UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

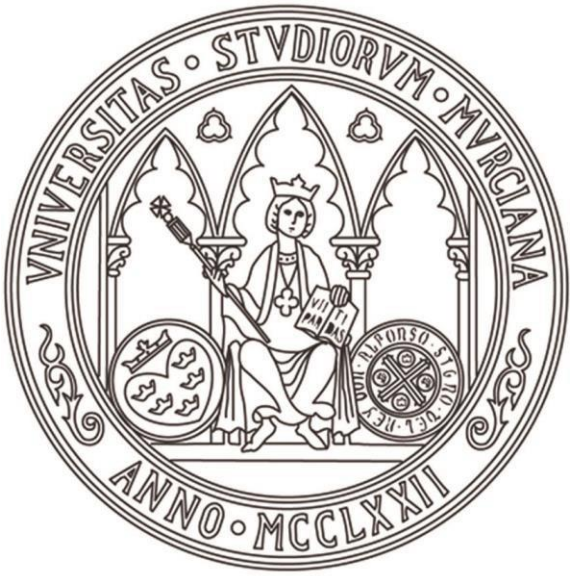
TESIS DOCTORAL

SPLITTING ALGORITHMS FOR STRUCTURED OPTIMIZATION: THEORY AND APPLICATIONS

ALGORITMOS DE DESGLOSE PARA OPTIMIZACIÓN ESTRUCTURADA: TEORÍA Y APLICACIONES

D. DAVID TORREGROSA BELÉN

2024



UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

SPLITTING ALGORITHMS FOR STRUCTURED OPTIMIZATION: THEORY AND APPLICATIONS

ALGORITMOS DE DESGLOSE PARA OPTIMIZACIÓN ESTRUCTURADA: TEORÍA Y APLICACIONES

Autor: D. David Torregrosa Belén

Director/es: D. Francisco J. Aragón Artacho

UNIVERSIDAD DE MURCIA

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD
DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR
Aprobado por la Comisión General de Doctorado el 19-10-2022

D./Dña. David Torregrosa Belén

doctorando del Programa de Doctorado en

Matemáticas

de la Escuela Internacional de Doctorado de la Universidad Murcia, como autor/a de la tesis presentada para la obtención del título de Doctor y titulada:

Algoritmos de Desglose para Optimización Estructurada: Teoría y Aplicaciones

y dirigida por,

D./Dña. Francisco Javier Aragón Artacho

D./Dña.

D./Dña.

DECLARO QUE:

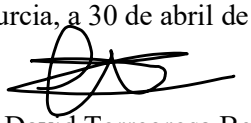
La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita, cuando se han utilizado sus resultados o publicaciones.

Si la tesis hubiera sido autorizada como tesis por compendio de publicaciones o incluyese 1 o 2 publicaciones (como prevé el artículo 29.8 del reglamento), declarar que cuenta con:

- *La aceptación por escrito de los coautores de las publicaciones de que el doctorando las presente como parte de la tesis.*
- *En su caso, la renuncia por escrito de los coautores no doctores de dichos trabajos a presentarlos como parte de otras tesis doctorales en la Universidad de Murcia o en cualquier otra universidad.*

Del mismo modo, asumo ante la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada, en caso de plagio, de conformidad con el ordenamiento jurídico vigente.

En Murcia, a 30 de abril de 2024



Fdo.: David Torregrosa Belén

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en la primera página de la tesis presentada para la obtención del título de Doctor.

Información básica sobre protección de sus datos personales aportados	
Responsable:	Universidad de Murcia. Avenida teniente Flomesta, 5. Edificio de la Convalecencia. 30003; Murcia. Delegado de Protección de Datos: dpd@um.es
Legitimación:	La Universidad de Murcia se encuentra legitimada para el tratamiento de sus datos por ser necesario para el cumplimiento de una obligación legal aplicable al responsable del tratamiento. art. 6.1.c) del Reglamento General de Protección de Datos
Finalidad:	Gestionar su declaración de autoría y originalidad
Destinatarios:	No se prevén comunicaciones de datos
Derechos:	Los interesados pueden ejercer sus derechos de acceso, rectificación, cancelación, oposición, limitación del tratamiento, olvido y portabilidad a través del procedimiento establecido a tal efecto en el Registro Electrónico o mediante la presentación de la correspondiente solicitud en las Oficinas de Asistencia en Materia de Registro de la Universidad de Murcia

Los resultados de esta tesis se enmarcan dentro del proyecto de investigación PGC2018-097960-B-C22 (*Contribuciones al análisis variacional: fundamentos, dualidad, robustez y algoritmos*) del Ministerio de Ciencia, Innovación y Universidades (MICIN), cofinanciado por el Fondo Europeo de Desarrollo Regional de la Unión Europea (UE). Dicho proyecto se ha desarrollado en el Departamento de Matemáticas de la Universidad de Alicante. Durante este periodo, el autor ha disfrutado de la ayuda predoctoral PRE2019-090751, dentro del programa "Ayudas para contratos predoctorales para la formación de doctores 2019" del MICIN, cofinanciada por el Fondo Social Europeo de la UE.

This dissertation was partially supported by the research project PGC2018-097960-B-C22 (*Contributions to variational analysis: fundamentals, duality, robustness and algorithms*) by the Ministry of Science, Innovation and University of Spain (MICIN) and the European Regional Development Fund of the European Union (EU). This project was carried out in the Department of Mathematics at the University of Alicante. The author was additionally supported by MICIN of Spain and the European Social Fund of EU (grant PRE2019-090751), under the program "Ayudas para contratos predoctorales para la formación de doctores 2019".

Agradecimientos
Acknowledgements

I cannot start without expressing my immense appreciation to my PhD supervisor, Francisco J. Aragón-Artacho. Working with him has been the most gratifying experience I have had during my academic career. Fran is undoubtedly the person who has taught me the most lessons, both in mathematics and as a professional, over the last four years, always transmitting his passion for what he does. Thank you, Fran, for your closeness, your unconditional support, and for making my experience of working with you enjoyable throughout this period. Above all, thank you for instilling in me the feeling that I could always count on you for anything. It is for all of this that you will always stand as a model to me as both a researcher and a person.

I am truly grateful to Walaa Moursi and Henry Wolkowicz for their hospitality and support during my stay at the University of Waterloo. They welcomed me with open arms as one more of their students, and it is thanks to them that I spent a beautiful eight months in Canada.

I would like to thank Radu Boţ for his sincere kindness throughout my stay at the University of Vienna, as well as on the occasions I have had the luck to encounter him.

Many thanks to Pedro Pérez-Aros for hosting me at the Universidad de O'Higgins. I am tremendously grateful for his keen interest in supervising my postdoctoral stage at the CMM. A special thank you for his patience during discussions about generalized subdifferentials!

The completion of this thesis also owes to the contributions of my coauthors. Thanks to Yair Censor, Aviv Gibali, Yura Malitsky and Matthew Tam for their kind collaboration.

Me siento muy afortunado de haber llevado a cabo mi etapa predoctoral en el Departamento de Matemáticas de la Universidad de Alicante. Les doy las gracias a todos sus miembros, que me han tratado estupendamente durante estos cuatro años. En particular, quiero agradecer a los integrantes del Laboratorio de Optimización por su ayuda en toda ocasión y por haberme acogido como uno más del grupo, ya sea en la universidad o en

los buenos momentos en los viajes a congresos. Gracias a Marco López y Miguel Ángel Goberna por sus consejos sobre la carrera investigadora, incluso ya desde que cursaba los estudios de grado. Un especial agradecimiento a Rubén Campoy por siempre animarme frente a los distintos retos a los que se enfrenta un joven investigador. Gracias a Mohammed, con el que he compartido la experiencia de ser doctorando, por el tiempo como compañeros de despacho y los agradables momentos comiendo en la universidad.

Muchas gracias a mi tutor en la Universidad de Murcia, Matías Raja, por su amabilidad y siempre haber estado disponible cuando lo he necesitado. Quisiera acordarme de José Bonet, quien me animó a emprender el camino de matemático cuando estaba cursando el grado de arquitectura en la Universidad Politécnica de Valencia. También de Isabel Fernández, que me dirigió y apoyó durante mis estudios de máster en la Universidad de Sevilla.

Por suerte muchos amigos me han acompañado durante este (a veces duro) proceso. Me siento especialmente contento de haber conocido a algunos de ellos durante el camino. Gracias a Andrés y David por ser mis guías durante los paseos por Viena. I will always cherish the great moments I spent during the harsh Canadian winter with Lia, Mengmeng and Leo. I feel so lucky to count you among my closest friends. Gracias también a mis amigos de siempre. Especialmente a Andreu, José, Sergio y Víctor, por siempre estar ahí para soportar mis quejas sobre la tesis, sus incansables esfuerzos por hacer que me despegara de la pantalla del ordenador y ayudarme a desconectar del trabajo.

Por último agradecer a mi familia. A mis hermanas Julia y Fátima, por nuestra particular relación de (más) amor (que) odio y por ser mis compañeras de muchas otras cosas. A mi primo Pablo, por ser la persona que ha aguantado mis quejas más matemáticas durante los viajes en coche a la universidad o al gimnasio. A mi abuela María Dolores, por haber sido como una segunda madre para mí durante tantos años. Quiero acordarme de mi abuelo Jero y mi abuela Mari Carmen que, aunque ya no estén, sé que estarían orgullosos de este momento. Y de Chuck, que es fundamental para mí, por ser mi compañero de excursiones y siempre transmitirme su felicidad. Finalmente, quiero agradecer a las dos personas más importantes para mí, mis padres. Gracias por siempre estar ahí para lo que necesitara y haberme apoyado en todo lo que me he propuesto. Es gracias a vuestro amor y dedicación que he logrado lo que soy hoy en día. Os quiero mucho...

David Torregrosa Belén
Alicante, 2024

Contents

Resumen

El leitmtotiv de la optimización matemática consiste en la búsqueda de un mejor elemento con respecto a un determinado criterio. Una de las manera más simple de describir este proceso es por medio del problema de minimización

$$\min_{x \in \mathcal{H}} \varphi(x),$$

donde $\mathcal{H}$ es un espacio de Hilbert real mientras que $\varphi : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ es una función con imagen sobre la recta real extendida que refleja las condiciones a optimizar. Los avances actuales en tecnología y el incremento de información disponible hacen que los problemas de optimización aumenten progresivamente en tamaño y complejidad. Una correcta aproximación a su tratamiento numérico precisa de un estudio cuidadoso de los datos de partida. En otras palabras, es fundamental ser capaz de sacar provecho de la estructura matemática de la función φ. Siguiendo la estrategia de divide y vencerás, los *algoritmos de desglose* se especializan en abordar programas matemáticos a través de la resolución iterativa de tareas simples, las cuales se definen empleando partes del problema original de manera independiente. Esto ha hecho que esta clase de algoritmos se consolide como una de las más fructíferas en el área de la optimización numérica moderna.

Esta tesis contribuye a la teoría de los algoritmos de desglose para optimización convexa y no convexa, campos significativamente dispares. Nuestras aportaciones se presentan en dos partes claramente diferenciadas, pero que comparten un mismo objetivo común: mejorar la eficiencia de los procesos computacionales de los algoritmos. Cada uno de los programas matemáticos abordados a lo largo de la tesis requerirá de un enfoque específico para la consecución de dicha meta. Las diferentes herramientas empleadas en el análisis de las técnicas numéricas para la resolución de estos problemas delineará la organización de la tesis. Asimismo, la eficacia de nuestros desarrollos teóricos se ilustrará en experimentos numéricos con datos, tanto sintéticos como reales, que surgen en diversas aplicaciones, como recuperación de imágenes, localización de instalaciones, computación distribuida y planificación de tratamientos de radioterapia de intensidad modulada.

Los Capítulos 1 y 2 están dedicados respectivamente a presentar una breve introducción así como notación básica y resultados preliminares que serán requeridos más adelante. Destacamos la última sección del Capítulo 2, que pretende ser un escueto primer contacto con el papel de la programación matemática en el modelado y resolución de problemas que surgen en el campo del tratamiento de imágenes. El resto de capítulos de la tesis se distribuyen en dos partes claramente diferenciadas.

En la primera parte de la tesis nos focalizamos en los comúnmente conocidos como *algoritmos de desglose para operadores monótonos*. Diremos que un operador multi-valuado $A : \mathcal{H} \rightrightarrows \mathcal{H}$ es *monótono* si verifica la desigualdad

$$\langle x - y, u - v \rangle \geq 0,$$

para todo (x, u) e (y, v) pertenecientes al grafo de A, y donde $\langle \cdot, \cdot \rangle$ representa el producto interior de $\mathcal{H}$. El problema de *inclusión monótona*, que consiste en hallar un *cero* de un operador monótono, es decir,

$$\text{encontrar } x \in \mathcal{H} \text{ tal que } 0 \in A(x),$$

permite modelar una gran variedad de situaciones que surgen en optimización convexa. Esto engloba la minimización de sumas de funciones convexas, problemas de factibilidad, problemas de punto de silla y desigualdades variacionales, entre muchas otras. En la Sección 2.2 detallamos una primera conexión entre la optimización convexa y los operadores monótonos. Además, introducimos herramientas fundamentales para el análisis de los algoritmos de desglose para operadores monótonos, como son los *resolventes*, los *operadores no-expansivos* y algunos resultados de *teoría de punto fijo*. También presentamos una selección de fórmulas cerradas de algunos resolventes que aparecerán recurrentemente a lo largo de la tesis.

En las últimas décadas, un gran número de métodos de desglose han sido diseñados para abordar un amplio rango de inclusiones monótonas. El objetivo del Capítulo 3 es iniciar al lector en la teoría de métodos de desglose para operadores monótonos y establecer una base común para los desarrollos presentados en los capítulos siguientes. En concreto, recopilamos las inclusiones monótonas más relevantes, destacamos sus aplicaciones en optimización y discutimos las propiedades de convergencia de los algoritmos más populares para su resolución. Entre otros, introducimos los algoritmos de *Douglas–Rachford*, *Davis–Yin*, *Chambolle–Pock* y *Tseng*.

El tema central alrededor del cual gira la primera parte de la tesis es presentado en

el Capítulo 4. Más concretamente, definimos con rigurosidad el concepto de dimensión del espacio subyacente al algoritmo, que denotaremos como *lifting*. A pesar de que los algoritmos de desglose para operadores monótonos han sido minuciosamente estudiados en la literatura, una anomalía común ha persistido en el diseño de algoritmos en esta familia: su *lifting* crece atípicamente conforme aumenta el tamaño del problema. Esto afecta directamente al rendimiento del algoritmo debido al incremento de los requisitos de memoria. Como ejemplificación de este hecho, considérese que estamos aplicando un algoritmo de desglose para la recuperación de una imagen borrosa con una resolución de 1000×1000 píxeles (un tamaño razonable). Matemáticamente esta imagen se representa a través de una matriz en el espacio $\mathbb{R}^{1000\times1000}$, la cual tiene un total de $n = 1\,000\,000$ entradas. Además, supongamos que el *lifting* de nuestro algoritmo viene dado por un número natural d. Esto implicaría que en cada iteración del algoritmo estamos arrastrando un total de $d \times n$ variables que deben ser almacenadas temporalmente en el ordenador. Reducir el *lifting* del algoritmo es por tanto sumamente recomendable, especialmente si el número n de variables del problema es grande.

En el Capítulo 4 introducimos un marco general para el estudio del *lifting* de los métodos de desglose para operadores monótonos y recopilamos los teoremas hasta ahora existentes que caracterizan el *lifting mínimo* que un método puede tener bajo ciertas condiciones. En particular, nuestras nuevas aportaciones teóricas son una generalización del *teorema de lifting mínimo de Malitsky–Tam* que engloba el uso de parámetros de tamaño de paso en los algoritmos de desglose, lo que aumenta considerablemente la gama de métodos que verifican las hipótesis del teorema, así como el establecimiento del primer resultado de *lifting* mínimo para *algoritmos de desglose primales-duales*.

El Capítulo 5 continene una demostración novedosa de la convergencia del algoritmo de desglose de Davis–Yin que permite duplicar el rango de valores admitidos del parámetro de tamaño de paso del algoritmo. Aunque la discusión sobre el *lifting* mínimo del método de Davis–Yin no tiene lugar en este capítulo, la prueba propuesta recoge la esencia del análisis de convergencia del método de desglose con *lifting* mínimo desarrollado en el Capítulo 6.

Esta primera parte concluye con los Capítulos 6 y 7, donde presentamos cuatro nuevos algoritmos con *lifting* mínimo (o reducido) para la resolución de los problemas de inclusión monótona discutidos en el Capítulo 3. En concreto, en el Capítulo 6 desarrollamos un algoritmo con *lifting* mínimo para la resolución de inclusiones monótonas conformadas por la suma de operadores *monótonos maximales* y *cocoercivos*. A su vez, la inclusión de unos términos de *tipo reflexión* nos permite diseñar una modificación del algoritmo adecuada

para abordar inclusiones monótonas donde la hipótesis de cocoercividad es remplazada por una menos exigente, esto es, los operadores pueden ser monótonos y *Lipschitz continuos*. En el Capítulo 7 introducimos el único método de desglose primal-dual existente para la resolución de *inclusiones monótonas compuestas*, las cuales involucran composiciones de operadores monónotonos maximales con operadores lineales. Por último, coordinamos los tres métodos previos para desarrollar un esquema capaz de abordar combinaciones de las inclusiones monótonas anteriores. Experimentos numéricos ilustran las ventajas de los métodos propuestos con respecto a otros algoritmos con mayor *lifting*.

La primera parte de la tesis se fundamenta en los siguientes artículos publicados:

[10] Aragón-Artacho, F. J., Boţ, R. I., and Torregrosa-Belén, D. A primal-dual splitting algorithm for composite monotone inclusions with minimal lifting. *Numer. Algorithms 93*, 1 (2023), 103–130.

[25] Aragón-Artacho, F. J., Malitsky, Y., Tam, M. K., and Torregrosa-Belén, D. Distributed forward-backward methods for ring networks. *Comput. Optim. Appl. 86*, 3 (2023), 845–870.

[26] Aragón-Artacho, F. J., and Torregrosa-Belén, D. A direct proof of convergence of Davis–Yin splitting algorithm allowing larger stepsizes. *Set-Val. Var. Anal. 30*, 3 (2022), 1011–1029.

En la segunda parte de la tesis introducimos dos avances independientes en la teoría de los algoritmos de desglose distribuidos en dos capítulos. El marco necesario para el análisis matemático de estas contribuciones no se sustenta en la teoría de los operadores monótonos. Este hecho ha motivado la separación de la tesis en dos partes diferenciadas de acuerdo a las distintas metodologías empleadas en cada una de ellas.

En el Capítulo 8 nos trasladamos al dominio no convexo. Citando a Tyrell Rockafellar, uno de los padres de la optimización moderna, "*...de hecho, la gran línea divisoria en optimización no separa la linealidad y la no linealidad, sino la convexidad y la no convexidad*". En efecto, el análisis de algoritmos de desglose para problemas no convexos no se ha desarrollado en la misma medida que en el caso convexo, solo habiéndose conseguido resultados de convergencia para un número reducido de problemas. Aquí introducimos el algoritmo BDSA (*Boosted Double-proximal Subgradient Algorithm*), un nuevo método de desglose diseñado para abordar programas matemáticos que involucran estructuras no convexas y no diferenciables. BDSA explota la estructura de los datos del problema a través del uso combinado de *subgradientes* y la evaluación de *operadores de proximidad*.

Además, integra una búsqueda lineal al final de cada iteración con el objetivo de mejorar su rendimiento computacional. Mientras que BDSA engloba algoritmos ya existentes en la literatura, extiende su aplicación a configuraciones de problemas más diversas. Por medio de experimentos numéricos con aplicaciones al *problema de conglomerados de suma de cuadrados mínima* y una generalización del clásico *problema de Heron*, mostramos que la búsqueda lineal reduce considerablemente el número de iteraciones y el tiempo que BDSA necesita para converger en comparación con algoritmos con términos de inercia. Asimismo, presentamos dos nuevas familias de funciones *test* que ilustran la eficacia de BDSA para evitar *puntos críticos* no óptimos. Este capítulo se basa en el artículo sometido

[29] Aragón-Artacho, F. J., Pérez-Aros, P., and Torregrosa-Belén, D. The boosted double-proximal subgradient algorithm for nonconvex optimization. *Preprint, arXiv:2306.17144 [math.OC], 2023*.

Finalmente, en el Capítulo 9 consideramos el *problema de minimización dividido* que consta de dos subproblemas de minimización con restricciones en dos espacios distintos bajo un operador lineal que asigna un espacio al otro. Este es un problema complejo que surge en aplicaciones prácticas como la planificación de tratamientos de radioterapia de intensidad modulada. Con frecuencia, la complejidad de los datos de dicho problema hace que la obtención de una solución exacta suponga un gran desafío computacional. Para afrontar esta situación distinguimos dos objetivos de diferente importancia en el problema de minimización dividido: un objetivo mayor consistente en la verificación de la restricciones y un objetivo secundario, pero deseable, de reducción de las funciones objetivo. Seguidamente desarrollamos un enfoque basado en la metodología de la *superiorización*. Los algoritmos superiorizados son una clase de métodos "semi-heurísticos" que se basan en entrelazar los pasos de dos procesos iterativos separados e independientes, perturbando las iteraciones de un proceso de acuerdo con los pasos dictados por el otro proceso. En este caso, el esquema de nuestro algoritmo se rige por un método de proyección, que asegura la convergencia a un punto factible, mientras que escogemos las perturbaciones de acuerdo a la meta secundaria de reducir (no minimizar) las funciones objetivo presentes en el problema. De esta manera sacrificamos el objetivo de minimización global a cambio de conseguir una mayor eficiencia numérica. Además, incluimos dos elementos novedosos en la metodología de la superiorización. El primero es la posibilidad de reiniciar las perturbaciones en el algoritmo superiorizado, lo que resulta en una aceleración significativa y aumenta el rendimiento del método. El segundo elemento es la capacidad de superiorizar subvectores de forma independiente. Las contribuciones presentes en este capítulo fueron publicadas en primer lugar en

[23] Aragón-Artacho, F. J., Censor, Y., Gibali, A., and Torregrosa-Belén, D. The superiorization method with restarted perturbations for split minimization problems with an application to radiotherapy treatment planning. *Appl. Math. Comput. 440* (2023), Paper No. 127627, 17.

Abstract

With modern advances in technologies and quantity of information, optimization problems become increasingly large in size and complexity. Successfully handling these ever evolving programs requires a careful processing of the available data, namely, taking advantage of the inherent mathematical structure of the problem. Following the divide-and-conquer paradigm, *splitting algorithms* specialize in tackling mathematical programs by iteratively solving simpler subtasks, which are defined by separately using some parts of the original problem. This has led this class of algorithms to emerge as one of the most fruitful among modern numerical optimization methods.

This thesis contributes to the theory of splitting algorithms for both convex and non-convex optimization. Our contributions are presented in two clearly differentiated parts, but which share a common core objective: to enhance the efficiency of the algorithms' computational processes. This goal is achieved through distinct approaches tailored to each specific mathematical program faced throughout the thesis. In addition, the effectiveness of our theoretical developments is validated in numerical experiments with synthetic and realistic data arising in multiple real-world applications, such as image recovery, facility location and intensity-modulated radiation therapy treatment planning.

In Part I, we concentrate on the so-called *monotone operator splitting methods*. These algorithms are employed for solving *monotone inclusions*, a problem which models several situations in convex optimization. Although these methods have been thoroughly studied in the last decades, a common anomaly has persisted in the design of algorithms in this family: the dimension of the underlying space —which we denote as *lifting*— of the algorithms abnormally increases as the problem size grows. This has direct implications on the computational performance of the methods as a result of the escalation of memory requirements. In this framework, we characterize the minimal lifting that can be obtained by splitting algorithms adept at solving certain general monotone inclusions. Moreover, we pioneer the development of splitting methods matching these lifting bounds, and thus having minimal lifting. The analysis developed in this context also leads to a new proof of convergence of the popular *Davis–Yin splitting algorithm* which allows to double the range of admitted stepsize parameter values.

Two independent advances to the family of splitting algorithms are presented in Part II. In Chapter 8, we move to the nonconvex realm. The analysis of splitting methods for nonconvex problems has not been developed to the same extent as in the convex setting, with convergence guarantees only given for some restricted problem structures. We introduce the *Boosted Double-proximal Subgradient Algorithm* (BDSA), a novel splitting algorithm designed to address general structured nonsmooth and nonconvex mathematical programs. While BDSA encompasses existing schemes in the literature, it extends its applicability to more diverse problem domains. One of the features of BDSA, which differentiates it from previously proposed methods, is the integration of a linesearch procedure to enhance (or "*boost*") its performance. Numerical experience reveals that this linesearch considerably reduces both the number of iterations and the time that BDSA needs to converge in comparison with algorithms including inertial terms. In addition, we introduce two new families of test functions to illustrate BDSA's ability to effectively escape non-optimal critical points.

Finally, in Chapter 9, we study the *split minimization problem* that consists of two constrained minimization problems in two separate spaces that are connected via a linear operator that maps one space into the other. To handle the data of such a problem, we develop a *superiorization* approach that can reach a feasible point with reduced (not necessarily minimal) objective function values. The superiorization methodology is based on interlacing the iterative steps of two separate and independent iterative processes by perturbing the iterates of one process according to the steps dictated by the other. Further, we include in our developed method two novel elements. The first one is the permission to restart the perturbations in the superiorized algorithm, which results in a significant acceleration and increases the computational efficiency. The second element is the ability to independently superiorize subvectors.

Chapter 1

Introduction: A guide for the reader

The leitmotiv of mathematical optimization consists in the search for a best element with respect to some criterion. The simplest way to mathematically describe this is by means of the minimization problem

$$\min_{x \in \mathcal{H}} \varphi(x), \tag{1.1}$$

where $\mathcal{H}$ is a real Hilbert space and the extended real-valued function $\varphi : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ captures the conditions to optimize. The modern advances in technologies and quantity of information lead to optimization problems becoming increasingly large in size and complexity. Successfully handling these ever evolving problems requires a careful processing of the available data, namely, taking advantage of the mathematical structure of the function φ. Following the divide-and-conquer paradigm, *splitting algorithms* specialize in tackling mathematical programs by iteratively solving simpler subtasks, which are defined by separately using some parts of the original problem. This has led this class of algorithms to emerge as one of the most fruitful among modern numerical optimization methods.

Throughout this thesis, we shall investigate splitting algorithms applicable to a wide spectrum of problems in the form of (1.1), encompassed in both convex and nonconvex optimization. The different tools required for analyzing numerical approaches to each of such programs will delineate the organization of the thesis. Nevertheless, a common central objective underlies and motivates the developments furnished in this dissertation: the enhancement of the efficiency of splitting algorithm's computational processes. This goal will be achieved through distinct approaches tailored to each specific problem setting.

The purpose of this introductory chapter is to provide an overview of the contents and organization of the thesis. Furthermore, each chapter begins with a brief motivation and listing of the main achievements contained therein. The reader is referred to those sections for further specific details of each chapter.

Besides Chapter 2, which introduces basic notation and preliminary results, the thesis is divided into two clearly differentiated parts.

1.1 Overview of Part I

In Part I, we concentrate on the subclass of *monotone operator splitting algorithms*. A set-valued operator $A : \mathcal{H} \rightrightarrows \mathcal{H}$ is said to be *monotone* if

$$\langle x - y, u - v \rangle \geq 0,$$

for all (x, u) and (y, v) belonging to the graph of A. The *monotone inclusion problem* consisting in finding a *zero* of a monotone operator, i.e.,

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in A(x), \tag{1.2}$$

can be used to model a great variety of situations in convex optimization. This encompasses the minimization of sums of convex functions, feasibility problems, and many more. A first connection between convex optimization and monotone operators is detailed in Section 2.2.

In the last five decades, a large number of splitting methods have been devised to address a broad range of inclusions in the form of (1.2), which are determined by the specific structure of A. Chapter 3 aims to introduce the reader to the family of monotone operator splitting methods and to establish a common basis for the developments presented in subsequent chapters. Therein, we survey relevant monotone inclusion problems, emphasizing their applications to optimization and introducing the most popular splitting methods for their resolution.

The central theme on which this part of the thesis revolves around is rigorously defined in Chapter 4. More concretely, the dimension of the underlying space of an algorithm, also called *lifting*, which has a big influence on its computational performance. In this chapter, we describe a unifying framework for analyzing the lifting of monotone operator splitting algorithms and present the known theorems characterizing the *minimal lifting* that a method can achieve under certain conditions.

Chapter 5 contains a new proof of convergence of the *Davis–Yin splitting algorithm* which allows to double the range of admitted stepsize parameter values. Although the discussion on the minimal lifting of the Davis–Yin method is not conducted there, the

newly proposed proof gathers the essence of the convergence analysis of the minimal lifting method devised in Chapter 6.

This part concludes with Chapters 6 and 7, where we introduce four new algorithms with minimal (or reduced) lifting specialized in solving the monotone inclusion problems described in Chapter 3. Numerical experiments illustrate the advantages of the proposed algorithms with respect to other methods with higher lifting.

Part I of the thesis is founded on the following published papers:

[10] Aragón-Artacho, F. J., Boţ, R. I., and Torregrosa-Belén, D. A primal-dual splitting algorithm for composite monotone inclusions with minimal lifting. *Numer. Algorithms 93*, 1 (2023), 103–130.

[25] Aragón-Artacho, F. J., Malitsky, Y., Tam, M. K., and Torregrosa-Belén, D. Distributed forward-backward methods for ring networks. *Comput. Optim. Appl. 86*, 3 (2023), 845–870.

[26] Aragón-Artacho, F. J., and Torregrosa-Belén, D. A direct proof of convergence of Davis–Yin splitting algorithm allowing larger stepsizes. *Set-Val. Var. Anal. 30*, 3 (2022), 1011–1029.

1.2 Overview of Part II

This part presents two independent advances to the theory of general splitting algorithms, which are distributed in two different chapters. The necessary setting for the analysis of these contributions cannot be drawn upon the framework established by monotone operators. This is what led us to split the thesis into two separate parts according to the disparate methodology employed.

Rather than including here a detailed motivation of the topics to discuss in Part II, we refer the reader to the introductions of Chapters 8 and 9. In what follows, we highlight the main contributions of each chapter.

Chapter 8 introduces the novel *Boosted Double-proximal Subgradient Algorithm* (abbreviated as BDSA) for structured nonsmooth and nonconvex mathematical programs. BDSA exploits the combined nature of subgradients from the data and proximal steps, and integrates a linesearch procedure to enhance its performance. To evaluate the effectiveness of BDSA, we introduce two novel test functions with an abundance of critical points. We conduct comparative evaluations, including algorithms with inertial terms,

that illustrate its ability to effectively escape non-optimal critical points. Additionally, we present two practical applications of BDSA for testing its efficacy, namely, a constrained minimum-sum-of-squares clustering problem and a nonconvex generalization of Heron's problem. This chapter is based on the submitted work

[29] Aragón-Artacho, F. J., Pérez-Aros, P., and Torregrosa-Belén, D. The boosted double-proximal subgradient algorithm for nonconvex optimization. *Preprint, arXiv:2306.17144 [math.OC], 2023*.

Chapter 9 proposes an algorithm based on the *superiorization methodology* for tackling a class of constrained optimization problem, with applications in intensity-modulated radiation therapy treatment planning. This algorithm provides an efficient way of handling the data from this challenging problem, for which it is computationally difficult, or sometimes even impossible, to obtain exact solutions by standard splitting methods. Superiorization algorithms are a class of "semi-heuristic" methods that interlace feasibility seeking steps with the inclusion of certain perturbations which aim to reduce (not minimize) a target function while ensuring constraint satisfiability. Hence, in the pursuit of greater numerical efficiency, the objective of global minimization is sacrificed. Two new elements are included. The first is a permission to restart the perturbations in the superiorized algorithm, which increases the computational efficiency. The second is the ability to superiorize independently over subvectors. The developments in this chapter were first published in

[23] Aragón-Artacho, F. J., Censor, Y., Gibali, A., and Torregrosa-Belén, D. The superiorization method with restarted perturbations for split minimization problems with an application to radiotherapy treatment planning. *Appl. Math. Comput. 440* (2023), Paper No. 127627, 17.

Chapter 2

Preliminaries

The purpose of the chapter is to introduce the basic notation and preliminary results that will be needed throughout the thesis. In Section 2.1, we fix the nomenclature. Section 2.2 establishes a first bond between optimization and monotone operators. This section contains some key results from the areas of convex analysis, fixed point theory and monotone operator theory. Section 2.3 provides closed-formulas for some proximal point and projection mappings that will be employed in the thesis. In subsequent chapters we shall test algorithm performance by conducting numerical experiments in image processing. We devote Section 2.4 to illustrate the mathematical formulation of problems in this field.

2.1 Notational conventions

Throughout this thesis, $\mathcal{H}$ and $\mathcal{G}$ are *real Hilbert spaces*. Otherwise stated, to simplify the notation, we will employ $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ to denote the inner product and the induced norm, respectively, of any space. We use $\to$ to denote *norm convergence* of a sequence, while $\rightharpoonup$ refers to *weak convergence*. We denote by $\mathcal{H}^n$ the *product Hilbert space* $\mathcal{H}^n := \mathcal{H} \times \overset{(n)}{\cdots} \times \mathcal{H}$ with inner product defined as

$$\langle (x_1, \ldots, x_n), (y_1, \ldots, y_n) \rangle := \sum_{i=1}^{n} \langle x_i, y_i \rangle, \quad \forall (x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \mathcal{H}^n.$$

Sequences and sets in product spaces are marked with bold, e.g., $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{H}^n$. We use $\mathbb{B}_\varepsilon(\bar{x})$ to denote the closed ball of radius $\varepsilon > 0$ centered at $\bar{x}$, namely,

$$\mathbb{B}_\varepsilon(\bar{x}) := \{x \in \mathcal{H} \,:\, \|x - \bar{x}\| \leq \varepsilon\}.$$

For a *set-valued operator*, we write $A : \mathcal{H} \rightrightarrows \mathcal{H}$, in opposition to $A : \mathcal{H} \to \mathcal{H}$, which

denotes a *single-valued operator*. The notation dom, Fix, zer, ran and gra is used for the *domain*, the set of *fixed point*, the *zeros*, the *range* and the *graph* of A, respectively, i.e.,

$$\begin{aligned} \operatorname{dom} A &:= \{x \in \mathcal{H} : A(x) \neq \emptyset\}, & \operatorname{ran} A &:= \{u \in \mathcal{H} : \exists x \in \mathcal{H} : u \in A(x)\}, \\ \operatorname{gra} A &:= \{(x,u) \in \mathcal{H} \times \mathcal{H} : u \in A(x)\}, & \operatorname{Fix} A &:= \{x \in \mathcal{H} : x \in A(x)\}, \\ & \text{and} & \operatorname{zer} A &:= \{x \in \mathcal{H} : 0 \in A(x)\}. \end{aligned}$$

The *inverse operator* of A, denoted by A^{-1}, is the operator whose graph is given by $\operatorname{gra} A^{-1} = \{(u,x) \in \mathcal{H} \times \mathcal{H} : u \in A(x)\}$. The *identity operator* is denoted by Id. When $L : \mathcal{H} \to \mathcal{G}$ is a bounded linear operator, we use $L^* : \mathcal{G} \to \mathcal{H}$ to denote its *adjoint*, which is the unique bounded linear operator such that $\langle Lx, y\rangle = \langle x, L^*y\rangle$, for all $x \in \mathcal{H}$ and $y \in \mathcal{G}$.

We recall that a set $C \subseteq \mathcal{H}$ is said to be *convex* if for any $x, y \in C$,

$$(1-\alpha)x + \alpha y \in C, \quad \forall \alpha \in]0,1[.$$

For an arbitrary set $C \subseteq \mathcal{H}$, we use the notation span to denote its *span* and co for its *convex hull*, which are defined as

$$\begin{aligned} \operatorname{span} C &:= \left\{ \sum_{i=1}^{n} \lambda_i x_i \;:\; n \in \{1,2,\ldots\},\, x_i \in C,\, \lambda_i > 0 \right\}, \\ \operatorname{co} C &:= \left\{ \sum_{i=1}^{n} \lambda_i x_i \;:\; n \in \{1,2,\ldots\},\, x_i \in C,\, \lambda_i \in]0,1],\, \sum_{i=1}^{n} \lambda_i = 1 \right\}. \end{aligned}$$

The span of C coincides with the smallest linear subspace of $\mathcal{H}$ containing C, its closure is the smallest closed linear subspace of $\mathcal{H}$ that contains C and is denoted by $\overline{\operatorname{span}} C$. A set $K \subseteq \mathcal{H}$ is said to be a *cone* if $K = \mathbb{R}_{++}K$, where $R_{++} = \{\lambda : \lambda > 0\}$. The *conicall hull* of a set $C \subseteq \mathcal{H}$ is the intersection of all the cones in $\mathcal{H}$ containing C. It is denoted by $\operatorname{cone} C$.

If C is a convex set, its *relative interior* and *strong relative interior* are denoted as ri and sri, respectively:

$$\begin{aligned} \operatorname{ri} C &:= \{x \in C \;:\; \operatorname{cone}(C-x) = \operatorname{span}(C-x)\}, \\ \operatorname{sri} C &:= \{x \in C \;:\; \operatorname{cone}(C-x) = \overline{\operatorname{span}}(C-x)\}. \end{aligned}$$

The symbol $\mathbb{N}$ is used for denoting the set of nonnegative integers, i.e., $\mathbb{N} := \{0,1,2,\ldots\}$.

We use $\mathbb{R}^n$ to denote the Euclidean space of dimension n, while we define the extended-real-valued line by $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ and adopt the convention $1/0 = +\infty$. The *domain* of a function $f : \mathcal{H} \to \overline{\mathbb{R}}$ is defined as $\operatorname{dom} f := \{x \in \mathcal{H} \,:\, f(x) < +\infty\}$. We say that f is *proper* if it does not attain the value $-\infty$ and $\operatorname{dom} f \neq \emptyset$. The function f is said to be *lower-semicontinuous (l.s.c.)* at some point $\bar{x} \in \mathcal{H}$ if $\liminf_{x \to \bar{x}} f(x) \geq f(\bar{x})$.

A function $f : \mathcal{H} \to \overline{\mathbb{R}}$ is *convex* if for any $x, y \in \operatorname{dom} f$

$$f((1-\alpha)x + \alpha y) \leq (1-\alpha)f(x) + \alpha f(y), \quad \forall \alpha \in]0,1[.$$

The function $f : \mathcal{H} \to \mathbb{R}$ is said to be (*Fréchet*) *differentiable* at a point $\bar{x} \in \mathcal{H}$ if there exists $\nabla f(\bar{x}) \in \mathcal{H}$ such that

$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \langle \nabla f(\bar{x}), x - \bar{x} \rangle}{\|x - \bar{x}\|} = 0.$$

In this case $\nabla f(\bar{x})$ is called the (*Fréchet*) *gradient* of f at $\bar{x}$.

Finally, to simplify the notation we will usually employ the compact expression $[\![k, l]\!]$ to denote the set of integers between $k, l \in \mathbb{N}$, i.e.,

$$[\![k, l]\!] := \begin{cases} \{k, k+1, \ldots, l\}, & \text{if } k \leq l, \\ \emptyset, & \text{otherwise.} \end{cases}$$

2.2 Convex analysis and monotone operator theory

This section contains a brief presentation of the basic concepts required from three interconnected areas: the theory of nonexpansive mappings, monotone operator theory and convex optimization. Our exposition is mainly based on the book by Bauschke and Combettes [44], which is an indispensable reference in this field. The recent book of Bauschke and Moursi [52] contains an accessible introduction to convex analysis and optimization, so we strongly recommend it to people who wish to get started in the area of splitting algorithms.

2.2.1 Nonexpansiveness and fixed point theorems

The analysis of many numerical methods in optimization relies on the use of tools from fixed point theory. A considerable number of the algorithms studied in this thesis can be

described as a *fixed point iteration* of a single-valued operator $T : \mathcal{H} \to \mathcal{H}$ given by

$$x^{k+1} = T(x^k), \quad \forall k \in \mathbb{N}. \tag{2.1}$$

The following definition gathers some properties that are crucial in the convergence study of the above sequence.

Definition 2.1. *An operator $T : \mathcal{H} \to \mathcal{H}$ is said to be*

(i) β-Lipschitz continuous *for $\beta > 0$ if*

$$\|T(x) - T(y)\| \leq \beta\|x - y\|, \quad \forall x, y \in \mathcal{H};$$

(ii) nonexpansive *if it is 1-Lipschitz continuous, i.e.,*

$$\|T(x) - T(y)\| \leq \|x - y\|, \quad \forall x, y \in \mathcal{H};$$

(iii) quasi-nonexpansive *if*

$$\|T(x) - y\| \leq \|x - y\|, \quad \forall x \in \mathcal{H}, \forall y \in \operatorname{Fix} T;$$

(iv) $\frac{1}{\beta}$-cocoercive *for $\beta > 0$ if*

$$\langle T(x) - T(y), x - y\rangle \geq \frac{1}{\beta}\|T(x) - T(y)\|^2, \quad \forall x, y \in \mathcal{H};$$

(v) α-averaged nonexpansive *for $\alpha \in]0, 1[$ if*

$$\|T(x) - T(y)\|^2 + \frac{1-\alpha}{\alpha}\|(\mathrm{Id} - T)(x) - (\mathrm{Id} - T)(y)\|^2 \leq \|x - y\|^2, \quad \forall x, y \in \mathcal{H};$$

(vi) $\frac{1-\alpha}{\alpha}$-strongly quasi-nonexpansive *for $\alpha \in]0, 1[$ if*

$$\|T(x) - y\|^2 + \frac{1-\alpha}{\alpha}\|(\mathrm{Id} - T)(x)\|^2 \leq \|x - y\|^2, \quad \forall x \in \mathcal{H}, \forall y \in \operatorname{Fix} T;$$

(vii) firmly nonexpansive *if*

$$\|T(x) - T(y)\|^2 + \|(\mathrm{Id} - T)(x) - (\mathrm{Id} - T)(y)\|^2 \leq \|x - y\|^2, \quad \forall x, y \in \mathcal{H}.$$

For brevity, we sometimes omit to mention the corresponding constant when referring to some of the above operators. For instance, we say averaged nonexpansive instead of α-averaged nonexpansive.

REMARK 2.2. The properties in Definition 2.1 are closely intertwined.
(a) The following implications hold: $(v) \Rightarrow (ii) \Rightarrow (iii)$ and $(v) \Rightarrow (vi) \Rightarrow (iii)$.
(b) Firmly nonexpansiveness equals α-averagedness with $\alpha = \frac{1}{2}$. Hence, any α-averaged operator with $\alpha \in \left]0, \frac{1}{2}\right]$ is firmly nonexpansive.
(c) By the Cauchy–Schwarz inequality, a $\frac{1}{\beta}$-cocoercive operator is β-Lipschitz continuous. The *Baillon–Haddad theorem* [37, Corolaire 10] states that the opposite inclusion also holds when the operator is the gradient of a convex function.
(d) By [44, Proposition 4.4], an operator T is $\frac{1}{\beta}$-cocoercive if and only if $\frac{1}{\beta}T$ is firmly nonexpansive.
(e) An equivalent definition of averaged nonexpansive operator is the following. An operator T is α-averaged, with $\alpha \in]0, 1[$, if and only if it is of the form

$$T = (1 - \alpha)\,\mathrm{Id} + \alpha R,$$

with R a nonexpansive operator.

The following characterization of firmly nonexpansive operators will be useful in the sequel. For further equivalent expressions we refer the reader to [44, Chapter 4].

Proposition 2.3 ([44, Proposition 4.4]). *An operator $T : \mathcal{H} \to \mathcal{H}$ is firmly nonexpansive if and only if*

$$0 \leq \langle T(x) - T(y), (\mathrm{Id} - T)(x) - (\mathrm{Id} - T)(y) \rangle, \quad \forall x, y \in \mathcal{H}.$$

The next result states that the sum of cocoercive operators is also cocoercive.

Proposition 2.4 ([44, Proposition 4.12]). *For every $i \in \{1, \ldots, n\}$, let $T_i : \mathcal{H} \to \mathcal{H}$ be a $\frac{1}{\beta_i}$-cocoercive operator with $\beta_i > 0$. Then the operator $T = \sum_{i=1}^{n} T_i$ is $\frac{1}{\beta}$-cocoercive with $\beta = \sum_{i=1}^{n} \beta_i$.*

Fejér monotonicity is a key property in fixed point theory, as it allows to derive weak convergence of iterative sequences (see, e.g, [44, Chapter 5]).

Definition 2.5 (Fejér monotone). *Let $C \subseteq \mathcal{H}$ be a nonempty set and let $(x^k)_{k \in \mathbb{N}}$ be a sequence in $\mathcal{H}$. Then $(x^k)_{k \in \mathbb{N}}$ is* Fejér monotone *with respect to C if, for all $x \in C$,*

$$\|x^{k+1} - x\| \leq \|x^k - x\|, \quad \forall k \in \mathbb{N}.$$

Proposition 2.6 ([44, Theorem 5.5]). *Let C be a nonempty subset of $\mathcal{H}$ and let $(x^k)_{k\in\mathbb{N}}$ be a sequence in $\mathcal{H}$. Suppose that $(x^k)_{k\in\mathbb{N}}$ is Fejér monotone with respect to C and that every weak sequential cluster point of $(x^k)_{k\in\mathbb{N}}$ belongs to C. Then $(x^k)_{k\in\mathbb{N}}$ converges weakly to a point in C.*

According to the widely-known Banach–Picard theorem, the iterative sequence $(x^k)_{k\in\mathbb{N}}$ defined in (2.1) converges strongly to a fixed point of T provided this operator is *contractive*, i.e., it is β-Lipschitz continuous with $\beta \in [0, 1[$. However, the contraction assumption is too restrictive as the operators that one encounters in practice are at most nonexpansive, namely, $\beta = 1$. When T is a nonexpansive operator, the fixed point iteration (2.1) is well-known to usually diverge (for instance, set $T = -\operatorname{Id}$). This situation can be overcome by considering a *Krasnosel'skiĭ–Mann iteration*, which might be regarded as a relaxed version of (2.1).

Theorem 2.7 (Krasnosel'skiĭ–Mann iteration, [44, Theorem 5.15]). *Given a nonexpansive operator $T : \mathcal{H} \to \mathcal{H}$ such that $\operatorname{Fix} T \neq \emptyset$, let $(\lambda_k)_{k\in\mathbb{N}}$ be a sequence in $[0, 1]$ such that $\sum_{k\in\mathbb{N}} \lambda_k(1 - \lambda_k) = +\infty$, and let $x^0 \in \mathcal{H}$. Set*

$$x^{k+1} = x^k + \lambda_k\big(T(x^k) - x^k\big), \quad \forall k \in \mathbb{N}.$$

Then the following hold.

(i) $(x^k)_{k\in\mathbb{N}}$ *is Fejér monotone with respect to* $\operatorname{Fix} T$.

(ii) $\big(T(x^k) - x^k\big)_{k\in\mathbb{N}}$ *converges strongly to* 0.

(iii) $(x^k)_{k\in\mathbb{N}}$ *converges weakly to a point in* $\operatorname{Fix} T$.

The following is a refined version of Theorem 2.7 for averaged nonexpansive operators that admits larger *relaxation parameters* λ_k.

Proposition 2.8 ([44, Proposition 5.16]). *Let $\alpha \in]0, 1[$ and let $T : \mathcal{H} \to \mathcal{H}$ be an α-averaged nonexpansive operator such that $\operatorname{Fix} T \neq \emptyset$. Let $(\lambda_k)_{k\in\mathbb{N}}$ be a sequence in $[0, \frac{1}{\alpha}]$ such that $\sum_{k\in\mathbb{N}} \lambda_k(1 - \alpha\lambda_k) = +\infty$. Given $x^0 \in \mathcal{H}$, set*

$$x^{k+1} = x^k + \lambda_k\big(T(x^k) - x^k\big), \quad \forall k \in \mathbb{N}.$$

Then the statements (i), (ii) and (iii) in Theorem 2.7 also hold.

REMARK 2.9. (a) In view of Remark 2.2 (e), assertions (i), (ii) and (iii) in Theorem 2.7 also hold for the sequence generated by the fixed point iteration (2.1) of an α-averaged nonexpansive operator by just setting $T := R$ and $\lambda_k := \alpha$, for all $k \in \mathbb{N}$.
(b) If T is a firmly nonexpansive operator in Proposition 2.8, a constant relaxation parameter can be chosen in the interval $]0, 2[$.

We say that a fixed point sequence $(x^k)_{k\in\mathbb{N}}$ defined by an operator T satisfies the *asymptotic regularity property* if

$$T(x^k) - x^k \to 0. \tag{2.2}$$

This property is fundamental for guaranteeing the weak convergence of the sequence generated by the Krasnosel'skiĭ–Mann iteration. In the sequel we shall encounter fixed point iterations defined by operators that only possess "one-sided" nonexpansive properties, such as strong quasi-nonexpansiveness. Theorem 2.7 and Proposition 2.8 are not applicable in this situation. The verification of (2.2) will be the first step in the convergence analysis of such sequences.

Finally, we say that a single-valued operator $T : \mathcal{H} \to \mathcal{H}$ is *demiregular* at $x \in \mathcal{H}$ if for all sequences $(x^k)_{k\in\mathbb{N}}$ with $x^k \rightharpoonup x$ and $T(x^k) \to T(x)$, we have that $x^k \to x$.

2.2.2 Monotone operator theory

Many problems arising in convex optimization can be described using monotone operators. In the following, we introduce some basic concepts about monotone operator theory.

Definition 2.10 (Monotone operator). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be a set-valued operator*

(i) A is said to be μ-monotone *for $\mu \in \mathbb{R}$ if*

$$\langle x - y, u - v\rangle \geq \mu\|x - y\|^2, \quad \forall (x, u), (y, v) \in \operatorname{gra} A.$$

Furthermore, A is said to be μ-maximally monotone *if there exists no μ-monotone operator $B : \mathcal{H} \rightrightarrows \mathcal{H}$ such that* $\operatorname{gra} B$ *properly contains* $\operatorname{gra} A$.

(ii) A is said to be uniformly monotone *with modulus $\phi : \mathbb{R}_+ \to [0, +\infty[$ if ϕ is increasing, vanishes only at 0, and*

$$\langle x - y, u - v\rangle \geq \phi(\|x - y\|), \quad \forall (x, u), (y, v) \in \operatorname{gra} A.$$

An operator is *monotone* (in the classical sense) if it is 0-monotone and it is *μ-strongly monotone* (in the classical sense) if it is μ-monotone with $\mu > 0$, in which case it is uniformly monotone with modulus $\phi(t) = \mu t^2$, for $t \in \mathbb{R}_+$. Obviously any strongly monotone operator is in particular monotone. The next result states that strong monotonicity is equivalent to cocoercivity of the inverse operator.

Proposition 2.11 ([44, Example 22.7]). *Let $\beta > 0$, $T : \mathcal{H} \to \mathcal{H}$, and set $A = T^{-1}$. Then T is $\frac{1}{\beta}$-cocoercive if and only if A is $\frac{1}{\beta}$-strongly monotone.*

2.2.2.1 Examples of maximally monotone operators

Example 2.12. *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be a maximally monotone operator, and let $\gamma > 0$. Then the operators A^{-1} and γA are maximally monotone.*

Example 2.13 (Skew symmetric operator). *Let $L : \mathcal{H} \to \mathcal{H}$ be a linear bounded operator. If L is* skew symmetric*, i.e., $L^* = -L$, then it is maximally monotone.*

By definition, it is clear that every cocoercive operator is monotone. Furthermore, the following proposition implies that cocoercive operators are maximally monotone.

Proposition 2.14 ([44, Corollary 20.38]). *Let $T : \mathcal{H} \to \mathcal{H}$ be monotone and continuous. Then T is maximally monotone.*

In general, the sum of maximally monotone operators may fail to be maximally monotone. The next result provides a condition under which maximal monotonicity is preserved.

Proposition 2.15 ([44, Corollary 25.5]). *Let $A, B : \mathcal{H} \rightrightarrows \mathcal{H}$ be two maximally monotone operators such that at least one of them has full domain. Then $A + B$ is maximally monotone.*

2.2.2.2 Resolvent of monotone operators

In this section, we present the concept of *resolvent* of a monotone operator, which will later become one of the main building blocks of splitting algorithms.

Definition 2.16. *Given an operator $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$, the* resolvent *of A with parameter $\gamma > 0$ is the operator $J_{\gamma A}\colon \mathcal{H} \rightrightarrows \mathcal{H}$ defined by $J_{\gamma A} := (\mathrm{Id} + \gamma A)^{-1}$.*

The following useful characterization of maximally monotone operators was proved by Minty [179]. One of its fundamental consequences is gathered in Theorem 2.18.

Theorem 2.17 (Minty's theorem). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be a monotone operator. Then A is maximally monotone if and only if* $\operatorname{ran}(\mathrm{Id} + A) = \mathcal{H}$.

Theorem 2.18 ([124, Proposition 3.4]). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be μ-monotone and let $\gamma > 0$ be such that $1 + \gamma\mu > 0$. Then*

(i) $J_{\gamma A}$ is single-valued;

(ii) $\operatorname{dom} J_{\gamma A} = \mathcal{H}$ *if and only if A is μ-maximally monotone.*

In particular, if A is monotone, then (i) and (ii) hold for every $\gamma > 0$.

Proposition 2.19 ([44, Corollary 23.9]). *Let $T : \mathcal{H} \to \mathcal{H}$. Then T is firmly nonexpansive if and only if it is the resolvent of a maximally monotone operator $A : \mathcal{H} \rightrightarrows \mathcal{H}$.*

The following propositions provide some formulas for the resolvents of certain operators. The first of them relates the resolvent of a maximally monotone operator to that of its inverse.

Proposition 2.20 ([44, Proposition 23.20]). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone and let $\gamma > 0$. Then*

$$\mathrm{Id} = J_{\gamma A} + \gamma J_{\gamma^{-1} A^{-1}} \circ \gamma^{-1} \mathrm{Id}\,.$$

Proposition 2.21 (Cartesian product of operators, [44, Proposition 23.18]). *Let $A_i : \mathcal{H}_i \rightrightarrows \mathcal{H}_i$ be maximally monotone operators, for integers $i \in \{1, \dots, n\}$. Define $\mathbf{A} := (A_1, A_2, \dots, A_n)$, also denoted by $\mathbf{A} := \bigtimes_{i=1}^n A_i$, as the operator given by*

$$\begin{aligned} \mathbf{A} : \quad \mathcal{H}_1 \times \cdots \times \mathcal{H}_n &\rightrightarrows \mathcal{H}_1 \times \cdots \times \mathcal{H}_n \\ (x_1, \dots, x_n) &\mapsto \big(A_1(x_1), \dots, A_n(x_n)\big). \end{aligned}$$

Then $\mathbf{A}$ is maximally monotone. Further, for any $\gamma > 0$, the resolvent of $\mathbf{A}$ with parameter γ is given component-wise as $(x_i)_{i=1}^n \mapsto \big(J_{\gamma A_i}(x_i)\big)_{i=1}^n$.

In principle, the composition of a maximally monotone operator with linear operators does not preserve monotoniciy (see Section 3.3). Nevertheless, the following result ensures maximal monotonicity for some particular compositions as well as characterizes its resolvent.

Proposition 2.22 ([44, Proposition 23.25]). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be a maximally monotone operator and $L : \mathcal{H} \to \mathcal{G}$ be a bounded linear operator. Assume LL^* is invertible and let $B = L^* A L$. Then the following hold.*

(i) $B : \mathcal{H} \rightrightarrows \mathcal{H}$ *is maximally monotone.*

(ii) Suppose that $LL^* = \mu \operatorname{Id}$ *for some* $\mu > 0$. *Then* $J_B = \operatorname{Id} - L^* \circ \frac{1}{\mu}(\operatorname{Id} - J_{\mu A}) \circ L$.

We conclude this section with the following result that is vital for the convergence analysis of splitting algorithms. It states that the graph of a maximally monotone operator $A : \mathcal{H} \rightrightarrows \mathcal{H}$ is sequentially closed in the *weak-strong topology* of $\mathcal{H} \times \mathcal{H}$.

Proposition 2.23 (Demiclosedness of a maximally monotone operator, [44, Proposition 20.38]). *Let* $A : \mathcal{H} \rightrightarrows \mathcal{H}$ *be a maximally monotone operator. Then* A *is* demiclosed, *i.e., for every sequence* $(x^k, u^k)_{k\in\mathbb{N}} \subseteq \operatorname{gra} A$, *if* $x^k \rightharpoonup x$ *and* $u^k \to u$, *then* $(x, u) \in \operatorname{gra} A$.

2.2.3 Convex optimization

The most elementary problem in convex optimization is that of minimizing a proper convex function $f : \mathcal{H} \to \overline{\mathbb{R}}$, that is,

$$\min_{x\in\mathcal{H}} f(x). \tag{2.3}$$

A solution to (2.3) can be characterized in terms of the *(convex) subdifferential* of f, which is the set-valued operator $\partial f : \mathcal{H} \rightrightarrows \mathcal{H}$ defined as

$$\partial f(x) = \{u \in \mathcal{H} \,:\, f(x) + \langle u, y - x\rangle \leq f(y),\ \forall y \in \mathcal{H}\}. \tag{2.4}$$

A vector $u \in \mathcal{H}$ belonging to $\partial f(x)$ is said to be a *subgradient* of f at $x \in \mathcal{H}$. Moreover, if f is proper, convex and differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$. *Fermat's rule* relates the global minimizers of f with the zeros of its subdifferential.

Theorem 2.24 (Fermat's rule, [44, Theorem 16.3]). *Let* $f : \mathcal{H} \to \overline{\mathbb{R}}$ *be a proper function. Then*

$$\operatorname{argmin} f = \operatorname{zer} \partial f := \{x \in \mathcal{H} \,:\, 0 \in \partial f(x)\}.$$

The next theorem establishes maximal monotonicity of the subdifferential of a proper l.s.c. convex function, and can be understood as the generalization of the fact that the derivative of a convex function of one real variable is monotonic non-decreasing. It dates back to Moreau [186, Chapter 12]. An earlier result assuming continuity was due to Minty in [180] while it was extended to Banach spaces by Rockafellar in [214].

Theorem 2.25. *Let $f : \mathcal{H} \to \overline{\mathbb{R}}$ be a proper lower-semicontinuous convex function. Then ∂f is maximally monotone*

From Theorem 2.25 and Fermat's rule, it follows that the minimization of a proper l.s.c. convex function is equivalent to the problem of finding a zero of a maximally monotone operator, a problem which will be investigated in detail in Chapter 3. Furthermore, it holds that $J_{\gamma\partial f} = \mathrm{prox}_{\gamma f} : \mathcal{H} \rightrightarrows \mathcal{H}$, where $\mathrm{prox}_{\gamma f}$ is the *proximity operator* of f (with parameter $\gamma > 0$) defined at $x \in \mathcal{H}$ by

$$\mathrm{prox}_{\gamma f}(x) := \operatorname*{argmin}_{u \in \mathcal{H}} \left\{ f(u) + \frac{1}{2\gamma}\|x - u\|^2 \right\},$$

see, e.g., [44, Theorem 20.25 & Example 23.3].

Example 2.26. *Given a nonempty set $C \subseteq \mathcal{H}$, the* indicator function *of C, $\iota_C : \mathcal{H} \to \overline{\mathbb{R}}$, is defined as*

$$\iota_C(x) := \begin{cases} 0, & \text{if } x \in C, \\ +\infty, & \text{if } x \notin C. \end{cases}$$

When C is a convex set, ι_C is a convex function whose subdifferential becomes the normal cone *to C, $N_C : \mathcal{H} \rightrightarrows \mathcal{H}$, given by*

$$\partial \iota_C(x) = N_C(x) := \begin{cases} \{u \in \mathcal{H} : \langle u, c - x\rangle \leq 0,\ \forall c \in C\}, & \text{if } x \in C, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Therefore, when C is nonempty, closed and convex, the normal cone N_C is maximally monotone. Furthermore, $J_{N_C} = P_C$, where $P_C : \mathcal{H} \to \mathcal{H}$ denotes the projection operator *onto C, which is defined at $x \in \mathcal{H}$ by*

$$P_C(x) := \operatorname*{argmin}_{c \in C} \|x - c\|,$$

see, e.g., [44, Example 20.26 & Example 23.4].

Let $f, g : \mathcal{H} \to \overline{\mathbb{R}}$ be proper functions. It is clear from the definition of the convex subdifferential that the following inclusion holds

$$\partial f(x) + \partial g(x) \subseteq \partial(f + g)(x),$$

for every $x \in \mathcal{H}$. However, in order to ensure the equality, some *constraint qualification* needs to be assumed.

Theorem 2.27 (Sum rule, [44, Theorem 16.47]). *Let $f : \mathcal{H} \to \overline{\mathbb{R}}$ and $g : \mathcal{H} \to \overline{\mathbb{R}}$ be proper l.s.c. convex functions and let $L : \mathcal{H} \to \mathcal{G}$ be a bounded linear operator. Suppose that $0 \in \operatorname{sri}\big(\operatorname{dom} g - L(\operatorname{dom} f)\big)$. Then,*

$$\partial(f + g \circ L) = \partial f + L^* \circ \partial g \circ L.$$

The conjugate function is vital for establishing a duality framework for convex optimization.

Definition 2.28 (Conjugate function). *Let $f : \mathcal{H} \to \overline{\mathbb{R}}$. The* conjugate *(also called* convex conjugate *or* Fenchel conjugate*) of f is the convex function $f^* : \mathcal{H} \to \overline{\mathbb{R}}$ given by*

$$f^*(x) := \sup_{u \in \mathcal{H}} \{\langle u, x\rangle - f(u)\}.$$

Proposition 2.29 (Fenchel–Young inequality, [44, Proposition 13.15 & Proposition 16.10]). *Let $f : \mathcal{H} \to \overline{\mathbb{R}}$ be proper. Then*

$$f(x) + f^*(u) \geq \langle x, u\rangle, \quad \forall x, u \in \mathcal{H}. \tag{2.5}$$

In addition, (2.5) holds with equality if and only if $u \in \partial f(x)$. In that case, $x \in \partial f^*(u)$.

If f is proper, the *Fenchel–Moreau theorem* [44, Theorem 13.37] states that f is convex and l.s.c. if and only if $f = f^{**}$, where $f^{**} := (f^*)^*$ denotes the biconjugate of f. In this case, f^* is also proper. Hence, its subdifferential is also maximally monotone and, by Proposition 2.29, it coincides with the inverse of the subdifferential of f, namely

$$\partial f^* = (\partial f)^{-1}.$$

Thus, using Proposition 2.20, the proximity operator of f^* can be immediately computed in terms of prox_f. More concretely, for any $\gamma > 0$, we have the expression

$$\operatorname{prox}_{\gamma f^*}(x) = x - \gamma \operatorname{prox}_{\gamma^{-1} f}(\gamma^{-1} x),$$

for every $x \in \mathcal{H}$.

Theorem 2.30 (Fenchel–Rockafellar duality, [44, Theorem 15.23]). *Assume that $f : \mathcal{H} \to \overline{\mathbb{R}}$ and $g : \mathcal{G} \to \overline{\mathbb{R}}$ are proper l.s.c. convex functions, and let $L : \mathcal{H} \to \mathcal{G}$ be a*

bounded linear operator. Suppose that $0 \in \operatorname{sri}\big(\operatorname{dom} g - L(\operatorname{dom} f)\big)$. *Then*

$$\inf_{x\in\mathcal{H}} f(x) + g(Lx) = -\min_{u\in\mathcal{G}} g^*(u) + f^*(-L^*u). \tag{2.6}$$

The minimization problem in the right-hand side of (2.6) is known as the *Fenchel–Rockafellar dual problem*. Note that by setting $L = \mathrm{Id}$ we recover the classical *Fenchel dual problem* [44, Proposition 15.13].

2.3 A selection of proximity operators

Some functions are said to be *prox-friendly*, which means that their proximity operators are easy to compute. In the following we present closed-form expressions of some selected proximity operators that shall appear in this thesis. Formulas for a wide range of proximity operators can be found in [110] or at the vast online repository [102].

Proposition 2.31 (Proximity operator of the ℓ_1-norm, [102]). *Let* $\gamma > 0$. *The proximal point mapping of the* ℓ_1*-norm of* $\mathbb{R}^n$, *denoted as* $\|\cdot\|_1$, *at a point* $x \in \mathbb{R}^n$ *is given by*

$$\operatorname{prox}_{\gamma\|\cdot\|_1}(x) = \operatorname{sign}(x) \odot [|x| - \gamma]_+,$$

where $\odot$ *denotes element-wise product, and* $[\,\cdot\,]_+$ *and* $|\cdot|$ *represent the positive part and absolute value, respectively, applied element-wise. That is, for each* $i \in \{1, \dots, n\}$, *its* i*-th component is given by*

$$\operatorname{prox}_{\gamma\|\cdot\|_1}(x)_i = \begin{cases} x_i + \gamma, & \text{if } x_i < -\gamma, \\ 0, & \text{if } |x_i| \le \gamma, \\ x_i - \gamma, & \text{if } x_i > \gamma. \end{cases}$$

As described in Example 2.26, the proximity operator of the indicator function of a nonempty convex closed set recovers the projection operator. The following propositions gather some projection formulas.

Proposition 2.32 (Projection onto a box, [102]). *Given* $a, b \in \mathbb{R}$ *with* $a < b$. *The projection onto the box* $[a, b]^n$ *of a point* $x \in \mathbb{R}^n$ *is given component-wise by*

$$P_{[a,b]^n}(x)_i = \max\{a, \min\{x_i, b\}\}, \qquad \text{for } i \in \{1, \dots, n\}.$$

Proposition 2.33 (Projection onto a half-space, [44, Example 29.20]). *Given $a \in \mathcal{H} \setminus \{0\}$ and $b \in \mathbb{R}$. The projection onto the half-space $H := \{x \in \mathcal{H} : \langle x, a\rangle \leq b\}$ of a point $x \in \mathcal{H}$ is given by*

$$P_H(x) = \begin{cases} x, & \text{if } \langle x, a\rangle \leq b, \\ x + \frac{b - \langle x, a\rangle}{\|a\|^2} a, & \text{if } \langle x, a\rangle > b. \end{cases}$$

Proposition 2.34 (Projection onto a ball, [44, Example 29.10]). *Let $\bar{x} \in \mathcal{H}$ and $\varepsilon > 0$. The projection onto the ball $C := \mathbb{B}_\varepsilon(\bar{x})$ of a point $x \in \mathcal{H}$ is given by*

$$P_C(x) = \begin{cases} x, & \text{if } \|x - \bar{x}\| \leq \varepsilon, \\ \bar{x} + \varepsilon \frac{x - \bar{x}}{\|x - \bar{x}\|}, & \text{otherwise.} \end{cases}$$

Proposition 2.35 (Projection onto an affine subspace, [44, Example 29.17]). *Let $A \in \mathbb{R}^{m \times n}$ with full row rank and $b \in \mathbb{R}^m$. The projection onto the affine subspace $C := \{x \in \mathbb{R}^n : Ax = b\}$ of a point $x \in \mathbb{R}^n$ is given by $P_C(x) = x - A^T(AA^T)^{-1}(Ax - b)$.*

Proposition 2.36 (Projection on the diagonal subspace, [44, Proposition 26.4]). *Let Δ_n denote the* diagonal subspace *of $\mathcal{H}^n$, defined as*

$$\Delta_n := \{(x, x, \ldots, x) \in \mathcal{H}^n : x \in \mathcal{H}\}.$$

The projection of a point $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{H}^n$ onto Δ_n is given by

$$P_{\Delta_n}(\mathbf{x}) = \left(\frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n x_i, \ldots, \frac{1}{n} \sum_{i=1}^n x_i \right).$$

Proposition 2.37 ([64, Section 4]). *Let $\alpha > 0$. Let $S \subseteq \mathbb{R}^n \times \mathbb{R}^n$ be the set defined as*

$$S := \left\{ (p, q) \in \mathbb{R}^n \times \mathbb{R}^n : \max_{1 \leq i \leq n} \sqrt{p_i^2 + q_i^2} \leq \alpha \right\}. \tag{2.7}$$

The projection operator $P_S : \mathbb{R}^n \times \mathbb{R}^n \to S$ onto S is given component-wise by

$$(x_i, y_i) \mapsto \alpha \frac{(x_i, y_i)}{\max\{\alpha, \sqrt{x_i^2 + y_i^2}\}}, \qquad \text{for } i \in \{1, \ldots, n\}.$$

2.4 Regularization techniques for sparse image recovery

Splitting methods, and in general first order methods, have been found specially successful for solving large-scale optimization problems arising in signal and image processing (see, e.g., [55, 64, 97, 100, 142]). In this section we briefly introduce some basic concepts on imaging that will enlighten the numerical experiments conducted throughout the thesis. Our main goal is to motivate the different formulations of the image recovery problem as an optimization program. We recommend the intrigued reader to refer to the work of Chambolle and Pock [101] for a thorough introduction to the role of convex optimization in imaging.

In the following, we face the task of recovering an unknown image $\bar{x}$ which has been transformed via a linear operator M (for instance, a blur), and perturbed by an additive random noise ε. This results in the observed image b to be expressible as

$$b = M\bar{x} + \varepsilon.$$

The recovery problem then amounts to obtaining an accurate approximation of $\bar{x}$ assuming the knowledge of M and b. In most situations the system $Mx = b$ is ill-posed, which encourages the employment of regularization techniques to avoid overfitting and induce some special structure in the recovered data. This leads to considering the optimization problem

$$\min_{x \in C} \frac{1}{2}\|Mx - b\|_p^2 + \alpha\rho(x),$$

where C is a restriction on the pixel values and $\|\cdot\|_p$ stands for some p-norm with $p \in \{1, 2, \ldots\} \cup \{\infty\}$. The regularization term is composed of the *regularization function* ρ and the parameter $\alpha > 0$, which is adapted to the noise level of b.

The most typical choice for the regularization function is the ℓ_1-norm, which is devoted to promote sparsity on the recovered image. In the celebrated paper [219] of 1992 , Rudin, Osher and Fatemi introduced the so-called *ROF model*, which employs the *total variation* function as a regularizer. In order to define the total variation, let us assume that $x \in \mathbb{R}^n$ describes an $N_1 \times N_2$ pixel image. We start by considering the discrete gradient operator $D : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n : x \to (D_1x, D_2x)$, where D_1 and D_2 are defined component-wise as

$$(D_1x)_{i,j} = \begin{cases} x_{i+1,j} - x_{i,j}, & \text{if } i < N_1, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad (D_2x)_{i,j} = \begin{cases} x_{i,j+1} - x_{i,j}, & \text{if } j < N_2, \\ 0, & \text{otherwise.} \end{cases} \tag{2.8}$$

The (*isotropic*) *total variation* function, also called *TV-norm*, is the convex function $TV : \mathbb{R}^n \to \mathbb{R}$ defined as

$$TV(x) = \|D(x)\|_{\times}, \qquad \text{where} \qquad \|(p,q)\|_{\times} := \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sqrt{p_{i,j}^2 + q_{i,j}^2}. \tag{2.9}$$

For practical applications, it is useful to recall that an upper bound of the Lipschitz constant of D is given by $\|D\| \leq \sqrt{8}$ (see [97, 98] for details).

The underlying idea to the total variation regularization is to induce sparsity to the gradients of the image, hence it favours the restoration of piece-wise constant images. In some circumstances it is convenient to replace the gradient operator with more general analysis operators. A popular choice is the *wavelet-based* approach, usually employed for signal compression. The simplest form of wavelet is given by the *Haar wavelet transform* [146], which allows to express an image by an alternative basis collecting information of pixel value averages. The image can be compressed in different stages or levels of scaling depending on how many averages are considered. An exhaustive mathematical description of this approach is out of the scope of this introductory section. The interested reader is referred to [228] for an intuitive presentation of the linear algebra governing the Haar wavelet.

Part I

Monotone operator splitting methods with minimal lifting

Chapter 3

A stroll through monotone inclusion problems and their splitting algorithms

A fundamental problem in mathematical programming is that of finding a zero of a monotone operator. Given a monotone operator $A : \mathcal{H} \rightrightarrows \mathcal{H}$, the inclusion problem

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in A(x) \tag{3.1}$$

finds applications in a broad range of areas, such as dynamical systems [1], image processing [99, 101, 142, 224], machine learning [111, 150] and optimal control [77, 78], among many others (see [106] and he references therein).

The inclusion (3.1) can alternatively be written as the problem of finding a fixed point of the resolvent of A. Indeed, for any $\gamma > 0$, we have the following chain of equivalences

$$0 \in A(\bar{x}) \iff \bar{x} \in (\mathrm{Id} + \gamma A)(\bar{x}) \iff \bar{x} = J_{\gamma A}(\bar{x}) \iff \bar{x} \in \operatorname{Fix} J_{\gamma A}, \tag{3.2}$$

where the equality in the third expression is due to Theorem 2.18 (i). Note that, when A is the subdifferential of a proper l.s.c. convex function f, (3.2) recovers the well-known characterization of optimality in terms of the proximity operator

$$\operatorname{argmin} f = \operatorname{Fix} \operatorname{prox}_{\gamma f},$$

for any $\gamma > 0$ (see, e.g., [44, Proposition 12.29]).

Equation (3.2) entails that the set of solutions to the inclusion (3.1) coincides with the set of fixed points of the resolvent of A. If, in addition, A is a maximally monotone operator, then the firm nonexpansiveness of its resolvent can be harnessed to successively approximate a zero by constructing a fixed point iteration using $J_{\gamma A}$. This is the

idea underlying the *proximal point algorithm*. This method was first introduced by Martinet [174] in the setting of variational inequalities and convex optimization problems, and later extended in 1976 to the more general framework of maximally monotone operators by Rockafellar [217].

Theorem 3.1 (Proximal point algorithm). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be a maximally monotone operator. Let $\gamma > 0$ and $\lambda \in]0, 2[$. Given an initial point $x^0 \in \mathcal{H}$, consider the sequence $(x^k)_{k\in\mathbb{N}}$ generated by the iteration*

$$x^{k+1} = (1 - \lambda)x^k + \lambda J_{\gamma A}(x^k), \quad \forall k \in \mathbb{N}. \tag{3.3}$$

Then, if $\operatorname{zer} A \neq \emptyset$, the sequence $(x^k)_{k\in\mathbb{N}}$ converges weakly to a point in $\operatorname{zer} A$.

In the next remark we provide some comments on the proof of Theorem 3.1, as it gathers the essence of the convergence analysis of the algorithmic schemes that will be studied throughout this thesis.

Remark 3.2 (On the convergence of the proximal point algorithm). The fixed point iteration given by (3.3) is a Krasnosel'skiĭ–Mann iteration of the firmly nonexpansive operator $J_{\gamma A}$ (see Proposition 2.19). In addition, by (3.2),

$$\operatorname{Fix} J_{\gamma A} = \operatorname{zer} A.$$

By assumption, this equality yields that $\operatorname{Fix} J_{\gamma A} \neq \emptyset$. Consequently, Proposition 2.8 implies that $(x^k)_{k\in\mathbb{N}}$ converges weakly to a point in $\operatorname{zer} A$.

Although in this section we will only discuss iterations with constant relaxation parameters, we recall that in view of Proposition 2.8 a sequence $(\lambda_k)_{k\in\mathbb{N}}$ of varying stepsizes can be taken in (3.3). For more details on the admitted stepsizes in the proximal point algorithm, we refer the reader to [106, Theorem 5.1].

While the proximal point algorithm is a powerful tool for addressing problem (3.1) when A is maximally monotone, in most situations in optimization we aim to find a zero of an operator which is not maximally monotone. *Monotone inclusions* gather a vast family of challenging problems with unique features. In order to successfully address them it is vital to be able to exploit their inherent characteristics and structure. The algorithms in the family of the *splitting methods* are able to do this by iteratively solving simpler problems which are defined by separately using some parts of the original problem. Further, as

these methods typically only use first-order information, they are well suited for large-scale problems. All this have contributed to make splitting algorithms the methods par excellence for addressing monotone inclusions.

In this section, we will introduce some of the most relevant monotone inclusion problems and present their applications to optimization. We will describe the different difficulties arising in their treatment, which urge to consider specific splitting schemes suitable for each problem.

3.1 Classical monotone inclusion

We start by considering one of the most basic monotone inclusion problems. It consists in finding a zero in the sum of two maximally monotone operators. Namely, given two maximally monotone operators $A_1, A_2 : \mathcal{H} \rightrightarrows \mathcal{H}$, we are interested in the problem

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in A_1(x) + A_2(x). \tag{3.4}$$

We shall refer to the inclusion in (3.4) as *classical monotone inclusion*, in order to distinguish it from problems where the operators hold additional properties beyond maximal monotonicity, which will be discussed in the subsequent sections.

In addition to the (primal) inclusion problem (3.4), in most situations it is useful to consider an associate dual problem. Duality in this setting was first studied by Mercier in [178, page 40] and widely disseminated after the work of Attouch and Théra [35].

Theorem 3.3 (Attouch–Théra duality, [35, Theorem 3.1]). *Let $A_1, A_2 : \mathcal{H} \rightrightarrows \mathcal{H}$ be two operators. The* primal inclusion problem (3.4) *and its* dual inclusion problem*, which is given by*

$$\textit{find } u \in \mathcal{H} \textit{ such that } 0 \in A_1^{-1}(u) - A_2^{-1}(-u), \tag{3.5}$$

are equivalent in the sense:

(i) If $x \in \mathcal{H}$ is a solution of (3.4), then there exists $u \in A_1(x)$ that solves (3.5).

(ii) If $u \in \mathcal{H}$ is a solution of (3.5), then there exists $x \in A_1^{-1}(u)$ that solves (3.4).

Problem (3.5) is usually referred to as the *Attouch–Théra dual problem* of (3.4). It is worth noting that this is an algebraic duality result and it does not require any assumptions concerning the operators. Nevertheless, if the operators A_1 and A_2 are maximally monotone, (3.5) also becomes a classical monotone inclusion, where the operators involved

are maximally monotone due to Example 2.12 and Proposition 2.22. Further results on duality for monotone inclusions can be found in [43, 49, 203].

3.1.1 Applications to optimization

We now analyze the subtle interplay between the monotone inclusion (3.4) and two core problems in mathematical optimization: the minimization of the sum of two convex functions and convex feasibility problems.

3.1.1.1 Minimization of the sum of two convex functions

Let $f, g : \mathcal{H} \to \overline{\mathbb{R}}$ be proper l.s.c. convex functions. We study the optimization problem

$$\min_{x \in \mathcal{H}} f(x) + g(x), \tag{3.6}$$

which we assume to have at least one solution. Provided that the constraint qualification $0 \in \operatorname{sri}(\operatorname{dom} g - \operatorname{dom} f)$ is satisfied, the subdifferential equality $\partial f(x) + \partial g(x) = \partial(f+g)(x)$ holds for every $x \in \mathcal{H}$, by Theorem 2.27. Consequently, Fermat's rule yields

$$\operatorname{argmin}(f+g) = \operatorname{zer}(\partial f + \partial g).$$

In other words, the minimization problem (3.6) becomes equivalent to finding a zero in the sum of the subdifferentials of f and g. This is, solving the monotone inclusion (3.4) setting

$$A_1 = \partial g \quad \text{and} \quad A_2 = \partial f.$$

Furthermore, with this replacement, the Attouch–Théra dual problem (3.5) corresponds to the Fenched dual problem in Theorem 2.30, which in this case reads as

$$\min_{u \in \mathcal{H}} g^*(u) + f^*(-u).$$

For further details on the tight interconnection between Fenchel and Attouch–Théra duality we recommend [189, Section 4.6].

At this point it is worth noting that, although it may seem that only working with subdifferential operators (instead of maximally monotone operators) is enough for addressing minimization problems, this is not the case. Indeed, resolvent operators of maximally monotone operators which are not subdifferentials naturally arise when tackling problems in the form of (3.6) (see, e.g., [53],[132, Proposition 4.10],[189, Example 5.32]).

3.1.1.2 Convex feasibility problems

Given two nonempty closed convex sets $C_1, C_2 \subseteq \mathcal{H}$, the *feasibility problem* consists in finding a point in the intersection of the sets, namely,

$$\text{find } x \in \mathcal{H} \text{ such that } C_1 \cap C_2. \tag{3.7}$$

This problem can be posed as the monotone inclusion (3.4) by considering the normal cone operators of the sets, i.e., by setting

$$A_1 = N_{C_1} \quad \text{and} \quad A_2 = N_{C_2}.$$

Equivalently, (3.7) is recovered from the minimization problem (3.6) with the choice of functions $f = \iota_{C_1}$ and $g = \iota_{C_2}$. Note that no constraint qualification is required when addressing feasibility problems by means of its associated monotone inclusion, besides the fact that $C_1 \cap C_2 \neq \emptyset$.

3.1.2 The Douglas–Rachford splitting algorithm

A first naive approach for the numerical resolution of (3.4) would be trying to apply the fixed point algorithm with the resolvent of $A_1 + A_2$. However, this strategy exhibits some inconveniences. First, the operator $A_1 + A_2$ might not be maximally monotone (see, e.g., [44, Example 25.1]), which is a necessary assumption to guarantee the convergence by Theorem 3.1. Second, we may not be able to precisely compute the resolvent $J_{A_1+A_2}$, even having closed formulas for the resolvents of A_1 and A_2. *Monotone operator splitting algorithms* provide an alternative to avoid these pitfalls, as they permit to separately tackle the different elements in the problem. For instance, in the framework of (3.4), it would be desirable to handle the maximally monotone operators A_1 and A_2 individually, by means of each of their resolvents.

This is the case of the *Douglas–Rachford splitting algorithm*, which is undoubtedly the most celebrated scheme among the class of splitting methods (see, e.g., [19, 48, 50, 51, 61, 164]). This algorithm owes its name to the 1956 paper of Jim Douglas and Henry H. Rachford [131], where it was originally proposed for tackling systems of linear equations where the matrices are symmetric and positive semidefinite. The scheme given there was solely applied to a particular problem in heat conduction, and in fact it is not trivial to find an expression that resembles the current form of the Douglas–Rachford splitting algorithm in it. It was Lions and Mercier [165] who presented in 1979 the generalization to

maximally monotone operators widely-known nowadays. In the last decades, many authors have contributed to analyzing the convergence properties of the Douglas–Rachford method scheme introduced by Lions and Mercier (see, e.g., [41, 45, 107, 108, 133, 134]). However, it was not until 2011 when Svaiter [230] provided the complete result guaranteeing weak convergence to a solution of (3.4).

Theorem 3.4 (Douglas–Rachford splitting algorithm). *Let $A_1, A_2 : \mathcal{H} \rightrightarrows \mathcal{H}$ be two maximally monotone operators such that $\operatorname{zer}(A_1 + A_2) \neq \emptyset$. Let $\gamma > 0$ and $\lambda \in]0, 2[$. Given an initial point $z^0 \in \mathcal{H}$, consider the sequences $(z^k)_{k\in\mathbb{N}}$ and $(x^k)_{k\in\mathbb{N}}$ given by the iteration*

$$\begin{cases} x^k = J_{\gamma A_1}(z^k), \\ z^{k+1} = z^k + \lambda\big(J_{\gamma A_2}(2x^k - z^k) - x^k\big), \end{cases} \tag{3.8}$$

for all $k \in \mathbb{N}$. Then the following statements hold.

(i) The sequence $(z^k)_{k\in\mathbb{N}}$ converges weakly to a point $\bar{z} \in \mathcal{H}$.

(ii) The sequence $(x^k)_{k\in\mathbb{N}}$ converges weakly to the point $\bar{x} := J_{\gamma A_1}(\bar{z})$, which solves the monotone inclusion (3.4).

Proof. This follows as a particular case of Theorem 5.3. It can also be found in [44, Theorem 26.11]. □

The iteration (3.8) generated by the Douglas–Rachford algorithm can be alternatively obtained as a fixed point iteration of the commonly-known as *Douglas–Rachford splitting operator*.

Remark 3.5 (Douglas–Rachford algorithm as a fixed point iteration). Define the *Douglas–Rachford splitting operator*, $\mathcal{T}_{DR} : \mathcal{H} \to \mathcal{H}$, as

$$\mathcal{T}_{DR}(z) = z + \lambda\big(J_{\gamma A_2}(2J_{\gamma A_1}(z) - z) - J_{\gamma A_1}(z)\big).$$

Then the sequence $(z^k)_{k\in\mathbb{N}}$ generated by the Douglas–Rachford algorithm is obtained as the fixed point iteration

$$z^{k+1} = \mathcal{T}_{DR}(z^k), \quad \forall k \in \mathbb{N}.$$

We can now refine the statement in Theorem 3.4 (i) by highlighting that the sequence $(z^k)_{k\in\mathbb{N}}$ converges weakly to a fixed point of $\mathcal{T}_{DR}$, namely, $z^k \rightharpoonup \bar{z} \in \operatorname{Fix} \mathcal{T}_{DR}$. It is important to note that, in general, the fixed point $\bar{z}$ does not necessarily belong to $\operatorname{zer}(A_1 + A_2)$.

Nonetheless, by Theorem 3.4 (ii) we obtain a solution to the monotone inclusion (3.4) by applying the resolvent of A_1 to $\bar{z}$, i.e.,

$$\bar{x} := J_{\gamma A_1}(\bar{z}) \in \operatorname{zer}(A_1 + A_2).$$

Hence, the operator $J_{\gamma A_1}$ is what we term a *solution operator* of the Douglas–Rachford splitting. We shall pay special attention to this class of mappings in Chapter 4.

Finally, we recall that the Douglas–Rachford splitting operator has an equivalent formulation in terms of *reflected resolvents*. The reflected resolvent of a maximally monotone operator A is the nonexpansive operator given as $R_{\gamma A} := 2J_{\gamma A} - \mathrm{Id}$. In particular, if A is the normal cone of a nonempty closed convex set C, then R_{N_C} coincides with the reflection onto C. In view of this, the Douglas–Rachford operator can be expressed as

$$\mathcal{T}_{DR}(z) = \frac{(2-\lambda)z + \lambda R_{\gamma A_2} R_{\gamma A_1}(z)}{2}. \tag{3.9}$$

For the Douglas–Rachford algorithm, and also for other splittings schemes, we call the fixed point sequence $(z^k)_{k\in\mathbb{N}}$ as *governing sequence*, while we use the term *shadow sequence* for referring to the sequence converging to a solution of the monotone inclusion, in this case $(x^k)_{k\in\mathbb{N}}$.

In order to illustrate the interplay between the fixed point iteration $(z^k)_{k\in\mathbb{N}}$ and the solutions of the monotone inclusion problem, we concentrate on a feasibility problem in the form of (3.7) where $C_1, C_2 \subseteq \mathbb{R}^2$ are the closed balls displayed in Figure 3.1. Let $\lambda = 1$. Then the formulation of the Douglas–Rachford operator given in (3.9) becomes particularly intuitive. Indeed, $(z^k)_{k\in\mathbb{N}}$ is generated by the iteration

$$z^{k+1} = \frac{z^k + R_{C_2} R_{C_1}(z^k)}{2}, \quad \forall k \in \mathbb{N},$$

where R_{C_1} and R_{C_2} denote the reflections onto the balls C_1 and C_2, respectively. Namely, at each iteration the method performs the double reflection $R_{C_2} R_{C_1}(z^k)$ and computes the average with z^k to obtain the following iterate. For the first iteration these computations are shown in Figure 3.1 left. In Figure 3.1 right, we observe how $(z^k)_{k\in\mathbb{N}}$ converges to a fixed point z which is not in the intersection of the two balls. However, $P_{C_1}(z)$ provides a solution to the feasibility problem.

Remark 3.6 (Peaceman–Rachford algorithm). The Peaceman–Rachford algorithm [165, 200] can be regarded as the limiting case of the Douglas–Rachford splitting in which $\lambda = 2$. In [44, Proposition 26.13], an analogous result to Theorem 3.4 for the Peaceman–Rachford

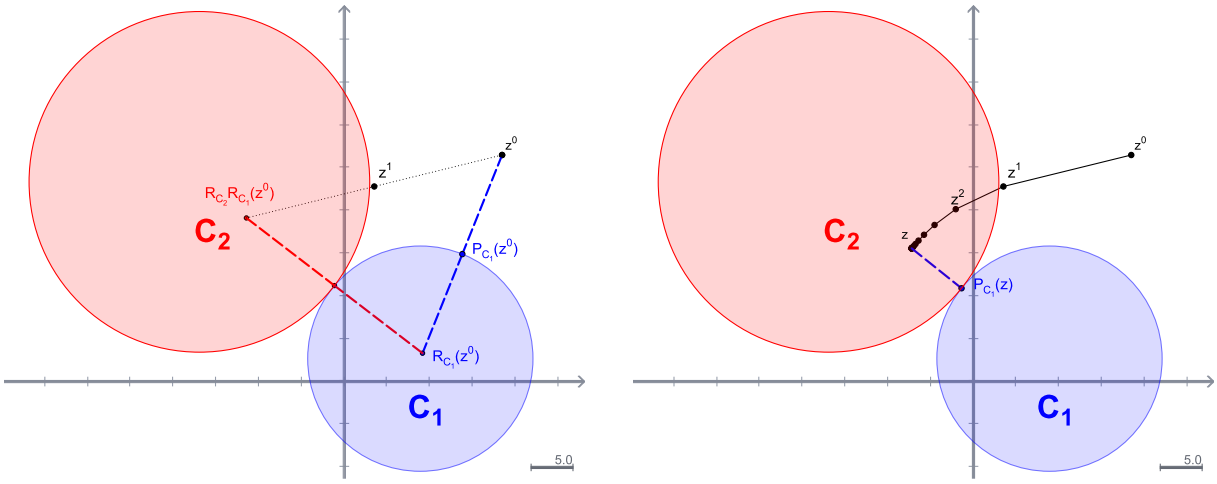


FIGURE 3.1: Geometric interpretation of the Douglas–Rachford splitting method for a feasibility problem involving two balls.

splitting follows provided that A_2 is uniformly monotone. In this context, the sequence $(x^k)_{k\in\mathbb{N}}$ converges strongly to a point in $\operatorname{zer}(A_1 + A_2)$. Observe that strong convergence under the assumption of uniform monotonicity can also be obtained by the Douglas–Rachford algorithm (see Theorem 5.3 below).

3.1.3 Pierra's product space reformulation

Now we draw our attention to the monotone inclusion problem consisting in finding a zero of the sum of an arbitrary finite number of maximally monotone operators.

Problem 3.7. *Let $A_1, A_2, \ldots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$ be a family of maximally monotone operators. We aim to solve the monotone inclusion*

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^{n} A_i(x).$$

The above problem finds immediate applications to feasibility problems and convex optimization problems with more than two sets and functions, respectively. Of course, if $n = 2$, we recover the monotone inclusion in (3.4) and the Douglas–Rachford algorithm becomes the benchmark method for addressing the problem. Until quite recently the approach for tackling more than two operators has solely consisted in the use of the so-called *Pierra's product space reformulation* [206]. Specifically, this reformulation allows solving Problem 3.7 by applying Douglas–Rachford to a concrete choice of operators.

Let $n \geq 2$ in Problem 3.7. Let us consider the normal cone $N_{\Delta_n} : \mathcal{H}^n \rightrightarrows \mathcal{H}^n$ onto the diagonal subspace of $\mathcal{H}^n$ and the Cartesian product of operators

$$\mathbf{A} := (A_1, A_2, \ldots, A_n) : \mathcal{H}^n \rightrightarrows \mathcal{H}^n,$$

whose resolvents are given by Proposition 2.36 and Proposition 2.21, respectively. By Theorem 3.4, the Douglas–Rachford splitting algorithm applied to N_{Δ_n} and $\mathbf{A}$ converges to a point $\bar{\mathbf{x}} \in \mathcal{H}^n$ such that

$$0 \in N_{\Delta_n}(\bar{\mathbf{x}}) + \mathbf{A}(\bar{\mathbf{x}}).$$

In particular, this implies that $N_{\Delta_n}(\bar{\mathbf{x}}) \neq \emptyset$, and therefore $\bar{\mathbf{x}} = (\bar{x}, \ldots, \bar{x})$ for some $\bar{x} \in \mathcal{H}$. Let $\mathbf{a} = (a_1, \ldots, a_n) \in \mathcal{H}^n$ be such that $\mathbf{a} \in A(\bar{\mathbf{x}})$ and $-\mathbf{a} \in N_{\Delta_n}(\bar{\mathbf{x}})$. The latter yields $\sum_{i=1}^n a_i = 0$, by [44, Proposition 26.4 (ii)]. Therefore, we get that

$$0 = a_1 + a_2 + \ldots + a_n \in A_1(\bar{x}) + A_2(\bar{x}) + \ldots + A_n(\bar{x}),$$

which concludes that $\bar{x}$ solves Problem 3.7.

Remark 3.8 (Pierra's reformulation does not recover the Douglas–Rachford operator). In principle, one would expect that applying Pierra's product space reformulation with $n = 2$ may recover the Douglas–Rachord splitting operator $\mathcal{T}_{DR}$. However, by setting $A_1 := N_{\Delta_2}$ and $A_2 := \mathbf{A}$ in Theorem 3.4, we obtain the fixed point iteration generated by the operator $\mathcal{T}_{PR} : \mathcal{H}^2 \to \mathcal{H}^2$ defined as

$$\mathcal{T}_{PR}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} + \frac{\lambda}{2}\begin{pmatrix} 2J_{\gamma A_1}(z_2) - z_1 - z_2 \\ 2J_{\gamma A_2}(z_1) - z_2 - z_1 \end{pmatrix},$$

where $\lambda, \gamma > 0$. The operator $\mathcal{T}_{PR}$ not only does not coincide with $\mathcal{T}_{DR}$, but it is defined in a different space! Observe that inverting the order of the operators when applying Douglas–Rachford to Pierrra's product space reformulation, i.e., setting $A_1 := \mathbf{A}$ and $A_2 := N_{\Delta_2}$, modifies the definition of $\mathcal{T}_{PR}$. Nonetheless, the resulting operator remains a mapping from $\mathcal{H}^2$ to $\mathcal{H}^2$.

For the general monotone inclusion with any $n \geq 2$, Pierra's reformulation is defined by an operator in the space $\mathcal{H}^n$ (see (4.2) below). Nonetheless, the Douglas–Rachford splitting suggests the plausibility of obtaining algorithms for this problem whose underlying ambient space is $\mathcal{H}^{n-1}$. This was a long-standing open problem, already posed by Lions and Mercier in 1979, and finally answered by Ryu [220] in 2020. In this work, it was proposed the first splitting algorithm for Problem 3.7 with $n = 3$, which is defined

by an operator in $\mathcal{H}^2$. Moreover, *Ryu's splitting*, as it is commonly called, leads to the Douglas–Rachford method when applied to a problem with two operators. The study of monotone operator splittings which are embeddable in spaces of reduced dimension is one of the main topics of Part I of this thesis. We shall start to delve into it in Chapter 4.

3.2 Monotone inclusions with Lipschitzian operators

In the previous section we focused on Problem 3.7, where only maximal monotonicity of the operators was assumed. The splitting algorithms studied there to address such monotone inclusion are referred to as *resolvent splittings*, as they handle each operator A_i by means of its resolvent. Resolvent operators cannot always be computed efficiently and closed formulas are only known for a selection of operators. In some problems, the operators involved possess stronger properties beyond maximal monotonicity, which allows avoiding the use of resolvents. In this section, we focus on a class of monotone inclusions encompassing operators with Lipchitzian properties.

Problem 3.9 (Monotone inclusion with Lipschitzian operators). *Let the operators $A_1, \ldots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone and let $T_1, \ldots, T_m : \mathcal{H} \to \mathcal{H}$ be cocoercive operators or monotone and Lipschitz continuous operators. The problem consists in solving the inclusion*

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \left(\sum_{i=1}^{n} A_i + \sum_{j=1}^{m} T_j\right)(x). \tag{3.10}$$

For simplicity, we will omit the index when only one operator of a class is considered. For instance, if $n = 1$, we simply write A instead of A_1.

We start by recalling that a cocoercive operator is Lipschitz continuous due to the Cauchy–Schwarz inequality. However, the converse implication is not true in general. For instance, the matrix

$$T = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \tag{3.11}$$

is clearly Lipschitz continuous and monotone, as it is skew symmetric. Nevertheless, by taking $x := (1, 0)^T$ and $y := (0, 0)^T$ it follows that

$$0 = \langle T(x) - T(y), x - y \rangle < \frac{1}{\beta}\|T(x) - T(y)\|^2 = \frac{1}{\beta}, \quad \forall \beta > 0.$$

Hence, T is not cocoercive. In any case, both cocoercive operators and monotone and Lipschitz continuous operators are maximally monotone by Proposition 2.14. Therefore, the operators comprising the monotone inclusion in Problem 3.9 are all maximally monotone.

3.2.1 Applications to optimization

Problem 3.9 arises in a number of settings of fundamental importance in mathematical optimization. In what follows, we describe three such examples.

3.2.1.1 Structured convex minimization

Consider the minimization problem given by

$$\min_{x\in\mathcal{H}} \sum_{i=1}^{n} g_i(x) + \sum_{j=1}^{m} f_j(x), \tag{3.12}$$

where $g_1, \ldots, g_n \colon \mathcal{H} \to \overline{\mathbb{R}}$ are proper l.s.c. convex functions, and $f_1, \ldots, f_m \colon \mathcal{H} \to \mathbb{R}$ are convex and differentiable with β-Lipschitz continuous gradients. Through its first order optimality condition and under appropriate constraint qualifications (see, e.g, [44, Corollary 16.50]), problem (3.12) can be posed as the monotone inclusion (3.10) with

$$A_i = \partial g_i \quad \text{and} \quad T_j = \nabla f_j.$$

Recall from Remark 2.2 (c), that in this case the operators $T_1, \ldots, T_m$ are both β-Lipschitz and $\frac{1}{\beta}$-cocoercive, due to the *Baillon–Haddad theorem*.

3.2.1.2 Structured saddle-point problems

Let us now work over the saddle-point problem given by

$$\min_{x\in\mathcal{H}} \max_{y\in\mathcal{G}} \sum_{i=1}^{n} g_i(x) + \sum_{j=1}^{m} \Phi_j(x,y) - \sum_{i=1}^{n} h_i(y), \tag{3.13}$$

where $g_1, \ldots, g_n \colon \mathcal{H} \to \overline{\mathbb{R}}$, $h_1, \ldots, h_n \colon \mathcal{G} \to \overline{\mathbb{R}}$ are proper l.s.c. convex functions, and the saddle-functions $\Phi_1, \ldots, \Phi_m \colon \mathcal{H} \times \mathcal{G} \to \mathbb{R}$ are differentiable convex-concave functions with Lipschitz continuous gradient. Assuming a saddle-point exists, (3.13) can be posed as

(3.10) in the space $\mathcal{H} \times \mathcal{G}$ with

$$A_i(x,y) = \begin{pmatrix} \partial g_i(x) \\ \partial h_i(y) \end{pmatrix} \quad \text{and} \quad T_j(x,y) = \begin{pmatrix} \nabla_x \Phi_j(x,y) \\ -\nabla_y \Phi_j(x,y) \end{pmatrix},$$

where we note that the operators $T_1, \ldots, T_m \colon \mathcal{H} \times \mathcal{G} \to \mathcal{H} \times \mathcal{G}$ are monotone, due to [216, Corollary 2], and β-Lipschitz continuous, but generally not cocoercive.

3.2.1.3 Variational inequalities

Consider the variational inequality problem given by

$$\text{find } \bar{x} \in \mathcal{H} \text{ such that } \sum_{i=1}^{n} g_i(x) - \sum_{i=1}^{n} g_i(\bar{x}) + \sum_{j=1}^{m} \langle T_j(\bar{x}), x - \bar{x} \rangle \geq 0, \quad \forall x \in \mathcal{H}, \tag{3.14}$$

where $g_1, \ldots, g_n \colon \mathcal{H} \to \overline{\mathbb{R}}$ are proper l.s.c. convex functions, and $T_1, \ldots, T_m \colon \mathcal{H} \to \mathcal{H}$ are monotone and β-Lipschitz. Then (3.14) can be modeled as (3.10) with $A_i = \partial g_i$. An important special case of (3.14) is the constrained variational inequality problem given by

$$\text{find } \bar{x} \in \mathcal{H} \text{ such that } \sum_{j=1}^{m} \langle T_j(\bar{x}), x - \bar{x} \rangle \geq 0, \quad \forall x \in C := \bigcap_{i=1}^{n} C_i,$$

where $C_1, \ldots, C_n \subseteq \mathcal{H}$ are nonempty, closed and convex sets. This formulation allows one to exploit a representation of the set C in terms of the simpler sets $C_1, \ldots, C_n$.

3.2.2 Forward-backward-type splitting methods for cocoercive operators

When addressing Problem 3.9, it is desirable to employ methods adept at harnessing the Lipchitzian properties of the single-valued operators. The family of *forward-backward splitting methods* is comprised of a collection of algorithms whose iterations make use of resolvents —also called *backward steps*— of the operators $A_1, \ldots, A_n$, and direct evaluations —*forward steps*— of the single-valued operators $T_1, \ldots, T_m$. In the following, we introduce some of the most popular algorithms in this class.

Observe that it is always possible to reduce Problem 3.9 to the $m = 1$ case by combining the single-valued operators into a single operator $T := \sum_{j=1}^{m} T_j$ whilst preserving the above features. However, since the resolvent of a sum is generally not related to the individual resolvents, the same cannot be said for the set-valued operators, and so it makes sense to distinguish algorithms for Problem 3.9 based on the value of n. In addition, as we will

demonstrate later, problems with monotone and Lipschitz continuous operators require the use of different algorithmic schemes than the ones consisting of cocoercive operators. We devote this section to tackling Problem 3.9 when all the single-valued operators are cocoercive, and will delve into the monotone and Lipschitz case in the subsequent section. In the following, we make the recurrent assumption that $T : \mathcal{H} \to \mathcal{H}$ is a $\frac{1}{\beta}$-cocoercive operator, with $\beta > 0$.

3.2.2.1 The forward-backward splitting algorithm

The *forward-backward method* is arguably the best-known method for the case $n = 1$. This algorithm is a generalization of the proximal point algorithm which includes a forward evaluation of the cocoercive operator before the computation of the resolvent. Let $\lambda, \gamma > 0$. Given $x^0 \in \mathcal{H}$, the method is governed by the iteration

$$x^{k+1} = (1-\lambda)x^k + \lambda J_{\gamma A}\big(x^k - \gamma T(x^k)\big), \quad \forall k \in \mathbb{N}. \tag{3.15}$$

The sequence $(x^k)_{k\in\mathbb{N}}$ thus generated converges weakly to a point in $\operatorname{zer}(A+T)$ provided that the stepsize and relaxation parameters satisfy $\gamma \in \left]0, 4/\beta\right[$ and $\lambda \in \left]0, 2 - \frac{\gamma\beta}{2}\right[$, respectively. The theorem of convergence of the forward-backward method is detailed in Corollary 5.5 below.

The iteration (3.15) has its roots in the *projected-gradient method* for the minimization of a smooth function over a closed convex constraint set [143, 161]. The method was extended to variational inequalities in 1979 by Mercier [177, Chapter 9], who first identified cocoercivity as a key property to establish weak convergence. Since then it has been widely studied by many authors (see, e.g., [70, 106, 107, 112, 113, 114, 140, 198, 236]).

3.2.2.2 The Davis–Yin splitting algorithm

Algorithms for the $n = 2$ case have only recently been proposed. One of them is the commonly-known as *Davis–Yin splitting algorithm*, which was introduced in 2017 by Damek Davis and Wotao Yin [128]. Let $\lambda, \gamma > 0$. Given an initial point $z^0 \in \mathcal{H}$, this method is defined by the iteration

$$\begin{cases} x^k = J_{\gamma A_1}(z^k), \\ z^{k+1} = z^k + \lambda\big(J_{\gamma A_2}\big(2x^k - z^k - \gamma T(x^k)\big) - x^k\big), \end{cases} \tag{3.16}$$

for all $k \in \mathbb{N}$. Observe that both the Douglas–Rachford and the forward-backward algorithms can be recovered as particular cases of the Davis–Yin method. It suffices to set $T = 0$ for Douglas–Rachford, while $A_1 = 0$ for the forward-backward. When $A_2 = 0$, the scheme (3.16) becomes the backward-forward method of [34].

In their seminal work, Davis and Yin proved the weak convergence of the sequence $(x^k)_{k\in\mathbb{N}}$ to a a point in zer $(A_1 + A_2 + T)$ when $\gamma \in]0, 2/\beta[$ and $\lambda \in \left]0, 2 - \frac{\gamma\beta}{2}\right[$. In the last few years, the Davis–Yin splitting algorithm has been a constant object of research (see, e.g., [117, 122, 160, 181, 201, 202, 232, 241, 245, 247]). The convergence of the method will be deeply studied in Chapter 5. Specifically, in Theorem 5.3 we detail a new proof which enables the stepsize γ to be chosen in the interval $]0, 4/\beta[$.

3.2.2.3 The generalized forward-backward algorithm

For an arbitrary finite number of operators, Problem 3.9 can be tackled by the *generalized forward-backward algorithm* of [209, 210]. This method can be understood as an application of the Davis–Yin algorithm to a modification of Pierra's product space reformulation with an additional cocoercive operator. Indeed, let $(\omega_i)_{i=1}^n \in]0, 1[^n$ with $\sum_{i=1}^n \omega_i = 1$, and consider the Hilbert space $\mathcal{H}^n$ endowed with the inner product $\langle \cdot, \cdot \rangle$ given by

$$\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^{n} \omega_i \langle x_i, y_i \rangle, \quad \forall \mathbf{x} = (x_1, \ldots, x_n), \forall \mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{H}^n.$$

Then the generalized forward-backward amounts to applying the Davis–Yin algorithm to the problem

$$\text{find } \mathbf{x} \in \mathcal{H}^n \text{ such that } 0 \in N_{\Delta_n}(\mathbf{x}) + \mathbf{A}(\mathbf{x}) + \mathbf{T}(\mathbf{x}),$$

where $\mathbf{A} : \mathcal{H}^n \rightrightarrows \mathcal{H}^n$ is the maximally monotone operator defined as

$$\mathbf{A}(\mathbf{x}) = \left(\frac{1}{\omega_1} A_1(x_1), \frac{1}{\omega_2} A_2(x_2), \ldots, \frac{1}{\omega_n} A_n(x_n) \right), \tag{3.17}$$

and $\mathbf{T} : \mathcal{H}^n \to \mathcal{H}^n$ is the cocoercive operator given by

$$\mathbf{T}(\mathbf{x}) = \big(T(x_1), T(x_2), \ldots, T(x_n)\big). \tag{3.18}$$

The resulting scheme is presented in Theorem 3.10. Despite the above interpretation, the generalized forward-backward goes back to before the Davis–Yin method, as it was originally proposed in 2013. We refer the interested reader to [73, Section 6] for further

details on the connection between the generalized forward-backward method and the Davis–Yin splitting.

Theorem 3.10 (Generalized forward-backward algorithm, [210, Theorem 2.1]). *Let $A_1, \ldots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone operators and let $T : \mathcal{H} \to \mathcal{H}$ be a $\frac{1}{\beta}$-cocoercive operator such that $\operatorname{zer}\left(\sum_{i=1}^n A_i + T\right) \neq \emptyset$. Choose $(\omega_i)_{i=1}^n \in \left]0,1\right[^n$ with $\sum_{i=1}^n \omega_i = 1$. Take $\gamma \in \left]0, 2/\beta\right[$ and set $\lambda \in \left]0, \min\left\{\frac{3}{2}, \frac{1}{2} + \frac{1}{\gamma\beta}\right\}\right[$. Given an initial guess $\mathbf{z}^0 = (z_1^0, \ldots, z_n^0) \in \mathcal{H}^n$, consider the sequences $(x^k)_{k\in\mathbb{N}}$ and $(\mathbf{z}^k)_{k\in\mathbb{N}} = (z_1^k, \ldots, z_n^k)_{k\in\mathbb{N}}$ generated by the iteration*

$$\begin{cases} x^k = \displaystyle\sum_{i=1}^n \omega_i z_i^k, \\ z_i^{k+1} = z_i^k + \lambda\big(J_{\gamma/\omega_i A_i}\big(2x^k - z_i^k - \gamma T(x^k)\big) - x^k\big), \quad \forall i \in [\![1, n]\!], \end{cases} \tag{3.19}$$

for all $k \in \mathbb{N}$. Then the sequence $(x^k)_{k\in\mathbb{N}}$ converges weakly to a point in $\operatorname{zer}\left(\sum_{i=1}^n A_i + T\right)$.

The new proof of the Davis–Yin splitting algorithm presented in Chapter 5 also leads to an enlargement of the parameters range of the generalized forward-backward. This is discussed in Remark 5.6 below.

3.2.3 Forward-backward splitting methods for Lipschitz continuous operators

We now concentrate on Problem 3.9 with the operators $T_1, \ldots, T_m$ being monotone and Lipschitz continuous. Again, we gather the m single-valued operators into the sum operator $T := \sum_{j=1}^m T_j$ and assume it to be β-Lipschitz continuous. Developing splitting algorithms which use forward evaluations of Lipschitz continuous monotone operators is generally more intricate than those exploiting cocoercivity, such as the ones in the previous subsection. For a concrete example, consider the special case of (6.22) with two operators given by

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in (A + T)(x), \tag{3.20}$$

with $\mathcal{H} = \mathbb{R}^2$, $A = 0$ and T is the matrix in (3.11), which is not cocoercive. The monotone inclusion (3.20) is equivalent to solving the linear equation

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0, \tag{3.21}$$

whose unique solution is $(0,0)^T$. The forward-backward method applied to this problem results in the iteration

$$x^{k+1} = (\mathrm{Id} - \lambda\gamma T)(x^k) = \begin{pmatrix} 1 & \lambda\gamma \\ -\lambda\gamma & 1 \end{pmatrix} x^k, \quad \forall k \in \mathbb{N}.$$

The sequence $(x^k)_{k\in\mathbb{N}}$ diverges for any $\lambda, \gamma > 0$ and any initial point $x^0 \neq (0,0)^T$. Indeed, since the eigenvalues of $\mathrm{Id} - \lambda\gamma T$ are $1 \pm \lambda\gamma i$, at every iteration the forward-backward method amplifies the norm of the previous iterate by a factor $|1 + \lambda\gamma i| > 1$ (see Figure 3.2 below).

In general, the methods that successfully solve this problem require one extra forward evaluation of the Lipschitz monotone operators per iteration. This is the case of the algorithms we present next.

3.2.3.1 The forward-reflected-backward splitting algorithm

A subtle modification in the forward-backward algorithm gives rise to the *forward-reflected-backward method* [172]. Given $\gamma > 0$ and $x^{-1}, x^0 \in \mathcal{H}$ this method iterates as

$$x^{k+1} = J_{\gamma A}\big(x^k - 2\gamma T(x^k) + \gamma T(x^{k-1})\big), \quad \forall k \in \mathbb{N}. \tag{3.22}$$

The schemes coincides with the one in (3.15) (when $\lambda = 1$) with the addition of the reflected-like term $-\gamma\big(T(x^k) - T(x^{k-1})\big)$. Note that the Lipschitzian operator is invoked twice per iteration, as T is evaluated at both x^k and x^{k-1}. This also entails that knowledge from the two previous iterates is required at every iteration. Unlike (3.15), the forward-reflected-backward method converges weakly to a point in $\operatorname{zer}(A+T)$ for any $\gamma \in \left]0, \frac{1}{2\beta}\right[$. This is illustrated for the linear system (3.21) in Figure 3.2. Other methods making use of reflected-like terms are the ones in [121, 212, 223].

3.2.3.2 Tseng's forward-backward-forward splitting algorithm

One of the most renowned splittings for addressing (3.20) is the *forward-backward-forward method* proposed by Tseng in [237], which is itself a generalization of the scheme investigated in [8]. At each iteration, the algorithm first activates the Lipschitz continuous operator T with a forward evaluation, followed by a backward evaluation of A and finally performs a second forward evaluation of T.

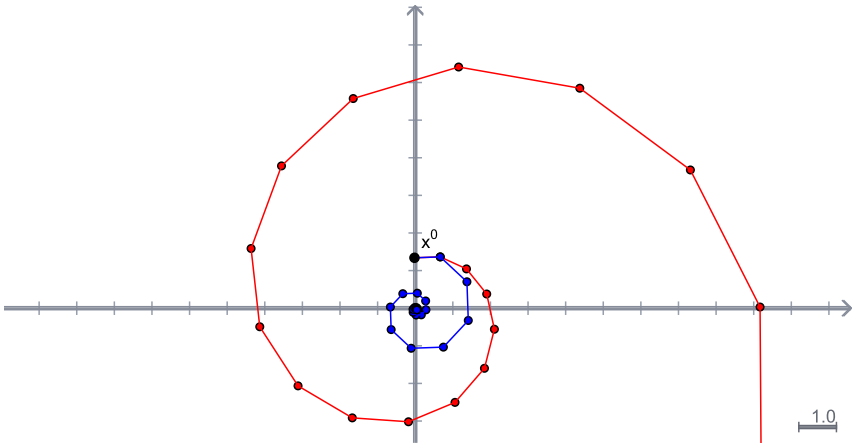


FIGURE 3.2: Forward-backward (red) and forward-reflected-backward (blue) for the linear equation (3.21).

Theorem 3.11 (Forward-backward-forward algorithm, [44, Theorem 26.17]). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be a maximally monotone operator and $T : \mathcal{H} \to \mathcal{H}$ be a monotone and β-Lipschitz continuous operator such that $\operatorname{zer}(A+T) \neq \emptyset$. Given $\gamma \in]0, 1/\beta[$ and $x^0 \in \mathcal{H}$, iterate as*

$$\begin{cases} u^k = \gamma T(x^k), \\ v^k = J_{\gamma A}(x^k - u^k), \\ x^{k+1} = v^k - \gamma T(v^k) + u^k, \end{cases}$$

for all $k \in \mathbb{N}$. Then the sequences $(x^k)_{k\in\mathbb{N}}$ and $(v^k)_{k\in\mathbb{N}}$ converge weakly to a point in $\operatorname{zer}(A+T)$.

Observe that, in contrast to the forward-reflected-backward, only information from the previous iterate is required in Tseng's algorithm. Extensive results and generalizations of this method can be found in [44, 68, 74, 75, 106, 109, 188, 223].

3.3 Composite monotone inclusions

Numerous relevant problems in mathematical optimization can be modeled as a special class of monotone inclusions in which some maximally monotone operators are composed with linear mappings. These problems are usually referred to as *composite monotone inclusions* and consist in simultaneously solving two inclusions which are closely related to each other.

Problem 3.12 (Composite monotone inclusion). *Let $\mathcal{H}$ and $(\mathcal{G}_j)_{j=1}^m$ be real Hilbert spaces. Let $A_1, \ldots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone operators. Let $B_j : \mathcal{G}_j \rightrightarrows \mathcal{G}_j$*

be maximally monotone and $L_j : \mathcal{H} \to \mathcal{G}_j$ *be a bounded linear operator whose adjoint is denoted by* L_j^**, for all* $j \in \{1, \ldots, m\}$*. The problem consists in solving the primal inclusion*

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^{n} A_i(x) + \sum_{j=1}^{m} L_j^* B_j(L_j x), \tag{3.23}$$

together with its associated dual inclusion

$$\text{find } (u_1, \ldots, u_m) \in \mathcal{G}_1 \times \cdots \times \mathcal{G}_m \text{ such that } (\exists\, x \in \mathcal{H}) \begin{cases} -\sum_{j=1}^{m} L_j^* u_j \in \sum_{i=1}^{n} A_i(x), \\ u_j \in B_j(L_j x), \quad j \in \{1, \ldots, m\}. \end{cases} \tag{3.24}$$

As will be later demonstrated in Section 7.1, the primal and dual inclusions of Problem (3.12) are equivalent in the following sense: the set of solutions of (3.23) is nonempty if there exists a solution to (3.24), and vice versa.

Another particularity of the inclusion (3.23) is that it is not comprised of a sum of maximally monotone operators. Notice that the composition $L_j^* B_j L_j : \mathcal{H} \rightrightarrows \mathcal{H}$ is a monotone operator, but in general it is not maximally monotone. For instance, suppose $\mathcal{H} = \mathbb{R}^2$, take $L = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ and $B = N_{\mathbb{B}_1((0,1)^T)}$. Then $\operatorname{ran}(\mathrm{Id} + L^* B L) = \{0\} \times \mathbb{R}$, which implies that $L^* B L$ is not maximally monotone by Theorem 2.17. This contrasts with the inclusions in Problems 3.7 and 3.9, and urges to consider a new class of splitting schemes. We will discuss about them in Section 3.3.2.

3.3.1 Applications to optimization

We now describe some variational problems expressible as Problem 3.12.

3.3.1.1 Composite convex minimization

Consider the optimization problem given by

$$\min_{x \in \mathcal{H}} f(x) + g(Lx), \tag{3.25}$$

where $f : \mathcal{H} \to \overline{\mathbb{R}}$ and $g : \mathcal{G} \to \overline{\mathbb{R}}$ are proper l.s.c. convex functions and $L : \mathcal{H} \to \mathcal{G}$ is a bounded linear operator. By Theorem 2.27, if $0 \in \operatorname{sri}\big(\operatorname{dom} g - L(\operatorname{dom} f)\big)$, then (3.25) is

equivalent to the monotone inclusion (3.23) with $n = m = 1$,

$$A = \partial f \quad \text{and} \quad B = \partial g.$$

Moreover, if (3.25) has at least one solution, its Fenchel–Rockafellar dual problem reduces to the dual inclusion (3.24) (see [44, Proposition 28.21]), which in this case is expressed as

$$\text{find } u \in \mathcal{G} \text{ such that } 0 \in -LA^{-1}(-L^*u) + B^{-1}(u).$$

In a similar fashion, given some proper l.s.c. convex functions $f_1, \ldots, f_n : \mathcal{H} \to \overline{\mathbb{R}}$ and $g_j : \mathcal{G}_j \to \overline{\mathbb{R}}$, for $j \in \{1, \ldots, m\}$, and some bounded linear operators $L_j : \mathcal{H} \to \mathcal{G}_j$, for $j \in \{1, \ldots, m\}$, we can address the more general problem

$$\min_{x \in \mathcal{H}} \sum_{i=1}^{n} f_i(x) + \sum_{j=1}^{m} g_j(L_j x), \tag{3.26}$$

by setting $A_i = \partial f_i$ and $B_j = \partial g_j$ in Problem 3.12.

3.3.1.2 Saddle-point problems

Let $f : \mathcal{H} \to \overline{\mathbb{R}}$ be a proper l.s.c. convex function and let $(g_j)_{j=1}^m$ and $(L_j)_{j=1}^m$ be as in (3.26). The *minimax problem* given by

$$\min_{x \in \mathcal{H}} \max_{(u_1, \ldots, u_m) \in \mathcal{G}_1 \times \cdots \times \mathcal{G}_m} f(x) + \sum_{j=1}^{m} \langle u_j, L_j x \rangle - \sum_{j=1}^{m} g_j^*(u_j), \tag{3.27}$$

has been extensively studied in the literature, in part due to its applications to multiple areas such as imaging or mechanics [36, 100, 101, 213, 215]. In view of the Fenchel–Moreau theorem, we observe that (3.27) is a primal-dual formulation of its counterpart problem in the form of (3.26), and thus it can be tackled in the same way by means of Problem 3.12. In this framework, the dual inclusion acquires a special importance as it recovers the solution of the maximization problem in (3.27). Indeed, a pair $(\bar{x}, \bar{\mathbf{u}}) \in \mathcal{H} \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$ solves (3.27) if and only if $\bar{x}$ solves (3.23) and $\bar{\mathbf{u}}$ solves (3.24).

3.3.2 Primal-dual splitting methods for composite monotone inclusions

The particular features of Problem 3.12 lead to a series of requirements that should be taken into consideration when designing splitting algorithms for its resolution. Firstly, it is

desirable to simultaneously obtain solutions to both inclusions (3.23) and (3.24), namely, a *primal-dual solution*. Secondly, the lack of maximality of the compositions $L_j^* B_j L_j$ makes the evaluation of its resolvents a non-viable option (see Proposition 2.22 for an exemption). Therefore, the methods should be capable of handling the composed operators separately. More precisely, we will consider schemes which only employ the resolvents of the maximally monotone operators $A_1, \ldots, A_n, B_1, \ldots, B_m$, and forward evaluations of the linear operators $L_1, \ldots, L_m$ and their adjoints $L_1^*, \ldots, L_m^*$. The use of the (generalized) inverse of the linear operators is not recommended as its computational cost is high.

The above are standard conditions satisfied by most of the methods employed to tackle Problem 3.12 (see, e.g., [58, 59, 69, 64, 75, 109, 167, 239, 246]). The term *primal-dual splitting algorithms* is used to encompass all theses algorithmic schemes. In the following, we present some of the most relevant algorithms within this class.

3.3.2.1 The Chambolle–Pock algorithm

The *primal-dual hybrid gradient algorithm* —popularly known as the *Chambolle–Pock algorithm*— is probably one of the most famous methods for tackling Problem 3.12 in the case $n = m = 1$. This algorithm generates a sequence of *primal variables* in $\mathcal{H}$ and a sequence of *dual variables* in $\mathcal{G}$ by making use of the resolvents of A and B^{-1}, respectively. At each variable update, both the primal and dual variables are activated by resorting to evaluations of L or its adjoint. It should be noted that, by Proposition 2.20, evaluating the resolvent of B^{-1} is essentially as challenging as computing the resolvent of B. Hence, we remain in the framework of primal-dual splitting algorihms.

An initial version of the method was proposed by Zhu and Chan [249] for an application in total variation image restoration. Later, multiple modifications for the minimization problem (3.25) were studied by different authors in [115, 136, 147, 207], and by Chambolle and Pock in [100]. Now, the algorithm can be recovered as a particular case of various schemes for addressing the more general Problem 3.12 with monotone operators (see, e.g., [59, 239]). The scheme of the Chambolle–Pock algorithm we present next is obtained as a simplified version of [239, Theorem 3.1].

Theorem 3.13 (Chambolle–Pock algorithm). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ and $B : \mathcal{G} \rightrightarrows \mathcal{G}$ be maximally monotone operators, and let $L : \mathcal{H} \to \mathcal{G}$ be a bounded linear operator. Assume that $\operatorname{zer}(A + L^* B L) \neq \emptyset$. Let γ_1 and γ_2 be positive constants such that $\gamma_1 \gamma_2 \leq \|L\|^2$ and*

$\lambda \in]0,2[$. *Given initial points* $(x^0, u^0) \in \mathcal{H} \times \mathcal{G}$, *consider the iteration*

$$\begin{cases} z^k = J_{\gamma_1 A}(x^k - \gamma_1 L^* u^k), \\ v^k = J_{\gamma_2 B^{-1}}\big(u^k + \gamma_2 L(2z^k - x^k)\big), \\ x^{k+1} = x^k + \lambda(z^k - x^k), \\ u^{k+1} = u^k + \lambda(v^k - u^k), \end{cases} \tag{3.28}$$

for all $k \in \mathbb{N}$. *Then the following hold.*

(i) The sequence $(x^k)_{k\in\mathbb{N}} \subseteq \mathcal{H}$ *converges weakly to a primal solution of Problem 3.12:*

$$x^k \rightharpoonup \bar{x} \in \operatorname{zer}\big(A + L^* B L\big).$$

(ii) The sequence $(u^k)_{k\in\mathbb{N}} \subseteq \mathcal{G}_1$ *converges weakly to a dual solution of Problem 3.12:*

$$u^k \rightharpoonup \bar{u} \in \operatorname{zer}\big(-LA^{-1}(-L^*) + B^{-1}\big).$$

Finally, we refer the interested reader to the papers [191, 196] for a beautiful connection between the Chambolle–Pock algorithm and the Douglas–Rachford splitting.

3.3.2.2 A monotone + skew splitting model

A fair number of splitting schemes developed for Problem 3.12 rely on applying an existing splitting algorithm to a suitable choice of maximally monotone operators. For instance, the method developed by Vũ in [239] has the structure of the forward-backward method, while two different algorithms having its core on the Douglas–Rachford splitting were introduced in [69]. In this section, we present the primal-dual algorithm proposed by Briceño-Arias and Combettes in [75], which is based on an application of the forward-backward-forward method.

To derive this scheme, let us first consider the particular instance of Problem 3.12 in which $n = m = 1$ and define the pair of operators M and N given by

$$\begin{cases} M : \mathcal{H} \times \mathcal{G} \rightrightarrows \mathcal{H} \times \mathcal{G} : (x,u) \mapsto A(x) \times B^{-1}(u), \\ N : \mathcal{H} \times \mathcal{G} \to \mathcal{H} \times \mathcal{G} : (x,u) \mapsto (L^* u, -Lx). \end{cases} \tag{3.29}$$

By Proposition 2.21, the operator M is maximally monotone and N is a skew symmetric bounded linear operator. Further, the set of zeros of the sum $M+N$ consists of primal-dual solutions to Problem 3.12. Applying the forward-backward-forward algorithm to the problem of finding the zeros of $M+N$ generates a sequence $(\mathbf{w}^k)_{k\in\mathbb{N}} := (x^k, u^k)_{k\in\mathbb{N}}$ in $\mathcal{H}\times\mathcal{G}$ given by the fixed point iteration

$$\mathbf{w}^{k+1} = \big(J_{\gamma M}\left(\mathrm{Id}-\gamma N\right) + \gamma N\left(\mathrm{Id}-J_{\gamma M}\left(\mathrm{Id}-\gamma N\right)\right)\big)(\mathbf{w}^k), \quad \forall k\in\mathbb{N}, \tag{3.30}$$

where $\gamma>0$. The weak convergence of the forward-backward-forward method to a zero of $M+N$ leads to the sequence $(x^k)_{k\in\mathbb{N}}$ converging weakly to a solution of (3.23) while $(u^k)_{k\in\mathbb{N}}$ converges weakly to a solution of (3.24).

The general case involving more than two operators can be easily addressed as follows.

Theorem 3.14 (Briceño-Arias–Combettes splitting). *Let $A_1,\dots,A_n : \mathcal{H}\rightrightarrows\mathcal{H}$ and $B_j : \mathcal{G}_j\rightrightarrows\mathcal{G}_j$, for $j\in\{1,\dots,m\}$, be maximally monotone operators. Let $L_j:\mathcal{H}\to\mathcal{G}_j$ be bounded linear operators, for $j\in\{1,\dots,m\}$. Assume* $\operatorname{zer}\left(\sum_{i=1}^n A_i + \sum_{j=1}^m L_j^* B_j L_j\right)\neq\emptyset$. *Take*

$$\gamma \in \left]0, \left((n-1)+\sum_{j=1}^m \|L_j\|^2\right)^{-\frac{1}{2}}\right[,$$

consider initial points $(x_1^0,\dots,x_n^0)\in\mathcal{H}^n$ and $(u_1^0,\dots,u_m^0)\in\mathcal{G}_1\times\cdots\times\mathcal{G}_m$, and set

$$\begin{cases} p_1^k = x_1^k - \gamma\left(\sum_{i=2}^n x_i^k + \sum_{j=1}^m L_j^* u_j^k\right), & \\ z_1^k = J_{\gamma A_1}(p_1^k), & \\ z_i^k = J_{\gamma A_i^{-1}}(x_i^k+\gamma x_1^k), & \forall i\in[\![2,n]\!], \\ v_j^k = J_{\gamma B_j^{-1}}(u_j^k+\gamma L_j x_1^k), & \forall j\in[\![1,m]\!], \\ x_1^{k+1} = x_1^k - p_1^k + z_1^k - \gamma\left(\sum_{i=2}^n z_i^k + \sum_{j=1}^m L_j^* v_j^k\right), & \\ x_i^{k+1} = z_i^k + \gamma(z_1^k - x_1^k), & \forall i\in[\![2,n]\!], \\ u_j^{k+1} = v_j^k + \gamma L_j(z_1^k - x_1^k), & \forall j\in[\![1,m]\!], \end{cases} \tag{3.31}$$

for all $k\in\mathbb{N}$. Then the following statements hold.

(i) The sequence $(x_1^k)_{k\in\mathbb{N}}$ converges weakly to a solution of the primal inclusion (3.23).

(ii) The sequence $(u_1^k,\dots,u_m^k)_{k\in\mathbb{N}}$ converges weakly to a solution of the inclusion (3.24).

Proof. Simply modify the definition of M and N in (3.29) by setting

$$A := A_1, \quad B := A_2 \times \cdots \times A_n \times B_1 \times \cdots \times B_m, \qquad \text{and}$$

$$L : \mathcal{H} \to \mathcal{H}^{n-1} \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m : x \mapsto (x, \overset{(n-1)}{\cdots}, x, L_1 x, \cdots, L_m x).$$

Then the result trivially follows. □

It can be seen that (3.31) is a *full splitting algorithm*, as it only requires evaluations of the resolvents of the maximally monotone operators $A_1, A_2^{-1} \ldots, A_n^{-1}, B_1^{-1}, \ldots, B_m^{-1}$, and of the linear operators and their adjoints.

In addition, it will be important for the sequel to notice that, according to the equivalent formulation (3.30), the scheme in (3.31) is generated by a fixed point iteration of an operator defined in the ambient space $\mathcal{H}^n \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$.

For the last 20 years, the study of monotone operators splitting algorithms has been an active topic of research beyond the mere design of new schemes. Among the most popular lines of investigation we find the development of acceleration techniques [33, 55, 154, 176, 222], the connection with dynamical systems [32, 63], or the analysis of algorithms' behavior in the inconsistent case [51, 190, 221], to name a few. In the next chapter, we explore one of the directions that has attracted the most attention in recent years: the determination of the *minimal dimension* of a splitting scheme.

Chapter 4

Minimal lifting for monotone operator splitting methods

Consider the abstract problem of finding a zero of a monotone operator $\mathcal{M} : \mathcal{H} \rightrightarrows \mathcal{H}$, i.e.,

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \mathcal{M}(x). \tag{4.1}$$

As discussed in Chapter 3, the monotone inclusion (4.1) is considerably broad, as it covers Problems 3.7, 3.9 and 3.12. In this chapter, our main focus will be the underlying algebra of the methods employed for addressing this kind of monotone inclusion. Thus, it will be useful to consider the following unifying approach for devising algorithms for (4.1).

Framework 4.1. *Abstract approach for solving* (4.1)*:*

(i) ***Encoding:*** *Find a real Hilbert space* $\mathcal{X}$*, a* fixed point mapping $\mathcal{T} : \mathcal{X} \to \mathcal{X}$ *and a* solution mapping $\mathcal{S} : \mathcal{X} \to \mathcal{H}$ *such that*

$$\operatorname{Fix} \mathcal{T} \neq \emptyset \iff \operatorname{zer} \mathcal{M} \neq \emptyset \quad \text{and} \quad \mathcal{S}(\operatorname{Fix} \mathcal{T}) \subseteq \operatorname{zer} \mathcal{M}.$$

(ii) ***Iterate:*** *Perform a fixed point iteration of* $\mathcal{T}$ *to obtain a point* $\bar{z} \in \operatorname{Fix} \mathcal{T}$*.*

(iii) ***Solve:*** *Recover a solution to* (4.1) *as* $\mathcal{S}(\bar{z}) \in \operatorname{zer} \mathcal{M}$*.*

Framework 4.1 is general enough, it groups all the methods previously presented as well as the ones that will be introduced in subsequent chapters. For instance, if we consider the classical monotone inclusion with two maximally monotone operators in (3.4), the Douglas–Rachford splitting algorithm is gathered under the above scheme by setting $\mathcal{X} = \mathcal{H}$, $\mathcal{T} = \mathcal{T}_{DR}$ and $\mathcal{S} = J_{\gamma A_1}$ in (i). Therefore, we will say that the Douglas–Rachford splitting is a *fixed point encoding* for (3.4) determined by the pair $(\mathcal{T}_{DR}, J_{\gamma A_1})$ (see Definition 4.2 for details).

The key idea behind Framework 4.1 is very intuitive. In order to find a zero of $\mathcal{M}$, we resort to an operator $\mathcal{T}$ whose set of fixed points is related to $\operatorname{zer}\mathcal{M}$. Further, the solution mapping provides an explicit expression for this connection: from a point $\bar{z} \in \operatorname{Fix}\mathcal{T}$ we recover a solution to the original problem by evaluating $\mathcal{S}(\bar{z})$. Obviously, additional assumptions on $\mathcal{T}$ might be required for being able to obtain a fixed point in step (ii). In practice, one looks for a fixed point operator with nonexpansive properties, but this is not required for the theoretical framework discussed in this chapter. This procedure aligns with the convergence proof of the proximal point algorithm described in Remark 3.2 and is latent in the convergence analysis of the methods studied in the sequel.

In this chapter, we concentrate on the encoding step in Framework 4.1 (i) and pay special attention to $\mathcal{X}$, the space in which $\mathcal{T}$ and $\mathcal{S}$ are defined, which can be understood as the ambient space of an algorithm described by Framework 4.1. We are particularly interested in the dimension of $\mathcal{X}$, which for now we will roughly refer to as *lifting*.

Recall the classical monotone inclusion in Problem 3.7. We say that an algorithm for solving Problem 3.7 has *d-fold lifting* if $\mathcal{X}$ is the d-fold Cartesian product space $\mathcal{H}^d$. When $n = 2$ in Problem 3.7, the Douglas–Rachford operator has 1-fold lifting in view of Remark 3.5. Until very recently, the only way to tackle the problem when $n > 2$ was using Pierra's product space reformulation, which implies an n-fold lifting. Indeed, applying Douglas–Rachford splitting to the operators N_{Δ_n} and $\mathbf{A}$ in Section 3.1.3, we obtain the fixed point iteration of the operator $\mathcal{T}_{PR} : \mathcal{H}^n \to \mathcal{H}^n$ given by

$$\mathcal{T}_{PR}(\mathbf{z}) := \mathbf{z} + \lambda \begin{pmatrix} x_1 - x_0 \\ x_2 - x_0 \\ \vdots \\ x_n - x_0 \end{pmatrix}, \tag{4.2}$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_n) \in \mathcal{H}^n$ and $\mathbf{x} = (x_0, x_1, \ldots, x_n) \in \mathcal{H}^{n+1}$ is the vector defined as

$$\begin{cases} x_0 = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} z_i, \\ x_i = J_{\gamma A_i}(2x_0 - z_i), \quad \forall i \in [\![1, n]\!], \end{cases}$$

for any $\gamma, \lambda > 0$. Note that this scheme is also gathered by Framework 4.1, by defining the solution operator as $\mathcal{S}_{PR} : \mathcal{H}^n \to \mathcal{H} : \mathbf{z} \mapsto (z_1 + \ldots + z_n)/n$. In particular, when $n = 2$, Pierra's product space reformulation not only does not recover the Douglas–Rachford splitting but it has 2-fold lifting, as discussed in Remark 3.8.

Although this reformulation always allows to tackle Problem 3.7 for any finite number of operators, numerical experience shows that it usually slows down the resulting algorithm (see, e.g., [20, Section 6.1] and [81, Section 6.1]), especially when the number of operators is large (see, e.g., [22, Section 4] and [67, Section 5]).

In general, for the abstract monotone inclusion (4.1), the dimension of the underlying space is directly related to the memory requirements of the resulting algorithm. A smaller dimension of $\mathcal{X}$ usually translates into less consumption of computational resources. For this reason, the development of algorithms with *reduced lifting* for solving monotone inclusion problems has recently become an active topic of research (see, e.g., [72, 81, 123, 173, 220, 231, 250]). Besides, different works have been devoted to determine the *minimal lifting*, namely, the minimal dimensional reduction that can be achieved under certain conditions, see [173, 187, 220].

The main contributions of this chapter are the following:

- We survey the existing minimal lifting results for the monotone inclusion problems studied in Chapter 3.
- Specifically, in Theorem 4.9, we extend the minimal lifting theorem for Problem 3.7 proposed by Malitsky and Tam [173] to algorithmic schemes that admit the use of parameters in the evaluation of resolvents.
- In Section 4.3, we formalize the notion of lifting for the composite monotone inclusions in Problem 3.12, and use it to prove the first minimal lifting theorem for primal-dual splitting algorithms in Theorem 4.21.

Except for Fact 4.13, there referenced, the results of this chapter were first developed in [10].

4.1 The case of resolvent splittings

In this section, we present a minimal lifting theorem for Problem 3.7. We start by introducing the definition of *fixed point encoding*, which was already anticipated in Framework 4.1 (i) and is vital for our analysis. More concretely, Definition 4.2 specifies the encoding step in Framework 4.1 in the context of Problem 3.7. Again, we employ $\mathcal{T}$ for denoting a *fixed point operator* and $\mathcal{S}$ a *solution operator*, both depending on the maximally monotone operators appearing in the problem.

Definition 4.2 (Fixed point encoding). *A pair of operators* $(\mathcal{T}, \mathcal{S})$ *is a* fixed point encoding *for Problem 3.7 if, for any particular instance of the problem,*

$$\operatorname{Fix}\mathcal{T} \neq \emptyset \Longleftrightarrow \operatorname{zer}\left(\sum_{i=1}^{n} A_i\right) \neq \emptyset \text{ and } \mathbf{z} \in \operatorname{Fix}\mathcal{T} \Longrightarrow \mathcal{S}(\mathbf{z}) \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i\right).$$

As shown in Section 3.1, the resolvents of the operators $A_1, A_2, \ldots, A_n$ are the fundamental blocks used for building splitting algorithms for addressing Problem 3.7. A fixed point encoding that only makes use of resolvents is known as a *resolvent splitting*.

Definition 4.3 (Resolvent splitting). *A fixed point encoding* $(\mathcal{T}, \mathcal{S})$ *for Problem 3.7 is a* resolvent splitting *if, for any particular instance of the problem, there is a finite procedure that evaluates* $\mathcal{T}$ *and* $\mathcal{S}$ *at a given point which only uses vector addition, scalar multiplication, and the resolvents of* $A_1, \ldots, A_n$.

In general, resolvents are not effortless to compute, even when closed formulas are known. If the aim is to improve the efficiency of an algorithm, it is desirable to properly limit the number of resolvent evaluations per iteration. This motivates the concept of *frugal resolvent splitting*.

Definition 4.4 (Frugality). *A resolvent splitting* $(\mathcal{T}, \mathcal{S})$ *for Problem 3.7 is* frugal *if, in addition, each of the resolvents of* $A_1, \ldots, A_n$ *is used exactly once.*

We now present the precise definition of lifting for fixed point encodings of Problem 3.7.

Definition 4.5 (Lifting). *Let* $d \in \mathbb{N} \setminus \{0\}$. *A fixed point encoding* $(\mathcal{T}, \mathcal{S})$ *is a* d-fold lifting *for Problem 3.7 if* $\mathcal{T} : \mathcal{H}^d \to \mathcal{H}^d$ *and* $\mathcal{S} : \mathcal{H}^d \to \mathcal{H}$.

For concision, we will say that a splitting algorithm has d-fold lifting if it can be described as a fixed point encoding with d-fold lifting.

At this point, we consider convenient to provide some comments on the history of minimal lifting for classical monotone inclusions. The concept of lifting in the framework of splitting algorithms was introduced by Ryu [220] in 2020. In addition, he was the first to consider the notion of resolvent splitting as an important property for studying the lifting reduction of an algorithm. More precisely, Ryu proved the uniqueness of the Douglas–Rachford splitting as a frugal resolvent splitting having 1-fold lifting for solving Problem 3.7 with $n = 2$. The definitions of these concepts were later refined by Malitsky and Tam [173] to generalize Ryu's result to the case $n > 2$. In [173, Theorem 1], they

proved that the minimal lifting that can be achieved for Problem 3.7 with frugal resolvent splittings is $n-1$.

In both works [173] and [220], the definition of resolvent splitting considered differs from the one in Definition 4.3, as they do not allow including parameters in the resolvents (i.e., it only permits computation of the resolvents $J_{A_1}, \ldots, J_{A_n}$). This definition comprises the Douglas–Rachford algorithm and the scheme derived from Pierra's product space reformulation. Since in these methods there is no upper bound for the stepsize parameter, it can always be set to $\gamma = 1$. Nonetheless, their framework leaves aside various well-known splitting methods (see, for instance, Example 4.6 below). Anticipating the discussion on minimal lifting for Problem 3.9, the inclusion of resolvent parameters is also fundamental for controlling the constants of cocoercivity, as well as the norms of the linear operators in Problem 3.12.

From the proof of Malitsky and Tam, it cannot be directly determined whether their result holds when the resolvents are allowed to have different parameters. Here we present the adaptation of the minimality theorem in [173, Theorem 1] to the more general parametrized setting. In the original publication of this work (see [10]), the authors opted for using the term *parameterized resolvent splitting* to make this distinction clear. However, we believe that the appropriate framework for this analysis involves the admission of different parameters in the resolvents. Therefore, we omit the term parametrized in Definition 4.3 for simplicity. Nonetheless, the reasoning below is very similar and follows the lines of [173, Section 3].

Example 4.6 (Campoy's minimal lifting splitting algorithm). *Slightly prior to the work of Malitsky and Tam, one of the first resolvent splittings with minimal lifting was derived by Campoy [81] by resorting to a product space reformulation with reduced dimension. This reformulation was actually employed earlier for a completely different purpose by Kruger [156], where he devised necessary conditions for extreme points of collections of closed sets. Given any $\gamma > 0$ and $\lambda \in]0, 2[$, the reformulation considered by Campoy applied to Problem 3.7 leads to the algorithm in [81, Theorem 5.1] (see also [54, 82, 117]), defined by the operator $\mathcal{T} : \mathcal{H}^{n-1} \to \mathcal{H}^{n-1}$ given by*

$$\mathcal{T}(\mathbf{z}) := \mathbf{z} + \lambda \begin{pmatrix} x_1 - x_0 \\ x_2 - x_0 \\ \vdots \\ x_{n-1} - x_0 \end{pmatrix},$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_{n-1})$ *and* $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1}) \in \mathcal{H}^n$ *is the vector defined as*

$$\begin{cases} x_0 = J_{\frac{\gamma}{n-1}A_n}\left(\dfrac{1}{n-1}\displaystyle\sum_{i=1}^{n-1} z_i\right), \\ x_i = J_{\gamma A_i}(2x_0 - z_i), \quad \forall i \in [\![1, n-1]\!]. \end{cases}$$

Moreover, given the operator $\mathcal{S} : \mathcal{H}^{n-1} \to \mathcal{H}$ *defined as*

$$\mathcal{S}(\mathbf{z}) := J_{\frac{\gamma}{n-1}A_n}\left(\frac{1}{n-1}\sum_{i=1}^{n-1} z_i\right),$$

then the pair $(\mathcal{T}, \mathcal{S})$ *is a frugal resolvent splitting with* $(n-1)$*-fold lifting according to Definition 4.3, but it is not covered by the setting of [173], as different parameters are required for the resolvents of* $A_1, \ldots, A_{n-1}$ *and* A_n*.*

In the following, we assume that $n \geq 2$. Let us also denote by $\mathcal{A}_n$ the collection of all n-tuples of maximally monotone operators in $\mathcal{H}$. Hence, an element $\mathbf{A} \in \mathcal{A}_n$ is of the form $\mathbf{A} = (A_1, \ldots, A_n)$, where $A_i : \mathcal{H} \rightrightarrows \mathcal{H}$ are maximally monotone operators for all $i \in [\![1, n]\!]$. Every instance of Problem 3.7 is determined by the choice of $\mathbf{A} \in \mathcal{A}_n$. In particular, when considering a fixed point encoding for this problem, the fixed point operator and the solution operator are both parametrized in terms of $\mathbf{A}$. To emphasize this idea, and to facilitate the exposition, we denote these operators by $\mathcal{T}_{\mathbf{A}}$ and $\mathcal{S}_{\mathbf{A}}$ in the following.

Let $(\mathcal{T}_{\mathbf{A}}, \mathcal{S}_{\mathbf{A}})$ be a d-fold lifted frugal resolvent splitting for Problem 3.7. By definition, there exists a finite procedure for evaluating $\mathcal{T}_{\mathbf{A}}$ and $\mathcal{S}_{\mathbf{A}}$ using only vector addition, scalar multiplication and the resolvents $J_{\gamma_1 A_1}, \ldots, J_{\gamma_n A_n}$ precisely once, where $\gamma = (\gamma_1, \ldots, \gamma_n)^T$ is a vector of positive stepsize parameters. This suggests that the evaluation of a point $\mathbf{z} = (z_1, \ldots, z_d) \in \mathcal{H}^d$ by $\mathcal{T}_{\mathbf{A}}$ is subject to certain algebraic rules. In order to deduce them, we proceed as follows.

First of all, since the resolvents are expected to be employed, there should exist points $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{H}^n$ and $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{H}^n$ such that

$$\mathbf{x} = J_{\gamma \mathbf{A}}(\mathbf{y}) \iff 0 \in \mathbf{x} - \mathbf{y} + \gamma \mathbf{A}(\mathbf{x}), \tag{4.3}$$

where $\gamma \mathbf{A} := (\gamma_1 A_1, \ldots, \gamma_n A_n) \in \mathcal{A}_n$.

According to the above equation, each resolvent $J_{\gamma_i A_i}$ is evaluated at the point y_i (which is not specified yet) to get the point $x_i = J_{\gamma_i A_i}(y_i)$. Since apart from resolvent

evaluations only sums and scalar products are allowed, each y_i should have been obtained as a linear combination of vectors already computed in the process. Assuming without loss of generality that the resolvents of $A_1, A_2, \ldots, A_n$ are calculated in order, we get that

$$\begin{aligned} y_i \in\ & \operatorname{span}\{z_1, z_2, \ldots, z_d, x_1, x_2, \ldots, x_{i-1}, y_1, y_2, \ldots, y_{i-1}\} \\ =\ & \operatorname{span}\{z_1, z_2, \ldots, z_d, x_1, x_2, \ldots, x_{i-1}\}, \end{aligned}$$

for all $i \in [\![1, n]\!]$. Equivalently, this can be compactly expressed by the equation

$$\mathbf{y} = Y_z \mathbf{z} + Y_x \mathbf{x}, \tag{4.4}$$

where $Y_z \in \mathbb{R}^{n\times d}$ and $Y_x \in \mathbb{R}^{n\times n}$ is a lower-triangular matrix with zeros in the diagonal[1]. Finally, by frugality, there exists $T_z \in \mathbb{R}^{d\times d}$ and $T_x \in \mathbb{R}^{d\times n}$ such that

$$\mathcal{T}_{\mathbf{A}}(\mathbf{z}) = T_z \mathbf{z} + T_x \mathbf{x}. \tag{4.5}$$

Similarly, also by frugality, the evaluation of $\mathbf{z}$ by the solution operator $\mathcal{S}_{\mathbf{A}}$ can be expressed as

$$\mathcal{S}_{\mathbf{A}}(\mathbf{z}) = S_z \mathbf{z} + S_x \mathbf{x}, \tag{4.6}$$

where $S_z \in \mathbb{R}^{1\times d}$ and $S_x \in \mathbb{R}^{1\times n}$.

On the whole, equations (4.3)-(4.6) completely describe any frugal resolvent splitting $(\mathcal{T}_{\mathbf{A}}, \mathcal{S}_{\mathbf{A}})$. In the following technical lemma, these equations are employed to characterize the fixed points of the operator $\mathcal{T}_{\mathbf{A}}$.

Lemma 4.7. *Let $(\mathcal{T}_{\mathbf{A}}, \mathcal{S}_{\mathbf{A}})$ be a frugal resolvent splitting for Problem 3.7. Let M denote the block matrix given by*

$$M := \begin{bmatrix} 0 & \mathrm{Id} & -\mathrm{Id} & \gamma^T \mathrm{Id} \\ Y_z & Y_x & -\mathrm{Id} & 0 \\ T_z - \mathrm{Id} & T_x & 0 & 0 \end{bmatrix}.$$

If $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{\mathbf{A}}$, then there exists $\bar{\mathbf{v}} = [\bar{\mathbf{z}}, \bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{a}}]^T \in \ker M$ with $\bar{\mathbf{a}} \in \mathbf{A}(\bar{\mathbf{x}})$. Conversely, if $\bar{\mathbf{v}} = [\bar{\mathbf{z}}, \bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{a}}]^T \in \ker M$ and $\bar{\mathbf{a}} \in \mathbf{A}(\bar{\mathbf{x}})$, then it holds that $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{\mathbf{A}}$, $\bar{\mathbf{x}} = J_{\gamma \mathbf{A}}(\bar{\mathbf{y}})$ and $\mathcal{S}_{\mathbf{A}}(\bar{\mathbf{z}}) = S_z \bar{\mathbf{z}} + S_x \bar{\mathbf{x}}$.

[1] Here we are making use of an abuse of notation. Strictly speaking, (4.4) should be written as the equality $\mathbf{y} = (Y_z \otimes \mathrm{Id})\mathbf{z} + (Y_x \otimes \mathrm{Id})\mathbf{x}$, where $\otimes$ denotes the Kronecker product.

Proof. Let $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{\mathbf{A}}$. Then (4.3) and (4.4) imply the existence of $\bar{\mathbf{x}}, \bar{\mathbf{y}} \in \mathcal{H}^n$ and $\bar{\mathbf{a}} \in \mathbf{A}(\bar{\mathbf{x}})$ such that

$$0 = \bar{\mathbf{x}} - \bar{\mathbf{y}} + \gamma \bar{\mathbf{a}} \quad \text{and} \quad \bar{\mathbf{y}} = Y_z \bar{\mathbf{z}} + Y_x \bar{\mathbf{x}}.$$

Together with the fact that $\bar{\mathbf{z}} = T_z \bar{\mathbf{z}} + T_x \bar{\mathbf{x}}$, this yields that $\bar{\mathbf{v}} = [\bar{\mathbf{z}}, \bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{a}}]^T \in \ker M$.

The converse implication follows similarly. The fact that $\mathcal{S}_{\mathbf{A}}(\bar{\mathbf{z}}) = S_z \bar{\mathbf{z}} + S_x \bar{\mathbf{x}}$ is implied by equations (4.3)-(4.5). □

Proposition 4.8 (Solution operator). *Let $(\mathcal{T}_{\mathbf{A}}, \mathcal{S}_{\mathbf{A}})$ be a frugal resolvent splitting for Problem 3.7. Let $\bar{z} \in \operatorname{Fix} \mathcal{T}_{\mathbf{A}}$ and $\bar{\mathbf{v}} = [\bar{\mathbf{z}}, \bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{a}}]^T \in \ker M$, where M is the block matrix defined in the above lemma. Then*

$$\mathcal{S}_{\mathbf{A}}(\bar{\mathbf{z}}) = \frac{1}{n} \sum_{i=1}^{n} (\bar{y}_i - \gamma_i \bar{a}_i) = \bar{x}_1 = \cdots = \bar{x}_n, \tag{4.7}$$

with $\bar{\mathbf{a}} \in \mathbf{A}(\bar{\mathbf{x}})$ and $\bar{\mathbf{x}} = J_{\gamma \mathbf{A}}(\bar{\mathbf{y}})$.

Proof. Consider a particular instance of Problem 3.7 given by some operators $\mathbf{A} \in \mathcal{A}_n$. Let $\mathcal{T}_{\mathbf{A}}$ and $\mathcal{S}_{\mathbf{A}}$ be the fixed point and the solution operators of this particular instance, respectively. Let $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{\mathbf{A}}$. The existence of $\bar{\mathbf{v}} := [\bar{\mathbf{z}}, \bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{a}}]^T \in \ker M$ is guaranteed by Lemma 4.7, which in addition ensures that $\bar{\mathbf{a}} \in \mathbf{A}(\bar{\mathbf{x}})$ and $\bar{\mathbf{x}} = J_{\gamma \mathbf{A}}(\bar{\mathbf{y}})$. Therefore we only need to prove the equalities in (4.7).

Define $x^* := \mathcal{S}_{\mathbf{A}}(\bar{\mathbf{z}})$. By (4.6), we have that $x^* = S_z(\bar{\mathbf{z}}) + S_x(\bar{\mathbf{x}})$. Consider now the $n+1$ instances of Problem 3.7 given by the n-tuples of maximally monotone operators $\mathbf{A}^{(0)}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(n)} \in \mathcal{A}_n$ defined as

$$\mathbf{A}^{(0)}(\mathbf{x}) := \bar{\mathbf{a}} \quad \text{and} \quad \mathbf{A}^{(j)}(\mathbf{x}) := \bar{\mathbf{a}} + \begin{bmatrix} 0 \\ \vdots \\ x_j - \bar{x}_j \\ \vdots \\ 0 \end{bmatrix}, \quad \forall j \in [\![1, n]\!].$$

Since $\bar{\mathbf{v}} \in \ker M$ and $\bar{\mathbf{a}} = \mathbf{A}^{(j)}(\bar{\mathbf{x}})$, for all $j \in [\![0, n]\!]$, Lemma 4.7 implies that $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{\mathbf{A}^{(j)}}$, $\bar{\mathbf{x}} = J_{\gamma \mathbf{A}^{(j)}}(\bar{\mathbf{y}})$ and thus, $S_{\mathbf{A}^{(j)}}(\bar{\mathbf{z}}) = S_z \bar{\mathbf{z}} + S_x \bar{\mathbf{x}} = x^*$ is a solution to every instance. Therefore, we have $0 = \sum_{i=1}^n A_i^{(0)}(x^*) = \sum_{i=1}^n \bar{a}_i$. Hence,

$$0 = \sum_{i=1}^{n} A_i^{(j)}(x^*) = \sum_{i=1}^{n} \bar{a}_i + x^* - \bar{x}_j = x^* - \bar{x}_j, \quad \forall j \in [\![1, n]\!],$$

from where it follows that $x^* = \bar{x}_1 = \cdots = \bar{x}_n$. Finally, since $\bar{\mathbf{x}} = J_{\gamma \mathbf{A}^{(0)}}(\bar{\mathbf{y}})$, we have that $\bar{\mathbf{y}} - \bar{\mathbf{x}} = \gamma \mathbf{A}^{(0)}(\bar{\mathbf{x}}) = (\gamma_1 \bar{a}_1, \ldots, \gamma_n \bar{a}_n)$. Consequently, $\sum_{i=1}^n \bar{y}_i - n x^* = \sum_{i=1}^n \gamma_i \bar{a}_i$, which completes the proof. □

Note that, although the expression for the solution operator given by (4.7) differs from the one obtained in [173, Proposition 1], it still holds that the vector $\bar{\mathbf{x}}$ belongs to the diagonal subspace Δ_n. This is what we employ to prove the minimal lifting theorem for frugal resolvent splittings.

Theorem 4.9 (Minimal lifting for frugal resolvent splittings). *Let $(\mathcal{T}_{\mathbf{A}}, \mathcal{S}_{\mathbf{A}})$ be a frugal resolvent splitting with d-fold lifting for Problem 3.7 with $n \geq 2$. Then $d \geq n - 1$.*

Proof. Suppose, by contradiction, that $(\mathcal{T}_{\mathbf{A}}, \mathcal{S}_{\mathbf{A}})$ is a frugal resolvent splitting for Problem 3.7 with d-fold lifting such that $d \leq n - 2$. Consider a particular instance of the problem given by $\mathbf{A} \in \mathcal{A}_n$ such that $\operatorname{zer}(\sum_{i=1}^n A_i) \neq \emptyset$ and take $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{\mathbf{A}}$. By Lemma 4.7, there exists $\bar{\mathbf{v}} := [\bar{\mathbf{z}}, \bar{\mathbf{x}}, \bar{\mathbf{y}}, \bar{\mathbf{a}}]^T \in \ker M$ with $\bar{\mathbf{a}} \in \mathbf{A}(\bar{\mathbf{x}})$. The last row of M implies that $0 = (T_z - \mathrm{Id})\bar{\mathbf{z}} + T_x \bar{\mathbf{x}}$. Since $T_x \in \mathbb{R}^{d \times n}$ and $d \leq n - 2$, by the rank-nullity theorem, $\dim \ker T_x = n - \dim \operatorname{rank} T_x \geq n - d \geq 2$. Since the dimension of Δ_n as a subspace of $\mathcal{H}^n$ is 1, there exists $\tilde{\mathbf{x}} \in \mathcal{H}^n \setminus \Delta_n$ such that $T_x \bar{\mathbf{x}} = T_x \tilde{\mathbf{x}}$.

Now, set $\tilde{\mathbf{z}} := \bar{\mathbf{z}}$, $\tilde{\mathbf{y}} := Y_z \tilde{\mathbf{z}} + Y_x \tilde{\mathbf{x}}$ and $\tilde{\mathbf{a}} := ((\tilde{y}_1 - \tilde{x}_1)/\gamma_1, \ldots, (\tilde{y}_n - \tilde{x}_n)/\gamma_n)$ and consider the instance of the problem given by $\tilde{\mathbf{A}} \in \mathcal{A}_n$ defined as $\tilde{\mathbf{A}}(\mathbf{s}) := \tilde{\mathbf{a}}$ for all $\mathbf{s} \in \mathcal{H}^n$. Then, $\tilde{\mathbf{v}} := [\tilde{\mathbf{z}}, \tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{a}}]^T \in \ker M$ with $\tilde{\mathbf{a}} = \tilde{\mathbf{A}}(\tilde{\mathbf{x}})$. By Lemma 4.7 and Proposition 4.8, this implies that $\tilde{\mathbf{x}} \in \Delta_n$, obtaining thus a contradiction which completes the proof. □

In accordance with Theorem 4.9, the Douglas–Rachford algorithm is a resolvent splitting with minimal lifting for Problem 3.7 with $n = 2$. In contrast, Pierra's product space reformulation leads to a frugal resolvent splitting which is not minimal, as it has n-fold lifting. Besides Douglas–Rachford, only recently various frugal resolvent splittings with minimal lifting have been developed. The first of them was Ryu's splitting (see [220] and [20, Appendix A]) for $n = 3$, which has 2-fold lifting for Problem 3.7. For an arbitrary finite number n of operators, Campoy's splitting [81], the Malitsky–Tam splitting (see Remark 6.10) and the methods in [72], [187, Theorem 8.1] and [231] describe frugal resolvent splittings with $(n - 1)$-fold lifting.

4.2 The case of forward-backward splittings

We now draw our attention to the monotone inclusion in Problem 3.9. Recall that this problem amounts to finding a zero in the sum of n maximally monotone operators and m single-valued operators that could be cocoercive or monotone and Lipschitz. Nonetheless, the minimal lifting theorem presented here does not rely on cocoercivity nor Lipschitz continuity and just assumes the operators $T_1, \ldots, T_m$ to be single-valued.

The characterization of the minimal lifting for Problem 3.9 is due to Morin, Banert and Giselsson [187], who proved under some mild assumptions that the lifting of the methods for addressing this problem has the same lower bound than that of frugal resolvent splittings. In essence, their result shows that including forward evaluations of single-valued operators in resolvent splittings does not alter the minimal dimension of the algorithm's underlying space.

Definition 4.10 (Fixed point encoding). *A pair of operators $(\mathcal{T}, \mathcal{S})$ is a fixed point encoding for Problem 3.9 if, for any particular instance of the problem,*

$$\operatorname{Fix}\mathcal{T} \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{j=1}^{m} T_j\right) \neq \emptyset \text{ and } \mathbf{z} \in \operatorname{Fix}\mathcal{T} \implies \mathcal{S}(\mathbf{z}) \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{j=1}^{m} T_j\right).$$

The definition of lifting for fixed point encoding for Problem 3.9 coincides with Definition 4.5, so we do not replicate it here.

Definition 4.11 (Frugal forward-backward splitting). *A fixed point encoding $(\mathcal{T}, \mathcal{S})$ for Problem 3.9 is a* frugal forward-backward splitting *if, for any particular instance of the problem, there is a finite procedure that evaluates $\mathcal{T}$ and $\mathcal{S}$ at a given point which only uses vector addition, scalar multiplication, and the resolvents of $A_1, \ldots, A_n$ and forward evaluations of $T_1, \ldots, T_m$ exactly once.*

In the following remark we determine the lifting of the forward-backward methods introduced in Section 3.2. For simplicity, we again set $T := \sum_{j=1}^{m} T_j$ and note that an evaluation of T amounts to an evaluation of each one of the operators $T_1, \ldots, T_m$.

Remark 4.12 (Lifting of forward-backward methods). (i): By Lemma 5.1 below, the pair $(\mathcal{T}_{DY}^{\gamma}, J_{\gamma A_1})$ determines a fixed point encoding for the Davis–Yin algorithm. Hence, Davis–Yin is a 1-fold frugal forward-backward splitting for Problem 3.9 with $n = 2$. As a particular case, the forward-backward algorithm also has 1-fold lifting. For Problem 3.9 with $n \geq 2$, the product space reformulation in Section 3.2.2.3 leads to a fixed point

encoding for the generalized forward-backward, which is frugal and has n-fold lifting in view of Theorem 3.10.
(ii): At each iteration, the forward-reflected-backward method makes use of two previously computed iterates. This urges to include an additional variable in order to express the algorithm as a fixed point iteration. The sequence in (3.22) can be written as the fixed point iteration of the operator $\mathcal{T} : \mathcal{H}^2 \to \mathcal{H}^2$ given by

$$\mathcal{T}\begin{pmatrix} x \\ y \end{pmatrix} := \begin{pmatrix} J_{\gamma A}(x - 2\gamma T(x) - \gamma y) \\ T(x) \end{pmatrix}.$$

It is easy to check that together with the immersion $\mathcal{S} : \mathcal{H}^2 \to \mathcal{H} : (x, y) \mapsto x$, the operator $\mathcal{T}$ determines a fixed point encoding for the forward-reflected-backward. In addition, note that T is only evaluated once per iteration. Hence, the forward-reflected-backward is a frugal forward-backward splitting with 2-fold lifting for Problem 3.9 with $n = 1$.
(iii): A fixed point encoding for Tseng's forward-backward method is given by the operators $\mathcal{T} : \mathcal{H} \to \mathcal{H}$ and $\mathcal{S} : \mathcal{H} \to \mathcal{H}$ defined as

$$\mathcal{S}(x) := J_{\gamma A}\big(x - \gamma T(x)\big) \quad \textit{and} \quad \mathcal{T}(x) := S(x) - \gamma T(S(x)) + \gamma T(x).$$

This makes the forward-backward-forward to have 1-fold lifting. However, $(\mathcal{T}, \mathcal{S})$ is not a frugal forward-backward splitting, as T is evaluated twice, specifically at x and $S(x)$. As noted in [187], Tseng's method is not expressible as a frugal forward-backward splitting.

Fact 4.13 (Minimal lifting for frugal forward-backward splittings, [187, Corollary 6.7]). *Let $(\mathcal{T}, \mathcal{S})$ be a frugal forward-backward splitting with d-fold lifting for Problem 3.9 with $n \geq 2$. Then $d \geq n - 1$.*

Fact 4.13 and Remark 4.12 imply that the lifting of the Davis–Yin splitting is minimal for Problem 3.9 with $n = 2$. In contrast, the generalized forward-backward method of Theorem 3.10 does not have minimal lifting. Together with the proof of Fact 4.13, the authors in [187] also proposed a frugal forward-backward splitting with $(n-1)$-fold lifting for solving Problem 3.9 with $n \geq 2$ and when the single-valued operators are exclusively cocoercive. When $n = 3$, the convergence of a 2-fold lifting forward-backward splitting which extends Ryu's algorithm has been investigated in [250]. These were not the first methods with minimal lifting for this class of problems. Previously, the authors in [25] devised a forward-backward type scheme with minimal lifting that has the advantage of being implementable in decentralized networks. We shall present it in Chapter 6.

To the best of the author's knowledge, there are no known frugal forward-backward splittings with minimal lifting capable of solving Problem 3.9 when the single-valued operators are monotone and Lipschitz continuous. When $n = 2$, some frugal resolvent splitting methods are noted (see, e.g., the *forward-reflected-Douglas–Rachford* [223] and the methods in [212]). However, these algorithms suffer from the same pathology than the forward-reflected-backward (see, Remark 4.12 (ii)), which urges to resort to a 3-fold lifting to express them as a fixed point iteration. The *forward-Douglas–Rachford-forward* [223] has 1-fold lifting and solves the problem with $n = 2$, but it is not frugal as it requires two forward evaluations of the Lipschitz continuous operator. In Chapter 6, we present a method which solves the general problem with $n \geq 2$ and has $(n-1)$-fold lifting, but also two evaluations of the monotone and Lipschitz continuous operator are needed per iteration.

4.3 The case of primal-dual resolvent splittings

We now concentrate on establishing the minimal lifting reduction for the composite monotone inclusion in Problem 3.12. We recall the importance of being able to simultaneously identify solutions to both the primal inclusion (3.23) and the dual inclusion (3.24). This motivates the following definition of fixed point encoding for Problem 3.12.

Definition 4.14 (Primal-dual fixed point encoding). *A pair of operators $(\mathcal{T}, \mathcal{S})$ is a* primal-dual fixed point encoding *for Problem 3.12 if, for any particular instance of the problem,*

$$\operatorname{Fix}\mathcal{T} \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{j=1}^{m} L_j^* B_j L_j\right) \neq \emptyset \text{ and } \mathbf{w} \in \operatorname{Fix}\mathcal{T} \Longrightarrow \mathcal{S}(\mathbf{w}) \in \mathbf{Z},$$

where $\boldsymbol{Z}$ is the set of primal-dual solutions *of Problem 3.12 defined as*

$$\mathbf{Z} := \left\{(x, u_1, \ldots, u_m) \in \mathcal{H} \times \mathcal{G}_1 \times \ldots \times \mathcal{G}_m \;\middle|\; -\sum_{j=1}^{m} L_j^* u_j \in \sum_{i=1}^{n} A_i(x) \text{ and } u_j \in B_j(L_j x),\ \forall j \in [\![1, m]\!]\right\}.$$

Observe that if $(x, u_1, \ldots, u_m) \in \mathbf{Z}$, then x solves (3.23) while $(u_1, \ldots, u_m)$ solves (3.24), the converse being also true. This will be discussed in detail in Section 7.1.

When talking about lifting for primal-dual problems, the need to distinguish between variables in the space of primal solutions and dual solutions arises. Hence, we introduce the notion of *primal-dual lifting*.

Definition 4.15 (Primal-dual lifting). *Let $d, f \in \mathbb{N}$. A primal-dual fixed point encoding (T, S) is a* (d, f)-fold lifting *for Problem 3.12 if*

$$T : \mathcal{H}^d \times \mathcal{G}_1^{f_1} \times \cdots \times \mathcal{G}_m^{f_m} \to \mathcal{H}^d \times \mathcal{G}_1^{f_1} \times \cdots \times \mathcal{G}_m^{f_m}$$

and

$$S : \mathcal{H}^d \times \mathcal{G}_1^{f_1} \times \cdots \times \mathcal{G}_m^{f_m} \to \mathcal{H} \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m,$$

where $f_j \geq 0$ for all $j \in [\![1, m]\!]$ and $f = \sum_{j=1}^m f_j$. We adopt the convention that the space $\mathcal{H}$ (respectively $\mathcal{G}_j$) vanishes from the equation when $d = 0$ (respectively $f_j = 0$).

For the sake of brevity, we will say that a primal-dual splitting has (d, f)-fold lifting if it can be described as a primal-dual fixed point encoding with (d, f)-fold lifting.

We now extend the definitions of resolvent splitting and frugality to the context of composite monotone inclusions. The first of them is inspired by the standard conditions satisfied by primal-dual splitting algorithms discussed at the beginning of Section 3.3.2.

Definition 4.16 (Primal-dual resolvent splitting). *A fixed point encoding $(\mathcal{T}, \mathcal{S})$ for Problem 3.12 is a* primal-dual resolvent splitting *if, for any particular instance of the problem, there is a finite procedure that evaluates $\mathcal{T}$ and $\mathcal{S}$ at a given point which only uses vector addition, scalar multiplication, the resolvents of $A_1, \ldots A_n$ and $B_1, \ldots, B_m$, and forward evaluations of $L_1, \ldots, L_m$ and their adjoints.*

As previously mentioned, the evaluation of the resolvents of $B_1^{-1}, \ldots, B_m^{-1}$ is common in primal-dual algorithmic schemes. Since it is comparable to computing the resolvents of $B_1, \ldots, B_m$ (see Proposition 2.20), their use is also included in the definition of primal-dual resolvent splitting.

Definition 4.17 (Frugality). *A primal-dual resolvent splitting $(\mathcal{T}, \mathcal{S})$ for Problem 3.12 is* frugal *if, in addition, each of the resolvents of $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ is used exactly once.*

Remark 4.18 (On the absence of restrictions on the evaluation of the linear operators). Since in the finite case, a forward evaluation of a linear operator is computationally equivalent to performing vector addition and scalar multiplication, this suggests that for practical applications there is no computational need to control the number of evaluations of the linear operators in the definition of frugality.

Example 4.19 (Chambolle–Pock as a primal-dual resolvent splitting). *Consider Chambolle–Pock's algorithm presented in Theorem 3.13 for Problem 3.12 with $n = m = 1$.*

The iteration (3.28) can be written as the fixed point iteration defined by the operator $\mathcal{T} : \mathcal{H} \times \mathcal{G} \to \mathcal{H} \times \mathcal{G}$ *given as*

$$\mathcal{T}\begin{pmatrix} x \\ u \end{pmatrix} := \begin{pmatrix} x \\ u \end{pmatrix} + \lambda \begin{pmatrix} z - x \\ v - u \end{pmatrix},$$

where

$$\begin{cases} z = J_{\gamma_1 A}(x - \gamma_1 L^* u), \\ v = J_{\gamma_2 B^{-1}}(u - \gamma_2 L(2z - x)). \end{cases}$$

It is a simple exercise to check that $(\bar{x}, \bar{u}) \in \operatorname{Fix} \mathcal{T}$ *if and only if* $(\bar{x}, \bar{u}) \in \mathbf{Z}$*. Hence, defining the solution operator* $\mathcal{S}$ *as the identity mapping in the space* $\mathcal{H} \times \mathcal{G}$*, the Chambolle–Pock method becomes a frugal primal-dual resolvent splitting with* $(1, 1)$*-fold lifting.*

Example 4.20 (Briceño-Arias–Combettes method as a primal-dual resolvent splitting). *In the setting of Problem 3.12 with an arbitrary number of operators, the Briceño-Arias–Combettes algorithm is derived from a suitable reformulation of Tseng's forward-backward-forward. In view of Remark 4.12 (iii) and Section 3.3.2.2, this method can be expressed as primal-dual fixed point encoding. Furthermore, the resulting scheme is a frugal primal-dual resolvent splitting with* (n, m)*-fold lifting, as shown in (3.31).*

The following result characterizes the minimal lifting of frugal primal-dual resolvent splitting algorithms with m dual variables.

Theorem 4.21 (Minimality theorem for frugal primal-dual resolvent splittings). *Let* $(\mathcal{T}, \mathcal{S})$ *be a frugal primal-dual resolvent splitting for Problem 3.12 with* (d, m)*-fold lifting. Then, if* $n \geq 2$*, necessarily* $d \geq n - 1$*.*

Proof. By way of contradiction, let $(\mathcal{T}, \mathcal{S})$ be a frugal primal-dual resolvent splitting for Problem 3.12 with (d, m) fold lifting and $d < n - 1$. Consider the instance of the problem in which $L_j = \mathrm{Id} : \mathcal{H} \to \mathcal{H}$, for all $j \in [\![1, m]\!]$. Then, Problem 3.12 becomes the classical monotone inclusion problem with $n+m$ operators and $(\mathcal{T}, \mathcal{S})$ is a frugal resolvent splitting with $(d + m)$-fold lifting for such problem with $d + m < n + m - 1$, which contradicts Theorem 4.9. □

The lifting of the Briceño-Arias–Combettes algorithm is not minimal, as it has (n, m)-fold lifting. Although, the above theorem is not applicable to Chambolle–Pock's method (it is applied with $n = 1$), its extensions to the general Problem 3.12 in [59, 116, 239] also have (n, m)-fold lifting. We will introduce the first primal-dual resolvent splitting with minimal lifting in Chapter 7.

Finally, note that Theorem 4.21 only gives a lower bound for the lifting of the primal variables, provided that the lifting of the dual variables coincides with the number of composed operators in Problem 3.12. It remains open the question of whether it is possible to reduce the dimension of the underlying space associated to the linearly composed operators.

4.4 A brief note on the convergence of fixed point encodings

In this chapter, we have solely concentrated on the encoding step in Framework 4.1, which aims to describe an algorithm as a fixed point iteration of a suitable operator. Nevertheless, being a fixed point encoding is not sufficient for guaranteeing convergence of an algorithm. A motivating example is the forward-backward method, which is a fixed point encoding for the problem of finding a zero in the sum of a maximally monotone operator and a monotone and Lipschitz continuous operator, but might fail to converge to a solution of the problem. A central topic in the following chapters will be the convergence analysis of algorithms in the context of Framework 4.1. We now make a brief preview of what is to come.

Let us consider a fixed point encoding $(\mathcal{T}, \mathcal{S})$ of a monotone inclusion problem associated to an operator $\mathcal{M}$. If $\mathcal{T}$ is averaged nonexpansive, the convergence of a sequence $(z^k)_{k\in\mathbb{N}}$, obtained by performing a fixed point iteration or a Krasnosel'skiĭ–Mann iteration of $\mathcal{T}$, to a point in $\operatorname{Fix}\mathcal{T}$ directly follows from Theorem 2.7 and Proposition 2.8, respectively. However, in some cases $\mathcal{T}$ has milder nonexpansive properties (for instance, it can be strongly quasi-nonexpansive), which prevents from resorting to Theorem 2.7 or Proposition 2.8. Likewise, the proof of convergence of the *shadow sequence* $(\mathcal{S}(z^k))_{k\in\mathbb{N}}$ to a zero of $\mathcal{M}$ cannot be deduced by applying tools from fixed point theory. In both cases, a more sophisticated analysis is required, which involves further techniques from monotone operator theory.

Chapter 5

A new proof of convergence of Davis–Yin splitting algorithm allowing larger stepsizes

This chapter is devoted to the study of the Davis–Yin splitting algorithm, already introduced in Section 3.2.2.2. The algorithm is designed for solving the monotone inclusion

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in (A_1 + A_2 + T)(x), \tag{5.1}$$

where all three operators involved are maximally monotone and act on the Hilbert space $\mathcal{H}$, and T is also $\frac{1}{\beta}$-cocoercive, for $\beta > 0$. In order to analyze the convergence of the sequence governing the algorithm, Davis and Yin defined the operator

$$\mathcal{T}_{DY}^{\gamma} := J_{\gamma A_2} \circ (2J_{\gamma A_1} - \mathrm{Id} - \gamma T \circ J_{\gamma A_1}) + \mathrm{Id} - J_{\gamma A_1}, \tag{5.2}$$

which we denote as $\mathcal{T}_{DY}^{\gamma}$ to emphasize the dependence on γ. Subsequently, they proved that $\mathcal{T}_{DY}^{\gamma}$ is α-averaged for $\alpha = \frac{2}{4-\gamma\beta}$ when $\gamma \in]0, 2/\beta[$. Observing that the sequence $(z^k)_{k\in\mathbb{N}}$ in (3.16) (where now the relaxation parameters may vary per iteration) is obtained through the standard Krasnosel'skiĭ–Mann iteration

$$z^{k+1} = (1 - \lambda_k)z^k + \lambda_k \mathcal{T}_{DY}^{\gamma}(z^k), \quad \forall k \in \mathbb{N}, \tag{5.3}$$

with $\lambda_k \in]0, 1/\alpha]$, the convergence of $(z^k)_{k\in\mathbb{N}}$ to a fixed point of $\mathcal{T}_{DY}^{\gamma}$ follows if the remaining assumptions of Proposition 2.8 hold. Further, the *shadow sequence* $\left(J_{\gamma A_1}(z^k)\right)_{k\in\mathbb{N}}$ weakly converges to a solution to (5.1), and convergence is strong under additional assumptions.

The main contributions of this chapter are summarized as follows:

- In Theorem 5.3, we provide a new proof of convergence of the iterative method (5.3)

without relying on the averagedness of the operator $\mathcal{T}_{DY}^{\gamma}$. The proof has two key advantages: (i) it permits to simplify the assumptions on the relaxation parameters, and (ii) it allows to choose the stepsize γ in $]0, 4/\beta[$ instead of $]0, 2/\beta[$.

Observe that the operator $\mathcal{T}_{DY}^{\gamma}$ does not need to be averaged when $\gamma > 2/\beta$ (for instance, take $A_1 = A_2 = 0$, T the identity, and apply $\mathcal{T}_{DY}^{\gamma}$ to the points $x = 1$ and $z = -1$). As a by-product, this shows that the stepsize in the forward-backward and the backward-forward algorithms can be also chosen in $]0, 4/\beta[$.

- In Theorem 5.10, we derive a *strengthened* version of the Davis–Yin splitting algorithm which permits to compute the resolvent of $A_1 + A_2 + T$.
- We provide multiple numerical experiments illustrating the importance of appropriately choosing the stepsize and the relaxation parameters of the algorithm.

Otherwise stated, the results in this chapter were extracted from [26].

5.1 The importance of properly adjusting the parameters of an algorithm

Let us present a simple motivating example of the significance of the algorithm parameters. Consider the problem of finding the minimum norm point in the intersection of two balls $\mathbb{A}$ and $\mathbb{B}$ in the Euclidean space whose intersection has nonempty interior. The problem can be solved with the Davis–Yin splitting algorithm, by taking A_1 and A_2 as the normal cones to the respective balls, and T as the identity mapping. Since the resolvents of the normal cones are the projectors (see Example 2.26), which we denote by $P_{\mathbb{A}}$ and $P_{\mathbb{B}}$, the iterative scheme is given by

$$z^{k+1} = z^k - \lambda_k P_{\mathbb{A}}(z^k) + \lambda_k P_{\mathbb{B}}\left((2-\gamma)P_{\mathbb{A}}(z^k) - z^k\right), \quad \forall k \in \mathbb{N},$$

and $\left(P_{\mathbb{A}}(z^k)\right)_{k\in\mathbb{N}}$ converges to the minimum norm point in $\mathbb{A} \cap \mathbb{B}$ (the normal cone sum rule holds by Theorem 2.27). Both the relaxation parameter λ_k and the stepsize γ have a big influence on the behavior of the algorithm, as shown in Figure 5.1.

In this example, since the cocoercivity constant $\frac{1}{\beta}$ is equal to 1, [128, Theorem 2.1] guarantees the convergence when the parameter γ is taken in $]0, 2[$, while Theorem 5.3 below allows to take $\gamma \in]0, 4[$. When the Davis–Yin splitting algorithm is applied to the

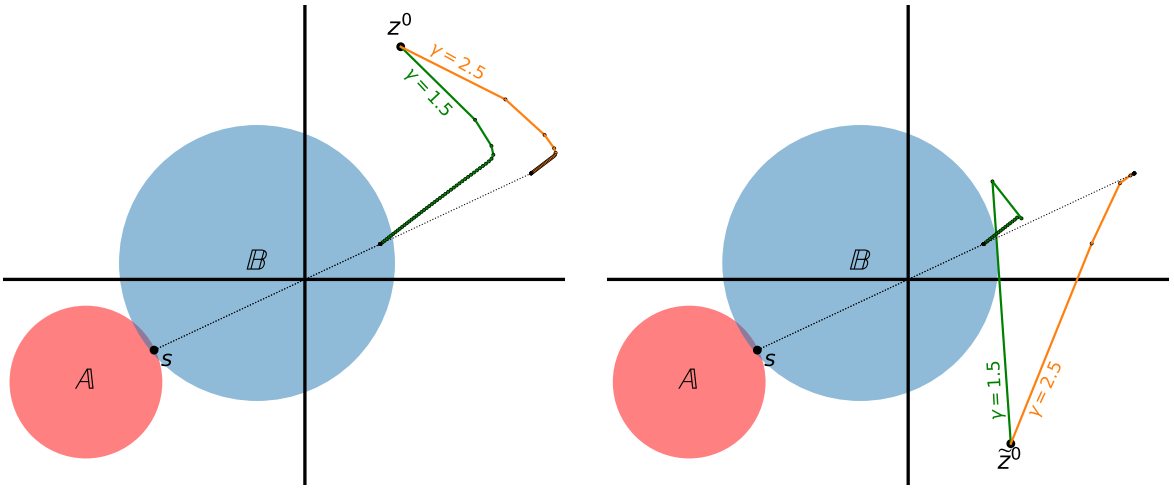


FIGURE 5.1: Behavior of Davis–Yin splitting algorithm for two starting points z^0 and $\tilde{z}^0$ and two stepsize parameters γ, with $\lambda_k = 0.99(2 - \gamma/2)$. The solution s is obtained after projecting the fixed point onto $\mathbb{A}$.

same problem with different starting points z^0, it can behave very differently depending on the parameters, as shown in Figures 5.1 and 5.2.

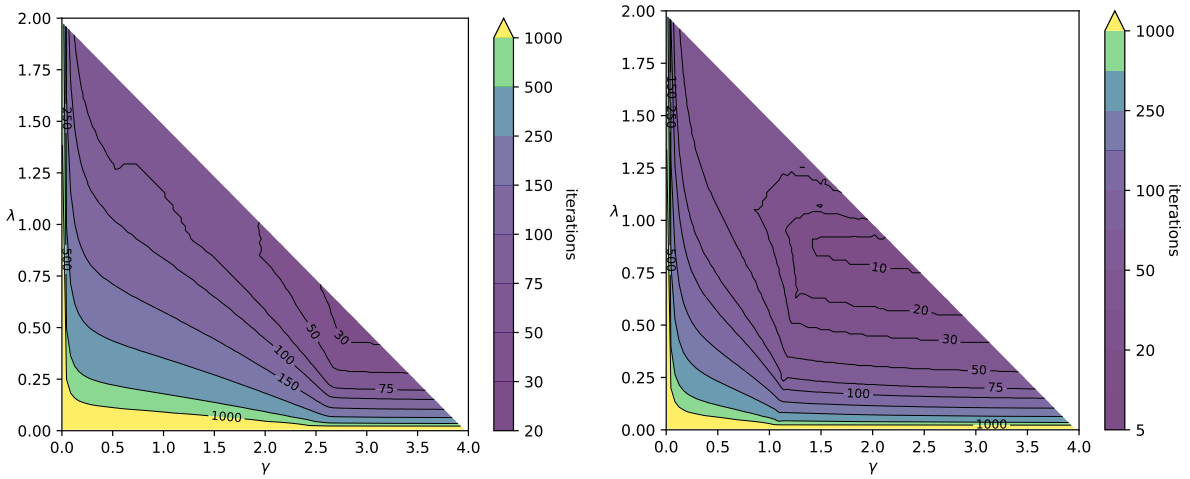


FIGURE 5.2: Number of iterations needed until the shadow sequence gets sufficiently close to the solution s (precisely, $\|P_{\mathbb{A}}(z^k) - s\| < 10^{-10}$) for different values of γ and $\lambda_k = \lambda$, with starting points z^0 (left) and $\tilde{z}^0$ (right) shown in Figure 5.1.

In general, larger stepsizes are commonly believed to be associated with faster convergence of algorithms, but this is not always the case, particularly when an algorithm has several parameters. It is important to have in mind that the relaxation parameter λ_k of the Davis–Yin splitting algorithm is upper bounded by $2 - \frac{\gamma\beta}{2}$ and that its value has an

important effect. If $\gamma \in]0, 2/\beta[$, *overrelaxed steps* (i.e., $\lambda_k > 1$) are allowed in (5.3), while only *underrelaxed steps* can be taken when $\gamma \geq \frac{2}{\beta}$. The fact that both the stepsize and the relaxation parameters are significant is especially apparent when one considers the particular case of $A_1 = A_2 = 0$ and $T = \nabla f$ for a differentiable function f whose gradient is β-Lipschitz continuous. In this case, the iteration (5.3) reduces to the gradient descent scheme:

$$z^{k+1} = z^k - \gamma\lambda_k \nabla f(z^k), \quad \forall k \in \mathbb{N}. \tag{5.4}$$

We observe in (5.4) that the stepsize of the algorithm is actually $\gamma\lambda_k$, so the upper bound $2 - \frac{\gamma\beta}{2}$ on the relaxation parameters λ_k entails $\gamma\lambda_k < \frac{2}{\beta}$, as expected.

Finally, it is important to recall that in practical applications only a lower bound of the best cocoercivity constant $\frac{1}{\beta}$ is usually known, and this can affect the performance of the algorithms. For instance, consider again the application of the Davis–Yin algorithm with starting point $\widetilde{z}_0$ shown on the right in Figure 5.2 and imagine that we underestimate $1/\beta$ to $1/\widehat{\beta} = 0.65 < 1 = 1/\beta$. Then we observe in Figure 5.3 how the choice of a stepsize parameter $\gamma \in]0, 2/\widehat{\beta}[$ excludes better values like $\widehat{\gamma} \in]2/\widehat{\beta}, 4/\widehat{\beta}[$.

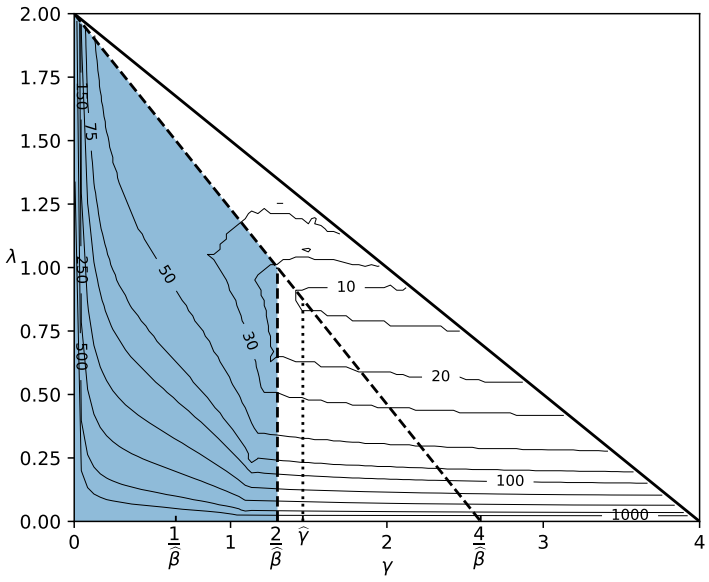


FIGURE 5.3: Repetition of the experiment shown on the right of Figure 5.2. When only an approximate value $\frac{1}{\widehat{\beta}}$ of the cocoercivity constant is known, choosing the stepsize $\gamma \in]0, 2/\widehat{\beta}[$ (shaded area) can exclude better choices like $\widehat{\gamma}$.

A typical choice for the parameters of the forward-backward algorithm is $\gamma = (2-\varepsilon)/\beta$ and $\lambda_k = 1$, for a small $\varepsilon > 0$ (see, e.g., [117]). This example shows that, when only an

estimate $\frac{1}{\widehat{\beta}}$ of the best value of $\frac{1}{\beta}$ is known, it can be worth testing the performance of the algorithm with parameters $\gamma = (2+\varepsilon)/\widehat{\beta}$ and $\lambda_k = 1-\varepsilon$ (i.e., with underrelaxation).

5.2 The Davis–Yin splitting algorithm

The following lemma characterizes the set of zeros of the sum of operators in (5.1) in terms of the set

$$\Omega_\gamma := \big\{z \in \mathcal{H} : J_{\gamma A_1}(z) = J_{\gamma A_2}\big(2J_{\gamma A_1}(z) - z - \gamma T(J_{\gamma A_1}(z))\big)\big\}, \tag{5.5}$$

with $\gamma > 0$, and shows that $\Omega_\gamma = \operatorname{Fix} \mathcal{T}_{DY}^\gamma$, where

$$\operatorname{Fix} \mathcal{T}_{DY}^\gamma = \{x + \gamma y : x \in \operatorname{zer}(A_1 + A_2 + T), y \in (-A_2(x) - T(x)) \cap A_1(x)\}, \tag{5.6}$$

as shown in [128, Lemma 2.2]. Furthermore, it reveals that the pair $(\mathcal{T}_{DY}^\gamma, J_{\gamma A_1})$ determines a fixed point encoding for the Davis–Yin algorithm.

Lemma 5.1. *For every $\gamma > 0$, it holds*

$$\operatorname{zer}(A_1 + A_2 + T) = J_{\gamma A_1}(\Omega_\gamma).$$

In particular, $\operatorname{zer}(A_1 + A_2 + T) \neq \emptyset \iff \Omega_\gamma \neq \emptyset$. *Further,* $\Omega_\gamma = \operatorname{Fix} \mathcal{T}_{DY}^\gamma$.

Proof. Observe that

$$\begin{aligned}
x \in \operatorname{zer}(A_1 + A_2 + T) &\Leftrightarrow -\gamma T(x) \in (\gamma A_1 + \gamma A_2)(x), \\
&\Leftrightarrow (\exists\, z \in \mathcal{H}) \quad z - x \in \gamma A_1(x), \quad x - z - \gamma T(x) \in \gamma A_2(x), \\
&\Leftrightarrow (\exists\, z \in \mathcal{H}) \quad x = J_{\gamma A_1}(z), \quad 2x - z - \gamma T(x) \in (\mathrm{Id} + \gamma A_2)(x), \\
&\Leftrightarrow (\exists\, z \in \mathcal{H}) \quad x = J_{\gamma A_1}(z), \quad x = J_{\gamma A_2}(2x - z - \gamma T(x)),
\end{aligned}$$

from where the first claim follows. Further, we have

$$\begin{aligned}
z \in \Omega_\gamma &\Leftrightarrow (\exists\, x \in \operatorname{zer}(A_1 + A_2 + T)) \quad x = J_{\gamma A_1}(z), \quad x = J_{\gamma A_2}(2x - z - \gamma T(x)), \\
&\Leftrightarrow (\exists\, x \in \operatorname{zer}(A_1 + A_2 + T)) \quad z - x \in \gamma A_1(x), \quad z - x \in (-\gamma A_2(x) - \gamma T(u)), \\
&\Leftrightarrow (\exists\, x \in \operatorname{zer}(A_1 + A_2 + T))\, (\exists\, y \in (-A_2(x) - T(x)) \cap A_1(x)) \quad z = x + \gamma y,
\end{aligned}$$

and thus, $\Omega_\gamma = \operatorname{Fix} \mathcal{T}_{DY}^\gamma$, by (5.6). □

Using a technique similar to the one employed in [20, Theorem 8], we can provide a direct proof of the convergence of Davis–Yin splitting algorithm with the additional advantages of both allowing a larger stepsize and having a simpler condition on the relaxation parameters than [128, Theorem 2.1]. The proof makes use of the following technical lemma.

Lemma 5.2. *Let $A_1, A_2 : \mathcal{H} \rightrightarrows \mathcal{H}$ be two maximally monotone operators and $T : \mathcal{H} \to \mathcal{H}$. Let $z, \hat{z} \in \mathcal{H}$ and $\gamma > 0$, and set $x := J_{\gamma A_1}(z)$, $\hat{x} := J_{\gamma A_1}(\hat{z})$, $u := J_{\gamma A_2}(2x - z - \gamma T(x))$ and $\hat{u} := J_{\gamma A_2}(2\hat{x} - \hat{z} - \gamma T(\hat{x}))$. Then it holds*

$$0 \leq \langle z - \hat{z}, (x - u) - (\hat{x} - \hat{u})\rangle - \|(x - u) - (\hat{x} - \hat{u})\|^2 - \gamma\langle T(x) - T(\hat{x}), u - \hat{u}\rangle. \tag{5.7}$$

Further, if A_1 (respectively A_2) is uniformly monotone with modulus ϕ, then (5.7) holds with 0 replaced by $\gamma\phi(\|x - \hat{x}\|)$ (respectively $\gamma\phi(\|u - \hat{u}\|)$).

Proof. Since $z - x \in \gamma A_1(x)$ and $\hat{z} - \hat{x} \in \gamma A_1(\hat{x})$, monotonicity of γA_1 yields

$$0 \leq \langle (z - x) - (\hat{z} - \hat{x}), x - \hat{x}\rangle. \tag{5.8}$$

Likewise, since $2x - z - \gamma T(x) - u \in \gamma A_2(u)$ and $2\hat{x} - \hat{z} - \gamma T(\hat{x}) - \hat{u} \in \gamma A_2(\hat{u})$, monotonicity of γA_2 implies

$$\begin{aligned} 0 &\leq \langle (2x - z - \gamma T(x) - u) - (2\hat{x} - \hat{z} - \gamma T(\hat{x}) - \hat{u}), u - \hat{u}\rangle \\ &= \langle (\hat{u} - \hat{x}) - (u - x), u - \hat{u}\rangle - \langle (z - x) - (\hat{z} - \hat{x}), u - \hat{u}\rangle - \gamma\langle T(x) - T(\hat{x}), u - \hat{u}\rangle. \end{aligned} \tag{5.9}$$

Summing together (5.8) and (5.9), we obtain

$$\begin{aligned} 0 &\leq \langle (z - x) - (\hat{z} - \hat{x}), (x - u) - (\hat{x} - \hat{u})\rangle + \langle (\hat{u} - \hat{x}) - (u - x), u - \hat{u}\rangle - \gamma\langle T(x) - T(\hat{x}), u - \hat{u}\rangle \\ &= \langle z - \hat{z}, (x - u) - (\hat{x} - \hat{u})\rangle - \|(x - u) - (\hat{x} - \hat{u})\|^2 - \gamma\langle T(x) - T(\hat{x}), u - \hat{u}\rangle, \end{aligned}$$

which proves (5.7). The last assertion easily follows from the definition of uniform monotonicity. □

5.2.1 A new proof of convergence of the Davis–Yin algorithm

In this section, we demonstrate the new convergence theorem for the Davis–Yin splitting and sketch its consequences for the forward-backward algorithm. The following is the main result of this chapter.

Theorem 5.3 (Davis–Yin splitting algorithm). *Let $A_1, A_2 : \mathcal{H} \rightrightarrows \mathcal{H}$ be two maximally monotone operators and $T : \mathcal{H} \to \mathcal{H}$ be a $\frac{1}{\beta}$-cocoercive operator, with $\beta > 0$, such that $\operatorname{zer}(A_1 + A_2 + T) \neq \emptyset$. Set a stepsize $\gamma \in]0, 4/\beta[$ and consider a sequence of relaxation parameters $(\lambda_k)_{k\in\mathbb{N}}$ in $\left]0, 2 - \frac{\gamma\beta}{2}\right]$ such that $\sum_{k\in\mathbb{N}} \lambda_k \left(2 - \frac{\gamma\beta}{2} - \lambda_k\right) = +\infty$. Given some initial point $z^0 \in \mathcal{H}$, consider the sequences defined by*

$$\begin{cases} x^k = J_{\gamma A_1}(z^k), \\ u^k = J_{\gamma A_2}\big(2x^k - z^k - \gamma T(x^k)\big), \\ z^{k+1} = z^k + \lambda_k(u^k - x^k), \end{cases} \tag{5.10}$$

for all $k \in \mathbb{N}$. Then the sequence $(z^k)_{k\in\mathbb{N}}$ is Fejér monotone with respect to the set Ω_γ given in (5.5). Moreover, the following assertions hold.

(i) $z^k \rightharpoonup \bar{z} \in \Omega_\gamma$, $x^k \rightharpoonup \bar{x}$, $u^k \rightharpoonup \bar{x}$, $u^k - x^k \to 0$ and $T(x^k) \to T(\bar{x})$ with

$$\bar{x} = J_{\gamma A_1}(\bar{z}) = J_{\gamma A_2}\big(2\bar{x} - \bar{z} - \gamma T(\bar{x})\big) \in \operatorname{zer}(A_1 + A_2 + T).$$

Further, $T(\operatorname{zer}(A_1 + A_2 + T)) = \{T(\bar{x})\}$.

(ii) If either A_1 or A_2 is uniformly monotone on every bounded subset of its domain, or T is demiregular at every point in $\operatorname{zer}(A_1 + A_2 + T)$, then $(x^k)_{k\in\mathbb{N}}$ and $(u^k)_{k\in\mathbb{N}}$ converge strongly to $\bar{x} \in \operatorname{zer}(A_1 + A_2 + T)$.

Proof. Define the sequences

$$v^k := \gamma T(x^k) \qquad \text{and} \qquad w^k := u^k - x^k, \quad \forall k \in \mathbb{N},$$

and note the following relations that (5.10) yields

$$(x^k, z^k - x^k) \in \operatorname{gra} \gamma A_1 \quad \text{and} \quad (u^k, 2x^k - z^k - v^k - u^k) \in \operatorname{gra} \gamma A_2. \tag{5.11}$$

Select any $z \in \Omega_\gamma$ and denote $x := J_{\gamma A_1}(z)$. By the definition of Ω_γ, we have that $x = J_{\gamma A_2}(2x - z - \gamma T(x))$. Applying Lemma 5.2 to z and $\hat{z} := z^k$, observing that $\hat{x} = x^k$, $u = x$ and $\hat{u} = u^k$, yields

$$0 \leq \langle z - z^k, w^k \rangle - \|w^k\|^2 - \gamma \langle T(x) - T(x^k), x - u^k \rangle. \tag{5.12}$$

The first two terms in (5.12) multiplied by $2\lambda_k$ can be expressed as

$$\begin{aligned} 2\lambda_k\left(\langle z - z^k, w^k\rangle - \|w^k\|^2\right) &= 2\langle z - z^k, z^{k+1} - z^k\rangle - 2\lambda_k\|w^k\|^2 \\ &= \|z^k - z\|^2 - \|z^{k+1} - z\|^2 + \lambda_k(\lambda_k - 2)\|w^k\|^2. \end{aligned} \tag{5.13}$$

Now, using the $\frac{1}{\beta}$-cocoercivity of T, the last term in (5.12) can be expressed as

$$\begin{aligned} -\gamma\langle T(x) - T(x^k), x - u^k\rangle &= -\gamma\langle T(x) - T(x^k), x - x^k\rangle + \gamma\langle T(x) - T(x^k), w^k\rangle \\ &\leq -\frac{\gamma}{\beta}\|T(x) - T(x^k)\|^2 + \gamma\langle T(x) - T(x^k), w^k\rangle. \end{aligned} \tag{5.14}$$

Using Cauchy–Schwarz and Young's inequalities, the last term in (5.14) can be estimated as

$$\gamma\langle T(x) - T(x^k), w^k\rangle \leq \frac{\gamma}{\beta}\|T(x) - T(x^k)\|^2 + \frac{\gamma\beta}{4}\|w^k\|^2. \tag{5.15}$$

Combining (5.12)-(5.15), we have

$$\|z^{k+1} - z\|^2 + \lambda_k(2 - \lambda_k)\|w^k\|^2 \leq \|z^k - z\|^2 + \frac{\gamma\beta}{2}\lambda_k\|w^k\|^2.$$

As a result, we reach the expression

$$\|z^{k+1} - z\|^2 + \lambda_k\left(2 - \frac{\gamma\beta}{2} - \lambda_k\right)\|w^k\|^2 \leq \|z^k - z\|^2. \tag{5.16}$$

Since $\lambda_k \leq 2 - \gamma\beta/2$, equation (5.16) implies that $(z^k)_{k\in\mathbb{N}}$ is Fejér monotone with respect to Ω_γ and thus, bounded. Since resolvents are nonexpansive and T is β-Lipschitz continuous (by Cauchy–Schwarz), it follows that $(x^k)_{k\in\mathbb{N}}$, $(v^k)_{k\in\mathbb{N}}$ and $(u^k)_{k\in\mathbb{N}}$ are bounded.

(i): The Fejér monotonicity of $(z^k)_{k\in\mathbb{N}}$ implies that the sequence $(\|z^k - z\|)_{k\in\mathbb{N}}$ is nonincreasing and convergent. Telescoping (5.16), we obtain

$$\sum_{k\in\mathbb{N}} \lambda_k\left(2 - \frac{\gamma\beta}{2} - \lambda_k\right)\|w^k\|^2 \leq \|z^0 - z\|^2,$$

which implies $\liminf_{k\to\infty}\|w^k\| = 0$, since $\sum_{k\in\mathbb{N}}\lambda_k\left(2 - \frac{\gamma}{2\beta} - \lambda_k\right) = +\infty$. To prove that $w^k \to 0$, it suffices to show that the sequence $(\|w^k\|)_{k\in\mathbb{N}}$ is nonincreasing. Applying Lemma 5.2 with $z := z^{k+1}$ and $\hat{z} := z^k$ yields

$$0 \leq \langle z^{k+1} - z^k, w^k - w^{k+1}\rangle - \|w^{k+1} - w^k\|^2 - \gamma\langle T(x^{k+1}) - T(x^k), u^{k+1} - u^k\rangle.$$

The first two terms multiplied by 2 can be expressed as

$$2\langle \lambda_k w^k, w^k - w^{k+1}\rangle - 2\|w^{k+1} - w^k\|^2 = \lambda_k^2\|w^k\|^2 - \|w^{k+1} - w^k\|^2 - \|w^{k+1} - w^k + \lambda_k w^k\|^2,$$

while the third term is equal to

$$\begin{aligned}
&-\gamma\langle T(x^{k+1}) - T(x^k), u^{k+1} - u^k\rangle \\
&\qquad = -\gamma\langle T(x^{k+1}) - T(x^k), w^{k+1} - w^k\rangle - \gamma\langle T(x^{k+1}) - T(x^k), x^{k+1} - x^k\rangle \\
&\qquad \le \frac{\gamma}{\beta}\|T(x^{k+1}) - T(x^k)\|^2 + \frac{\gamma\beta}{4}\|w^{k+1} - w^k\|^2 - \frac{\gamma}{\beta}\|T(x^{k+1}) - T(x^k)\|^2 \\
&\qquad = \frac{\gamma\beta}{4}\|w^{k+1} - w^k\|^2,
\end{aligned}$$

where we have used again Young's inequality and the cocoercivity of T. Therefore, we deduce

$$\begin{aligned}
0 &\le \lambda_k^2\|w^k\|^2 - \left\|w^{k+1} - w^k + \lambda_k w^k\right\|^2 + \left(\frac{\gamma\beta}{2} - 1\right)\|w^{k+1} - w^k\|^2 \\
&= \lambda_k^2\|w^k\|^2 - \lambda_k^2\|w^k\|^2 + 2\lambda_k\langle w^{k+1} - w^k, -w^k\rangle + \left(\frac{\gamma\beta}{2} - 2\right)\|w^{k+1} - w^k\|^2 \\
&= \lambda_k\|w^k\|^2 - \lambda_k\|w^{k+1}\|^2 + \left(\frac{\gamma\beta}{2} - 2 + \lambda_k\right)\|w^{k+1} - w^k\|^2,
\end{aligned}$$

that is,

$$\lambda_k\|w^{k+1}\|^2 \le \lambda_k\|w^k\|^2 - \left(2 - \frac{\gamma\beta}{2} - \lambda_k\right)\|w^{k+1} - w^k\|^2 \le \lambda_k\|w^k\|^2,$$

so $(\|w^k\|)_{k\in\mathbb{N}}$ is nonincreasing, since $\lambda_k > 0$. Hence, we have proved that $w^k \to 0$.

Let $(\bar{z}, \bar{x}, \bar{v})$ be a weak sequential cluster point of the bounded sequence $(z^k, x^k, v^k)_{k\in\mathbb{N}}$. Hence, there is a subsequence $(z^{k_n}, x^{k_n}, v^{k_n})_{n\in\mathbb{N}}$ which is weakly convergent to $(\bar{z}, \bar{x}, \bar{v})$. Now, consider the operator $C : \mathcal{H}^3 \rightrightarrows \mathcal{H}^3$ given by

$$C := \begin{pmatrix} (\gamma A_1)^{-1} \\ (\gamma T)^{-1} \\ \gamma A_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 & -\mathrm{Id} \\ 0 & 0 & -\mathrm{Id} \\ \mathrm{Id} & \mathrm{Id} & 0 \end{pmatrix},$$

which is maximally monotone, because it is the sum of a maximally monotone operator and a skew-symmetric matrix (see Example 2.13 and Proposition 2.15). From (5.11), it

follows that

$$\begin{pmatrix} x^{k_n} - u^{k_n} \\ x^{k_n} - u^{k_n} \\ x^{k_n} - u^{k_n} \end{pmatrix} \in C \begin{pmatrix} z^{k_n} - x^{k_n} \\ v^{k_n} \\ u^{k_n} \end{pmatrix}.$$

Since by Proposition 2.23 the graph of a maximally monotone operator is sequentially closed in the weak-strong topology, taking the limit as $n \to \infty$ and noting $z^{k_n} - x^{k_n} \rightharpoonup \bar{z} - \bar{x}$ and $u^{k_n} \rightharpoonup \bar{x}$ (since $w^{k_n} = u^{k_n} - x^{k_n} \to 0$), we deduce that

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \in \left(\begin{pmatrix} (\gamma A_1)^{-1} \\ (\gamma T)^{-1} \\ \gamma A_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 & -\mathrm{Id} \\ 0 & 0 & -\mathrm{Id} \\ \mathrm{Id} & \mathrm{Id} & 0 \end{pmatrix} \right) \begin{pmatrix} \bar{z} - \bar{x} \\ \bar{v} \\ \bar{x} \end{pmatrix}.$$

The latter inclusion is equivalent to

$$\bar{x} = J_{\gamma A_1}(\bar{z}), \quad \bar{v} = \gamma T(\bar{x}) \quad \text{and} \quad \bar{x} = J_{\gamma A_2}(2\bar{x} - \bar{z} - \bar{v}), \tag{5.17}$$

which implies $\bar{z} \in \Omega_\gamma$. Therefore, every weak sequential cluster point of $(z^k)_{k\in\mathbb{N}}$ is contained in Ω_γ, and Proposition 2.6 implies that $(z^k)_{k\in\mathbb{N}}$ is weakly convergent to a point $\bar{z} \in \Omega_\gamma$. Then (5.17) shows that $\bar{x} = J_{\gamma A_1}(\bar{z})$ and $\bar{v} = \gamma T(\bar{x})$ are the unique cluster points of $(x^k)_{k\in\mathbb{N}}$ and $(v^k)_{k\in\mathbb{N}}$, respectively, and hence $x^k \rightharpoonup \bar{u}$, $u^k \rightharpoonup \bar{x}$ and $v^k \rightharpoonup \bar{v}$.

Moreover, since z was arbitrarily chosen in Ω_γ, (5.12) and (5.14) also hold with x replaced by $\bar{x}$ and z replaced by $\bar{z}$. From the resulting inequalities, we obtain

$$\begin{aligned} \frac{\gamma}{\beta}\|T(\bar{x}) - T(x^k)\|^2 \leq{}& \langle \bar{z} - z^k, w^k \rangle + \langle x^k - \bar{x}, w^k \rangle \\ &+ \langle \bar{x} - u^k, w^k \rangle + \gamma \langle T(\bar{x}) - T(x^k), w^k \rangle, \end{aligned}$$

and thus $T(x^k) \to T(\bar{x})$. Now, by Lemma 5.1, we know that $\bar{x} \in \operatorname{zer}(A_1 + A_2 + T)$.

Finally, we conclude that $T(\operatorname{zer}(A_1 + A_2 + T)) = \{T(\bar{x})\}$ due to the uniqueness of solution of the Attouch–Théra dual problem of (5.1).

(ii): Assume first that A_1 is uniformly monotone. Since the sequence $(x^k)_{k\in\mathbb{N}}$ is bounded, the set $\{\bar{x}\} \cup \{x^k, k \geq 0\} \subset \operatorname{dom} A_1$ is bounded. Thus, using uniform monotonicity in Lemma 5.2 with $z := \bar{z}$ and $\hat{z} := z^k$, we obtain the stronger inequality

$$\gamma\phi(\|\bar{x} - x^k\|) \leq \langle \bar{z} - z^k, w^k \rangle - \|w^k\|^2 - \gamma \langle T(\bar{x}) - T(x^k), \bar{x} - u^k \rangle,$$

which entails $\gamma\phi(\|\bar{x} - x^k\|) \to 0$. Since ϕ is increasing, we deduce that $x^k \to \bar{x}$, which implies $u^k \to \bar{x}$. When A_2 is uniformly monotone, the result similarly follows.

Finally, suppose that the demiregularity assumption holds. Since (i) yields $x^k \rightharpoonup \bar{x}$ and $T(x^k) \to T(\bar{x})$, the demiregularity of T at $\bar{x}$ then implies that $x^k \to \bar{x}$. Since $u^k - x^k \to 0$, we also obtain that $u^k \to \bar{x}$. □

REMARK 5.4. (i) The stepsize γ in [128, Theorem 2.1] is assumed to be in $]0, 2\varepsilon/\beta[$, with $\varepsilon \in]0, 1[$, while Theorem 5.3 allows to take stepsizes in the interval $]0, 4/\beta[$, which is twice larger. Note that our assumption is required to guarantee that $2 - \gamma\beta/2 > 0$. The relaxation parameters $(\lambda_k)_{k\in\mathbb{N}}$ in [128, Theorem 2.1] must be taken in $]0, 2 - \varepsilon[$, while the interval given in Theorem 5.3 is $]0, 2 - \frac{\gamma\beta}{2}]$. If $\gamma \in]0, 2\varepsilon/\beta[$, we have $2 - \varepsilon < 2 - \gamma\beta/2$. Thus, Theorem 5.3 additionally allows to take some of the relaxation parameters equal to $2 - \gamma\beta/2$ (but not all of them, as we need $\sum_{k\in\mathbb{N}} \lambda_k \left(2 - \frac{\gamma\beta}{2} - \lambda_k\right) = +\infty$, unless either A_1 or A_2 is uniformly monotone). Finally, unlike [128, Theorem 2.1], we do not require the assumption $\inf_{k\in\mathbb{N}} \lambda_k > 0$.
(ii) In Theorem 5.3 (ii), even when $\sum_{k\in\mathbb{N}} \lambda_k \left(2 - \frac{\gamma\beta}{2} - \lambda_k\right) < +\infty$, we have proved that the sequence $(x^k)_{k\in\mathbb{N}}$ (respectively $(u^k)_{k\in\mathbb{N}}$) is strongly convergent to $\bar{x}$ when A_1 (respectively A_2) is uniformly monotone.
(iii) Very close in time to [26], a similar convergence result allowing γ in the interval $]0, 4/\beta[$ was proved in [126] using the notion of *conically averaged operators* recently introduced in [39]. Observe that the proof there can be refined to prove $z^k \rightharpoonup \bar{z} \in \Omega_\gamma$ not only for a fixed relaxation parameter $\lambda_k = \lambda$, as it was done in [126, Corollary 4.2]. Indeed, by [126, Theorem 4.1], the operator $\mathcal{T}_{DY}^{\gamma}$ in (5.2) is conically $(2-\gamma\beta/2)^{-1}$-averaged, so [39, Proposition 2.9] can be applied to deduce the convergence of the Krasnosel'skiĭ–Mann iteration (5.3) to a fixed point of $\mathcal{T}_{DY}^{\gamma}$, which belongs to Ω_γ by Lemma 5.1.

As a corollary, we obtain the following convergence result for the forward-backward splitting algorithm that allows doubling the range of the stepsizes assumed in [44, Theorem 26.14] (which is a particular case of [114, Proposition 4.4] and [116, Lemma 4.4]). Although this wider range of the stepsizes has been shown before in [140, 141, 159], it has not yet become widely known in the literature.

Corollary 5.5 (Forward-backward splitting algorithm). *Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be a maximally monotone operator and $T : \mathcal{H} \to \mathcal{H}$ be a $\frac{1}{\beta}$-cocoercive operator, with $\beta > 0$, such that $\operatorname{zer}(A + T) \neq \emptyset$. Set a stepsize $\gamma \in]0, 4/\beta[$ and consider a sequence of relaxation parameters $(\lambda_k)_{k\in\mathbb{N}}$ in $]0, 2 - \frac{\gamma\beta}{2}]$ such that $\sum_{k\in\mathbb{N}} \lambda_k \left(2 - \frac{\gamma\beta}{2} - \lambda_k\right) = +\infty$. Given some*

initial point $x^0 \in \mathcal{H}$*, consider the sequence* $(x^k)_{k\in\mathbb{N}}$ *defined by*

$$x^{k+1} = (1-\lambda_k)x^k + \lambda_k J_{\gamma A}\big(x^k - \gamma T(x^k)\big), \quad \forall k \in \mathbb{N}.$$

Then the following assertions hold.

(i) $(x^k)_{k\in\mathbb{N}}$ *converges weakly to a point* $\bar{x} \in \operatorname{zer}(A+T)$ *and* $\big(T(x^k)\big)_{k\in\mathbb{N}}$ *converges strongly to the unique dual solution* $T(\bar{x})$*.*

(ii) If either A *is uniformly monotone on every bounded subset of its domain, or* T *is demiregular at every point belonging to* $\operatorname{zer}(A+T)$*, then* $(x^k)_{k\in\mathbb{N}}$ *converges strongly to* $\bar{x} \in \operatorname{zer}(A+T)$*.*

Proof. Apply Theorem 5.3 with $A_1 = 0$ and $A_2 = A$. Theorem 5.3 (i) then yields that $T(x^k) \to T(\bar{x})$ and $T(\operatorname{zer}(A+T)) = \{T(\bar{x})\}$, which is the unique solution to the dual problem. □

Remark 5.6 (Generalized forward-backward algorithm). The original proof of the *generalized forward-backward method* [210] relies on the averagedness of a composition of operators, which leads to the parameter bounds in Theorem 3.10. Nonetheless, as described in Section 3.2.2.3, this method can be regarded as an application of the Davis–Yin splitting to the operators in (3.17) and (3.18). As the latter preserves the same cocoercive constant than T (see, e.g., [73, Proposition 6.2]), it follows from Theorem 5.3 that the stepsize and relaxation parameters in Theorem 3.10 can be chosen as $\gamma \in]0, 4/\beta[$ and $\lambda \in \left]0, 2 - \frac{\gamma\beta}{2}\right[$, respectively. This enlargement of the parameters range appears to have not been previously reported.

5.2.2 Strengthened Davis–Yin splitting algorithm

Recently in [20], the authors developed a systematic framework for computing the resolvent of sums of maximally monotone operators, which draws upon the notion of *strengthening* of an operator. The analysis there leads to new splitting schemes such as the *averaged alternating modified reflections* [16], which is an enhanced version of the Douglas–Rachford splitting suitable for addressing best approximation problems (see, e.g., [5, 14, 17]). We now employ the techniques from [20] to derive a modification of the Davis–Yin splitting for the calculation of the resolvent of $A_1 + A_2 + T$.

Definition 5.7. *Let $\theta > 0$, $\sigma \in \mathbb{R}$ and let $w \in \mathcal{H}$. Given $A\colon \mathcal{H} \rightrightarrows \mathcal{H}$, the (θ,σ)-*strengthening with w *of A is the operator ${}_wA^{(\theta,\sigma)}\colon \mathcal{H} \rightrightarrows \mathcal{H}$ defined by*

$$ {}_wA^{(\theta,\sigma)} := A \circ (\theta \operatorname{Id} - w) + \sigma \operatorname{Id}. $$

We now present two facts containing results from [20] and [125] that will be used to derive the strengthened version of the Davis–Yin algorithm. The first of them explores the monotonicity of strengthened operators and characterizes their resolvents. The second relates the resolvent of sums of operators with the zeros of the sum of their strengthenings.

Fact 5.8. *Let $\beta, \theta > 0$, $\alpha, \sigma \in \mathbb{R}$ and let $w \in \mathcal{H}$. Given $A : \mathcal{H} \rightrightarrows \mathcal{H}$, $T : \mathcal{H} \to \mathcal{H}$ and $\gamma > 0$, the following hold.*

(i) A is (maximally) α-monotone if and only if ${}_wA^{(\theta,\sigma)}$ is (maximally) $(\theta\alpha+\sigma)$-monotone.

(ii) T is $\frac{1}{\beta}$-cocoercive if and only if ${}_wT^{(\theta,0)}$ is $\frac{1}{\theta\beta}$-cocoercive. Consequently, if $\sigma > 0$ then ${}_wT^{(\theta,\sigma)}$ is $\frac{1}{\mu}$-cocoercive with $\mu := \theta\beta + \sigma$.

(iii) If $1 + \gamma\sigma \neq 0$, then

$$ J_{\gamma\, {}_wA^{(\theta,\sigma)}} = \frac{1}{\theta}\left(J_{\frac{\gamma\theta}{1+\gamma\sigma}A} \circ \left(\frac{\theta}{1+\gamma\sigma} \operatorname{Id} - w \right) + w \right). $$

If, in addition, A is maximally α-monotone and $1 + \gamma(\theta\alpha + \sigma) > 0$, then $J_{\gamma\, {}_wA^{(\theta,\sigma)}}$ and $J_{\frac{\gamma\theta}{1+\gamma\sigma}A}$ are single-valued and have full domain.

Proof. (i) and (iii): [125, Proposition 2.1]. (ii): [20, Theorem 1 (iii)]. □

Fact 5.9 ([20, Proposition 3]). *Let $A_i : \mathcal{H} \rightrightarrows \mathcal{H}$ and $\alpha_i, \sigma_i \in \mathbb{R}$, for $i \in \{1, \ldots, n\}$, and set $\sigma := \sum_{i=1}^n \sigma_i > 0$. Let $q \in \mathcal{H}$ and $\theta > 0$. Then*

$$ J_{\frac{\theta}{\sigma}(\sum_{i=1}^n A_i)}(q) := \left\{ \theta x + q \;:\; x \in \operatorname{zer}\left(\sum_{i=1}^n {}_{-q}A_i^{(\theta,\sigma_i)} \right) \right\}. $$

Moreover, if each A_i is α_i-monotone, $\sum_{i=1}^n(\theta\alpha_i + \sigma_i) > 0$ and $q \in \operatorname{ran}\left(\operatorname{Id} + \frac{\theta}{\sigma}\sum_{i=1}^n A_i\right)$, then $J_{\frac{\theta}{\sigma}(\sum_{i=1}^n A_i)}(q)$ is a singleton and

$$ \operatorname{zer}\left(\sum_{i=1}^n {}_{-q}A_i^{(\theta,\sigma_i)} \right) = \left\{ \frac{1}{\theta}\left(J_{\frac{\theta}{\sigma}(\sum_{i=1}^n A_i)}(q) - q \right) \right\}. $$

Theorem 5.10 (Strengthened-Davis–Yin splitting). *Let $A_1, A_2 : \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally α_{A_1}-monotone and α_{A_2}-monotone operators, respectively, with $\alpha_{A_1}, \alpha_{A_2} \in \mathbb{R}$. Let $T : \mathcal{H} \to \mathcal{H}$ be a $\frac{1}{\beta}$-cocoercive and maximally α_T-monotone operator, with $\beta > 0$ and $\alpha_T \in \mathbb{R}$. Let $\theta > 0$, $\sigma_{A_1}, \sigma_{A_2} \in \mathbb{R}$ and $\sigma_T \geq 0$ be such that*

$$\sigma_{A_1} + \sigma_{A_2} + \sigma_T > 0 \quad \text{and} \quad (\theta\alpha_{A_1} + \sigma_{A_1}, \theta\alpha_{A_2} + \sigma_{A_2}, \theta\alpha_T + \sigma_T) \in \mathbb{R}^3_+ \setminus \{0\}. \tag{5.18}$$

Let $\mu := \theta\beta + \sigma_T$ and take $\gamma \in \left]0, 4/\mu\right[$. Consider a sequence of relaxation parameters $(\lambda_k)_{k\in\mathbb{N}}$ in the interval $\left]0, 2 - \frac{\gamma\mu}{2}\right]$ verifying $\sum_{k\in\mathbb{N}} \lambda_k \left(2 - \frac{\gamma\mu}{2} - \lambda_k\right) = +\infty$. Suppose that $q \in \operatorname{ran}\left(\operatorname{Id} + \frac{\theta}{\sigma_{A_1}+\sigma_{A_2}+\sigma_T}(A_1 + A_2 + T)\right)$. Given any $z^0 \in \mathcal{H}$, consider the sequences

$$\begin{cases} x^k = J_{\frac{\gamma\theta}{1+\gamma\sigma_{A_1}}A_1}\left(\dfrac{1}{1+\gamma\sigma_{A_1}}(z^k + \gamma\sigma_{A_1}q)\right), \\ u^k = J_{\frac{\gamma\theta}{1+\gamma\sigma_{A_2}}A_2}\left(\dfrac{1}{1+\gamma\sigma_{A_2}}\big((2-\gamma\sigma_T)x^k - z^k - \theta\gamma T(x^k) + \gamma(\sigma_{A_2}+\sigma_T)q\big)\right), \\ z^{k+1} = z^k + \lambda_k(u^k - x^k), \end{cases} \tag{5.19}$$

for all $k \in \mathbb{N}$. Then $(x^k)_{k\in\mathbb{N}}$ and $(u^k)_{k\in\mathbb{N}}$ are weakly convergent to $J_{\frac{\theta}{\sigma_{A_1}+\sigma_{A_2}+\sigma_T}(A_1+A_2+T)}(q)$, and $(z^k)_{k\in\mathbb{N}}$ is weakly convergent to $\bar{z}$, with

$$J_{\frac{\gamma\theta}{1+\gamma\sigma_{A_1}}A_1}\left(\frac{1}{1+\gamma\sigma_{A_1}}(\bar{z} + \gamma\sigma_{A_1}q)\right) = J_{\frac{\theta}{\sigma_{A_1}+\sigma_{A_2}+\sigma_T}(A_1+A_2+T)}(q).$$

Further, if $\theta\alpha_{A_1} + \sigma_{A_1} > 0$ (respectively $\theta\alpha_{A_2} + \sigma_{A_2} > 0$) then the convergence of $(x^k)_{k\in\mathbb{N}}$ (respectively $(u^k)_{k\in\mathbb{N}}$) is strong, even when $\sum_{k\in\mathbb{N}} \lambda_k \left(2 - \frac{\gamma\mu}{2} - \lambda_k\right) < +\infty$.

Proof. Set $\hat{z}^0 := \frac{1}{\theta}(z^0 - q)$ and consider the sequences

$$\begin{cases} \hat{x}^k = J_{\gamma\,{}_{-q}A_1{}^{(\theta,\sigma_{A_1})}}(\hat{z}^k), \\ \hat{u}^k = J_{\gamma\,{}_{-q}A_2{}^{(\theta,\sigma_{A_2})}}\left(2\hat{x}^k - \hat{z}^k - \gamma\,{}_{-q}T^{(\theta,\sigma_T)}(\hat{x}^k)\right), \\ \hat{z}^{k+1} = \hat{z}^k + \lambda_k(\hat{u}^k - \hat{x}^k). \end{cases} \tag{5.20}$$

By (5.18) and Fact 5.8 (i)-(ii), the operators ${}_{-q}A_1{}^{(\theta,\sigma_{A_1})}$, ${}_{-q}A_2{}^{(\theta,\sigma_{A_2})}$ are maximally monotone, and ${}_{-q}T^{(\theta,\sigma_T)}$ is $\frac{1}{\mu}$-cocoercive. Since $q \in \operatorname{ran}\left(\operatorname{Id} + \frac{\theta}{\sigma_{A_1}+\sigma_{A_2}+\sigma_T}(A_1 + A_2 + T)\right)$ by assumption, (5.18) and Fact 5.9 imply that

$$\operatorname{zer}\left({}_{-q}A_1{}^{(\theta,\sigma_{A_1})} + {}_{-q}A_2{}^{(\theta,\sigma_{A_2})} + {}_{-q}T^{(\theta,\sigma_T)}\right) = \left\{\frac{1}{\theta}\left(J_{\frac{\theta}{\sigma_{A_1}+\sigma_{A_2}+\sigma_T}(A_1+A_2+T)}(q) - q\right)\right\}. \tag{5.21}$$

By Theorem 5.3 (i), $\hat{x}^k \rightharpoonup \hat{x}$ and $\hat{u}^k \rightharpoonup \hat{x}$, with

$$\hat{x} \in \operatorname{zer}\left({}_{-q}A_1^{(\theta,\sigma_{A_1})} + {}_{-q}A_2^{(\theta,\sigma_{A_2})} + {}_{-q}T^{(\theta,\sigma_T)}\right),$$

and $\hat{z}^k \rightharpoonup \hat{z}$, where $\hat{z}$ satisfies

$$\hat{x} = J_{\gamma\, {}_{-q}A_1^{(\theta,\sigma_{A_1})}}(\hat{z}) \in \operatorname{zer}\left({}_{-q}A_1^{(\theta,\sigma_{A_1})} + {}_{-q}A_2^{(\theta,\sigma_{A_2})} + {}_{-q}T^{(\theta,\sigma_T)}\right). \tag{5.22}$$

If $\theta\alpha_{A_1} + \sigma_{A_1} > 0$ (respectively $\theta\alpha_{A_2} + \sigma_{A_2} > 0$), then $\hat{x}^k \to \hat{x}$ (respectively $\hat{u}^k \to \hat{x}$) by Theorem 5.3 (ii), even if $\sum_{k\in\mathbb{N}} \lambda_k \left(2 - \frac{\gamma\mu}{2} - \lambda_k\right) < +\infty$. Thanks to Fact 5.8 (iii), we may rewrite (5.20) as

$$\begin{cases} \theta\hat{x}^k + q = J_{\frac{\gamma\theta}{1+\gamma\sigma_{A_1}}A_1}\left(\dfrac{\theta}{1+\gamma\sigma_{A_1}}\hat{z}^k + q\right), \\ \theta\hat{u}^k + q = J_{\frac{\gamma\theta}{1+\gamma\sigma_{A_2}}A_2}\left(\dfrac{\theta}{1+\gamma\sigma_{A_2}}\left(2\hat{x}^k - \hat{z}^k - \gamma\left(T(\theta\hat{x}^k + q) + \sigma_T\hat{x}^k\right)\right) + q\right). \end{cases}$$

Further, by (5.21), (5.22) and Fact 5.8 (iii),

$$J_{\frac{\theta}{\sigma_{A_1}+\sigma_{A_2}+\sigma_T}(A_1+A_2+T)}(q) = \theta J_{\gamma\, {}_{-q}A_1^{(\theta,\sigma_{A_1})}}(\hat{z}) + q = J_{\frac{\gamma\theta}{1+\gamma\sigma_{A_1}}A_1}\left(\frac{\theta}{1+\gamma\sigma_{A_1}}\hat{z} + q\right).$$

The result follows by making the change of variables $(z^k, x^k, u^k) := (\theta\hat{z}^k+q, \theta\hat{x}^k+q, \theta\hat{u}^k+q)$ for all $k \in \mathbb{N}$ and $\bar{z} := \theta\hat{z} + q$. The final assertion is a consequence of Remark 5.4 (ii). □

Remark 5.11. Another way of computing the resolvent with parameter $\delta > 0$ of the sum $A_1 + A_2 + T$ at $q \in \mathcal{H}$ is applying the Davis–Yin splitting algorithm to A_1, A_2 and $\widetilde{T} := \frac{1}{\delta}(\mathrm{Id} - q) + T$, where $\widetilde{T}$ is $\frac{\delta+\beta}{\delta\beta}$-cocoercive, by Fact 5.8 (ii), and $\frac{1}{\beta}$ is the cocoercivity constant of T. Note that this is a particular instance covered by Theorem 5.10, taking $\sigma_T = \frac{1}{\delta}$, $\sigma_{A_1} = \sigma_{A_2} = 0$ and $\theta = 1$.

5.3 Numerical experiments

In this section we provide some numerical examples of the algorithms developed in the chapter. These experiments aim not to be exhaustive and only intend to show the importance of appropriately choosing the stepsize and the relaxation parameters of the algorithms.

5.3.1 A feasibility problem with hard and soft constraints

Let $\mathbb{A}, \mathbb{B}, \mathbb{C} \subseteq \mathbb{R}^n$ be three closed and convex sets with nonempty intersection of the relative interiors of $\mathbb{A}$ and $\mathbb{B}$. Suppose $\mathbb{A}$ and $\mathbb{B}$ are *hard constraints*, which need to be satisfied, and $\mathbb{C}$ is a third *soft constraint*, which does not necessarily need to be fulfilled, but whose violation we want to reduce as much as possible. Imagine that, at the same time, we would like to find a point in $\mathbb{A} \cap \mathbb{B}$ as close as possible to a point $q \in \mathbb{R}^n$. This problem can be written as

$$\operatorname*{argmin}_{x \in \mathbb{A} \cap \mathbb{B}} \frac{1}{2} d^2(x, \mathbb{C}) + \frac{\rho}{2} \|x - q\|^2, \tag{5.23}$$

where $d^2(x, \mathbb{C}) := \|x - P_{\mathbb{C}}(x)\|^2$ and $\rho > 0$ is a regularization parameter specifying the importance of remaining close to the point q. Problem (5.23) can be reformulated as

$$\operatorname*{argmin}_{x \in \mathbb{R}^n} \iota_{\mathbb{A}}(x) + \iota_{\mathbb{B}}(x) + \frac{1}{2} \|x - q\|^2 + \frac{1}{2\rho} d^2(x, \mathbb{C}),$$

whose solution is given by $\operatorname{prox}_{\left(\iota_{\mathbb{A}} + \iota_{\mathbb{B}} + \frac{1}{2\rho} d^2(\cdot, \mathbb{C})\right)}(q)$. The subdifferential sum rule (see, e.g., [44, Corollary 16.50(v)]) guarantees the equality

$$\operatorname{prox}_{\left(\iota_{\mathbb{A}} + \iota_{\mathbb{B}} + \frac{1}{2\rho} d^2(\cdot, \mathbb{C})\right)}(q) = J_{\left(\partial \iota_{\mathbb{A}} + \partial \iota_{\mathbb{B}} + \nabla\left(\frac{1}{2\rho} d^2(\cdot, \mathbb{C})\right)\right)}(q) = J_{\left(N_{\mathbb{A}} + N_{\mathbb{B}} + \frac{1}{\rho}(\mathrm{Id} - P_{\mathbb{C}})\right)}(q),$$

and thus, solving (5.23) boils down to computing the resolvent at q of the sum of the three maximally monotone operators $A_1 := N_{\mathbb{A}}$, $A_2 := N_{\mathbb{B}}$ and $T := \frac{1}{\rho}(\mathrm{Id} - P_{\mathbb{C}})$, with T being $\frac{1}{\rho}$-cocoercive (see, e.g., [44, Corollary 12.31]).

To illustrate on the problem (5.23) the behavior of the Davis–Yin algorithm and its strengthened version derived in Theorem 5.10, we retake our simple introductory example of two balls $\mathbb{A}$ and $\mathbb{B}$ centered at $(-1.6, -0.75)$ and $(-0.35, 0.12)$, with radii 0.55 and 1, respectively. We chose these values to make the problem slightly challenging. We now add a new third ball $\mathbb{C}$ with center $(1, -1)$ and radius 0.5, the point $q := (-1.75, 1.5)$ and take $\rho := 1$. Observe that any combination of $\sigma_{A_1} \geq 0$, $\sigma_{A_2} \geq 0$ and $\sigma_T \geq 0$ such that $\theta := \sigma_{A_1} + \sigma_{A_2} + \sigma_T > 0$ satisfies the hypotheses of Theorem 5.10. Although finding the best values is beyond our scope, for comparison, we tested the result of running the algorithm (5.19) with $(\sigma_{A_1}, \sigma_{A_2}, \sigma_T) = (0, 0, 1)$ (which corresponds to the Davis–Yin splitting algorithm, see Remark 5.11) and $(\sigma_{A_1}, \sigma_{A_2}, \sigma_T) = (0, 1, 1)$, using as starting point $z^0 := (0.7, 1.7)$. In accordance with Theorem 5.10, the stepsize γ must be chosen so that $\gamma\mu \in \,]0, 4[$, for $\mu = ((\sigma_{A_1} + \sigma_{A_2} + \sigma_T)\rho + \sigma_T)$. In Figure 5.4 we have represented the

iterates for $\lambda_k = 0.99(2 - \frac{\gamma\mu}{2})$ and for two values of $\gamma\mu$, namely 1.5 (overrelaxation) and 2.5 (underrelaxation).

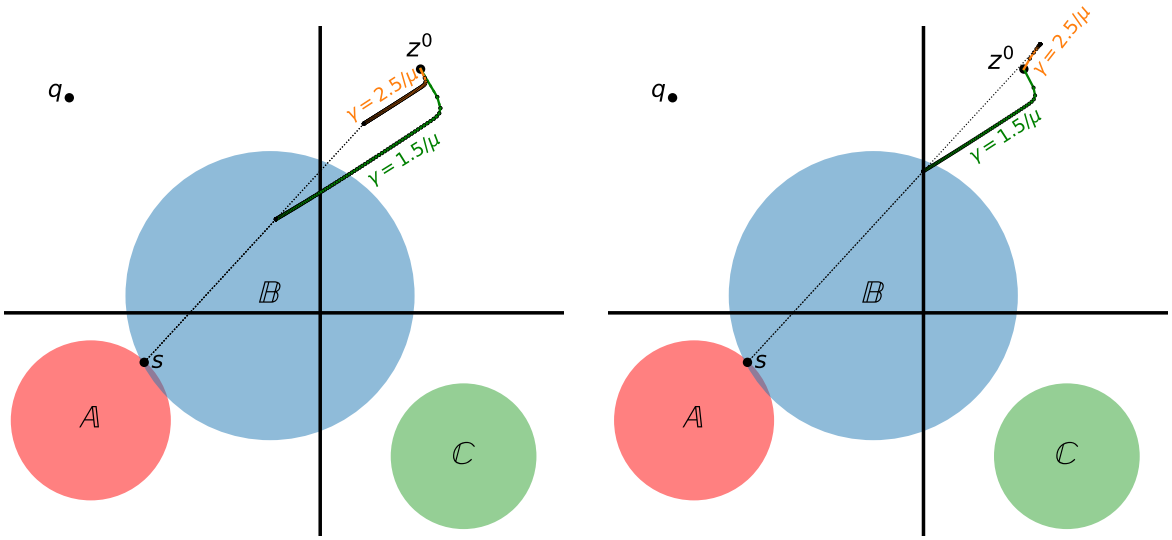


FIGURE 5.4: Behavior of the iterates of the Davis–Yin (left) and the strengthened-Davis–Yin (right) splitting algorithms for the problem (5.23) for two stepsize parameters γ and relaxation parameters $\lambda_k = 0.99(2 - \gamma\mu/2)$. Since $\sigma_{A_1} = 0$, the solution is obtained after projecting the fixed point onto the set $\mathbb{A}$.

In order to obtain the best combination of the stepsize and relaxation parameters, we run the algorithms for every possible value of $(\gamma\mu, \lambda)$ on a grid with 4950 points in $]0, 4[\times]0, 2[$. The algorithms were stopped when the norm of the difference between the shadow sequence $\big(P_{\mathbb{A}}(z^k)\big)_{k\in\mathbb{N}}$ and the solution to the problem was smaller than 10^{-8}. The solution, which is approximately equal to $(-1.227559, -0.3452923)$, was computed in Maple by numerically solving the KKT conditions with high precision. A contour plot representing the number of iterations is shown in Figure 5.5. The minimum number of iterations for Davis–Yin was 17 and it was attained at $(\gamma\mu, \lambda) = (3.11, 0.43)$, and for the strengthened-Davis–Yin was 16 and it was reached at three pair of values of $\gamma\mu$ and λ, namely $\gamma\mu = 2.34, \lambda \in \{0.79, 0.81\}$ and $\gamma\mu = 2.39, \lambda = 0.79$.

5.3.2 Image recovery via ℓ_1 regularization

The restoration of blurred images using ℓ_1 regularization has become a standard application in the literature to test the performance of forward-backward algorithms, see [55].

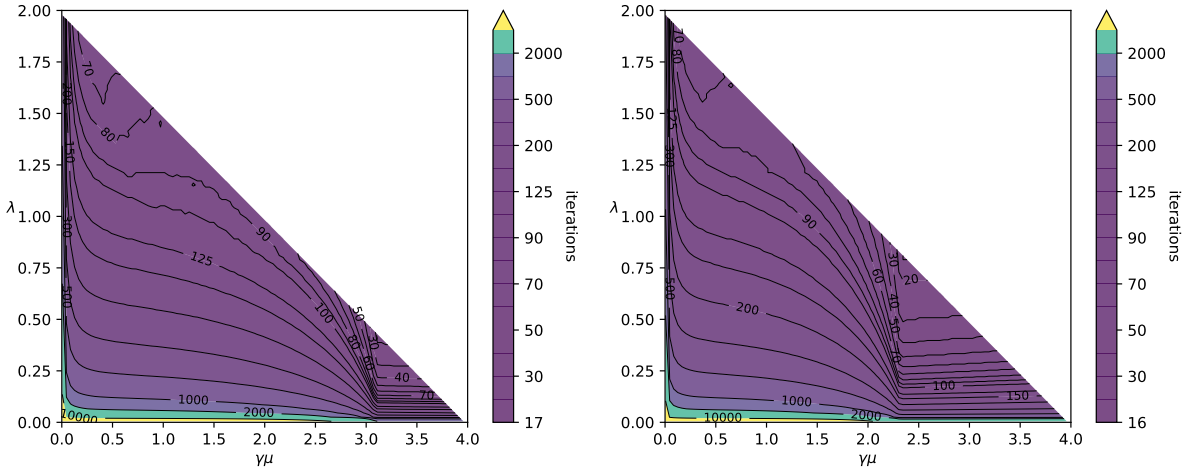


FIGURE 5.5: Number of iterations needed until the shadow sequence is sufficiently close to the solution s when the Davis–Yin (left) and the strengthened-Davis–Yin (right) splitting algorithms are applied for different values of γ and $\lambda_k = \lambda$, with the experiment setting shown in Figure 5.4.

This consists in solving a minimization problem of the form

$$\operatorname*{argmin}_{x \in \mathbb{R}^n} \ \mu \|x\|_1 + \frac{1}{2}\|Mx - b\|^2, \tag{5.24}$$

where $M \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ is the observed blurred image (the vectorization of the two-dimensional matrix) and $\mu > 0$ is a regularization parameter. Setting $A_2 = \partial(\mu\|\cdot\|_1)$ and $T = M^T(Mx - b)$, this problem can be reformulated as finding a zero of the sum $A_2 + T$ of two maximally monotone operators. Since T is Lipschitz continuous, we can employ the forward-backward algorithm (i.e., Davis–Yin with $A_1 = 0$), to solve (5.24). Recall that the proximity operator of the ℓ_1-norm is the well-known soft thresholding function from Proposition 2.31. As pixel values must be in $[0, 1]$, it is more realistic to solve instead the problem

$$\operatorname*{argmin}_{x \in [0,1]^n} \ \mu \|x\|_1 + \frac{1}{2}\|Mx - b\|^2,$$

By setting $A_1 = N_{[0,1]^n}$ and A_2 and T as above, this problem can be solved without much additional effort using the Davis–Yin splitting algorithm.

For our tests we replicated the wavelet-based restoration method in [55, Section 5.1], including the additional constraint $x \in [0, 1]^n$. We also ran our experiments without this constraint (applying thus forward-backward) and the results were basically the same, so we do not include them for brevity. We employed as observed images the widely-used 256×256 pixels cameraman image and a picture of a symbol from the University of Alicante: the sculpture "Dibuixar l'espai" (by Pepe Azorín), with a resolution of 600×800 pixels. The

images, shown in Figure 5.6, were subjected to a Gaussian 9×9 blur with standard deviation 4, followed by an additive zero-mean Gaussian noise with standard deviation 10^{-3}. We chose $M = RW$, where R is the matrix representing the blur operator and W is the inverse of the three stage Haar wavelet transform. The regularization parameter was taken as $\mu = 2 \cdot 10^{-5}$. The Lipschitz constant of T is the spectral radius of $M^T M$, which is equal to 1. Thus, T is 1-cocoercive and the stepsize in the Davis–Yin algorithm can be chosen in the interval $]0, 4[$. For values of (γ, λ) on a grid with 4950 points in $]0, 4[\times]0, 2[$, we performed 200 iterations of the algorithm taking as initial image the observed blurred image. Figure 5.7 shows the value of the objective function in the final iteration. We observe a symmetry with respect to the diagonal. The lowest values of the objective function were 0.349 for the cameraman and 2.684 for the sculpture, and they were both attained at $(\gamma, \lambda) = (1.98, 0.99)$.

FIGURE 5.6: Original (left), observed blurred (middle) and restored (right) images, showing the cameraman at the top and the sculpture "Dibuixar l'espai" at the bottom. The Davis–Yin algorithm was applied for 200 iterations with $\gamma = 1.98$ and $\lambda = 0.99$, using as starting point the observed blurred image.

Interpretation of the results of the experiments The numerical experiments show the importance of appropriately selecting the stepsize and relaxation parameters. In most of our tests, the behavior of the algorithm with respect to the parameters was symmetric,

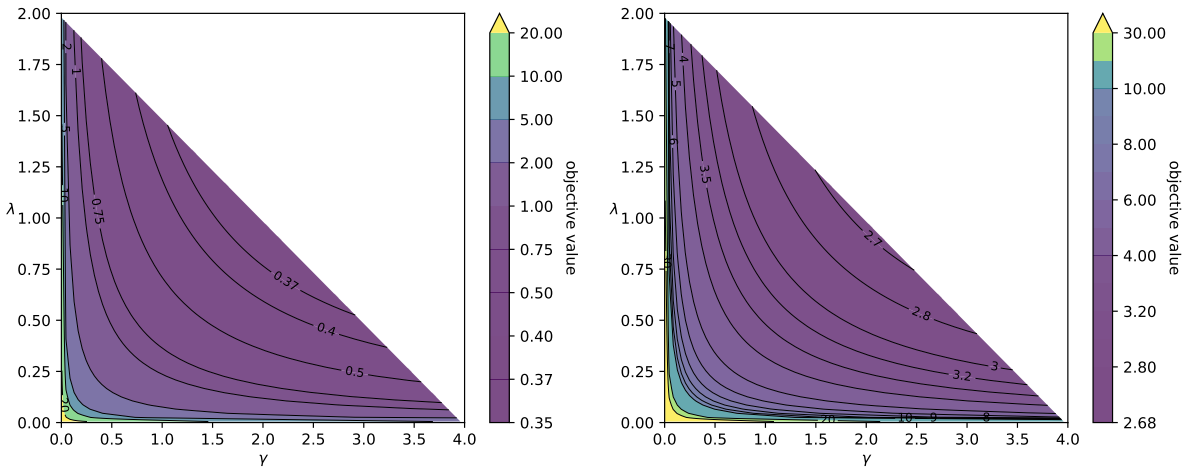


FIGURE 5.7: Objective function value after 200 iterations of the Davis–Yin splitting algorithm applied to the cameraman (left) and the sculpture "Dibuixar l'espai" (right), for different values of γ and λ, and taking as starting point the observed blurred image.

as the one shown in Figure 5.7. Selecting the best parameters is not a simple task, but even so, it is clear that having more freedom in the choice of the stepsize parameter can only be advantageous.

Chapter 6

Forward-backward methods with reduced lifting

In this chapter, we study forward-backward type schemes with reduced lifting for solving the monotone inclusion in Problem 3.9. As previously demonstrated, the resolution of this problem requires the employment of different algorithms depending on the properties satisfied by the single-valued operators. In the same fashion, our analysis will distinguish between the single-valued operators being cocoercive or monotone and Lipschitz continuous.

We summarize the main topics of this chapter as follows:

- In Section 6.1, we concentrate on the case in which the single-valued operators are cocoercive and analyze the convergence of a frugal forward-backward splitting with minimal lifting (see Algorithm 6.6) in Theorem 6.7.
- We propose Algorithm 6.14 as a "reflected-like modification" of Algorithm 6.6 for solving Problem 3.9 when the single-valued operators are monotone and Lipschitz continuous. Algorithm 6.14 is the first algorithm for this problem with $(n-1)$-fold lifting. However, Fact 4.13 cannot be used to guarantee that this lifting is minimal, as the method performs two evaluations of the single-valued operators per iteration.
- One of the advantages of the proposed algorithms is that they do not rely on product space reformulations, which makes them conducive for distributed decentralized optimization. We illustrate this in Section 6.1.3.
- Section 6.3 contains a numerical experiment comparing the performance of our proposed minimal lifting forward-backward method with the generalized forward-backward algorithm (see Section 3.2.2.3).

The results presented here first appeared in [25].

6.1 A forward-backward method with minimal lifting

Let $n \geq 2$. In this section, we consider the following instance of Problem 3.9:

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\right)(x), \tag{6.1}$$

with $A_1, \ldots, A_n \colon \mathcal{H} \rightrightarrows \mathcal{H}$ being maximally monotone operators and $T_1, \ldots, T_{n-1} \colon \mathcal{H} \to \mathcal{H}$ being $\frac{1}{\beta}$-cocoercive, with $\beta > 0$.

6.1.1 A fixed point encoding with $(n-1)$-fold lifting

Let $\lambda, \gamma > 0$. In order to tackle (6.1), we propose a fixed point algorithm consisting in the fixed point iteration generated by the operator $\mathcal{T}_{FB} : \mathcal{H}^{n-1} \to \mathcal{H}^{n-1}$ defined as

$$\mathcal{T}_{FB}(\mathbf{z}) := \mathbf{z} + \lambda \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix}, \tag{6.2}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{H}^n$ depends on $\mathbf{z} = (z_1, \ldots, z_{n-1}) \in \mathcal{H}^{n-1}$ and is given by

$$\begin{cases} x_1 = J_{\gamma A_1}(z_1), \\ x_i = J_{\gamma A_i}\big(z_i + x_{i-1} - z_{i-1} - \gamma T_{i-1}(x_{i-1})\big), \quad \forall i \in [\![2, n-1]\!], \\ x_n = J_{\gamma A_n}\big(x_1 + x_{n-1} - z_{n-1} - \gamma T_{n-1}(x_{n-1})\big). \end{cases} \tag{6.3}$$

Remark 6.1. Note that, although in view of Proposition 2.4 the sum of cocoercive operators is cocoercive, considering the sum of $n-1$ operators in Problem 3.9 gives the freedom of either applying each operator as a forward step before the corresponding backward step, or to apply the sum of all of them before a particular backward step (by setting all the operators to be equal to zero except for one of them, which would be equal to the sum). One of the advantages of the first alternative is that it enables a distributed decentralized implementation of the method, which is presented in Section 6.1.3.

Remark 6.2 (Special cases). If $n = 2$, then $x_1 = x_{n-1}$ and the fixed point iteration generated by $\mathcal{T}_{FB}$ recovers the *Davis–Yin splitting* for finding a zero of $A_1 + A_2 + T_1$. In turn, this includes the *forward-backward splitting* and *Douglas–Rachford splitting* as special cases by further taking $A_1 = 0$ or $T_1 = 0$, respectively.

If $T_1 = \cdots = T_{n-1} = 0$, then $\mathcal{T}_{FB}$ in (6.2) reduces to the *Malitsky–Tam resolvent splitting* proposed in [173] and further studied in [54, 72, 231].

REMARK 6.3 (On the number of single-valued operators in (6.1)). Although the number of set-valued and single-valued monotone operators in (6.1) differ by one, it is straightforward to derive a scheme where this is not the case by letting $A_1 = 0$. In this case, setting $x_1 = J_{\gamma A_1}(z_1) = z_1$ can be used to eliminate x_1 so that (6.2) and (6.3) respectively become

$$\mathcal{T}_{FB}(\mathbf{z}) := \mathbf{z} + \lambda \begin{pmatrix} x_2 - z_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix},$$

where

$$\begin{cases} x_2 = J_{\gamma A_2}\big(z_2 - \gamma T_1(z_1)\big), \\ x_i = J_{\gamma A_i}\big(z_i + x_{i-1} - z_{i-1} - \gamma T_{i-1}(x_{i-1})\big), \quad \forall i \in [\![3, n-1]\!], \\ x_n = J_{\gamma A_n}\big(z_1 + x_{n-1} - z_{n-1} - \gamma T_{n-1}(x_{n-1})\big). \end{cases}$$

Nevertheless, the resulting operator $\mathcal{T}_{FB}$ would now be defined in a space with dimension equal to the number of (nonzero) set-valued operators in (6.1). We discuss in Remark 6.10 how this may affect determining whether the resulting scheme has minimal lifting.

Finally, observe that while at first it may seem unusual that the number of set-valued and single-valued monotone operators in (6.1) are not the same, we note that this same situation arises in Davis–Yin splitting, as described above.

The following lemma implies that the fixed point operator $\mathcal{T}_{FB}$ (6.2) together with $J_{\gamma A_1}$ as a solution mapping, constitute a fixed point encoding for the monotone inclusion (6.1) which has $(n-1)$-fold lifting.

Lemma 6.4. *Let $n \geq 2$ and $\lambda, \gamma > 0$. The following assertions hold.*

(i) If $\bar{x} \in \operatorname{zer}\big(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\big)$, then there exists $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{FB}$.

(ii) If $(\bar{z}_1, \ldots, \bar{z}_{n-1}) \in \operatorname{Fix} \mathcal{T}_{FB}$, then $\bar{x} := J_{\gamma A_1}(\bar{z}_1) \in \operatorname{zer}\big(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\big)$. Moreover,

$$\bar{x} = J_{\gamma A_i}\big(\bar{z}_i - \bar{z}_{i-1} + \bar{x} - \gamma T_{i-1}(\bar{x})\big) = J_{\gamma A_n}\big(2\bar{x} - \bar{z}_{n-1} - \gamma T_{n-1}(\bar{x})\big), \tag{6.4}$$

for all $i \in [\![2, n-1]\!]$.

Consequently,

$$\operatorname{Fix} \mathcal{T}_{FB} \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\right) \neq \emptyset.$$

Proof. (i): Let $\bar{x} \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\right)$. Then there exists $(a_1, \ldots, a_n) \in \mathcal{H}^n$ such that $a_i \in A_i(\bar{x})$ and $\sum_{i=1}^{n} a_i + \sum_{i=1}^{n-1} T_i(\bar{x}) = 0$. Define the vector $\bar{\mathbf{z}} = (\bar{z}_1, \ldots, \bar{z}_{n-1}) \in \mathcal{H}^{n-1}$ according to

$$\begin{cases} \bar{z}_1 := \bar{x} + \gamma a_1 \in (\mathrm{Id} + \gamma A_1)(\bar{x}), \\ \bar{z}_i := \gamma a_i + \bar{z}_{i-1} + \gamma T_{i-1}(\bar{x}) \in (\mathrm{Id} + \gamma A_i)(\bar{x}) - \bar{x} + \bar{z}_{i-1} + \gamma T_{i-1}(\bar{x}), \end{cases}$$

for every $i \in [\![2, n-1]\!]$. Then $\bar{x} = J_{\gamma A_1}(z_1)$ and $\bar{x} = J_{\gamma A_i}(\bar{z}_i + \bar{x} - \bar{z}_{i-1} - \gamma T_{i-1}(\bar{x}))$ for $i \in [\![2, n-1]\!]$. Furthermore, we have

$$\begin{aligned} 2\bar{x} - \bar{z}_{n-1} - \gamma T_{n-1}(\bar{x}) &= \bar{x} + \gamma a_n + \bar{x} - \bar{z}_{n-1} + \gamma \sum_{i=2}^{n-1} (a_i + T_{i-1}(\bar{x})) + \gamma a_1 \\ &= \bar{x} + \gamma a_n + \bar{x} - \bar{z}_{n-1} + \sum_{i=2}^{n-1} (\bar{z}_i - \bar{z}_{i-1}) + \bar{z}_1 - \bar{x} \\ &\in (\mathrm{Id} + \gamma A_n)(\bar{x}), \end{aligned}$$

which implies that $\bar{x} = J_{\gamma A_n}(2\bar{x} - \bar{z}_{n-1} - \gamma T_{n-1}(\bar{x}))$. Altogether, it follows that $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{FB}$.

(ii): Let $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{FB}$ and set $\bar{x} := J_{\gamma A_1}(\bar{z}_1)$. Then (6.4) holds thanks to the expression of $\mathcal{T}_{FB}$. The definition of the resolvent therefore implies

$$\begin{cases} \bar{z}_1 - \bar{x} \in \gamma A_1(\bar{x}), \\ \bar{z}_i - \bar{z}_{i-1} - \gamma T_{i-1}(\bar{x}) \in \gamma A_i(\bar{x}), \quad \forall i \in [\![2, n-1]\!], \\ \bar{x} - z_{n-1} - \gamma T_{n-1}(\bar{x}) \in \gamma A_n(\bar{x}). \end{cases}$$

Summing together the above inclusions finally gives $\bar{x} \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\right)$, as claimed. □

6.1.2 Convergence analysis

The convergence analysis of our forward-backward algorithm with minimal lifting is based on an adaptation of the methodology developed in Theorem 5.3 for the Davis–Yin splitting to the framework presented in [173]. We start by studying the nonexpansive properties of the operator $\mathcal{T}_{FB}$ in (6.2).

Lemma 6.5. *For all* $\mathbf{z} = (z_1, \dots, z_n) \in \mathcal{H}^{n-1}$ *and* $\bar{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_n) \in \mathcal{H}^{n-1}$*, we have*

$$\begin{aligned}\|\mathcal{T}_{FB}(\mathbf{z}) - \mathcal{T}_{FB}(\bar{\mathbf{z}})\|^2 + \left(\frac{1-\lambda}{\lambda} - \frac{\gamma\beta}{2\lambda}\right)\|(\mathrm{Id} - \mathcal{T}_{FB})(\mathbf{z}) - (\mathrm{Id} - \mathcal{T}_{FB})(\bar{\mathbf{z}})\|^2 \\ + \frac{1}{\lambda}\Big\|\sum_{i=1}^{n-1}(\mathrm{Id} - \mathcal{T}_{FB})(\mathbf{z})_i - \sum_{i=1}^{n-1}(\mathrm{Id} - \mathcal{T}_{FB})(\bar{\mathbf{z}})_i\Big\|^2 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2.\end{aligned} \tag{6.5}$$

In particular, if $\gamma \in \left]0, \frac{2}{\beta}\right[$ *and* $\lambda \in \left]0, 1 - \frac{\gamma\beta}{2}\right[$*, then* $\mathcal{T}_{FB}$ *is* α*-averaged nonexpansive for* $\alpha = \frac{2\lambda}{2-\gamma\beta} \in]0, 1[$.

Proof. This proof mainly uses the monotonicity property of the operators $A_1, \dots, A_n$ together with the cocoercivity property of the operators $T_1, \dots, T_{n-1}$ to obtain some bounds which yield (6.5), from where the averagedness of operator $\mathcal{T}_{FB}$ can be directly deduced. For convenience, set $\mathbf{z}^+ := \mathcal{T}_{FB}(\mathbf{z})$ and $\bar{\mathbf{z}}^+ := \mathcal{T}_{FB}(\bar{\mathbf{z}})$. Further, let $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$ be given by (6.3) from $\mathbf{z}$, and let $\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_n) \in \mathcal{H}^n$ be given analogously, this time from $\bar{\mathbf{z}}$. Since $z_1 - x_1 \in \gamma A_1(x_1)$ and $\bar{z}_1 - \bar{x}_1 \in \gamma A_1(\bar{x}_1)$, monotonicity of γA_1 implies

$$\begin{aligned}0 &\leq \langle x_1 - \bar{x}_1, (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1)\rangle \\ &= \langle x_2 - \bar{x}_1, (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1)\rangle + \langle x_1 - x_2, (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1)\rangle.\end{aligned} \tag{6.6}$$

For all $i \in [\![2, n-1]\!]$, we have that $z_i - z_{i-1} + x_{i-1} - x_i - \gamma T_{i-1}(x_{i-1}) \in \gamma A_i(x_i)$ and $\bar{z}_i - \bar{z}_{i-1} + \bar{x}_{i-1} - \bar{x}_i - \gamma T_{i-1}(\bar{x}_{i-1}) \in \gamma A_i(\bar{x}_i)$. Thus, monotonicity of γA_i yields

$$\begin{aligned}0 &\leq \langle x_i - \bar{x}_i, z_i - z_{i-1} + x_{i-1} - x_i - \gamma T_{i-1}(x_{i-1})\rangle \\ &\quad - \langle x_i - \bar{x}_i, \bar{z}_i - \bar{z}_{i-1} + \bar{x}_{i-1} - \bar{x}_i - \gamma T_{i-1}(\bar{x}_{i-1})\rangle \\ &= \langle x_i - \bar{x}_i, (z_i - z_{i-1} + x_{i-1} - x_i) - (\bar{z}_i - \bar{z}_{i-1} + \bar{x}_{i-1} - \bar{x}_i)\rangle \\ &\quad - \gamma\langle x_i - \bar{x}_i, T_{i-1}(x_{i-1}) - T_{i-1}(\bar{x}_{i-1})\rangle \\ &= \langle x_{i+1} - \bar{x}_i, (z_i - x_i) - (\bar{z}_i - \bar{x}_i)\rangle + \langle x_i - x_{i+1}, (z_i - x_i) - (\bar{z}_i - \bar{x}_i)\rangle \\ &\quad - \langle x_i - \bar{x}_{i-1}, (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x}_{i-1})\rangle \\ &\quad - \langle \bar{x}_{i-1} - \bar{x}_i, (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x}_{i-1})\rangle \\ &\quad - \gamma\langle x_i - \bar{x}_i, T_{i-1}(x_{i-1}) - T_{i-1}(\bar{x}_{i-1})\rangle.\end{aligned}$$

Summing this inequality for $i \in [\![2, n-1]\!]$ and simplifying gives

$$\begin{aligned} 0 \leq & \sum_{i=2}^{n-1} \langle x_i - x_{i+1}, (z_i - x_i) - (\bar{z}_i - \bar{x}_i) \rangle - \sum_{i=1}^{n-2} \langle \bar{x}_i - \bar{x}_{i+1}, (z_i - x_i) - (\bar{z}_i - \bar{x}_i) \rangle \\ & - \langle x_2 - \bar{x}_1, (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1) \rangle + \langle x_n - \bar{x}_{n-1}, (z_{n-1} - x_{n-1}) - (\bar{z}_{n-1} - \bar{x}_{n-1}) \rangle \\ & - \gamma \sum_{i=2}^{n-1} \langle x_i - \bar{x}_i, T_{i-1}(x_{i-1}) - T_{i-1}(\bar{x}_{i-1}) \rangle. \end{aligned} \tag{6.7}$$

Since we have that $x_1 + x_{n-1} - x_n - z_{n-1} - \gamma T_{n-1}(x_{n-1}) \in \gamma A_n(x_n)$ and it holds that $\bar{x}_1 + \bar{x}_{n-1} - \bar{x}_n - \bar{z}_{n-1} - \gamma T_{n-1}(\bar{x}_{n-1}) \in \gamma A_n(\bar{x}_n)$, monotonicity of γA_n gives

$$\begin{aligned} 0 \leq & \langle x_n - \bar{x}_n, x_1 + x_{n-1} - x_n - z_{n-1} - \gamma T_{n-1}(x_{n-1}) \rangle \\ & - \langle x_n - \bar{x}_n, \bar{x}_1 + \bar{x}_{n-1} - \bar{x}_n - \bar{z}_{n-1} - \gamma T_{n-1}(\bar{x}_{n-1}) \rangle \\ = & \langle x_n - \bar{x}_n, (x_1 - x_n) - (\bar{x}_1 - \bar{x}_n) \rangle \\ & + \langle x_n - \bar{x}_n, (x_{n-1} - z_{n-1}) - (\bar{x}_{n-1} - \bar{z}_{n-1}) \rangle \\ & - \gamma \langle x_n - \bar{x}_n, T_{n-1}(x_{n-1}) - T_{n-1}(\bar{x}_{n-1}) \rangle \\ = & - \langle x_n - \bar{x}_{n-1}, (z_{n-1} - x_{n-1}) - (\bar{z}_{n-1} - \bar{x}_{n-1}) \rangle \\ & + \langle \bar{x}_n - \bar{x}_{n-1}, (z_{n-1} - x_{n-1}) - (\bar{z}_{n-1} - \bar{x}_{n-1}) \rangle \\ & + \frac{1}{2} \left(\|x_1 - \bar{x}_1\|^2 - \|x_n - \bar{x}_n\|^2 - \|(x_1 - x_n) - (\bar{x}_1 - \bar{x}_n)\|^2 \right) \\ & - \gamma \langle x_n - \bar{x}_n, T_{n-1}(x_{n-1}) - T_{n-1}(\bar{x}_{n-1}) \rangle. \end{aligned} \tag{6.8}$$

Adding (6.6), (6.7) and (6.8) and rearranging yields

$$\begin{aligned} 0 \leq & \sum_{i=1}^{n-1} \langle (x_i - \bar{x}_i) - (x_{i+1} - \bar{x}_{i+1}), \bar{x}_i - x_i \rangle \\ & + \sum_{i=1}^{n-1} \langle (x_i - \bar{x}_i) - (x_{i+1} - \bar{x}_{i+1}), z_i - \bar{z}_i \rangle \\ & + \frac{1}{2} \left(\|x_1 - \bar{x}_1\|^2 - \|x_n - \bar{x}_n\|^2 - \|(x_1 - x_n) - (\bar{x}_1 - \bar{x}_n)\|^2 \right) \\ & - \gamma \sum_{i=1}^{n-1} \langle x_{i+1} - \bar{x}_{i+1}, T_i(x_i) - T_i(\bar{x}_i) \rangle. \end{aligned} \tag{6.9}$$

The first term in (6.9) can be expressed as

$$\begin{aligned}
&\sum_{i=1}^{n-1}\langle (x_i-\bar{x}_i)-(x_{i+1}-\bar{x}_{i+1}),\bar{x}_i-x_i\rangle \\
&\quad=\frac{1}{2}\sum_{i=1}^{n-1}\left(\|x_{i+1}-\bar{x}_{i+1}\|^2-\|x_i-\bar{x}_i\|^2-\|(x_i-x_{i+1})-(\bar{x}_i-\bar{x}_{i+1})\|^2\right) \\
&\quad=\frac{1}{2}\left(\|x_n-\bar{x}_n\|^2-\|x_1-\bar{x}_1\|^2-\frac{1}{\lambda^2}\|(\mathbf{z}-\mathbf{z}^+)-(\bar{\mathbf{z}}-\bar{\mathbf{z}}^+)\|^2\right),
\end{aligned} \tag{6.10}$$

and the second term in (6.9) can be written as

$$\begin{aligned}
&\sum_{i=1}^{n-1}\langle (x_i-x_{i+1})-(\bar{x}_i-\bar{x}_{i+1}),z_i-\bar{z}_i\rangle \\
&\quad=\frac{1}{\lambda}\sum_{i=1}^{n-1}\langle (z_i-z_i^+)-(\bar{z}_i-\bar{z}_i^+),z_i-\bar{z}_i\rangle \\
&\quad=\frac{1}{\lambda}\langle (\mathbf{z}-\mathbf{z}^+)-(\bar{\mathbf{z}}-\bar{\mathbf{z}}^+),\mathbf{z}-\bar{\mathbf{z}}\rangle \\
&\quad=\frac{1}{2\lambda}\left(\|(\mathbf{z}-\mathbf{z}^+)-(\bar{\mathbf{z}}-\bar{\mathbf{z}}^+)\|^2+\|\mathbf{z}-\bar{\mathbf{z}}\|^2-\|\mathbf{z}^+-\bar{\mathbf{z}}^+\|^2\right).
\end{aligned} \tag{6.11}$$

To estimate the last term, Young's inequality and $\frac{1}{\beta}$-cocoercivity of $T_1,\dots,T_{n-1}$ gives

$$\begin{aligned}
&-\sum_{i=1}^{n-1}\langle x_{i+1}-\bar{x}_{i+1},T_i(x_i)-T_i(\bar{x}_i)\rangle \\
&\quad=\sum_{i=1}^{n-1}\langle (\bar{x}_{i+1}-\bar{x}_i)-(x_{i+1}-x_i),T_i(x_i)-T_i(\bar{x}_i)\rangle \\
&\qquad+\sum_{i=1}^{n-1}\langle \bar{x}_i-x_i,T_i(x_i)-T_i(\bar{x}_i)\rangle \\
&\quad\le\frac{\beta}{4}\sum_{i=1}^{n-1}\|(\bar{x}_{i+1}-\bar{x}_i)-(x_{i+1}-x_i)\|^2+\frac{1}{\beta}\sum_{i=1}^{n-1}\|T_i(x_i)-T_i(\bar{x}_i)\|^2 \\
&\qquad-\frac{1}{\beta}\sum_{i=1}^{n-1}\|T_i(x_i)-T_i(\bar{x}_i)\|^2 \\
&\quad=\frac{\beta}{4}\sum_{i=1}^{n-1}\|(\bar{x}_{i+1}-\bar{x}_i)-(x_{i+1}-x_i)\|^2 \\
&\quad=\frac{\beta}{4\lambda^2}\|(\mathbf{z}-\mathbf{z}^+)-(\bar{\mathbf{z}}-\bar{\mathbf{z}}^+)\|^2.
\end{aligned} \tag{6.12}$$

Thus, substituting (6.10) and (6.11) into (6.9), using (6.12) and simplifying gives the claimed inequality (6.5). Finally, to show that (6.5) implies $\mathcal{T}_{FB}$ is α-averaged nonexpansive with $\alpha := \frac{2\lambda}{2-\gamma\beta}$, note that $\alpha \in]0,1[$ and satisfies $\frac{1-\alpha}{\alpha} = \frac{1-\lambda}{\lambda} - \frac{\gamma\beta}{2\lambda}$. This completes the proof. □

We present the scheme obtained from the fixed point iteration of operator $\mathcal{T}_{FB}$ in Algorithm 6.6.

Algorithm 6.6 Forward-backward splitting for (6.1) with minimal lifting.

Require: $\gamma \in]0, 2/\beta[$ and $\lambda \in \left]0, 1 - \frac{\gamma\beta}{2}\right[$.

1: Choose $\mathbf{z}^0 = (z_1^0, \ldots, z_{n-1}^0) \in \mathcal{H}^{n-1}$.

2: **for** $k = 0, 1, \ldots$ **do**

3: Compute

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \lambda \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \end{pmatrix}, \tag{6.13}$$

with $\mathbf{x}^k = (x_1^k, \ldots, x_n^k) \in \mathcal{H}^n$ computed as

$$\begin{cases} x_1^k = J_{\gamma A_1}(z_1^k), \\ x_i^k = J_{\gamma A_i}\big(z_i^k + x_{i-1}^k - z_{i-1}^k - \gamma T_{i-1}(x_{i-1}^k)\big), \quad \forall i \in [\![2, n-1]\!], \\ x_n^k = J_{\gamma A_n}\big(x_1^k + x_{n-1}^k - z_{n-1}^k - \gamma T_{n-1}(x_{n-1}^k)\big). \end{cases}$$

4: **end for**

The global convergence of Algorithm 6.6 is demonstrated in Theorem 6.7.

Theorem 6.7. *Let $n \geq 2$, let $A_1, \ldots, A_n \colon \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone and let $T_1, \ldots, T_{n-1} \colon \mathcal{H} \to \mathcal{H}$ be $\frac{1}{\beta}$-cocoercive with $\operatorname{zer}\left(\sum_{i=1}^n A_i + \sum_{i=1}^{n-1} T_i\right) \neq \emptyset$. Further, let $\gamma \in]0, 2/\beta[$ and $\lambda \in \left]0, 1 - \frac{\gamma\beta}{2}\right[$. Given $\mathbf{z}^0 \in \mathcal{H}^{n-1}$, let $(\mathbf{z}^k)_{k\in\mathbb{N}}$ and $(\mathbf{x}^k)_{k\in\mathbb{N}}$ be the sequences generated by Algorithm 6.6. Then the following assertions hold.*

(i) The sequence $(\mathbf{z}^k)_{k\in\mathbb{N}}$ converges weakly to a point $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{FB}$.

(ii) The sequence $(\mathbf{x}^k)_{k\in\mathbb{N}}$ converges weakly to a point $(\bar{x}, \ldots, \bar{x}) \in \mathcal{H}^n$ with $\bar{x}$ solving the monotone inclusion (6.1).

(iii) The sequence $\left(T_i(x_i^k)\right)_{k\in\mathbb{N}}$ converges strongly to $T_i(\bar{x})$ for all $i \in [\![1, n-1]\!]$.

Proof. (i): Since $\operatorname{zer}\left(\sum_{i=1}^n A_i + \sum_{i=1}^{n-1} T_i\right) \neq \emptyset$, Lemma 6.4 (i) implies $\operatorname{Fix} \mathcal{T}_{FB} \neq \emptyset$. Since $\gamma \in]0, 2/\beta[$ and $\lambda \in \left]0, 1 - \frac{\gamma\beta}{2}\right[$, Lemma 6.5 implies $\mathcal{T}_{FB}$ is averaged nonexpansive. By

applying Theorem 2.7, we deduce that $(\mathbf{z}^k)_{k\in\mathbb{N}}$ converges weakly to a point $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{FB}$ and that $\lim_{k\to\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$.

(ii): By nonexpansiveness of resolvents, β-Lipschitz continuity of $T_1, \ldots, T_{n-1}$, and boundedness of $(\mathbf{z}^k)_{k\in\mathbb{N}}$, it follows that $(\mathbf{x}^k)_{k\in\mathbb{N}}$ is also bounded. Further, (6.2) and the fact that $\lim_{k\to\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$ implies that

$$\lim_{k\to\infty} \|x_i^k - x_{i-1}^k\| = 0, \quad \forall i \in [\![2, n]\!]. \tag{6.14}$$

Next, using the definition of the resolvent together with (6.13), we have

$$C \begin{pmatrix} z_1^k - x_1^k \\ (z_2^k - x_2^k) - (z_1^k - x_1^k) + \gamma t_2^k \\ \vdots \\ (z_{n-1}^k - x_{n-1}^k) - (z_{n-2}^k - x_{n-2}^k) + \gamma t_{n-1}^k \\ x_n^k \end{pmatrix} \ni \begin{pmatrix} x_1^k - x_n^k \\ x_2^k - x_n^k \\ \vdots \\ x_{n-1}^k - x_n^k \\ x_1^k - x_n^k + \gamma \sum_{i=1}^{n-1} t_{i+1}^k \end{pmatrix}, \tag{6.15}$$

where $t_i^k := T_{i-1}(x_i^k) - T_{i-1}(x_{i-1}^k)$ and the operator $C\colon \mathcal{H}^n \rightrightarrows \mathcal{H}^n$ is given by

$$C := \begin{pmatrix} (\gamma A_1)^{-1} \\ \big(\gamma(A_2 + T_1)\big)^{-1} \\ \vdots \\ \big(\gamma(A_{n-1} + T_{n-2})\big)^{-1} \\ \gamma(A_n + T_{n-1}) \end{pmatrix} + \begin{pmatrix} 0 & 0 & \ldots & 0 & -\operatorname{Id} \\ 0 & 0 & \ldots & 0 & -\operatorname{Id} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & -\operatorname{Id} \\ \operatorname{Id} & \operatorname{Id} & \ldots & \operatorname{Id} & 0 \end{pmatrix}. \tag{6.16}$$

As the sum of two maximally monotone operators is again maximally monotone provided that one of the operators has full domain (see Proposition 2.15), it follows that C is maximally monotone. Consequently, it is demiclosed by Proposition 2.23. That is, its graph is sequentially closed in the weak-strong topology.

Let $\bar{\mathbf{x}} \in \mathcal{H}^n$ be an arbitrary weak cluster point of the sequence $(\mathbf{x}^k)_{k\in\mathbb{N}}$. As a consequence of (6.14), $\bar{\mathbf{x}} = (\bar{x}, \ldots, \bar{x})$ for some $\bar{x} \in \mathcal{H}$. Taking the limit along a subsequence of $(\mathbf{x}^k)_{k\in\mathbb{N}}$ which converges weakly to $\bar{\mathbf{x}}$ in (6.15), using demiclosedness of C together with β-Lipschitz continuity of $T_1, \ldots, T_{n-1}$, and unravelling the resulting expression gives

$$\begin{cases} \bar{z}_1 - \bar{x} \in \gamma A_1(\bar{x}), \\ \bar{z}_i - \bar{z}_{i-1} \in \gamma(A_i + T_{i-1})(\bar{x}), \quad \forall i \in [\![2, n-1]\!], \\ \bar{x} - \bar{z}_{n-1} \in \gamma(A_n + T_{n-1})(\bar{x}), \end{cases}$$

which implies $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{FB}$ and $\bar{x} = J_{\gamma A_1}(\bar{z}_1) \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\right)$.

In other words, $\bar{\mathbf{x}} = (\bar{x}, \ldots, \bar{x}) \in \mathcal{H}^n$ with $\bar{x} := J_{\gamma A_1}(\bar{z}_1)$ being the unique weak sequential cluster point of the bounded sequence $(\mathbf{x}^k)_{k\in\mathbb{N}}$. We therefore deduce that $(\mathbf{x}^k)_{k\in\mathbb{N}}$ converges weakly to $\bar{\mathbf{x}}$, which completes this part of the proof.

(iii): For convenience, denote $\mathbf{y}^k = (y_1^k, \ldots, y_n^k)$ where

$$\begin{cases} y_1^k := z_1^k, \\ y_i^k := z_i^k + x_{i-1}^k - z_{i-1}^k - \gamma T_{i-1}(x_{i-1}^k), \quad \forall i \in [\![2, n-1]\!], \\ y_n^k := x_1^k + x_{n-1}^k - z_{n-1}^k - \gamma T_{n-1}(x_{n-1}^k), \end{cases}$$

so that $x_i^k = J_{\gamma A_i}(y_i^k)$ for all $i \in [\![1, n]\!]$. Define $\bar{\mathbf{y}} = (\bar{y}_1, \ldots, \bar{y}_n)$ in an analogous way with $\bar{\mathbf{z}}$ in place of $\mathbf{z}^k$ and $(\bar{x}, \ldots, \bar{x})$ in place of $\mathbf{x}^k$, so that $\bar{x} = J_{\gamma A_i}(\bar{y}_i)$ for all $i \in [\![1, n]\!]$. As resolvents are firmly nonexpansive, Proposition 2.3 yields

$$\begin{aligned} 0 &\leq \sum_{i=1}^{n} \langle J_{\gamma A_i}(y_i^k) - J_{\gamma A_i}(\bar{y}_i), (\mathrm{Id} - J_{\gamma A_i})(y_i^k) - (\mathrm{Id} - J_{\gamma A_i})(\bar{y}_i) \rangle \\ &= \langle x_1^k - \bar{x}, (z_1^k - x_1^k) - (\bar{z}_1 - \bar{x}) \rangle \\ &\quad + \sum_{i=2}^{n-1} \langle x_i^k - \bar{x}, (z_i^k - x_i^k) - (z_{i-1}^k - x_{i-1}^k) - \gamma T_{i-1}(x_{i-1}^k) \rangle \\ &\quad + \langle x_n^k - \bar{x}, x_1^k - x_n^k - (z_{n-1}^k - x_{n-1}^k) - \gamma T_{n-1}(x_{n-1}^k) \rangle \\ &\quad - \sum_{i=2}^{n-1} \langle x_i^k - \bar{x}, \bar{z}_i - \bar{z}_{i-1} - \gamma T_{i-1}(\bar{x}) \rangle - \langle x_n^k - \bar{x}, \bar{x} - \bar{z}_{n-1} - \gamma T_{n-1}(\bar{x}) \rangle \\ &= \langle x_1^k - x_n^k, (z_1^k - x_1^k) - (\bar{z}_1 - \bar{x}) \rangle + \langle x_n^k - \bar{x}, (z_1^k - x_1^k) - (\bar{z}_1 - \bar{x}) \rangle \\ &\quad + \sum_{i=2}^{n-1} \langle x_i^k - x_n^k, (z_i^k - x_i^k) - (z_{i-1}^k - x_{i-1}^k) - (\bar{z}_i - \bar{z}_{i-1}) \rangle \\ &\quad + \langle x_n^k - \bar{x}, (z_{n-1}^k - x_{n-1}^k) - (z_1^k - x_1^k) - (\bar{z}_{n-1} - \bar{z}_1) \rangle \\ &\quad - \gamma \sum_{i=1}^{n-1} \langle x_{i+1}^k - x_i^k, T_i(x_i^k) - T_i(\bar{x}) \rangle - \gamma \sum_{i=1}^{n-1} \langle x_i^k - \bar{x}, T_i(x_i^k) - T_i(\bar{x}) \rangle \\ &\quad + \langle x_n^k - \bar{x}, x_1^k - x_n^k \rangle - \langle x_n^k - \bar{x}, (z_{n-1}^k - x_{n-1}^k) + (\bar{x} - \bar{z}_{n-1}) \rangle. \end{aligned} \tag{6.17}$$

Rearranging (6.17) followed by applying $\frac{1}{\beta}$-cocoercivity of $T_1, \ldots, T_{n-1}$ gives

$$\begin{aligned}
&\langle x_n^k - \bar{x}, x_1^k - x_n^k\rangle + \langle x_1^k - x_n^k, (z_1^k - x_1^k) - (\bar{z}_1 - \bar{x})\rangle - \gamma \sum_{i=1}^{n-1} \langle x_{i+1}^k - x_i^k, T_i(x_i^k) - T_i(\bar{x})\rangle \\
&\quad + \sum_{i=2}^{n-1} \langle x_i^k - x_n^k, (z_i^k - x_i^k) - (z_{i-1}^k - x_{i-1}^k) - (\bar{z}_i - \bar{z}_{i-1})\rangle \qquad (6.18) \\
&\quad \geq \gamma \sum_{i=1}^{n-1} \langle x_i^k - \bar{x}, T_i(x_i^k) - T_i(x)\rangle \geq \frac{\gamma}{\beta} \sum_{i=1}^{n-1} \|T_i(x_i^k) - T_i(\bar{x})\|^2.
\end{aligned}$$

Note that the left-hand side of (6.18) converges to zero due to (6.14) and the boundedness of the sequences $(\mathbf{z}^k)_{k\in\mathbb{N}}, (\mathbf{x}^k)_{k\in\mathbb{N}}$ and $\big(T_i(x_i^k)\big)_{k\in\mathbb{N}}$ for $i \in [\![1, n-1]\!]$. It then follows that $T_i(x_i^k) \to T_i(\bar{x})$ for all $i \in [\![1, n-1]\!]$, as claimed. □

Remark 6.8 (Attouch–Théra duality). Let $I \subseteq \{1, \ldots, n-1\}$ be a non-empty index set with cardinality denoted by $|I|$. Express the monotone inclusion (6.1) as

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i\in I} T_i(x) + \left(\sum_{i=1}^{n} A_i + \sum_{i\notin I} T_i\right)(x), \qquad (6.19)$$

and note that the first operator $\sum_{i\in I} T_i$ is $\frac{1}{|I|\beta}$-cocoercive, by Proposition 2.4. The Attouch–Théra dual problem [35] associated with (6.19) takes the form

$$\text{find } u \in \mathcal{H} \text{ such that } 0 \in \left(\sum_{i\in I} T_i\right)^{-1}(u) - \left(\sum_{i=1}^{n} A_i + \sum_{i\notin I} T_i\right)^{-1}(-u), \qquad (6.20)$$

where we note that the first operator $\big(\sum_{i\in I} T_i\big)^{-1}$ is $\frac{1}{|I|\beta}$-strongly monotone. Hence, as a strongly monotone inclusion, (6.20) has a unique solution $\bar{u} \in \mathcal{H}$. Moreover, for any solution $\bar{x} \in \mathcal{H}$ of (6.19), Theorem 3.3 implies $\bar{u} = \big(\sum_{i\in I} T_i\big)(\bar{x})$. In the context of the previous result, Theorem 6.7 (iii) implies $\sum_{i\in I} T_i(x_i^k) \to \bar{u}$ as $k \to \infty$. In other words, Algorithm 6.6 also produces a sequence which converges strongly to the unique solution of the dual inclusion (6.20).

Remark 6.9. Theorem 6.7 is linked to the convergence result of different splitting algorithms:

(i) When $T_1 = \cdots = T_{n-1} = 0$, Theorem 6.7 recovers the convergence proof of the Malitsky–Tam resolvent splitting in [173, Theorem 2].

(ii) In the special case when $n = 2$, $\mathcal{T}_{FB} : \mathcal{H} \to \mathcal{H}$ and equality (6.5) from Lemma 6.5 simplifies to give the stronger inequality

$$\|\mathcal{T}_{FB}(\mathbf{z}) - \mathcal{T}_{FB}(\bar{\mathbf{z}})\|^2 + \left(\frac{2-\lambda}{\lambda} - \frac{\gamma\beta}{2\lambda}\right) \|(\mathrm{Id} - \mathcal{T}_{FB})(\mathbf{z}) - (\mathrm{Id} - \mathcal{T}_{FB})(\bar{\mathbf{z}})\|^2 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2. \quad (6.21)$$

This ensures averagedness of $\mathcal{T}_{FB}$, which in this case recovers the Davis–Yin splitting, provided that $\lambda \in \left]0, 2 - \frac{\gamma\beta}{2}\right[$, which is larger than the range of permissible values for λ in the statement of Theorem 6.7. However, by using (6.21), and modifying the proof of Theorem 6.7 accordingly, guarantees the convergence for the same range of parameters as Theorem 5.3, namely, when $\gamma \in \left]0, 4/\beta\right[$ and $\lambda \in \left]0, 2 - \frac{\gamma\beta}{2}\right[$. Indeed, note that (6.21) is equivalent to (5.16) when $\lambda_k = \lambda$ for all $k \in \mathbb{N}$.

In turn, observe that (6.21) does not imply that the Davis–Yin splitting operator $\mathcal{T}_{DY}^{\gamma}$, defined in (5.2), is $\frac{2\lambda}{4-\gamma\beta}$-averaged nonexpansive, as what we have is that

$$\mathcal{T}_{FB} = (1 - \lambda)\,\mathrm{Id} + \lambda \mathcal{T}_{DY}^{\gamma}.$$

Remark 6.10 (Algorithm 6.6 has minimal lifting). Algorithm 6.6 is a frugal forward-backward splitting with $(n-1)$-fold lifting for (6.1) with $n \geq 2$. Consequently, Fact 4.13 implies that its lifting is minimal. In turn, the Malitsky–Tam resolvent splitting, which is directly recovered by Algorithm 6.6 when $T_1 = \ldots = T_{n-1} = 0$, is a frugal resolvent splitting with minimal lifting due to Theorem 4.9.

We recall that the version of Problem 3.9 with $m > n - 1$ cocoercive operators, $\widetilde{T}_1, \ldots, \widetilde{T}_m$, can be tackled with Algorithm 6.6 by just distributing the operators between $T_1, \ldots, T_{n-1}$ (for instance, setting $T_1 := \widetilde{T}_1$, $\ldots$, $T_{n-2} := \widetilde{T}_{n-2}$, $T_{n-1} := \sum_{j=n-1}^{m} \widetilde{T}_m$). In view of Fact 4.13, the resulting method will continue to have minimal lifting.

In contrast, if we force the number of set-valued operators to be the same than the number of single-valued operators (i.e., $n-1$), Algorithm 6.6 becomes the iteration given in Remark 6.3, which has $(n-1)$-fold lifting. Hence, there is no reduction in the lifting with respect to the number of set-valued operators, and the resulting scheme is not minimal according to Fact 4.13. Nonetheless, [187, Corollary 6.7] implies that the lifting of this scheme is minimal among the frugal forward-backward splittings that perform a forward evaluation before any backward evaluation.

6.1.3 Application to distributed decentralized optimization

Advances in hardware (parallel computation) and increasing the size of datasets (decentralized storage) have made distributed algorithms one of the most prevalent trends in algorithm development. Such algorithms rely on a network of devices that perform subtasks and are able to communicate with each other. For details on the topic, the reader is referred to the book of Bertsekas and Tsitsiklis [56].

From the perspective of distributed computing, product space formulations generally require the computation of a global sum across all nodes in every iteration. To be more concrete, consider a distributed implementation of the generalized forward-backward in Theorem 3.10 in which node i performs the z_i-updates by using the operators A_i and T. To perform the x-update, the local variables $z_1, \ldots, z_n$ must be aggregated and the result then broadcast to the entire network. There may be many reasons why this is not desirable including default network setting, privacy or cost issues.

Another important aspect of distributed communication is parallelism and synchronization. Returning to our example involving the generalized forward-backward, the product space reformulation provides a fully parallel algorithm in the sense that all nodes performing z-update can compute their updates in parallel before sending to the central coordinator. This parallelization comes at cost of requiring global synchronization between nodes. Specially, the algorithm defined by (3.19) cannot move from the k-th to the $(k+1)$-th iteration until all nodes $1, \ldots, n$ have completed their computation. This can be overcome with *asynchronous algorithms*, that is, those which only require little or no global synchronization. However, their development and mathematical analysis are significantly more delicate.

The structure of Algorithm 6.6 lends itself to a distributed decentralized implementation. More precisely, consider a cycle graph with n nodes labeled 1 through n. Each node in the graph represents an agent, and two agents can communicate only if their nodes are adjacent. In our setting, this means that Agent i can only communicate with Agents $i-1$ and $i+1 \mod n$, for $i \in [\![1, n]\!]$. We assume that each agent only knows its operators in (6.1). Specifically, we assume that only Agent 1 knows the operator A_1 and that, for each $i \in [\![2, n]\!]$, only Agent i knows the operators A_i and T_{i-1}. The responsibility of updating x_i is assigned to Agent i for all $i \in [\![1, n]\!]$ and the responsibility of updating z_i is assigned to Agent i for $i \in [\![2, n]\!]$. This way, the order in which variables are updated can vary significantly between executions: z_i^{k+1} can be computed before evaluation of $z_{i+1}^k, z_{i+2}^k, \ldots$ Altogether, this gives rise to the protocol for decentralized implementation

of Algorithm 6.6 described in Algorithm 6.11 and illustrated in Figure 6.1.

Algorithm 6.11 Protocol for decentralized implementation of Algorithm 6.6.

Require: $\gamma \in \left]0, 2/\beta\right[$ and $\lambda \in \left]0, 1 - \frac{\gamma\beta}{2}\right[$.

For each $i \in [\![2, n]\!]$, Agent i chooses $z_{i-1}^0 \in \mathcal{H}$ and sends it to Agent $i-1$.

for $k = 0, 1, \ldots$ **do**

Agent 1 computes $x_1^k = J_{\gamma A_1}(z_1^k)$ and sends it to Agents 2 and n.

for $i = 2, \ldots, n-1$ **do**

Agent i computes

$$\begin{cases} x_i^k = J_{\gamma A_i}\big(z_i^k + x_{i-1}^k - z_{i-1}^k - \gamma T_{i-1}(x_{i-1}^k)\big), \\ z_{i-1}^{k+1} = z_{i-1}^k + \lambda(x_i^k - x_{i-1}^k), \end{cases}$$

sends x_i^k to Agent $i+1$ and sends z_{i-1}^{k+1} to Agent $i-1$.

end for

Agent n computes

$$\begin{cases} x_n^k = J_{\gamma A_n}\big(x_1^k + x_{n-1}^k - z_{n-1}^k - \gamma T_{n-1}(x_{n-1}^k)\big), \\ z_{n-1}^{k+1} = z_{n-1}^k + \lambda(x_n^k - x_{n-1}^k), \end{cases}$$

sends z_{n-1}^{k+1} to Agent $n-1$.

end for

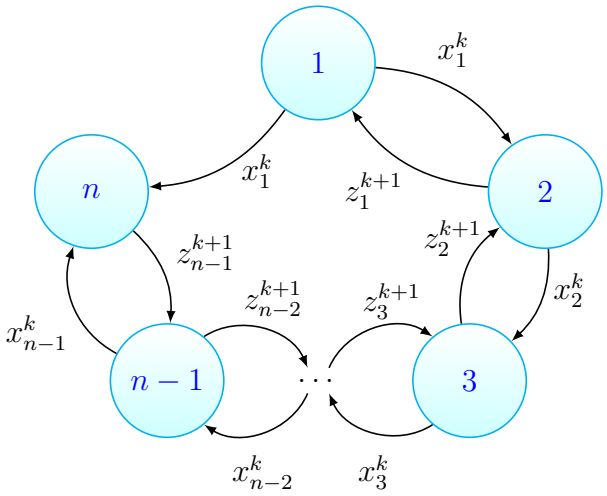


FIGURE 6.1: Distributed implementation of Algorithm 6.6 in a decentralized ring network with n nodes following the protocol in Algorithm 6.11.

REMARK 6.12 (Termination criterion for Algorithm 6.11). Let $(\mathbf{z}^k)_{k\in\mathbb{N}}$ be the sequence generated by Algorithm 6.11. In order to detect termination, one could compute (possibly

periodically) the residual given by

$$\|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2 = \sum_{i=1}^{n-1} \|z_i^{k+1} - z_i^k\|^2.$$

The structure of this residual is suitable for the decentralized implementation within the protocol in the algorithm. Indeed, the i-th term in the sum, given by $\|z_i^{k+1} - z_i^k\|^2$, can already be evaluated by Agent $i+1$, and therefore the full residual $\|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2$ can be computed by a global summation and broadcast operation. The same stopping criterion can also be applied to the algorithm presented in Section 6.2 generated by the iteration given in (6.23) and (6.24).

Remark 6.13 (Connection to other methods for decentralized optimization). The algorithm given by Algorithm 6.11 appears new even in the special case with $A_i = 0$ and $T_i = \nabla f_i$ for convex smooth functions f_i. In this case, one of the most popular algorithms for solving $\min_x \sum_{i=1}^m f_i(x)$ in a decentralized way is EXTRA, proposed in [225]. They are similar in spirit, but also have quite different properties. In particular, the main update of EXTRA is

$$\mathbf{x}^{k+1} = (\mathrm{Id} + W)\mathbf{x}^k - \widetilde{W}\mathbf{x}^{k-1} - \gamma[\nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^{k-1})], \quad \forall k \in \mathbb{N},$$

where W and $\widetilde{W}$ are certain mixing matrices and $\mathbf{x}^1 = W\mathbf{x}^0 - \gamma\nabla f(\mathbf{x}^0)$. The method was later extended to PG-EXTRA [226] which also includes proximal steps of nonsmooth convex functions. Likewise, PG-EXTRA does not recover Algorithm 6.11. Undoubtedly, an advantage of EXTRA and PG-EXTRA is the ability to use a wider range of mixing matrices which, in terms of communication, generalizes better for network topology.

Importantly, we believe that Algorithm 6.11 constitutes an important starting point towards a more general template that will allow for a full distributed implementation of forward-backward splittings with minimal lifting for different network topologies. Some steps in this direction have been done for the case $T_1 = \ldots = T_{n-1} = 0$ in [72, 231].

6.2 A forward-reflected-backward method with reduced lifting

We now focus our attention on designing a forward-backward type method for the instance of Problem 3.9 in which the single-valued operators are monotone and Lipschitz

continuous. In this section, we assume $n \geq 3$ and consider the monotone inclusion

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-2} T_i\right)(x), \tag{6.22}$$

where $A_1, \ldots, A_n \colon \mathcal{H} \rightrightarrows \mathcal{H}$ are maximally monotone and $T_1, \ldots, T_{n-1} \colon \mathcal{H} \to \mathcal{H}$ are monotone and β-Lipschitz continuous operators, with $\beta > 0$. Again, the motivation behind the different number of operators is to provide more freedom on how to distribute the evaluation of single-valued operators. Nevertheless, it is always possible to equate them by setting some of the set-valued operators to zero, or equivalently, by considering the sum of certain single-valued operators.

6.2.1 A fixed point encoding with $(n-1)$-fold lifting

As demonstrated in Section 3.2.3, the algorithmic schemes capable of solving (6.22) are fundamentally different to those used to tackle (6.1). In this section, we develop a modification of the method from the previous section which converges for Lipschitz continuous operators by drawing inspiration from the differences between the iterations of the forward-backward and the forward-reflected-backward methods in (3.15) and (3.22), respectively.

Algorithm 6.14 Forward-reflected-backward splitting for (6.22) with reduced lifting.

Require: $\gamma \in \left]0, \frac{1}{2\beta}\right[$ and $\lambda \in]0, 1 - 2\gamma\beta[$.

1: Choose $\mathbf{z}^0 = (z_1^0, \ldots, z_{n-1}^0) \in \mathcal{H}^{n-1}$.

2: **for** $k = 0, 1, \ldots$ **do**

3: Compute

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \lambda \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \end{pmatrix}, \tag{6.23}$$

with $\mathbf{x}^k = (x_1^k, \ldots, x_n^k) \in \mathcal{H}^n$ computed as

$$\begin{cases} x_1^k = J_{\gamma A_1}(z_1^k), \\ x_2^k = J_{\gamma A_2}\big(z_2^k + x_1^k - z_1^k - \gamma T_1(x_1^k)\big), \\ x_i^k = J_{\gamma_i A_i}\big(z_i^k + x_{i-1}^k - z_{i-1}^k - \gamma T_{i-1}(x_{i-1}^k) - \gamma(T_{i-2}(x_{i-1}^k) - T_{i-2}(x_{i-2}^k))\big), \quad \forall i \in [\![3, n-1]\!], \\ x_n^k = J_{\gamma A_n}\big(x_1^k + x_{n-1}^k - z_{n-1}^k - \gamma(T_{n-2}(x_{n-1}^k) - T_{n-2}(x_{n-2}^k))\big). \end{cases} \tag{6.24}$$

4: **end for**

Compared to Algorithm 6.6, the only major change here is that some expressions for x_i^k in (6.24) incorporate a reflection-type term involving the operator T_{i-2}. This precise form seems important for our subsequence convergence analysis and it seems not easy to incorporate reflection-type terms involving the operator T_{i-1}. Observe that now the single-valued operators T_i, for $i \in [\![1, n-2]\!]$, are evaluated twice per iteration. The structure of Algorithm 6.14 allows for a similar protocol to the one described in Algorithm 6.11 to be used for a distributed decentralized implementation. The only change to the protocol (in terms of communication) is that Agent i must also now send $\gamma\big(T_{i-1}(x_i^k) - T_{i-1}(x_{i-1}^k)\big)$ to Agent $i+1$ for all $i \in [\![2, n-1]\!]$.

REMARK 6.15. To the best of our knowledge, the scheme given by Algorithm 6.14 does not directly recover any existing forward-backward-type scheme as special case (although it is clearly related to the forward-reflected-backward). For example, if we take $n = 3$, $A_2 = A$, $T_1 = T$, and $A_1 = A_3 = 0$. Then x_1^k and x_3^k can be eliminated from Algorithm 6.14 to give

$$\begin{cases} x_2^k = J_{\gamma A}\big(z_2^k - \gamma T(z_1^k)\big), \\ z_1^{k+1} = z_1^k + \lambda\big(x_2^k - z_1^k\big), \\ z_2^{k+1} = z_2^k + \lambda\big(z_1^k - z_2^k - \gamma(T(x_2^k) - T(z_1^k))\big). \end{cases}$$

To better understand the relationship between this and (3.22), it is instructive to consider the limiting case with $\lambda = 1$. Indeed, when $\lambda = 1$, x_2^k and z_2^k can be eliminated to give

$$z_1^{k+1} = J_{\gamma A}\big(z_1^{k-1} - 2\gamma T(z_1^k) + \gamma T(z_1^{k-1})\big).$$

Although this closely resembles (3.22) for finding a zero of $A + T$, it is not exactly the same due to the index of the first term inside the resolvent.

In order to analyze Algorithm 6.14, we introduce the underlying fixed point operator $\mathcal{T}_{FRB}\colon \mathcal{H}^{n-1} \to \mathcal{H}^{n-1}$ given by

$$\mathcal{T}_{FRB}(\mathbf{z}) := \mathbf{z} + \lambda \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix}, \tag{6.25}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{H}^n$ depends on $\mathbf{z} = (z_1, \ldots, z_n) \in \mathcal{H}$ and is given by

$$\begin{cases} x_1 = J_{\gamma A_1}(z_1), \\ x_2 = J_{\gamma A_2}\big(z_2 + x_1 - z_1 - \gamma T_1(x_1)\big), \\ x_i = J_{\gamma A_i}\big(z_i + x_{i-1} - z_{i-1} - \gamma T_{i-1}(x_{i-1}) - \gamma(T_{i-2}(x_{i-1}) - T_{i-2}(x_{i-2}))\big), \quad \forall i \in [\![3, n-1]\!], \\ x_n = J_{\gamma A_n}\big(x_1 + x_{n-1} - z_{n-1} - \gamma(T_{n-2}(x_{n-1}) - T_{n-2}(x_{n-2}))\big). \end{cases} \tag{6.26}$$

In this way, the sequence $(\mathbf{z}^k)$ given by (6.23) satisfies $\mathbf{z}^{k+1} = \mathcal{T}_{FRB}(\mathbf{z}^k)$ for all $k \in \mathbb{N}$. Furthermore, $(\mathcal{T}_{FRB}, J_{\gamma A_1})$ defines a fixed point encoding with $(n-1)$-fold lifting for (6.22), as proved in the following lemma.

Lemma 6.16. *Let $n \geq 3$ and $\lambda, \gamma > 0$. Then the following assertions hold.*

(i) If $\bar{x} \in \operatorname{zer}\big(\sum_{i=1}^n A_i + \sum_{i=1}^{n-2} T_i\big)$, then there exists $\bar{z} \in \operatorname{Fix} \mathcal{T}_{FRB}$.

(ii) If $(\bar{z}_1, \ldots, \bar{z}_{n-1}) \in \operatorname{Fix} \mathcal{T}_{FRB}$, then $\bar{x} := J_{\gamma A_1}(\bar{z}_1) \in \operatorname{zer}\big(\sum_{i=1}^n A_i + \sum_{i=1}^{n-2} T_i\big)$. Moreover,

$$\bar{x} = J_{\gamma A_i}\big(\bar{z}_i - \bar{z}_{i-1} + \bar{x} - \gamma T_{i-1}(\bar{x})\big) = J_{\gamma A_n}(2\bar{x} - \bar{z}_{n-1}), \tag{6.27}$$

for all $i \in [\![2, n-1]\!]$.

Consequently,

$$\operatorname{Fix} \mathcal{T}_{FRB} \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^n A_i + \sum_{i=1}^{n-2} T_i\right) \neq \emptyset.$$

Proof. The proof follows similarly to that of Lemma 6.4. □

6.2.2 Convergence analysis

Next, we analyze the nonexpansiveness properties of the operator $\mathcal{T}_{FRB}$. The proof of the following result is similar to that of Lemma 6.5, but using the Lipschitzian properties of the operators $T_1, \ldots, T_{n-2}$ instead of cocoercivity.

Lemma 6.17. *Let $\bar{\mathbf{z}} = (\bar{z}_1, \ldots, \bar{z}_{n-1}) \in \operatorname{Fix} \mathcal{T}_{FRB}$. For all $\mathbf{z} = (z_1, \ldots, z_{n-1}) \in \mathcal{H}^{n-1}$, we have*

$$\begin{aligned} \|\mathcal{T}_{FRB}(\mathbf{z}) - \bar{\mathbf{z}}\|^2 + \left(\frac{1-\lambda}{\lambda} - \frac{2\gamma\beta}{\lambda}\right) \|(\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})\|^2 + \frac{1}{\lambda}\Big\|\sum_{i=1}^{n-1}(\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})_i\Big\|^2 \\ + \frac{\gamma\beta}{\lambda}\|(\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})_1\|^2 + \frac{\gamma\beta}{\lambda}\|(\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})_{n-1}\|^2 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2. \end{aligned} \tag{6.28}$$

In particular, if $\gamma \in \left]0, \frac{1}{2\beta}\right[$ *and* $\lambda \in]0, 1-2\gamma\beta[$, *then* $\mathcal{T}_{FRB}$ *is* σ*-strongly quasi-nonexpansive for* $\sigma = \frac{1-\lambda}{\lambda} - \frac{2\gamma\beta}{\lambda} > 0$.

Proof. For convenience, denote $\mathbf{z}^+ = \mathcal{T}_{FRB}(\mathbf{z})$. Further, let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{H}^n$ be given by (6.26) and let $\bar{\mathbf{x}} = (\bar{x}, \ldots, \bar{x}) \in \mathcal{H}^{n-1}$ be given analogously. Note that the expression of $\bar{\mathbf{x}}$ is justified by (6.27) as $\bar{\mathbf{z}} = \mathcal{T}_{FRB}(\bar{\mathbf{z}})$. Monotonicity of γA_1 implies

$$0 \leq \langle x_2 - \bar{x}, (z_1 - x_1) - (\bar{z}_1 - \bar{x})\rangle + \langle x_1 - x_2, (z_1 - x_1) - (\bar{z}_1 - \bar{x})\rangle. \tag{6.29}$$

In order to simplify the case study, we introduce the zero operator $T_0 := 0$. By monotonicity of γA_i, we deduce

$$\begin{aligned} 0 \leq\ & \langle x_{i+1} - \bar{x}, (z_i - x_i) - (\bar{z}_i - \bar{x})\rangle + \langle x_i - x_{i+1}, (z_i - x_i) - (\bar{z}_i - \bar{x})\rangle \\ & - \langle x_i - \bar{x}, (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x})\rangle \\ & - \gamma\langle x_i - \bar{x}, T_{i-1}(x_{i-1}) - T_{i-1}(\bar{x})\rangle \\ & - \gamma\langle x_i - \bar{x}, T_{i-2}(x_{i-1}) - T_{i-2}(x_{i-2})\rangle, \end{aligned} \tag{6.30}$$

for all $i \in [\![2, n-1]\!]$, and monotonicity of γA_n yields

$$\begin{aligned} 0 \leq\ & -\langle x_n - \bar{x}, (z_{n-1} - x_{n-1}) - (\bar{z}_{n-1} - \bar{x})\rangle \\ & - \gamma\langle x_n - \bar{x}, T_{n-2}(x_{n-1}) - T_{n-2}(x_{n-2})\rangle \\ & + \frac{1}{2}\left(\|x_1 - \bar{x}\|^2 - \|x_n - \bar{x}\|^2 - \|x_1 - x_n\|^2\right). \end{aligned} \tag{6.31}$$

Summing together (6.29)-(6.31), we obtain the inequality

$$\begin{aligned} 0 \leq\ & \sum_{i=1}^{n-1} \langle (\bar{z}_i - \bar{x}) - (z_i - x_i), x_{i+1} - x_i\rangle \\ & + \frac{1}{2}\left(\|x_1 - \bar{x}\|^2 - \|x_n - \bar{x}\|^2 - \|x_1 - x_n\|^2\right) \\ & - \gamma \sum_{i=2}^{n-1} \langle x_i - \bar{x}, T_{i-1}(x_{i-1}) - T_{i-1}(\bar{x})\rangle \\ & - \gamma \sum_{i=3}^{n} \langle x_i - \bar{x}, T_{i-2}(x_{i-1}) - T_{i-2}(x_{i-2})\rangle, \end{aligned} \tag{6.32}$$

where we have omitted the index $i = 2$ in the last sum, since $T_0 := 0$. The first term in (6.32) multiplied by 2λ can be written as

$$\begin{aligned} &2\lambda \sum_{i=1}^{n-1} \langle (\bar{z}_i - \bar{x}) - (z_i - x_i), x_{i+1} - x_i \rangle \\ &\qquad = \sum_{i=1}^{n-1} \left(\|\bar{z}_i - z_i\|^2 + \|z_i^+ - z_i\|^2 - \|z_i^+ - \bar{z}_i\|^2 \right) \\ &\qquad\quad - \frac{1}{\lambda} \sum_{i=1}^{n-1} \|z_i^+ - z_i\|^2 + \lambda \left(\|x_n - \bar{x}\|^2 - \|x_1 - \bar{x}\|^2 \right). \end{aligned} \tag{6.33}$$

Therefore, multiplying (6.32) by 2λ and substituting (6.33), we reach the inequality

$$\begin{aligned} &\|\mathcal{T}_{FRB}(\mathbf{z}) - \bar{\mathbf{z}}\|^2 + \frac{1-\lambda}{\lambda} \|(\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})\|^2 + \frac{1}{\lambda} \Big\| \sum_{i=1}^{n-1} (\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})_i \Big\|^2 \\ &\qquad \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2 - 2\lambda\gamma \sum_{i=2}^{n-1} \langle x_i - \bar{x}, T_{i-1}(x_{i-1}) - T_{i-1}(\bar{x}) \rangle \\ &\qquad\quad - 2\lambda\gamma \sum_{i=3}^{n} \langle x_i - \bar{x}, T_{i-2}(x_{i-1}) - T_{i-2}(x_{i-2}) \rangle. \end{aligned} \tag{6.34}$$

Using monotonicity of $T_1, \ldots, T_{n-2}$, the second last term can be estimated as

$$-\sum_{i=2}^{n-1} \langle x_i - \bar{x}, T_{i-1}(x_{i-1}) - T_{i-1}(\bar{x}) \rangle \leq \sum_{i=2}^{n-1} \langle x_i - \bar{x}, T_{i-1}(x_i) - T_{i-1}(x_{i-1}) \rangle \tag{6.35}$$

and, using β-Lipschitz continuity of $T_1, \ldots, T_{n-2}$, the last term can be estimated as

$$\begin{aligned} &-\sum_{i=3}^{n} \langle x_i - \bar{x}, T_{i-2}(x_{i-1}) - T_{i-2}(x_{i-2}) \rangle \\ &\qquad = -\sum_{i=3}^{n} \langle x_{i-1} - \bar{x}, T_{i-2}(x_{i-1}) - T_{i-2}(x_{i-2}) \rangle \\ &\qquad\quad + \sum_{i=3}^{n} \langle x_{i-1} - x_i, T_{i-2}(x_{i-1}) - T_{i-2}(x_{i-2}) \rangle \\ &\qquad \leq -\sum_{i=3}^{n} \langle x_{i-1} - \bar{x}, T_{i-2}(x_{i-1}) - T_{i-2}(x_{i-2}) \rangle \\ &\qquad\quad + \frac{\beta}{2} \sum_{i=3}^{n} \left(\|x_{i-1} - x_i\|^2 + \|x_{i-1} - x_{i-2}\|^2 \right) \end{aligned}$$

$$
\begin{aligned}
&= -\sum_{i=2}^{n-1}\langle x_i - \bar{x}, T_{i-1}(x_i) - T_{i-1}(x_{i-1})\rangle + \beta\sum_{i=2}^{n}\|x_i - x_{i-1}\|^2 \\
&\quad - \frac{\beta}{2}\|x_2 - x_1\|^2 - \frac{\beta}{2}\|x_n - x_{n-1}\|^2 \\
&= -\sum_{i=2}^{n-1}\langle x_i - \bar{x}, T_{i-1}(x_i) - T_{i-1}(x_{i-1})\rangle + \frac{\beta}{\lambda^2}\|(\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})\|^2 \\
&\quad - \frac{\beta}{2\lambda^2}\|(\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})_1\|^2 - \frac{\beta}{2\lambda^2}\|(\mathrm{Id} - \mathcal{T}_{FRB})(\mathbf{z})_{n-1}\|^2. \qquad (6.36)
\end{aligned}
$$

Thus, substituting (6.35) and (6.36) into (6.34) gives (6.28). □

REMARK 6.18. Compared to Lemma 6.5 from the previous section, the conclusions of Lemma 6.17 are weaker in two ways. Firstly, the permissible stepsize range of $\gamma \in \left]0, \frac{1}{2\beta}\right[$ is smaller than in Lemma 6.5, which allowed $\gamma \in \left]0, \frac{2}{\beta}\right[$. And, secondly, the operator $\mathcal{T}_{FRB}$ is only shown to be strongly quasi-nonexpansive in Lemma 6.17 whereas $\mathcal{T}_{FB}$ is known to be averaged nonexpansive.

The following theorem is our main result regarding convergence of Algorithm 6.14. Observe that, as $\mathcal{T}_{FRB}$ is not averaged nonexpansive, we cannot prove the weak convergence of the sequence $(\mathbf{z}^k)_{k\in\mathbb{N}}$ generated by Algorithm 6.14 to a fixed point of $\mathcal{T}_{FRB}$ using Theorem 2.7.

Theorem 6.19. *Let $n \geq 3$, let $A_1, \ldots, A_n \colon \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone and let $T_1, \ldots, T_{n-2} \colon \mathcal{H} \to \mathcal{H}$ be monotone and β-Lipschitz continuous. Further, assume that $\operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-2} T_i\right) \neq \emptyset$ and let $\gamma \in \left]0, \frac{1}{2\beta}\right[$ and $\lambda \in \left]0, 1 - 2\gamma\beta\right[$. Given $\mathbf{z}^0 \in \mathcal{H}^{n-1}$, let $(\mathbf{z}^k) \subseteq \mathcal{H}^{n-1}$ and $(\mathbf{x}^k) \subseteq \mathcal{H}^n$ be the sequences generated by Algorithm 6.14. Then the following assertions hold.*

(i) The sequence $(\mathbf{z}^k)_{k\in\mathbb{N}}$ converges weakly to a point $\bar{\mathbf{z}} \in \operatorname{Fix}\mathcal{T}_{FRB}$.

(ii) The sequence $(\mathbf{x}^k)_{k\in\mathbb{N}}$ converges weakly to a point $(\bar{x}, \ldots, \bar{x}) \in \mathcal{H}^n$ with $\bar{x}$ solving the monotone inclusion (6.22).

Proof. (i): Since $\operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-2} T_i\right) \neq \emptyset$, Lemma 6.16 (i) implies that the set of fixed points of operator $\mathcal{T}_{FRB}$ is nonempty. Since $\gamma \in \left]0, \frac{1}{2\beta}\right[$ and $\lambda \in \left]0, 1 - 2\gamma\beta\right[$, Lemma 6.17 implies that $(\mathbf{z}^k)_{k\in\mathbb{N}}$ is Fejér monotone with respect to $\operatorname{Fix}\mathcal{T}_{FRB}$ and that $\lim_{k\to+\infty}\|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$. By nonexpansiveness of resolvents, β-Lipschitz continuity of $T_1, \ldots, T_{n-2}$, and boundedness of $(\mathbf{z}^k)_{k\in\mathbb{N}}$, it follows that $(\mathbf{x}^k)_{k\in\mathbb{N}}$ is also bounded. Further,

(6.25) and the fact that $\lim_{k\to\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$ implies that

$$\lim_{k\to\infty} \|x_i^k - x_{i-1}^k\| = 0, \quad \forall i \in [\![2, n]\!]. \tag{6.37}$$

Let $\mathbf{u} = (u_1, \ldots, u_{n-1}) \in \mathcal{H}^{n-1}$ be an arbitrary weak cluster point of $(\mathbf{z}^k)_{k\in\mathbb{N}}$. Then, due to (6.37), there exists a point $x \in \mathcal{H}$ such that $(\mathbf{u}, \mathbf{w})$ is a weak cluster point of $(\mathbf{z}^k, \mathbf{x}^k)_{k\in\mathbb{N}}$, where $\mathbf{w} = (x, \ldots, x) \in \mathcal{H}^n$. Let C denote the maximally monotone operator defined by (6.16) when $T_{n-1} = 0$. Then (6.24) implies

$$C \begin{pmatrix} z_1^k - x_1^k \\ (z_2^k - x_2^k) - (z_1^k - x_1^k) + \gamma t_2^k \\ (z_3^k - x_3^k) - (z_2^k - x_2^k) + \gamma t_3^k - \gamma t_2^k \\ \vdots \\ (z_{n-1}^k - x_{n-1}^k) - (z_{n-2}^k - x_{n-2}^k) + \gamma t_{n-1}^k - \gamma t_{n-2}^k \\ x_n^k \end{pmatrix} \ni \begin{pmatrix} x_1^k - x_n^k \\ x_2^k - x_n^k \\ x_3^k - x_n^k \\ \vdots \\ x_{n-1}^k - x_n^k \\ x_1^k - x_n^k \end{pmatrix}, \tag{6.38}$$

where $t_i^k := T_{i-1}(x_i^k) - T_{i-1}(x_{i-1}^k)$. Taking the limit along a subsequence of $(\mathbf{z}^k, \mathbf{x}^k)_{k\in\mathbb{N}}$ which converges weakly to $(\mathbf{u}, \mathbf{w})$ in (6.38), using demiclosedness of C together with the β-Lipschitz continuity of $T_1, \ldots, T_{n-2}$, and unravelling the resulting expression gives that $\mathbf{u} \in \operatorname{Fix} \mathcal{T}_{FRB}$ and $x = J_{\gamma A_1}(u_1) \in \operatorname{zer}\left(\sum_{i=1}^n A_i + \sum_{i=1}^{n-2} T_i\right)$. Thus, by Proposition 2.6, it follows that $(\mathbf{z}^k)_{k\in\mathbb{N}}$ converges weakly to a point $\bar{\mathbf{z}} \in \operatorname{Fix} \mathcal{T}_{FRB}$.

(ii): Follows by using an argument analogous to the one in Theorem 6.7 (ii). □

Remark 6.20 (Algorithm 6.14 has reduced lifting). To the best of the author's knowledge, Algorithm 6.14 is the first splitting with $(n-1)$-fold lifting for solving (6.22), when $n \geq 3$. However, as the single-valued operator are evaluated twice per iteration, the method is not frugal. Therefore, we cannot apply Fact 4.13 to ensure that its lifting is minimal.

The case $n = 2$ is of particular interest. By setting $A_1 = 0$ Algorithm 6.14 reduces to the iteration

$$\begin{cases} x_2^k = J_{\gamma A_2}\big(z_2^k - \gamma T_1(z_1^k)\big), \\ x_3^k = J_{\gamma A_3}\big(z_1^k + x_2^k - z_2^k - \gamma(T_1(x_2^k) - T_1(z_1^k))\big), \\ z_1^{k+1} = z_1^k + \lambda(x_2^k - z_1^k), \\ z_2^{k+1} = z_2^k + \lambda(x_3^k - x_2^k), \end{cases}$$

which has 2-fold lifting. Hence, in this case, no reduction in the lifting is achieved with respect to the number of set-valued operators appearing in the problem.

REMARK 6.21 (Exploiting cocoercivity). If a Lipschitz continuous operator T_i in (6.22) is actually cocoercive, then it is possible to reduce the number evaluations of T_i per iteration by combining the ideas in Sections 6.1 and 6.2. In fact, we can consider the problem

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\right)(x),$$

where $T_1, \ldots, T_{n-2}$ are each either monotone and Lipschitz continuous or cocoercive, and T_{n-1} is cocoercive. For this problem, we can replace (6.26) in the definition of $\mathcal{T}_{FRB}$ with

$$\begin{cases} x_1 = J_{\gamma A_1}(z_1), \\ x_2 = J_{\gamma A_2}\big(z_2 + x_1 - z_1 - \gamma T_1(x_1)\big), \\ x_i = J_{\gamma A_i}\big(z_i + x_{i-1} - z_{i-1} - \gamma T_{i-1}(x_{i-1}) - \gamma t_{i-1}\big), \quad \forall i \in [\![3, n-1]\!], \\ x_n = J_{\lambda A_n}\big(x_1 + x_{n-1} - z_{n-1} - \gamma T_{n-1}(x_{n-1}) - \lambda t_{n-1}\big), \end{cases}$$

where $t_2, \ldots, t_{n-1} \in \mathcal{H}$ are given by

$$t_i = \begin{cases} 0, & \text{if } T_{i-1} \text{ is cocoercive,} \\ T_{i-1}(x_i) - T_{i-1}(x_{i-1}), & \text{if } T_{i-1} \text{ is monotone and Lipschitz.} \end{cases}$$

This modification can be shown to converge using a proof similar to Theorem 6.19 for $\gamma \in \left]0, \frac{1}{2\beta}\right[$. However, it is not straightforward to recover Theorem 6.7 as a special case of such a result because the stepsizes range of $\gamma \in \left]0, \frac{2}{\beta}\right[$ in the cocoercive only case (i.e., Theorem 6.7) are larger than the range in the mixed case. Moreover, Theorem 6.7 (iii) (strong convergence to dual solutions) does not have an analogue in the statement of Theorem 6.19. In addition, keeping the two cases separate allows the analysis to be as transparent as possible.

6.3 A numerical experiment in quadratic optimization

In the following, we present a simple numerical experiment that aims to compare the performance of the *Minimal lifting Forward-Backward method* (MFB) of Algorithm 6.6 and the *Generalized Forward-Backward* (GFB). We consider the constrained convex quadratic

optimization problem given as

$$\begin{aligned} \min_{x\in\mathbb{R}^m} \quad & \frac{1}{2}x^TQx + c^Tx + \mu\|x\|_1, \\ \text{s.t.} \quad & Mx = b,\ -1 \le x \le 1, \end{aligned} \tag{6.39}$$

where $\mu > 0$, $b \in \mathbb{R}^p$, $c \in \mathbb{R}^m$, $M \in \mathbb{R}^{p\times m}$ is a matrix with full row rank and $Q \in \mathbb{R}^{m\times m}$ is a symmetric positive definite matrix.

Let $S := \{x \in \mathbb{R}^m : Mx = b\}$. Assuming that S intersects the interior of $[-1,1]^m$, the minimization problem (6.39) is equivalent to the monotone inclusion

$$\text{find } x \in \mathbb{R}^m \text{ such that } 0 \in \left(\partial_{\mu\|\cdot\|_1}(x) + N_S(x) + N_{[-1,1]^m}(x) + Qx + c\right), \tag{6.40}$$

due to [215, Corollary 23.8.1] and [44, Corollary 16.50]. Observe that the single-valued operator $T(\cdot) := Q(\cdot) + c$ is $\frac{1}{\beta}$-cocoercive, with β the spectral radius of Q. Hence, (6.40) is of the form of problem (6.1). Recall that the closed formulas for the projection operators onto $[-1,1]^m$ and S were given in Propositions 2.32 and 2.35, respectively.

Altogether, taking $A_1 := \partial_{\mu\|\cdot\|_1}$, $A_2 := N_S$, $A_3 := N_{[-1,1]^m}$, $T_1 := 0$, $T_2 := T$ and choosing an initial guess $(z_0^1, z_0^2) \in \mathbb{R}^m \times \mathbb{R}^m$, the iteration in Algorithm 6.6 applied to (6.40) becomes

$$\left\lfloor \begin{aligned} x_1^k &= \operatorname{prox}_{\gamma\mu\|\cdot\|_1}(z_1^k), \\ x_2^k &= P_S(z_2^k - x_1^k - z_1^k), \\ x_3^k &= P_{[-1,1]^m}\left(x_1^k + x_2^k - z_2^k - \gamma(Qx_2^k + c)\right), \\ z_1^{k+1} &= z_1^k + \lambda(x_2^k - x_1^k), \\ z_2^{k+1} &= z_2^k + \lambda(x_3^k - x_2^k), \end{aligned} \right.$$

for all $k \in \mathbb{N}$.

In order to fairly compare MFB and GFB, we first performed an experiment for determining the best combination of parameters. We generated random problems in the follwing way. We set $\mu := 2$, $[m,p] := [750, 500]$ and generated M and c with entries randomly distributed in the interval $[-1,1]$. The matrix Q was a sparse positive definite matrix generated by Matlab's `sprandsym` inbuilt function. To ensure feasibility, we chose a vector $w \in \mathbb{R}^m$ with random entries in $]-1,1[$ and set $b := Mw$. We tried the following combination of parameters. The stepsizes of both MFB and GFB were tested in the range obtained by setting $\gamma = \bar{\gamma}\frac{1}{\beta}$ with $\bar{\gamma} \in \{0.3, 0.5, 0.7, 0.9, 1, 1.1, 1.3, 1.5, 1.7, 1.9\}$. For every γ, we tried the relaxation parameters resulting from choosing $\bar{\lambda} \in \{0.3, 0.5, 0.7, 0.9, 0.99\}$

and setting $\lambda = \bar{\lambda}\big(1 - \frac{\gamma\beta}{2}\big)$ and $\lambda = \bar{\lambda}\min\big\{\frac{3}{2}, \frac{1}{\gamma\beta} + \frac{1}{2}\big\}$, for MFB and GFB, respectively. Finally, the weights for GFB were $w_i = 1/3$, for $i \in \{1, 2, 3\}$. Figure 6.2 displays the average number of iterations over 10 random instances obtained by every method and combination of parameters. Both algorithms were initialized at the origin and stopped when the shadow sequence satisfied $\max\big\{\|Mz_1^k - b\|, \frac{\|z_1^{k+1} - z_1^k\|}{1+\|z_1^k\|^2}\big\} < 10^{-8}$. The best choice of parameters for MFB resulted from setting $(\bar{\gamma}, \bar{\lambda}) = (0.9, 0.99)$, while $(\bar{\gamma}, \bar{\lambda}) = (0.5, 0.99)$ were the best performing ones for GFB.

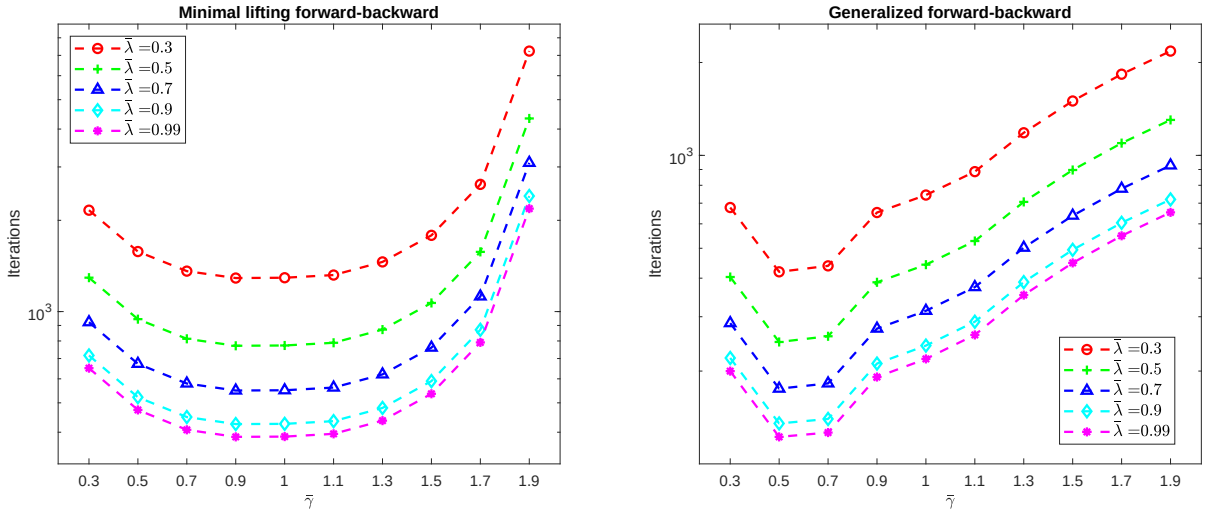


FIGURE 6.2: For 10 randomly generated problems, average number of iterations performed by MFB (left) and GFB (right) before reaching the stopping criterion using different combinations of parameters.

For this specific choice of parameters we repeated the same experiment with different dimensions. Specifically, we took $m \in \{750, 1125, 1500, 1875, 2250, 3000\}$ and $p = \frac{2}{3}m$. Figure 6.3 shows that the forward-backward method with minimal lifting was approximately 2 times faster in time than the generalized forward-backward for this particular experiment[1].

[1]The experiment was ran in a computer of Intel Core i7-12700H 2.30 GHz with 16GB RAM, under Windows 11 (64-bit)

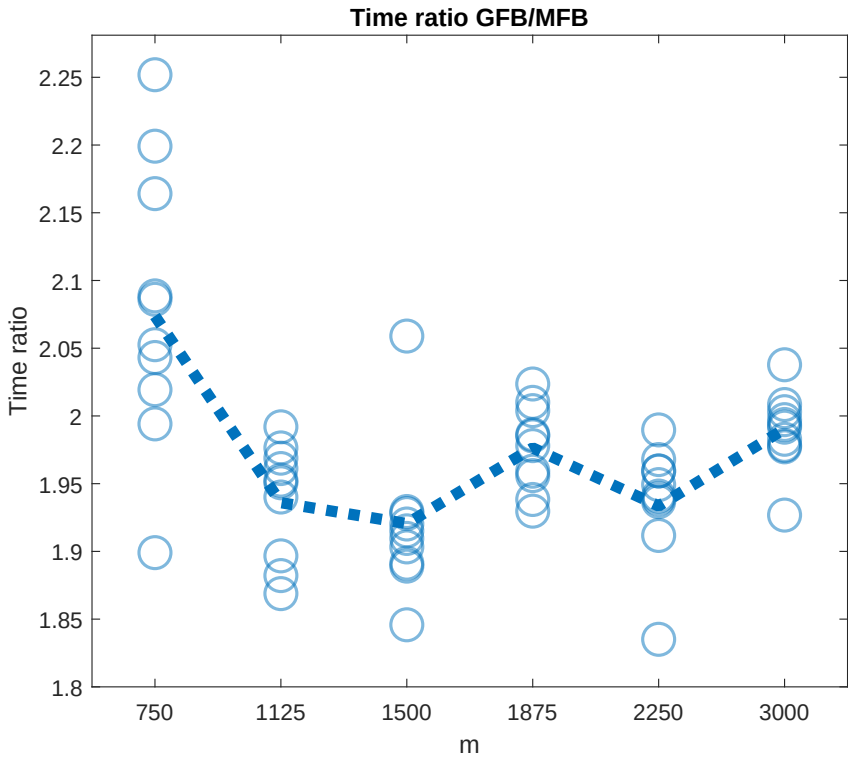


FIGURE 6.3: For each $m \in \{750, 1125, 1500, 1875, 2250, 3000\}$ and $p = \frac{2}{3}m$, we randomly generated 10 different problems and ran MFB and GFB using their respective best performing parameters. We plot with circles the ratio of the running time of every particular instance. The dotted line represents the average ratio for each choice of m.

Chapter 7

A primal-dual splitting algorithm for composite monotone inclusions with minimal lifting

In this chapter, we devise the first primal-dual resolvent splitting algorithm with minimal lifting for composite monotone inclusions in the form of Problem 3.12. The chapter is organized as follows:

- In Section 7.1 we first concentrate on Problem 3.12 with $m = 1$, i.e., only one maximally monotone operator is linearly composed. We propose a frugal primal-dual splitting with $(n-1, 1)$-fold lifting and analyze its convergence to a primal-dual solution of the problem in Theorem 7.4.
- By resorting to a product space reformulation, in Section 7.2, we obtain the general splitting with $(n-1, m)$-fold lifting presented in Algorithm 7.8. This method is minimal among all frugal primal-dual resolvent splittings with m dual variables, as discussed in Section 7.2.1.
- In Section 7.3, we devise a new splitting method for solving mixtures of monotone inclusions, namely, monotone inclusions that involve linearly composed maximally monotone operators, cocoercive operators and monotone and Lipschitz continuous operators, jointly. The proposed method arises as a combination of Algorithm 6.6, Algorithm 6.14 and Algorithm 7.8, and so it has reduced lifting with respect to the existing algorithms in the literature for this class of problems.
- Finally, in Section 7.4, we include a numerical experiment on image recovery and compare the performance of the new algorithm with the best performing primal-dual scheme for this problem.

This chapter is mainly based on the work in [10].

7.1 The case with one linear composition

In this section, we base our analysis in the case in which the primal problem involves only one linear composition, i.e., $m = 1$. Later, in Section 7.2, it will be extended to an arbitrary finite number of linearly composed maximally monotone operators by appealing to a product space reformulation.

Let $n \geq 2$. We start by considering the primal-dual composite monotone inclusion given by

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^{n} A_i(x) + L^*B(Lx), \tag{7.1}$$

and

$$\text{find } u \in \mathcal{G} \text{ such that } 0 \in -L\left(\sum_{i=1}^{n} A_i\right)^{-1}\left(-L^*u\right) + B^{-1}(u), \tag{7.2}$$

where $A_1, \ldots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$ and $B : \mathcal{G} \rightrightarrows \mathcal{G}$ are maximally monotone operators and $L : \mathcal{H} \to \mathcal{G}$ is a bounded linear operator. Note that in this case (7.2) corresponds to the Attouch–Théra dual problem of (7.1). In the following, we denote the set of solutions of (7.1) and (7.2) by $\mathcal{P}$ and $\mathcal{D}$, respectively, and consider the set $\mathbf{Z}$ defined as

$$\mathbf{Z} := \left\{(x,u) \in \mathcal{H} \times \mathcal{G} : -L^*u \in \sum_{i=1}^{n} A_i(x) \text{ and } u \in B(Lx)\right\},$$

which is useful for tackling primal-dual inclusion problems. It is well-known that $\mathbf{Z}$ is a subset of $\mathcal{P} \times \mathcal{D}$ and that

$$\mathcal{P} \neq \emptyset \Longleftrightarrow \mathbf{Z} \neq \emptyset \Longleftrightarrow \mathcal{D} \neq \emptyset.$$

Indeed, we have

$$\begin{aligned}
\exists\, x \in \mathcal{P} &\Longleftrightarrow (\exists\, x \in \mathcal{H}) \quad 0 \in \sum_{i=1}^{n} A_i(x) + L^*B(Lx) \\
&\Longleftrightarrow (\exists\, (x,u) \in \mathcal{H} \times \mathcal{G}) \begin{cases} -L^*(u) \in \sum_{i=1}^{n} A_i(x), \\ u \in B(Lx), \end{cases} \\
&\Longleftrightarrow (\exists\, (x,u) \in \mathcal{H} \times \mathcal{G}) \begin{cases} x \in \left(\sum_{i=1}^{n} A_i\right)^{-1}(-L^*u), \\ Lx \in B^{-1}(u), \end{cases} \\
&\Longleftrightarrow (\exists\, u \in \mathcal{G}) \quad 0 \in -L\left(\sum_{i=1}^{n} A_i\right)^{-1}\left(-L^*u\right) + B^{-1}(u) \Longleftrightarrow \exists\, u \in \mathcal{D}.
\end{aligned}$$

We refer to an element of $\mathbf{Z}$ as a *primal-dual solution* of (7.1)-(7.2).

7.1.1 A primal-dual fixed point encoding with $(n-1,1)$-fold lifting

Now, we introduce a fixed point algorithm for solving the primal-dual problem given by (7.1)-(7.2). Let $\lambda, \gamma > 0$ and $\mathcal{T}_{\mathcal{PD}} : \mathcal{H}^{n-1} \times \mathcal{G} \to \mathcal{H}^{n-1} \times \mathcal{G}$ be the fixed point operator given by

$$\mathcal{T}_{\mathcal{PD}} \begin{pmatrix} \mathbf{z} \\ v \end{pmatrix} := \begin{pmatrix} \mathbf{z} \\ v \end{pmatrix} + \lambda \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \\ \gamma(y - Lx_n) \end{pmatrix}, \tag{7.3}$$

where $(\mathbf{x}, y) = (x_1, \ldots, x_n, y) \in \mathcal{H}^n \times \mathcal{G}$ depends on $(\mathbf{z}, v) = (z_1, \ldots, z_{n-1}, v) \in \mathcal{H}^{n-1} \times \mathcal{G}$ in the following way

$$\begin{cases} x_1 = J_{A_1}(z_1), \\ x_i = J_{A_i}(z_i + x_{i-1} - z_{i-1}), \quad \forall i \in [\![2, n-1]\!], \\ x_n = J_{A_n}\big(x_1 + x_{n-1} - z_{n-1} - L^*(\gamma L x_1 - v)\big), \\ y = J_{B/\gamma}\left(L(x_1 + x_n) - \dfrac{v}{\gamma}\right). \end{cases} \tag{7.4}$$

In the next lemma we characterize the set of fixed points of the operator $\mathcal{T}_{\mathcal{PD}}$ by means of the set of primal-dual solutions to (7.1)-(7.2).

Lemma 7.1. *Let $n \geq 2$ and $\lambda, \gamma > 0$. The following assertions hold.*

(i) If $(\bar{x}, \bar{u}) \in \mathbf{Z}$, then there exists $\bar{\mathbf{z}} \in \mathcal{H}^{n-1}$ such that $(\bar{\mathbf{z}}, \gamma L\bar{x} - \bar{u}) \in \operatorname{Fix} \mathcal{T}_{\mathcal{PD}}$.

(ii) If $(\bar{z}_1, \ldots, \bar{z}_{n-1}, \bar{v}) \in \operatorname{Fix} \mathcal{T}_{\mathcal{PD}}$, then $(J_{A_1}(\bar{z}_1), \gamma L\bar{x} - \bar{v}) \in \mathbf{Z}$.

As a result,

$$\operatorname{Fix} \mathcal{T}_{\mathcal{PD}} \neq \emptyset \iff \mathbf{Z} \neq \emptyset.$$

Proof. (i) Let $(\bar{x}, \bar{u}) \in \mathbf{Z}$. Then $\bar{u} \in B(L\bar{x})$ and there exists $(a_1, \ldots, a_n) \in \mathcal{H}^n$ such that $a_i \in A_i(\bar{x})$ and $-L^*\bar{u} = \sum_{i=1}^n a_i$. Consider the vectors $(\bar{z}_1, \ldots, \bar{z}_{n-1}, \bar{v}) \in \mathcal{H}^{n-1} \times \mathcal{G}$ defined as

$$\begin{cases} \bar{z}_1 := \bar{x} + a_1 \in (\operatorname{Id} + A_1)(\bar{x}), \\ \bar{z}_i := a_i + \bar{z}_{i-1} \in (\operatorname{Id} + A_i)(\bar{x}) - \bar{x} + \bar{z}_{i-1}, \quad \forall i \in [\![2, n-1]\!], \\ \bar{v} := \gamma L\bar{x} - \bar{u} \in (\gamma \operatorname{Id} - B)(L\bar{x}). \end{cases}$$

Then, we deduce that $\bar{x} = J_{A_1}(\bar{z}_1)$ and $\bar{x} = J_{A_i}(\bar{z}_i + \bar{x} - \bar{z}_{i-1})$ for all $i \in [\![2, n-1]\!]$. Moreover, we have

$$\begin{aligned} 2\bar{x} - \bar{z}_{n-1} - L^*(\gamma L\bar{x} - \bar{v}) &= 2\bar{x} - \bar{z}_{n-1} - L^*(\bar{u}) \\ &= \bar{x} + a_n + \bar{x} - \bar{z}_{n-1} + \sum_{i=1}^{n-1} a_i \\ &= \bar{x} + a_n + \bar{x} - \bar{z}_{n-1} + \sum_{i=2}^{n-1} (\bar{z}_i - \bar{z}_{i-1}) + \bar{z}_1 - \bar{x} \\ &\in (\mathrm{Id} + A_n)(\bar{x}). \end{aligned}$$

Altogether, we obtain

$$\begin{cases} \bar{x} = J_{A_1}(\bar{z}_1), \\ \bar{x} = J_{A_i}(\bar{z}_i + \bar{x} - \bar{z}_{i-1}), \quad \forall i \in [\![2, n-1]\!], \\ \bar{x} = J_{A_n}\big(2\bar{x} - \bar{z}_{n-1} - L^*(\gamma L\bar{x} - \bar{v})\big), \\ L\bar{x} = J_{B/\gamma}\left(2L\bar{x} - \dfrac{\bar{v}}{\gamma}\right), \end{cases}$$

which implies that $(\bar{z}_1, \ldots, \bar{z}_{n-1}, \bar{v}) \in \operatorname{Fix} \mathcal{T}_{\mathcal{PD}}$.

(ii) Let $(\bar{z}_1, \ldots, \bar{z}_{n-1}, \bar{v}) \in \operatorname{Fix} \mathcal{T}_{\mathcal{PD}}$ and set $\bar{x} := J_A(\bar{z}_1)$. By (7.3), $y = L\bar{x}$ and $x_i = \bar{x}$ for all $i \in [\![1, n]\!]$. Consequently, from (7.4) we derive

$$\begin{cases} \bar{z}_1 - \bar{x} \in A_1(\bar{x}), \\ \bar{z}_i - \bar{z}_{i-1} \in A_i(\bar{x}), \quad \forall i \in [\![2, n-1]\!], \\ \bar{x} - \bar{z}_{n-1} - L^*(\gamma L\bar{x} - \bar{v}) \in A_n(\bar{x}), \\ \gamma L\bar{x} - \bar{v} \in B(L\bar{x}). \end{cases}$$

Summing together the first n inclusions above and setting $\bar{u} := \gamma L\bar{x} - \bar{v}$, we deduce

$$\begin{cases} -L^*\bar{u} \in \sum_{i=1}^{n} A_i(\bar{x}), \\ \bar{u} \in B(L\bar{x}), \end{cases}$$

which implies $(\bar{x}, \bar{u}) \in \mathbf{Z}$, as claimed. □

REMARK 7.2 (Primal-dual fixed point encoding). Let $\mathcal{S}_{PD} : \mathcal{H}^{n-1} \times \mathcal{G} \to \mathcal{H} \times \mathcal{G}$ be the solution operator defined as

$$\mathcal{S}_{PD} \begin{pmatrix} \mathbf{z} \\ v \end{pmatrix} := \begin{pmatrix} J_{A_1}(z_1) \\ \gamma L J_{A_1}(z_1) - v \end{pmatrix}.$$

In view of Lemma 7.1, the pair $(\mathcal{T}_{PD}, \mathcal{S}_{PD})$ is a fixed point encoding, which is a frugal primal-dual resolvent splitting with $(n-1, 1)$-fold lifting, for the composite monotone inclusion determined by (7.1)-(7.2).

7.1.2 Convergence analysis

The following technical lemma provides nonexpansive properties of the operator $\mathcal{T}_{\mathcal{PD}}$ in the Hilbert space $\mathcal{H}^{n-1} \times \mathcal{G}$ with scalar product given by

$$\langle (z_1, \ldots, z_{n-1}, v), (\bar{z}_1, \ldots, \bar{z}_{n-1}, \bar{v}) \rangle_\gamma := \sum_{i=1}^{n-1} \langle z_i, \bar{z}_i \rangle_{\mathcal{H}} + \frac{1}{\gamma} \langle v, \bar{v} \rangle_{\mathcal{G}}, \tag{7.5}$$

for $(z_1, \ldots, z_{n-1}, v), (\bar{z}_1, \ldots, \bar{z}_{n-1}, \bar{v}) \in \mathcal{H}^{n-1} \times \mathcal{G}$ and $\gamma > 0$.

Lemma 7.3. *For all $(\mathbf{z}, v) = (z_1, \ldots, z_{n-1}, v) \in \mathcal{H}^{n-1} \times \mathcal{G}$ and $(\bar{\mathbf{z}}, \bar{v}) = (\bar{z}_1, \ldots, \bar{z}_{n-1}, \bar{v}) \in \mathcal{H}^{n-1} \times \mathcal{G}$,*

$$\begin{aligned} &\left\| \mathcal{T}_{\mathcal{PD}}(\mathbf{z}, v) - \mathcal{T}_{\mathcal{PD}}(\bar{\mathbf{z}}, \bar{v}) \right\|_\gamma^2 + \frac{1-\lambda}{\lambda} \left\| \left(\mathrm{Id} - \mathcal{T}_{\mathcal{PD}}\right)(\mathbf{z}, v) - \left(\mathrm{Id} - \mathcal{T}_{\mathcal{PD}}\right)(\bar{\mathbf{z}}, \bar{v}) \right\|_\gamma^2 \\ &\qquad + \frac{1 - \gamma \|L\|^2}{\lambda} \left\| \sum_{i=1}^{n-1} \left(\mathrm{Id} - \mathcal{T}_{\mathcal{PD}}\right)(\mathbf{z}, v)_i - \sum_{i=1}^{n-1} \left(\mathrm{Id} - \mathcal{T}_{\mathcal{PD}}\right)(\bar{\mathbf{z}}, \bar{v})_i \right\|^2 \\ &\qquad \leq \|(\mathbf{z}, v) - (\bar{\mathbf{z}}, \bar{v})\|_\gamma^2, \end{aligned} \tag{7.6}$$

where $\|\cdot\|_\gamma$ denotes the norm induced by the scalar product (7.5). In particular, if $\lambda \in]0, 1[$ and $\gamma \in \left]0, \frac{1}{\|L\|^2}\right]$, the operator $\mathcal{T}_{\mathcal{PD}}$ is λ-averaged nonexpansive.

Proof. Let $(x_1, \ldots, x_n, y) \in \mathcal{H}^n \times \mathcal{G}$ and $(\bar{x}_1, \ldots, \bar{x}_n, \bar{y}) \in \mathcal{H}^n \times \mathcal{G}$ be given by (7.4) from $(\mathbf{z}, v)$ and $(\bar{\mathbf{z}}, \bar{v})$, respectively. For simplicity, we denote $(\mathbf{z}^+, v^+) = \mathcal{T}_{\mathcal{PD}}(\mathbf{z}, v)$ and $(\bar{\mathbf{z}}^+, \bar{v}^+) = \mathcal{T}_{\mathcal{PD}}(\bar{\mathbf{z}}, \bar{v})$. Since $z_1 - x_1 \in A_1(x_1)$ and $\bar{z}_1 - \bar{x}_1 \in A_1(\bar{x}_1)$, by monotonicity of A_1

$$\begin{aligned} 0 &\leq \langle (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1), x_1 - \bar{x}_1 \rangle \\ &= \langle (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1), x_1 - x_2 \rangle + \langle (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1), x_2 - \bar{x}_1 \rangle. \end{aligned} \tag{7.7}$$

For every $i \in [\![2, n-1]\!]$, we have $z_i + x_{i-1} - z_{i-1} - x_i \in A_i(x_i)$ and $\bar{z}_i + \bar{x}_{i-1} - \bar{z}_{i-1} - \bar{x}_i \in A_i(\bar{x}_i)$ and thus, by monotonicity of A_i

$$\begin{aligned}
0 &\leq \langle (z_i + x_{i-1} - z_{i-1} - x_i) - (\bar{z}_i + \bar{x}_{i-1} - \bar{z}_{i-1} - \bar{x}_i), x_i - \bar{x}_i \rangle \\
&= \langle (z_i - x_i) - (\bar{z}_i - \bar{x}_i), x_i - \bar{x}_i \rangle - \langle (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x}_{i-1}), x_i - \bar{x}_i \rangle \\
&= \langle (z_i - x_i) - (\bar{z}_i - \bar{x}_i), x_i - x_{i+1} \rangle + \langle (z_i - x_i) - (\bar{z}_i - \bar{x}_i), x_{i+1} - \bar{x}_i \rangle \\
&\quad - \langle (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x}_{i-1}), x_i - \bar{x}_{i-1} \rangle \\
&\quad - \langle (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x}_{i-1}), \bar{x}_{i-1} - \bar{x}_i \rangle.
\end{aligned} \tag{7.8}$$

Now, since $x_1 + x_{n-1} - z_{n-1} - x_n - L^* (\gamma L x_1 - v) \in A_n(x_n)$ and $\bar{x}_1 + \bar{x}_{n-1} - \bar{z}_{n-1} - \bar{x}_n - L^* (\gamma L \bar{x}_1 - \bar{v}) \in A_n(\bar{x}_n)$, again monotonicity of A_n results in the inequality

$$\begin{aligned}
0 &\leq \langle x_1 + x_{n-1} - z_{n-1} - x_n - L^* (\gamma L x_1 - v), x_n - \bar{x}_n \rangle \\
&\quad - \langle \bar{x}_1 + \bar{x}_{n-1} - \bar{z}_{n-1} - \bar{x}_n - L^* (\gamma L \bar{x}_1 - \bar{v}), x_n - \bar{x}_n \rangle \\
&= \langle (x_{n-1} - z_{n-1}) - (\bar{x}_{n-1} - \bar{z}_{n-1}), x_n - \bar{x}_n \rangle + \langle (x_1 - \bar{x}_1) - (x_n - \bar{x}_n), x_n - \bar{x}_n \rangle \\
&\quad - \langle \gamma (L x_1 - L \bar{x}_1) - (v - \bar{v}), L x_n - L \bar{x}_n \rangle \\
&= \langle (x_{n-1} - z_{n-1}) - (\bar{x}_{n-1} - \bar{z}_{n-1}), x_n - \bar{x}_{n-1} \rangle \\
&\quad + \langle (x_1 - \bar{x}_1) - (x_n - \bar{x}_n), x_n - \bar{x}_n \rangle \\
&\quad + \langle (x_{n-1} - z_{n-1}) - (\bar{x}_{n-1} - \bar{z}_{n-1}), \bar{x}_{n-1} - \bar{x}_n \rangle \\
&\quad - \langle \gamma (L x_1 - L \bar{x}_1) - (v - \bar{v}), L x_n - L \bar{x}_n \rangle.
\end{aligned} \tag{7.9}$$

Finally, we have $\gamma L(x_1 + x_n) - v - \gamma y \in B(y)$ and $\gamma L(\bar{x}_1 + \bar{x}_n) - \bar{v} - \gamma \bar{y} \in B(\bar{y})$, so by monotonicity of B, we get

$$0 \leq \langle (\gamma L(x_1 + x_n) - v - \gamma\, y) - (\gamma L(\bar{x}_1 + \bar{x}_n) - \bar{v} - \gamma\, \bar{y}), y - \bar{y} \rangle. \tag{7.10}$$

Summing together (7.7)-(7.10) and rearranging, yields

$$\begin{aligned}
0 &\leq \sum_{i=1}^{n-1} \langle (x_i - x_{i+1}) - (\bar{x}_i - \bar{x}_{i+1}), z_i - \bar{z}_i \rangle \\
&\quad + \sum_{i=1}^{n-1} \langle (x_i - \bar{x}_i) - (x_{i+1} - \bar{x}_{i+1}), \bar{x}_i - x_i \rangle \\
&\quad + \langle (x_1 - \bar{x}_1) - (x_n - \bar{x}_n), x_n - \bar{x}_n \rangle + \langle (L x_n - L \bar{x}_n) - (y - \bar{y}), v - \bar{v} \rangle \\
&\quad + \gamma \langle \big(L(x_1 + x_n) - L(\bar{x}_1 + \bar{x}_n)\big) - (y - \bar{y}), y - \bar{y} \rangle \\
&\quad - \gamma \langle L x_1 - L \bar{x}_1, L x_n - L \bar{x}_n \rangle.
\end{aligned} \tag{7.11}$$

The sums in (7.11) can be written, respectively, as

$$\begin{aligned}
&\sum_{i=1}^{n-1}\langle (x_i - x_{i+1}) - (\bar{x}_i - \bar{x}_{i+1}), z_i - \bar{z}_i\rangle \\
&\quad = \frac{1}{\lambda}\sum_{i=1}^{n-1}\langle (z_i - z_i^+) - (\bar{z}_i - \bar{z}_i^+), z_i - \bar{z}_i\rangle \\
&\quad = \frac{1}{\lambda}\langle (\mathbf{z} - \mathbf{z}^+) - (\bar{\mathbf{z}} - \bar{\mathbf{z}}^+), \mathbf{z} - \bar{\mathbf{z}}\rangle \\
&\quad = \frac{1}{2\lambda}\left(\|(\mathbf{z} - \mathbf{z}^+) - (\bar{\mathbf{z}} - \bar{\mathbf{z}}^+)\|^2 - \|\mathbf{z}^+ - \bar{\mathbf{z}}^+\|^2 + \|\mathbf{z} - \bar{\mathbf{z}}\|^2\right),
\end{aligned} \tag{7.12}$$

and

$$\begin{aligned}
&\sum_{i=1}^{n-1}\langle (x_i - \bar{x}_i) - (x_{i+1} - \bar{x}_{i+1}), \bar{x}_i - x_i\rangle \\
&\quad = \frac{1}{2}\sum_{i=1}^{n-1}\left(\|x_{i+1} - \bar{x}_{i+1}\|^2 - \|x_i - \bar{x}_i\|^2 - \|(x_i - x_{i+1}) - (\bar{x}_i - \bar{x}_{i+1})\|^2\right) \\
&\quad = \frac{1}{2}\left(\|x_n - \bar{x}_n\|^2 - \|x_1 - \bar{x}_1\|^2 - \frac{1}{\lambda^2}\sum_{i=1}^{n-1}\|(z_i - z_i^+) - (\bar{z}_i - \bar{z}_i^+)\|^2\right) \\
&\quad = \frac{1}{2}\left(\|x_n - \bar{x}_n\|^2 - \|x_1 - \bar{x}_1\|^2 - \frac{1}{\lambda^2}\|(\mathbf{z} - \mathbf{z}^+) - (\bar{\mathbf{z}} - \bar{\mathbf{z}}^+)\|^2\right).
\end{aligned} \tag{7.13}$$

The third term in (7.11), becomes

$$\begin{aligned}
&\langle (x_1 - \bar{x}_1) - (x_n - \bar{x}_n), x_n - \bar{x}_n\rangle \\
&\qquad = \frac{1}{2}\left(\|x_1 - \bar{x}_1\|^2 - \|x_n - \bar{x}_n\|^2 - \|(x_1 - \bar{x}_1) - (x_n - \bar{x}_n)\|^2\right),
\end{aligned} \tag{7.14}$$

while the fourth term yields

$$\begin{aligned}
&\langle (Lx_n - L\bar{x}_n) - (y - \bar{y}), v - \bar{v}\rangle \\
&\qquad = \frac{1}{\gamma\lambda}\langle (v - v^+) - (\bar{v} - \bar{v}^+), v - \bar{v}\rangle \\
&\qquad = \frac{1}{2\gamma\lambda}\left(\|(v - v^+) - (\bar{v} - \bar{v}^+)\|^2 - \|v^+ - \bar{v}^+\|^2 + \|v - \bar{v}\|^2\right).
\end{aligned} \tag{7.15}$$

Lastly, making use of the Cauchy–Schwarz and Young's inequalities, the second last term of (7.11) gives

$$\begin{aligned}
&\gamma\langle\big(L(x_1+x_n)-L(\bar{x}_1+\bar{x}_n)\big)-(y-\bar{y}),y-\bar{y}\rangle\\
&\quad=\gamma\left(\langle Lx_1-L\bar{x}_1,y-\bar{y}\rangle+\langle(Lx_n-L\bar{x}_n)-(y-\bar{y}),y-\bar{y}\rangle\right)\\
&\quad=\frac{\gamma}{2}\left(\|Lx_n-L\bar{x}_n\|^2-\|\left(Lx_n-L\bar{x}_n\right)-(y-\bar{y})\|^2-\|y-\bar{y}\|^2\right)\\
&\qquad+\gamma\langle Lx_1-L\bar{x}_1,y-\bar{y}\rangle\\
&\quad\leq\frac{\gamma}{2}\left(\|Lx_n-L\bar{x}_n\|^2-\frac{1}{\gamma^2\lambda^2}\|(v-v^+)-(\bar{v}-\bar{v}^+)\|^2-\|y-\bar{y}\|^2\right)\\
&\qquad+\frac{\gamma}{2}\|Lx_1-L\bar{x}_1\|^2+\frac{\gamma}{2}\|y-\bar{y}\|^2\\
&\quad=\frac{\gamma}{2}\|Lx_1-L\bar{x}_1\|^2+\frac{\gamma}{2}\|Lx_n-L\bar{x}_n\|^2-\frac{1}{2\gamma\lambda^2}\|(v-v^+)-(\bar{v}-\bar{v}^+)\|^2,
\end{aligned}\tag{7.16}$$

while the last term can be rearranged as follows

$$\begin{aligned}
&-\gamma\langle Lx_1-L\bar{x}_1,Lx_n-L\bar{x}_n\rangle\\
&\quad=\frac{\gamma}{2}\left(\|L(x_1-x_n)-L(\bar{x}_1-\bar{x}_n)\|^2-\|Lx_1-L\bar{x}_1\|^2-\|Lx_n-L\bar{x}_n\|^2\right).
\end{aligned}\tag{7.17}$$

Summing together (7.16) and (7.17) and using the Lipschitz continuity of L, we get

$$\begin{aligned}
&\gamma\langle\big(L(x_1+x_n)-L(\bar{x}_1+\bar{x}_n)\big)-(y-\bar{y}),y-\bar{y}\rangle-\gamma\langle Lx_1-L\bar{x}_1,Lx_n-L\bar{x}_n\rangle\\
&\quad=\frac{\gamma}{2}\|L(x_1-x_n)-L(\bar{x}_1-\bar{x}_n)\|^2-\frac{1}{2\gamma\lambda^2}\|(v-v^+)-(\bar{v}-\bar{v}^+)\|^2\\
&\quad\leq\frac{\gamma\|L\|^2}{2}\|(x_1-x_n)-(\bar{x}_1-\bar{x}_n)\|^2-\frac{1}{2\gamma\lambda^2}\|(v-v^+)-(\bar{v}-\bar{v}^+)\|^2.
\end{aligned}\tag{7.18}$$

Multiplying (7.11) by 2λ and substituting equations (7.12)-(7.18), we obtain the final inequality

$$\begin{aligned}
&\|\mathbf{z}^+-\bar{\mathbf{z}}^+\|^2+\left(\frac{1}{\lambda}-1\right)\left(\|(\mathbf{z}-\mathbf{z}^+)-(\bar{\mathbf{z}}-\bar{\mathbf{z}}^+)\|^2+\frac{1}{\gamma}\|(v-v^+)-(\bar{v}-\bar{v}^+)\|^2\right)\\
&+\frac{1}{\gamma}\|v^+-\bar{v}^+\|^2+\lambda\left(1-\gamma\|L\|^2\right)\|(x_1-x_n)-(\bar{x}_1-\bar{x}_n)\|^2\\
&\quad\leq\|\mathbf{z}-\bar{\mathbf{z}}\|^2+\frac{1}{\gamma}\|v-\bar{v}\|^2.
\end{aligned}$$

To complete the proof, just note that

$$\begin{aligned}\lambda(x_1 - x_n) - \lambda(\bar{x}_1 - \bar{x}_n) &= \lambda \sum_{i=1}^{n-1}(x_i - x_{i+1}) - \lambda \sum_{i=1}^{n-1}(\bar{x}_i - \bar{x}_{i+1}) \\ &= \sum_{i=1}^{n-1}(z_i - z_i^+) - \sum_{i=1}^{n-1}(\bar{z}_i - \bar{z}_i^+),\end{aligned}$$

from where (7.6) finally follows. □

Next, we state the main result of this section, which establishes the convergence of the iterative algorithm defined by the operator $\mathcal{T}_{\mathcal{PD}}$.

Theorem 7.4. *Let $n \geq 2$, let $L : \mathcal{H} \to \mathcal{G}$ be a bounded linear operator and assume $A_1, \ldots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$ and $B : \mathcal{G} \rightrightarrows \mathcal{G}$ are maximally monotone operators verifying $\operatorname{zer}\left(\sum_{i=1}^n A_i + L^*BL\right) \neq \emptyset$. Further, let $\lambda \in]0,1[$ and $\gamma \in \left]0, \frac{1}{\|L\|^2}\right]$. Given an initial point $(\mathbf{z}^0, v^0) = (z_1^0, \ldots, z_{n-1}^0, v^0) \in \mathcal{H}^{n-1} \times \mathcal{G}$, consider the sequences defined by*

$$\begin{pmatrix} \mathbf{z}^{k+1} \\ v^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{z}^{k} \\ v^{k} \end{pmatrix} + \lambda \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \\ \gamma(y^k - Lx_n^k) \end{pmatrix}, \quad \forall k \in \mathbb{N}, \tag{7.19}$$

with

$$\begin{cases} x_1^k = J_{A_1}(z_1^k), \\ x_i^k = J_{A_i}(z_i^k + x_{i-1}^k - z_{i-1}^k), \quad \forall i \in [\![2, n-1]\!], \\ x_n^k = J_{A_n}\big(x_1^k + x_{n-1}^k - z_{n-1}^k - L^*(\gamma L x_1^k - v^k)\big), \\ y^k = J_{B/\gamma}\left(L(x_1^k + x_n^k) - \dfrac{v^k}{\gamma}\right). \end{cases} \tag{7.20}$$

Then the following statements hold.

(i) The sequence $(\mathbf{z}^k, v^k)_{k\in\mathbb{N}}$ converges weakly to a point $(\bar{\mathbf{z}}, \bar{v}) \in \operatorname{Fix} \mathcal{T}_{\mathcal{PD}}$.

(ii) The sequence $(x_1^k, \ldots, x_n^k, y^k)_{k\in\mathbb{N}}$ converges weakly to $(\bar{x}, \ldots, \bar{x}, L\bar{x})$ with $\bar{x} \in \mathcal{P}$.

(iii) The sequence $\left(\gamma L x_i^k - v^k\right)_{k\in\mathbb{N}}$ converges weakly to $\gamma L\bar{x} - \bar{v} \in \mathcal{D}$, for all $i \in [\![1, n]\!]$.

Proof. (i) The sequence in (7.19) is the fixed point iteration generated as

$$\begin{pmatrix} \mathbf{z}^{k+1} \\ v^{k+1} \end{pmatrix} = \mathcal{T}_{\mathcal{PD}} \begin{pmatrix} \mathbf{z}^{k} \\ v^{k} \end{pmatrix}, \quad \forall k \in \mathbb{N}.$$

Since $\lambda \in]0,1[$ and $\gamma \in]0, \|L\|^{-2}]$, $\mathcal{T}_{\mathcal{PD}}$ is averaged nonexpansive by Lemma 7.3 and, moreover, $\operatorname{Fix} \mathcal{T}_{\mathcal{PD}} = \emptyset$, due to $\mathbf{Z} \neq \emptyset$ and Lemma 7.1(i). Then, by Theorem 2.7, we conclude that the sequence $(\mathbf{z}^k, v^k)_{k\in\mathbb{N}}$ converges weakly to a point $(\bar{\mathbf{z}}, \bar{v}) \in \operatorname{Fix} \mathcal{T}_{\mathcal{PD}}$ and $\lim_{k\to\infty} \|(\mathbf{z}^{k+1}, v^{k+1}) - (\mathbf{z}^k, v^k)\|_\gamma = 0$.

(ii) Due to the weak convergence proved in (i), the sequence $(\mathbf{z}^k, v^k)_{k\in\mathbb{N}}$ is bounded. Then, nonexpansiveness of the resolvents and boundedness of the linear operator L imply that the sequence $(\mathbf{x}^k, y^k)_{k\in\mathbb{N}} = (x_1^k, \ldots, x_n^k, y^k)_{k\in\mathbb{N}}$ is also bounded. Further, the fact that $(\mathbf{z}^{k+1}, v^{k+1})_{k\in\mathbb{N}} - (\mathbf{z}^k, v^k)_{k\in\mathbb{N}} \to 0$, as $k \to \infty$, implies by (7.19) that

$$y^k - Lx_n^k \to 0 \text{ and } x_{i+1}^k - x_i^k \to 0, \text{ for all } i \in [\![1, n-2]\!]. \tag{7.21}$$

Next, by making use of the definition of resolvents and (7.20), we can write

$$C \begin{pmatrix} z_1^k - x_1^k \\ (z_2^k - x_2^k) - (z_1^k - x_1^k) \\ \vdots \\ (z_{n-1}^k - x_{n-1}^k) - (z_{n-2}^k - x_{n-2}^k) \\ x_n^k \\ \gamma\left(L(x_1^k + x_n^k) - y^k\right) - v^k \end{pmatrix} \ni \begin{pmatrix} x_1^k - x_n^k \\ x_2^k - x_n^k \\ \vdots \\ x_{n-1}^k - x_n^k \\ x_1^k - x_n^k + \gamma L^*\left(Lx_n^k - y^k\right) \\ y^k - Lx_n^k \end{pmatrix}, \tag{7.22}$$

where the operator $C\colon \mathcal{H}^n \times \mathcal{G} \rightrightarrows \mathcal{H}^n \times \mathcal{G}$ is given by

$$C := \begin{pmatrix} A_1^{-1} \\ A_2^{-1} \\ \vdots \\ A_{n-1}^{-1} \\ A_n \\ B^{-1} \end{pmatrix} + \begin{pmatrix} 0 & 0 & \ldots & 0 & -\mathrm{Id} & 0 \\ 0 & 0 & \ldots & 0 & -\mathrm{Id} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & -\mathrm{Id} & 0 \\ \mathrm{Id} & \mathrm{Id} & \ldots & \mathrm{Id} & 0 & L^* \\ 0 & 0 & \ldots & 0 & -L & 0 \end{pmatrix}. \tag{7.23}$$

The operator C is maximally monotone as the sum of a maximally monotone operator and a skew symmetric linear operator (see Example 2.13 and Proposition 2.15). By Proposition 2.23, the graph of C is sequentially closed in the weak-strong topology, by demiclosedness of maximally monotone operators.

Now, let $(\bar{\mathbf{x}}, \bar{y})$ be a weak sequential cluster point of $(\mathbf{x}^k, y^k)_{k\in\mathbb{N}}$. Due to (7.21), $\bar{\mathbf{x}}$ is of the form $\bar{\mathbf{x}} = (\bar{x}, \ldots, \bar{x}) \in \mathcal{H}^n$ and $\bar{y} = L\bar{x}$. Taking the limit along a subsequence of $(\mathbf{x}^k, y^k)_{k\in\mathbb{N}}$ which converges weakly to $(\bar{\mathbf{x}}, \bar{y})$ and using demiclosedness of C, equations (7.22) and (7.23) yield the expression

$$\begin{cases} \bar{z}_1 - \bar{x} \in A_1(\bar{x}), \\ \bar{z}_i - \bar{z}_{i-1} \in A_i(\bar{x}), \quad \forall i \in [\![2, n-1]\!], \\ \bar{x} - \bar{z}_{n-1} - L^*(\gamma L\bar{x} - \bar{v}) \in A_n(\bar{x}), \\ \gamma L\bar{x} - \bar{v} \in B(L\bar{x}), \end{cases}$$

which, by summing the first n equations, implies that $(\bar{x}, \gamma L\bar{x} - \bar{v}) \in \mathbf{Z}$ with $\bar{x} = J_{A_1}(\bar{z}_1)$. In particular, we have shown that $(\bar{\mathbf{x}}, \bar{y})$ is directly obtained from $\bar{\mathbf{z}}$, implying that it is the unique weak sequential cluster point of the bounded sequence $(\mathbf{x}^k, y^k)_{k\in\mathbb{N}}$. Thus, the full sequence converges weakly to this point.

(iii) From (i)-(ii), for all $i \in [\![1, n]\!]$, we deduce that the sequence $(\gamma L x_i^k - v^k)_{k\in\mathbb{N}}$ weakly converges to $\gamma L\bar{x} - \bar{v}$, which belongs to $\mathcal{D}$ since $(\bar{x}, \gamma L\bar{x} - \bar{v}) \in \mathbf{Z}$. □

Remark 7.5 (Malitsky–Tam resolvent splitting [173] as a special case). Consider Problem (7.1)-(7.2) in the particular case in which $L = \mathrm{Id}$. Then, $B : \mathcal{H} \rightrightarrows \mathcal{H}$ and equation (7.1) becomes the classical monotone inclusion problem with $(n+1)$-operators. Furthermore, by setting $\gamma = 1$ in Theorem 7.4, it is straightforward to see that the sequences in (7.19)-(7.20) yield the Malitsky–Tam resolvent splitting with minimal lifting for $(n+1)$-operators.

Remark 7.6. In the case in which $n = m = 1$, making use of Proposition 2.20 and considering the change of variable $u^k := \gamma L z^k - v^k$, for all $k \in \mathbb{N}$, the iteration in (7.19) takes the form

$$\begin{cases} x^k = J_A(z^k - L^*u^k), \\ w^k = J_{\gamma B^{-1}}\big(u^k + \gamma L x^k\big), \\ z^{k+1} = z^k + \lambda(x^k - z^k), \\ u^{k+1} = u^k - \gamma L(z^{k+1} - z^k) + \lambda(w^k - u^k). \end{cases} \tag{7.24}$$

The scheme (7.24) resembles that of the Chambolle–Pock algorithm, but does not recover it, as the update of the dual variable differs. It is not clear to us whether Chambolle–Pock splitting can be obtained as a particular case of (7.19)-(7.20).

Remark 7.7 (On the parameter γ in the definition of the norm $\|\cdot\|_\gamma$). In Lemma 7.3, we proved that the operator T is λ-averaged nonexpansive with respect to the norm $\|\cdot\|_\gamma$ induced by the scalar product defined in (7.5). Although the use of this norm did not require

detours from the usual procedure to prove convergence of the fixed point iteration in Theorem 7.4, it may numerically affect the performance of the algorithm. To give an intuition about this, consider the norm of the *sequence of residuals* $\left(\|(\mathbf{z}^{k+1}, v^{k+1}) - (\mathbf{z}^k, v^k)\|_\gamma\right)_{k\in\mathbb{N}}$, which converges to 0 as the algorithm reaches a fixed point, and note that we have

$$\left\|(\mathbf{z}^{k+1}, v^{k+1}) - (\mathbf{z}^k, v^k)\right\|_\gamma^2 = \|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2 + \frac{1}{\gamma}\|v^{k+1} - v^k\|^2, \quad \forall k \in \mathbb{N}.$$

Lemma 7.3 implies that this sequence is monotone decreasing, but if γ is very small, the weight of the sequence of dual variables $(v^{k+1} - v^k)_{k\in\mathbb{N}}$ in the norm would be much larger than the one of the sequence of primal variables $(\mathbf{z}^{k+1} - \mathbf{z}^k)_{k\in\mathbb{N}}$, so a small decrease in the value of $\|v^{k+1} - v^k\|$ will readily imply a decrease of the norm of the sequence of residuals even if $\|\mathbf{z}^{k+1} - \mathbf{z}^k\|$ does not diminish much. Because of that, a larger number of iterations might be needed to achieve convergence of the primal sequence, which can slow down the overall convergence of the algorithm. Nonetheless, it is possible to perform some sort of preconditioning to prevent from having a large constant in the definition of the norm. We will further comment on this in the numerical experiment in Section 7.4.

7.2 Extension to multiple linear compositions

A standard product space reformulation permits to extend our method to the more general inclusion Problem 3.12, which has finitely many linearly composed maximally monotone operators. We detail this in Corollary 7.9, while the resulting scheme is displayed in Algorithm 7.8.

Corollary 7.9. *Let $n \geq 2$ and assume that Problem 3.12 has a solution. Let $\lambda \in]0,1[$ and $\gamma \in \left]0, 1/\sum_{j=1}^m \|L_j\|^2\right]$. Given some initial points $\mathbf{z}^0 = (z_1, \ldots, z_{n-1}) \in \mathcal{H}^{n-1}$ and $\mathbf{v}^0 = (v_1^0, \ldots, v_m^0) \in \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$, consider the sequences $(\mathbf{z}^k, \mathbf{v}^k)_{k\in\mathbb{N}}$ and $(\mathbf{x}^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ generated by Algorithm 7.8. Then the following assertions hold.*

(i) The sequence $(\mathbf{z}^k, \mathbf{v}^k)_{k\in\mathbb{N}}$ converges weakly to a point $(\bar{\mathbf{z}}, \bar{\mathbf{v}}) \in \mathcal{H}^{n-1} \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$.

(ii) The sequence $(x_1^k, \ldots, x_n^k, y_1^k, \ldots, y_m^k)_{k\in\mathbb{N}}$ converges weakly to $(\bar{x}, \ldots, \bar{x}, L_1\bar{x}, \ldots, L_m\bar{x})$ with $\bar{x} \in \mathcal{H}$ solving the primal inclusion (3.23).

(iii) For all $i \in [\![1, n]\!]$, the sequence $(\gamma L_1 x_i^k - v_1^k, \ldots, \gamma L_m x_i^k - v_m^k)_{k\in\mathbb{N}}$ converges weakly to $(\gamma L_1\bar{x} - \bar{v}_1, \ldots, \gamma L_m\bar{x} - \bar{v}_m)$, which solves the dual inclusion (3.24).

Algorithm 7.8 Primal-dual splitting for Problem 3.12 with minimal lifting.

Require: $\lambda \in]0,1[$ and $\gamma \in \left]0, 1/\sum_{j=1}^{m} \|L_j\|^2\right[$.

1: Choose $\mathbf{z}^0 = (z_1^0, \dots, z_{n-1}^0) \in \mathcal{H}^{n-1}$ and $\mathbf{v}^0 = (v_1^0, \dots, v_m^0) \in \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$.

2: **for** $k = 0, 1, \dots$ **do**

3: Compute

$$\begin{pmatrix} \mathbf{z}^{k+1} \\ \mathbf{v}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{z}^{k} \\ \mathbf{v}^{k} \end{pmatrix} + \lambda \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \\ \gamma(y_1^k - L_1 x_n^k) \\ \vdots \\ \gamma(y_m^k - L_m x_n^k) \end{pmatrix}, \tag{7.25}$$

with $\mathbf{x}^k = (x_1^k, \dots, x_n^k) \in \mathcal{H}^n$ and $\mathbf{y}^k = (y_1^k, \dots, y_m^k) \in \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$ computed as

$$\begin{cases} x_1^k = J_{A_1}(z_1^k), \\ x_i^k = J_{A_i}(z_i^k + x_{i-1}^k - z_{i-1}^k), \quad \forall i \in [\![2, n-1]\!], \\ x_n^k = J_{A_n}\left(x_1^k + x_{n-1}^k - z_{n-1}^k - \sum_{j=1}^{m} L_j^*(\gamma L_j x_1^k - v_j^k)\right), \\ y_j^k = J_{B_j/\gamma}\left(L_j(x_1^k + x_n^k) - \dfrac{v_j^k}{\gamma}\right), \quad \forall j \in [\![1, m]\!]. \end{cases} \tag{7.26}$$

4: **end for**

Proof. Just note that Problem 3.12 can be reformulated as an instance of Problem (7.1)-(7.2) by replacing B by the operator $\mathbf{B} : \mathcal{G}_1 \times \cdots \times \mathcal{G}_m \rightrightarrows \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$ defined as the Cartesian product $\mathbf{B} := \bigtimes_{j=1}^{m} B_j$ and L by $\mathbf{L} : \mathcal{H} \to \mathcal{G}_1 \times \cdots \times \mathcal{G}_m : x \mapsto (L_j x)_{j=1}^m$. In particular, it holds that $\|\mathbf{L}\|^2 = \sum_{j=1}^{n} \|L_j\|^2$ and that its linear adjoint operator is $\mathbf{L}^* : \mathcal{G}_1 \times \cdots \times \mathcal{G}_m \to \mathcal{H} : (v_1, \dots, v_m) \mapsto \sum_{j=1}^{m} L_j^* v_j$. Hence, the result follows by considering the averaged nonexpansive operator $\mathcal{T}_{\mathcal{PD}}$ in (7.3) for this choice of operators and applying Theorem 7.4. □

7.2.1 Discussion on minimal lifting

The fixed point encoding for the case $m = 1$ in Remark 7.2 can easily be extended to the more general Problem 3.12 with $m \geq 1$. For this, let $n \geq 2$ in Problem 3.12. Let $\mathcal{T}_{PD} : \mathcal{H}^{n-1} \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m \to \mathcal{H}^{n-1} \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$ be the operator defined in (7.3) by setting $B := \bigtimes_{j=1}^{m} B_j$ and letting L be the operator $\mathbf{L}$ described in the proof of Corollary 7.9.

Define the solution operator $\mathcal{S}_{PD} : \mathcal{H}^{n-1} \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m \to \mathcal{H} \times \mathcal{G}_1 \times \cdots \times \mathcal{G}_m$ as

$$\mathcal{S}_{PD}\begin{pmatrix} \mathbf{z} \\ \mathbf{v} \end{pmatrix} := \begin{pmatrix} J_{A_1}(z_1) \\ \gamma L_1 J_{A_1}(z_1) - v_1 \\ \vdots \\ \gamma L_m J_{A_1}(z_1) - v_m \end{pmatrix}.$$

Then, by Lemma 7.1 and Corollary 7.9, the pair $(\mathcal{T}_{PD}, \mathcal{S}_{PD})$ is a frugal resolvent splitting with $(n-1, m)$-fold lifting.

As a consequence, it follows from Theorem 4.21 that the lifting of Algorithm 7.8 is minimal among frugal primal-dual resolvent splitting algorithms with m dual variables. To the best of the author's knowledge, Algorithm 7.8 is the only primal-dual method with minimal lifting devised so far.

Finally, we conclude this section by highlighting that Algorithm 7.8 can be applied with $n < 2$, by setting $A_i = 0$ if required. However, a reduction in the lifting is not obtained in this case.

Remark 7.10 (Algorithm 7.8 when $n \leq 1$). Consider Algorithm 7.8 applied to Problem 3.12 with $n \leq 1$. We distinguish the two cases:

(i) If $n = 1$, then Algorithm 7.8 has $(1, m)$-fold lifting. Indeed, equations (7.25) and (7.26) become

$$\begin{pmatrix} z^{k+1} \\ \mathbf{v}^{k+1} \end{pmatrix} = \begin{pmatrix} z^k \\ \mathbf{v}^k \end{pmatrix} + \lambda \begin{pmatrix} x^k - z^k \\ \gamma(y_1^k - L_1 x^k) \\ \vdots \\ \gamma(y_m^k - L_m x^k) \end{pmatrix}, \quad \forall k \in \mathbb{N},$$

and

$$\begin{cases} x^k = J_A\Big(z^k - \sum_{j=1}^m L_j^*(\gamma L_j z^k - v_j^k)\Big), \\ y_j^k = J_{B_j/\gamma}\Big(L_j(z^k + x^k) - \dfrac{v_j^k}{\gamma}\Big), \quad \forall j \in [\![1, m]\!], \end{cases} \tag{7.27}$$

respectively. This means that, in contrast with what happens when $n \geq 2$, there is no reduction in the lifting with respect to the number of operators involved.

(ii) If $n = 0$, the scheme also has $(1, m)$-fold lifting. In fact, the scheme is the same as in the previous case but substituting J_A by Id in (7.27). Note that this is also the lifting obtained by the already known algorithms in the literature applied to this case (see, e.g., [59, 75, 116, 239]).

7.3 A unified splitting for mixtures of monotone inclusions

Many methods developed in the literature are suited for solving monotone inclusions resulting from a mixture of Problems 3.9 and 3.12 (see, e.g., [59, 65, 109, 239]). We now illustrate how to combine the algorithms in Chapter 6 and Section 7.1 to achieve the same goal.

Let $A_1, \ldots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$ and $B : \mathcal{G} \rightrightarrows \mathcal{G}$ be maximally monotone operators, let $T_1, \ldots, T_{n-1} : \mathcal{H} \to \mathcal{H}$ be $\frac{1}{\beta}$-cocoercive operators, with $\beta > 0$, and $L : \mathcal{H} \to \mathcal{G}$ be a bounded linear operator. We consider the primal inclusion

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^{n} A_i(x) + \sum_{i=1}^{n-1} T_i(x) + L^*B(Lx), \tag{7.28}$$

together with its associated dual inclusion

$$\text{find } u \in \mathcal{G} \text{ such that } 0 \in -L\left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i\right)^{-1}(-L^*u) + B^{-1}(u). \tag{7.29}$$

It is not difficult to check that both inclusions are equivalent in the same way as (7.1)-(7.2). For simplicity, we are limiting the number of linearly composed operators in (7.28)-(7.29) to one. The case with an arbitrary finite number of linear compositions can be addressed by resorting to the same trick as in Section 7.2.1. In order to tackle (7.28)-(7.29), we propose the following algorithm which merges the iterations in (6.6) and (7.8).

Algorithm 7.11 Splitting for (7.28)-(7.29) with $(n-1, 1)$-fold lifting.

Require: $\delta \in \left]0, 2/\beta\right[$, $\lambda \in \left]0, 1 - \frac{\delta\beta}{2}\right[$ and $\gamma \in \left]0, \frac{1}{\|L\|^2}\right]$.

1: Choose $\mathbf{z}^0 = (z_1^0, \ldots, z_{n-1}^0) \in \mathcal{H}^{n-1}$ and $v^0 \in \mathcal{G}$.

2: **for** $k = 0, 1, \ldots$ **do**

3: Compute

$$\begin{pmatrix} \mathbf{z}^{k+1} \\ v^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{z}^{k} \\ v^{k} \end{pmatrix} + \lambda \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \\ \gamma(y^k - Lx_n^k) \end{pmatrix}, \tag{7.30}$$

with $\mathbf{x}^k = (x_1^k, \ldots, x_n^k) \in \mathcal{H}^n$ and $y^k \in \mathcal{G}$ computed as

$$\begin{cases} x_1^k = J_{\delta A_1}(z_1^k), \\ x_i^k = J_{\delta A_i}\big(z_i^k + x_{i-1}^k - z_{i-1}^k - \delta T_{i-1}(x_{i-1}^k)\big), \quad \forall i \in [\![2, n-1]\!], \\ x_n^k = J_{\delta A_n}\big(x_1^k + x_{n-1}^k - z_{n-1}^k - \delta T_{n-1}(x_{n-1}^k) - L^*(\gamma L x_1^k - v^k)\big), \\ y^k = J_{\frac{\delta}{\gamma}B}\left(L(x_1^k + x_n^k) - \dfrac{v^k}{\gamma}\right). \end{cases} \tag{7.31}$$

4: **end for**

The sequence $(\mathbf{z}^k, v^k)_{k\in\mathbb{N}}$ in (7.30) is obtained as the fixed point iteration of the operator $\widetilde{\mathcal{T}} : \mathcal{H}^{n-1} \times \mathcal{G} \to \mathcal{H}^{n-1} \times \mathcal{G}$ given by

$$\widetilde{\mathcal{T}}\begin{pmatrix}\mathbf{z}\\ v\end{pmatrix} := \begin{pmatrix}\mathbf{z}\\ v\end{pmatrix} + \lambda \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \\ \gamma(y - Lx_n)\end{pmatrix}, \tag{7.32}$$

with

$$\begin{cases} x_1 = J_{\delta A_1}(z_1), \\ x_i = J_{\delta A_i}\big(z_i + x_{i-1} - z_{i-1} - \delta T_{i-1}(x_{i-1})\big), \quad \forall i \in [\![2, n-1]\!], \\ x_n = J_{\delta A_n}\big(x_1 + x_{n-1} - z_{n-1} - \delta T_{n-1}(x_{n-1}) - L^*(\gamma L x_1 - v)\big), \\ y = J_{\frac{\delta}{\gamma}B}\left(L(x_1 + x_n) - \dfrac{v}{\gamma}\right). \end{cases} \tag{7.33}$$

This operator will be used in the convergence theorem of Algorithm 7.11 that comes next.

Theorem 7.12. *Let $n \geq 2$. Let $A_1, \ldots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$ and $B : \mathcal{G} \rightrightarrows \mathcal{G}$ be maximally monotone operators, $T_1, \ldots, T_{n-1} : \mathcal{H} \to \mathcal{H}$ be $\frac{1}{\beta}$-cocoercive operators and $L : \mathcal{H} \to \mathcal{G}$ be a bounded linear operator. Assume that $\operatorname{zer}\big(\sum_{i=1}^n A_i + \sum_{i=1}^{n-1} T_i + L^*BL\big) \neq \emptyset$. Further, set $\delta \in \left]0, 2/\beta\right[$ and take $\lambda \in \left]0, 1 - \frac{\delta\beta}{2}\right[$ and $\gamma \in \left]0, \frac{1}{\|L\|^2}\right]$. Given an initial guess $(\mathbf{z}^0, v^0) \in \mathcal{H}^{n-1} \times \mathcal{G}$, consider the sequence $(\mathbf{z}^k, v^k)_{k\in\mathbb{N}}$ generated by Algorithm 7.11. Then the following assertions hold.*

(i) The sequence $(\mathbf{z}^k, v^k)_{k\in\mathbb{N}}$ converges weakly to a point $(\bar{\mathbf{z}}, \bar{v}) \in \operatorname{Fix}\widetilde{\mathcal{T}}$, where $\widetilde{\mathcal{T}}$ is the fixed point operator defined by (7.32)-(7.33).

(ii) The sequence $(x_1^k, \ldots, x_n^k, y^k)_{k\in\mathbb{N}}$ converges weakly to $(\bar{x}, \ldots, \bar{x}, L\bar{x})$ with $\bar{x}$ solving the primal inclusion (7.28).

(iii) For all $i \in [\![1, n]\!]$, the sequence $(\gamma L x_i^k - v^k)_{k\in\mathbb{N}}$ converges weakly to a point $\gamma L\bar{x} - \bar{v}$ solving the dual inclusion (7.29).

(iv) The sequence $\big(T_i(x_i^k)\big)_{k\in\mathbb{N}}$ converges strongly to $T_i(\bar{x})$ for all $i \in [\![1, n-1]\!]$.

We briefly sketch the proof of Theorem 7.12, which follows in an analogous manner to some of the results already presented.

Proof. By an argument similar to that in Lemmas 6.4 and 7.1, we can prove that

$$\operatorname{Fix}\widetilde{\mathcal{T}} \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^{n} A_i + \sum_{i=1}^{n-1} T_i + L^*BL\right) \neq \emptyset.$$

Next, let $(\mathbf{z}, v) \in \mathcal{H}^{n-1} \times \mathcal{G}$ and $(\bar{\mathbf{z}}, \bar{v}) \in \operatorname{Fix}\widetilde{\mathcal{T}}$. Combining the proofs of Lemmas 6.5 and 7.3 yields the inequality

$$\begin{aligned}\big\|\widetilde{\mathcal{T}}(\mathbf{z}, v) - (\bar{\mathbf{z}}, \bar{v})\big\|_\gamma^2 + \left(\frac{1-\lambda}{\lambda} - \frac{\delta\beta}{2\lambda}\right)\big\|\big((\mathrm{Id} - \widetilde{\mathcal{T}})(\mathbf{z}, v)\big)_{i=1}^{n-1}\big\|^2 + \frac{1}{\gamma}\left(\frac{1-\lambda}{\lambda}\right)\big\|(\mathrm{Id} - \widetilde{\mathcal{T}})(\mathbf{z}, v)_n\big\|^2 \\ + \frac{1-\gamma\|L\|^2}{\lambda}\Big\|\sum_{i=1}^{n-1}(\mathrm{Id} - \widetilde{\mathcal{T}})(\mathbf{z}, v)_i\Big\|^2 \leq \big\|(\mathbf{z}, v) - (\bar{\mathbf{z}}, \bar{v})\big\|_\gamma^2.\end{aligned}$$

Thus, the sequence $(\mathbf{z}^k, v^k)_{k\in\mathbb{N}}$ is Fejér monotone with respect to $\operatorname{Fix}\widetilde{\mathcal{T}}$ and $(\mathbf{x}^k, y^k)_{k\in\mathbb{N}}$ is bounded. Further, $\mathbf{z}^{k+1} - \mathbf{z}^k \to 0$ and $v^{k+1} - v^k \to 0$, which in turn imply $Lx_n^k - y^k \to 0$ and $x_i^k - x_{i-1}^k \to 0$, for all $i \in [\![1, n-1]\!]$.

Now, consider the maximally monotone operator $C : \mathcal{H}^{n-1} \times \mathcal{G} \rightrightarrows \mathcal{H}^{n-1} \times \mathcal{G}$ defined as

$$C := \begin{pmatrix} (\delta A_1)^{-1} \\ \big(\delta(A_2 + T_1)\big)^{-1} \\ \vdots \\ \big(\delta(A_{n-1} + T_{n-2})\big)^{-1} \\ \delta(A_n + T_{n-1}) \\ (\delta B)^{-1} \end{pmatrix} + \begin{pmatrix} 0 & 0 & \dots & 0 & -\mathrm{Id} & 0 \\ 0 & 0 & \dots & 0 & -\mathrm{Id} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & -\mathrm{Id} & 0 \\ \mathrm{Id} & \mathrm{Id} & \dots & \mathrm{Id} & 0 & L^* \\ 0 & 0 & \dots & 0 & -L & 0 \end{pmatrix}.$$

Then, (i)-(iii) follow from taking the limit along a subsequence weakly convergent to a weak cluster point of $(\mathbf{z}^k, v^k, \mathbf{x}^k, y^k)_{k\in\mathbb{N}}$, using β-Lipschitz continuity of $T_1, \dots, T_{n-1}$, demiclosedness of C and noting that (7.31) can be written as

$$C\begin{pmatrix} z_1^k - x_1^k \\ (z_2^k - x_2^k) - (z_1^k - x_1^k) + \delta t_2^k \\ \vdots \\ (z_{n-1}^k - x_{n-1}^k) - (z_{n-2}^k - x_{n-2}^k) + \delta t_{n-1}^k \\ x_n^k \\ \gamma\left(L(x_1^k + x_n^k) - y^k\right) - v^k \end{pmatrix} \ni \begin{pmatrix} x_1^k - x_n^k \\ x_2^k - x_n^k \\ \vdots \\ x_{n-1}^k - x_n^k \\ x_1^k - x_n^k + \delta\sum_{i=1}^{n-1} t_{i+1}^k + \gamma L^*\left(Lx_n^k - y^k\right) \\ y^k - Lx_n^k \end{pmatrix},$$

where $t_i^k := T_{i-1}(x_i^k) - T_{i-1}(x_{i-1}^k)$.

It only remains to prove (iv). Proceeding in the same fashion as in Theorem 6.7 (iii), firm nonexpansiveness of the resolvents $J_{\delta A_1}, \dots, J_{\delta A_n}$ and $\frac{1}{\beta}$-cocoercivity of $T_1, \dots, T_{n-1}$

yield the inequality

$$\begin{aligned}\frac{\delta}{\beta}\sum_{i=1}^{n-1}\|T_i(x_i^k)-T_i(\bar{x})\|^2 &\\ \leq \langle x_n^k-\bar{x}, x_1^k-x_n^k\rangle + \langle x_1^k-x_n^k, (z_i^k-x_i^k)-(\bar{z}_i-\bar{x})\rangle &\\ +\sum_{i=2}^{n-1}\langle x_i^k-x_n^k, (z_i^k-x_i^k)-(z_{i-1}^k-x_{i-1}^k)-(\bar{z}_i-\bar{z}_{i-1})\rangle &\\ -\delta\sum_{i=1}^{n-1}\langle x_{i+1}^k-x_i^k, T_i(x_i^k)-T_i(\bar{x})\rangle - \langle Lx_n^k-L\bar{x}, \gamma L(x_1^k-\bar{x})-(v^k-\bar{v})\rangle.&\end{aligned} \tag{7.34}$$

Further, resorting to the monotonicity of δB we get

$$0 \leq \langle y^k-L\bar{x}, \gamma L(x_1^k-\bar{x})-(v^k-\bar{v})\rangle + \gamma\langle y^k-L\bar{x}, Lx_n^k-y^k\rangle. \tag{7.35}$$

Combining (7.34) and (7.35) and taking limits, the claim follows. □

Remark 7.13. Determining whether Theorem 7.12 has minimal lifting would require to rigorously establish a minimal lifting result for the monotone inclusion problem (7.28)-(7.29), which is out of the scope of this thesis. Nevertheless, from the above it is clear that Theorem 7.12 results in a method with $(n-1,m)$-fold lifting when applied to (7.28)-(7.29) with $n \geq 2$ and m linearly composed maximally monotone operators. Therefore, it also reduces the dimension of the underlying space of the algorithm with respect to other existing methods in the literature, see [59, 65, 109, 239].

In addition, if some of the operators $T_1, \ldots, T_{n-2}$ are not cocoercive but monotone and Lipschitz continuous, we can modify Algorithm 7.11 in the way described in Remark 6.21. In this context, the proof of Theorem 7.12 can be tailored to obtain (i)-(iii) with a smaller range of admissible parameters.

7.4 A numerical experiment in image recovery

In this section, we test our algorithm for solving an ill-conditioned linear inverse problem which arises in image deblurring and denoising. Let $b \in \mathbb{R}^n$ be an observed blurred and noisy image of size $N_1 \times N_2$, with $n = N_1N_2$ for grayscale and $n = 3N_1N_2$ for color images, and denote by $M \in \mathbb{R}^{n\times n}$ the blur operator. The model considered on this occasion is

given by the following regularized convex non-differentiable problem

$$\inf_{s\in\mathbb{R}^n}\left\{\|Ms-b\|_1+\alpha_1\|Ws\|_1+\alpha_2 TV(s)+\iota_{[0,1]^n}(s)\right\}, \tag{7.36}$$

where $\alpha_1,\alpha_2>0$ are regularization parameters, $\iota_{[0,1]^n}$ denotes the indicator function of the set $[0,1]^n$, $TV:\mathbb{R}^n\to\mathbb{R}$ is the isotropic total variation function and W is the linear operator given by the normalized *nonstandard Haar transform*, which is orthogonal (see, e.g., [228]).

Recalling Remark 7.7, it is of interest to consider a mechanism which allows tuning the parameter γ appearing in the definition of the norm given by the inner product in (7.5) to an appropriate value. To this aim, we perform in (7.36) a change of variable of the form $s=\mu x$, with $\mu>0$, and instead handle the problem

$$\inf_{x\in\mathbb{R}^n}\left\{\mu\left\|Mx-\frac{b}{\mu}\right\|_1+\alpha_1\mu\|Wx\|_1+\alpha_2 TV(\mu x)+\iota_{[0,1/\mu]^n}(x)\right\}. \tag{7.37}$$

Below we shall see the way in which the choice of μ can help setting a suitable parameter γ. First, note that the parametrized total variation can be written as $TV(\mu\,\cdot)=\|L(\cdot)\|_\times$, with $L=D/\mu$ and D the discrete gradient operator in (2.8). As a consequence, an upper bound of the squared Lipschitz constant of L is given by $\|L\|^2\leq 8\mu^2$.

By [44, Proposition 27.5], solving (7.37) is equivalent to obtaining a solution to the composite inclusion

$$\text{find } x\in\operatorname{zer}\left(N_{[0,1/\mu]^n}+W^*\circ\partial g_1\circ W+M^T\circ\partial g_2\circ M+L^*\circ\partial g_3\circ L\right), \tag{7.38}$$

with $g_1:\mathbb{R}^n\to\mathbb{R}$, $g_1(y)=\alpha_1\mu\|y\|_1$, $g_2:\mathbb{R}^n\to\mathbb{R}$, $g_2(y)=\mu\|y-b/\mu\|_1$, $g_3:\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}$, $g_3(p,q)=\alpha_2\|(p,q)\|_\times$, and $N_{[0,1/\mu]^n}$ the normal cone operator to the set $[0,1/\mu]^n$. In order to implement Algorithm 7.8 for solving (7.38), we need the expression of the following resolvents and proximity operators. The orthogonality of W and Proposition 2.22 imply that the second term in (7.38) is a maximally monotone operator with resolvent

$$J_{W^*\circ\partial g_1\circ W}=\mathrm{Id}-W^*\circ\left(\mathrm{Id}-\operatorname{prox}_{g_1}\right)\circ W=\mathrm{Id}-W^*\circ\operatorname{prox}_{g_1^*}\circ W,$$

where the conjugate function to g_1 is equal to the indicator function $\iota_{[-\alpha_1\mu,\alpha_1\mu]^n}$, and thus $\operatorname{prox}_{g_1^*}=P_{[-\alpha_1\mu,\alpha_1\mu]^n}$. Given $\sigma>0$, the proximity operators of g_2 and g_3 are, respectively,

$$\operatorname{prox}_{\sigma g_2}(x)=\frac{b}{\mu}+\operatorname{prox}_{\sigma\mu\|\cdot\|_1}\left(x-\frac{b}{\mu}\right)=\frac{b}{\mu}+\operatorname{sign}\left(x-\frac{b}{\mu}\right)\odot\left[\left|x-\frac{b}{\mu}\right|-\sigma\mu\right]_+,$$

where $\odot$ denotes element-wise product and $[\cdot]_+$ and $|\cdot|$ are applied element-wise, and

$$\operatorname{prox}_{\sigma g_3} = \operatorname{Id} - \sigma \operatorname{prox}_{\frac{1}{\sigma} g_3^*} \circ \frac{1}{\sigma} \operatorname{Id} \quad = \operatorname{Id} - \sigma P_S \circ \frac{1}{\sigma} \operatorname{Id},$$

since the conjugate function of g_3 is $g_3^* = \iota_S$, where S is defined likewise to (2.7) and whose projection is given in Proposition 2.37.

Hence, when choosing $z^0 \in \mathbb{R}^n$, $v_1^0 \in \mathbb{R}^n$ and $v_2^0 \in \mathbb{R}^n \times \mathbb{R}^n$ as starting values, and letting $\lambda \in]0,1[$ and $\gamma \in]0, 1/(\|M\|^2 + \|L\|^2)]$, the iterative scheme in Algorithm 7.8 becomes

$$\left\lfloor \begin{aligned}
x_1^k &= P_{[0,1/\mu]^n}(z^k), \\
x_2^k &= \big(\operatorname{Id} - W^* \circ P_{[-\alpha_1\mu, \alpha_1\mu]^n} \circ W\big) \\
&\qquad \big(2x_1^k - z^k - M^T(\gamma M x_1^k - v_1^k) - L^*(\gamma L x_1^k - v_2^k)\big), \\
y_1^k &= \frac{b}{\mu} + \operatorname{prox}_{\frac{\mu}{\gamma}\|\cdot\|_1}\left(M(x_1^k + x_2^k) - \frac{v_1^k}{\gamma} - \frac{b}{\mu}\right), \\
y_2^k &= \left(\operatorname{Id} - \frac{1}{\gamma} P_S\right)\left(\gamma L(x_1^k + x_2^k) - v_2^k\right), \\
z^{k+1} &= z^k + \lambda(x_2^k - x_1^k), \\
v_1^{k+1} &= v_1^k + \lambda\gamma(y_1^k - M x_2^k), \\
v_2^{k+1} &= v_2^k + \lambda\gamma(y_2^k - L x_2^k).
\end{aligned} \right.$$

In our experiment, we replicate the problem in the survey [60, Section 4.2], where an extensive comparison between different primal-dual algorithms is presented. Since the best performing algorithm is the Douglas–Rachford type primal-dual method in [60, Algorithm 11], we limit our comparison to this algorithm, whose detailed implementation is given in the cited work. The method has $(2,2)$-fold lifting for this problem. We ran our experiments in Matlab, making use of the inbuilt functions `fspecial` and `imfilter` to define an operator M which is a Gaussian blur operator of size 9×9 with standard deviation 4 and reflexive boundary conditions. In particular, M verifies $\|M\| = 1$ and $M^T = M$. We employed as observed image b a picture taken at the Schönbrunn Palace Gardens (Vienna) subjected to the already specified blur followed by the addition of a zero-mean Gaussian noise with standard deviation 10^{-3} (see Figure 7.2). To test the influence on the performance of the picture size, we resized the original picture to different pixel resolutions (see Table 7.1).

When measuring the quality of the restored images, we use the *improvement in signal-to-noise-ratio* (ISNR), which is given by

$$\mathrm{ISNR}_k = 10 \log_{10} \left(\frac{\|x - b\|^2}{\|x - x^k\|^2} \right),$$

where x and x^k are the original and the reconstructed image at iteration k, respectively. We tuned the regularization parameters in order to guarantee an adequate ISNR value for the restored images, setting $\alpha_1 := 0.005$ and $\alpha_2 := 0.009$.

We recall that the stepsize parameter γ of Algorithm 7.8 must be taken in the interval $\gamma \in]0, 1/(\|M\|^2 + \|L\|^2)] \supseteq]0, 1/(1 + 8\mu^2)]$. When $\mu = 1$ (i.e., we solve (7.36)), the latter interval is $]0, 0.111]$. In our numerical experiments we empirically observed that a very small stepsize negatively affects the performance of the algorithm, as mentioned in Remark 7.7. After testing different options, the most convenient one seems to be $\mu = 1/\sqrt{8}$, which implies making the Lipschitz constant of both linear operators in the problem equal to 1.

The initialization of each of the methods was the following:

- DR1([60, Algorithm 3.1]): starting points $x_0 = b$ and $(v_{1,0}, v_{2,0}, v_{3,0}) = (0, 0, 0)$, $\sigma_1 = 1$, $\sigma_2 = 0.05$, $\sigma_3 = 0.05$, $\tau = 1(\sigma_1 + \sigma_2 + 8\sigma_3)^{-1} - 0.01$, $\lambda_n = 1.5$ for all $n \in \mathbb{N}$.
- Algorithm 7.8 with $\mu = 1$: starting points $z^0 = b$ and $(v_1^0, v_2^0) = (0, 0)$, $\lambda = 0.99$ and $\gamma = 1/9$.
- Algorithm 7.8 with $\mu = 1/\sqrt{8}$: starting points $z^0 = b/\mu$ and $(v_1^0, v_2^0) = (0, 0)$, $\lambda = 0.99$ and $\gamma = 1/2$.

We performed 400 iterations of each of the algorithms and compared the values of the objective function in (7.37) and the ISNR with respect to the CPU time, which provides a more realistic comparison than iteration count, since DR1 has a higher computational cost per iteration than Algorithm 7.8. The tests were ran on a computer of Intel Core i7-12700H 2.30 GHz with 16GB RAM, under Windows 11 (64-bit). The algorithms were ran 3 times, once for each of the RGB components of the picture. The evolution in CPU time of adding these 3 values of the objective function and those of the ISNR for the 640×768-sized picture are represented in Figure 7.1, where we observe that Algorithm 7.8 with $\mu = 1/\sqrt{8}$ obtains slightly better values than those returned by DR1, but in significantly less time.

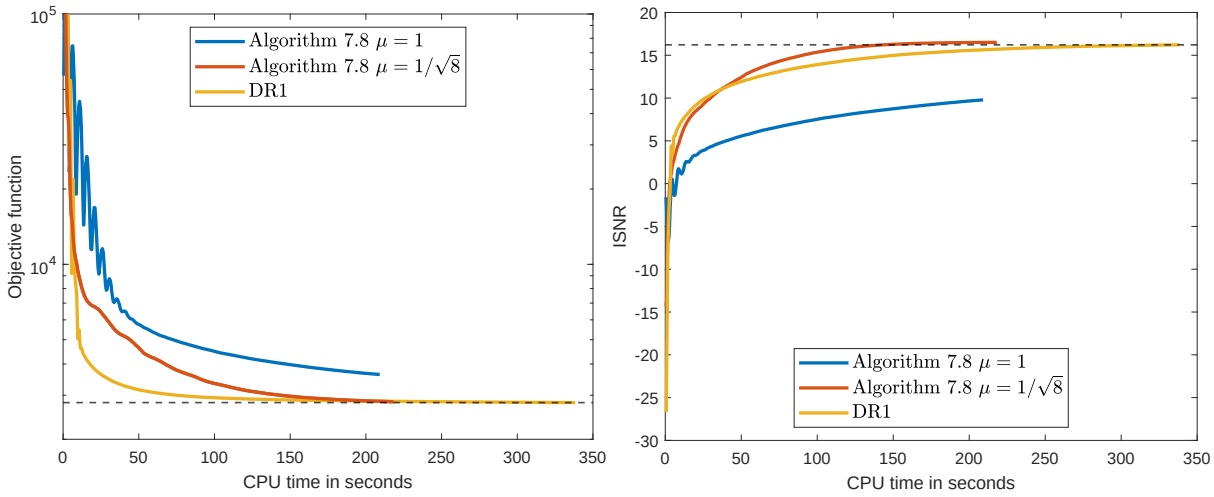


FIGURE 7.1: The evolution of the values of the objective function and of the ISNR in CPU time for 400 iterations of Algorithm 7.8 with $\mu = 1$ and $\mu = 1\sqrt{8}$ and DR1, using the 640×768 pixels image displayed in Figure 7.2.

The restored images are presented in Figure 7.2. There is no much difference between the ones corresponding to Algorithm 7.8 with $\mu = 1/\sqrt{8}$ (bottom-middle) and DR1 (bottom-right), but a close look at the image obtained with Algorithm 7.8 with $\mu = 1$ permits to observe its worse quality. To show that this trend in the performance of the algorithms is not affected by the image size, we present in Table 7.1 the results from running the algorithms on the same picture for five different pixel resolutions. Overall, we notice that the CPU time required for computing the 400 iterations is significantly lower for Algorithm 7.8, as expected. On average, DR1 required 50% more time than Algorithm 7.8 to compute the 400 iterations, independently of the size of the image. Regarding the parameter μ, Algorithm 7.8 with $\mu = 1$ is notably outperformed by the other two methods, making thus clear the influence that this parameter has on it. The function values obtained were slightly lower for DR1, while the ISNR was slightly lower for Algorithm 7.8 with $\mu = 1/\sqrt{8}$, which implies that both algorithms performed similarly with respect to the restored image quality.

Interpretation of the results of the experiments The experimental results show that, after performing the same number of iterations, Algorithm 7.8 with $\mu = 1/\sqrt{8}$ obtains similar results in the function values and the measurement in the quality of the image recovery than those obtained by DR1, but in considerably less time. This decrease in the running time can be attributed to the reduction in the lifting of the operator. Although in the first iterations DR1 achieves a larger reduction of the objective function,

the quality of the restored image is not sufficient, as assessed by the low ISNR values. On the other hand, Algorithm 7.8 with $\mu = 1$ can be discarded, as it obtains higher objective and lower ISNR values. Consequently, Algorithm 7.8 with $\mu = 1/\sqrt{8}$ is the preferable choice to address problem (7.36).

FIGURE 7.2: On the top, the original 640×768 pixels image and the blurred and noisy image. On the bottom the images restored after computing 400 iterations of Algorithm 7.8 with $\mu = 1$ (left) and $\mu = 1/\sqrt{8}$ (middle), and DR1 (right).

Resolution		80×96	160×192	320×384	640×768	1280×1536
Function values	$\mu = 1$	55.4	226.3	919.4	3630.3	13 177.5
	$\mu = 1/\sqrt{8}$	43.1	174.3	710.6	2825.2	10 410.6
	DR1	42.9	173.4	705.2	2804.5	10 377.4
ISNR	$\mu = 1$	9.7	8.4	8.7	9.8	12.8
	$\mu = 1/\sqrt{8}$	15.8	14.3	14.9	16.5	21.0
	DR1	15.8	14.2	14.8	16.4	21.0
CPU time	$\mu = 1$	1.8	7.6	27.9	209.1	1453.3
	$\mu = 1/\sqrt{8}$	4.5	10.1	44.8	218.1	1476.9
	DR1	6.7	12.0	120.2	338.2	2561.3

TABLE 7.1: Results from running on the picture displayed in Figure 7.2 (for various pixel resolutions) 400 iterations of Algorithm 7.8 with $\mu = 1$ and $\mu = 1/\sqrt{8}$, and DR1.

Part II

Splitting methods beyond the monotonicity framework

Chapter 8

The Boosted Double-proximal Subgradient Algorithm for nonconvex optimization

In this chapter we leave the *convex world* and delve into the field of nonconvex optimization, ruled by completely different principles. Quoting Tyrrel Rockafellar, one of the fathers of modern optimization, "*...in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity*". In the following, we shall observe that the machinery employed for analyzing nonconvex problems immensely differs from the one presented in Part I of this thesis. Nevertheless, the search for numerical methods adept at harnessing the inherent structure of the optimization problem prevails. Being able to duly split the original problem into simpler tasks becomes imperative, as an appropriate choice of *"splitting"* crucially influences the quality of the solution.

In this context, splitting methods have risen as a preferable option for structured nonconvex minimization. The Douglas–Rachford algorithm performs exceptionally well for the resolution of combinatorial problems [12, 15, 18, 164]. However, it is usually employed as a heuristic, as very few is known about its convergence in the nonconvex setting besides some particular cases (see, e.g., [11, 13, 47]). The so-called *Difference of Convex functions Algorithm* (DCA), initially introduced by Pham Dinh and El Bernoussi [233], pioneered the development of methods for the minimization of differences of convex functions [24, 129, 205, 227, 234].

One of the cornerstones in the understanding of the minimization of a nonsmooth nonconvex function was the abstract convergence theorem devised by Attouch, Bolte and Svaiter [31]. The theory there developed relies on the verification of the *Kurdyka–Łojasiewicz property* [157, 166] and is suitable for *descent methods*, namely, those algorithms that generate a nonincreasing sequence with respect to the objective function. This framework furnished the necessary tools for the convergence analysis of well-known

methods applied to nonconvex problems (for instance, the proximal point algorithm and the forward-backward method [31]), and propitiated the development of new schemes, see, e.g., [7, 38, 66, 139].

In this chapter, we propose a novel descent method for the structured nonconvex optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) + g(x) - \sum_{i=1}^{p} h_i(\Psi_i(x)), \tag{$\mathcal{P}$}$$

where the function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz and satisfies the *descent lemma*, $g : \mathbb{R}^n \to]-\infty, +\infty]$ is lower-semicontinuous and prox-bounded, $h_i : \mathbb{R}^{m_i} \to \mathbb{R}$ are convex continuous functions and $\Psi_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ are differentiable functions with Lipschitz continuous gradients, for $i = 1, \ldots, p$ (see Assumption 8.14 for more specific details).

Problems in this form appear in a broad variety of fields such as machine learning, image recovery or signal processing [169, 243, 248]. Numerous algorithms have been developed to handle simpler instances of ($\mathcal{P}$) (see, e.g., [7, 38, 171, 229]), but as far as we are aware of, not for the more general problem that we address here.

We present a new splitting algorithm, named *Boosted Double-proximal Subgradient Algorithm* (BDSA), that makes use of the inherent structure of problem ($\mathcal{P}$). The inspiration of the underlying scheme comes from the *Double-proximal Gradient Algorithm* (DGA) proposed by Banert and Boţ [38]. More specifically, the method employs the subdifferential of f, the gradients of Ψ_i and the *proximal point operators* of the functions g and h_i, for $i = 1, \ldots, p$. Subsequently, an optional linesearch, not included in DGA, can be performed to obtain the final update, which intends to steer (or *"boost"*) the iteration to a point with a reduced value of the objective function, in a similar manner than the linesearch introduced in [24, 27] permits to accelerate the DCA.

In our numerical tests, the addition of the linesearch is observed to provide substantial improvements in the performance of the method. On the one hand, it *accelerates* the algorithm, significantly reducing both the number of iterations and the time that it needs to converge. On the other hand, it may help the sequence to converge to *better solutions*. Indeed, note that the algorithms employed for tackling this class of nonconvex problems usually converge to critical points (see Section 8.2). Being a critical point is a necessary condition for local optimality, but not sufficient. Therefore, the algorithms often converge to critical points which are not even local minima. The linesearch introduced in our scheme seems to help the method to converge to critical points with lower value of the objective function.

The main contributions of this chapter are the following.

- In Section 8.1.2, we develop the framework for allowing f to be taken in the class of upper-$\mathcal{C}^2$ functions. This family of functions increases the possibilities of distribution of the elements for splitting the problem and seems to not have been integrated into numerical schemes before our work.
- The convergence of BDSA is analyzed in Section 8.2.1. In addition, in Section 8.2.2, we make use of the Kurdyka–Łojasiewicz property to prove in Theorems 8.28 and 8.30 its global convergence and deduce some convergence rates, respectively.
- Section 8.3 contains multiple numerical experiments showing the good performance of BDSA in comparison with *"non-boosted"* methods and some *inertial algorithms*.
- The benefit of the linesearch for escaping non-optimal critical points is illustrated with the introduction of two new challenging test functions in Section 8.3.1.
- The reduction in time and iterations is exemplified with an application of the *minimum sum-of-squares clustering problem* and a generalization of the classical *Heron problem* in Sections 8.3.2 and 8.3.3, respectively.

Unless otherwise stated, the results in this chapter first appeared in [29].

8.1 Some further notions of variational analysis

We start by presenting the essential concepts for the analysis of nonconvex optimization problems. All the results in Section 8.1.1 can be found in the fundamental books of Rockafellar and Wets [218], and Mordukhovich [183].

8.1.1 Notational conventions

Throughout this chapter, the notations $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ represent the Euclidean norm and inner product in $\mathbb{R}^n$, respectively. Given p positive integers $m_1,\dots,m_p$, the inner product of the product space $\mathbb{R}^{m_1}\times\cdots\times\mathbb{R}^{m_p}$ is defined by

$$\langle(x_1,\dots,x_p),(y_1,\dots,y_p)\rangle := \sum_{i=1}^{p}\langle x_i,y_i\rangle, \quad \text{for all } (x_1,\dots,x_p),(y_1,\dots,y_p)\in\mathbb{R}^m,$$

with $m := \sum_{i=1}^{p} m_i$, and its induced norm is denoted by $\|(x_1,\dots,x_p)\|$. Again, vectors in product spaces will be marked with bold, e.g., $\mathbf{x}=(x_1,\dots,x_p)\in\mathbb{R}^m$.

Given some constant $L \geq 0$, a vector-valued function $F : C \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is said to be *L-Lipschitz continuous* on C if

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \text{for all } x, y \in C,$$

and *locally Lipschitz continuous* around $\bar{x} \in C$ if it is Lipschitz continuous in some neighborhood of $\bar{x}$. Observe that, in contrast to the definition of Lipschitz continuous operator in Definition 2.1, here we are allowing $L = 0$. Constant functions are 0-Lipschitz continuous.

A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is *strictly differentiable* at a point $\bar{x} \in \mathbb{R}^n$ if $f(\bar{x})$ is finite and there exists a vector $v \in \mathbb{R}^n$ such that

$$\lim_{x, x' \to \bar{x}} \frac{f(x') - f(x) - \langle v, x' - x \rangle}{\|x' - x\|} = 0, \quad \text{with } x' \neq x.$$

In this case, v coincides with the gradient of f at $\bar{x}$, denoted as $\nabla f(\bar{x})$. In the following, we also use the same symbol to denote the transpose of the Jacobian matrix of a multivariable function $F = (F_1, \ldots, F_m) : \mathbb{R}^n \to \mathbb{R}^m$, namely, the matrix of gradients given by $\nabla F(x) = (\nabla F_1(x), \ldots, \nabla F_m(x))$. We say that a function is of class $\mathcal{C}^1$ if it is differentiable with continuous gradient. Furthermore, if its gradient is L-Lipschitz continuous we say f is *L-smooth* .

The *upper Dini directional derivative* of f at some point $\bar{x} \in \operatorname{dom} f$ in the direction $d \in \mathbb{R}^n$ is defined as

$$d^+ f(\bar{x}; d) := \limsup_{t \downarrow 0} \frac{f(\bar{x} + td) - f(\bar{x})}{t}.$$

Finally, given a set-valued mapping $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$, the *Painlevé–Kuratowski upper limit* of F at $\bar{x}$ is defined as

$$\operatorname*{Lim\,sup}_{x \to \bar{x}} F(x) := \left\{ y \in \mathbb{R}^m \;\middle|\; \exists\, x^k \to \bar{x},\, y^k \to y \text{ with } y^k \in F(x^k),\, \forall k \in \mathbb{N} \right\}.$$

8.1.1.1 Generalized subdifferentials

We devote this section to briefly motivate the use of generalized subdifferentials, which are essential for deriving optimality conditions for the minimization of (nonsmooth) nonconvex functions.

Definition 8.1 (Generalized subdifferentials). *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and $\bar{x} \in \operatorname{dom} f$.*

(i) The regular *(or* Fréchet*)* subdifferential *of f at $\bar{x} \in \mathbb{R}^n$ is the closed and convex set of* regular subgradients

$$\hat{\partial} f(\bar{x}) := \{v \in \mathbb{R}^n : f(x) \geq f(\bar{x}) + \langle v, x - \bar{x}\rangle + o(\|x - \bar{x}\|)\}.$$

*(ii) We say that $v \in \mathbb{R}^n$ is a (*basic*, or* limiting*, or* Mordukhovich*)* subgradient *of f at $\bar{x}$ if there exist sequences $(x^k)_{k\in\mathbb{N}}$ and $(v^k)_{k\in\mathbb{N}}$, with $v^k \in \hat{\partial} f(x^k)$ for all $k \in \mathbb{N}$, such that $x^k \to \bar{x}$, $f(x^k) \to f(\bar{x})$ and $v^k \to v$, as $k \to \infty$.*

*(iii) The (*basic*, or* limiting*, or* Mordukhovich*)* subdifferential *of f is the set of all (basic) subgradients of f at $\bar{x}$, denoted by $\partial f(\bar{x})$.*

We use the convention $\hat{\partial} f(\bar{x}) = \partial f(\bar{x}) := \emptyset$ if $|f(\bar{x})| = +\infty$. We say that f is lower regular *at $\bar{x} \in \operatorname{dom} f$ if $\partial f(\bar{x}) = \hat{\partial} f(\bar{x})$.*

Note that we are allowing ourselves to an abuse of notation in the nomenclature of the basic subdifferential, as ∂f was also employed for the convex subdifferential (2.4). The reason behind this will be clear soon.

The calculus rules for the regular and basic subdifferentials are limited. Therefore, it is often useful to consider the following convexification of the basic subdifferential. Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a locally Lipschitz continuous function around a point $\bar{x} \in \operatorname{dom} f$, *Clarke's generalized gradient* or *Clarke's subdifferential* of f at $\bar{x}$ is defined as

$$\partial_C f(\bar{x}) := \operatorname{co} \partial f(\bar{x}).$$

Rademacher's theorem [218, Theorem 9.60] states that f is almost everywhere differentiable in some neighborhood of $\bar{x}$, which yields the equivalent representation

$$\partial_C f(\bar{x}) = \operatorname{co}\left\{\lim_{x^k \to \bar{x}} \nabla f(x^k)\right\}. \tag{8.1}$$

Let $f : \mathbb{R} \to \overline{\mathbb{R}}$ be locally Lipschitz continuous and $\bar{x} \in \operatorname{dom} f$. We have the following relationships between the regular, the basic and Clarke's subdifferentials:

$$\hat{\partial} f(\bar{x}) \subseteq \partial f(\bar{x}) \subseteq \partial_C f(\bar{x}). \tag{8.2}$$

If in addition f is convex, the above inclusions become equalities and all three subdifferentials coincide with the convex subdifferential in (2.4). If f is strictly differentiable at $\bar{x}$, then all its subdifferentials are singletons and coincide with its gradient.

Proposition 8.2 (Generalized Fermat's rule, [183, Proposition 1.30]). *Consider $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and let $\bar{x} \in \operatorname{dom} f$. If $\bar{x}$ is a local minimizer of f, then*

$$0 \in \hat{\partial} f(\bar{x}) \quad \text{and} \quad 0 \in \partial f(\bar{x}).$$

Obviously, if f is convex the above inclusions become a necessary and sufficient condition for global optimality.

Remark 8.3. In view of (8.2), the subdifferential inclusion

$$0 \in \partial_C f(\bar{x}),$$

also constitutes a necessary condition for $\bar{x}$ to be a local minimizer of the locally Lipschitz continuous function f. Nevertheless, it is well recognized that Clarke's generalized gradient is less suited for optimization purposes. For a simple motivating example, let $f : \mathbb{R} \to \mathbb{R}$ be the opposite of the absolute value, i.e., $f(\cdot) = -|\cdot|$. Then Clarke's generalized gradient of f at the local maximum given by $\bar{x} = 0$ takes the form

$$\partial_f(0) = [-1, 1],$$

and hence satisfies the necessary optimality condition. In turn, we have

$$0 \notin \hat{\partial} f(0) = \emptyset \quad \text{and} \quad 0 \notin \partial f(0) = \{-1, 1\},$$

which do not verify the generalized Fermat rule.

Proposition 8.4 (Sum rules for regular subgradients). *Let $f, g : \mathbb{R}^n \to \overline{\mathbb{R}}$ be such that $\bar{x} \in \operatorname{dom} f \cap \operatorname{dom} g$.*

(i) If f is of class $\mathcal{C}^1$ around $\bar{x}$, then

$$\partial(f + g)(\bar{x}) = \nabla f(\bar{x}) + \partial g(\bar{x}).$$

(ii) If f is locally Lipschitz continuous around $\bar{x}$ and g is l.s.c. at this point, then

$$\partial(f + g)(\bar{x}) \subseteq \partial f(\bar{x}) + \partial g(\bar{x}). \tag{8.3}$$

If in addition, both f and g are lower regular, (8.3) holds as an equality.

Proof. (i) [183, Proposition 1.30]. (ii) [183, Corollary 2.20]. □

8.1.1.2 Prox-bounded functions

We now extend the definition of proximity operator applied to functions which are not necessarily convex.

We define the *proximal point mapping* of a proper l.s.c. function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ as the multifunction $\operatorname{prox}_{\gamma f} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ which maps a point x to the solution set of the optimization problem

$$\operatorname{prox}_{\gamma f}(x) := \operatorname*{argmin}_{u \in \mathbb{R}^n} \left\{ f(u) + \frac{1}{2\gamma}\|u - x\|^2 \right\}. \tag{8.4}$$

A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is said to be *prox-bounded* if there exists some $\gamma > 0$ such that $f(\cdot) + \frac{1}{2\gamma}\|\cdot - x\|^2$ is bounded from below for all $x \in \mathbb{R}^n$. The supremum of the set of all such constants γ is called the *prox-boundedness threshold* of f and is denoted by γ^f.

Proposition 8.5 ([218, Theorem 1.25]). *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be proper, l.s.c. and prox-bounded with threshold γ^f. The following assertions hold.*

(i) The proximal mapping $\operatorname{prox}_{\gamma f}$ has full domain for any $\gamma \in]0, \gamma^f[$.

(ii) Let $(x^k)_{k\in\mathbb{N}}$, $(v^k)_{k\in\mathbb{N}} \subseteq \mathbb{R}^n$ and $(\gamma_k)_{k\in\mathbb{N}} \subseteq]0, \gamma^f[$ be such that $x^k \to \bar{x}$, $\gamma_k \to \bar{\gamma} \in]0, \gamma^f[$ and

$$v^k \in \operatorname{prox}_{\gamma_k f}(x^k), \qquad \text{for all } k \in \mathbb{N}.$$

Then $(v^k)_{k\in\mathbb{N}}$ is bounded and all its cluster points lie in $\operatorname{prox}_{\bar{\gamma} f}(\bar{x})$.

All proper l.s.c. convex functions are prox-bounded, and in this case (8.4) is single-valued. Nonetheless, the class is much larger. Any proper l.s.c. function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ that is bounded from below by an affine function has a threshold of prox-boundedness $\gamma^f = +\infty$ (see [218, Example 3.28]). For example, the indicator function ι_C of a nonempty and closed set $C \subseteq \mathbb{R}^n$ is prox-bounded with threshold $\gamma^{\iota_C} = +\infty$.

The following examples provide formulas for the proximity operators of some nonconvex functions that will be needed in the sequel.

Example 8.6. *For all $i = 1, \ldots, p$, let $C_i \subseteq \mathbb{R}^n$ be a nonempty closed convex set and define the union $C := \bigcup_{i=1}^{p} C_i$. Since C is closed, the indicator ι_C is prox-bounded with proximity operator given by the* set-valued projection operator

$$\operatorname{prox}_{\gamma \iota_C}(x) = P_C(x) := \left\{ P_{C_i}(x) \,:\, d(x, C) = \|x - P_{C_i}(x)\|,\ i = 1, \ldots, p \right\},$$

for all $\gamma > 0$ and $x \in \mathbb{R}^n$, and where $d(\cdot, C)$ stands for the distance function to C and P_{C_i} is the single-valued projection on the closed convex set C_i. See Figure 8.1 for an illustration.

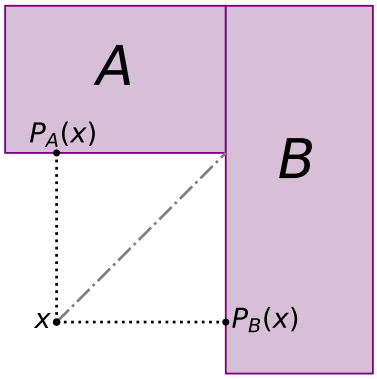


FIGURE 8.1: Set-valued projection onto the union of convex sets $C := A \cup B$. The projection of x is multi-valued, given by $P_C(x) = \{P_A(x), P_B(x)\}$.

Example 8.7. *The opposite of the ℓ_1-norm is prox-bounded with threshold $+\infty$. Moreover, its proximity operator, for any $\gamma > 0$, at a point $x \in \mathbb{R}^n$ is given component-wise, for $i \in \{1, \ldots, n\}$, by*

$$\operatorname{prox}_{\gamma(-\|\cdot\|_1)}(x)_i = \begin{cases} \{\gamma, -\gamma\}, & \text{if } x_i = 0, \\ x_i + \operatorname{sign}(x_i)\,\gamma, & \text{otherwise.} \end{cases}$$

8.1.2 The family of upper-$\mathcal{C}^2$ functions

The class of functions with a Lipschitz continuous gradient is very important in optimization. Its relevance relies on the fact that these functions verify the so-called *descent lemma* (see, e.g., [149, Lemma A.11]). Namely, given a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ with

L_f-Lipschitz continuous gradient, then the following inequality holds

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L_f}{2}\|y - x\|^2, \quad \text{for all } x, y \in \mathbb{R}^n. \tag{8.5}$$

Different authors have identified a larger family of functions that also satisfies an inequality similar to (8.5), preserving thus the same nice properties for optimization (see, e.g., [105, Theorem 5.1] and [218, Definition 10.29]). We extend our analysis to this broader class of functions, which we present next.

Definition 8.8. *Let $V \subseteq \mathbb{R}^n$ be an open, convex and bounded set and let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be Lipschitz continuous on V. We say that f is κ-upper-$\mathcal{C}^2$ on V for some $\kappa \geq 0$ if there exist a compact set S (in some topological space) and some continuous functions $b : S \to \mathbb{R}^n$ and $c : S \to \mathbb{R}$ such that*

$$f(x) = \min_{s \in S} \left\{ \kappa\|x\|^2 - \langle b(s), x \rangle - c(s) \right\}, \quad \textit{for all } x \in V. \tag{8.6}$$

From (8.6) it directly follows that κ-upper-$\mathcal{C}^2$ functions are DC (difference of convex) functions with the specific DC decomposition

$$f(x) = \kappa\|x\|^2 - \max_{s \in S} \left\{ \langle b(s), x \rangle + c(s) \right\}, \tag{8.7}$$

since $x \mapsto \max_{s \in S} \left\{ \langle b(s), x \rangle + c(s) \right\}$ is a convex continuous function.

The following result, based on [105, Theorem 5.1], establishes the relationship between upper-$\mathcal{C}^2$ functions and the descent lemma. This permits to relax the Lipschitz continuous gradient assumption on f in [38] to nonsmooth functions while preserving the useful inequality.

Proposition 8.9. *Let U be an open and convex set such that $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is locally Lipschitz on U. Then the following assertions are equivalent for a parameter $\kappa \geq 0$:*

(i) f is κ-upper-$\mathcal{C}^2$ on every open bounded subset of U;

(ii) for all $x \in U$ and $\xi \in \operatorname{co} \partial f(x)$, it holds

$$f(y) \leq f(x) + \langle \xi, y - x \rangle + \kappa\|y - x\|^2, \quad \textit{for all } y \in U; \tag{8.8}$$

(iii) for each $x \in U$, there exits $\xi \in \mathbb{R}^n$ such that (8.8) holds.

Proof. The case $\kappa = 0$ can be straightforwardly established. Therefore, for the remainder of the proof, we assume $\kappa > 0$.

(i)⇒(ii) Let $x, y \in U$ and $\xi \in \operatorname{co} \partial f(x)$. Let V be an open, convex and bounded subset of U that contains both x and y. By [105, Theorem 5.1 (a)⇒(c) & Theorem 5.2] applied to $-f$ and V, it holds

$$f(w) \leq f(x) + \langle -\zeta, w - x \rangle + \kappa \|w - x\|^2, \quad \forall w \in V, \tag{8.9}$$

for all $\zeta \in \partial(-f)(x) = \hat{\partial}(-f)(x)$. By convexity of the regular subdifferential and (8.1),

$$\partial(-f)(x) = \operatorname{co} \partial(-f)(x) = -\operatorname{co} \partial f(x). \tag{8.10}$$

Since $-\xi \in -\operatorname{co} \partial f(x) = \partial(-f)(x)$, we can conclude that (8.9) holds for $\zeta := -\xi$ and $w := y \in V$, which implies (8.8).

(ii)⇒(iii) This is straightforward: $\partial f(x) \neq \emptyset$ since f is locally Lipschitz around x (see, e.g., [183, Theorem 1.22]).

(iii)⇒(i) If V is an open, convex and bounded subset of U, then (8.8) holds on V, so [105, Theorem 5.1 (b)⇒(a)] implies that f is κ-upper-$\mathcal{C}^2$ on V. This completes the proof. □

We conclude this section by providing some examples of upper-$\mathcal{C}^2$ functions, which are motivated by our subsequent numerical applications. The first such example was considered in [138, Lemma 5.2]. A local variation of this result can be found in [28, Lemma 3.6].

Example 8.10 (Difference of an L-smooth function and a convex function). *First, notice that by (8.7) any κ-upper-$\mathcal{C}^2$ function can be expressed as a difference of a smooth convex function and a continuous convex function. Now, let $f_1 : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function whose gradient is L_{f_1}-Lipschitz continuous and let $f_2 : \mathbb{R}^n \to \mathbb{R}$ be a l.s.c. convex function. Then the function $f := f_1 - f_2$ is $L_{f_1}/2$-upper-$\mathcal{C}^2$ on every bounded set of $\mathbb{R}^n$. Indeed, observe first that (8.5) holds for f_1. Moreover, for any $v \in \partial f_2(x)$, by definition of the convex subdifferential, we have*

$$-f_2(y) \leq -f_2(x) + \langle -v, y - x \rangle, \quad \text{for all } y \in \mathbb{R}^n.$$

Adding together (8.5) *for* f_1 *and the above inequality, we get that*

$$f(y) \leq f(x) + \langle \nabla f_1(x) - v, y - x \rangle + \frac{L_{f_1}}{2}\|y - x\|^2.$$

Hence, the assertion follows by Proposition 8.9.

Example 8.11 (Clustering problems). *An interesting problem where the collection of* κ*-upper-*$\mathcal{C}^2$ *functions naturally appears is the* minimum sum-of-squares clustering problem. *The task is to minimize an objective function of the form*

$$f(X) := \frac{1}{q}\sum_{i=1}^{q} \omega_i(X), \qquad \text{with} \qquad \omega_i(X) := \min\left\{\|x^j - a^i\|^2 : j = 1, \ldots, \ell\right\},$$

where $X := (x^1, \ldots, x^\ell) \in \mathbb{R}^{s\times\ell}$*, and* $\{a^1, \ldots, a^q\}$ *represents a set of data points. It is important to emphasize that the function* f *is nonsmooth. Therefore, the classical descent lemma cannot be applied. Nevertheless,* f *is* 1*-upper-*$\mathcal{C}^2$ *since each of the functions* ω_i *is* 1*-upper-*$\mathcal{C}^2$ *(simply by definition). According to Proposition 8.9, the function* f *satisfies the subgradient descent inequality* (8.8).

Example 8.12 (Squared distance to a nonconvex set). *Consider a nonempty closed set* $C \subseteq \mathbb{R}^s$ *and a matrix* $Q \in \mathbb{R}^{s\times n}$*. Then the mapping* $x \to \frac{1}{2}d^2(Qx, C)$ *is upper-*C^2 *on* $\mathbb{R}^n$*. Indeed, let us notice that*

$$\frac{1}{2}d^2(Qx, C) = \frac{1}{2}\|Qx\|^2 - \mathtt{A}_C(Qx), \tag{8.11}$$

where $\mathtt{A}_C$ *is the* Asplund function *associated to the set* C *given by*

$$\mathtt{A}_C(w) := \sup_{y\in C}\left\{\langle y, w\rangle - \frac{1}{2}\|y\|^2\right\} = \left(\iota_C + \frac{1}{2}\|\cdot\|^2\right)^*(w).$$

By representation (8.11) *and Example 8.10, we have that* $x \mapsto \frac{1}{2}d^2(Qx, C)$ *is* $\frac{\|Q\|^2}{2}$*-upper-*$\mathcal{C}^2$ *on* $\mathbb{R}^n$*.*

Remark 8.13. In view of (8.7), it is always possible to decompose a κ-upper-$\mathcal{C}^2$ function f as $f = f_1 - f_2$, with f_1 smooth and convex and f_2 continuous and convex. However, considering this DC decomposition may not be desirable in practice as it could enlarge the set of critical points of the problem (see equation (8.18)). This is essentially due to the fact that the basic subdifferential is not homogeneous with respect to the minus sign, which causes the equality

$$\partial f(x) = \partial f_1(x) - \partial f_2(x) \tag{8.12}$$

to not necessarily hold at every point x. For instance, let $Q = \mathrm{Id}$ in Example 8.12 (for simplicity of the explanation), and consider the DC decomposition of $f(x) = \frac{1}{2}d(x,C)^2$ given in (8.11). By [28, Theorem 5.3 & Proposition 2.7] one has that

$$\partial f(x) = x - P_C(x) \text{ while } \partial \mathtt{A}_C(x) = \operatorname{co} P_C(x),$$

where P_C stands for the (set-valued) projection operator onto C. Therefore, if the set C is nonconvex, the right hand side of (8.12) might be a larger set, resulting in the strict inclusion $\partial f(x) \subsetneq \partial f_1(x) - \partial f_2(x)$.

8.2 The Boosted Double-proximal Subgradient Algorithm

In this section, we design an algorithm for solving the nonconvex optimization problem ($\mathcal{P}$). The algorithm utilizes subgradients of f, the gradients of Ψ_i, and proximal steps of g and h_i^*, for $i \in [\![1,p]\!]$. Additionally, it incorporates a linesearch step, leading us to name it the *Boosted Double-proximal Subgradient Algorithm* (in short BDSA).

Le us define the function $\varphi : \mathbb{R}^n \to \overline{\mathbb{R}}$ as follows:

$$\varphi(x) := f(x) + g(x) - \sum_{i=1}^{p} h_i(\Psi_i(x)). \tag{8.13}$$

The following assumptions are made throughout the rest of this chapter.

Assumption 8.14. Let $U \subseteq \mathbb{R}^n$ be an open convex set such that $\operatorname{dom} g \subseteq U$. Suppose that $\inf_{x\in\mathbb{R}^n} \varphi(x) > -\infty$ and that the functions in ($\mathcal{P}$) satisfy:

(i) $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is locally Lipschitz on U and κ-upper-$\mathcal{C}^2$ on every open bounded subset of U;

(ii) $g : \mathbb{R}^n \to \overline{\mathbb{R}}$ is proper, l.s.c. and prox-bounded for some $\gamma^g > 0$;

(iii) $h_i : \mathbb{R}^{m_i} \to \mathbb{R}$ are convex and continuous functions for all $i \in [\![1,p]\!]$;

(iv) $\Psi_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ are differentiable functions with L_i-Lipschitz continuous gradients on U for all $i \in [\![1,p]\!]$.

Remark 8.15. Despite the fact that the class of upper-$\mathcal{C}^2$ functions is large, as demonstrated by the examples presented in Section 8.1.2, it is worth noting that, in general, one cannot subsume the function $-\sum_{i=1}^{p} h_i(\Psi_i(x))$ into the function f in (8.13), since

$-h_i(\Psi_i(x))$ may not belong to the upper-$\mathcal{C}^2$ family (e.g., if $h_i(y) = y^2$ and $\Psi_i(x) = x^2$, then $-h_i(\Psi_i(x)) = -x^4$ does not satisfy (8.8) on $U = \mathbb{R}$ for any fixed $\kappa \geq 0$).

Instead of directly addressing problem ($\mathcal{P}$), which consists in the minimization of the function φ in (8.13), we consider the primal-dual formulation

$$\min_{(x,\mathbf{y})\in\mathbb{R}^n\times\mathbb{R}^m} \Phi(x,\mathbf{y}), \tag{$\mathcal{PD}$}$$

where $\Phi : \mathbb{R}^n \times \mathbb{R}^m \to \overline{\mathbb{R}}$ is given by

$$\Phi(x,\mathbf{y}) := f(x) + g(x) + \sum_{i=1}^{p} \left(h_i^*(y_i) - \langle \Psi_i(x), y_i\rangle\right), \tag{8.14}$$

with $\mathbf{y} = (y_1, \ldots, y_p) \in \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_p} = \mathbb{R}^m$. It is easy to check that the optimal values of both problems coincide, that is,

$$\inf_{x\in\mathbb{R}^n} \varphi(x) = \inf_{(x,\mathbf{y})\in\mathbb{R}^n\times\mathbb{R}^m} \Phi(x,\mathbf{y}). \tag{8.15}$$

Indeed, by the Fenchel–Moreau theorem,

$$\begin{aligned}
\inf_{x\in\mathbb{R}^n} \varphi(x) &= \inf_{x\in\mathbb{R}^n} \left\{ f(x) + g(x) - \sum_{i=1}^{p} h_i(\Psi_i(x)) \right\} \\
&= \inf_{x\in\mathbb{R}^n} \left\{ f(x) + g(x) - \sum_{i=1}^{p} \sup_{y_i\in\mathbb{R}^{m_i}} \{\langle \Psi_i(x), y_i\rangle - h_i^*(y_i)\} \right\} \\
&= \inf_{x\in\mathbb{R}^n} \inf_{\mathbf{y}\in\mathbb{R}^m} \left\{ f(x) + g(x) + \sum_{i=1}^{p} \left(h_i^*(y_i) - \langle \Psi_i(x), y_i\rangle\right) \right\} \\
&= \inf_{(x,\mathbf{y})\in\mathbb{R}^n\times\mathbb{R}^m} \Phi(x,\mathbf{y}).
\end{aligned}$$

By the generalized Fermat rule, a necessary condition for a point $(\bar{x}, \bar{\mathbf{y}}) \in \mathbb{R}^n \times \mathbb{R}^m$ to be a local minimum of Φ is that $0 \in \partial\Phi(\bar{x}, \bar{\mathbf{y}})$. Observe that we can express $\Phi = \Phi_1 + \Phi_2$, with $\Phi_1(x,\mathbf{y}) := f(x) + g(x) + \sum_{i=1}^{p} h_i^*(y_i)$ and $\Phi_2(x,\mathbf{y}) := \sum_{i=1}^{p} \langle \Psi_i(x), y_i\rangle$. Since Φ_2 is $\mathcal{C}^1$, by the sum rule in Proposition 8.4 (i),

$$\begin{aligned}
\partial\Phi(x,\mathbf{y}) &= \partial\Phi_1(x,\mathbf{y}) - \nabla\Phi_2(x,\mathbf{y}) \\
&= \partial(f+g)(x) \times \left(\bigtimes_{i=1}^{p} \partial h_i^*(y_i) \right) - \left(\sum_{i=1}^{p} \nabla\Psi_i(x) y_i, \Psi_1(x), \ldots, \Psi_p(x) \right),
\end{aligned} \tag{8.16}$$

where the second equality holds because Φ_1 has separate variables. Therefore, the necessary condition $0 \in \partial\Phi(\bar{x}, \bar{\mathbf{y}})$ is equivalent to

$$\begin{cases} \sum_{i=1}^{p} \nabla\Psi_i(\bar{x})\bar{y}_i \in \partial(f+g)(\bar{x}), \\ \Psi_i(\bar{x}) \in \partial h_i^*(\bar{y}_i), \quad \forall i \in [\![1,p]\!]. \end{cases} \tag{8.17}$$

By Proposition 8.4 (ii), we have $\partial(f+g)(\bar{x}) \subseteq \partial f(\bar{x}) + \partial g(\bar{x})$. Thus, the inclusions (8.17) imply

$$\begin{cases} \sum_{i=1}^{p} \nabla\Psi_i(\bar{x})\bar{y}_i \in \partial f(\bar{x}) + \partial g(\bar{x}), \\ \bar{y}_i \in \partial h_i\big(\Psi_i(\bar{x})\big), \quad \forall i \in [\![1,p]\!]. \end{cases} \tag{8.18}$$

We refer to a point $(\bar{x}, \bar{\mathbf{y}}) \in \mathbb{R}^n \times \mathbb{R}^m$ that satisfies (8.18) as a *critical point of* $(\mathcal{PD})$.

On the other hand, given a point $\bar{x} \in \operatorname{dom}\varphi$, it is well-known (see, e.g., [6, 119, 120]) that the subdifferential inclusion

$$\partial\left(\sum_{i=1}^{p} h_i \circ \Psi_i\right)(\bar{x}) \subset \partial(f+g)(\bar{x}). \tag{8.19}$$

constitutes a necessary condition for local optimality of problem $(\mathcal{P})$. A point verifying (8.19) is called a *stationary point* of $(\mathcal{P})$. In general, finding stationary points is highly challenging, so it is useful to consider relaxed notions. We say that a point $\bar{x} \in \operatorname{dom}\varphi$ is a *critical point* of $(\mathcal{P})$ if there exists $\bar{\mathbf{y}} \in \mathbb{R}^m$ such that (8.18) holds. By [218, Theorem 10.6] and [183, Corollary 2.21], it easily follows that every stationary point of $(\mathcal{P})$ is a critical point, but the converse is not true in general.

Therefore, if $(\bar{x}, \bar{\mathbf{y}}) \in \mathbb{R}^n \times \mathbb{R}^m$ is a critical point of $(\mathcal{PD})$, then $\bar{x}$ is a critical point of $(\mathcal{P})$. Conversely, if $\bar{x} \in \mathbb{R}^n$ is a critical point of $(\mathcal{P})$, then there exists $\bar{\mathbf{y}} \in \mathbb{R}^m$ such that $(\bar{x}, \bar{\mathbf{y}})$ is a critical point of $(\mathcal{PD})$. The next result establishes further connections between critical points and solutions of the two minimization problems $(\mathcal{P})$ and $(\mathcal{PD})$ presented above.

Proposition 8.16. *Let $(\bar{x}, \bar{\mathbf{y}}) \in \mathbb{R}^n \times \mathbb{R}^m$. Then the following claims hold.*

(i) If $(\bar{x}, \bar{\mathbf{y}})$ is a critical point of $(\mathcal{PD})$, then $\Phi(\bar{x}, \bar{\mathbf{y}}) = \varphi(\bar{x})$.

(ii) If $(\bar{x}, \bar{\mathbf{y}})$ is a solution of $(\mathcal{PD})$, then $\bar{x}$ is a solution of $(\mathcal{P})$.

(iii) If $\bar{x}$ is a solution of $(\mathcal{P})$, then there exists $\bar{\mathbf{y}} \in \mathbb{R}^m$ such that $(\bar{x}, \bar{\mathbf{y}})$ is a solution of $(\mathcal{PD})$.

Proof. (i) Let $(\bar{x}, \bar{\mathbf{y}}) = (\bar{x}, \bar{y}_1, \ldots, \bar{y}_p)$ be a critical point of $(\mathcal{PD})$. This implies that $\bar{y}_i \in \partial h_i(\Psi_i(\bar{x}))$ for all $i \in [\![1, p]\!]$, so for these points the Fenchel–Young inequality in Proposition 2.29 becomes an equality, and we obtain the following expressions

$$h_i(\Psi_i(\bar{x})) + h_i^*(\bar{y}_i) = \langle \Psi_i(\bar{x}), \bar{y}_i \rangle, \quad \forall i \in [\![1, p]\!]. \tag{8.20}$$

This yields the equality

$$\begin{aligned} \varphi(\bar{x}) &= f(\bar{x}) + g(\bar{x}) - \sum_{i=1}^{p} h_i(\Psi_i(\bar{x})) \\ &= f(\bar{x}) + g(\bar{x}) + \sum_{i=1}^{p} (h_i^*(\bar{y}_i) - \langle \Psi_i(\bar{x}), \bar{y}_i \rangle) = \Phi(\bar{x}, \bar{y}_1, \ldots, \bar{y}_p). \end{aligned}$$

(ii) If $(\bar{x}, \bar{\mathbf{y}})$ is a solution of $(\mathcal{PD})$, as argued above, it must be a critical point. Then, by (i), we have that $\Phi(\bar{x}, \bar{\mathbf{y}}) = \varphi(\bar{x})$. Since the optimal values of problems $(\mathcal{P})$ and $(\mathcal{PD})$ coincide (recall (8.15)), the above expression implies that $\bar{x}$ is a solution to $(\mathcal{P})$.

(iii) Finally, let us suppose that $\bar{x}$ is a solution of $(\mathcal{P})$. Then $\bar{x}$ is necessarily a critical point of $(\mathcal{P})$, and thus there exists $\bar{y}_i \in \partial h_i(\Psi_i(\bar{x}))$ such that (8.20) holds. Once more, this implies that $\varphi(\bar{x}) = \Phi(\bar{x}, \bar{\mathbf{y}})$, where $\bar{\mathbf{y}} := (\bar{y}_1, \ldots, \bar{y}_p)$. Therefore, using again the fact that optimal values of problems $(\mathcal{P})$ and $(\mathcal{PD})$ coincide, we get that $(\bar{x}, \bar{\mathbf{y}})$ is a solution of problem $(\mathcal{PD})$. □

Now, we present in Algorithm 8.17 the pseudo-code of the *Boosted Double-proximal Subgradient Algorithm*. Steps 2 and 3 are motivated by the *Double-proximal Gradient Algorithm* proposed by Banert and Boţ [38], which can be derived as an application of Bolte–Sabach–Teboulle's *Proximal Alternating Linearized Minimization* [66] to the primal-dual formulation $(\mathcal{PD})$ when f is L-smooth, g is convex and the Ψ_i's are linear. In addition, Steps 4-8 in BDSA allow an optional linesearch procedure that permits to enhance the performance of the method.

Algorithm 8.17 Boosted Double-proximal Subgradient Algorithm for problem $(\mathcal{P})$.

Require: $(x^0, \mathbf{y}^0) = (x^0, y_1^0, \ldots, y_p^0) \in \mathbb{R}^n \times \mathbb{R}^m$, $R \geq 0$, $\rho \in]0,1[$ and $\alpha \geq 0$. Set $k := 0$.

1: Choose $v^k \in \partial f(x^k)$.

2: Take some positive $\gamma_k < \min\left\{\gamma^g, \left(2\kappa + \sum_{i=1}^p L_i \left\|y_i^k\right\|\right)^{-1}\right\}$ and compute

$$\hat{x}^k \in \operatorname{prox}_{\gamma_k g}\left(x^k + \gamma_k \sum_{i=1}^p \nabla\Psi_i(x^k) y_i^k - \gamma_k v^k\right). \tag{8.21}$$

3: For each $i \in [\![1,p]\!]$, take $\mu_i^k > 0$ and compute

$$\hat{y}_i^k = \operatorname{prox}_{\mu_i^k h_i^*}\left(y_i^k + \mu_i^k \Psi_i(\hat{x}^k)\right). \tag{8.22}$$

4: Choose any $\overline{\lambda}_k \geq 0$. Set $\lambda_k := \overline{\lambda}_k$, $r := 0$ and $(d^k, \mathbf{e}^k) := (\hat{x}^k, \hat{\mathbf{y}}^k) - (x^k, \mathbf{y}^k)$.

5: **if** $(d^k, \mathbf{e}^k) = 0$ **then** STOP and return x^k.

6: **while** $r < R$ and

$$\Phi\big((\hat{x}^k, \hat{\mathbf{y}}^k) + \lambda_k (d^k, \mathbf{e}^k)\big) > \Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \alpha\lambda_k^2 \|(d^k, \mathbf{e}^k)\|^2 \tag{8.23}$$

do $r := r + 1$ and $\lambda_k := \rho^r \overline{\lambda}_k$.

7: **if** $r = R$ **then** $\lambda_k := 0$.

8: Set $(x^{k+1}, \mathbf{y}^{k+1}) := (\hat{x}^k, \hat{\mathbf{y}}^k) + \lambda_k (d^k, \mathbf{e}^k)$, $k := k + 1$ and go to Step 1.

Remark 8.18. From the definition of the proximity operator, (8.21) and (8.22) imply that the sequences generated by Algorithm 8.17 verify

$$\begin{aligned} &\frac{x^k - \hat{x}^k}{\gamma_k} + \sum_{i=1}^p \nabla\Psi_i(x^k) y_i^k - v^k \in \partial g(\hat{x}^k), \\ &\frac{y_i^k - \hat{y}_i^k}{\mu_i^k} + \Psi_i(\hat{x}^k) \in \partial h_i^*(\hat{y}_i^k), \quad \forall i \in [\![1,p]\!]. \end{aligned} \tag{8.24}$$

In particular, if $(d^k, \mathbf{e}^k) = 0$, the above inclusions become (8.18) for $(\bar{x}, \bar{\mathbf{y}}) := (x^k, \mathbf{y}^k)$ and hence $\bar{x}$ is a critical point of problem $(\mathcal{P})$.

Remark 8.19. It is possible to replace $v^k \in \partial f(x^k)$ by $v^k \in \operatorname{co} \partial f(x^k)$ in Algorithm 8.17. This is justified by Proposition 8.9, which shows that such subgradients satisfy the descent inequality (8.8). This straightforward modification in Step 1 of Algorithm 8.17 can be advantageous in numerical applications, particularly when the calculus rules for the basic subdifferential only provide an upper estimate rather than an equality. For example, if the objective function $\varphi(x)$ contains a term of the form $-\upsilon(x)$ and $\upsilon : \mathbb{R}^n \to \overline{\mathbb{R}}$ is convex, it is natural to set $f := -\upsilon$, which is upper-$\mathcal{C}^2$ by Example 8.10. Algorithm 8.17 requires

choosing $v^k \in \partial f(x^k) = \partial(-\upsilon)(x^k) \subset -\partial\upsilon(x^k)$ (see [218, Corollary 9.21]), while the modification $v^k \in \operatorname{co}\partial f(x^k) = -\partial\upsilon(x^k)$ (see (8.10)) allows choosing v^k from the possibly larger set $-\partial\upsilon(x^k)$.

On the other hand, according to (8.24), allowing $v^k \in \operatorname{co}\partial f(x^k) = \partial_C f(x^k)$ in Step 1 entails the notion of criticality

$$\begin{cases} \displaystyle\sum_{i=1}^{p} \nabla\Psi_i(\bar{x})\bar{y}_i \in \partial_C f(\bar{x}) + \partial g(\bar{x}), \\ \bar{y}_i \in \partial h_i\big(\Psi_i(\bar{x})\big), \quad \forall i \in [\![1,p]\!], \end{cases}$$

when $(d^k, \mathbf{e}^k) = 0$ and $(\bar{x}, \bar{\mathbf{y}}) := (x^k, \mathbf{y}^k)$, which in view of Remark 8.3 is weaker than the one in (8.18). Thus, the price to pay for having more freedom in the choice of v^k is the possibility of having a larger set of non-optimal critical points to which the algorithm might converge. Nonetheless, a more subtle analysis can be performed to derive stronger notions of criticality, see Remark 8.26 and Section 8.3.2 for a numerical example.

Remark 8.20 (Particular cases of Algorithm 8.17). Different known algorithms can be obtained as particular cases of Algorithm 8.17.

(i) Consider the problem of minimizing $\varphi(x) := g_1(x) - g_2(x)$, with g_1 and g_2 being convex. If we let $f := -g_2$, $g := g_1$ and $\Psi_i := 0 =: h_i$ for all $i \in [\![1,p]\!]$, then Step 1 of Algorithm 8.17 becomes $v^k \in \partial(-g_2)(x^k)$. Since $\partial(-g_2)(x^k) \subseteq -\partial g_2(x^k)$, when $R = 0$ one recovers a specific choice for the *Proximal DC Algorithm* of [229] in which v^k is chosen from a smaller set of subgradients. If $x^{k+1} = x^k =: \bar{x}$, one gets $-v^k \in \partial g_1(\bar{x}) \cap (-\partial(-g_2)(\bar{x}))$. Observe that the more restrictive condition $\partial g_1(\bar{x}) \cap (-\partial(-g_2)(\bar{x})) \neq \emptyset$ can serve to discard some points which are not local minima (e.g., if $g_1(x) = x^2$ and $g_2(x) = |x|$, then $\bar{x} := 0$ is a local maximum which satisfies $0 \in \nabla g_1(\bar{x}) \cap \partial g_2(\bar{x})$, but $0 \notin \nabla g_1(\bar{x}) \cap (-\partial(-g_2)(\bar{x}))$. When taken $R = \infty$, one obtains the *Boosted Proximal DC Algorithm* introduced in [4], which is a modification of the *Boosted Difference of Convex functions Algorithm* from [24, 27] that adds a proximal term.

(ii) Likewise, if $\varphi(x) := f_1(x) - f_2(x) + g(x)$ with g being proper and l.s.c. with $\inf_{x\in\mathbb{R}^n} g(x) > -\infty$, f_2 being convex and f_1 being L-smooth, letting $f := f_1 - f_2$ and the rest of the functions as in the previous case with $R = 0$, we obtain a specific choice for the *Generalized Proximal Point Algorithm* presented in [7].

(iii) The *Double-Proximal Gradient Algorithm* proposed in [38] is also recovered by Algorithm 8.17 in the case in which f is convex and L-smooth, g is convex, $R = 0$ and Ψ_i is a linear operator, for all $i \in [\![1, p]\!]$. Also note that p is set to 1 in [38], but this is sufficient to address the case $p > 1$ by considering the choice of functions $\Psi : \mathbb{R}^n \to \mathbb{R}^m : x \mapsto (\Psi_1(x), \ldots, \Psi_p(x))$ and $h : \mathbb{R}^m \to \mathbb{R} : \mathbf{y} \mapsto \sum_{i=1}^{p} h_i(y_i)$. The larger upper bound on γ_k in the Double-Proximal Gradient Algorithm can also be obtained for BDSA when g is assumed to be convex (see Remark 8.23 below).

For the reader's convenience, we collect in Table 8.1 the main characteristics of the above-mentioned algorithms.

	f	g	h_i	Ψ_i	γ_k bound
Proximal DC [229]	$-f$ l.s.c. convex	l.s.c. convex	0	0	$+\infty$
Generalized Proximal-Point [7]	f_1-f_2 f_1 L-smooth f_2 convex	l.s.c. $\inf_{x\in\mathbb{R}^n} g(x) > -\infty$	0	0	$\frac{1}{L}$
Double-Proximal Gradient [38]	convex L-smooth	l.s.c. convex	convex continuous	linear	$\frac{2}{L}$

TABLE 8.1: Summary of particular cases of Algorithm 8.17.

Steps 6-7 of Algorithm 8.17 correspond to an optional linesearch step in the direction $(d^k, \mathbf{e}^k)$ with a fixed number R of attempts. On the one hand, note that the computational burden of these steps can be avoided if the user either chooses $R = 0$ or $\overline{\lambda}_k = 0$, in which case we refer to the resulting algorithm as *Double-proximal Subgradient Algorithm* (abbr. *DSA*). On the other hand, if $R > 0$ and $\overline{\lambda}_k > 0$, Step 6 allows to achieve a further decrease of the primal-dual objective function Φ. Step 7 sets $\lambda_k = 0$ when the linesearch was not successful.

For general problems satisfying Assumption 8.14 there is no guarantee that the vector $(d^k, \mathbf{e}^k)$ defined in Step 4 of BDSA provides a descent direction for the function Φ. The motivation for the linesearch step comes from the case in which the functions g and h_i^* are differentiable at the points $\hat{x}^k$ and $\hat{y}_i^k$, respectively. In this case, $(d^k, \mathbf{e}^k)$ is a descent direction for Φ at $(\hat{x}^k, \hat{\mathbf{y}}^k)$, since the upper Dini directional derivative of Φ at $(\hat{x}^k, \hat{\mathbf{y}}^k)$ in the direction $(d^k, \mathbf{e}^k)$ is negative. This fact is proved in the next result.

Proposition 8.21. *Suppose that Assumption 8.14 holds and consider the sequences generated by Algorithm 8.17 for problem $(\mathcal{P})$. Assume also that*

(i) g is differentiable at $\hat{x}^k$;

(ii) h_i^* *is differentiable at* $\hat{y}_i{}^k$ *for all* $i \in [\![1,p]\!]$;

(iii) $\gamma_k \in \left]0, \left(2\kappa + \frac{p}{2} + \sum_{i=1}^p L_i\|y_i^k\|\right)^{-1}\right[$ *and* $\mu_i^k \in \left]0, 2\,\|\nabla\Psi_i(\hat{x}^k)\|^{-2}\right[$ *for all* $i \in [\![1,p]\!]$.

Then, for all $k \geq 0$,

$$d^+\Phi\big((\hat{x}^k,\hat{\mathbf{y}}^k);(d^k,\mathbf{e}^k)\big) \leq \left(2\kappa + \frac{p}{2} + \sum_{i=1}^p L_i\|y_i^k\| - \frac{1}{\gamma_k}\right)\|d^k\|^2 + \sum_{i=1}^p\left(\frac{1}{2}\|\nabla\Psi_i(\hat{x}^k)\|^2 - \frac{1}{\mu_i^k}\right)\|e_i^k\|^2. \tag{8.25}$$

Consequently, if $(d^k,\mathbf{e}^k) \neq 0$, *then for every* $\alpha > 0$ *there is some* $\delta_k > 0$ *such that*

$$\Phi\big((\hat{x}^k,\hat{\mathbf{y}}^k) + \lambda(d^k,\mathbf{e}^k)\big) \leq \Phi(\hat{x}^k,\hat{\mathbf{y}}^k) - \alpha\lambda^2\|(d^k,\mathbf{e}^k)\|^2, \quad \text{for all } \lambda \in [0,\delta_k]. \tag{8.26}$$

Proof. Indeed, for any $\hat{v}^k \in \partial f(\hat{x}^k)$, we get

$$\begin{aligned} d^+\Phi\big((\hat{x}^k,\hat{\mathbf{y}}^k);(d^k,\mathbf{e}^k)\big) &\leq \limsup_{t\downarrow 0}\frac{f(\hat{x}^k+td^k)-f(\hat{x}^k)}{t} + \limsup_{t\downarrow 0}\frac{g(\hat{x}^k+td^k)-g(\hat{x}^k)}{t} \\ &\quad + \sum_{i=1}^p \limsup_{t\downarrow 0}\frac{h_i^*(\hat{y}_i^k+te_i^k)-h_i^*(\hat{y}_i^k)}{t} \\ &\quad - \sum_{i=1}^p \liminf_{t\downarrow 0}\frac{\langle\Psi_i(\hat{x}^k+td^k)-\Psi_i(\hat{x}^k),\hat{y}_i^k\rangle + t\langle\Psi_i(\hat{x}^k+td^k),e_i^k\rangle}{t} \\ &\leq \langle\hat{v}^k,d^k\rangle + \langle\nabla g(\hat{x}^k),d^k\rangle + \sum_{i=1}^p\langle\nabla h_i^*(\hat{y}_i^k),e_i^k\rangle \\ &\quad - \sum_{i=1}^p\Big(\langle\nabla\Psi_i(\hat{x}^k)\hat{y}_i^k,d^k\rangle + \langle\Psi_i(\hat{x}^k),e_i^k\rangle\Big), \end{aligned} \tag{8.27}$$

where the second inequality is due to Proposition 8.9. Now, since g and h_i^* are assumed to be differentiable at $\hat{x}^k$ and $\hat{y}_i^k$, respectively, (8.24) yields

$$\begin{aligned} &\frac{x^k-\hat{x}^k}{\gamma_k} + \sum_{i=1}^p\nabla\Psi_i(x^k)y_i^k - v^k = \nabla g(\hat{x}^k), \\ &\frac{y_i^k-\hat{y}_i^k}{\mu_i^k} + \Psi_i(\hat{x}^k) = \nabla h_i^*(\hat{y}_i^k), \quad \forall i \in [\![1,\ldots,p]\!]. \end{aligned} \tag{8.28}$$

On the other hand, again making use of Proposition 8.9, we get the following inequality by setting $y := \hat{x}^k$, $x := x^k$ and $\xi := v^k$ in equation (8.8)

$$f(\hat{x}^k) - f(x^k) - \langle v^k,\hat{x}^k-x^k\rangle \leq \kappa\|\hat{x}^k-x^k\|^2.$$

Likewise, setting $y := x^k$, $x := \hat{x}^k$ and $\xi := \hat{v}^k$ in (8.8) yields

$$f(x^k) - f(\hat{x}^k) + \langle \hat{v}^k, \hat{x}^k - x^k \rangle \leq \kappa \|\hat{x}^k - x^k\|^2.$$

Summing together these two equations, we get

$$\langle \hat{v}^k - v^k, \hat{x}^k - x^k \rangle \leq 2\kappa \|\hat{x}^k - x^k\|^2, \quad \forall v^k \in \partial f(x^k). \tag{8.29}$$

Substituting (8.28) and (8.29) in (8.27), it becomes

$$\begin{aligned} d^+\Phi\big((\hat{x}^k, \hat{\mathbf{y}}^k); (d^k, \mathbf{e}^k)\big) &\leq \langle \hat{v}^k - v^k, d^k \rangle - \frac{1}{\gamma_k}\langle \hat{x}^k - x^k, d^k \rangle + \sum_{i=1}^{p} \langle \nabla \Psi_i(x^k) y_i^k, d^k \rangle \\ &\quad - \sum_{i=1}^{p} \frac{1}{\mu_i^k} \langle \hat{y}_i^k - y_i^k, e_i^k \rangle + \sum_{i=1}^{p} \langle \Psi_i(\hat{x}^k), e_i^k \rangle \\ &\quad - \sum_{i=1}^{p} \big(\langle \nabla \Psi_i(\hat{x}^k) \hat{y}_i^k, d^k \rangle + \langle \Psi_i(\hat{x}^k), e_i^k \rangle \big) \\ &\leq \left(2\kappa - \frac{1}{\gamma_k} \right) \|d^k\|^2 - \sum_{i=1}^{p} \frac{1}{\mu_i^k} \|e_i^k\|^2 \\ &\quad - \sum_{i=1}^{p} \langle \nabla \Psi_i(\hat{x}^k) \hat{y}_i^k - \nabla \Psi_i(x^k) y_i^k, d^k \rangle. \end{aligned} \tag{8.30}$$

Finally, using the Cauchy–Schwartz inequality and Young's inequality, the terms in the last summation can be upper bounded as

$$\begin{aligned} -\langle \nabla \Psi_i(\hat{x}^k) \hat{y}_i^k - \nabla \Psi_i(x^k) y_i^k, d^k \rangle &= -\langle \nabla \Psi_i(\hat{x}^k) e_i^k, d^k \rangle + \langle (\nabla \Psi_i(x^k) - \nabla \Psi_i(\hat{x}^k)) y_i^k, d^k \rangle \\ &\leq \|\nabla \Psi_i(\hat{x}^k)\| \|e_i^k\| \|d^k\| + L_i \|y_i^k\| \|d^k\|^2 \\ &\leq \frac{1}{2} \left(\|\nabla \Psi_i(\hat{x}^k)\|^2 \|e_i^k\|^2 + \|d^k\|^2 \right) + L_i \|y_i^k\| \|d^k\|^2 \\ &= \frac{1}{2} \|\nabla \Psi_i(\hat{x}^k)\|^2 \|e_i^k\|^2 + \left(\frac{1}{2} + L_i \|y_i^k\| \right) \|d^k\|^2, \end{aligned}$$

for all $i \in [\![1, p]\!]$. Putting this into (8.30), we deduce (8.25).

Thanks to assumption (iii), we have

$$K := \min_{i=1,\ldots,p} \left\{ \frac{1}{\gamma_k} - 2\kappa + \frac{p}{2} - \sum_{j=1}^{p} L_j \|y_j^k\|, \frac{1}{\mu_i^k} - \frac{1}{2} \|\nabla \Psi_i(\hat{x}^k)\|^2 \right\} > 0.$$

Thus, if $(d^k, \mathbf{e}^k) \neq 0$, one has

$$d^+\Phi\big((\hat{x}^k, \hat{\mathbf{y}}^k); (d^k, \mathbf{e}^k)\big) \leq -K\|(d^k, \mathbf{e}^k)\|^2 < -\frac{K}{2}\|(d^k, \mathbf{e}^k)\|^2,$$

so there exist $\tau_k > 0$ such that

$$\Phi\big((\hat{x}^k, \hat{\mathbf{y}}^k) + \lambda(d^k, \mathbf{e}^k)\big) \leq \Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \lambda\frac{K}{2}\|(d^k, \mathbf{e}^k)\|^2, \quad \text{for all } \lambda \in [0, \tau_k].$$

Then, given any $\alpha > 0$, letting $\delta_k := \min\{K/(2\alpha), \tau_k\} > 0$, we have that $-K/2 \leq -\lambda\alpha$ for all $\lambda \in [0, \delta_k]$, so we obtain (8.26). □

The differentiability of h_i^* is guaranteed when h_i is strictly convex. Actually, for proper l.s.c. convex functions, *essential strict convexity* is equivalent to *essential smoothness* of the conjugate function, cf. [215, Theorem 26.3].

Before moving to the convergence analysis in the next subsection, let us explain the rationale behind Algorithm 8.17 in the simplest case in which $h_i = \Psi_i = 0$. Thanks to the κ-upper-$\mathcal{C}^2$ assumption on f, since $v^k \in \partial f(x^k)$ (Step 1), it holds

$$f(z) \leq f(x^k) + \langle v^k, z - x^k\rangle + \kappa\|z - x^k\|^2, \quad \forall z \in \mathbb{R}^n.$$

Thus, if $\gamma_k \in \left]0, \frac{1}{2\kappa}\right[$ (as in Step 2), one gets

$$\begin{aligned}\varphi(z) &= f(z) + g(z)\\ &\leq g(z) + f(x^k) + \langle v^k, z - x^k\rangle + \frac{1}{2\gamma_k}\|z - x^k\|^2 =: \widetilde{\varphi}_k(z).\end{aligned}$$

for all $z \in \mathbb{R}^n$. Therefore, the function $\widetilde{\varphi}_k$ provides an upper bound to φ, so it makes sense to take

$$\hat{x}^k \in \operatorname*{argmin}_{z\in\mathbb{R}^n} \widetilde{\varphi}_k(z) = \operatorname{prox}_{\gamma_k g}\left(x^k - \gamma_k v^k\right),$$

which coincides with (8.21). Finally, when g is differentiable at $\hat{x}^k$, the linesearch condition (8.23) in Step 6 permits to further reduce the original function φ.

For illustration, consider the function $\varphi(x) := x_1 + x_2 - \|x\|_1 + \|x\|^2$, for $x \in \mathbb{R}^2$, from [27, Example 2.4]. If we let $f(x) := x_1 + x_2 - \|x\|_1$ and $g(x) := \|x\|^2$, then f is κ-upper-$\mathcal{C}^2$ for $\kappa = 0$, by Example 8.10. If we take $x^0 := (0, 1)^T$ and $\gamma_0 := 1$, we have $v^0 := (2, 0)^T \in \partial f(x^0) = \{(2, 0)^T, (0, 0)^T\}$ and $\hat{x}^0 = \operatorname{prox}_{\gamma_0 g}(x^0 - \gamma_0 v^0) = \frac{1}{3}(-2, 1)^T$, which minimizes the function $\widetilde{\varphi}_0$. In Figure 8.2, we represent the sections of φ and $\widetilde{\varphi}$ at $\hat{x}^0$ in the direction $d^0 = \hat{x}^0 - x^0 = -\frac{2}{3}(1, 1)^T$. Taking for instance $\alpha = 0.1$, we can observe how

the linesearch step in the direction d^0 can help to achieve an additional reduction of the objective function.

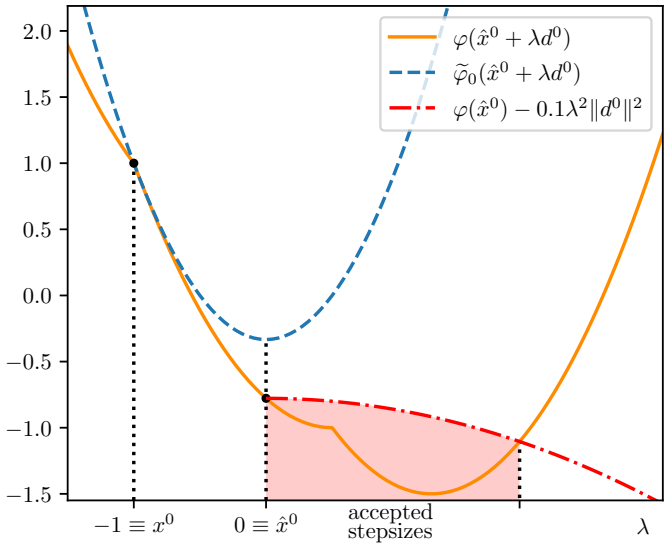


FIGURE 8.2: Sections of the functions φ and $\widetilde{\varphi}_0$ at $\hat{x}^0$ in the direction d^0. The point x^0 corresponds with the stepsize $\lambda = -1$.

We emphasize that the differentiability of g and h_i^* is only required in Proposition 8.21, and is not assumed in our subsequent results for analyzing the convergence of BDSA. The fixed number R of attempts in BDSA's linesearch works as a safeguard for the nondifferentiable case, in which the vector $(d^k, \mathbf{e}^k)$ may not define a descent direction for Φ, stopping the linesearch if the decrease condition fails R times. Nonetheless, numerical experience shows that this linesearch usually leads to an improved performance of the method.

8.2.1 Convergence analysis

The following result shows that the primal-dual functional Φ of problem ($\mathcal{PD}$) evaluated at the sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ generated by BDSA decreases after every iteration of the algorithm.

Proposition 8.22. *Let Φ be the function defined in (8.14) and suppose that Assumption 8.14 holds. Given a starting point $(x^0, \mathbf{y}^0) = (x^0, y_1^0, \ldots, y_p^0) \in \mathbb{R}^n \times \mathbb{R}^m$, consider the*

sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ *generated by Algorithm 8.17. Then, for all* $k \geq 1$,

$$\Phi(x^{k+1},\mathbf{y}^{k+1}) - \Phi(x^k,\mathbf{y}^k) \leq -a_k\|x^{k+1}-x^k\|^2 - \sum_{i=1}^{p} b_i^k\|y_i^{k+1}-y_i^k\|^2, \tag{8.31}$$

where

$$a_k := \frac{2\alpha\lambda_k^2 + \gamma_k^{-1} - 2\kappa - \sum_{i=1}^p L_i\|y_i^k\|}{2(1+\lambda_k)^2} > 0 \quad \textit{and} \quad b_i^k := \frac{1+\alpha\lambda_k^2\mu_i^k}{\mu_i^k(1+\lambda_k)^2} > 0,$$

for $i \in [\![1,p]\!]$.

Proof. First, note that for $k \geq 1$ the vector $(x^k, \mathbf{y}^k)$ belongs to $\operatorname{dom}\Phi$. By the definition of the proximal point operator, equation (8.21) yields the inequality

$$g(\hat{x}^k) + \frac{1}{2\gamma_k}\left\|\hat{x}^k - x^k - \gamma_k\left(\sum_{i=1}^p \nabla\Psi_i(x^k)y_i^k - v^k\right)\right\|^2 \leq g(x^k) + \frac{\gamma_k}{2}\left\|\sum_{i=1}^p \nabla\Psi_i(x^k)y_i^k - v^k\right\|^2.$$

Rearranging this expression and remembering that $d^k = \hat{x}^k - x^k$ we get

$$g(\hat{x}^k) - g(x^k) \leq \left\langle d^k, \sum_{i=1}^p \nabla\Psi_i(x^k)y_i^k - v^k\right\rangle - \frac{1}{2\gamma_k}\|d^k\|^2. \tag{8.32}$$

Now, let us notice that since the function $x \mapsto -\langle\Psi_i(x), y_i^k\rangle$ is $\mathcal{C}^1$ with $L_i\|y_i^k\|$-Lipschitz gradient, then by (8.5) we have

$$\langle\Psi_i(x^k) - \Psi_i(\hat{x}^k), y_i^k\rangle \leq -\langle\nabla\Psi_i(x^k)y_i^k, d^k\rangle + \frac{L_i\|y_i^k\|}{2}\|d^k\|^2.$$

Using this expression and (8.32), we obtain

$$\begin{aligned}
\Phi(\hat{x}^k,\mathbf{y}^k) - \Phi(x^k,\mathbf{y}^k) =& f(\hat{x}^k) - f(x^k) + g(\hat{x}^k) - g(x^k) \\
&+ \sum_{i=1}^p \langle\Psi_i(x^k) - \Psi_i(\hat{x}^k), y_i^k\rangle \\
\leq& f(\hat{x}^k) - f(x^k) + g(\hat{x}^k) - g(x^k) \\
&+ \sum_{i=1}^p \left(-\langle\nabla\Psi_i(x^k)y_i^k, d^k\rangle + \frac{L_i\|y_i^k\|}{2}\|d^k\|^2\right) \\
\leq& f(\hat{x}^k) - f(x^k) - \frac{1}{2\gamma_k}\|d^k\|^2 - \langle v^k, d^k\rangle + \frac{1}{2}\sum_{i=1}^p L_i\|y_i^k\|\|d^k\|^2 \\
\leq& \left(\kappa + \frac{1}{2}\sum_{i=1}^p L_i\|y_i^k\| - \frac{1}{2\gamma_k}\right)\|d^k\|^2,
\end{aligned} \tag{8.33}$$

where the last inequality is due to Proposition 8.9. On the other hand, in equation (8.22) we are computing the proximity operator of the convex function h_i^*, which yields the subgradient inequality

$$h_i^*(\hat{y}^k) + \left\langle \frac{y_i^k - \hat{y}_i^k}{\mu_i^k} + \Psi_i(\hat{x}^k), y_i^k - \hat{y}_i^k \right\rangle \leq h_i^*(y_i^k), \quad \forall i \in [\![1, p]\!].$$

Making use of this expression and $\mathbf{e}^k = \hat{\mathbf{y}}^k - \mathbf{y}^k$, we obtain

$$\Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \Phi(\hat{x}^k, \mathbf{y}^k) = \sum_{i=1}^{p} \left(h_i^*(\hat{y}_i^k) - h_i^*(y^k) - \langle \Psi_i(\hat{x}^k), e_i^k \rangle \right) \leq -\sum_{i=1}^{p} \frac{1}{\mu_i^k} \|e_i^k\|^2. \tag{8.34}$$

Then, (8.34) and (8.33) give

$$\Phi(\hat{x}^k, \hat{\mathbf{y}}^k) \leq \Phi(x^k, \mathbf{y}^k) - \left(\frac{1}{2\gamma_k} - \kappa - \frac{1}{2} \sum_{i=1}^{p} L_i \|y_i^k\| \right) \|d^k\|^2 - \sum_{i=1}^{p} \frac{1}{\mu_i^k} \|e_i^k\|^2. \tag{8.35}$$

Finally, using the linesearch (8.23) and (8.35), we get

$$\begin{aligned}
\Phi(x^{k+1}, \mathbf{y}^{k+1}) &\leq \Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \alpha \lambda_k^2 \|(d^k, \mathbf{e}^k)\|^2 \\
&\leq \Phi(x^k, \mathbf{y}^k) - \left(\alpha \lambda_k^2 + \frac{1}{2\gamma_k} - \kappa - \frac{1}{2} \sum_{i=1}^{p} L_i \|y_i^k\| \right) \|d^k\|^2 \\
&\quad - \sum_{i=1}^{p} \left(\alpha \lambda_k^2 + \frac{1}{\mu_i^k} \right) \|e_i^k\|^2 \\
&= \Phi(x^k, \mathbf{y}^k) - a_k \|x^{k+1} - x^k\|^2 - \sum_{i=1}^{p} b_i^k \|y_i^{k+1} - y_i^k\|^2,
\end{aligned}$$

where we note that the first inequality trivially holds when the linesearch procedure was not successful, as in that case $\lambda_k = 0$ by Step 7. Therefore, (8.31) holds. □

Remark 8.23. In the case in which g is assumed to be convex, the convex subdifferential characterization of the proximity operator allows to replace (8.32) by the stronger inequality

$$g(\hat{x}^k) - g(x^k) \leq \left\langle d^k, \sum_{i=1}^{p} \nabla \Psi_i(x^k) y_i^k - v^k \right\rangle - \frac{1}{\gamma_k} \|d^k\|^2.$$

This leads to the larger upper bound for the parameter γ_k given by

$$\gamma_k < \left(\kappa + \sum_{i=1}^{p} L_i \left\|y_i^k\right\|\right)^{-1}.$$

Moreover, if f is L-smooth and Ψ_i is linear, for all $i \in [\![1, p]\!]$, we recover the bound for the Double-Proximal Gradient Algorithm presented in Table 8.1.

Finally, we present the main convergence result of Algorithm 8.17.

Theorem 8.24. *Suppose that Assumption 8.14 holds. Let φ and Φ be the functions in (8.13) and (8.14), respectively. Given $(x^0, \mathbf{y}^0) \in \mathbb{R}^n \times \mathbb{R}^m$ and $\eta \in]0, 1[$, consider the pair of sequences $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ generated by Algorithm 8.17 with $\sup_{k\in\mathbb{N}, i=1,\ldots,p} \mu_i^k < +\infty$ and $\gamma_k \in \left]0, \eta \min\left\{\gamma^g, \left(2\kappa + \sum_{i=1}^{p} L_i \left\|y_i^k\right\|\right)^{-1}\right\}\right[$, for all $k \in \mathbb{N}$, and chosen such that $\inf_{k\in\mathbb{N}, i=1,\ldots,p}\{\gamma_k, \mu_i^k\} > 0$. Then either Algorithm 8.17 stops at a critical point of $(\mathcal{PD})$ after a finite number of iterations or it generates an infinite sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ such that the following assertions hold.*

(i) The sequence $\left(\Phi(x^k, \mathbf{y}^k)\right)_{k\in\mathbb{N}}$ monotonically (strictly) decreases and converges. Moreover, the sequences $(x^k)_{k\in\mathbb{N}}$ and $(\mathbf{y}^k)_{k\in\mathbb{N}}$ verify that

$$\sum_{k=0}^{\infty} \|x^{k+1} - x^k\|^2 < \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 < \infty. \tag{8.36}$$

(ii) If the sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ is bounded, the set of its accumulation points is nonempty, closed and connected.

(iii) If $(\bar{x}, \bar{\mathbf{y}}) \in \mathbb{R}^n \times \mathbb{R}^m$ is an accumulation point of the sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$, then there exists $\bar{v} \in \partial f(\bar{x})$ such that (8.18) holds, i.e., $\bar{x}$ is a critical point of $(\mathcal{P})$. In addition, $\varphi(\bar{x}) = \inf_{k\in\mathbb{N}} \Phi(x^k, y^k)$.

(iv) If $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ has at least one isolated accumulation point, then the whole sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ converges to a critical point of $(\mathcal{PD})$. Consequently, $(x^k)_{k\in\mathbb{N}}$ converges to a critical point of problem $(\mathcal{P})$.

Proof. If Algorithm 8.17 stops at some iteration $k+1$ with $x^* := x^{k+1} = x^k$ and $\mathbf{y}^{k+1} = \mathbf{y}^k$, then x^* is a critical point of $(\mathcal{P})$, as shown in Remark 8.18. Otherwise, Algorithm 8.17 generates an infinite sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$.

(i) Again, observe that $(x^k, \mathbf{y}^k) \in \operatorname{dom}\Phi$ for all $k \geq 1$. By Proposition 8.22, summing (8.31) for all $k \geq 1$, we get

$$\begin{aligned}\Phi(x^1, \mathbf{y}^1) - \inf_{k\in\mathbb{N}} \Phi(x^k, \mathbf{y}^k) &\geq \sum_{k=1}^{\infty} a_k \|x^{k+1} - x^k\|^2 + \sum_{k=1}^{\infty}\sum_{i=1}^{p} b_i^k \|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 \\ &\geq C\left(\sum_{k=1}^{\infty} \|x^{k+1} - x^k\|^2 + \sum_{k=1}^{\infty} \|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2\right),\end{aligned} \tag{8.37}$$

where $C := \inf_{k\in\mathbb{N}, i=1,\ldots,p}\left\{a_k, b_i^k\right\}$. Let us see that $C > 0$. Indeed, minimizing the value of a_k with respect to λ_k, we deduce

$$a_k = \frac{2\alpha\lambda_k^2 + \gamma_k^{-1} - 2\kappa - \sum_{i=1}^p L_i\|y_i^k\|}{2(1+\lambda_k)^2} \geq \frac{(\gamma_k^{-1} - 2\kappa - \sum_{i=1}^p L_i\|y_i^k\|)\alpha}{\gamma_k^{-1} - 2\kappa - \sum_{i=1}^p L_i\|y_i^k\| + 2\alpha},$$

whose right-hand side, as a function of γ_k, is strictly decreasing in the interval

$$\left]0, \eta\left(2\kappa + \sum_{i=1}^{p} L_i \left\|y_i^k\right\|\right)^{-1}\right].$$

Hence,

$$a_k \geq \frac{(1-\eta)\alpha}{1 - \eta + 2\alpha\eta/(2\kappa + \sum_{i=1}^p L_i\|y_i^k\|)} \geq \frac{(1-\eta)\alpha}{1 - \eta + \alpha\eta/\kappa} > 0, \quad \forall k \in \mathbb{N}.$$

Likewise,

$$b_i^k = \frac{1 + \alpha\lambda_k^2\mu_i^k}{\mu_i^k(1+\lambda_k)^2} \geq \frac{\alpha}{1+\alpha\mu_i^k} \geq \frac{\alpha}{1 + \alpha\sup_{k\in\mathbb{N}, i=1,\ldots,p}\mu_i^k} > 0, \quad \forall k \in \mathbb{N}.$$

Therefore, $C > 0$ and we obtain from (8.37) that

$$\sum_{k=1}^{\infty} \|x^{k+1} - x^k\|^2 + \sum_{k=1}^{\infty} \|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 \leq C^{-1}\left(\Phi(x^1, \mathbf{y}^1) - \inf_{k\in\mathbb{N}} \Phi(x^k, \mathbf{y}^k)\right).$$

By assumption, the right-hand side of the equation is bounded from above, so the sums in the left-hand side are finite, which proves (8.36).

(ii) Equation (8.36) implies that the sequences $(x^k)_{k\in\mathbb{N}}$ and $(\mathbf{y}^k)_{k\in\mathbb{N}}$ verify the so-called *Ostrowski's condition*, that is,

$$\lim_{k\to\infty} \|x^{k+1} - x^k\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \|\mathbf{y}^{k+1} - \mathbf{y}^k\| = 0. \tag{8.38}$$

Now, the result directly follows from [137, Theorem 8.3.9].

(iii) Let $(x^{k_j}, \mathbf{y}^{k_j})_{j\in\mathbb{N}}$ be a subsequence of $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ such that $(x^{k_j}, \mathbf{y}^{k_j}) \to (\bar{x}, \bar{\mathbf{y}})$. By (8.36), since

$$(x^{k_j+1} - x^{k_j}, \mathbf{y}^{k_j+1} - \mathbf{y}^{k_j}) = (1 + \lambda_{k_j})(\hat{x}^{k_j} - x^{k_j}, \hat{\mathbf{y}}^{k_j} - \mathbf{y}^{k_j}),$$

the sequence $(\hat{x}^{k_j}, \hat{\mathbf{y}}^{k_j})_{k\in\mathbb{N}}$ also converges to $(\bar{x}, \bar{\mathbf{y}})$. Now, we can assume without loss of generality that $\gamma_{k_j} \to \bar{\gamma} \in]0, \infty[$. Consider the subsequence $(v^{k_j})_{j\in\mathbb{N}}$ of $(v^k)_{k\in\mathbb{N}}$. In particular, $v^{k_j} \in \partial f(x^{k_j})$ for all $j \in \mathbb{N}$. Since f is locally Lipschitz continuous, v^{k_j} is bounded for all sufficiently large $j \in \mathbb{N}$ (see, e.g., [218, Proposition 9.13]). Therefore, we can also assume without loss of generality that $(v^{k_j})_{j\in\mathbb{N}}$ converges to some point $\bar{v} \in \partial f(\bar{x})$. Thus, the sequence

$$\left(x^{k_j} + \gamma_{k_j} \sum_{i=1}^{m} \nabla\Psi_i(x^{k_j}) y_i^{k_j} - \gamma_{k_j} v^{k_j}, \hat{x}^{k_j} \right)_{j\in\mathbb{N}} \subseteq \operatorname{gra} \operatorname{prox}_{\gamma_{k_j} g}$$

converges to $(\bar{x} + \bar{\gamma} \sum_{i=1}^{p} \nabla\Psi_i(\bar{x})\bar{y}_i - \bar{\gamma}\bar{v}, \bar{x})$ as $j \to \infty$. Hence, by Proposition 8.5, we get that

$$\bar{x} \in \operatorname{prox}_{\bar{\gamma} g} \left(\bar{x} + \bar{\gamma} \sum_{i=1}^{p} \nabla\Psi_i(\bar{x})\bar{y}_i - \bar{\gamma}\bar{v} \right),$$

which implies that

$$\sum_{i=1}^{p} \nabla\Psi_i(\bar{x})\bar{y}_i \in \bar{v} + \partial g(\bar{x}) \subseteq \partial f(\bar{x}) + \partial g(\bar{x}). \tag{8.39}$$

Further, using (8.24), we get

$$\frac{y_i^{k_j} - \hat{y}_i^{k_j}}{\mu_i^{k_j}} + \Psi_i(\hat{x}^{k_j}) \in \partial h_i^*(\hat{y}_i^{k_j}), \quad \forall i \in [\![1, p]\!].$$

Again, we can assume without loss of generality that $\mu_i^{k_j} \to \bar{\mu}_i \in]0, \infty[$ as $j \to \infty$ for all $i \in [\![1, p]\!]$. Taking limits as $j \to \infty$, the closedness of the subdifferential of convex functions results in

$$\Psi_i(\bar{x}) \in \partial h_i^*(\bar{y}_i), \quad \forall i \in [\![1, p]\!]. \tag{8.40}$$

Therefore, (8.39) and (8.40) imply that $\bar{x}$ is a critical point of $(\mathcal{P})$.

Let us now prove that $\varphi(\bar{x}) = \inf_{k\in\mathbb{N}} \Phi(x^k, y^k)$. On the one hand, due to (8.24), for all $i \in [\![1, p]\!]$, we have

$$h_i^*(\hat{y}_i^{k_j}) + \left\langle \frac{y_i^{k_j} - \hat{y}_i^{k_j}}{\mu_i^{k_j}} + \Psi_i(\hat{x}^{k_j}), \bar{y}_i - \hat{y}_i^{k_j} \right\rangle \leq h_i^*(\bar{y}_i), \tag{8.41}$$

from the definition of the convex subdifferential. On the other hand, for all $j \in \mathbb{N}$, we have

$$\begin{aligned} g(\hat{x}^{k_j}) + \frac{1}{2\gamma_{k_j}} \left\| \hat{x}^{k_j} - \left(x^{k_j} + \gamma_{k_j} \sum_{i=1}^{p} \nabla\Psi_i(x^{k_j}) y_i^{k_j} - \gamma_{k_j} v^{k_j} \right) \right\|^2 \\ \leq g(\bar{x}) + \frac{1}{2\gamma_{k_j}} \left\| \bar{x} - \left(x^{k_j} + \gamma_{k_j} \sum_{i=1}^{p} \nabla\Psi_i(x^{k_j}) y_i^{k_j} - \gamma_{k_j} v^{k_j} \right) \right\|^2, \end{aligned} \tag{8.42}$$

Thus, we deduce from (8.41) and (8.42) that $\limsup_{j\to\infty} \Phi(\hat{x}^{k_j}, \hat{\mathbf{y}}^{k_j}) \leq \Phi(\bar{x}, \bar{\mathbf{y}})$.

By (8.38), we have that $(x^{k_j+1}, \mathbf{y}^{k_j+1})_{j\in\mathbb{N}}$ also converges to $(\bar{x}, \bar{\mathbf{y}})$, so by lower semi-continuity of the functions defining Φ in (8.14), we have

$$\begin{aligned} \liminf_{j\to\infty} \Phi(x^{k_j+1}, \mathbf{y}^{k_j+1}) &\geq \Phi(\bar{x}, \bar{\mathbf{y}}) \\ &\geq \limsup_{j\to\infty} \Phi(\hat{x}^{k_j}, \hat{\mathbf{y}}^{k_j}) \\ &\geq \limsup_{j\to\infty} \Phi(x^{k_j+1}, \mathbf{y}^{k_j+1}), \end{aligned}$$

where the last inequality is a consequence of the linesearch (8.23). Therefore, using Proposition 8.16 (i) and item (i), we obtain

$$\varphi(\bar{x}) = \Phi(\bar{x}, \bar{\mathbf{y}}) = \lim_{j\to\infty} \Phi(x^{k_j+1}, \mathbf{y}^{k_j+1}) = \inf_{k\in\mathbb{N}} \Phi(x^k, \mathbf{y}^k),$$

which proves the claim.

(iv) In this case, by [137, Proposition 8.3.10] the sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ converges to some point $(\bar{x}, \bar{\mathbf{y}})$, which is a critical point of problem $(\mathcal{PD})$ by (iii). □

Remark 8.25 (Possible modification of Algorithm 8.17). It is worth mentioning that it would be possible to replace $\hat{x}^k$ in (8.22) by some interpolation between the points x^k and $\hat{x}^k$ of the form $(1 - \beta_i^k)\hat{x}^k + \beta_i^k x^k$, where $\beta_i^k \in \mathbb{R}$ is arbitrarily chosen at each iteration. In principle, this could allow to improve the overall performance of the algorithm. For example, setting $\beta_i = 1$ for all $i \in [\![1, p]\!]$ would permit to fully run the algorithm in parallel,

since only x^k and y_i^k would be required to compute $\hat{y}_i^k$. This would allow to simultaneously compute $\hat{x}^k$ and $\hat{y}_i^k$, which could improve the algorithm's overall efficiency when the computation of the proximal mappings is time-consuming. However, it is important to note that the bounds obtained in the accordingly modified version of Proposition 8.22 would be considerably more technical and difficult to be satisfied in practical applications when $\beta_i \neq 0$. As, in addition, in our experiments we did not find numerical evidence of the benefits to justify the inclusion of such extra linear terms, for conciseness we only consider the case where $\beta_i^k = 0$.

Remark 8.26. As mentioned in Remark 8.19, one can allow choosing $v^k \in \operatorname{co} \partial f(x^k)$ in Algorithm 8.17. The only modification in Theorem 8.24 is that if $(x^{k_j}, \mathbf{y}^{k_j})_{k\in\mathbb{N}}$ converges to $(\bar{x}, \bar{\mathbf{y}})$, then there exists $\bar{y}_i \in \partial h_i(\Psi_i(\bar{x}))$ for all $i \in [\![1, p]\!]$, such that

$$\sum_{i=1}^{p} \nabla\Psi_i(\bar{x})\bar{y}_i \in \operatorname*{Lim\,sup}_{j\to+\infty} \left\{v^{k_j}\right\} + \partial g(\bar{x}), \tag{8.43}$$

where here Lim sup refers to the Painlevé–Kuratowski upper-limit of the sequence $(v^{k_j})_{j\in\mathbb{N}}$. Furthermore, it is easy to prove that $\operatorname{Lim\,sup}_{j\to+\infty} \left\{v^{k_j}\right\} \subseteq \operatorname{co} \partial f(\bar{x})$.

8.2.2 Convergence under the Kurdyka–Łojasiewicz property

In this subsection, we establish the global convergence of Algorithm 8.17 and some convergence rates. In addition to the assumptions required by Theorem 8.24, we assume that the primal-dual function Φ satisfies the Kurdyka–Łojasiewicz property [157, 166] at some accumulation point of the sequence generated by Algorithm 8.17.

Recall that the Kurdyka–Łojasiewicz property holds for $\Phi : \mathbb{R}^n \to \mathbb{R}$ at $\bar{x} \in \mathbb{R}^n$ if there exists $\beta > 0$ and a continuous concave function $\theta : [0, \beta] \to [0, +\infty[$ such that $\theta(0) = 0$, θ is of class $\mathcal{C}^1$ on $]0, \beta[$ with a strictly positive derivative θ' and

$$\theta'\big(\Phi(x) - \Phi(\bar{x})\big)\, d(0, \partial\Phi(x)) \geq 1, \tag{8.44}$$

for all $x \in \mathbb{B}_\beta(\bar{x})$ with $\Phi(\bar{x}) < \Phi(x) < \Phi(\bar{x}) + \beta$, where $d(\cdot, \Omega)$ stands for the distance function to a set Ω. This property is shared by a vast spectrum of functions that arise in a wide variety of applications (see, e.g., [66, Appendix 5]).

The Kurdyka–Łojasiewicz property has been extensively used for proving the convergence of descent methods for nonconvex nonsmooth optimization after the seminal works [30, 31, 66, 195], which are based on Łojasiewicz's original idea [166]. Our analysis,

though, requires the use of some ad hoc techniques that do not directly follow the general methodology described in [30, Section 3.2], mainly because of the boosting linesearch. Further, unlike in [27], we do not require the strong Kurdyka–Łojasiewicz property (which allows using Clarke's subdifferential). Two of the three requirements in [30], namely the *sufficient decrease property* and the *subgradient lower bound for the iterates gap*, must be adapted to the sequence $(\hat{x}^k, \hat{\mathbf{y}}^k)_{k\in\mathbb{N}}$. We begin with a lemma that permits us to derive a modified subgradient lower bound for the iterates gap related to elements in $\partial\Phi(\hat{x}^k, \hat{\mathbf{y}}^k)$.

Lemma 8.27. *Consider a point $(\bar{x}, \bar{\mathbf{y}}) \in \mathbb{R}^n \times \mathbb{R}^m$. In addition to the assumptions of Theorem 8.24, suppose that f is L_1-smooth around $\bar{x}$. Then, there exists $r > 0$, $\rho > 0$ and $\hat{k} \in \mathbb{N}$ such that, for all $(x^k, \mathbf{y}^k) \in \mathbb{B}_r(\bar{x}, \bar{\mathbf{y}})$ with $k \geq \hat{k}$, there exists $(u^k, \mathbf{w}^k) \in \partial\Phi(\hat{x}^k, \hat{\mathbf{y}}^k)$ verifying*

$$\|(u^k, \mathbf{w}^k)\| \leq \rho \|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\|. \tag{8.45}$$

Proof. Let $r > 0$ be such that f is continuously differentiable with L_1-Lipschitz gradient on $\mathbb{B}_{2r}(\bar{x}, \bar{\mathbf{y}})$. Let $\hat{k} \in \mathbb{N}$ be such that $\|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\| \leq r$ for all $k \geq \hat{k}$. Now, consider $(x^k, \mathbf{y}^k) \in \mathbb{B}_r(\bar{x}, \bar{\mathbf{y}})$ with $k \geq \hat{k}$. It follows that $(\hat{x}^k, \hat{\mathbf{y}}^k)$ belongs to $\mathbb{B}_{2r}(\bar{x}, \bar{\mathbf{y}})$. Using (8.24), we get that

$$\begin{aligned}
&\frac{x^k - \hat{x}^k}{\gamma_k} + \sum_{i=1}^{p} \nabla\Psi_i(x^k) y_i^k - \nabla f(x^k) \in \partial g(\hat{x}^k), \\
&\frac{y_i^k - \hat{y}_i^k}{\mu_i^k} + \Psi_i(\hat{x}^k) \in \partial h_i^*(\hat{y}_i^k), \quad \forall i \in [\![1, p]\!].
\end{aligned}$$

Let us define $(u^k, \mathbf{w}^k) = (u^k, w_1^k, \ldots, w_p^k)$ by

$$\begin{aligned}
u^k &:= \frac{x^k - \hat{x}^k}{\gamma_k} + \sum_{i=1}^{p} \left(\nabla\Psi_i(x^k) y_i^k - \nabla\Psi_i(\hat{x}^k)\hat{y}_i^k\right) + \nabla f(\hat{x}^k) - \nabla f(x^k), \\
w_i^k &:= \frac{y_i^k - \hat{y}_i^k}{\mu_i^k}, \quad \forall i \in [\![1, p]\!].
\end{aligned}$$

It follows from (8.16) that $(u^k, \mathbf{w}^k) \in \partial\Phi(\hat{x}^k, \hat{\mathbf{y}}^k)$. Furthermore, $(x, \mathbf{y}) \mapsto \sum_{i=1}^{p} \nabla\Psi_i(x) y_i$ is Lipschitz continuous on $\mathbb{B}_{2r}(\bar{x})$, let us say L_2-Lipschitz continuous, so we can make the

following estimations

$$\|u^k\| \leq \left(\frac{1}{\gamma_k} + L_1\right)\|x^k - \hat{x}^k\| + L_2\|(x^k - \hat{x}^k, \mathbf{y}^k - \hat{\mathbf{y}}^k)\|,$$
$$\|w_i^k\| \leq \frac{1}{\mu_i^k}\|y_i^k - \hat{y}_i^k\|, \quad \forall i \in [\![1, p]\!].$$

Finally, let us notice that, by Step 8 of Algorithm 8.17, we have that

$$\begin{aligned}\|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\| &= (1 + \lambda_k)\|(\hat{x}^k, \hat{\mathbf{y}}^k) - (x^k, \mathbf{y}^k)\| \\ &\geq \|(\hat{x}^k, \hat{\mathbf{y}}^k) - (x^k, -\mathbf{y}^k)\|,\end{aligned} \tag{8.46}$$

which yields (8.45) taking $\rho > 0$ sufficiently large. □

Theorem 8.28. *In addition to the assumptions of Theorem 8.24, suppose the sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ generated by Algorithm 8.17 has an accumulation point $(\bar{x}, \bar{\mathbf{y}}) \in \mathbb{R}^n \times \mathbb{R}^m$ at which the Kurdyka–Łojasiewicz property (8.44) holds, assume that f is L_1-smooth around $\bar{x}$, and $\sup_{k\in\mathbb{N}} \lambda_k < +\infty$. Then $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ converges to $(\bar{x}, \bar{\mathbf{y}})$ as $k \to \infty$.*

Proof. If Algorithm 8.17 stops after a finite number of iterations, then the results clearly holds. Otherwise, Algorithm 8.17 produces an infinite sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$. Let r, ρ and $\hat{k}$ be the constants given by Lemma 8.27, let β and θ be the constant and function in the definition of the Kurdyka–Łojasiewicz property, and let $c_0 := \inf_{k\in\mathbb{N}, i=1,\ldots,p}\left\{\frac{(1-\eta)\kappa}{\eta}, \frac{1}{\mu_i^k}\right\}$, $\lambda_\infty := \sup_{k\in\mathbb{N}} \lambda_k$ and $\sigma := \rho(1 + \lambda_\infty)^2/c_0$. Consider an arbitrary $\varepsilon \in]0, \min\{r, \beta/2\}]$ and pick $k_0 \geq \hat{k}$ large enough such that the following conditions hold:

- $\|(x^{k_0}, \mathbf{y}^{k_0}) - (\bar{x}, \bar{\mathbf{y}})\| \leq \varepsilon/4$,
- $\|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\| \leq \varepsilon/4$, for all $k \geq k_0$,
- $\sigma\theta(\Phi(x^{k_0}, \mathbf{y}^{k_0}) - \Phi(\bar{x}, \bar{\mathbf{y}})) \leq \varepsilon/4$,
- $\Phi(\bar{x}, \bar{\mathbf{y}}) < \Phi(\hat{x}^k, \hat{\mathbf{y}}^k) < \Phi(\bar{x}, \bar{\mathbf{y}}) + \beta$, for all $k \geq k_0$,

where in the last assertion we have used the fact that $(\Phi(\hat{x}^k, \hat{\mathbf{y}}^k))_{k\in\mathbb{N}}$ also converges to $\Phi(\bar{x}, \bar{\mathbf{y}})$, since

$$\Phi(x^{k+1}, \mathbf{y}^{k+1}) \leq \Phi(\hat{x}^k, \hat{\mathbf{y}}^k) \leq \Phi(x^k, \mathbf{y}^k),$$

by the linesearch (8.23) and (8.35).

The rest of the proof is split into three claims.

Claim 1: *Let $k \geq k_0$ be such that $(x^k, \mathbf{y}^k) \in \mathbb{B}_\varepsilon(\bar{x}, \bar{\mathbf{y}})$. Then, the following estimation holds*

$$\Delta_{k+1} \leq (\sigma \Delta s_k \Delta_k)^{\frac{1}{2}}, \tag{8.47}$$

where $\Delta_k := \|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\|$, $\Delta s_k := s_k - s_{k+1}$ and $s_k := \theta(\Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \Phi(\bar{x}, \bar{\mathbf{y}}))$. Indeed, let $(u^k, \mathbf{w}^k) \in \partial\Phi(\hat{x}^k, \hat{\mathbf{y}}^k)$ be the vector given by Lemma 8.27 (recall that $(x^k, \mathbf{y}^k) \in \mathbb{B}_r(\bar{x}, \bar{\mathbf{y}})$). Observe that $(\hat{x}^k, \hat{\mathbf{y}}^k) \in \mathbb{B}_\beta(\bar{x}, \bar{\mathbf{y}})$, since by (8.46), it holds

$$\begin{aligned}\|(\hat{x}^k, \hat{\mathbf{y}}^k) - (\bar{x}, \bar{\mathbf{y}})\| &\leq \|(\hat{x}^k, \hat{\mathbf{y}}^k) - (x^k, \mathbf{y}^k)\| + \|(x^k, \mathbf{y}^k) - (\bar{x}, \bar{\mathbf{y}})\| \\ &\leq \|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\| + \varepsilon \leq 2\varepsilon \leq \beta\end{aligned}$$

Then, by the concavity of θ, the Kurdyka–Łojasiewicz property (8.44) applied to $(\hat{x}^k, \hat{\mathbf{y}}^k)$, inequality (8.35), and the linesearch (8.23), we have that

$$\begin{aligned}\Delta s_k \|(u^k, \mathbf{w}^k)\| &\geq \theta'\big(\Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \Phi(\bar{x}, \bar{\mathbf{y}})\big)\big(\Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \Phi(\hat{x}^{k+1}, \hat{\mathbf{y}}^{k+1})\big)\|(u^k, \mathbf{w}^k)\| \\ &\geq \Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \Phi(\hat{x}^{k+1}, \hat{\mathbf{y}}^{k+1}) \\ &\geq \Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \Phi(x^{k+1}, \mathbf{y}^{k+1}) \\ &\quad + \frac{1}{2}\left(\frac{1}{\gamma_{k+1}} - 2\kappa - \sum_{i=1}^p L_i \|y_i^{k+1}\|\right)\|d^{k+1}\|^2 + \sum_{i=1}^p \frac{1}{\mu_i^{k+1}}\|e_i^{k+1}\|^2 \\ &\geq \frac{(1-\eta)\kappa}{\eta}\|d^{k+1}\|^2 + \sum_{i=1}^p \frac{1}{\mu_i^{k+1}}\|e_i^{k+1}\|^2 \\ &\geq c_0\|(\hat{x}^{k+1}, \hat{\mathbf{y}}^{k+1}) - (x^{k+1}, \mathbf{y}^{k+1})\|^2 \\ &= \frac{c_0}{(1+\lambda_{k+1})^2}\Delta_{k+1}^2 \geq \frac{c_0}{(1+\lambda_\infty)^2}\Delta_{k+1}^2.\end{aligned}$$

Hence,

$$\Delta_{k+1} \leq (1+\lambda_\infty)\sqrt{\frac{\Delta s_k \|(u^k, \mathbf{w}^k)\|}{c_0}} \leq (1+\lambda_\infty)\sqrt{\frac{\rho}{c_0}\Delta s_k \Delta_k} = \sqrt{\sigma \Delta s_k \Delta_k},$$

which proves the claim.

Claim 2: *Let $k \geq k_0$ and assume that $(x^j, \mathbf{y}^j) \in \mathbb{B}_\varepsilon(\bar{x}, \bar{\mathbf{y}})$, for all $j \in \{k_0, \ldots, k\}$. Then*

$$\Delta_{k+1} \leq \sigma\left(\sum_{j=0}^{k-k_0} \frac{1}{2^{j+1}}\Delta s_{k-j}\right) + \frac{1}{2^{k+1-k_0}}\Delta_{k_0}. \tag{8.48}$$

Using (8.47) inductively for $j \in \{k_0, \dots, k\}$, we get

$$\Delta_{k+1} \leq \left(\prod_{j=0}^{k-k_0} (\sigma \Delta s_{k-j})^{\frac{1}{2^{j+1}}} \right) \Delta_{k_0}^{\frac{1}{2^{k+1-k_0}}}.$$

Now, let us recall the (generalized) inequality of arithmetic and geometric means (see, e.g., [9, Proposition 3.14]), which states that for any nonnegative numbers $b_0, \dots, b_{\ell+1}$,

$$\prod_{j=0}^{\ell+1} b_j^{\nu_j} \leq \sum_{j=0}^{\ell+1} \nu_j b_j, \text{ whenever } \nu_j \geq 0, \text{ with } \sum_{j=0}^{\ell+1} \nu_j = 1.$$

Using this inequality with $b_j := \sigma \Delta s_{k-j}$ and $\nu_j := \frac{1}{2^{j+1}}$ for $j = 0, \dots, k - k_0 =: \ell$, and $b_{\ell+1} := \Delta_{k_0}$ and $\nu_{\ell+1} := \frac{1}{2^{k+1-k_0}}$, we have that

$$\left(\prod_{j=0}^{k-k_0} (\sigma \Delta s_{k-j})^{\frac{1}{2^{j+1}}} \right) \Delta_{k_0}^{\frac{1}{2^{k+1-k_0}}} \leq \sigma \left(\sum_{j=0}^{k-k_0} \frac{1}{2^{j+1}} \Delta s_{k-j} \right) + \frac{1}{2^{k+1-k_0}} \Delta_{k_0},$$

which concludes the proof of (8.48).

Claim 3: *For all $k \geq k_0$, $(x^k, \mathbf{y}^k) \in \mathbb{B}_\varepsilon(\bar{x}, \bar{\mathbf{y}})$. Therefore, $(x^k, \mathbf{y}^k)_{k \in \mathbb{N}}$ converges to $(\bar{x}, \bar{\mathbf{y}})$.*

We prove by induction that $(x^k, \mathbf{y}^k) \in \mathbb{B}_\varepsilon(\bar{x}, \bar{\mathbf{y}})$ for all $k \geq k_0$. The assertion clearly holds for $k = k_0$ and $k = k_0 + 1$, so we can assume that there is $k_1 > k_0 + 1$ such that $(x^k, \mathbf{y}^k) \in \mathbb{B}_\varepsilon(\bar{x}, \bar{\mathbf{y}})$ for all $k \in \{k_0, \dots, k_1\}$. Then (8.48) holds for all $k \in \{k_0, \dots, k_1\}$, so we get that

$$\begin{aligned}
\sum_{k=k_0}^{k_1} \|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\| &= \Delta_{k_0} + \sum_{k=k_0}^{k_1-1} \Delta_{k+1} \\
&\leq \Delta_{k_0} + \sum_{k=k_0}^{k_1-1} \left(\sigma \left(\sum_{j=0}^{k-k_0} \frac{1}{2^{j+1}} \Delta s_{k-j} \right) + \frac{1}{2^{k+1-k_0}} \Delta_{k_0} \right) \\
&\leq \Delta_{k_0} + \sigma \sum_{k=k_0}^{k_1-1} \left(\sum_{j=0}^{k-k_0} \frac{1}{2^{j+1}} \Delta s_{k-j} \right) + \left(\sum_{k=1}^{\infty} \frac{1}{2^j} \right) \Delta_{k_0} \\
&\leq 2\Delta_{k_0} + \sigma \sum_{k=k_0}^{k_1-1} \left(\sum_{j=0}^{k-k_0} \frac{1}{2^{j+1}} \Delta s_{k-j} \right) \\
&= 2\Delta_{k_0} + \sigma \sum_{j=1}^{k_1-k_0} \frac{1}{2^j} \left(\sum_{k=k_0}^{k_1-j} \Delta s_k \right) \\
&\leq 2\Delta_{k_0} + \sigma \sum_{j=1}^{k_1-k_0} \frac{1}{2^j} s_{k_0} \leq 2\Delta_{k_0} + \sigma s_{k_0}.
\end{aligned}$$

Hence,

$$\|(x^{k_1+1}, \mathbf{y}^{k_1+1}) - (\bar{x}, \bar{\mathbf{y}})\| \leq \|(x^{k_0}, \mathbf{y}^{k_0}) - (\bar{x}, \bar{\mathbf{y}})\| + \sum_{k=k_0}^{k_1} \|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\|$$
$$\leq \frac{\varepsilon}{4} + 2\Delta_{k_0} + \sigma s_{k_0} \leq \frac{\varepsilon}{4} + \frac{2\varepsilon}{4} + \frac{\varepsilon}{4} = \varepsilon,$$

which demonstrates the assertion for $k = k_1 + 1$.

Therefore,

$$\sum_{k=k_0}^{\infty} \|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\| \leq 2\Delta_{k_0} + \sigma s_{k_0}, \tag{8.49}$$

which proves that $(x^k, \mathbf{y}^k)$ is a Cauchy sequence, so it converges to $(\bar{x}, \bar{\mathbf{y}})$. □

The next theorem allows to deduce convergence rates of the sequence generated by Algorithm 8.17 when the Kurdyka–Łojasiewicz property holds for a specific choice of function θ. We will make use of the following lemma.

Fact 8.29 ([24, Lemma 1]). *Let $(z_k)_{k\in\mathbb{N}}$ be a nonnegative sequence in $\mathbb{R}$ and let ξ, ζ be some positive constants. Suppose that $z^k \to 0$ and that the sequence satisfies*

$$z_k^{\xi} \leq \zeta(z_k - z_{k+1}),$$

for all k sufficiently large. Then the following assertions hold.

(i) If $\xi = 0$, then the sequence $(z_k)_{k\in\mathbb{N}}$ converges to 0 in a finite number of steps.

(ii) If $\xi \in]0, 1]$, the sequence $(z_k)_{k\in\mathbb{N}}$ converges linearly to 0 with rate $1 - \frac{1}{\zeta}$.

(iii) If $\xi > 1$, there exists $\varrho > 0$ such that

$$z_k \leq \varrho k^{-\frac{1}{\xi-1}},$$

for all k sufficiently large.

Theorem 8.30 (Convergence rates). *In addition to the assumptions of Theorem 8.28, suppose that the function θ in the definition of the Kurdyka–Łojasiewicz property is given by $\theta(t) := Mt^{1-\vartheta}$ for some $M > 0$ and $0 \leq \vartheta < 1$. Then, we obtain the following convergence rates.*

(i) If $\vartheta = 0$, the sequence $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ converges in a finite number of steps to $(\bar{x}, \bar{\mathbf{y}})$.

(ii) If $\vartheta \in \left]0, \frac{1}{2}\right]$, *then the sequence* $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ *converges linearly to* $(\bar{x}, \bar{\mathbf{y}})$.

(iii) If $\vartheta \in \left]\frac{1}{2}, 1\right[$, *then there exists a positive constant* ϱ *such that for all* k *large enough*

$$\|(x^k, \mathbf{y}^k) - (\bar{x}, \bar{\mathbf{y}})\| \leq \varrho k^{-\frac{1-\vartheta}{2\vartheta-1}}.$$

Proof. For proving (i), let $\vartheta = 0$. By (8.44), (8.45) and Claim 3 from previous theorem, we have that

$$1 \leq M\|(u^k, \mathbf{w}^k)\| \leq \rho\|(x^{k+1}, \mathbf{y}^{k+1}) - (x^k, \mathbf{y}^k)\|,$$

for all k sufficiently large. Therefore, Theorem 8.24 concludes that $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ stops after a finite number of iterations, or otherwise we would enter into contradiction with (8.36).

For the remaining cases, consider the sequence $S_k := \sum_{\ell=k}^{\infty} \|(x^{\ell+1}, \mathbf{y}^{\ell+1}) - (x^\ell, \mathbf{y}^\ell)\|$, which is finite for any $k \geq 0$ due to (8.49). The convergence of $(x^k, \mathbf{y}^k)_{k\in\mathbb{N}}$ to $(\bar{x}, \bar{\mathbf{y}})$ can be studied by means of S_k, since $\|(x^k, \mathbf{y}^k) - (\bar{x}, \bar{\mathbf{y}})\| \leq S_k$.

Recall that $\theta'(t) = (1-\vartheta)Mt^{-\vartheta}$. Then, for any k large enough (8.49) implies

$$\begin{aligned} S_k &= \sum_{\ell=k}^{\infty} \Delta_\ell \leq 2\Delta_k + \sigma s_k \\ &= 2\Delta_k + \sigma M \left(\Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \Phi(\bar{x}, \bar{\mathbf{y}})\right)^{1-\vartheta} \\ &= 2\Delta_k + \frac{\sigma M^{\frac{1}{\vartheta}}(1-\vartheta)^{\frac{1-\vartheta}{\vartheta}}}{\theta'\left(\Phi(\hat{x}^k, \hat{\mathbf{y}}^k) - \Phi(\bar{x}, \bar{\mathbf{y}})\right)^{\frac{1-\vartheta}{\vartheta}}} \\ &\leq 2\Delta_k + \sigma M^{\frac{1}{\vartheta}}(1-\vartheta)^{\frac{1-\vartheta}{\vartheta}} \rho^{\frac{1-\vartheta}{\vartheta}} \Delta_k^{\frac{1-\vartheta}{\vartheta}}, \end{aligned} \tag{8.50}$$

where the last inequality is due to (8.44) and (8.45). If $\vartheta \in \left]0, \frac{1}{2}\right]$, the dominant term in the right hand side of the above equation is the first summand. Therefore, there exists $k_1 > 0$ and $K_1 > 0$ such that

$$S_k \leq K_1 \Delta_k, \quad \text{for all } k \geq k_1.$$

This implies (ii) by resorting to Fact 8.29 (ii). On the other hand, if $\vartheta \in \left]\frac{1}{2}, 1\right[$, the second term in the right hand side of (8.50) would be the dominant one. This yields the existence of some $k_2 > 0$ and $K_2 > 0$ such that

$$S_k^{\frac{\vartheta}{1-\vartheta}} \leq K_2 \Delta_k, \quad \text{for all } k \geq k_2.$$

Finally, the conclusion of (iii) similarly follows from Fact 8.29 (iii). □

8.3 Numerical experiments

In this section, we present some computational experiments where we evaluate the performance of Algorithm 8.17. We recall that when $R > 0$, so the linesearch in Steps 6-7 is performed, the resulting algorithm is referred to as BDSA; otherwise, Algorithm 8.17 without linesearch is named as DSA.

The linesearch of BDSA requires the selection of a number of hyperparameters, namely, the initial stepsize $\overline{\lambda}_k$, the backtracking constant ρ and the number of trials R. This may seem to be a drawback, as each particular problem could require of a specific tuning of all these parameters in order to obtain a good performance of the linesearch. Quite the opposite, the next numerical experiments on very different applications evidence that this is not the case: we ran all instances of BDSA with the same choice of parameters specified below and this general tuning was good enough for BDSA to significantly outperform its counterpart DSA with no linesearch, as well as some inertial methods.

Parameter tuning for Algorithm 8.17 linesearch All the linesearches for BDSA in our numerical experiments were performed with the following choice of parameters: $R = 2$, $\rho = 0.5$ and $\alpha = 0.1$. The initial stepsize $\overline{\lambda}_k$ was chosen according to the self-adaptive trial stepsize scheme presented in Algorithm 8.31 with $\overline{\lambda}_0 = 2$ and $\delta = 2$.

Algorithm 8.31 Self-adaptive trial stepsize.

Require: $\delta > 1$ and $\overline{\lambda}_0 > 0$. Obtain λ_k from $\overline{\lambda}_k$ by Steps 4-7 of BDSA (Algorithm 8.17).

1: **if** $r = 0$ **then**
2: set $\overline{\lambda}_{k+1} := \delta\overline{\lambda}_k$;
3: **else**
4: set $\overline{\lambda}_{k+1} := \max\{\overline{\lambda}_0, \rho^r\,\overline{\lambda}_k\}$.
5: **end if**

The *self-adaptive* trial stepsize given by Algorithm 8.31 is based on the one proposed in [27] for the *Boosted Difference of Convex functions Algorithm* (BDCA). We note that a similar adaptive scheme for the *gradient descend method* was recently introduced in [235]. The procedure works as follows. Algorithm 8.31 determines how to choose the starting stepsize $\overline{\lambda}_{k+1}$ of the next iteration of the method. If in the current iteration a decrease of Φ was achieved in the first attempt of the linesearch (i.e., when $r = 0$), then the starting stepsize for the next iteration of the main algorithm is increased by setting $\overline{\lambda}_{k+1} := \delta\lambda_k$, with $\delta > 1$. Otherwise, $\overline{\lambda}_{k+1}$ is set as the maximum between the default initial stepsize

and the smallest stepsize tested in the previous iteration, i.e., $\overline{\lambda}_{k+1} := \max\{\overline{\lambda}_0, \rho^r \overline{\lambda}_k\}$ (observe that λ_k could be zero if the linesearch was not successful).

This section is divided into three subsections, each containing a different application. The purpose of the experiments in the first subsection is twofold. First, to illustrate how the linesearch from the boosting step can help reaching better critical points. Second, to show that the assignment of the terms of the objective function of ($\mathcal{P}$) to each of the functions f, g, h_i and Ψ_i has a big impact in the success of the resulting scheme derived from Algorithm 8.17. In Section 8.3.2, we consider an application with real-data for clustering cities in a region and show how the linesearch of BDSA helps finding better solutions in considerably less time than DSA (which, in this context, coincides with the Generalized Proximal Point Algorithm (GPPA) [7]). Lastly, Section 8.3.3 contains a nonconvex generalization of Heron's problem that can be addressed with BDSA. In this case, BDSA is not a particular instance of any other known algorithm.

All the experiments were ran in a computer of Intel Core i7-12700H 2.30 GHz with 16GB RAM, under Windows 11 (64-bit).

8.3.1 Avoiding non-optimal critical points

Theorems 8.24 and 8.28 prove the convergence of Algorithm 8.17 to some critical point of ($\mathcal{P}$). We recall that being a critical point is a necessary (but not sufficient) condition for local optimality of problem ($\mathcal{P}$). In [27, Example 3.3] it was shown how the linesearch performed by the BDCA helps prevent the algorithm from being trapped by critical points which are not local minima. In this subsection we illustrate the same phenomenon by considering different known algorithms that can be obtained as particular cases of Algorithm 8.17. We show that its *boosted* version, with the additional linesearch, outperforms the basic methods in avoiding these non-desirable critical points. To this aim, we introduce two novel types of test functions which entail a challenge to this class of methods.

We first consider a family of test functions that have a large number of critical points where the algorithm can easily get stuck, but a unique global minimum. Specifically, for any $q \in \mathbb{N}$, we define the functions $\varphi_q : \mathbb{R}^n \to \mathbb{R}$ as

$$\varphi_q(x) := \|x\|^2 - \|x\|_1 - \sum_{j=1}^{q} (\|x - je\|_1 + \|x + je\|_1) - \|x - (q+1)e\|_1, \tag{8.51}$$

where e is the vector of ones in $\mathbb{R}^n$. It is a simple task to check that the function φ_q possesses $(2q+3)^n$ critical points, given by the set $\{-(q+1), -q, \ldots, 0, \ldots, q, q+1\}^n$, and

a unique local minimum at $x^* := (-(q+1), -(q+1), \ldots, -(q+1))^T$, which corresponds to its global optimum, with optimal value $\varphi_q(x^*) = -n(q^2+3q+2)$.

Note that the function φ_q admits diverse representations as an instance of ($\mathcal{P}$), and different algorithms are derived from BDSA depending on which terms one assigns to each of the functions f, g and h_i (recall Remark 8.20):

- Setting $f(x) := 0$, $g(x) := \|x\|^2$ and h_i, for $i = 1, \ldots, 2q+2$, to be the remaining terms involving the ℓ_1-norm, the Double-proximal Gradient Algorithm (DGA) by Banert–Boţ [38] is obtained.

- If instead we take $f(x) := -\|x\|_1 - \sum_{j=1}^{q} (\|x - je\|_1 + \|x + je\|_1) - \|x - (q+1)e\|_1$, $g(x) := \|x\|^2$ and $h(x) := 0$, we recover the particular case of the Proximal DC Algorithm (PDCA) discussed in Remark 8.20 (i), which would become the Boosted Proximal DC Algorithm (BPDCA) from [4] when $R = \infty$ (but recall that we take $R = 2$ in our experiments). Due to variable separability of the ℓ_1-norm, it can be proved that the subdifferential of f coincides with the sum of subdifferentials of the ℓ_1-norm terms. Therefore, for every $k \geq 0$, we take $v^k \in \partial f(x^k)$ as a sum of subgradients of the form

$$v^k = \sum_{s \in I} v_s^k, \quad \text{where} \quad v_s^k \in \partial\left(-\|\cdot - se\|_1\right)(x^k),$$

  where $I := \{0, 1, -1, \ldots, q, -q, q+1\}$ and every subgradient is component-wise chosen as

$$(v_s^k)_i = \begin{cases} 1 \text{ if } x_i^k \leq s, \\ -1 \text{ if } x_i^k > s, \end{cases} \quad \text{for all } i \in [\![1, n]\!].$$

- Finally, we can set $f(x) := \|x\|^2 - \sum_{j=1}^{q} (\|x - je\|_1 + \|x + je\|_1) - \|x - (q+1)e\|_1$, $g(x) := -\|x\|_1$ and $h(x) := 0$. The resulting scheme coincides with the one from the Generalized Proximal Point Algorithm [7]. However, in [7] it is required that $\inf_{x \in \mathbb{R}^n} g(x) > -\infty$, which does not hold in this case. This assumption is not needed by Algorithm 8.17 and therefore we use DSA as an abbreviation for the resulting scheme here.

The use of *inertial techniques* has also been reported to help algorithms avoid non-optimal critical points (see, e.g., [27, 62, 130]). Here, we consider the *Oliveira–Tcheou's inertial DC Algorithm* (iDCA) [130]. For our problem at hand, iDCA can be written as the *linearized proximal method with inertial term* in [130, Section 3.1.2], given by the

iteration

$$x^{k+1} = \operatorname{prox}_{\frac{\|\cdot\|^2}{\sigma}}\left(x^k + \frac{1}{\sigma}v^k + \frac{\beta}{\sigma}(x^k - x^{k-1})\right), \tag{8.52}$$

where $v^k \in \partial f(x^k)$, with $f(x) := \|x\|_1 + \sum_{j=1}^{q}(\|x - je\|_1 + \|x + je\|_1) + \|x - (q+1)e\|_1$, $\sigma > 0$ and $\beta \in [0, \sigma/2[$. Observe that the iteration (8.52) coincides with that of PDCA except for the addition of the inertial term $\frac{\beta}{\sigma}(x^k - x^{k-1})$.

For illustration, we display in Figure 8.3 for $n = 2$ and $q = 3$ the sequences generated by DGA (with $\mu = \gamma = 1$), PDCA (with $\gamma = 1$), DSA (with $\gamma = 0.49$, as one must take $\gamma < (2\kappa)^{-1}$ for $\kappa = 1$), their boosted counterparts with linesearch (abbr. as BDGA, BPDCA and BDSA) and iDCA (with $\sigma = 1$ and $\beta = 0.4995$) from the starting point $x^0 = (1.8, 0.3)^T$, using $y^0 = (0, 0)^T$ as the dual initial point for DGA and BDGA. We observe that PDCA got caught by $(1, 0)^T$, which is the nearest critical point that it encountered, while DGA and DSA converged to the slightly better critical points $(-1, -1)^T$ and $(1, -1)^T$, respectively. On the other hand, BDSA, BPDCA, BDGA and iDGA, managed to converge to $x^* = (-4, -4)^T$.

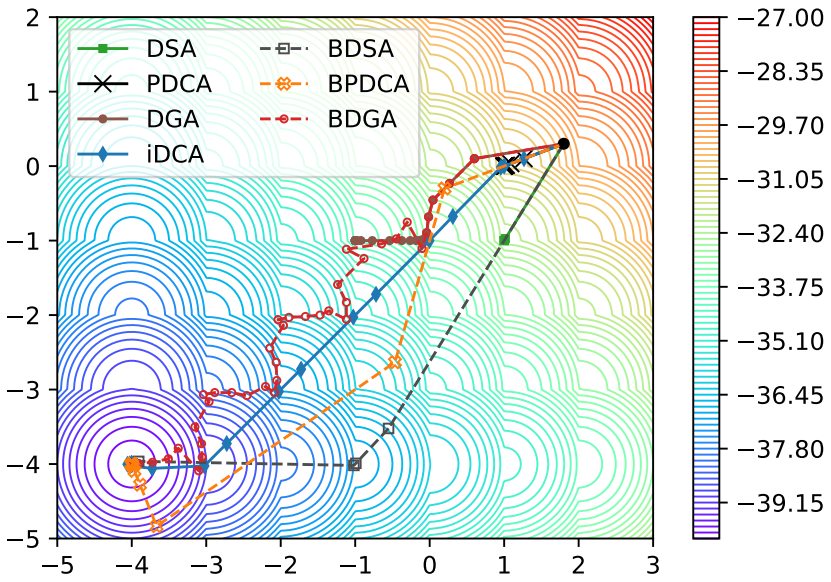


FIGURE 8.3: Sequence of iterates generated by DSA, PDCA, DGA, iDCA and the boosted versions for the same starting point, marked with a black dot, when they were applied to the function φ_3 in (8.51) for $n = 2$. Only the boosted algorithms and iDCA managed to converge to the global optimum $x^* = (-4, -4)^T$.

For different combinations of n and q, we performed 10 000 runs of the algorithms all initialized at the same starting points randomly chosen in the interval $[-q-2, q+2]^n$, and $[-1, 1]^n$ for the dual variables (when necessary), using the same parameters as before. We

note that the conjugate of the ℓ_1-norm is the indicator function of $[-1,1]^n$, so this seems a fair set in which to choose the initial dual variables. We stopped all the algorithms when the norm of the difference between two consecutive iterates was smaller than $n \times 10^{-6}$ and counted how many times each of the methods converged to the optimal solution x^*. The results are summarized in Table 8.2.

n	q	DGA	BDGA	PDCA	BPDCA	iDCA	DSA	BDSA
2	3	273	1202	410	10 000	10 000	410	10 000
2	5	72	774	201	10 000	10 000	201	10 000
2	10	10	440	71	10 000	10 000	71	10 000
2	20	0	253	21	10 000	10 000	21	10 000
10	3	0	2229	0	10 000	10 000	0	10 000
20	3	0	2076	0	10 000	10 000	0	10 000

TABLE 8.2: For different values of n and q, and 10 000 random starting points, we count the number of instances that each of the algorithms converged to the global minimum $x^* = (-q-1,\ldots,-q-1)^T$ of the function φ_q in (8.51). All algorithms are particular instances of Algorithm 8.17 except for iDCA.

The most remarkable fact is that BDSA, BPDCA and iDCA converged to the optimal point x^* in every single instance. By contrast, the non accelerated versions DSA and PDCA very rarely managed to reach the optimum (1.17% of the overall instances). In fact, these two methods only reached the global optimum when the initial points belonged to the region $[-q-2,-q]^n$ in which the global minimum is located. Their iterates, though, do not necessarily coincide (see Figure 8.3) and DSA attained a lower function value in 32.96% of the instances (the values were the same in the rest). On the other side, DGA also got trapped very often by non-optimal critical points, only converging to the global minimum in 0.59% of the instances. Its accelerated version BDGA greatly improved this poor result and converged 11.62% of the times to the optimal solution.

The fact that PDCA, DGA and DSA have such a low rate of success in reaching the global minimum is an indicator of how challenging the proposed family of functions is for this type of algorithms. The advantage of the boosted versions of the algorithms for this family is clear. Even so, it is important to mention that, although the linesearch in BDGA only succeeded in improving its success rate up to 11.62%, it consistently improved the objective values. BDGA converged to a point with lower objective value than DGA in 46.61% of the instances, while both algorithms attained the same value in the remaining 53.39%. DGA did not surpass BDGA in any of the 60 000 instances.

The complete success of BPDCA, iDCA and BDSA for finding the global optimum in the previous experiment might be due to the lack of any local (non-global) minima. Let us now consider the function $\psi : \mathbb{R}^n \to \mathbb{R}$ given by

$$\psi(x) = \|x\|^2 - \sum_{i=1}^{n} \log\left(2 + e^{2x_i}\right) - \|x\|_1, \tag{8.53}$$

which has a unique global minimum at $x^* := (a, a, \ldots, a)^T$ and $2^n - 1$ local minima at the points $\{a, b\}^n \setminus \{x^*\}$, with $a \approx 1.3895$ and $b \approx -0.2767$ (their values can be approximated numerically by solving the necessary optimality condition).

If we let $f(x) := -\sum_{i=1}^{n} \log\left(2 + e^{2x_i}\right) - \|x\|_1$ and $g(x) := \|x\|^2$, we observe that $\psi = f + g$ is expressible as a difference of convex functions. To study how the most successful methods in our previous experiment behave under the presence of multiple local minima, we apply both BPDCA (with $\gamma = 1$) and iDCA (with $\beta = 0.4995$ and $\sigma = 1$) to this decomposition, and BDSA (with $\gamma = 0.49$) to the splitting $\psi = f + g$ with $f(x) := ||x||^2 - \sum_{i=1}^{n} \log\left(2 + e^{2x_i}\right)$ and $g(x) := -||x||_1$. For $n = 2$, we show in Figure 8.4 the sequences generated by these three algorithms from two starting points.

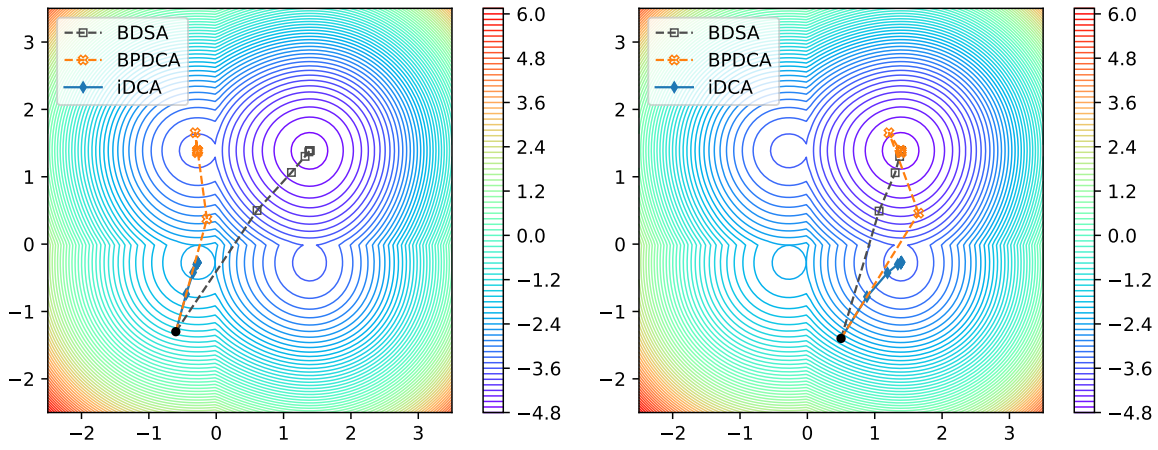


FIGURE 8.4: Iterates of BDSA, BPDCA and iDCA for the minimization of the function ψ in (8.53) with two different starting points, represented with black dots.

Figure 8.4 suggests that the inertial algorithm could be less likely to escape from a local minimum. BDSA and BPDCA managed to avoid these critical points and converged to the global optimum for the starting point shown in the right, but this clearly depends on the stepsize considered. In general there is no guarantee that any of these techniques (the inertial nor the boosting linesearch) will succeed in making the algorithms converge to global minima. However, the inertial term seems to be more influenced by the geometry of

the function, while the freedom in the choice of the stepsize in the linesearch could provide better chances of not getting stuck. To test if this was the case, for different values of n, we ran the three algorithms from the same 10 000 starting points randomly chosen in the box $[-2.5, 3.5]^n$, stopping them when the difference between two consecutive iterates was smaller than $n \times 10^{-6}$. In Table 8.3, we display the number of instances that each method converged to the global optimum. BDSA managed to attain the global minimum in every single instance, but the other two algorithms never succeeded when $n \geq 100$ for any starting point. To compare the performance of iDCA and BPDCA, we also counted the number of times that each of these algorithms achieved a strictly lower objective value than the other.

	Global minimum			Lowest value	
n	iDCA	BPDCA	BDSA	iDCA	BPDCA
2	3374	6764	10 000	0	4129
5	647	2939	10 000	0	5847
10	48	798	10 000	0	7251
20	0	54	10 000	0	8818
100	0	0	10 000	0	9998
1000	0	0	10 000	0	10 000
5000	0	0	10 000	0	10 000
10 000	0	0	10 000	0	10 000

TABLE 8.3: For different values of n and 10 000 random starting points, we count the number of instances that each of the algorithms converged to the global minimum x^* of the function ψ in (8.53). In the last two columns we only compare iDCA with BPDCA and count the number of times that each method attains a strictly lower objective value than the other.

8.3.2 Minimum sum-of-squares constrained clustering problem

Clustering analysis is a widely-employed technique in data science for classifying a collection of objects into groups, called *clusters*, whose elements share similar characteristics. In order to mathematically describe the clustering problem, we can think of our data as a finite set of points $A = \{a^1, \ldots, a^q\}$ in $\mathbb{R}^s$. Our goal is to group A into ℓ disjoint subsets $A^1, \ldots, A^\ell$, based on the minimization of some clustering measure.

In the *minimum sum-of-squares clustering problem*, the groups are determined by the minimization of the squared Euclidean distance of each data to the *centroid* of its cluster. In this way, each cluster A^j is identified by its centroid, which we denote by $x^j \in \mathbb{R}^s$, for $j = 1, \ldots, \ell$. Letting $X := (x^1, \ldots, x^\ell) \in \mathbb{R}^{s \times \ell}$, this clustering problem can be reformulated

as the optimization problem

$$\min_{X\in\mathbb{R}^{s\times\ell}} f(X) := \frac{1}{q}\sum_{i=1}^{q}\omega_i(X), \tag{8.54}$$

where $\omega_i(X) := \min\left\{\|x^j - a^i\|^2 : j = 1,\dots,\ell\right\}$. The function f is 1-upper-$\mathcal{C}^2$ (recall Example 8.11).

In [27], the authors considered the clustering problem (8.54) with the aim of grouping the 4001 Spanish cities in the peninsula with more than 500 residents. In this work, we consider a more challenging version of the above problem in which we add a nonconvex constraint on X. This is useful for example when the centroids represent facilities (e.g., hospitals or government administrations). In this case, the centroids cannot be located in the sea, or even in certain areas that should be avoided. Therefore, we are interested in solving the problem

$$\min_{X\in C} f(X), \tag{8.55}$$

where $C \subseteq \mathbb{R}^{s\times\ell}$ is the newly introduced (not necessarily convex) constraint. This allows us to make the experiment in [27] more challenging, in the following way:

- We consider the cities with more than 500 residents in the Spanish peninsula, but also those in the Balearic Islands, which is an archipelago in the Mediterranean Sea. They sum a total of 4049 cities.
- We exclude a region in the center of Spain as a possible location for centroids, which would be useful if decentralization policies were aimed.
- We exclude Portugal, which is also contained in the same peninsula as Spain.

The resulting closed nonconvex constraint is depicted in Figure 8.5.

Now, considering the objective function of problem (8.55) as a large sum of nonsmooth functions, the sum rule for the basic subdifferential only offers an upper estimation rather than an equality. Consequently, it becomes more convenient to compute subgradients of individual functions ω_i instead of examining the entire function f. In this context, the following proposition formally provides the computation of the subdifferential of the functions ω_i, for $i = 1,\dots,q$.

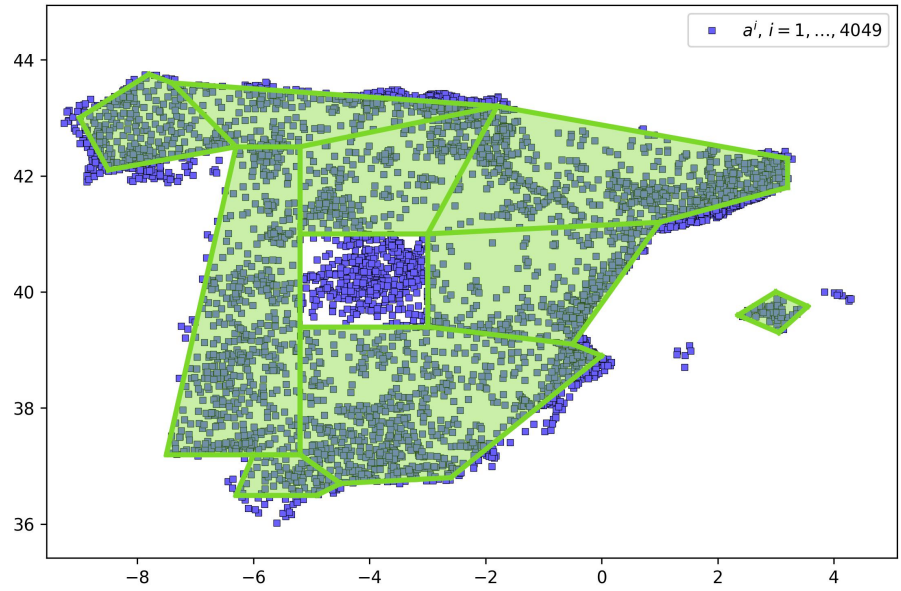


FIGURE 8.5: The blue squares represent the 4049 cities of Spain peninsula and Balearic Islands with more than 500 inhabitants. In order to accurately gather all the area of Spain including these cities, we build our constraint C as the union of a finite number of shaded polyhedral sets. Note that the rectangle in the center of Spain is excluded.

Proposition 8.32. *Given $a \in \mathbb{R}^s$, consider the function $\omega : \mathbb{R}^{s\times\ell} \to \mathbb{R}$ given by*

$$\omega(X) := \min\left\{\|x^j - a\|^2 : j = 1, \ldots, \ell\right\},$$

with $X = (x^1, \ldots, x^\ell) \in \mathbb{R}^{s\times\ell}$. Then, for all $X \in \mathbb{R}^{s\times\ell}$, the following formula holds

$$\partial\omega(X) = \left\{(0, \ldots, 0, \underbrace{2(x^j - a)}_{j\text{-th position}}, 0, \ldots, 0) : \omega(X) = \|x^j - a\|^2\right\}. \tag{8.56}$$

Proof. To prove (8.56) let us notice that the inclusion $\subseteq$ follows from the calculus rule for the minimum function (see, e.g., [182, Proposition 1.113]). To prove the opposite inclusion, we recall that by (8.10) the following equality holds

$$\operatorname{co}\partial\omega(X) = -\partial(-\omega)(X). \tag{8.57}$$

Now, since $-\omega$ is a maximum of quadratic forms, we can apply [182, Theorem 3.46] to $-\omega$ to conclude that

$$-\partial(-\omega)(X) = \operatorname{co} B(X),$$

where $B(X)$ is the set in the right-hand side of (8.56). Finally, since all the points in the set $B(X)$ are linearly independent, we get that $\supseteq$ must hold in (8.56), as otherwise it would contradict (8.57). □

Based on the aforementioned observation, we are motivated to present Algorithm 8.33 as a well-suited variant of the Boosted Double-proximal Subgradient Algorithm for effectively addressing the constrained clustering problem (8.54). When C is defined by linear inequality constraints, observe that feasibility of the direction D^k defined in Step 3 can be checked as in [21, Algorithm 1] (see also Lemma 3.1 there), so the boosting in Step 5 is only run when D^k is in the cone of feasible directions.

Algorithm 8.33 Boosted proximal Subgradient Algorithm for constrained clustering.

Require: $X^0 \in \mathbb{R}^{s\times\ell}$, $R \geq 0$, $\rho \in]0,1[$ and $\alpha \geq 0$. Set $k := 0$.

1: Choose $v_i^k \in \partial\omega_i(X^k)$ for $i = 1,\ldots,q$ and set $V^k = \frac{1}{q}\sum_{i=1}^q v_i^k$.

2: Take some positive $\gamma_k < \frac{1}{2}$ and compute

$$\hat{X}^k \in P_C\left(X^k - \gamma_k V^k\right).$$

3: Choose any $\overline{\lambda}_k \geq 0$. Set $\lambda_k := \overline{\lambda}_k$, $r := 0$ and $D^k := \hat{X}^k - X^k$.

4: **if** $D^k = 0$ **then** STOP and return x^k.

5: **while** $r < R$ and

$$\hat{X}^k + \lambda_k D^k \notin C \text{ or } f\left(\hat{X}^k + \lambda_k D^k\right) > f(\hat{X}^k) - \alpha\lambda_k^2\|D^k\|^2,$$

do $r := r + 1$ and $\lambda_k := \rho^r\overline{\lambda}_k$.

6: **if** $r = R$ **then** $\lambda_k := 0$.

7: Set $X^{k+1} := \hat{X}^k + \lambda_k D^k$, $k := k + 1$ and go to Step 1.

Now, in order to present our convergence result for Algorithm 8.33, we define a suitable notion of critical point: we say that $\bar{X}$ is a *critical point* of the constrained clustering problem (8.55) if

$$0 \in \frac{1}{q}\sum_{i=1}^q \partial\omega_i(\bar{X}) + N_C(\bar{X}),$$

where N_C denotes the (*basic*) *normal cone* to C, which coincides with $\partial\iota_C$ (see, e.g.,[182, Proposition 1.79]).

Finally, the following corollary gathers the convergence result of Algorithm 8.33.

Corollary 8.34. *Given $X^0 \in \mathbb{R}^{s\times\ell}$ and $\eta \in]0,1[$, consider the sequence $(X^k)_{k\in\mathbb{N}}$ generated by Algorithm 8.33 with $\gamma_k \in \left]0, \frac{\eta}{2}\right]$ for all $k \in \mathbb{N}$. Then either Algorithm 8.33 stops at a critical point of (8.55) after a finite number of iterations or it generates an infinite sequence $(X^k)_{k\in\mathbb{N}}$ such that the following assertions hold.*

(i) The sequence $\left(f(X^k)\right)_{k\in\mathbb{N}}$ monotonically (strictly) decreases and converges, and $X^k \in C$ for all $k \geq 1$. Moreover, the sequences $(X^k)_{k\in\mathbb{N}}$ verifies that

$$\sum_{k=0}^{\infty} \|X^{k+1} - X^k\|^2 < \infty.$$

(ii) If the sequence $(X^k)_{k\in\mathbb{N}}$ is bounded, the set of its accumulation points is nonempty, closed and connected.

(iii) If $\bar{X} \in \mathbb{R}^{s\times\ell}$ is an accumulation point of the sequence $(X^k)_{k\in\mathbb{N}}$, then $\bar{X}$ is a critical point of (8.55). In addition, $f(\bar{X}) = \inf_{k\in\mathbb{N}} f(X^k)$.

(iv) If $(X^k)_{k\in\mathbb{N}}$ has at least one isolated accumulation point, then the whole sequence $(X^k)_{k\in\mathbb{N}}$ converges to a critical point of (8.55).

Proof. To prove this corollary, let us first notice that every subgradient V^k belongs to $\operatorname{co}\partial f(X^k)$. Indeed, by [182, Theorem 3.46 (ii)], we have that each function $-\omega_i$ is lower regular at any point. Hence, by resorting to (8.10) and the sum rule for the basic subdifferential, we get that

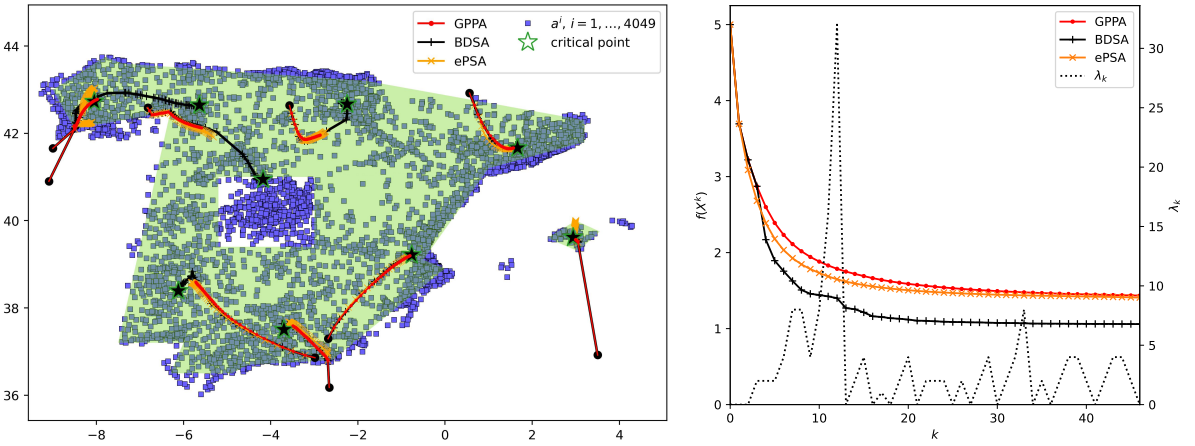
$$\operatorname{co}\partial f(X) = -\partial(-f)(X) = -\frac{1}{q}\sum_{i=1}^{q} \partial(-\omega_i)(X) = \frac{1}{q}\sum_{i=1}^{q} \operatorname{co}\partial\omega_i(X) \supseteq \frac{1}{q}\sum_{i=1}^{q} \partial\omega_i(X).$$

Therefore, we obtain the result by using Remark 8.26, where the justification that $\bar{X}$ is a critical point of (8.55) follows from (8.43). □

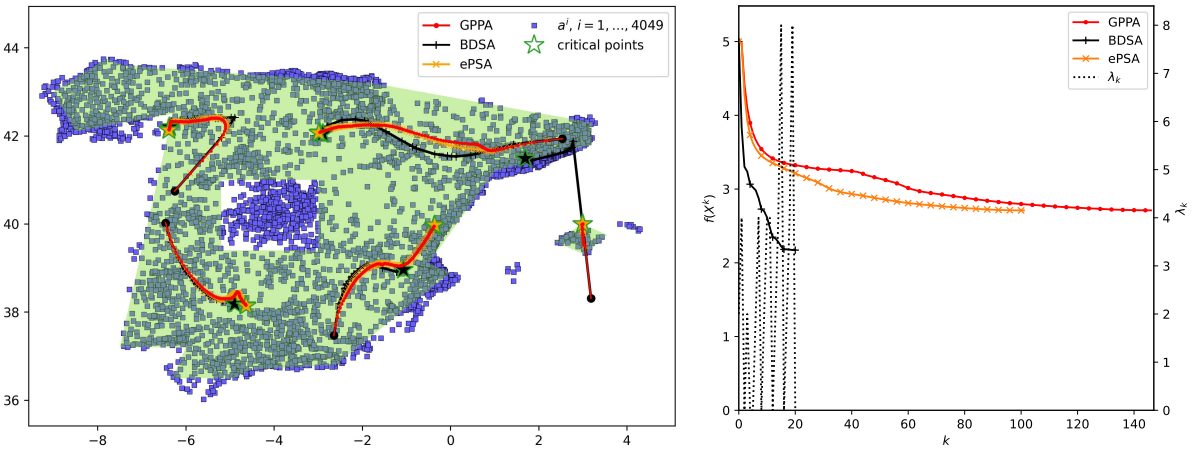
In this setting, we performed various experiments in which we compared BDSA with GPPA (i.e., Algorithm 8.33 without linesearch) and the recently proposed *Proximal Subgradient Algorithm with extrapolation* (ePSA) of [204]. The latter method is a generalization of GPPA which includes two inertial terms: one in the evaluation of the proximity operator and one in the evaluation of the gradient. The algorithm also extends the *proximal linearized algorithm* [130, 227] and the *proximal DC algorithm* with extrapolation of [242]. In particular, it allows the extrapolation parameters to vary in a similar fashion to the parameter used in FISTA. In our experiments, we set $\gamma := 0.45$ for both GPPA and BDSA. We note that ePSA has a large number of parameters and determining the best combination is out of the scope of this work. We initially tuned the algorithm according to [204, Section 4.2.], but it did not show any advantage with respect to GPPA. Increasing the extrapolation parameter $\bar{\mu}$ reported some clear benefits, so we set it to 1 and kept the remaining parameters as in [204, Section 4.2.].

In our first experiment, we aim to find a partition into 9 clusters of the 4049 cities in consideration. We ran the three methods starting from the same random initial point, until they reached a relative error in the objective function (i.e., $|f(X^{k+1}) - f(X^k)|/f(X^{k+1})$) smaller than 10^{-4}. This stopping criterion was achieved by BDSA after 46 iterations, see Figure 8.6a. In particular, in the left figure we observe that BDSA progresses faster towards regions with a higher concentration of cities. Both GPPA and ePSA converged to the same critical point than BDSA, but needed 179 and 142 iterations to reach it, respectively. Figure 8.6b presents an illustrative example where the algorithms converge to distinct critical points. In this occasion, GPPA and ePSA reached the same critical point after 146 and 101 iterations, respectively, while BDSA only needed 20 iterations to converge to a better point with a lower function value. The plots on both Figure 8.6a and 8.6b give more insight into how the linesearch helps the boosted algorithm to achieve a more significant decrease in the objective value.

To show that this is the general trend, we solved the same problem with the Spanish cities for a different number of clusters $\ell \in \{3, 5, 10, 15, 20, 30, 40, 50\}$. The results are summarized in Figure 8.7. For each of these values and for 10 different random starting points, we ran BDSA until the relative error in the objective function was smaller than 10^{-4}. Then, GPPA and ePSA were run from the same starting point until they reached the same objective value than BDSA or until the relative error in the objective function was smaller than 10^{-4}. In particular, GPPA failed to reach the same value than BDSA in 62 out of the 80 runs, while ePSA failed to do so in 44 of the instances. Nevertheless, both methods were considerably slower. In Figure 8.7, we present the iterations ratio (left) and



(a)



(b)

FIGURE 8.6: We ran GPPA, BDSA and ePSA from the same initial random point, for finding centroids satisfying the constraints shown in Figure 8.5. On the left, the iterations are drawn on the map. On the right, we compare the value of f for GPPA, BDSA and ePSA along the iterations. The figure has two vertical axes, the right one corresponds with the stepsize taken by BDSA in every iteration, represented with a dotted line.

time ratio (right) of the algorithms for all the instances. On average, BDSA was more than 3 and 2.5 times faster in iterations than GPPA and ePSA, respectively. With respect to time, BDSA was more than 2 times faster than the other two methods. There was only one instance where BDSA was slower than GPPA (only in time), for $\ell = 20$, but the values of objective function f at the stopping point for ePSA and BDSA were 0.503 and 0.478, respectively.

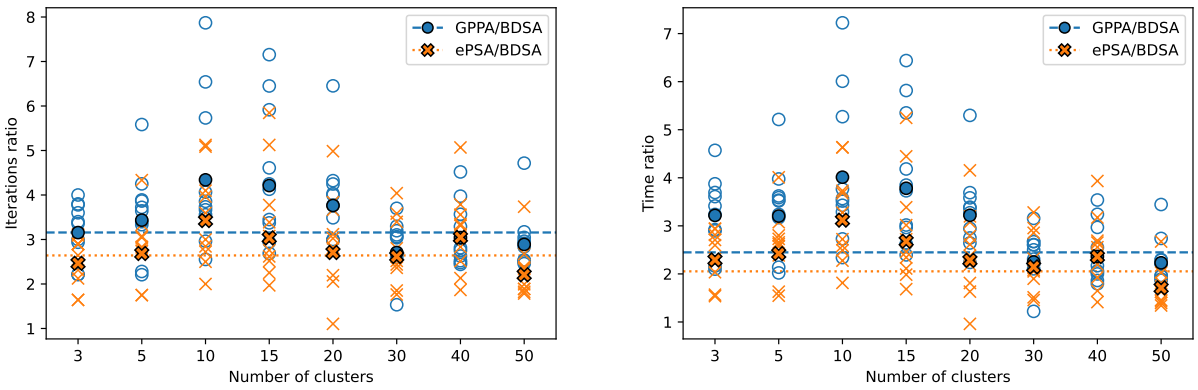


FIGURE 8.7: Iteration and time ratios between GPPA and ePSA with respect to BDSA for solving problem (8.55) with different number of clusters. The algorithms were run until they reached the same relative error, or until they (GPPA and/or ePSA) attained the objective value obtained by BDSA. The unfilled markers show the ratio for every particular instance, the filled markers represent the ratio average for a fixed number of clusters and the lines the overall ratio average.

8.3.3 A nonconvex generalization of the squared Heron problem

The original formulation of Heron's problem consists in the following: given a straight line in the plane, find a point x in it such that the sum of the distances from x to two given points is minimal. A generalization of Heron's problem to a Euclidean space of arbitrary dimension $\mathbb{R}^n$ was introduced in [185], where the line was substituted by a closed convex set C_0 and the two given points by a finite family of closed convex sets $\{C_i\}_{i=1}^p$. This convex problem was then solved by means of a projected subgradient algorithm. Lately, different splitting methods have also been employed to tackle this generalization of Heron's problem [69, 81].

In this subsection, we go one step ahead and consider a more general version of the problem. Specifically, we seek to minimize a weighted sum of the squared distance of the images of x by certain differentiable functions $\Psi_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ with Lipschitz continuous gradients, for $i = 1, \ldots, p$. Namely, given some closed (but not necessarily convex) sets $C_0 \subseteq \mathbb{R}^n$ and $C_i \subseteq \mathbb{R}^{m_i}$, for $i = 1, \ldots, p$, we are interested in solving the following nonconvex generalization of the squared Heron's problem

$$\min_{x \in C_0} \sum_{i=1}^{p} \frac{w_i}{2} d^2(\Psi_i(x), C_i), \tag{8.58}$$

where $w_i > 0$ represents a weight associated to the i-th constraint.

Problem (8.58) can be easily reformulated as an instance of ($\mathcal{P}$). Indeed, as shown in Example 8.12, the squared distance function admits the following decomposition:

$$\frac{1}{2}d^2(\Psi_i(x), C_i) = \frac{1}{2}\|\Psi_i(x)\|^2 - \mathtt{A}_{C_i}(\Psi_i(x)).$$

Hence, problem (8.58) is equivalent to the unconstrained problem

$$\min_{x\in\mathbb{R}^n} \iota_{C_0}(x) + \sum_{i=1}^{p} w_i \left(\frac{1}{2}\|\Psi_i(x)\|^2 - \mathtt{A}_{C_i}(\Psi_i(x))\right).$$

It is clear that this problem can be expressed in the form of ($\mathcal{P}$) with the choice of functions $f := \sum_{i=1}^{p} \frac{w_i}{2}\|\Psi_i(\cdot)\|^2$, $g = \iota_{C_0}$ and $h_i = w_i\mathtt{A}_{C_i}$, for $i = 1, \ldots, p$. Note that, although the Asplund function $\mathtt{A}_{C_i}$ is always convex, $\mathtt{A}_{C_i} \circ \Psi_i$ may not be convex if Ψ_i is not linear. Therefore, $\frac{w_i}{2}d^2(\Psi_i(x), C_i)$ is not necessarily upper-$\mathcal{C}^2$.

In the following, we analyze the performance of DSA and BDSA for the particular instance of the problem in which $w_i := 1$ for all $i = 1, \ldots, p$, C_0 is the closed ball of radius $r_{C_0} := 5$ in $\mathbb{R}^n$, and the soft constraints C_i are hypercubes of edge length 2. To avoid intersections with C_0, the centroids of the hypercubes were randomly generated with norm between 7 and 10. We set all $\Psi_i := \Psi : \mathbb{R}^n \to \mathbb{R}^m$, with

$$\Psi(x) := \left(x^TQ_1x, x^TQ_2x, \ldots, x^TQ_mx\right)^T,$$

where, for simplicity, we chose $Q_1, \ldots, Q_m$ as diagonal matrices with randomly generated entries in $]-1, 1[$. Note, that the gradient of Ψ is the linear transformation given by

$$\nabla\Psi(x) = 2(Q_1x, Q_2x, \ldots, Q_mx),$$

which is Lipschitz continuous with constant $L_\Psi := 2\rho(Q)$, where $\rho(Q)$ denotes the spectral radius of $Q := (Q_1, Q_2, \ldots, Q_m)$. On the other hand, note that ∇f is also L_f-Lipschitz in the ball C_0 for $L_f := 6pr_{C_0}^2\rho(Q)\sqrt{\sum_{i=1}^{m}\rho(Q_i)^2}$, since

$$\begin{aligned}
\|\nabla f(x) - \nabla f(y)\| &= \|p\left(\nabla\Psi(x)\Psi(x) - \nabla\Psi(y)\Psi(y)\right)\| \\
&\leq p\|\nabla\Psi(x)\Psi(x) - \nabla\Psi(x)\Psi(y)\| + p\|\nabla\Psi(x)\Psi(y) - \nabla\Psi(y)\Psi(y)\| \\
&\leq p\|\nabla\Psi(x)\|\sqrt{\sum_{i=1}^{m}(x^TQ_ix - y^TQ_iy)^2} + p\|\Psi(y)\|L_\Psi\|x - y\|
\end{aligned}$$

$$\leq 2\rho(Q)p\left(\|x\|\sqrt{\sum_{i=1}^{m} 4r_{C_0}^2\rho(Q_i)^2} + \max_{z\in C_0}\|\Psi(z)\|\right)\|x-y\|$$
$$\leq 2\rho(Q)p\left(\|x\|2r_{C_0}\sqrt{\sum_{i=1}^{m}\rho(Q_i)^2} + \max_{z\in C_0}\sqrt{\sum_{i=1}^{m}\|z\|^4\ \rho(Q_i)^2}\right)\|x-y\|$$
$$\leq 6\rho(Q)pr_{C_0}^2\sqrt{\sum_{i=1}^{m}\rho(Q_i)^2}\|x-y\| = L_f\|x-y\|.$$

Therefore, according to (8.5) and Proposition 8.9, the function f is $L_f/2$-upper-$\mathcal{C}^2$.

In order to fairly illustrate the advantages of the linesearch step in BDSA, we initially perform an experiment to find some adequate performing parameters for DSA (i.e., when no linesearch was performed). In all the experiments in this subsection, we stopped the algorithms when

$$|\Phi(x^{k+1}, y^{k+1}) - \Phi(x^k, y^k)| < 10^{-6}. \tag{8.59}$$

Tuning the parameters for DSA We set $n = 3$, $m = 4$ and $p = 3$, and ran Algorithm 8.17 with different choices of parameters for 5 randomly generated problems and 5 different starting points for each problem (i.e., 25 instances in total). Having in mind the bounds for the parameters given in Theorem 8.24, we tested the algorithm for every combination of the following choices:

$$\gamma_k := \eta\left(L_f + L_\Psi\sum_{i=1}^{p}\|y_i^k\|\right)^{-1},\ \text{with } \eta\in\{0.1, 0.3, 0.5, 0.7, 0.9, 0.99\},$$
$$\mu_i^k := \mu\in\{0.5, 1, 5\},\ \text{for all } k\geq 0.$$

The algorithm always obtained the same value of Φ in the last iterate independently of the combination of stepsize parameters chosen. However, there is a considerable variability in the number of iterations needed for reaching the stopping criterion, which we show in Table 8.4. Note that the parameter γ_k is the only one providing significant differences, being $\eta = 0.99$ the best performing value. The parameter μ does not seem to have much influence in the results obtained.

	$\eta = 0.1$	$\eta = 0.3$	$\eta = 0.5$	$\eta = 0.7$	$\eta = 0.9$	$\eta = 0.99$
$\mu = 0.5$	8 013.0	3 104.8	1 989.0	1 486.6	1 201.5	1 104.9
$\mu = 1$	8 012.4	3 103.7	1 987.6	1 485.0	1 199.9	1 103.4
$\mu = 5$	8 011.7	3 102.9	1 986.5	1 483.9	1 198.6	1 102.1

TABLE 8.4: Average number of iterations of DSA for 5 random problems (8.58) and 5 random starting points for each problem, with $n = 3$, $m = 4$, $p = 3$ and different values of the parameters.

DSA vs. BDSA: Benefit of linesearches Now we compare both versions of Algorithm 8.17 with the best choice parameter $\eta = 0.99$. Since μ does not have much effect, we set a small value $\mu^k = 0.5$, which is more likely to satisfy the bound in Proposition 8.21 (iii). We tested DSA and BDSA for different values of n, m and p. Both algorithms obtained similar results regarding the objective function, showing only differences after the second decimal, in favor of BDSA in all but one instance, so we only present the results regarding number of iterations (without counting those needed for the linesearch) and the running time. The results are summarized in Figures 8.8 and 8.9, where we observe that BDSA clearly outperformed DSA in each of the 120 instances. In particular, BDSA was on average more than 2.5 times faster than DSA.

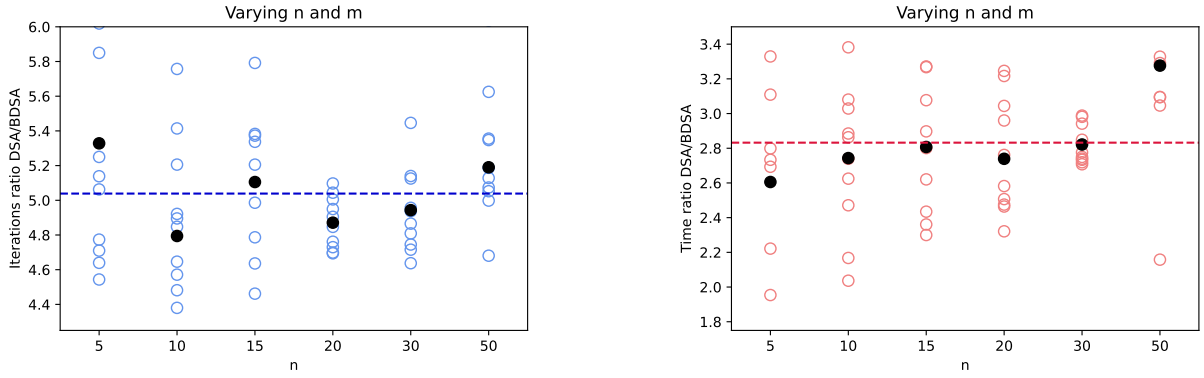


FIGURE 8.8: For $p = 3$, each $n \in \{5, 10, 15, 20, 30, 50\}$ and $m = 1.2\,n$, we randomly generated 10 different problems and ran DSA and BDSA initialized at the same random starting point. We plot with circles the ratio of the number of iterations (left) and the running time (right). The black dots represent the average ratio for a fixed n and the dashed line the overall average ratio.

The generalized squared Heron Problem with nonconvex sets In our last experiment, we consider examples of (8.58) in which the sets C_i are not necessarily convex. Instances of the generalized Heron problem with nonconvex sets have already been studied, for example in [184]. In this experiment, we let Ψ_i be a linear mapping of the form

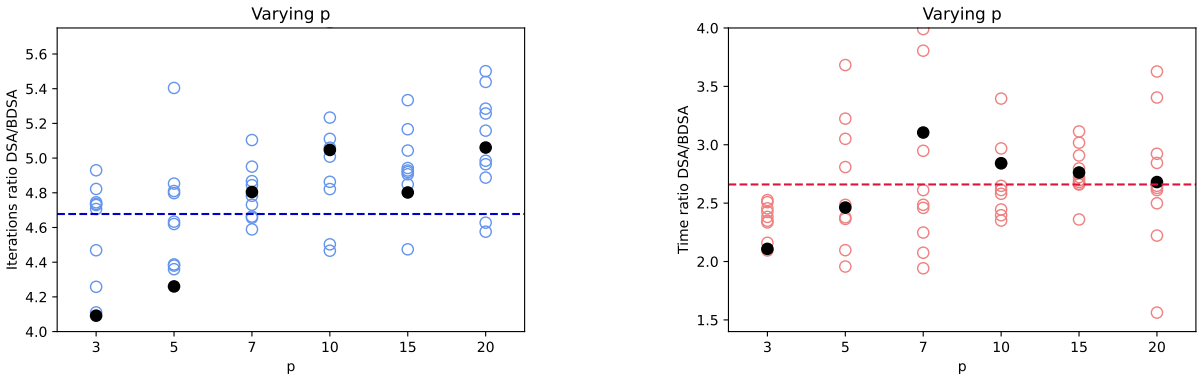


FIGURE 8.9: For $n = 20$, $m = 16$ and each $p \in \{3, 5, 7, 10, 15, 20\}$, we randomly generated 10 different problems and ran DSA and BDSA initialized at the same random starting point. We plot with circles the ratio of the number of iterations (left) and the running time (right). The black dots represent the average ratio for a fixed n and the dashed line the overall average ratio.

$\Psi_i(x) = Qx$, with $Q \in \mathbb{R}^{m \times n}$, for all $i = 1, \dots, p$. We showed in Example 8.12 that $x \mapsto \frac{1}{2} d^2(Qx, C)$ is a κ-upper-$\mathcal{C}^2$ function with constant $\kappa = \rho(Q)^2/2$. Moreover, note that by [28, Theorem 5.3 (iii)] its subdifferential at a point $x \in \mathbb{R}^n$ is given by

$$\partial \left(\frac{1}{2} d^2(Q(\cdot), C) \right)(x) = Q^T \Big(Qx - P_C(Qx) \Big).$$

These two facts allow us to tackle (8.58) when C_i are not necessarily convex as an instance of problem $(\mathcal{P})$ by setting $f := \sum_{i=1}^{p} \frac{w_i}{2} d^2(Q(\cdot), C_i)$, $g = \iota_{C_0}$ and $h = 0$.

We work with the particular instance of (8.58) in which $p = 1$, $w_1 = 1$ and C_1 is given as the union of 5 hypercubes which were generated in the same way as in the previous experiment. The entries of Q were randomly generated in the interval $]-1, 1[$. As in the previous experiment, up to the authors' knowledge, in this setting BDSA does not recover any method already proposed in the literature. In Figure 8.10 we show the results of running both DSA and BDSA for 8 different dimensions and 10 different randomly generated problems for each dimension. In this case, BDSA reached a better value of the objective function than DSA in every instance. Regarding the comparison in iterations and time, BDSA was again significantly faster: on average, DSA needed around 5 times more iterations and 2.5 more running time than BDSA to satisfy the stopping criterion (8.59).

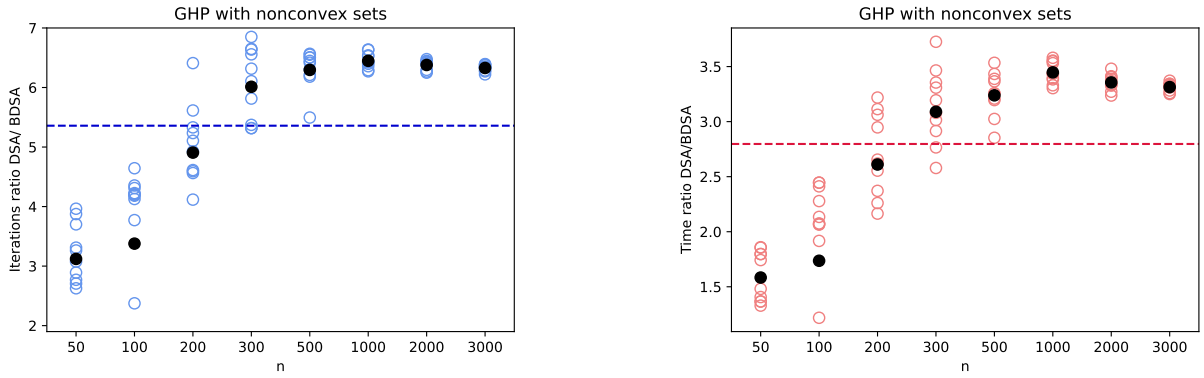


FIGURE 8.10: Let $n \in \{50, 100, 200, 300, 500, 1\,000, 2\,000, 3\,000\}$ and $m = 1.2\,n$. For every choice of dimensions we run both algorithms for 10 different randomly generated problems and initialized at the same random starting point. The unfilled markers show the ratio in iterations and CPU time of DSA over BDSA for every particular instance, the black dots represent the ratio average for a fixed n and the dashed line the overall ratio average.

Chapter 9

The superiorization method with restarted perturbations for split minimization problems with an application to radiotherapy treatment planning

In a fair number of applications the nature and size of the arising constrained optimization problems make it computationally difficult, or sometimes even impossible, to obtain exact solutions and alternative ways of handling the data of the optimization problem should be considered. A common approach is the regularization technique that replaces the constrained optimization problem by an unconstrained optimization one wherein the objective function is a linear combination of the original objective and a regularization term that "measures" in some way the constraint violations.

This approach is used for constrained minimization problems appearing in image processing, where the celebrated *Fast Iterative Shrinkage-Thresholding Algorithm* (FISTA) method was pioneered by Beck and Teboulle [55]. In situations when there are some constraints whose satisfaction is imperative ("hard constraints") the problem can be considered as being composed of two goals: a major goal of satisfying the constraints and a secondary, but desirable, goal of target (a.k.a. objective, merit, cost) function reduction.

In this setting, the *Superiorization Methodology* (SM) has proven capable of efficiently handling the data of very large constrained optimization problems. The idea behind superiorization is to apply a feasibility-seeking algorithm and introduce in each of its iterations a certain change, referred to as a *perturbation*, whose aim is to reduce the value of the target function. When the feasibility-seeking iterative method is *bounded perturbation*

resilient (see Definition 9.4 below), the superiorized version of the feasibility-seeking algorithm will converge to a feasible solution which is expected to have a reduced, not necessarily minimal, target function value.

In this chapter, we propose a novel superiorized algorithm for dealing with the data of the following *split minimization problem:*

Problem 9.1 (The Split Minimization Problem (SMP)). *Given two nonempty, closed and convex subsets $C \subseteq \mathbb{R}^n$ and $Q \subseteq \mathbb{R}^m$, an $m \times n$ real matrix A, and convex functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}$, find*

$$x^* \in C \text{ such that } x^* \in \operatorname{argmin}\{f(x) \mid x \in C\}, \quad \text{and such that} \tag{9.1}$$

$$y^* := Ax^* \in Q \text{ and } y^* \in \operatorname{argmin}\{g(y) \mid y \in Q\}. \tag{9.2}$$

It is important to observe that the two objective functions f and g in (9.1) and (9.2) may conflict with each other, and thus the existence of a solution to Problem 9.1 is not guaranteed even if $Ax \in Q$ for all $x \in C$. This is a new genre of problems which are not considered as multi-objective but rather split between two spaces. Problem 9.1 is a particular instance of the *Split Variational Inequality Problem* (SVIP), which employs, instead of the minimization problems in (9.1) and (9.2), variational inequalities. The SVIP, see [91], entails finding a solution of one Variational Inequality Problem (VIP), the image of which under a given bounded linear transformation is a solution of another VIP. Algorithms for solving the SVIP require computing the projections onto the corresponding constraint sets at every step, see [91]. In the case when C and Q are each given by an intersection of nonempty, closed and convex sets, auxiliary algorithms, such as Dykstra's algorithm [71] (see also [44, Subsection 30.2]), the *Halpern–Lions–Wittmann–Bauschke algorithm* [40] (see also [44, Subsection 30.1]) or the *averaged alternating modified reflections method* [14, 80] are needed for computing/approximating these projections, which will considerably increase the running time and the numerical errors of the algorithms. In this work, we do not aim to find an exact solution of the SMP, but rather obtain a feasible solution with reduced values of the objective functions f and g. This allows us to drop the usual assumption on the existence of a solution to the minimization problem, and instead we will only require that the set of solutions to the associated feasibility problem (see, Problem 9.11) is nonempty.

Our main findings are the following:

- In Section 9.2, we introduce a new technique for setting up the stepsizes in the perturbations of the SM, which results in a new version of the general structure of the

superiorized algorithm. This new scheme increases the efficiency of the superiorized algorithm by allowing restarts of stepsizes.

- Section 9.3 contains a new algorithm for dealing with the data of the split minimization problem that has the ability to superiorize independently over subvectors.
- Our contributions cater to real-world situations, as we illustrate in Section 9.4.3 for a nontrivial realistic problem arising in intensity-modulated radiation therapy treatment planning.

This chapter is based on the published work [23].

9.1 The superiorization methodology

In this section we present a brief introduction to the superiorization methodology (SM), which is a simplified version of the presentation in [86]. The SM has been shown to be a useful tool for handling the data of difficult constrained minimization problems of the form

$$\min \ \phi(x) \quad \text{s.t.} \quad x \in C,$$

where $\phi : \mathbb{R}^n \to \mathbb{R}$ is a target function and $C \subseteq \mathbb{R}^n$ is a nonempty feasible set, generally presented as an intersection of a finite family of constraint sets $C := \bigcap_{s=1}^{p} C_s$. When $\{C_s\}_{s=1}^{p}$ is a collection of nonempty, closed and convex sets in $\mathbb{R}^n$, there is a wide range of projection methods (see, e.g., [44, 80, 83]) that can be employed for solving the convex feasibility-seeking problem

$$\text{find } x^* \in C := \bigcap_{s=1}^{p} C_s. \tag{9.3}$$

The first building brick of the SM is an iterative feasibility-seeking algorithm, often a projection method, which is referred to as the *basic algorithm*, capable of (asymptotically) finding a solution to (9.3). This algorithm employs an *algorithmic operator* $\mathbf{T}_C : \mathbb{R}^n \to \mathbb{R}^n$ in the following iterative process.

Algorithm 9.2 The basic algorithm.

1: **Initialization:** Choose an arbitrary initialization point $x^0 \in \mathbb{R}^n$.
2: **Iterative Step:** Given the current iterate x^k, calculate the next iterate x^{k+1} by

$$x^{k+1} = \mathbf{T}_C(x^k).$$

Example 9.3. *A well-known feasibility-seeking algorithm for the set C given in (9.3) is the method of* sequential alternating projections *[240], whose algorithmic operator is given by*

$$\mathbf{T}_C := P_{C_p} P_{C_{p-1}} \cdots P_{C_2} P_{C_1}. \tag{9.4}$$

Many other iterative feasibility-seeking projection methods are available, see, e.g., the excellent review paper of Bauschke and Borwein [42] and [135]. Such methods have general algorithmic structures of *Block-Iterative Projection* (BIP), see, e.g., [2, 3] or *String-Averaging Projections* (SAP), see, e.g., [89, 93, 192].

In the SM one constructs from the basic algorithm a "superiorized version of the basic algorithm" which includes perturbations of the iterates of the basic algorithm. This requires the basic algorithm to be resilient to certain perturbations. The definition is given next with respect to the feasibility-seeking operator (9.4), but is phrased in the literature with algorithmic operators of any basic algorithm.

Definition 9.4 (Bounded perturbation resilience). *Let $\{C_s\}_{s=1}^{p}$ be a family of closed and convex sets in $\mathbb{R}^n$ such that $C = \bigcap_{s=1}^{p} C_s$ is nonempty. An algorithmic operator $\mathbf{T}_C : \mathbb{R}^n \to \mathbb{R}^n$ for solving the feasibility-seeking problem associated with C is said to be* bounded perturbation resilient *if the following holds: for all $x^0 \in \mathbb{R}^n$, if the sequence $\{x^k\}_{k=0}^{\infty}$ generated by Algorithm 1 converges to a solution of the feasibility-seeking problem, then any sequence $\{y^k\}_{k=0}^{\infty}$ generated by*

$$y^{k+1} = \mathbf{T}_C \left(y^k + \eta_k v^k \right), \quad \forall k \in \mathbb{N}, \tag{9.5}$$

for any $y^0 \in \mathbb{R}^n$, where the vector sequence $(v^k)_{k\in\mathbb{N}}$ is bounded and the scalars $(\eta_k)_{k\in\mathbb{N}}$ are nonnegative and summable, i.e., $\sum_{k=0}^{\infty} \eta_k < \infty$, also converges to a solution of the feasibility-seeking problem.

The property of bounded perturbation resilience has been validated for two major prototypical algorithmic operators that give rise to the string averaging projections method and the block iterative projections method mentioned above, see [79] and [127], respectively. These schemes include many well-known projection algorithms, such as the method of alternating projections and Cimmino's algorithm. The convexity and closedness of the sets C_s is present in the results proving the bounded perturbation resilience of the BIP algorithms and the SAP methods.

The importance of bounded perturbation resilience for the SM stems from the fact that it allows to include perturbations in the iterative steps of the basic algorithm without

compromising its convergence to a feasible solution, while steering the algorithm toward a feasible point with a reduced (not necessarily minimal) value of the target function.

The fundamental idea underlying the SM is to use the bounded perturbations in (9.5) in order to induce convergence to a feasible point which is *superior*, meaning that the value of the target function ϕ is smaller or equal than that of a point obtained by applying the basic algorithm alone without perturbations. To achieve this aim, the bounded perturbations in (9.5) should imply that

$$\phi(y^k + \eta_k v^k) \leq \phi(y^k), \quad \text{for all } k \in \mathbb{N}. \tag{9.6}$$

To do so, the sequence $(v^k)_{k\in\mathbb{N}}$ is chosen according to the next definition, which is closely related to the concept of *descent direction*.

Definition 9.5. *Given a function $\phi : \mathbb{R}^n \to \mathbb{R}$ and a point $y \in \mathbb{R}^n$, we say that a direction $v \in \mathbb{R}^n$ is* nonascending for ϕ at y *if $\|v\| \leq 1$ and there is some $\delta > 0$ such that*

$$\phi(y + \lambda v) \leq \phi(y), \quad \textit{for all } \lambda \in [0, \delta].$$

Obviously, the zero vector is a nonascending direction. However, it would not provide any perturbation of the sequence in (9.5). Denoting by $\frac{\partial \phi}{\partial x_i}(x)$ the partial derivatives, the next result provides a formula for obtaining nonascending vectors of convex functions.

Fact 9.6 ([148, Theorem 2]). *Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a convex function and let $x \in \mathbb{R}^n$. Let $u = (u_i)_{i=1}^n \in \mathbb{R}^n$ be defined, by*

$$u_i := \begin{cases} \frac{\partial \phi}{\partial x_i}(x), & \textit{if } \frac{\partial \phi}{\partial x_i}(x) \textit{ exists}, \\ 0, & \textit{otherwise}, \end{cases}$$

and define

$$v := \begin{cases} 0, & \textit{if } \|u\| = 0, \\ -u/\|u\|, & \textit{otherwise}. \end{cases}$$

Then v is a nonascending vector for ϕ at the point x.

Next we present the pseudo-code of the iterative process governing the superiorized version of the basic algorithm.

Algorithm 9.7 The superiorized version of the basic algorithm.

1: **Initialization:** Choose an arbitrary initialization point $y^0 \in \mathbb{R}^n$, a summable nonnegative sequence $(\eta_\ell)_{\ell\in\mathbb{N}}$ and a positive integer N.
2: Set $k := 0$ and $\ell := -1$.
3: **repeat**
4: Set $y^{k,0} = y^k$.
5: **for** $j = 0$ **to** $N - 1$ **do**
6: Choose a nonascending vector $v^{k,j}$ for ϕ at $y^{k,j}$.
7: Set $\ell := \ell + 1$.
8: **while** $\phi(y^{k,j} + \eta_\ell v^{k,j}) > \phi(y^k)$ **do**
9: Set $\ell := \ell + 1$.
10: **end while**
11: Set $y^{k,j+1} = y^{k,j} + \eta_\ell v^{k,j}$.
12: **end for**
13: Update $y^{k+1} = \mathbf{T}_C\left(y^{k,N}\right)$ and set $k := k + 1$.

The choice of nonascending vectors guarantees that the **while** loop in lines 8–10 is finite (see [148] for a complete proof on the termination of the algorithm). When a bounded perturbation resilient operator $\mathbf{T}_C$ is chosen as the basic algorithm, Algorithm 9.7 will converge to a solution of the feasibility-seeking problem. Moreover, it is expected that the perturbations in line 11, which reduce at each inner-loop step the value of the target function ϕ (line 8), will drive the iterates of Algorithm 9.7 to an output which will be superior (from the point of view of its ϕ value) to the output that would have been obtained by the original unperturbed basic algorithm.

9.1.1 The volatile nature of the superiorization methodology

In general, there is no theoretical guarantee that the point obtained by the superiorized algorithm would be superior to the one reached with the basic algorithm. The performance of the SM might be influenced on different factors. In what follows, we present a simple example where the ability of the SM to improve the output from the basic algorithm depends on the choice of the initialization point.

Consider the problem of finding the minimum norm point in the intersection of two half-spaces $A := \{x \in \mathbb{R}^2 \mid x_1 + x_2 \geq 1\}$ and $B := \{x \in \mathbb{R}^2 \mid x_1 - x_2 \leq 0\}$. If we use the method of alternating projections as the basic algorithm with starting point $y^0 := (3/10, 0)^T$, one obtains $y^1 = (1/2, 1/2)^T$, which is the solution to the problem. In Figure 9.1 (left) we show 50 iterations of its superiorized version for the same starting point with $\phi(y) := \|y\|^2$, step-sizes in the sequence $(2^{-\ell})_{\ell\in\mathbb{N}}$, $N = 1$ and taking as

nonascending direction $v^k := -\frac{\nabla\phi(y^k)}{\|\nabla\phi(y^k)\|} = -\frac{y^k}{\|y^k\|}$.

From this starting point the basic algorithm for feasibility-seeking alone without perturbations (i.e., the alternating projection method) converges in one iteration to the minimum norm point in the intersection of the two half-spaces, while its superiorized version remains far from the solution after 50 iterations. This happens because the first perturbation applied to this y^0 results in a point on the horizontal axis inside the set A. If we were to choose another y^0 on the x-axis, but far enough to the right inside the set B, then after one perturbation the next point will be outside both sets A and B on the positive x-axis and this would lead, following a single iteration of feasibility-seeking, to the minimum norm point whereas the feasibility-seeking-only from that y^0 onward would lead to a less "superior" feasible point, as shown in Figure 9.1 (right).

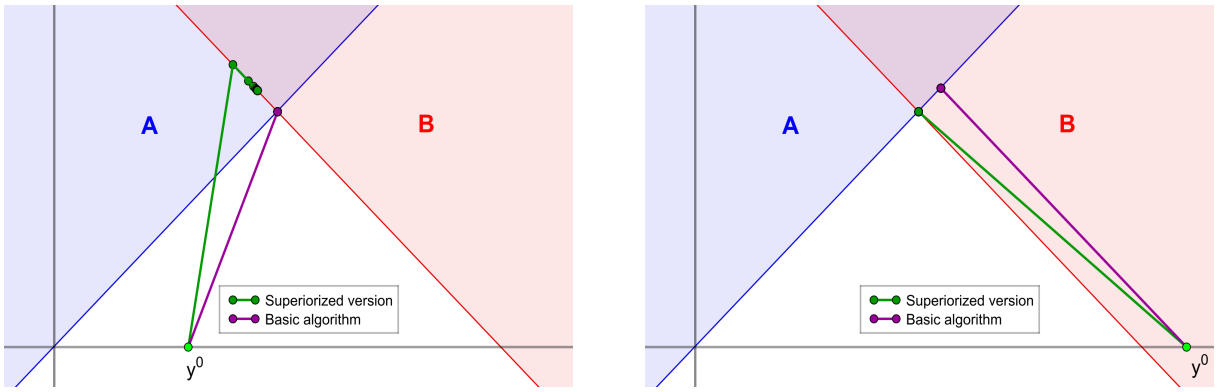


FIGURE 9.1: (Left) Taking $y^0 = (3/10, 0)^T$ as starting point, the alternating projection method converges in one iteration to the minimum norm point in the intersection of two half-spaces, while its superiorized version remains far from the solution after 50 iterations. (Right) If we take $y^0 = (11/10, 0)^T$, in one iteration, the superiorized version converges to the minimum norm point while the alternating projection method converges to a feasible point which is not the solution.

Observe that we only computed 50 iterations because after them the norm of the perturbations is smaller than $2^{-50} \approx 8.9 \cdot 10^{-16}$. Hence, the effect of the perturbations steering the basic algorithm to a superiorized solution vanishes, having no real effect on it. This phenomenon is inherent in the SM and the purpose of the restarts, proposed in the next section, is to improve this unwanted behavior.

Proving mathematically a guarantee of global function reduction of the SM will probably require some additional assumptions on the feasible set, the target function, the parameters involved, or even on the initialization points. In [92, Section 3] the authors gave a precise definition of the "guarantee problem" of the SM: "A mathematical guarantee has not been found to date that the overall process of the superiorized version of

the basic algorithm will not only retain its feasibility-seeking nature but also accumulate and preserve globally the target function reductions. We call this fundamental question of the SM '*the guarantee problem of the SM*' which is: under which conditions one can guarantee that a superiorized version of a bounded perturbation resilient feasibility-seeking algorithm converges to a feasible point that has target function value smaller or equal to that of a point to which this algorithm would have converged if no perturbations were applied – everything else being equal."

Numerous works that are cited in [84] show that this global function reduction of the SM occurs in practice in many real-world applications. In addition to a partial answer in [92] with the aid of the "concentration of measure" principle there is also the partial result of [96, Theorem 4.1] about strict Fejér monotonicity of sequences generated by an SM algorithm.

9.2 The superiorization method with perturbations restarts

We offer here a modification of the SM, applied to the superiorized version of the basic algorithm in Algorithm 9.7, by setting the perturbations stepsizes in a manner that allows restarts. Commonly, the summable sequence $(\eta_\ell)_{\ell\in\mathbb{N}}$ employed in Algorithm 9.7 is generated by taking a real number $\alpha \in]0,1[$, referred to as *kernel*, and setting $\eta_\ell := \alpha^\ell$, for $\ell \geq 0$. This strategy works well in practice, as witnessed by many works cited in [84], but it has though the inconvenience that, as the iterations progress, the stepsizes in (9.5) decrease toward zero quite fast, yielding insignificant perturbations.

In some applications, various methods have been studied for controlling the stepsizes, see, e.g., [85, 158, 208], see also the software package SNARK14 [193] which is an updated version of [155]. We propose here a new strategy which allows *restarting* the sequence of stepsizes to a previous value while maintaining the summability of the series of stepsizes. The restarting of stepsizes is a useful approach that allows to improve an algorithm's performance, see, e.g. [168, 197], where this technique is applied to the stochastic gradient descent and FISTA, respectively.

Our proposed scheme for restarting the stepsizes is controlled by a sequence of positive integers $\left(W_r\right)_{r\in\mathbb{N}}$, where we call the indices $r = 1, 2, \ldots$ *restart indices*. Bearing some similarity to a backtracking scheme, the initial stepsize at the beginning of a new loop of perturbations is reduced after each restart. Specifically, let $\alpha \in]0,1[$ be any fixed kernel, assume that r restarts have been already performed and let W_r denote the number of consecutive stepsizes that must be taken before allowing the next restart.

The algorithm begins with $r = 0$ and takes W_0 decreasing stepsizes in the sequence $\{1, \alpha, \alpha^2, \ldots\}$. After these W_0 stepsizes, the algorithm performs a restart in the stepsizes by setting $r = 1$ and taking anew W_1 decreasing stepsizes in the sequence $\{\alpha, \alpha^2, \alpha^3, \ldots\}$. Then the algorithm performs another restart with $r = 2$ and uses stepsizes in the sequence $\{\alpha^2, \alpha^3, \alpha^4, \ldots\}$, and so on.

This is accurately described in the pseudo-code of the *superiorized version of the basic algorithm with restarts* presented below. Observe that there are no restrictions on how the sequence $\left(W_r\right)_{r\in\mathbb{N}}$ is chosen. A simple possible choice is to take a positive constant value $W_r := W$, for all $r \in \mathbb{N}$.

Note that, since the kernel α needs to lie in the interval $]0, 1[$, despite performing restarts, the perturbations may happen to yield an insignificant decrease in the objective value, specially if the norm of the nonascending vectors is close to zero. This can be controlled by considering a positive real number c and performing the restarts to the sequence $(c\,\alpha^\ell)_{\ell\in\mathbb{N}}$ rather than to $(\alpha^\ell)_{\ell\in\mathbb{N}}$. This parameter is also included in the pseudo-code of Algorithm 9.8.

Algorithm 9.8 The superiorized version of the basic algorithm with restarts.

Initialization: Choose an arbitrary initialization point $y^0 \in \mathbb{R}^n$, $\alpha \in]0, 1[$, a positive number c, a positive integer N and a sequence of positive integers $\left(W_r\right)_{r\in\mathbb{N}}$.
Set $k := 0$, $\ell := -1$, $w := 0$ and $r := 0$.
repeat
    Set $y^{k,0} = y^k$.
    for $j = 0$ **to** $N - 1$ **do**
        Choose a nonascending vector $v^{k,j}$ for ϕ at $y^{k,j}$.
        Set $\ell := \ell + 1$.
        while $\phi(y^{k,j} + c\,\alpha^\ell v^{k,j}) > \phi(y^k)$ **do**
            Set $\ell := \ell + 1$.
        end while
        Set $y^{k,j+1} = y^{k,j} + c\,\alpha^\ell v^{k,j}$.
    end for
    Set $w := w + 1$.
    if $w = W_r$ **then**
        Set $r := r + 1$, $\ell := r$ and $w := 0$.
    end if
    Update $y^{k+1} = \mathbf{T}_C\left(y^{k,N}\right)$ and set $k := k + 1$.

Remark 9.9. The strategy of restarts in Algorithm 9.8 preserves the summability of the overall series of stepsizes. This is so because even if during the iterative process the largest

stepsizes allowed in each of the sets W_r were taken, the infinite sequence of all stepsizes,

$$\left(\left(\alpha^{r+\ell}\right)_{\ell=0}^{W_r-1}\right)_{r=0}^{\infty}, \tag{9.7}$$

would still form a bounded series. Indeed, since $\alpha \in]0,1[$, we have that

$$\sum_{r=0}^{\infty}\sum_{\ell=0}^{W_r-1}\alpha^{r+\ell} \leq \sum_{r=0}^{\infty}\alpha^{r}\sum_{\ell=0}^{\infty}\alpha^{\ell} = \frac{1}{(1-\alpha)^2}.$$

Hence, since only stepsizes leading to expected superior values of the target function are allowed by (9.6) (line 8 of Algorithm 9.8), each of the stepsizes taken will be smaller than the corresponding one in the sequence (9.7), so its sum will always be smaller than $1/(1-\alpha)^2$ and will, thus, define bounded perturbations.

For some applications, the SM with restarts is very useful, notably outperforming the current SM without restarts (see Section 9.4.3).

9.3 A superiorized algorithm for subvectors in the split minimization problem

We develop here a superiorized algorithm for tackling the data of the SMP in Problem 9.1 when $C := \bigcap_{s=1}^{p} C_s$ and $Q := \bigcap_{t=1}^{q} Q_t$, where p and q are two integers and $\{C_s\}_{s=1}^{p}$ and $\{Q_t\}_{t=1}^{q}$ are two families of closed and convex sets with nonempty intersections in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. To ease the discussion we will refer here to $\mathbb{R}^n$ and $\mathbb{R}^m$, as the "x-space" and the "y-space", respectively.

9.3.1 The SMP with subvectors

In some situations of practical interest, the minimization problem in (9.2) should be independently applied to subvectors of the y-space. We discuss an instance in the field of radiation therapy treatment planning where this is significant in Section 9.4 below.

For simplicity and without loss of generality, we assume that the subvectors are in consecutive order. The $m \times n$ real matrix A is divided into B blocks and is represented by

$$A := \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_B \end{pmatrix}, \tag{9.8}$$

where, for each $b = 1, \ldots, B$, the matrices $A_b \in \mathbb{R}^{m_b \times n}$ are blocks of rows of the matrix A, with $\sum_{b=1}^{B} m_b = m$. Thus, any vector $y := Ax$ is of the form

$$y = \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^B \end{pmatrix} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_B \end{pmatrix} x,$$

where $y^b \in \mathbb{R}^{m_b}$ are subvectors of $y \in \mathbb{R}^m$.

Problem 9.10 (The SMP with subvectors). *Given two families of closed and convex sets $\{C_s\}_{s=1}^{p} \subseteq \mathbb{R}^n$ and $\{Q_t\}_{t=1}^{q} \subseteq \mathbb{R}^m$ such that $C := \bigcap_{s=1}^{p} C_s \neq \emptyset$ and $Q := \bigcap_{t=1}^{q} Q_t \neq \emptyset$, an $m \times n$ real matrix A in the form (9.8) for given positive integer B, a convex function $f : \mathbb{R}^n \to \mathbb{R}$ and convex functions $\phi_b : \mathbb{R}^{m_b} \to \mathbb{R}$, for $b = 1, 2, \ldots, B$, find*

$$x^* \in C \text{ such that } x^* \in \operatorname{argmin}\{f(x) \mid x \in C\}, \quad \text{and such that} \tag{9.9}$$

$$y^* := Ax^* \in Q \text{ and } y^{b*} \in \operatorname{argmin}\{\phi_b(y^b) \mid y \in Q\}, \quad \text{for all } b \in \{1, \ldots, B\}. \tag{9.10}$$

Our algorithm, presented below, can also handle subvectors in the x-space, but for simplicity we restrict ourselves here to subvectors in the y-space.

9.3.2 Reformulation in the product space

To work out a superiorization method for the data of the SMP with subvectors in Problem 9.10 we look at a *Multiple Sets Split Feasibility Problem* (MSSFP), see, e.g., [175], as follows.

Problem 9.11 (The Multiple Sets Split Feasibility Problem (MSSFP)). *Given* $C := \bigcap_{s=1}^{p} C_s$ *and* $Q := \bigcap_{t=1}^{q} Q_t$, *where* p *and* q *are two integers, and* $\{C_s\}_{s=1}^{p}$ *and* $\{Q_t\}_{t=1}^{q}$ *are two families of closed and convex sets with nonempty intersections each in* $\mathbb{R}^n$ *and* $\mathbb{R}^m$, *respectively, and an* $m \times n$ *real matrix* A, *find*

$$x^* \in C = \bigcap_{s=1}^{p} C_s \quad \textit{such that} \quad y^* := Ax^* \in Q = \bigcap_{t=1}^{q} Q_t.$$

This is a generalization of the *Split Feasibility Problem* (SFP) that occurs when $p = q = 1$ in the MSSFP. The SFP, which plays an important role in signal processing, in medical image reconstruction and in many other applications, was introduced by Censor and Elfving [88] in order to model certain inverse problems. Since then, many iterative algorithms for solving the SFP have been proposed and analyzed. See, for instance, the references given in [211] or consult the section "A brief review of 'split problems' formulations and solution methods" in [76].

Our proposed algorithm deals with an equivalent reformulation of Problem 9.11 in the product space $\mathbb{R}^n \times \mathbb{R}^m$. Adopting once more the notation that quantities in the product space are denoted by boldface symbols, we define the sets

$$\mathbf{C} := \left(\bigcap_{s=1}^{p} C_s\right) \times \left(\bigcap_{t=1}^{q} Q_t\right) \quad \text{and} \quad \mathbf{V} := \{\mathbf{z} = (x, y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax = y\}.$$

By Proposition 2.35 the projection onto $\mathbf{V}$ is given by $P_{\mathbf{V}} = \mathrm{Id} - Z^T(ZZ^T)^{-1}Z$, with $Z := [A, -I]$, where I denotes the $m \times m$ identity matrix. Then Problem 9.11 is equivalent to the problem:

$$\text{find a point } \mathbf{z}^* \in \mathbf{C} \cap \mathbf{V}. \tag{9.11}$$

Without loss of generality, we assume that $p = q$, since otherwise the whole space (or one particular set) could be added repeatedly as a constraint until both indices are equal. Since the projection of a Cartesian product is the Cartesian product of the projections (see Proposition 2.21), the following implementation of the method of alternating projections can be employed to solve (9.11). We consider this as our basic algorithm for the superiorization method for subvectors, and gather it in Algorithm 9.12.

Algorithm 9.12 Basic algorithm for the MSSFP.

1: **Initialization:** Choose an arbitrary initialization point $x^0 \in \mathbb{R}^n$. Set $y^0 = Ax^0$.
2: **Iterative Step:** Given the current iterate $\mathbf{z}^k = (x^k, y^k)$, calculate the next iterate $\mathbf{z}^{k+1}$ by

$$\mathbf{z}^{k+1} = P_{\mathbf{V}}\left(\left(P_{C_p} \times P_{Q_p}\right) \cdots \left(P_{C_1} \times P_{Q_1}\right)(\mathbf{z}^k)\right). \tag{9.12}$$

In order to construct a superiorized version of Algorithm 9.12 that can cope with the data of the SMP with subvectors in Problem 9.10, we need to establish at each iteration some appropriate perturbations that will steer the algorithm to a superiorized solution. For this, we note that the vector y^k inside $\mathbf{z}^k$ in (9.12) is expressed as

$$y^k = \begin{pmatrix} y^{1,k} \\ y^{2,k} \\ \vdots \\ y^{B,k} \end{pmatrix} = \begin{pmatrix} A_1 x^k \\ A_2 x^k \\ \vdots \\ A_B x^k \end{pmatrix}.$$

Thus, we declare our perturbation vector to be

$$\begin{pmatrix} x^k \\ y^k \end{pmatrix} + \eta_k \begin{pmatrix} u^k \\ v^k \end{pmatrix}, \quad \text{for all } k \in \mathbb{N}, \tag{9.13}$$

where $\{\eta_k\}_{k=0}^{\infty}$ is a nonnegative summable sequence, u^k is a nonascending vector for f at x^k and

$$v^k = \begin{pmatrix} v^{1,k} \\ v^{2,k} \\ \vdots \\ v^{B,k} \end{pmatrix},$$

with each $v^{b,k}$ being a nonascending vector for ϕ_b at the point $A_b x^k$, for all $b = 1, 2, \ldots, B$. The complete pseudo-code of the superiorized version of the basic Algorithm 9.12 with perturbations of the form given by (9.13) is shown in Algorithm 9.13.

Since the method of alternating projections is bounded perturbation resilient [79], Algorithm 9.13 will converge to a solution of the feasibility problem (9.11). Moreover, by the nature of the SM, the algorithm is expected to converge to a point $\mathbf{z}^* = (x^*, y^*)$ which will be superior with respect to f for the component x in the x-space, and with respect to ϕ_b for the b-th subvector in the y-space, for $b = 1, 2, \ldots, B$.

Algorithm 9.13 Superiorized Algorithm for the data of the SMP with subvectors.

Initialization: Choose $x^0 \in \mathbb{R}^n$, a summable nonnegative sequence $\{\eta_\ell\}_{\ell=0}^{\infty}$ and a positive integer N.
Set $y^{b,0} = A_b x^0$ for $b \in [\![1, B]\!]$. Set $k := 0$ and $\ell := -1$.
repeat
    Set $x^{k,0} = x^k$.
    Set $y^{b,k,0} = y^{b,k}$ for $b \in [\![1, B]\!]$.
    for $j = 0$ **to** $N - 1$ **do**
        Take $v_x^{k,j}$ a nonascending vector for f at $x^{k,j}$.
        Take $v_y^{b,k,j}$ a nonascending vector for ϕ_b at the point $y^{b,k,j}$ for all b.
        Set $\ell := \ell + 1$.
        while $f(x^{k,j} + \eta_\ell v_x^{k,j}) > f(x^k)$ **or there exists** b **with**
            $\phi_b(y^{b,k,j} + \eta_\ell v_y^{b,k,j}) > \phi_b(y^{b,k})$ **do**
            Set $\ell := \ell + 1$.
        end while
        Set $x^{k,j+1} = x^{k,j} + \eta_\ell v_x^{k,j}$.
        Set $y^{b,k,j+1} = y^{b,k,j} + \eta_\ell v_y^{b,k,j}$.
    end for
    Update

$$(x^{k+1}, y^{1,k+1}, \ldots, y^{B,k+1}) = P_{\mathbf{V}}\Big(\big(P_{C_p} \times P_{Q_p}\big) \cdots \big(P_{C_1} \times P_{Q_1}\big) \big(x^{k,N}, y^{1,k,N}, \ldots, y^{B,k,N}\big) \Big).$$

    Set $k := k + 1$.

9.4 Numerical experiments

Our aim in this section is not to compare the superiorization method with constrained optimization methods. The SM is not a method intended to solve exact constrained optimization problems, although such comparisons were done elsewhere, see, e.g., [145] or [87]. Our goal is to show how the SM can be improved and this is achieved by comparing the SM with and without restarts and with and without perturbations. We present our results of numerical experiments performed on three different problems. The first two problems are simple illustrative examples. The computational performance of the SM with restart algorithms, proposed here, can be substantiated with exhaustive testing of the possible specific variants permitted by the general framework and their various user-chosen parameters. This should be done on larger problems, preferably within the context of a significant real-world application. Therefore, our third problem addresses an actual situation arising in the real-world application of Intensity-Modulated Radiation Therapy (IMRT) treatment planning.

The purpose of the first example is to illustrate the potential benefits of superiorization with restarts for finding a point with reduced norm in the intersection of two convex sets. We first consider the case of two balls and then explore the case of two half-spaces presented in Remark 9.1.1, in which superiorization did not achieve its purpose for a particular setting.

In the second example, we illustrate the behavior of Algorithm 9.13 in a simple setting with $C, Q \subset \mathbb{R}^2$, where each of the sets is an intersection of three half-spaces.

Finally, the last experiment shows the benefits of superiorization with restarts in a difficult realistic setting in IMRT, where a large-scale multiobjective optimization problem arises. All tests were run on a desktop of Intel Core i7-4770 CPU 3.40GHz with 32GB RAM, under Windows 10 (64-bit).

9.4.1 The benefits of superiorization with restarts

Consider the problem of finding the minimum norm point in the intersection of two balls A and B in the Euclidean 2-dimensional space, so $\phi : \mathbb{R}^2 \to \mathbb{R}$ is given by the squared norm, i.e., $\phi(x) := \frac{1}{2}\|x\|^2$ for $x \in \mathbb{R}^2$. The underlying feasibility problem can be solved by the method of alternating projections, which we chose as the basic algorithm. Hence, the feasibility-seeking algorithmic operator used in our computations is

$$\mathbf{T} = P_B P_A,$$

where P_A and P_B denote the projection operators onto the balls A and B, respectively. We tested the method of alternating projections, its superiorized version (with two different kernels $\alpha = 0.6$ and $\alpha = 0.999$) and its superiorized version with restarts (with $\alpha = 0.6$ and $W_r = 50$, for all $r \in \mathbb{N}$). We set $N = 1$ in all the superiorized algorithms. The nonascending directions were taken as $v^k := -\frac{\nabla\phi(y^k)}{\|\nabla\phi(y^k)\|} = -\frac{y^k}{\|y^k\|}$.

The behavior of these algorithms is shown in Figure 9.2. On the left, we represent 500 iterations generated by each algorithm. On the right, we plot the sequence of perturbations obtained before applying the algorithmic operator, that is, we draw the points $\left(y^{k,N}\right)_{k=0}^{500}$. This sequence coincides with the sequence of iterates $\left(y^k\right)_{k=0}^{500}$ in the case when no perturbations are performed at all and only the basic algorithm works, while it coincides with the sequence $\left(y^k + \eta_k v^k\right)_{k=0}^{500}$ for the superiorized algorithms.

As expected, the method of alternating projections converges to a point in the intersection which is not desirable according to the task of reducing the target function

value (the squared norm). Superiorization with kernel $\alpha = 0.6$ reaches a better point than the output of the basic algorithm, but is yet far from the solution to the problem. This might well be due to the step-sizes not being big enough for the perturbations to steer the algorithm to a proper function reduction.

Taking $\alpha = 0.999$ in the standard superiorized version of the algorithm results in a very slow convergence of the sequence, as can be observed on the right figure in Figure 9.2. These deficiencies are resolved by considering superiorization with restarts, which achieves fast convergence to a solution with reduced norm in the intersection.

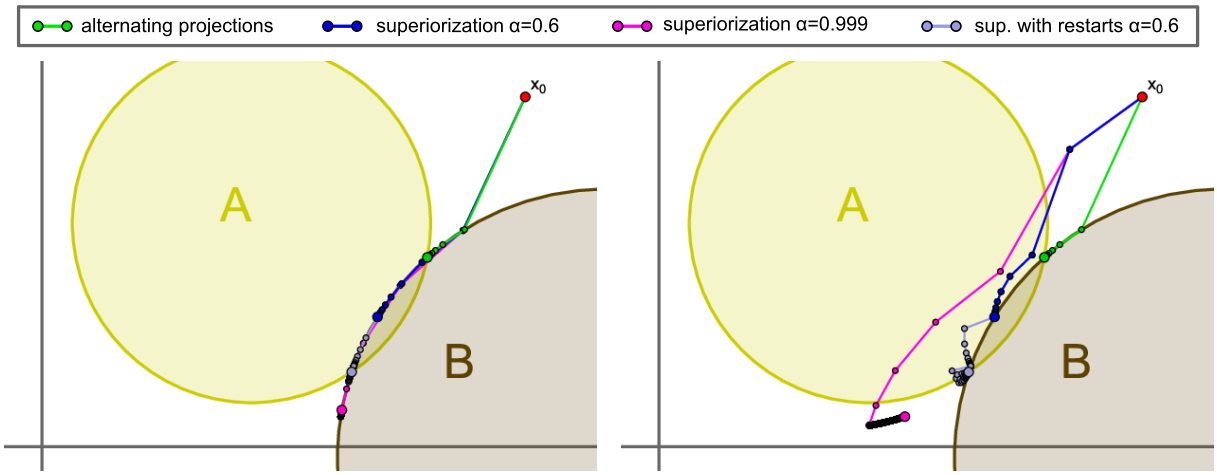


FIGURE 9.2: Behavior of the different algorithms considered applied to the data of the problem of finding the minimum norm point in the intersection of the balls A and B. The figures show the first 500 points in the sequence of iterates $\left(y^k\right)_{k=0}^{500}$ (left) and the sequence of perturbed iterations before applying the algorithmic operator $\left(y^{k,N}\right)_{k=0}^{500}$ (right) of each algorithm.

The example presented in Section 9.1.1 is artificial in the sense that the vectors defining the half-spaces are orthogonal and the starting point was chosen in a particular region of the plane which was less favorable to the superiorized algorithm. Also, the value of the kernel was chosen to be small ($\alpha = 0.5$), to aggravate the vanishing effect of the perturbations.

To study what happens with random data, we ran an experiment generating 1 million pairs of half-spaces $A := \{x \in \mathbb{R}^2 \mid \langle c_A, x\rangle \leq b_A\}$ and $B := \{x \in \mathbb{R}^2 \mid \langle c_B, x\rangle \leq b_B\}$ where the vectors $c_A, c_B \in \mathbb{R}^2$ were randomly chosen and then normalized, and $b_A, b_B \in]-1, 0[$ (to ensure that $(0,0)^T \notin A \cap B$). For each pair of half-spaces, we generated a random starting point $y^0 \in [-1,1]^2$ such that $y^0 \notin A \cap B$.

Then, we ran from y^0 the basic algorithm (feasibility-seeking alternating projections), its superiorized version with kernel $\alpha \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ and its superiorized version

with restarts with $W_r = 20$ and $N = 1$. The results are summarized in Table 9.1. In this table "AP" stands for "alternating projections", "Sup" stands for "the superiorized version", and "Sup. Res." stands for "the superiorized version with restarts". We count that one method is better than the other when the norm of its solution is smaller than the norm of the second method's output minus 10^{-3}. With kernel $\alpha = 0.5$, the superiorized algorithm failed to obtain a solution with lower norm than the basic algorithm in 12 931 of the 1 million instances; with kernel $\alpha = 0.9$, this number was reduced to 122. The superiorized algorithm with restarts with kernel $\alpha = 0.7$ only failed to get a better solution than the basic algorithm in 21 instances. Remarkably, when the kernels $\alpha \in \{0.8, 0.9\}$ were used, superiorization with restarts always reached the same or a better solution than both the basic and the superiorized algorithms without restarts.

	AP vs Sup.		AP vs Sup. Res.		Sup. vs Sup. Res.	
Kernel	AP	Sup.	AP	Sup.Res.	Sup.	Sup.Res.
$\alpha = 0.5$	1.29%	56.17%	0.08%	57.2%	0.01%	16.9%
$\alpha = 0.6$	0.73%	56.63%	0.02%	57.26%	0.001%	10.78%
$\alpha = 0.7$	0.32%	56.96%	0.002%	57.28%	0%	6.1%
$\alpha = 0.8$	0.10%	57.17%	0%	57.29%	0%	2.86%
$\alpha = 0.9$	0.01%	57.27%	0%	57.3%	0%	0.68%

TABLE 9.1: For each pair-wise comparative of methods and kernel choice, the numbers inside the table are the percentage of the 1 million runs in which each method obtains a solution with strictly lower norm than the other one with which it is compared.

9.4.2 Behavior of the superiorized algorithm for the data of the SMP

In this section, we present another illustrative example of the performance of Algorithm 9.13. To be able to display the iterates, we let both the x-space and the y-space in Problem (9.9)-(9.10) be the Euclidean 2-dimensional spaces. We take C as the intersection of three half-spaces. More precisely, we define them as $C_1 := \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 + x_2 \leq 10\}$, $C_2 := \{(x_1, x_2) \in \mathbb{R}^2 \mid -13x_1 + 3x_2 \leq -26\}$ and $C_3 := \{(x_1, x_2) \in \mathbb{R}^2 \mid x_2 \geq 1\}$.

We let A be the rotation matrix by an angle of $\pi/2$ and Q be the image of C under A (i.e., the intersection of the three half-spaces $Q_1 := \{(y_1, y_2) \in \mathbb{R}^2 \mid y_1 - y_2 \leq -10\}$, $Q_2 := \{(y_1, y_2) \in \mathbb{R}^2 \mid -3y_1 - 13y_2 \leq -26\}$ and $Q_3 := \{(y_1, y_2) \in \mathbb{R}^2 \mid y_1 \leq -1\}$). The function to be reduced in the x-space is the value of the second component $f(x_1, x_2) := x_2$, whereas in the y-space, we aim to find a point with increased first and second components (that is, $B := 2$, $\phi_1(y_1) := -y_1$ and $\phi_2(y_2) := -y_2$).

In other words, we want to tackle with the SM the data of the split minimization problem given by

$$\text{find } x^* = \begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix} \in C \text{ such that } x^* \in \operatorname{argmin} \left\{ x_2 \ : \ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in C \right\}, \tag{9.14}$$

$$\text{and such that } \begin{pmatrix} -x_2^* \\ x_1^* \end{pmatrix} \in Q, \quad -x_2^* \geq y_1 \text{ and } x_1^* \geq y_2 \text{ for all } \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \in Q, \tag{9.15}$$

with $C := \bigcap_{i=1}^{3} C_i$ and $Q := \bigcap_{j=1}^{3} Q_j$.

By looking at Figure 9.3, one easily identifies that the point $(9,1)^T$, obtained as the intersection of the lines $x_1 + x_2 = 10$ and $x_2 = 1$, is the unique solution to the SMP with the above described data.

Again, for the SM we choose the method of alternating projections as the basic algorithm. Consequently, the algorithmic operator that we use is

$$\mathbf{T} := P_{\mathbf{V}} \circ (P_{C_3} \times P_{Q_3}) \circ (P_{C_2} \times P_{Q_2}) \circ (P_{C_1} \times P_{Q_1}),$$

where we recall that $\mathbf{V} := \{(x,y) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid Ax = y\}$. In our experiment, we performed 50 iterations of both the basic algorithm and its superiorized version, taking $\alpha = 0.9$ as the kernel for generating the step-sizes of the perturbations and $N = 1$. The nonascending vectors were taken as $v_x^k := (0,1)^T$ for the perturbations in the x-space, and $v_y^{1,k} := 1$ and $v_y^{2,k} := 1$ in the y-space.

Figure 9.3 shows that, while the method of alternating projection converges to the closest point to the starting point in the intersection in each of the spaces, the superiorized algorithm converges to the solution of the SMP.

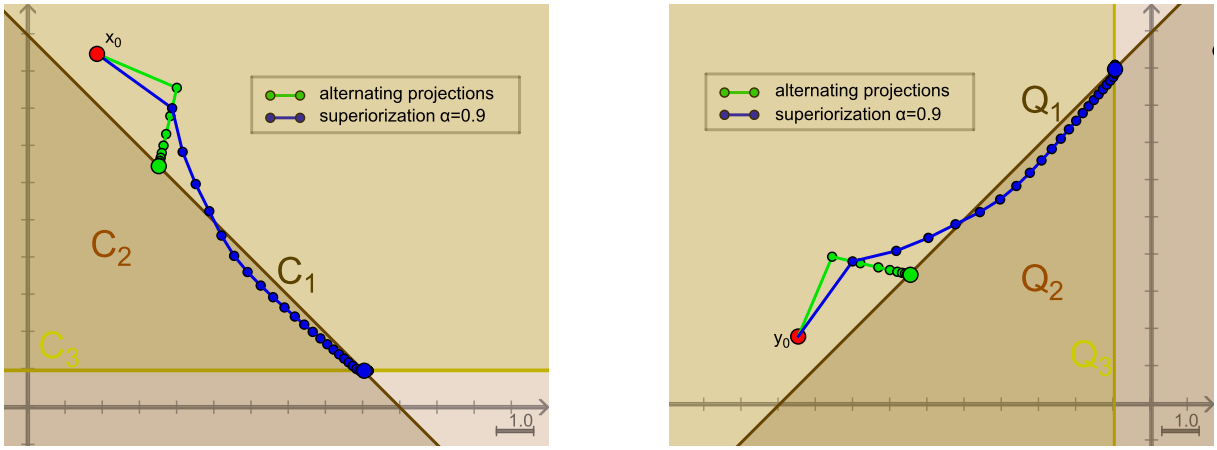


FIGURE 9.3: Performance of the method of alternating projections and its superiorized version algorithm applied in the setting of problem (9.14)-(9.15). The left image displays the iterates in the x-space, while the right one shows the iterates in the y-space.

9.4.3 An illustrative example in IMRT

In this section, we test our SM with restarts algorithm in a sophisticated multiobjective setting motivated from a split minimization problem in the field of Intensity-Modulated Radiation Therapy (IMRT) treatment planning. IMRT is a radiation therapy that manipulates particle beams (protons or photons or others) of varying directions and intensities that are directed toward a human patient to achieve a goal of eradicating tumorous tissues, henceforth called *tumor structures*, while keeping healthy tissues, called *organs-at-risk* below certain thresholds of absorbed dose of radiation. The beams are projected onto the *region of interest* from different angles. Many review papers in this field are available, see, e.g., [94, 95, 103, 170, 238] and references therein.

9.4.3.1 The fully-discretized model of the inverse problem of IMRT

In the fully-discretized model of the inverse problem of IMRT each external radiation beam is discretized into a finite number of *beamlets* (also called *pencil-beams* or *rays*) along which the particles (i.e., their energies) are transmitted. Let all beamlets from all directions be indexed by $i = 1, 2, \ldots, n$, and denote the *intensity* irradiated along the i-th beamlet by x_i. The vector $x = (x_i)_{i=1}^n \in \mathbb{R}^n$ is called the *intensities vector*.

The 2-dimensional (2D) cross-section[1] of the irradiated body is discretized. Assume that the cross-section is covered by a square that is discretized into a finite number of square pixels. This creates an $M \times M$ array of pixels. Let all pixels be indexed by $j = 1, 2, \ldots, m$, with $m = M^2$ and let y_j denote the *dose* of radiation absorbed in the j-th pixel. The vector $y = (y_j)_{j=1}^m \in \mathbb{R}^m$ is called the *dose vector*.

The *intensities space* $\mathbb{R}^n$ and the *dose space* $\mathbb{R}^m$, defined above are the *x-space* and the *y-space*, respectively, mentioned at the beginning of Section 9.3. The physics of the model assumes that there exists an $m \times n$ real matrix $A = (a_{ij})_{i=1,j=1}^{n,m}$(sometimes called the *dose matrix*) through which the intensities of the beamlets and the absorbed doses in pixels are related via the equation

$$Ax = y.$$

Each element a_{ij} in A is the dose absorbed in pixel j due to a unit of intensity along the i-th beamlet. This means that

$$\sum_{i=1}^{n} a_{ij} x_i$$

[1]Everything presented here can easily be extended to 3D wherein the pixels are replaced by voxels. The choice of the 2D case just makes the presentation simpler.

is the total dose absorbed in pixel j due to an intensity vector x. With these notions in mind, we consider the following feasibility-seeking problem of the fully-discretized inverse problem of IMRT.

Problem 9.14 (The feasibility-seeking problem of the fully-discretized inverse problem of IMRT). *Let $\mathbb{R}^n$ and $\mathbb{R}^m$ be the intensities space and the dose space (henceforth called the x-space and the y-space), respectively. Let $A = (a_{ij})_{i=1,j=1}^{n,m}$ be the dose matrix mapping the x-space onto the y-space. For $\ell = 1, 2, \ldots, L$, denote by $T_\ell \subseteq \{1, 2, \ldots, m\}$ the set of pixels corresponding to the ℓ-th tumor structure in the region of interest. For $r = 1, 2, \ldots, R$, denote by $S_r \subseteq \{1, 2, \ldots, m\}$ the set of pixels corresponding to the r-th organ-at-risk. Set $\underline{e}$ and $\overline{e}$ the lower and upper bounds for the available beamlets intensities. Let $\underline{d}_r$ and $\overline{d}_r$, and $\underline{c}_\ell$ and $\overline{c}_\ell$ be the lower and upper bounds for the dose deposited in each pixel of the r-th organ at risk and of the ℓ-th tumor, respectively.*

The task is to find an intensities vector x such that

$$\begin{aligned}
\underline{c}_\ell \leq \sum_{i=1}^{n} a_{ij} x_i \leq \overline{c}_\ell, \quad & \text{for all } j \in T_\ell, \quad \ell \in \{1, 2, \ldots, L\}, \\
\underline{d}_r \leq \sum_{i=1}^{n} a_{ij} x_i \leq \overline{d}_r, \quad & \text{for all } j \in S_r, \quad r \in \{1, 2, \ldots, R\}, \\
\underline{e} \leq x_i \leq \overline{e}, \quad & \text{for all } i \in \{1, 2, \ldots, n\}.
\end{aligned}$$

Problem 9.14 is a linear feasibility problem. Usually, $\underline{e} = \underline{d}_r = 0$ and the significant bounds are $\underline{c}_\ell$ for tumor structures and $\overline{d}_r$ for organs-at-risk. A pair (x^*, y^*) such that x^* is a solution of Problem 9.14 and $y^* = Ax^*$ will be henceforth called "a treatment plan" for the IMRT inverse planning problem.

9.4.3.2 The quest for smoothness and uniformness

In the IMRT inverse planning problem there is an advantage to generating treatment plans with intensity vectors x^* whose subvectors, related to parallel beamlets from the same beam, will be as "smooth" as possible and with dose vectors y^* whose subvectors, related to specific organs (a.k.a. *structures*), that will be as "uniform" as possible.

For the intensity vectors x^*, "smoothness" of subvectors, related to parallel beamlets from the same beam, means that the real numbers that are the individual intensities x_i^*, in each subvector separately, would be as close to each other as possible, subject to the

constraints of Problem 9.14. Such smoothness will allow for less extreme movements of the multileaf collimator[2] that modulates the parallel beamlets from the same beam.

For the dose vectors y^*, "uniformness" of subvectors, related to specific organs means that the real numbers that are the individual doses y_j^*, in each pixel of the subvector would be as close to each other as possible, subject to the constraints of Problem 9.14. Thus, it will guarantee uniformness of the dose deposited within each organ separately and help to avoid the presence of hot- and cold-spots in the dose distribution in each organ, see, e.g., [118].

Each of these aims can be achieved by attempting to minimize or just reduce the total variation of the associated subvectors. Choosing the TV-norm (see (2.9)) as the objective function in the x-space or the y-space, or both, and associating it with the feasibility-seeking problem of the fully-discretized inverse problem of IMRT (Problem 9.14) leads naturally to formulations of the SMP in Problem 9.1.

9.4.3.3 The experimental setup

For the purpose of our numerical experiment, we confine ourselves specifically to a case of the feasibility-seeking problem of the fully-discretized inverse problem of IMRT (Problem 9.14) where there are L tumor structures and the whole rest of the cross-section is considered as one single organ-at-risk, i.e., we let $S_r = S \subseteq \{1, 2, \ldots, m\}$ for all $r = 1, 2, \ldots, R$. This leads to the next split problem of minimizing the TV-norm of the dose subvectors so that uniformity of dose distribution will be achieved for each tumor structure separately.

Problem 9.15. *Let $\mathbb{R}^n$ be the x-space of intensity vectors, let $\mathbb{R}^m$ be the y-space of dose vectors, and A be the dose matrix relating them to each other. For $\ell = 1, 2, \ldots, L$, denote by $T_\ell \subseteq \{1, 2, \ldots, m\}$ the sets of pixels corresponding to the ℓ-th tumor structure and let $S \subseteq \{1, 2, \ldots, m\}$ be the complementary set of pixels that do not belong to any of the target structures and represent all organs at risk. Set $\underline{e}$ and $\overline{e}$ as the lower and upper bounds for the beamlets intensities. Let $\underline{d}$ and $\overline{d}$, and $\underline{c}_\ell$ and $\overline{c}_\ell$ be the dose bounds for pixels in an organ-at-risk and at the tumor structures, respectively. We further assume that the dose vector $y = Ax$ comprises $L + 1$ subvectors $y = (y^\ell)_{\ell=1}^{L+1}$ such that the first L subvectors consist of the doses absorbed in pixels of the L tumor structures and y^{L+1} is the dose absorbed in the complementary tissue S.*

[2] A multileaf collimator is a beam-limiting device that is made of individual "leaves" of a high atomic numbered material, usually tungsten, that can move independently in and out of the path of a radiotherapy beam in order to shape (i.e., modulate) it and vary its intensity. See, e.g., [151].

The task is to find an intensities vector x such that

$$\begin{aligned} \underline{c}_\ell \le y_j^\ell = \sum_{i=1}^{n} a_{ij}x_i \le \overline{c}_\ell, &\quad \text{for all } j \in T_\ell, \quad \ell \in \{1,2,\ldots,L\}, \\ \underline{d} \le y_j^{L+1} = \sum_{i=1}^{n} a_{ij}x_i \le \overline{d}, &\quad \text{for all } j \in S, \\ \underline{e} \le x_i \le \overline{e}, &\quad \text{for all } i \in \{1,2,\ldots,n\}, \end{aligned}$$

$$\text{and } y^\ell \in \operatorname{argmin}\{TV(u) \mid u \in [\underline{c}_\ell, \overline{c}_\ell]^{|T_\ell|}\} \text{ for all } \ell \in \{1,2,\ldots,L\},$$

where, for every $\ell \in \{1,2,\ldots,L\}$, y^ℓ denotes the subvector of the vector y associated with the ℓ-th tumor, and $|T_\ell|$ is the cardinality of the set T_ℓ.

This is the problem we worked on in our experiment. We do not use real data but replicate a realistic situation. In particular, we consider a cross-section of 50×50 square pixels, which translates into the dose vector in the y-space $\mathbb{R}^{2500}$. The number of external radiation beamlets is $n = 2840$, meaning that the x-space is $\mathbb{R}^{2840}$. In the cross-section we have two tumor structures of irregular shapes, whose location appears in Figure 9.5. In order to guarantee the existence of a feasible point for Problem 9.15, we generated the data as follows.

- We generate a vector $\overline{y} \in \mathbb{R}^{2500}$ with components randomly distributed in the interval $[0, 15]$ for the pixels corresponding to organs-at-risk, and in the interval $[10, 40]$ for pixels of tumor structures.
- We randomly generated a matrix $V \in \mathbb{R}^{2840\times 2500}$ with entries in the interval $[0, 1]$ and defined the dose matrix $A \in \mathbb{R}^{2500\times 2840}$, mapping the x-space onto the y-space, as the generalized left inverse of V, i.e., we took $A := (V^TV)^{-1}V^T$.
- We defined $\overline{x} := V\overline{y}$, which implies that $\overline{y} = A\overline{x}$.
- We set the bounds for the constraints of Problem 9.15 as

$$\begin{cases} \underline{d} = 0,\ \overline{d} = \max\{\overline{y}_j \mid j \in S\} + 5\varepsilon_1, \\ \underline{c}_1 = \min\{\overline{y}_j \mid j \in T_1\} - 5\varepsilon_2,\ \overline{c}_1 = \max\{\overline{y}_j \mid j \in T_1\} + 5\varepsilon_3, \\ \underline{c}_2 = \min\{\overline{y}_j \mid j \in T_2\} - 5\varepsilon_4,\ \overline{c}_2 = \max\{\overline{y}_j \mid j \in T_2\} + 5\varepsilon_5, \\ \underline{e} = (\varepsilon_6 + 1)/2 \min\{\overline{x}_i \mid i \in \{1,2,\ldots,n\}\}, \\ \overline{e} = (1 + \varepsilon_7/2) \max\{\overline{x}_i \mid i \in \{1,2,\ldots,n\}\}, \end{cases}$$

where the sub-indices in $\overline{c}$ and $\underline{c}$ refer to the first and second tumor structures and, for $i \in \{1, 2, \ldots, 7\}$, ε_i are randomly picked real numbers in the interval $(0, 1]$.

These choices during the data generation guarantee that there exists a feasible point for Problem 9.15 with these data, namely $\overline{x}$.

In our experimental work, we ran the basic algorithm (Algorithm 9.12) and the superiorized version of the basic algorithm (Algorithm 9.13) with and without restarts. For all of them we took the algorithmic operator as

$$\mathbf{T} := P_{\mathbf{V}} \circ \left(P_{[\underline{c},\overline{c}]} \times P_Q\right),$$

with $\mathbf{V} := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m \mid Ax = y\}$ and $P_Q : \mathbb{R}^m \to \mathbb{R}^m$ defined component-wise as

$$P_Q(y_j) := \begin{cases} P_{[\underline{d},\overline{d}]}(y_j), & \text{if } j \in S, \\ P_{[\underline{c}_1,\overline{c}_1]}(y_j), & \text{if } j \in T_1, \\ P_{[\underline{c}_2,\overline{c}_2]}(y_j), & \text{if } j \in T_2. \end{cases}$$

We tested the three algorithms with different choices of the parameters and present here the most advantageous for each one. Specifically, in Algorithm 9.13 with or without restart the step-sizes were taken in the sequence $\left(c\,\alpha^\ell\right)_{\ell=0}^{\infty}$ with a constant kernel α and a positive number c, and we took $N = 5$. We performed some experiments in order to determine the best choice of α and c for each method. The results are shown in Table 9.2. We chose $\alpha = 0.999$ and $c = 100\,000$ for the superiorized algorithm, since these parameters provide the best reduction in TV-norm values while performing the fastest. For the superiorization with restarts, all of the combinations of parameters, except for the first one, provide a great reduction in the TV-norm values with respect to superiorization with no restarts. Among these combinations, $\alpha = 0.99$ and $c = 100$ was the fastest, so we opted for it.

The target functions ϕ_b were always the appropriate TV-norms. Since no smoothing of the intensities vectors is included in the experiment, we took $v_x^{k,j} = 0$, for all k and j. The final parameters of the two methods are the following:

- **Superiorization:** We took $\alpha = 0.999$, $c = 100\,000$ and $N = 5$, and $v_y^{k,b,j}$ was defined as the nonascending direction given by Theorem 9.6.

- **Superiorization with restarts:** We took $\alpha = 0.99$, $c = 100$, $W_r = 20$ for all r and $N = 5$, and $v_y^{k,b,j}$ was defined as the nonascending direction given by Theorem 9.6.

		$\alpha = 0.99$					$\alpha = 0.999$				
		$c = 10$	$c = 100$	$c = 1000$	$c = 10\,000$	$c = 100\,000$	$c = 10$	$c = 100$	$c = 1000$	$c = 10\,000$	$c = 100\,000$
Sup.	$TV1$	2433.38	2433.36	2432.93	2430.40	2430.40	2432.26	2423.20	2300.88	2167.01	2166.97
	$TV2$	3056.41	3056.34	3055.51	3052.28	3052.28	3054.67	3039.15	2899.47	2714.62	2714.58
	Time	204.34	202.87	203.46	201.12	208.12	202.69	201.84	199.67	199.33	196.57
Sup. Restarts	$TV1$	2078.81	368.81	397.35	381.65	371.03	249.96	361.70	386.13	381.51	382.51
	$TV2$	2689.98	707.2	598.24	579.47	568.02	585.33	583.39	534.21	528.54	528.56
	Time	367.65	453.47	598.55	779.16	950.63	2362.15	3560.19	4426.75	6181.23	7174.09

TABLE 9.2: Average TV-norm values for the first and second subvectors and average time (in seconds) obtained by running the superiorized and superiorized with restarts algorithm with different choices of parameters for 10 random initial points. The algorithms were stopped when a proximity of 0.01 was reached.

We performed multiple runs of the three algorithms. At each run, each of the algorithms was initialized at the same starting point which was randomly generated in the interval $[\underline{e}, \overline{e}]$. We define the proximity of an iterate as the distance to the feasible region, i.e., for an iterate pair (x^k, y^k), we define its proximity as

$$\text{proximity}(x^k, y^k) := \|x^k - P_{[\underline{e},\overline{e}]}(x^k)\| + \|y^k - P_Q(y^k)\|.$$

Note that, due to the definition of the algorithmic operator $\mathbf{T}$, the distance of (x^k, y^k) to $\mathbf{V}$ is 0. All three algorithms were terminated once the proximity became less than 0.01. The obtained results for all different runs are summarized in Table 9.3. Our numerical experiments showed that superiorization with restarts was considerably the best performing algorithm regarding the target function reduction, while superiorization alone, without restarts, did not achieve a significant reduction with respect to the basic algorithm.

This fact can be graphically observed in the heat maps of Figure 9.5, where we represent the dose in the pixels of the cross-section at the last iteration of each algorithm. The uniformity of the heat distribution in a tumor structure represents the dose distribution in that structure. Clearly, superiorization with restarts provided a more homogeneous dose distribution in the tumorous pixels. We observed the increased uniformity of dose distributions in the tumors in all our algorithmic runs of the superiorization with restarts method. However, depending on the datasets and the allowable parameters the level of the uniformity may vary.

The evolution throughout the iterations of the proximity and the total variation of the algorithms is shown in Figure 9.4 with "proximity-target function curves" (which were introduced in [90]), where the iteration indices k increase from right to left in each of the plots. Finally, we note that superiorization with restarts needed more time and a larger number of iterations to reach the desired proximity.

Run		1	2	3	4	5
TV for subvector 1	Basic	2405.3	2498.4	2289.3	2624.74	2474.9
	Superiorized	2072.8	2230.1	2089.6	2362.7	227.2
	Sup. Restarts	421.9	315.6	404.3	349.3	301.1
TV for subvector 2	Basic	3019.1	3252.8	3002.5	3076.6	3096.3
	Superiorized	2703.3	2837.4	2558.9	2744.3	2848.2
	Sup. Restarts	817.7	759.7	688.1	709.2	531.2
Run time (sec.)	Basic	96.1	94.6	91.6	93.6	95.4
	Superiorized	261.6	263.3	253.8	269.4	267.3
	Sup. Restarts	577.5	588.0	630.9	580.3	582.7
No. of iterations	Basic	7352	7265	7117	7400	7505
	Superiorized	7327	7195	7095	7382	7478
	Sup. Restarts	14880	14819	17140	15275	14778

TABLE 9.3: TV-norm values for the first and second subvector, run times and number of iterations resulting from running the Basic, Superiorized and Superiorized with restarts algorithms (for runs with 5 different random initial points) until a proximity of 0.01 was reached.

In our experiments we have observed that other choices of parameters for the superiorization with restarts runs can be employed to reduce its running times and make them comparable to those of the superiorized algorithm without restarts and, at the same time, still achieve a significant reduction of the target function when compared to the other algorithms.

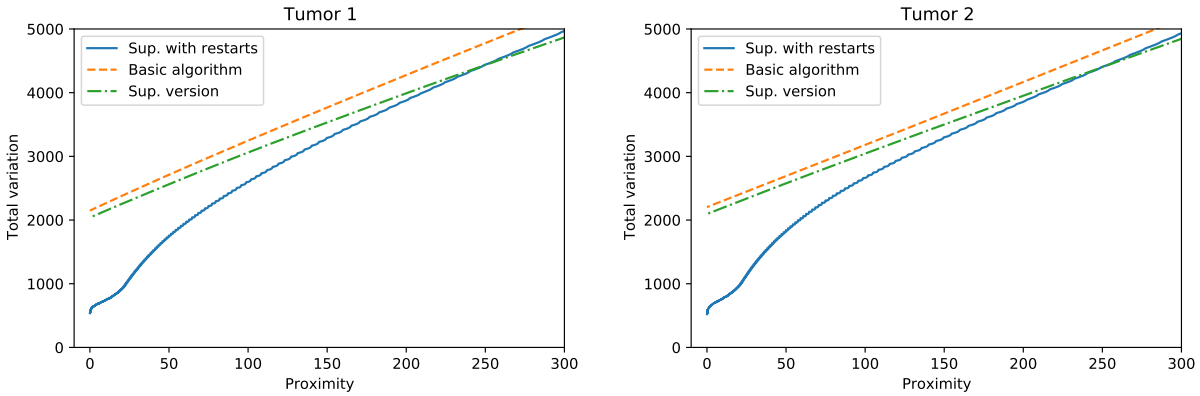


FIGURE 9.4: The evolution of the total variation and the proximity of the iterations of the first run of each of the algorithms for the subvector associated to the first tumor (left) and the second tumor (right). In these "proximity-target function curves" the iteration indices k increase from right to left in each of the plots.

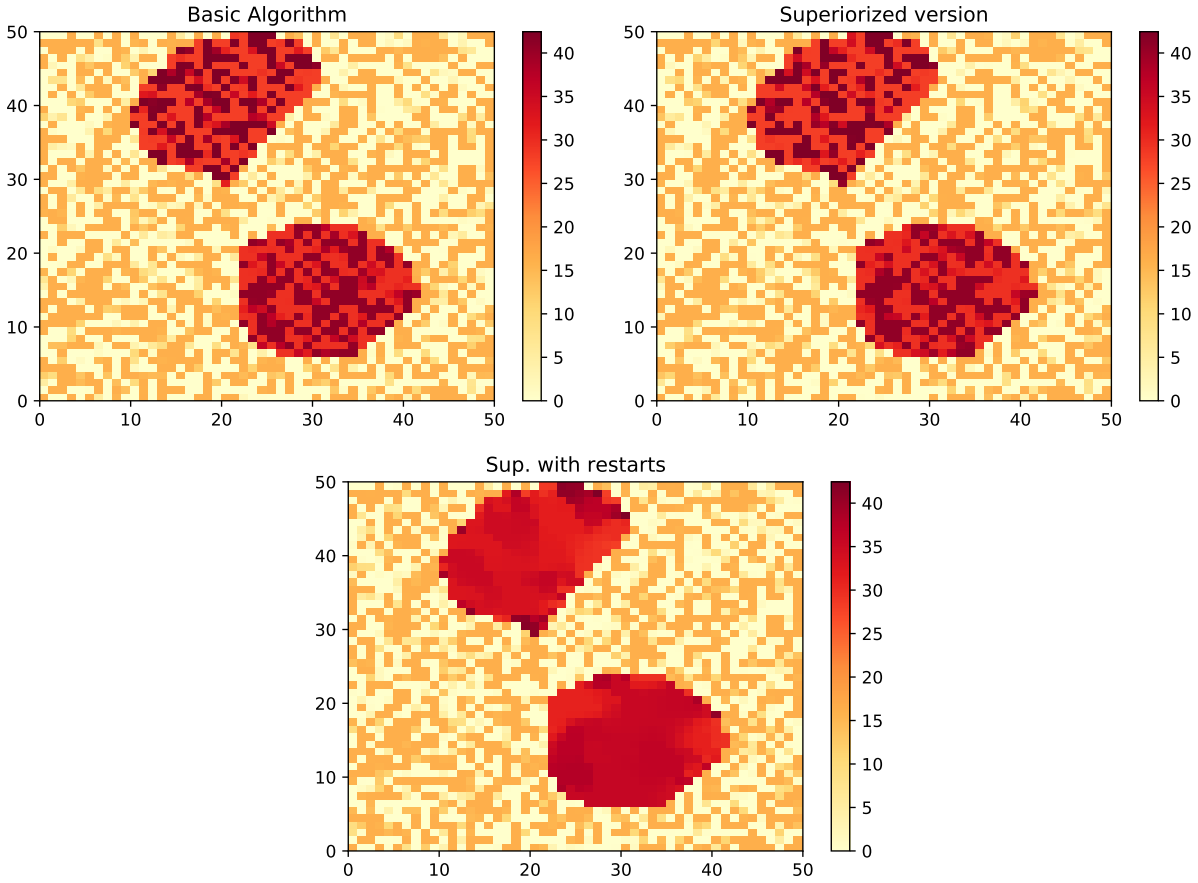


FIGURE 9.5: Heat maps of the solutions in the y-space of pixel doses for the first run of each one of the studied algorithms. Represented is a 50×50 square grid of pixels, where the color indicates the dose absorbed in each pixel.

Conclusions and future research

The material in this thesis is based on the author's joint work that appeared in the publications [10, 23, 25, 26], and the submitted manuscript [29].

Overview of Part I

In the first part of this thesis, we have contributed to the theory and development of monotone operator splitting methods. As an introduction to this topic, we presented a survey of monotone inclusion problems and introduced the foremost algorithms to tackle them. We stressed the applications to numerous situations arising in mathematical optimization.

With the aim of improving the computational efficiency of splitting algorithms, we studied the dimension of the underlying space, or *lifting*, of these methods, a concept that is directly related to the memory requirements of an algorithm. In Chapter 4, we established a unifying framework for presenting the existing results that analyze the minimal lifting that can be achieved by different classes of splitting schemes. The new theoretical contributions are the extension of the minimal lifting theorem of Malitsky and Tam [173] to emcompass the use of resolvent parameters in resolvent splittings, as well as a new characterization of the minimal lifting of primal-dual splitting algorithms for composite monotone inclusions.

The first algorithm with minimal lifting we took into consideration was the Davis–Yin splitting, first introduced in [128]. In Chapter 5, we have presented an alternative proof of convergence without requiring the Davis–Yin operator to be averaged. The proof was solely based on monotone operator theory and has the additional advantage of allowing larger stepsizes, up to four times the cocoercivity constant of the single-valued operator, doubling thus the range of values allowed in [128]. As a consequence, the same conclusion applies to the forward-backward splitting algorithm and the generalized forward-backward.

In Chapters 6 and 7, we analyze the convergence of four novel splitting schemes with reduced lifting. The first method in Chapter 6 is a forward-backward type algorithm with

minimal lifting for addressing monotone inclusions with cocoercive operators. One of the advantages of this algorithm is that it does not rely on a product space reformulation, which makes it suitable for distributed decentralized implementation on a ring network. The second method is obtained as a modification of the previous one. The incorporation of reflected-like terms allows to tackle inclusions involving monotone and Lipschitz continuous operators that are not cocoercive. The latter method cannot be guaranteed to have minimal lifting as it requires more than one forward evaluation of the single-valued operator, which is a necessary assumption for the application of the minimal lifting characterization. Nonetheless, it does reduce the lifting with respect to the existing algorithms in the literature.

Chapter 7 is devoted to composite monotone inclusions, those that include linear compositions of maximally monotone operators. There, we present the first primal-dual resolvent splitting method with minimal lifting for this problem. Further, we combine the three previous algorithms to devise a scheme for addressing mixtures of the monotone inclusions studied in the prequel. Our numerical experiments suggest that the lifting reduction furnishes a decrease in the running time of the algorithms while preserving the quality of the solution.

Overview of Part II

In the second part of the thesis, we provide advances in the theory of splitting methods for nonconvex optimization and in the application of the superiorization methodology.

In Chapter 8, we developed a new splitting algorithm for structured nonconvex optimization problems, which we named as *Boosted Double-proximal Subgradient Algorithm* (BDSA). One of the main features of our method is the inclusion of a linesearch at the end of each iteration. If the stepsizes in every iteration of the linesearch are set to 0, then algorithms such as the *proximal difference of convex functions algorithm* [229], the *generalized proximal point algorithm* [7] and the *double-proximal gradient algorithm* [38] can be recovered as particular cases of BDSA. Nevertheless, BDSA can also be applied to far more general problems and provides a wider range of possibilities to deal with the different elements of the objective function.

The convergence of the sequence generated by BDSA is guaranteed under the usual assumptions required for this class of nonconvex problems. In addition, when the Kurdyka–Łojasiewicz property holds, global convergence and convergence rates can be derived.

We illustrated the advantages of the additional linesearch included in BDSA with multiple experiments. By recurring to two new test functions, we showed that BDSA seems to have bigger chances of avoiding non-optimal critical points and converging to global minima than "non-boosted" and inertial methods. Indeed, in some of our test problems BDSA was the only method to have a 100% rate of success in doing so, for some adequate splittings of the objective function. Further, the boosting step significantly reduced the running time and the number of iterations employed by the algorithm. For example, BDSA was twice faster in reaching the same accuracy than the algorithm with extrapolation recently proposed in [204] for an application of the minimum sum-of-squares clustering problem. For different generalizations of the Heron problem, BDSA also managed to be much faster than its non-accelerated version, both in time and number of iterations.

Chapter 9 introduced a novel algorithm based on the superiorization methodology for tackling the split minimization problem. Superiorization algorithms are a class of methods that interlace feasibility seeking steps with the inclusion of certain perturbations which aim to reduce (not minimize) a target function while ensuring constraint satisfiability. This defines a "semi-heuristic" procedure, in the sense that a superiorized algorithm will always converge to a feasible point, but there is no theoretical guarantee that the perturbations will succeed in obtaining a point with objective function value smaller than the output of the feasibility seeking method without perturbations.

Two novel elements were included in our superiorization based scheme. The first is a permission to restart the perturbations in the superiorized algorithm, which can increase the computational efficiency. The second is the ability to superiorize independently over subvectors. We illustrated our developments in a realistic situation arising in the field of intensity-modulated radiation therapy treatment planning.

Future directions of research

The family of splitting algorithms with minimal or reduced lifting is quite recent. As these methods do not usually rely on product space reformulations, it is not clear if the developments in classical splitting algorithms can be immediately extended to them. The determination of convergence rates, the development of acceleration techniques or the convergence analysis in the presence of pathologies are some of the topics that yet have to be explored. We now enumerate some of the open questions that we consider more relevant for this class of methods.

Obtaining a theoretical certification of the efficiency of reducing the lifting of an algorithm Although there is sufficient numerical experience showing that splitting methods with reduced lifting usually outperform "higher dimensional" product space reformulations (see, e.g., [81, 173, 250] and the numerical experiments presented in this thesis), there exist no theoretical results supporting this phenomenon. One possible approach to this matter would be considering the *Friedrichs angle* of Pierra's product space reformulation. The cosine of the Friedrichs angle has been proven to determine the rate of linear convergence of Douglas–Rachford applied to feasibility problems involving two subspaces [46]. It would be interesting to examine whether the Friedrichs angle of Pierra's reformulation increases in correlation with the the number of subspaces. This would explain why the convergence of this product space technique slows down for large-scale problems.

Does it exist a frugal forward-backward method for monotone and Lipschitz continuous operators? Recall Problem 3.9, where we aim to find a zero in the sum of n set-valued operators and m single-valued operators. Fact 4.13 states that the lifting of frugal forward-backward splittings (see Definition 4.11) must be greater than or equal to $n-1$. When the single-valued operators are cocoercive, the minimal lifting problem is unraveled, in the sense that there exist algorithms with $(n-1)$-fold lifting that solve the monotone inclusion (for instance, Algorithm 6.6). Nonetheless, this cannot be said when the single-valued operators are just monotone and Lipschitz continuous. In this case, either the methods capable of solving Problem 3.9 have n-fold lifting, or they are not frugal, as more than one forward evaluation of the single-valued operators is needed per iteration. Therefore, two natural alternative questions arise: (i) Are the assumptions in Fact 4.13 too restrictive and two evaluations of the single-valued operators should be allowed for studying the lifting of this problem? (ii) Is n the lower bound for the algorithms' lifting in this case?

Achieving lifting reduction in the space of linearly composed operators The minimal lifting characterization presented in Theorem 4.21 assumes that the dimension of the space associated with the dual variables coincides with the number of linearly composed operators. In this context, it states that the lifting of the space of primal variables must be greater than or equal to the number of operators minus one, condition that is satisfied by Algorithm 7.8. However, it remains open the question of whether it is possible to reduce the dimension of the underlying space associated to the linearly

composed operators. More precisely, if we consider the problem given by

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{j=1}^{m} L_j^* B_j(L_j x),$$

is it possible to obtain an algorithm for solving this problem with $(0, m-1)$-fold lifting (according to Definition 4.15)? Or even with $(1, m-1)$ or $(0, m)$-fold lifting? To the best of the author's knowledge, there exists no primal-dual resolvent splitting with these liftings.

Regarding the developments studied in Part II, we propose the following lines of research. The first of them gathers a spectrum of applications where to continue testing the performance of BDSA.

Applications to nonconvex formulations of combinatorial and geometric problems Some NP-hard combinatorial and geometric problems such as the max-cut, the quadratic assignment problem or the Euclidean Distance Matrix Completion (EDMC) problem can be formulated as a semidefinite optimization problem with a nonconvex low-rank constraint (see [244] for more details). Recently, different works have addressed these programs by considering their convex relaxations, obtained by just removing the rank constraint, and applying first order splitting methods to tackle these simpler versions of the problem (see, e.g., [144, 163]). It is important to note that this approach may fail to solve the original problem, as the solution of the convex relaxation might not satisfy the low-rank condition. This is particularly problematic in applications such as protein reconstruction, where neglecting the rank constraint might lead to obtaining a protein structure not embeddable in the Euclidean space of dimension three. Nevertheless, low-rank constraints have been proven to be tractable. Indeed, making use of the projection onto the set $\{A \in \mathbb{R}^{n\times n} : A^T = A, \operatorname{rank}(A) \leq r\}$, the Douglas–Rachford method has been successfully applied to heuristically tackle nonconvex matrix completion problems (see [12]). Studying the behavior of BDSA for EDMC as well as other combinatorial problems seems an intriguing direction for future work.

Incorporating second order information The use of second order derivatives in Newton and Quasi-Newton methods has been well-recognized in the literature for providing faster convergence compared to first order numerical methods (see, e.g., [149, 194]). In the framework of nonsmooth structured optimization problems, the incorporation of second order information has only recently gained attention (see, e.g., [28, 152, 153] and

references therein). For some applications, it can be crucial to extend our results by incorporating Hessian information into the data. This becomes particularly significant when dealing with data that is not twice continuously differentiable. Recent studies propose the integration of generalized Hessians (see, e.g., [28] and the references therein) to enable linesearches in Newton-like methods.

Application of the superiorization methodology to the inverse problem of intensity-modulated radition therapy treatment planning with real data Besides the investigation of the guarantee problem of the SM already detailed in Section 9.1.1, we consider highly interesting to extend the experimental framework in Section 9.4.3.1 to case studies with realistic data. This would be a first step towards achieving the ultimate goal of translating our theoretical developments into real implementation in radiotherapy.

Finally we state what, in the author's opinion, is the most intriguing unanswered question about splitting algorithms for nonconvex problems.

Convergence analysis of the Douglas–Rachford splitting in the nonconvex setting The convergence analysis of the Douglas–Rachford algorithm for the minimization of two nonconvex functions is one of the most challenging open questions in the field of splitting methods. The main difficulty to face in this context is the fact that Douglas–Rachford is not a descent method. The sequences that it generates present an oscillatory behavior, in the sense that they do not consistently decrease the objective function value in every iteration. Some authors have resorted to the use of *envelopes*, auxiliary functions that allow establishing some descent inequality and subsequently applying the usual Kurdyka–Łojasiewicz property based approach, see, e.g., [162, 199]. However, stronger assumptions on the initial functions are usually required, such as weak or strong convexity and differentiability (see, e.g., [57, 104, 162]). As of today there are no theoretical results that explain the good behavior of Douglas–Rachford for combinatorial problems.

Bibliography

[1] Adly, S., Hantoute, A., and Le, B. K. Maximal monotonicity and cyclic monotonicity arising in nonsmooth Lur'e dynamical systems. *J. Math. Anal. Appl. 448*, 1 (2017), 691–706.

[2] Aharoni, R., and Censor, Y. Block-iterative projection methods for parallel computation of solutions to convex feasibility problems. In *Proceedings of the Fourth Haifa Matrix Theory Conference (Haifa, 1988)* (1989), vol. 120, pp. 165–175.

[3] Aleyner, A., and Reich, S. Block-iterative algorithms for solving convex feasibility problems in Hilbert and in Banach spaces. *J. Math. Anal. Appl. 343*, 1 (2008), 427–435.

[4] Alizadeh Tabrizian, A. H., and Bidabadi, N. A new boosted proximal point algorithm for minimizing nonsmooth DC functions. *Numer. Funct. Anal. Optim. 43*, 12 (2022), 1459–1483.

[5] Alwadani, S., Bauschke, H. H., Moursi, W. M., and Wang, X. On the asymptotic behaviour of the Aragón Artacho–Campoy algorithm. *Oper. Res. Lett. 46*, 6 (2018), 585–587.

[6] Amahroq, T., Penot, J.-P., and Syam, A. On the subdifferentiability of the difference of two functions and local minimization. *Set-Valued Anal. 16*, 4 (2008), 413–427.

[7] An, N. T., and Nam, N. M. Convergence analysis of a proximal point algorithm for minimizing differences of functions. *Optimization 66*, 1 (2017), 129–147.

[8] Antipin, A. S. On a method for convex programs using a symmetrical modification of the lagrange function. *Èkonom. i Mat. Metody 12* (1976), 1164–1173.

[9] Aragón, F. J., Goberna, M. A., López, M. A., and Rodríguez, M. M. L. *Nonlinear Optimization*. Springer, Cham, 2019.

[10] Aragón-Artacho, F. J., Boţ, R. I., and Torregrosa-Belén, D. A primal-dual splitting algorithm for composite monotone inclusions with minimal lifting. *Numer. Algorithms 93*, 1 (2023), 103–130.

[11] Aragón Artacho, F. J., and Borwein, J. M. Global convergence of a non-convex Douglas–Rachford iteration. *J. Global Optim. 57*, 3 (2013), 753–769.

[12] Aragón Artacho, F. J., Borwein, J. M., and Tam, M. K. Douglas–Rachford feasibility methods for matrix completion problems. *ANZIAM J. 55*, 4 (2014), 299–326.

[13] Aragón Artacho, F. J., Borwein, J. M., and Tam, M. K. Global behavior of the Douglas–Rachford method for a nonconvex feasibility problem. *J. Global Optim. 65*, 2 (2016), 309–327.

[14] Aragón Artacho, F. J., and Campoy, R. A new projection method for finding the closest point in the intersection of convex sets. *Comput. Optim. Appl. 69*, 1 (2018), 99–132.

[15] Aragón Artacho, F. J., and Campoy, R. Solving graph coloring problems with the Douglas–Rachford algorithm. *Set-Valued Var. Anal. 26*, 2 (2018), 277–304.

[16] Aragón Artacho, F. J., and Campoy, R. Computing the resolvent of the sum of maximally monotone operators with the averaged alternating modified reflections algorithm. *J. Optim. Theory Appl. 181*, 3 (2019), 709–726.

[17] Aragón Artacho, F. J., and Campoy, R. Optimal rates of linear convergence of the averaged alternating modified reflections method for two subspaces. *Numer. Algorithms 82*, 2 (2019), 397–421.

[18] Aragón Artacho, F. J., Campoy, R., and Elser, V. An enhanced formulation for solving graph coloring problems with the Douglas–Rachford algorithm. *J. Global Optim. 77*, 2 (2020), 383–403.

[19] Aragón Artacho, F. J., Campoy, R., and Tam, M. K. The Douglas–Rachford algorithm for convex and nonconvex feasibility problems. *Math. Methods Oper. Res. 91* (2020), 201–240.

[20] Aragón Artacho, F. J., Campoy, R., and Tam, M. K. Strengthened splitting methods for computing resolvents. *Comput. Optim. Appl. 80*, 2 (2021), 549–585.

[21] Aragón-Artacho, F. J., Campoy, R., and Vuong, P. T. The boosted DC algorithm for linearly constrained DC programming. *Set-Valued Var. Anal. 30*, 4 (2022), 1265–1289.

[22] Aragón Artacho, F. J., Censor, Y., and Gibali, A. The cyclic Douglas–Rachford algorithm with r-sets-Douglas–Rachford operators. *Optim. Methods Softw. 34*, 4 (2019), 875–889.

[23] Aragón-Artacho, F. J., Censor, Y., Gibali, A., and Torregrosa-Belén, D. The superiorization method with restarted perturbations for split minimization problems with an application to radiotherapy treatment planning. *Appl. Math. Comput. 440* (2023), Paper No. 127627, 17.

[24] Aragón Artacho, F. J., Fleming, R. M., and Vuong, P. T. Accelerating the DC algorithm for smooth functions. *Math. Program. 169* (2018), 95–118.

[25] Aragón-Artacho, F. J., Malitsky, Y., Tam, M. K., and Torregrosa-Belén, D. Distributed forward-backward methods for ring networks. *Comput. Optim. Appl. 86*, 3 (2023), 845–870.

[26] Aragón-Artacho, F. J., and Torregrosa-Belén, D. A direct proof of convergence of Davis–Yin splitting algorithm allowing larger stepsizes. *Set-Valued Var. Anal. 30*, 3 (2022), 1011–1029.

[27] Aragón Artacho, F. J., and Vuong, P. T. The boosted difference of convex functions algorithm for nonsmooth functions. *SIAM J. Optim. 30*, 1 (2020), 980–1006.

[28] Aragón-Artacho, F. J., Mordukhovich, B. S., and Pérez-Aros, P. Coderivative-based semi-Newton method in nonsmooth difference programming. *Preprint, arXiv:2301.03491 [math.OC], 2023*.

[29] Aragón-Artacho, F. J., Pérez-Aros, P., and Torregrosa-Belén, D. The boosted double-proximal subgradient algorithm for nonconvex optimization. *Preprint, arXiv:2306.17144 [math.OC], 2023*.

[30] Attouch, H., and Bolte, J. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Program. 116*, 1-2 (2009), 5–16.

[31] Attouch, H., Bolte, J., and Svaiter, B. F. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Math. Program. 137*, 1-2 (2013), 91–129.

[32] Attouch, H., and Peypouquet, J. Convergence of inertial dynamics and proximal algorithms governed by maximally monotone operators. *Math. Program. 174* (2019), 391–432.

[33] Attouch, H., Peypouquet, J., and Redont, P. A dynamical approach to an inertial forward-backward algorithm for convex minimization. *SIAM J. Optim. 24*, 1 (2014), 232–256.

[34] Attouch, H., Peypouquet, J., and Redont, P. Backward–forward algorithms for structured monotone inclusions in Hilbert spaces. *J. Math. Anal. Appl. 457*, 2 (2018), 1095–1117.

[35] Attouch, H., and Théra, M. A general duality principle for the sum of two operators. *J. Convex Anal. 3*, 1 (1996), 1–24.

[36] Azé, D., Attouch, H., and Wets, R. J.-B. Convergence of convex-concave saddle functions: applications to convex programming and mechanics. *Ann. Inst. Henri Poincaré, 5*, 6 (1988), 537–572.

[37] Baillon, J. B., and Haddad, G. Quelques propriétés des opérateurs angle-bornés etn-cycliquement monotones. *Israel Journal of Mathematics 26* (1977), 137–150.

[38] Banert, S., and Boţ, R. I. A general double-proximal gradient algorithm for DC programming. *Math. Program. 178*, 1 (2019), 301–326.

[39] Bartz, S., Dao, M. N., and Phan, H. M. Conical averagedness and convergence analysis of fixed point algorithms. *J. Global Optim. 82*, 2 (2022), 351–373.

[40] Bauschke, H. H. The approximation of fixed points of compositions of nonexpansive mappings in Hilbert space. *J. Math. Anal. Appl. 202*, 1 (1996), 150–159.

[41] Bauschke, H. H. A note on the paper by Eckstein and Svaiter on "General projective splitting methods for sums of maximal monotone operators". *SIAM J. Control Optim. 48*, 4 (2009), 2513–2515.

[42] Bauschke, H. H., and Borwein, J. M. On projection algorithms for solving convex feasibility problems. *SIAM Rev. 38*, 3 (1996), 367–426.

[43] Bauschke, H. H., Boţ, R. I., Hare, W. L., and Moursi, W. M. Attouch–Théra duality revisited: paramonotonicity and operator splitting. *J. Approx. Theory 164*, 8 (2012), 1065–1084.

[44] Bauschke, H. H., and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2 ed. Springer, 2017.

[45] Bauschke, H. H., Combettes, P. L., and Luke, D. R. Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization. *J. Opt. Soc. Amer. A 19*, 7 (2002), 1334–45.

[46] Bauschke, H. H., Cruz, Y. B., Nghia, T. T. A., Phan, H. M., and Wang, X. The rate of linear convergence of the Douglas–Rachford algorithm for subspaces is the cosine of the Friedrichs angle. *J. Approx. Theory 185* (2013), 63–79.

[47] Bauschke, H. H., Dao, M. N., and Lindstrom, S. B. The Douglas–Rachford algorithm for a hyperplane and a doubleton. *J. Global Optim. 74*, 1 (2019), 79–93.

[48] Bauschke, H. H., Dao, M. N., and Moursi, W. M. The Douglas–Rachford algorithm in the affine-convex case. *Oper. Res. Lett. 44*, 3 (2016), 379–382.

[49] Bauschke, H. H., Lukens, B., and Moursi, W. M. Affine nonexpansive operators, Attouch–Théra duality and the Douglas–Rachford algorithm. *Set-Valued Var. Anal. 25* (2017), 481–505.

[50] Bauschke, H. H., and Moursi, W. M. On the Douglas–Rachford algorithm. *Math. Program. 164*, 1-2 (2017), 263–284.

[51] Bauschke, H. H., and Moursi, W. M. On the behavior of the Douglas–Rachford algorithm for minimizing a convex function subject to a linear constraint. *SIAM J. Optim. 30*, 3 (2020), 2559–2576.

[52] Bauschke, H. H., and Moursi, W. M. *An Introduction to Convexity, Optimization, and Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2023.

[53] Bauschke, H. H., Schaad, J., and Wang, X. On Douglas–Rachford operators that fail to be proximal mappings. *Math. Program. 168* (2018), 55–61.

[54] Bauschke, H. H., Singh, S., and Wang, X. The splitting algorithms by Ryu, by Malitsky–Tam, and by Campoy applied to normal cones of linear subspaces converge strongly to the projection onto the intersection. *SIAM J. Optim. 33*, 2 (2023), 739–765.

[55] Beck, A., and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci. 2*, 1 (2009), 183–202.

[56] Bertsekas, D. P., and Tsitsiklis, J. N. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, MA, 2014. Originally published by Prentice-Hall, Inc. in 1989. Includes corrections (1997).

[57] Bian, F., and Zhang, X. A three-operator splitting algorithm for nonconvex sparsity regularization. *SIAM J. Sci. Comput. 43*, 4 (2021), A2809–A2839.

[58] Boţ, R. I., Csetnek, E. R., and Heinrich, A. A primal-dual splitting algorithm for finding zeros of sums of maximal monotone operators. *SIAM J. Optim. 23*, 4 (2013), 2011–2036.

[59] Boţ, R. I., Csetnek, E. R., Heinrich, A., and Hendrich, C. On the convergence rate improvement of a primal-dual splitting algorithm for solving monotone inclusion problems. *Math. Program. 150*, 2 (2015), 251–279.

[60] Boţ, R. I., Csetnek, E. R., and Hendrich, C. Recent developments on primal-dual splitting methods with applications to convex minimization. In *Mathematics without boundaries*. Springer, New York, 2014, pp. 57–99.

[61] Boţ, R. I., Csetnek, E. R., and Hendrich, C. Inertial Douglas–Rachford splitting for monotone inclusion problems. *Appl. Math. Comput. 256* (2015), 472–487.

[62] Boţ, R. I., Csetnek, E. R., and László, S. C. An inertial forward-backward algorithm for the minimization of the sum of two nonconvex functions. *EURO J. Comput. Optim. 4*, 1 (2016), 3–25.

[63] Boţ, R. I., Grad, S.-M., Meier, D., and Staudigl, M. Inducing strong convergence of trajectories in dynamical systems associated to monotone inclusions with composite structure. *Adv. Nonlinear Anal. 10*, 1 (2021), 450–476.

[64] Boţ, R. I., and Hendrich, C. Convergence analysis for a primal-dual monotone + skew splitting algorithm with applications to total variation minimization. *J. Math. Imaging Vision 49*, 3 (2014), 551–568.

[65] Boţ, R. I., and Hendrich, C. Solving monotone inclusions involving parallel sums of linearly composed maximally monotone operators. *Inverse Probl. Imaging 10*, 3 (2016), 617–640.

[66] Bolte, J., Sabach, S., and Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program. 146*, 1-2 (2014), 459–494.

[67] Borwein, J. M., and Tam, M. K. A cyclic Douglas–Rachford iteration scheme. *J. Optim. Theory Appl. 160*, 1 (2014), 1–29.

[68] Boţ, R. I., and Csetnek, E. R. A Tseng's type penalty scheme for solving inclusion problems involving linearly composed and parallel-sum type monotone operators. *Vietnam J. Math. 42*, 4 (2014), 451–465.

[69] Bot, R. I., and Hendrich, C. A Douglas–Rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators. *SIAM J. Optim. 23*, 4 (2013), 2541–2565.

[70] Bot, R. I., Nguyen, D.-K., and Zong, C. Fast forward-backward splitting for monotone inclusions with a convergence rate of the tangent residual of $o(1/k)$. *Preprint, arXiv:2312.12175 [math.OC], 2023*.

[71] Boyle, J. P., and Dykstra, R. L. A method for finding projections onto the intersection of convex sets in Hilbert spaces. In *Advances in order restricted statistical inference (Iowa City, Iowa, 1985)*, vol. 37 of *Lect. Notes Stat.* Springer, Berlin, 1986, pp. 28–47.

[72] Bredies, K., Chenchene, E., and Naldi, E. Graph and distributed extensions of the Douglas–Rachford method. *Preprint, arXiv:2312.12175 [math.OC], 2022*.

[73] Briceño Arias, L. M. Forward-Douglas–Rachford splitting and forward-partial inverse method for solving monotone inclusions. *Optimization 64*, 5 (2015), 1239–1261.

[74] Briceño Arias, L. M. Forward-partial inverse-forward splitting for solving monotone inclusions. *J. Optim. Theory Appl. 166*, 2 (2015), 391–413.

[75] Briceño Arias, L. M., and Combettes, P. L. A monotone + skew splitting model for composite monotone inclusions in duality. *SIAM J. Optim. 21*, 4 (2011), 1230–1250.

[76] Brooke, M., Censor, Y., and Gibali, A. Dynamic string-averaging CQ-methods for the split feasibility problem with percentage violation constraints arising in radiation therapy treatment planning. *Int. Trans. Oper. Res. 30*, 1 (2023), 181–205.

[77] Burachik, R. S., Caldwell, B. I., and Kaya, C. Y. Douglas–Rachford algorithm for control-constrained minimum-energy control problems. To be published in ESAIM Control Optim. Calc. Var (2024).

[78] Burachik, R. S., Caldwell, B. I., Kaya, C. Y., and Moursi, W. M. Optimal control duality and the Douglas–Rachford algorithm. *Preprint, arXiv:2303.06527 [math.OC], 2023*.

[79] Butnariu, D., Davidi, R., Herman, G. T., and Kazantsev, I. G. Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems. *IEEE J. Sel. Top. Signal Process. 1*, 4 (2007), 540–547.

[80] Campoy, R. *Contributions to the Theory and Applications of Projection Algorithms*. PhD thesis, Universidad de Murcia, 2018.

[81] Campoy, R. A product space reformulation with reduced dimension for splitting algorithms. *Comput. Optim. Appl. 83*, 1 (2022), 319–348.

[82] Campoy, R. Regularity of sets under a reformulation in a product space with reduced dimension. *Set-Valued Var. Anal. 31*, 4 (2023), Paper No. 40, 14.

[83] Cegielski, A. *Iterative methods for fixed point problems in Hilbert spaces*, vol. 2057 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 2012.

[84] Censor, Y. Superiorization and perturbation resilience of algorithms: A bibliography compiled and continuously updated. http://math.haifa.ac.il/yair/bib-superiorization-censor.htm. Accessed: 18 March 2024.

[85] Censor, Y. Can linear superiorization be useful for linear optimization problems? *Inverse Problems 33*, 4 (2017), 044006, 22.

[86] Censor, Y., Davidi, R., and Herman, G. T. Perturbation resilience and superiorization of iterative algorithms. *Inverse Problems 26*, 6 (2010), 065008, 12.

[87] Censor, Y., Davidi, R., Herman, G. T., Schulte, R. W., and Tetruashvili, L. Projected subgradient minimization versus superiorization. *J. Optim. Theory Appl. 160*, 3 (2014), 730–747.

[88] Censor, Y., and Elfving, T. A multiprojection algorithm using Bregman projections in a product space. *Numer. Algorithms 8*, 2-4 (1994), 221–239.

[89] Censor, Y., Elfving, T., and Herman, G. T. Averaging strings of sequential iterations for convex feasibility problems. In *Inherently parallel algorithms in feasibility and optimization and their applications (Haifa, 2000)*, vol. 8 of *Stud. Comput. Math.* North-Holland, Amsterdam, 2001, pp. 101–113.

[90] Censor, Y., Garduño, E., Helou, E. S., and Herman, G. T. Derivative-free superiorization: principle and algorithm. *Numer. Algorithms 88*, 1 (2021), 227–248.

[91] Censor, Y., Gibali, A., and Reich, S. Algorithms for the split variational inequality problem. *Numer. Algorithms 59*, 2 (2012), 301–323.

[92] Censor, Y., and Levy, E. An analysis of the superiorization method via the principle of concentration of measure. *Appl. Math. Optim. 83*, 3 (2021), 2273–2301.

[93] Censor, Y., and Nisenbaum, A. String-averaging methods for best approximation to common fixed point sets of operators: the finite and infinite cases. *Fixed Point Theory Algorithms Sci. Eng.* (2021), Paper No. 9, 21.

[94] Censor, Y., Schubert, K. E., and Schulte, R. W. Developments in mathematical algorithms and computational tools for proton CT and particle therapy treatment planning. *IEEE Trans. Radiat. Plasma Med. Sci. 6*, 3 (2021), 313–324.

[95] Censor, Y., and Unkelbach, J. From analytic inversion to contemporary IMRT optimization: Radiation therapy planning revisited from a mathematical perspective. *Phys. Med. 28*, 2 (2012), 109–118.

[96] Censor, Y., and Zaslavski, A. J. Strict Fejér monotonicity by superiorization of feasibility-seeking projection methods. *J. Optim. Theory Appl. 165*, 1 (2015), 172–187.

[97] Chambolle, A. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision 20*, 1-2 (2004), 89–97.

[98] Chambolle, A., Caselles, V., Cremers, D., Novaga, M., and Pock, T. An introduction to total variation for image analysis. In *Theoretical foundations and numerical methods for sparse recovery*, vol. 9 of *Radon Ser. Comput. Appl. Math.* Walter de Gruyter, Berlin, 2010, pp. 263–340.

[99] Chambolle, A., and Lions, P.-L. Image recovery via total variation minimization and related problems. *Numer. Math. 76*, 2 (1997), 167–188.

[100] Chambolle, A., and Pock, T. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision 40*, 1 (2011), 120–145.

[101] Chambolle, A., and Pock, T. An introduction to continuous optimization for imaging. *Acta Numer. 25* (2016), 161–319.

[102] Chierchia, G., Chouzenoux, E., Combettes, P. L., and Pesquet, J.-C. The Proximity Operator Repository. https://proximity-operator.net/. Accessed: 09 January 2024.

[103] Cho, P. S., and Marks II, R. J. Hardware-sensitive optimization for intensity modulated radiotherapy. *Phys. Med. Biol. 45*, 2 (2000), 429–440.

[104] Chuang, C.-S., He, H., and Zhang, Z. A unified Douglas–Rachford algorithm for generalized DC programming. *J. Global Optim. 82*, 2 (2022), 331–349.

[105] Clarke, F. H., Stern, R. J., and Wolenski, P. R. Proximal smoothness and the lower-$\mathcal{C}^2$ property. *J. Convex Anal. 2*, 1-2 (1995), 117–144.

[106] Combettes, P. L. The geometry of monotone operator splitting methods. *Preprint, arXiv:2310.08443 [math.OC], 2023*.

[107] Combettes, P. L. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization 53*, 5-6 (2004), 475–504.

[108] Combettes, P. L. Iterative construction of the resolvent of a sum of maximal monotone operators. *J. Convex Anal 16*, 4 (2009), 727–748.

[109] Combettes, P. L., and Pesquet, J.-C. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators. *Set-Valued Var. Anal. 20* (2011), 307 – 330.

[110] Combettes, P. L., and Pesquet, J.-C. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds., vol. 49 of *Springer Optim. Appl.* Springer, New York, 2011, pp. 185–212.

[111] Combettes, P. L., Salzo, S., and Villa, S. Consistent learning by composite proximal thresholding. *Math. Program. 167*, 1 (2018), 99–127.

[112] Combettes, P. L., and Vu, B. C. Variable metric forward–backward splitting with applications to monotone inclusions in duality. *Optimization 63* (2012), 1289–1318.

[113] Combettes, P. L., and Wajs, V. R. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul. 4*, 4 (2005), 1168–1200.

[114] Combettes, P. L., and Yamada, I. Compositions and convex combinations of averaged nonexpansive operators. *J. Math. Anal. Appl. 425* (2014), 55–70.

[115] Condat, L. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl. 158*, 2 (2013), 460–479.

[116] Condat, L. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl. 158*, 2 (2013), 460–479.

[117] Condat, L., Kitahara, D., Contreras, A., and Hirabayashi, A. Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists. *SIAM Review 65*, 2 (2023), 375–435.

[118] Cook, J. T., Tobler, M., Leavitt, D. D., and Watson, G. IMRT fluence map editing to control hot and cold spots. *Med. Dosim. 30*, 4 (2005), 201–204.

[119] Correa, R., López, M. A., and Pérez-Aros, P. Necessary and sufficient optimality conditions in DC semi-infinite programming. *SIAM J. Optim. 31*, 1 (2021), 837–865.

[120] Correa, R., López, M. A., and Pérez-Aros, P. Optimality conditions in DC-constrained mathematical programming problems. *J. Optim. Theory Appl. 198*, 3 (2023), 1191–1225.

[121] Csetnek, E. R., Malitsky, Y., and Tam, M. K. Shadow Douglas–Rachford splitting for monotone inclusions. *Appl. Math. Optim. 80*, 3 (2019), 665–678.

[122] Cui, F., Tang, Y., and Yang, Y. An inertial three-operator splitting algorithm with applications to image inpainting. *Appl. Set-Valued Anal. Optim. 1*, 2 (2019), 113–134.

[123] Dao, M. N., Dizon, N. D., Hogan, J. A., and Tam, M. K. Constraint reduction reformulations for projection algorithms with applications to wavelet construction. *J. Optim. Theory Appl. 190*, 1 (2021), 201–233.

[124] Dao, M. N., and Phan, H. M. Adaptive Douglas–Rachford splitting algorithm for the sum of two operators. *SIAM J. Optim. 29*, 4 (2019), 2697–2724.

[125] Dao, M. N., and Phan, H. M. Computing the resolvent of the sum of operators with application to best approximation problems. *Optim. Lett. 14*, 5 (2020), 1193–1205.

[126] Dao, M. N., and Phan, H. M. An adaptive splitting algorithm for the sum of two generalized operators and one cocoercive operator. *Fixed Point Theory and Algorithms Sci. Eng. 2021* (2021), 16.

[127] Davidi, R., Herman, G. T., and Censor, Y. Perturbation-resilient block-iterative projection methods with application to image reconstruction from projections. *Int. Trans. Oper. Res. 16*, 4 (2009), 505–524.

[128] Davis, D., and Yin, W. A three-operator splitting scheme and its optimization applications. *Set-Valued Var. Anal. 25*, 4 (2017), 829–858.

[129] de Oliveira, W. The ABC of DC programming. *Set-Valued Var. Anal. 28*, 4 (2020), 679–706.

[130] de Oliveira, W., and Tcheou, M. P. An inertial algorithm for DC programming. *Set-Valued Var. Anal. 27*, 4 (2019), 895–919.

[131] Douglas, J., and Rachford, H. H. J. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Am. Math. Soc. 82* (1956), 421–439.

[132] Eckstein, J. *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, Massachusetts Institute of Technology, 1989.

[133] Eckstein, J., and Bertsekas, D. P. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program. 55* (1992), 293–318.

[134] Eckstein, J., and Svaiter, B. F. A family of projective splitting methods for the sum of two maximal monotone operators. *Math. Program. 111* (2008), 173–199.

[135] Escalante, R., and Raydan, M. *Alternating projection methods*, vol. 8 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2011.

[136] Esser, E., Zhang, X., and Chan, T. F. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sci. 3*, 4 (2010), 1015–1046.

[137] Facchinei, F., and Pang, J.-S. *Finite-Dimensional Variational Inequalities and Complementarity Problems, Volume II*. Springer-Verlag, New York, 2003.

[138] Ferreira, O. P., Santos, E. M., and Souza, J. C. O. A boosted DC algorithm for non-differentiable DC components with non-monotone line search. *Preprint, arXiv:2111.01290v2 [math.OC], 2022*.

[139] Frankel, P., Garrigos, G., and Peypouquet, J. Splitting methods with variable metric for Kurdyka–Łojasiewicz functions and general convergence rates. *J. Optim. Theory Appl. 165*, 3 (2015), 874–900.

[140] Giselsson, P. Nonlinear forward-backward splitting with projection correction. *SIAM J. Optim. 31*, 3 (2021), 2199–2226.

[141] Giselsson, P., and Moursi, W. M. On compositions of special cases of Lipschitz continuous operators. *Fixed Point Theory Algorithms Sci. Eng.* (2021), Paper No. 25, 38.

[142] Glowinski, R., Osher, S. J., and Yin, W., Eds. *Splitting methods in communication, imaging, science, and engineering.* Scientific Computation. Springer, Cham, 2016.

[143] Goldstein, A. A. Convex programming in Hilbert space. *Bull. Amer. Math. Soc. 70* (1964), 709–710.

[144] Graham, N., Hu, H., Im, J., Li, X., and Wolkowicz, H. A restricted dual Peaceman–Rachford splitting method for a strengthened DNN relaxation for QAP. *INFORMS J. Comput. 34*, 4 (2022), 2125–2143.

[145] Guenter, M., Collins, S., Ogilvy, A., and Hare, W. Superiorization versus regularization: a comparison of algorithms for solving image reconstruction problems with applications in computed tomography. *Med. Phys. 49* (2022), 1065–1082.

[146] Haar, A. Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann. 69*, 3 (1910), 331–371.

[147] He, B., and Yuan, X. Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective. *SIAM J. Imaging Sci. 5*, 1 (2012), 119–149.

[148] Herman, G., Garduño, E., Davidi, R., and Censor, Y. Superiorization: an optimization heuristic for a medical physics. *Med. Phys. 39* (2012), 5532–5546.

[149] Izmailov, A. F., and Solodov, M. V. *Newton-Type Methods for Optimization and Variational Problems.* Springer, Cham, 2014.

[150] Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. Proximal methods for hierarchical sparse coding. *J. Mach. Learn. Res. 12* (2011), 2297–2334.

[151] Jeraj, M., and Robar, V. Multileaf collimator in radiotherapy. *Radiol. Oncol. 38*, 3 (2004), 235–240.

[152] Khanh, P. D., Mordukhovich, B., and Phat, V. T. A generalized Newton method for subgradient systems. *Math. Oper. Res. 48*, 4 (2023), 1811–1845.

[153] Khanh, P. D., Mordukhovich, B. S., Phat, V. T., and Tran, D. B. Generalized damped Newton algorithms in nonsmooth optimization via second-order subdifferentials. *J. Global Optim. 86*, 1 (2023), 93–122.

[154] Kim, D. Accelerated proximal point method for maximally monotone operators. *Math. Program. 190*, 1-2 (2021), 57–87.

[155] Klukowska, J., Davidi, R., and Herman, G. SNARK09 a software package for reconstruction of 2D images from 1D projections. *Comput. Methods Programs Biomed. 110* (2013), 424–440.

[156] Kruger, A. Y. Generalized differentials of nonsmooth functions and necessary conditions for an extremum. *Sibirsk. Mat. Zh. 26*, 3 (1985), 78–90, 224.

[157] Kurdyka, K. On gradients of functions definable in o-minimal structures. *Ann. Inst. Fourier (Grenoble) 48*, 3 (1998), 769–783.

[158] Langthaler, O. Incorporation of the superiorization methodology into biomedical imaging software. *Marshall Plan Scholarship Report, Salzburg University of Applied Sciences, Salzburg, Austria, and The Graduate Center of the City University of New York, NY, USA 76* (2014). https://www.marshallplan.at/images/All-Papers/MP-2014/Langthaler.pdf.

[159] Latafat, P., and Patrinos, P. Asymmetric forward-backward-adjoint splitting for solving monotone inclusions involving three operators. *Comput. Optim. Appl. 68*, 1 (2017), 57–93.

[160] Lee, J., Yi, S., and Ryu, E. K. Convergence analyses of Davis–Yin splitting via scaled relative graphs. *Preprint, arXiv:2207.04015 [math.OC], 2022*.

[161] Levitin, E., and Polyak, B. Constrained minimization methods. *U.S.S.R. Comput. Math. Math. Phys. 6*, 5 (1966), 1–50.

[162] Li, G., and Pong, T. K. Douglas–Rachford splitting for nonconvex optimization with application to nonconvex feasibility problems. *Math. Program. 159*, 1-2 (2016), 371–401.

[163] Li, X., Pong, T. K., Sun, H., and Wolkowicz, H. A strictly contractive Peaceman–Rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem. *Comput. Optim. Appl. 78*, 3 (2021), 853–891.

[164] Lindstrom, S. B., and Sims, B. Survey: Sixty years of Douglas–Rachford. *J. Aust. Math. Soc. 110*, 3 (2021), 333–370.

[165] Lions, P. L., and Mercier, B. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal. 16*, 6 (1979), 964–979.

[166] Łojasiewicz, S. *Ensembles semi-analytiques*. Institut des Hautes Etudes Scientifiques, Bures-sur-Yvette (Seine-et-Oise), France, 1965.

[167] Loris, I., and Verhoeven, C. On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. *Inverse Problems 27*, 12 (2011), 125007, 15.

[168] Loshchilov, I., and Hutter, F. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017), OpenReview.net.

[169] Lou, Y., Zeng, T., Osher, S., and Xin, J. A weighted difference of anisotropic and isotropic total variation model for image processing. *SIAM J. Imaging Sci. 8*, 3 (2015), 1798–1823.

[170] Maass, K., Kim, M., and Aravkin, A. A nonconvex optimization approach to IMRT planning with dose-volume constraints. *INFORMS J. Comput. 34*, 3 (2022), 1366–1386.

[171] Maingé, P.-E., and Moudafi, A. Convergence of new inertial proximal methods for DC programming. *SIAM J. Optim. 19*, 1 (2008), 397–413.

[172] Malitsky, Y., and Tam, M. K. A forward-backward splitting method for monotone inclusions without cocoercivity. *SIAM J. Optim. 30*, 2 (2020), 1451–1472.

[173] Malitsky, Y., and Tam, M. K. Resolvent splitting for sums of monotone operators with minimal lifting. *Math. Program. 201* (2023), 231–262.

[174] Martinet, B. Régularisation d'inéquations variationnelles par approximations successives. *Rev. Francaise Inf. Rech. Oper. 4*, R3 (1970), 154–158.

[175] Masad, E., and Reich, S. A note on the multiple-set split convex feasibility problem in Hilbert space. *J. Nonlinear Convex Anal. 8*, 3 (2007), 367–371.

[176] Maulén, J. J., Fierro, I., and Peypouquet, J. Inertial Krasnoselskiĭ–Mann iterations. *Preprint, arXiv:2210.03791 [math.OC], 2023*.

[177] Mercier, B. *Topics in Finite Element Solution of Elliptic Problems*. Lectures on Mathematics. Tata Institute of Fundamental Research, 1979.

[178] Mercier, B. *Inéquations Variationelles de la Mécanique*. Publications mathématiques d'Orsay. Université de Paris-Sud, Département de Mathématique, 1980.

[179] Minty, G. J. Monotone (nonlinear) operators in Hilbert space. *Duke Math. J. 29*, 3 (1962), 341–346.

[180] Minty, G. J. On the monotonicity of the gradient of a convex function. *Pac. J. Math. 14* (1964), 243–247.

[181] Molinari, C., Liang, J., and Fadili, J. Convergence rates of forward-Douglas–Rachford splitting method. *J. Optim. Theory Appl. 182*, 2 (2019), 606–639.

[182] Mordukhovich, B. S. *Variational Analysis and Generalized Differentiation I*. Springer-Verlag, Berlin, 2006.

[183] Mordukhovich, B. S. *Variational Analysis and Applications*. Springer, Cham, 2018.

[184] Mordukhovich, B. S., Nam, N. M., and Salinas, J. Applications of variational analysis to a generalized Heron problem. *Appl. Anal. 91*, 10 (2012), 1915–1942.

[185] Mordukhovich, B. S., Nam, N. M., and Salinas, J. Solving a generalized Heron problem by means of convex analysis. *Amer. Math. Monthly 119*, 2 (2012), 87–99.

[186] Moreau, J. J. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France 93* (1965), 273–299.

[187] Morin, M., Banert, S., and Giselsson, P. Frugal splitting operators: Representation, minimal lifting, and convergence. *Preprint, arXiv:2206.11177, [math.OC], 2023*.

[188] Morin, M., Banert, S., and Giselsson, P. Nonlinear forward-backward splitting with momentum correction. *Set-Valued Var. Anal. 31*, 4 (2023), 37.

[189] Moursi, W. M. *The Douglas–Rachford Operator in the Possibly Inconsistent Case: Static Properties and Dynamic behaviour*. PhD thesis, University of British Columbia, 2016.

[190] Moursi, W. M. The forward-backward algorithm and the normal problem. *J. Optim. Theory Appl. 176*, 3 (2018), 605–624.

[191] Moursi, W. M., and Zinchenko, Y. A note on the equivalence of operator splitting methods. In *Splitting algorithms, modern operator theory, and applications*. Springer, Cham, 2019, pp. 331–349.

[192] Nikazad, T., Abbasi, M., and Mirzapour, M. Convergence of string-averaging method for a class of operators. *Optim. Methods Softw. 31*, 6 (2016), 1189–1208.

[193] no, E. G., Ye, Z., and Herman, G. T. SNARK14. A programming system for the reconstruction of 2D images from 1D projections. Available at: https://turing.iimas.unam.mx/SNARK14M/index.php.

[194] Nocedal, J., and Wright, S. J. *Numerical optimization*, second ed. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006.

[195] Noll, D. Convergence of non-smooth descent methods using the Kurdyka–Łojasiewicz inequality. *J. Optim. Theory Appl. 160*, 2 (2014), 553–572.

[196] O'Connor, D., and Vandenberghe, L. On the equivalence of the primal-dual hybrid gradient method and Douglas–Rachford splitting. *Math. Program. 179*, 1-2 (2020), 85–108.

[197] O'Donogue, B., and Candés, E. Adaptative restart for accelerated gradient schemes. *Found. Comput. Math. 15* (2015), 715–732.

[198] Passty, G. B. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *J. Math. Anal. Appl. 72*, 2 (1979), 383–390.

[199] Patrinos, P., Stella, L., and Bemporad, A. Douglas–Rachford splitting: Complexity estimates and accelerated variants. In *53rd IEEE Conference on Decision and Control* (2014), pp. 4234–4239.

[200] Peaceman, D. W., and Rachford, Jr., H. H. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math. 3* (1955), 28–41.

[201] Pedregosa, F., Fatras, K., and Casotto, M. Proximal splitting meets variance reduction. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (16–18 Apr 2019), K. Chaudhuri and M. Sugiyama, Eds., vol. 89 of *Proceedings of Machine Learning Research*, PMLR, pp. 1–10.

[202] Pedregosa, F., and Gidel, G. Adaptive three operator splitting. In *Proceedings of the 35th International Conference on Machine Learning* (2018), J. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, pp. 4085–4094.

[203] Pennanen, T. Dualization of generalized equations of maximal monotone type. *SIAM J. Optim. 10*, 3 (2000), 809–835.

[204] Pham, T. N., Dao, M. N., Shah, R., Sultanova, N., Li, G., and Islam, S. A proximal subgradient algorithm with extrapolation for structured nonconvex nonsmooth problems. *Numer. Algorithms* (2023), 1–33.

[205] Pham Dinh, T., and Le Thi, H. A. Recent advances in DC programming and DCA. In *Transactions on Computational Intelligence XIII*, N.-T. Nguyen and H. A. Le-Thi, Eds. Springer, Berlin, Heidelberg, 2014, pp. 1–37.

[206] Pierra, G. Decomposition through formalization in a product space. *Math. Program. 28* (1984), 96–115.

[207] Pock, T., Cremers, D., Bischof, H., and Chambolle, A. An algorithm for minimizing the Mumford-Shah functional. In *2009 IEEE 12th International Conference on Computer Vision* (2009), pp. 1133–1140.

[208] Prommegger, B. Verification and evaluation of superiorized algorithms used in biomedical imaging: Comparison of iterative algorithms with and without superiorization for image reconstruction from projections. *Marshall Plan Scholarship Report, Salzburg University of Applied Sciences, Salzburg, Austria, and The Graduate Center of the City University of New York, NY, USA 84* (2014). https://www.marshallplan.at/images/All-Papers/MP-2014/Prommegger.pdf.

[209] Raguet, H. A note on the forward-Douglas–Rachford splitting for monotone inclusion and convex optimization. *Optim. Lett. 13*, 4 (2019), 717–740.

[210] Raguet, H., Fadili, J., and Peyré, G. A generalized forward-backward splitting. *SIAM J. Imaging Sci. 6*, 3 (2013), 1199–1226.

[211] Reich, S., and Tuyen, T. M. Projection algorithms for solving the split feasibility problem with multiple output sets. *J. Optim. Theory Appl. 190*, 3 (2021), 861–878.

[212] Rieger, J., and Tam, M. K. Backward-forward-reflected-backward splitting for three operator monotone inclusions. *Appl. Math. Comput. 381* (2020), 125248, 10.

[213] Rockafellar, R. Minimax theorems and conjugate saddle-functions. *Math. Scan. 14*, 2 (1964), 151–173.

[214] Rockafellar, R. T. Characterization of the subdifferentials of convex functions. *Pac. J. Math. 17* (1966), 497–510.

[215] Rockafellar, R. T. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[216] Rockafellar, R. T. Monotone operators associated with saddle-functions and minimax problems. In *Nonlinear Functional Analysis Part Proceedings of Symposia in Pure Mathematics* (1970), F. E. Browder, Ed., vol. 18, American Mathematical Soc., pp. 241–250.

[217] Rockafellar, R. T. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim. 14*, 5 (1976), 877–898.

[218] Rockafellar, R. T., and Wets, R. J.-B. *Variational Analysis*. Springer-Verlag, Berlin, 1998.

[219] Rudin, L. I., Osher, S., and Fatemi, E. Nonlinear total variation based noise removal algorithms. vol. 60. 1992, pp. 259–268. Experimental mathematics: computational issues in nonlinear science (Los Alamos, NM, 1991).

[220] Ryu, E. K. Uniqueness of DRS as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting. *Math. Program. 182*, 1-2 (2020), 233–273.

[221] Ryu, E. K., Liu, Y., and Yin, W. Douglas–Rachford splitting and ADMM for pathological convex optimization. *Comput. Optim. Appl. 74*, 3 (2019), 747–778.

[222] Ryu, E. K., Taylor, A. B., Bergeling, C., and Giselsson, P. Operator splitting performance estimation: tight contraction factors and optimal parameter selection. *SIAM J. Optim. 30*, 3 (2020).

[223] Ryu, E. K., and Vũ, B. C. Finding the forward-Douglas–Rachford-forward method. *J. Optim. Theory Appl. 184*, 3 (2020), 858–876.

[224] Setzer, S., Steidl, G., and Teuber, T. Infimal convolution regularizations with discrete ℓ_1-type functionals. *Commun. Math. Sci. 9*, 3 (2011), 797–827.

[225] Shi, W., Ling, Q., Wu, G., and Yin, W. EXTRA: an exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim. 25*, 2 (2015), 944–966.

[226] Shi, W., Ling, Q., Wu, G., and Yin, W. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing 63*, 22 (2015), 6013–6023.

[227] Souza, J. C. O., Oliveira, P. R., and Soubeyran, A. Global convergence of a proximal linearized algorithm for difference of convex functions. *Optim. Lett. 10*, 7 (2016), 1529–1539.

[228] Stollnitz, E. J., DeRose, A. D., and Salesin, D. H. Wavelets for computer graphics: A primer, part 1. *IEEE Comput. Graph. Appl. 15*, 3 (1995), 76–84.

[229] Sun, W.-Y., Sampaio, R. J. B., and Candido, M. A. B. Proximal point algorithm for minimization of DC function. *J. Comput. Math. 21* (2003), 451–462.

[230] Svaiter, B. F. On weak convergence of the Douglas–Rachford method. *SIAM J. Control Optim. 49*, 1 (2011), 280–287.

[231] Tam, M. K. Frugal and decentralised resolvent splittings defined by nonexpansive operators. *Optim. Lett.* (2023).

[232] Tang, Y., Wen, M., and Zeng, T. Preconditioned three-operator splitting algorithm with applications to image restoration. *J. Sci. Comput. 92*, 3 (2022), Paper No. 106, 26.

[233] Tao, P. D., and Souad, E. B. Algorithms for solving a class of nonconvex optimization problems. Methods of subgradients. In *FERMAT days 85: mathematics for optimization (Toulouse, 1985)*, vol. 129 of *North-Holland Math. Stud.* North-Holland, Amsterdam, 1986, pp. 249–271.

[234] Tao, P. D., and Souad, E. B. Algorithms for solving a class of nonconvex optimization problems. Methods of subgradients. In *Fermat Days 85: Mathematics for Optimization*, J.-B. Hiriart-Urruty, Ed., vol. 129 of *North-Holland Mathematics Studies*. North-Holland, Springer, Berlin, 1986, pp. 249–271.

[235] Truong, T. T., and Nguyen, H.-T. Backtracking gradient descent method and some applications in large scale optimisation. Part 2: Algorithms and experiments. *Appl. Math. Optim. 84*, 3 (2021), 2557–2586.

[236] Tseng, P. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optim. 29*, 1 (1991), 119–138.

[237] Tseng, P. A modified forward-backward splitting method for maximal monotone mappings. *SIAM J. Control Optim. 38*, 2 (2000), 431–446.

[238] ur Rehman, J., Ahmad, N., Khalid, M., Gilani, Z. A., Ullah, I., Nasar, G., Akhtar, M. M., Usmani, M. N., et al. Intensity modulated radiation therapy: A review of current practice and future outlooks. *J. Radiat. Res. Appl. Sci. 11*, 4 (2018), 361–367.

[239] Vũ, B. C. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math. 38*, 3 (2013), 667–681.

[240] von Neumann, J. *Functional Operators II: The Geometry of Orthogonal Spaces*. Princeton University Press, 1950.

[241] Watanabe, Y., and Sakurama, K. Distributed optimization of clique-wise coupled problems via three-operator splitting. *Preprint, arXiv:2310.18625 [math.OC], 2023*.

[242] Wen, B., Chen, X., and Pong, T. K. A proximal difference-of-convex algorithm with extrapolation. *Comput. Optim. Appl. 69* (2018), 297–324.

[243] Wen, F., Chu, L., Liu, P., and Qiu, R. C. A survey on nonconvex regularization-based sparse and low-rank recovery in signal processing, statistics, and machine learning. *IEEE Access 6* (2018), 69883–69906.

[244] Wolkowicz, H., Saigal, R., and Vandenberghe, L. *Handbook of semidefinite programming: theory, algorithms, and applications*, vol. 27. Springer Science & Business Media, 2012.

[245] Yan, M. A new primal-dual algorithm for minimizing the sum of three functions with a linear operator. *J. Sci. Comput. 76*, 3 (2018), 1698–1717.

[246] Yan, M. A new primal-dual algorithm for minimizing the sum of three functions with a linear operator. *J. Sci. Comput. 76*, 3 (2018), 1698–1717.

[247] Yi, S., and Ryu, E. K. Convergence analyses of Davis–Yin splitting via scaled relative graphs II: Convex optimization problems. *Preprint, arXiv:2211.15604 [math.OC], 2022*.

[248] Zhang, T. Some sharp performance bounds for least squares regression with l_1 regularization. *Ann. Stat. 37*, 5A (2009), 2109–2144.

[249] Zhu, M., and Chan, T. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Reports 34* (2008), 8–34.

[250] Zong, C., Tang, Y., and Zhang, G. Solving monotone inclusions involving the sum of three maximally monotone operators and a cocoercive operator with applications. *Set-Valued Var. Anal. 31*, 2 (2023), [Paper No. 16], 20.

List of Algorithms

List of Figures

List of Tables

Notation and Symbols

Basic notation

$\mathbb{N} := \{0, 1, 2, \ldots\}$	set of nonnegative integers
$\mathbb{R} :=]-\infty, \infty[$	real line
$\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$	extended real line
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{R}_{++}$	set of (strictly) positive real numbers
$\mathcal{H}$, $\mathcal{G}$	real Hilbert space
$\mathcal{H}^n$	product Hilbert space, i.e., $\mathcal{H}^n = \mathcal{H} \times \overset{(n)}{\cdots} \times \mathcal{H}$
$\langle x, y \rangle$	inner product of the vectors x and y
$\|x\|$	norm of x induced by the inner product, i.e., $\|x\| = \sqrt{\langle x, x \rangle}$
$\|x\|_p$	p-norm of $x \in \mathbb{R}^n$ with $p \in \{1, 2, \ldots\} \cup \{\infty\}$
$\lvert x \rvert$	absolute value of x
$[x]_+$	positive part of x, i.e., $[x]_+ = \max\{x, 0\}$
$\mathbb{B}_\varepsilon(\bar{x})$	closed ball centered at $\bar{x}$ with radius $\varepsilon > 0$
$x^k \rightharpoonup \bar{x}$	the sequence $(x^k)_{k \in \mathbb{N}}$ converges weakly to the point $\bar{x}$
$x^k \to \bar{x}$	the sequence $(x^k)_{k \in \mathbb{N}}$ converges strongly to the point $\bar{x}$
$[\![k, l]\!]$	set of integers between two integer numbers k and l
$\odot$	component-wise product
$\otimes$	Kronecker matrix product

Sets

$C \times D$	Cartesian product of the sets C and D
$\mathcal{H} \setminus C$	complementary of the set $C \subseteq \mathcal{H}$
$\operatorname{aff} C$	affine hull of the set C
$\operatorname{cone} C$	conical hull generated by the set C
$\operatorname{co} C$	convex hull of the set C

$d(\cdot, C)$	distance function to the set C
i_C	indicator function of the set C
$\operatorname{int} C$	interior of the set C
P_C	projector onto the set C
R_C	reflector with respect to the set C
$\mathtt{A}_C$	Asplund function associated to the set C
$\operatorname{ri} C$	relative interior of the set C
$\operatorname{span} C$	span of the set C
$\operatorname{sri} C$	strong relative interior of the set C
Δ_n	diagonal subspace of $\mathcal{H}^n$
$\dim U$	dimension of the linear subspace U

Functions

$f : \mathcal{H} \to \overline{\mathbb{R}}$	extended real-valued function
f^*	convex conjugate of a function f
$\operatorname{dom} f$	domain of the function f
∂f	basic (resp, convex) subdifferential of the (resp, convex) function f
$\hat{\partial} f$	regular subdifferential of the function f
$\partial_C f$	Clarke's generalized gradient of the function f
$\operatorname{prox}_{\gamma f}$	proximity operator of the function f with parameter γ
$d^+ f(\cdot, \cdot)$	upper Dini directional derivative of the function f

Operators

$A : \mathcal{H} \rightrightarrows \mathcal{H}$	set-valued operator from in $\mathcal{H}$
$T : \mathcal{H} \to \mathcal{H}$	single-valued operator in $\mathcal{H}$
Id	identity mapping
A^{-1}	inverse operator of A
$\operatorname{dom} A$	domain of the operator A
$\operatorname{Fix} A$	set of fixed points of the operator A
$\operatorname{gra} A$	graph of the operator A
$\operatorname{ran} A$	range of the operator A
$\operatorname{zer} A$	set of zeros of the operator A

$\|L\|$	norm of a linear operator L
L^*	adjoint of a linear bounded operator L
$J_{\gamma A}$	resolvent of the operator A with parameter $\gamma > 0$
$R_{\gamma A}$	reflected resolvent of the operator A with parameter $\gamma > 0$
${}_w A$	inner w-perturbation of the operator A
$A^{(\theta,\sigma)}$	(θ, σ)-strengthening of the operator A

Matrices

$\mathbb{R}^{n\times m}$	vector space of $n \times m$ real matrices
I	identity matrix
M^T	transpose of the matrix M
M^{-1}	inverse of the matrix M
$\ker M$	kernel of the matrix M
$\operatorname{rank} M$	rank of the matrix M
$\rho(M)$	spectral radius of the matrix M

Fixed point encodings

$\mathcal{T}_{DR}$	Douglas–Rachford splitting operator
$\mathcal{T}_{PR}$	fixed point operator associated to Pierra's product space reformulation
$\mathcal{T}_{FB}$	fixed point operator of minimal lifting forward-backward algorithm
$\mathcal{T}_{\mathcal{PD}}$	fixed point operator of minimal lifting primal-dual algorithm
$\mathcal{T}_{FRB}$	fixed point operator of the reduced lifting forward-reflected-backward

Index of Topics