A Successive Proximal DC Penalty Method with an Application to Mathematical Programs with Complementarity Constraints

Dominic C. Flocco, Steven A. Gabriel, Trine K. Boomsma, Martin Schmidt, Miguel A. Lejeune

Abstract. We develop a successive, proximal difference-of-convex (DC) function penalty method for solving DC programs with DC constraints. The proposed approach relies on a DC penalty function that measures the violation of constraints and leads to a penalty reformulation sharing the same solution set as the original problem. The resulting penalty problem is a DC program with convex constraints and is solved using a proximal DC algorithm (DCA) with successive DC decomposition. We establish convergence to a stationary point of the original problem under milder constraint qualifications than those typically required in the DC programming literature. We then specialize the framework to mathematical programs with complementarity constraints (MPCCs), which cover a large class of optimization and equilibrium problems arising in engineering, economics, and game theory. For this application, we introduce a novel DC penalty function that penalizes violations of the complementarity constraints and establish a correspondence between S-stationary points of the MPCC and stationary points of the resulting DC penalty reformulation. Numerical experiments on MPCCs arising from bilevel optimization show that the proposed successive, proximal DCA, referred to as SPDCA, reliably computes feasible solutions and achieves good solution quality, while often requiring significantly less runtime than conventional mixed-integer formulations, particularly on instances with quadratic upper-level DC objectives.

1. Introduction

Many optimization problems contain functions $F = G - H$ that can be expressed as a difference-of-convex (DC) functions G and H. As a result, DC programming represents a quickly growing sub-field of nonlinear optimization; see, e.g., Le Thi (2020), Le Thi et al. (2019), Le Thi and Pham (2023, 2018), and Oliveira (2020). For such DC optimization problems, the objective function and possibly some constraints are DC functions. The DC algorithm (DCA) and its variants start with a particular DC decomposition $F = G - H$ and solve a convex problem at each iteration, using the approximation $\hat{F}(x) := G(x) - \hat{H}(x)$, where $\hat{H}$ is a linear approximation of H (An and Tao 2005; Pham and Le Thi 1997). When the DC decomposition changes at each iteration k, i.e., $F^k = G^k - H^k$, we refer to the method as one with successive DC decomposition. In recent years, DC optimization has emerged as a promising approach for handling nonconvex problems by exploiting their DC structure. The approach has shown to be successful in solving nonconvex quadratic programs (Pham et al. 2008), binary linear and quadratic programs (Le Thi and Pham 2001), nonlinear bilevel programs (Hoai An et al. 2009), and complementarity problems (Gabriel et al. 2025; Le Thi and Pham 2011), to just name a few.

Date: March 12, 2026.

2020 *Mathematics Subject Classification.* 90C26, 90C33, 91A65.

Key words and phrases. DC programming, Mathematical programs with complementarity problems, Penalty methods, Convergence analysis, Bilevel optimization.

This work considers the specific class of DC programs with DC equality constraints

$$\xi = \underset{z\in C}{\text{minimize}}\ f(z) \quad \text{s.t.} \quad \varphi_i(z) = 0, \quad i = 1, \dots, n, \tag{1.1}$$

where $f : \mathbb{R}^N \to \mathbb{R}$ is assumed to be a continuous DC function with decomposition $f = g - h$ with $g, h : \mathbb{R}^N \to \mathbb{R}$ being convex functions. The set $C \subseteq \mathbb{R}^N$ is assumed to be non-empty, convex, and compact, representing the "well-behaved" constraints, and $\varphi_i : \mathbb{R}^N \to \mathbb{R}$ define nonconvex, DC equality constraints. As introduced by Le Thi et al. (2014), we consider the so-called DC penalty version

$$\xi(\gamma) = \underset{z\in C}{\text{minimize}}\ F(z) := f(z) + \gamma\theta(z), \tag{1.2}$$

of this problem, where $\theta : \mathbb{R}^N \to \mathbb{R}_{\geq 0}$ is the DC penalty function with decomposition $\theta = \phi - \psi$, i.e., ϕ and ψ are convex functions. The penalty function has the property that $\theta(z) = 0$ if and only if $\varphi_i(z) = 0$ for all $i = 1, \dots, n$. Moreover, we assume that the penalty function θ is such that that any stationary point $\bar{z} \in C$ of θ satisfies $\theta(\bar{z}) = 0$. This property holds for any nonnegative convex function with a zero minimum, and, more generally, functions whose stationary points are global minimizers with value 0. During successive DC decomposition, the parameter $\gamma > 0$ may be updated at each iteration of the DCA to dynamically penalize constraint violations. The benefit of considering (1.2) is that the "troublesome" constraints are now part of the objective and the feasible region is convex and compact. It is known that (1.2) is an exact penalty reformulation of (1.1), i.e., there is a sufficiently large sufficiently large $\gamma_0 \geq 0$, such that for all $\gamma > \gamma_0$, (1.2) and (1.1) share the same optimal value and optimal solution set; see, e.g., Le Thi et al. (2012a). In this work, we develop a proximal DCA with successive decomposition, referred to as SPDCA, to solve (1.2), prove its convergence to a stationary point of (1.1), and apply the framework to the special case of mathematical programs with complementarity constraints (MPCC).

1.1. **Application to MPCCs.** MPCCs represent a large class of optimization and equilibrium problems, which may be expressed in the form (1.1). These models arise naturally in engineering, economics, and game theory, where complementarity constraints often arise from the Karush–Kuhn–Tucker (KKT) optimality conditions of one or many convex optimization problems (Cottle et al. 2009; Francisco and Pang 2003; Zhi-Quan et al. 1996). MPCCs are challenging to solve due to their inherent nonconvexity and failure of standard constraint qualifications (Zhi-Quan et al. 1996). Traditional nonlinear optimization and combinatorial techniques often fail to converge for realistic problem instances or suffer from poor scaling. While the development of the successive proximal DCA and the corresponding convergence result is applicable to any DC penalty problem of the form (1.2), we later specialize its application to MPCCs with a DC objective function (MPCC-DC). Along these lines, we present a novel DC reformulation of the MPCC that relies on a DC penalty function θ that penalizes complementarity constraint violations. Finally, the approach is applied to two classes of MPCCs that arise from bilevel optimization problems with convex lower levels: (i) those with linear upper and lower levels, and (ii) those with quadratic upper and lower levels, where the upper-level objective is a DC function.

In general, we consider the MPCC-DC of the form

$$\xi = \min_{(x,y)\in C} \quad f(x, y) \tag{1.3a}$$

$$\text{s.t.} \quad \varphi_i(x, y) := x_i y_i = 0, \quad i = 1, \dots, n, \tag{1.3b}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$, and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a smooth DC function with decomposition $f(x, y) := g(x, y) - h(x, y)$, i.e., $g, h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ are convex functions. We assume that the set C is given by

$$C = \{(x, y) \colon r(x, y) \le 0,\ q(x, y) = 0,\ x \ge 0,\ y \ge 0\}, \tag{1.4}$$

where $r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m_1}$ is component-wise convex and smooth, and $q : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{m_2}$ is affine in both x and y. Note that C is closed and convex. We assume that C is non-empty and bounded, thus compact. Together, (1.3b) and the nonnegativity constraints on x and y represent complementarity constraints, sometimes abbreviated as $0 \le x \perp y \ge 0$. Such complementarity constraints may arise as part of the KKT conditions of an associated optimization problem. For example, problems of the form (1.3) may arise from bilevel optimization problems with a convex lower level (Dempe 2002; Kim et al. 2020; Zhi-Quan et al. 1996) or from mixed complementarity problems (Cottle et al. 2009; Gabriel et al. 2012).

Besides nonconvexity, the difficulty in solving (1.3) is the failure of standard constraint qualifications (CQ) due to the complementarity constraints. We address this difficulty by defining a suitable DC penalty function θ for the bilinear constraints (1.3b). The resulting penalty problem is thus

$$\xi(\gamma) = \min_{(x,y)\in C} \quad F(x, y) := f(x, y) + \gamma\theta_p(x, y) \tag{1.5}$$

where $\theta_p : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\ge 0}$ is an ℓ_p-norm DC penalty function with decomposition $\theta_p = \phi_p - \psi_p$. This penalty function should have the property that $\theta_p(x, y) = 0$ if and only if $x_i y_i = 0$ holds for all $i = 1, \dots, n$. An additional benefit of considering (1.5) is that, assuming that a CQ holds on the convex set C, a CQ holds for (1.5) since the troublesome constraints are now part of the objective. The resulting problem is a DC program with convex constraints of the form in (1.4), which we may solve with the proposed SPDCA. It is well-known from the MPCC literature that the problem (1.5) is an exact penalty reformulation of (1.3); see, e.g., Hu and Ralph (2004), Liu et al. (2008), Lucidi and Rinaldi (2010), Mangasarian and Pang (1997), Ralph and Wright (2004), Scholtes and Stöhr (1999), and Ye et al. (1997).

The specialization of the proposed SPDCA to solve (1.5) builds upon the work of Gabriel et al. (2025), who develop a DC-based approach for MCPs using an ℓ_∞-norm penalty function. The resulting penalty problem is a DC program with a linear objective and DC constraints. We extend this to consider a general ℓ_p-norm penalty function and by applying the method to the broader class of MPCCs with DC objectives. Moreover, in contrast to and extending the work by Jara-Moroni et al. (2018), Le Thi et al. (2023), and Le Thi and Pham (2011), the proposed approach expresses the complementarity relationship $x_i y_i = 0$ via a DC reformulation of bilinear terms and minimizes an updated penalty term in a sequential quadratic programming approach. Relative to the methods of Le Thi et al. (2012b) and Liu et al. (2024), the proposed approach applies to a broader class of bilevel problems. Namely, it accommodates problems with a DC quadratic upper-level objective and convex quadratic lower-level objective.

1.2. **Summary of Contributions.** The proposed approach for solving DC programs with DC equality constraints relies on successive decomposition of the DC penalty function and uses a proximal DCA to solve the resulting penalty problem. For the application of the framework to MPCCs, we carry out extensive numerical experiments on problems arising from bilevel optimization and demonstrate applicability of the proposed method for linear and quadratic upper and lower levels. Our main contributions are twofold.

(i) We prove global convergence to a stationary point of the general DC program (1.1) using the proposed SPDCA for the DC penalty reformulation (1.2), extending the result of Souza et al. (2016) to the case of successive DC decomposition. In this scheme, an adaptive penalty parameter is used to balance the original objective and the penalization of constraint violations. By applying proximal DCA, we ensure that the convex subproblem solved at each iteration has a strongly convex objective. The analysis applies to any suitably defined strongly convex proximal term defined by a symmetric and positive definite matrix, which allows for the adoption of, e.g., linear or quasi-Newton DCA proximal methods such as those presented by Rakotomamonjy et al. (2016) and Sun et al. (2003).

(ii) We present a novel DC penalty reformulation for general MPCCs (1.3), recasting the bilinear constraints $x_i y_i = 0$ as the difference of smooth convex quadratic functions and thereby generalizing previous suggestions from the complementarity literature (Constante-Flores et al. 2022; Gabriel et al. 2006; Siddiqui and Gabriel 2013). We establish a correspondence between strong stationary (S-stationary) points of the MPCC and stationary points of the resulting DC penalty reformulation. The choice of the DC decomposition allows us to efficiently evaluate subdifferentials as gradients and obtain linear approximations by uniquely determined supporting hyperplanes to the concave part of a DC function.

The remainder of the paper is organized as follows. In Section 2, we provide a brief introduction to DC optimization and penalty methods. In Section 3, we introduce the SDPCA framework and present the main convergence result in the context of the general DC program (1.1) in Section 4. In Sections 5 and 6 we apply the framework to MPCCs (1.3), first introducing the complementarity-tailored penalty function and then presenting the DC reformulation of the penalty problem. In Section 7 we apply the proposed algorithm to a set of MPCC test problems arising from linear and nonlinear bilevel optimization problems to exhibit its performance. Finally, we close this paper with some concluding remarks in Section 8.

2. Preliminaries

In this section, we review the necessary preliminaries on optimality conditions for general DC programs and MPCCs.

2.1. Optimality Conditions for DC Programs.

The theoretical analysis of DC programs relies on foundational tools from convex analysis, variational analysis, and duality theory.

For a comprehensive background on these concepts, the reader is referred to Bazaraa et al. (2006), Boyd and Vandenberghe (2004), and Rockafellar and Wets (2009). The following results on DC optimality conditions are presented without proofs. We refer the reader to Oliveira (2020) and the references therein for further details.

First, we require some basic concepts in convex analysis. The subdifferential of a convex function $\varphi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ at a point x generalizes the concept of a gradient for smooth functions and is defined by

$$\partial\varphi(x) := \{z \in \mathbb{R}^n : \varphi(y) \geq \varphi(x) + \langle z, y - x\rangle \ \forall y \in \mathbb{R}^n\}.$$

The subdifferential is defined for all $x \in \operatorname{dom}(\varphi) := \{x \in \mathbb{R}^n : \varphi(x) < \infty\}$. The directional derivative of φ at a point x in the direction d is defined by

$$\varphi'(x; d) := \lim_{t \downarrow 0} \frac{\varphi(x + td) - \varphi(x)}{t}.$$

Note that $z \in \partial\varphi(x)$ if and only if $\langle z, d\rangle \leq \varphi'(x; d)$ for all $d \in \mathbb{R}^n$. When computing subgradients, the notion of inexactness will be useful for our discussion. To this end, we define the ε-subdifferential for $\varepsilon \geq 0$ as

$$\partial_\varepsilon\varphi(x) := \{z \in \mathbb{R}^n : \varphi(y) \geq \varphi(x) + \langle z, y - x\rangle - \varepsilon \ \forall y \in C\}.$$

Clearly, if $\varepsilon = 0$, then $\partial_\varepsilon \varphi(x) = \partial \varphi(x)$. These notions are important for defining optimality conditions of general DC optimization problems.

We now consider the unconstrained DC program

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ F(x) := G(x) - H(x), \tag{2.1}$$

where $G, H : \mathbb{R}^n \to \mathbb{R}$ are proper lower semi-continuous convex functions, i.e., the domains of G and H are nonempty and $G(x), H(x) > -\infty$ holds for all x. Our subsequent convergence analysis considers the unconstrained version of the general DC program (1.2) using an indicator function, which is common in the DC literature.

We begin with a classic characterization of global solutions of a general DC program (2.1).

Theorem 2.1 (Oliveira (2020))**.** *A vector $\bar{x} \in \operatorname{ri} \operatorname{dom} F$, i.e., in the relative interior of $\operatorname{dom} F$, is a global solution of problem (2.1) if and only if*

$$\emptyset \neq \partial_\varepsilon H(\bar{x}) \subseteq \partial_\varepsilon G(\bar{x}) \quad \forall \varepsilon \geq 0. \tag{2.2}$$

Important here is the inclusion for all $\varepsilon \geq 0$, which characterizes the global optimality of a solution. If this inclusion holds only for a finite ε, then we say $\bar{x}$ is a local solution. Specifically, we have the following result.

Theorem 2.2 (Oliveira (2020))**.** *Let H be finite-valued in a neighborhood of $\bar{x} \in \operatorname{dom} G$. Suppose there exists an $\varepsilon^* > 0$ such that*

$$\emptyset \neq \partial_\varepsilon H(\bar{x}) \subseteq \partial_\varepsilon G(\bar{x}) \quad \forall \varepsilon \in [0, \varepsilon^*]. \tag{2.3}$$

Then, $\bar{x}$ is local solution to problem (2.1).

The complexity of DC programs, as in general nonconvex optimization, lies in the distinction between local and global solutions. The inclusions in (2.2) and (2.3) are difficult to check in practice because the ε-subdifferential usually does not have a closed, analytical form in general. It is often easier to determine if the inclusion in (2.2) and (2.3) hold for $\varepsilon = 0$, i.e., for the regular subdifferential. From this arises the important, and more easily verifiable, notion of stationarity and criticality.

Definition 2.1. *A vector $\bar{x} \in \operatorname{dom} G$ is said to be a d-stationary point of (2.1) if*

$$\emptyset \neq \partial H(\bar{x}) \subseteq \partial G(\bar{x}). \tag{2.4}$$

The definition of d-stationarity arises from the inclusion in (2.3) with $\varepsilon = 0$. For a point $\bar{x} \in (\operatorname{int} \operatorname{dom} H) \cap (\operatorname{int} \operatorname{dom} G)$, the inclusion in (2.4) is equivalent to $F'(\bar{x}; d) \geq 0$ for all directions $d \in \mathbb{R}^n$, i.e., the directional derivative of f at $\bar{x}$ is nonnegative for all directions, hence, the term "directional stationarity". Clearly, every local solution is a d-stationary point. However, the converse does not hold in general. The relationship is illustrated in the left-hand side of Figure 1. Along these lines, an even more useful but weaker condition is the notion of criticality of a vector $\bar{x}$.

Definition 2.2. *A vector $\bar{x} \in \operatorname{dom} G$ is a critical point of problem (2.1) if*

$$0 \in \partial G(\bar{x}) - \partial H(\bar{x}) \iff \partial H(\bar{x}) \cap \partial G(\bar{x}) \neq \emptyset. \tag{2.5}$$

Clearly, all d-stationary points are critical points. Since d-stationarity is the strongest notion of stationarity, it is often referred to as strong criticality in the literature. We note that if H is smooth, criticality implies d-stationarity since (2.5) implies $\partial H(\bar{x}) = \{\nabla H(\bar{x})\} \subseteq \partial G(\bar{x})$. This point is illustrated in the left-hand side of Figure 1. In the application to MPCCs discussed below, H is smooth, and consequently, d-stationarity and criticality are equivalent.

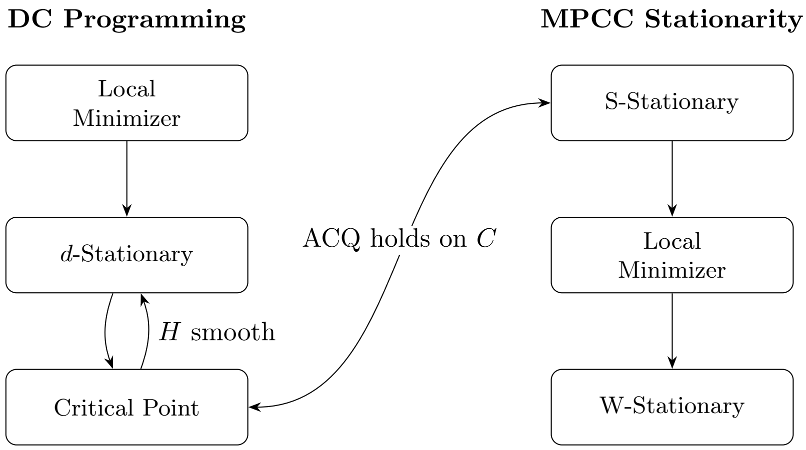


FIGURE 1. Relationship between DC Programming and MPCC notions of stationarity.

2.2. **Stationarity Conditions for MPCCs.** For the specialization to MPCCs, we introduce and analyze notions of MPCC stationarity, linking them directly to the DC literature. We utilize the classic concepts of weak and strong stationarity, which are pervasive in the MPCC field; see, e.g., Schwartz (2011) and Zhi-Quan et al. (1996). The analysis of MPCC stationarity is essential because the standard KKT conditions do not apply due to the lack of a common CQ holding for (1.3). In fact, due to the complementarity constraints on x and y, typical CQs, such as the linear independence CQ or the Mangasarian–Fromovitz CQ, do not hold for the MPCC (1.3) in general. The only standard constraint qualification, which is satisfied in general is the Guignard CQ (Schwartz 2011). The Abadie CQ (ACQ) holds at points satisfying strict complementarity, i.e., either $x_i > 0$ or $y_i > 0$ for all $i = 1, \dots, n$ (Flegel and Kanzow 2005). Note that, when considering MPCCs, we assume smoothness of the relevant functions. However, this assumption is relaxed in the general algorithmic description.

Definition 2.3. *Let (x^*, y^*) be feasible to the MPCC-DC (1.3). Then, (x^*, y^*) is said to be* weakly stationary (W-stationary)*, if there exist multipliers $\lambda \in \mathbb{R}^\ell$, $\nu \in \mathbb{R}^r$, $\mu, \xi, \zeta \in \mathbb{R}^n$, such that*

$$0 = \nabla_x f(x^*, y^*) + \sum_{i=1}^{\ell} \lambda_i \nabla_x r_i(x^*, y^*) + \sum_{j=1}^{r} \nu_j \nabla_x q_j(x^*, y^*) + \sum_{i=1}^{n} \left(\mu_i \nabla_x \varphi_i(x^*, y^*) - \xi_i\right), \tag{2.6a}$$

$$0 = \nabla_y f(x^*, y^*) + \sum_{i=1}^{\ell} \lambda_i \nabla_y r_i(x^*, y^*) + \sum_{j=1}^{r} \nu_j \nabla_y q_j(x^*, y^*) + \sum_{i=1}^{n} \left(\mu_i \nabla_y \varphi_i(x^*, y^*) - \zeta_i\right), \tag{2.6b}$$

$$\lambda_i \geq 0, \ r_i(x^*, y^*)\lambda_i = 0, \quad i = 1, \dots, \ell, \tag{2.6c}$$

$$\xi_i = 0, \quad i \in \mathcal{I}_y(x^*, y^*) := \{i \colon x_i^* > 0, y_i^* = 0\}, \tag{2.6d}$$

$$\zeta_i = 0, \quad i \in \mathcal{I}_x(x^*, y^*) := \{i \colon x_i^* = 0, y_i^* > 0\}. \tag{2.6e}$$

If, in addition, $\xi_i, \zeta_i \geq 0$ *holds for all* $i \in \mathcal{I}_0(x^*, y^*) := \{i\colon x_i^* = 0, y_i^* = 0\}$*, then* (x^*, y^*) *is said to be strongly stationary (S-stationary).*

As we will see, the benefit of the proposed penalty problem (1.5) is that the (complicating) complementarity constraints are transferred from the feasible set to the objective, allowing standard CQs to hold for the penalty problem. Along these lines, we make the following mild assumption on the convex and compact set C, which we use as a standing assumption for the remainder of the paper.

Assumption 1. *The ACQ holds, i.e., the tangential cone equals the linearized cone, for all feasible* $(x, y) \in C$ *to Problem* (1.5).

Based on this assumption, the KKT conditions constitute necessary optimality conditions for the penalty problem (1.5). Specifically, given a feasible $(x^*, y^*) \in C$ to the penalty problem (1.5), there exist multipliers $\lambda \in \mathbb{R}^\ell$, $\nu \in \mathbb{R}^r$, $\mu, \xi, \zeta \in \mathbb{R}^n$, such that

$$0 = \nabla_x f(x^*, y^*) + \gamma \nabla_y \theta(x, y) + \sum_{i=1}^{\ell} \lambda_i \nabla_x r_i(x^*, y^*) + \sum_{j=1}^{r} \nu_j \nabla_x q_j(x^*, y^*) - \sum_{i=1}^{n} \xi_i, \tag{2.7a}$$

$$0 = \nabla_y f(x^*, y^*) + \gamma \nabla_y \theta(x, y) + \sum_{i=1}^{\ell} \lambda_i \nabla_y r_i(x^*, y^*) + \sum_{j=1}^{r} \nu_j \nabla_y q_j(x^*, y^*) - \sum_{i=1}^{n} \zeta_i, \tag{2.7b}$$

$$\lambda_i \geq 0, \; r_i(x^*, y^*)\lambda_i = 0 \quad i = 1, \ldots, \ell, \tag{2.7c}$$

$$\xi_i \geq 0, \; x_i \xi_i = 0 \quad i = 1, \ldots, n, \tag{2.7d}$$

$$\zeta_i \geq 0, \; y_i \zeta_i = 0 \quad i = 1, \ldots, n. \tag{2.7e}$$

The relationship between S-Stationary points, local minimizers, and W-Stationary points are well documented in the MPCC literature; see, e.g., Jara-Moroni et al. (2018). This hierarchy is illustrated in the right-hand side of Figure 1.

Recall that our subsequent convergence analysis considers the unconstrained version of the general DC program (1.2) using an indicator function. When applied to MPCC, the convergence analysis relies on the following equivalence. Under the ACQ on C (Assumption 1), a critical point of this unconstrained DC program, i.e., a point satisfying (2.5), corresponds to an S-Stationary point of the MPCC (Definition 2.3). This is illustrated in Figure 1 and discussed in detail in Section 5. By demonstrating convergence to a critical point of the unconstrained DC program in the following Section 3, we effectively establish convergence to an S-Stationary point of the MPCC.

3. The Algorithm

We now develop the SPDCA for solving the DC program (1.1) via its penalty reformulation (1.2). The DCA is based on successive DC decomposition of the DC objective $F(z) = f(z) + \gamma\theta(z)$, where $f = g - h$ and $\theta = \phi - \psi$ are DC functions. Let $F^k = G^k - H^k$ be the DC decomposition of the penalty objective at iteration k. The penalty parameter γ_k is updated at each iteration k, which produces the DC decomposition

$$G^k(z) = g(z) + \gamma_k \phi(z) \text{ and } H^k(z) = h(z) + \gamma_k \psi(z) \tag{3.1}$$

at iteration k. As we will see below, the update rule for γ_k has great theoretical and practical impact. In particular, a desirable property of the update rule for γ_k is that the sequence (γ_k) is bounded. In addition, γ_k must become sufficiently large to effectively balance the two "competing" parts f and θ of the penalty objective.

Another desirable property of the penalty problem is the iterative improvement of the penalty term θ as γ_k increases, which is summarized in the following lemma.

Lemma 3.1 (Lemma 1 of Zangwill (1967))**.** *Let C be a non-empty, compact, and convex set. Define*

$$z_k^* \in \operatorname*{arg\,min}_{z \in C} \{f(z) + \gamma_k \theta(z)\}$$

as a global solution to (1.2) for some $\gamma_k > 0$ from a strictly increasing sequence $(\gamma_k) \subset \mathbb{R}_+$. Then, the sequence of penalty function values $\theta(z_k^*)$ is non-increasing, i.e.,

$$\theta(z_1^*) \geq \theta(z_2^*) \geq \cdots \geq 0.$$

At each iteration, we solve the convex approximation

$$\operatorname*{minimize}_{z \in C} \; G^k(z) - \left[H^k(z^k) + \langle w^k, z - z^k \rangle\right] \tag{3.2}$$

of (1.2), where $w^k \in \partial H^k(z^k) := \partial(h + \gamma_k \psi)(z^k)$.

The proposed algorithm makes use of a quadratic proximal term in the convex subproblem. The proximal term is governed by a symmetric and positive definite matrix $B^k \succ 0$, which can likewise be updated in each iteration depending on the proximal scheme employed. We append to the objective in (3.2) by the proximal term

$$\operatorname{prox}^k(z) = \frac{1}{2}\|z - z^k\|_{B^k}^2, \tag{3.3}$$

where $\|x\|_A = \sqrt{\langle x, x \rangle_A}$ is the norm induced by the matrix A with $\langle x, x \rangle_A = \langle x, Ax \rangle$. The prox operator (3.3) admits the gradient $\nabla \operatorname{prox}^k(z) = B^k(z - z^k)$. The fact that the matrix B^k induces a norm will be useful in the convergence analysis. Specifically, we use that $\|z\|_A^2 \geq \lambda_{\min}\|z\|_2^2$ holds for a symmetric and positive definite matrix $A \succ 0$, where $\lambda_{\min} > 0$ is the smallest eigenvalue of A.

The following are two choices for B^k that result in different variations of the basic DCA scheme.

(i) **Proximal Linear DCA (DCA-PL)**: If $B^k = \tau_k^{-1} I$, where $I \in \mathbb{R}^{N \times N}$ is the identity matrix and $\tau_k > 0$ is a prox-parameter, then Algorithm 1 becomes the linearized proximal method as it is discussed by Pang et al. (2017), Souza et al. (2016), and Sun et al. (2003).

(ii) **Proximal Newton DCA (DCA-PN)**: If B^k is a quasi-Newton matrix, i.e., it approximates the Hessian of G at iterate z^k, then Algorithm 1 becomes the proximal-Newton method as discussed by Rakotomamonjy et al. (2016).

We note that if B^k is the matrix of zeros, i.e., $B^k = 0$, then Algorithm 1 is the classic DCA scheme by Pham and Le Thi (1997). In this work, we extend the approaches by Rakotomamonjy et al. (2016) and Souza et al. (2016) by considering DCA with successive DC decomposition via the penalty parameter γ_k, as shown in Algorithm 1.

While Lines 3 and 9 allow for broad flexibility depending on the problem structure, their specific realizations for our test cases are discussed in Section 7. The scalars $\delta_1 > 1$ and $\delta_2 > 0$ govern the penalty parameter update scheme. Specifically, δ_1 determines the increase of the penalty parameter and δ_2 affects the frequency of penalty parameter updates.

Algorithm 1 Proximal DCA with Successive DC Decomposition (SPDCA)

Initialize a starting point $z^0 \in \mathbb{R}^N$, a penalty parameter $\gamma_0 > 0$, scalars $\delta_1 > 1$ and $\delta_2 > 0$, as well as a symmetric and positive definite matrix $B^0 \in \mathbb{R}^{N\times N}$ and a feasibility tolerance $\varepsilon > 0$.
for $k = 0, 1, \dots$ **do**
  Compute $w^k \in \partial H^k(z^k) = \partial(h + \gamma_k \psi)(z^k)$.
  Compute

$$z^{k+1} \in \arg\min_{z\in\mathcal{C}} \left\{ G^k(z) - \left[H^k(z^k) + \langle w^k, z - z^k\rangle\right] + \frac{1}{2}\|z - z^k\|^2_{B^k} \right\}. \tag{3.4}$$

  if $z^{k+1} = z^k$ and $\theta(z^{k+1}) \le \varepsilon$ **then**
    return z^{k+1}.
  end if
  Update the penalty parameter via

$$\gamma_{k+1} = \begin{cases} \delta_1\gamma_k, & \text{if } \gamma_k < \delta_2\|z^{k+1} - z^k\|^{-1} \text{ and } \theta(z^{k+1}) > \varepsilon, \\ \gamma_k, & \text{if } \gamma_k \ge \delta_2\|z^{k+1} - z^k\|^{-1} \text{ or } \theta(z^{k+1}) \le \varepsilon. \end{cases} \tag{3.5}$$

  Update prox-operator matrix $B^{k+1} \succ 0$.
end for

4. Convergence Analysis

To establish convergence to a critical point, we consider the unconstrained version of (1.2) through the introduction of the (convex) indicator function $\chi_C(z)$, which takes the value 0 if $z \in C$ and ∞ otherwise. Then, we augment the convex part of the DC decomposition (3.1) with $\chi_C(z)$, resulting in the equivalent DC program

$$\underset{z\in\mathbb{R}^n}{\text{minimize}}\ \ F^k(z) := G^k(z) + \chi_C(z) - H^k(z). \tag{4.1}$$

We denote by $\tilde{G}^k(z) = G^k(z) + \chi_C(z)$ the resulting convex part of the DC decomposition.

In addition, we make the following mild assumptions regarding Algorithm 1.

Assumption 2.

(i) *The eigenvalues of the symmetric and positive definite matrices B^k are strictly positive and bounded from above in the limit, i.e., there exists a $\rho > 0$ and $\Gamma < \infty$, such that*

$$\liminf_{k\to\infty} \lambda^k_{\min} \ge \rho \quad \text{and} \quad \limsup_{k\to\infty} \lambda^k_{\max} \le \Gamma,$$

where $\lambda^k_{\min}$ and $\lambda^k_{\max}$ are the smallest and largest eigenvalues of B^k.

(ii) *The functions g, h, ϕ, and ψ are continuous on the set C.*

(iii) *The penalty function θ is such that any stationary point $\bar{z} \in C$ of θ satisfies $\theta(\bar{z}) = 0$, i.e., $0 \in \partial\theta(\bar{z})$ implies $\theta(\bar{z}) = 0$ for any $\bar{z} \in C$.*

We note that the function f is bounded below on the set C, since it is compact, i.e., there exists a finite constant M such that $f(z) \ge M$ for all $z \in C$. Assumption 2(i) is easily guaranteed by the user-specified proximal-term matrix B^k and Assumption 2(ii) is satisfied by many functions encountered in practice. We further note that χ_C is lower semi-continuous, hence this assumption implies F^k and its convex decomposition functions G^k and H^k are continuous for all k; see Theorem 7.1 by Rockafellar (1970). Assumption 2(iii) holds for any nonnegative convex function with a zero minimum, e.g., it holds for norms and distance functions. More generally, it holds for any nonnegative function for which all stationary points are global minimizers with value 0, a property that characterizes the class of invex functions with minimum value 0, which generalize convex functions in the differentiable setting (Hanson 1981).

In the following discussion, we consider criticality in the context of DC programming. That is, z^* is a critical point of (1.2) if $\partial G(z^*) \cap \partial H(z^*) \neq \emptyset$ holds as stated in (2.5). The differentiability of G and H depends on the particular DC decomposition used, which depends both on the original objective $f = g - h$ and the penalty function $\theta = \phi - \psi$. For generality, the remaining discussion does not assume that G or H are differentiable everywhere. However, we note that in the case of H being differentiable, the criticality definition (2.5) implies the stronger notion of d-stationarity (2.4), as shown by Oliveira (2020). Nevertheless, this is not required for the following convergence analysis.

When considering the unconstrained version of (1.2), given by (4.1), DC-criticality implies

$$\begin{aligned} 0 \in \partial(G^k + \chi_C)(z) - \partial H^k(z) &= \partial G^k(z) + \mathcal{N}_C(z) - \partial H^k(z) \\ &= \partial \tilde{G}^k(z) - \partial H^k(z), \end{aligned} \tag{4.2}$$

where $\mathcal{N}_C(z) := \partial \chi_C(z) = \{\eta \in \mathbb{R}^N : \langle \eta, s - z \rangle \leq 0 \ \forall s \in C\}$ is the normal cone of the closed set $C \subset \mathbb{R}^N$ at the point $z \in C$. This cone is equivalent to the polar of the tangent cone (Rockafellar and Wets 2009, Theorem 6.28). Under the ACQ, this polar cone coincides with the linearized normal cone. By applying Farkas' Lemma to the linearized constraints at z, this cone is the conic hull of the active constraint gradients. This allows the right-hand side of (4.2) to be expressed as a weighted sum of gradients, equating the stationary condition with the KKT conditions of (1.2); see, e.g., Theorem 5.1.3 by Bazaraa et al. (2006). As a result, we consider the notion of DC criticality (2.5) applied to (4.1) under the assumption that the ACQ holds on C.

The following result shows that Algorithm 1 is a descent algorithm, and if the algorithm stops, it must stop at a critical point of (1.2).

Lemma 4.1. *The sequence (z^k) generated by Algorithm 1 has one of the following properties:*

- (i) *The algorithm stops at a critical point z^* of (1.2); i.e., a point satisfying (2.5).*
- (ii) *$F^{k_0}(z^k) = f(z^k) + \gamma_{k_0}\theta(z^k)$ is strictly decreasing for any fixed $\gamma_{k_0} > 0$; in particular, at least one of $f(z^{k+1}) < f(z^k)$ or $\theta(z^{k+1}) < \theta(z^k)$ holds at each iteration k.*

Proof. Let z^k be the current iterate and take $w^k \in \partial H^k(z^k)$. By (3.4), the new candidate solution satisfies

$$0 \in \partial \tilde{G}^k(z^{k+1}) - w^k + B^k(z^{k+1} - z^k) \implies w^k \in \partial \tilde{G}^k(z^{k+1}) + B^k(z^{k+1} - z^k).$$

If the algorithm stops at iteration $k+1$, then $z^{k+1} = z^k$, which implies $w^k \in \tilde{G}^k(z^{k+1}) = \tilde{G}^k(z^k)$. Therefore, since $w^k \in \partial H^k(z^k)$ by design, $w^k \in \partial \tilde{G}^k(z^k) \cap \partial H^k(z^k)$ holds, which implies $\partial \tilde{G}^k(z^k) \cap \partial H^k(z^k) \neq \emptyset$. So by (2.5), z^k is a critical point.

Now suppose $z^{k+1} \neq z^k$. Since H^k is convex and $w^k \in \partial H^k(z^k)$, we have

$$H^k(z^{k+1}) \geq H^k(z^k) + \langle w^k, z^{k+1} - z^k \rangle. \tag{4.3}$$

Moreover, since z^{k+1} is a minimizer of subproblem (3.4), we have

$$\begin{aligned} &\tilde{G}^k(z^{k+1}) - \left[H^k(z^k) + \langle w^k, z^{k+1} - z^k \rangle\right] + \frac{1}{2}\|z^{k+1} - z^k\|^2_{B^k} \\ \leq\ &\tilde{G}^k(z^k) - \left[H^k(z^k) + \langle w^k, z^k - z^k \rangle\right] + \frac{1}{2}\|z^k - z^k\|^2_{B^k}, \end{aligned}$$

which implies

$$\tilde{G}^k(z^k) \geq \tilde{G}^k(z^{k+1}) - \langle w^k, z^{k+1} - z^k\rangle + \frac{1}{2}\|z^{k+1} - z^k\|^2_{B^k}. \tag{4.4}$$

We now consider $k = k_0$ (note that k_0 can be chosen arbitrarily) and add the inequalities in (4.3) and (4.4). Since $z^k, z^{k+1} \in C$ holds so that we have $\chi_C(z^k) = \chi_C(z^{k+1}) = 0$, we obtain

$$F^k(z^k) - F^k(z^{k+1}) \geq \frac{1}{2}\|z^{k+1} - z^k\|^2_{B^k}. \tag{4.5}$$

Since $z^{k+1} \neq z^k$ and $B^k \succ 0$, we obtain $F^k(z^{k+1}) < F^k(z^k)$, i.e.,

$$f(z^{k+1}) + \gamma_k\theta(z^{k+1}) < f(z^k) + \gamma_k\theta(z^k).$$

□

In Lemma 4.3 below, the boundedness of the penalty parameter γ_k is established, i.e., there exists an iteration k_0 such that $\gamma_k = \gamma_{k_0}$ for all $k \geq k_0$. Hence, Lemma 4.1(ii), together with Lemma 3.1 and F^k being bounded from below, implies that the sequence $(F^k(z^k))$ is convergent. Next, we show that if Algorithm 1 does not stop, it generates an infinite sequence of iterates (z^k) such that $\lim_{k\to\infty}\|z^{k+1} - z^k\| = 0$, extending the result of Souza et al. (2016) to the case of a general proximal term and successive DC decomposition.

Lemma 4.2. *Suppose Assumption 2(i)–(ii) holds and let (z^k) be an infinite sequence of iterates generated by Algorithm 1 with bounded penalty parameters $(\gamma_k) \uparrow \gamma_* < \infty$. Then, $\lim_{k\to\infty}\|z^{k+1} - z^k\| = 0$ holds.*

Proof. Let (z^k) be an infinite sequence generated by Algorithm 1. This means that $z^k \neq z^{k+1}$ holds for all k. From (4.5), we have

$$[f(z^k) + \gamma_k\theta(z^k)] - [f(z^{k+1}) + \gamma_k\theta(z^{k+1})] \geq \frac{1}{2}\|z^{k+1} - z^k\|^2_{B^k} > 0.$$

Summing this inequality from $k = 0$ to $k = n-1$ for $n > 1$ yields

$$\sum_{k=0}^{n-1}[f(z^k) + \gamma_k\theta(z^k)] - [f(z^{k+1}) + \gamma_k\theta(z^{k+1})] \geq \frac{1}{2}\sum_{k=0}^{n-1}\|z^{k+1} - z^k\|^2_{B^k}. \tag{4.6}$$

By Lemma 3.1, we have that $\theta(z^{k+1}) \leq \theta(z^k)$ for all k. Also, (γ_k) is a non-decreasing sequence; therefore, the left-hand side reduces to

$$\begin{aligned} & f(z^0) - f(z^n) + \sum_{k=0}^{n-1}\gamma_k[\theta(z^k) - \theta(z^{k+1})] \\ \leq\ & f(z^0) - f(z^n) + \gamma_*\sum_{k=0}^{n-1}[\theta(z^k) - \theta(z^{k+1})] \\ =\ & f(z^0) - f(z^n) + \gamma_*[\theta(z^0) - \theta(z^n)] \\ =\ & [f(z^0) + \gamma_*\theta(z^0)] - [f(z^n) + \gamma_*\theta(z^n)]. \end{aligned} \tag{4.7}$$

Moreover, using $B^k \succ 0$, the right-hand side of (4.6) simplifies to

$$\frac{1}{2}\sum_{k=0}^{n-1}\|z^{k+1} - z^k\|^2_{B^k} \geq \frac{1}{2}\sum_{k=0}^{n-1}\lambda^k_{\min}\|z^{k+1} - z^k\|^2. \tag{4.8}$$

Using these simplifications, we take the limit as $n \to \infty$ of (4.6), using (4.7) and (4.8) as well as the continuity of f and θ and obtain

$$\begin{aligned} &[f(z^0) + \gamma_* \theta(z^0)] - \lim_{n\to\infty} [f(z^n) + \gamma_* \theta(z^n)] \geq \lim_{n\to\infty} \frac{1}{2} \sum_{k=0}^{n-1} \lambda_{\min}^k \|z^{k+1} - z^k\|^2 \\ \iff &[f(z^0) + \gamma_* \theta(z^0)] - [f(z^*) + \gamma_* \theta(z^*)] \geq \lim_{n\to\infty} \frac{1}{2} \sum_{k=0}^{n-1} \lambda_{\min}^k \|z^{k+1} - z^k\|^2. \end{aligned}$$

Since C is compact, f is bounded from below (and θ is bounded from below by definition), so the left-hand side is finite. Moreover, Assumption 2(i) implies $\lambda_{\min}^k > 0$ for all k. Together, this gives

$$\infty > \frac{1}{2} \sum_{k=0}^{\infty} \lambda_{\min}^k \|z^{k+1} - z^k\|^2 \geq \frac{\rho}{2} \sum_{k=0}^{\infty} \|z^{k+1} - z^k\|^2, \tag{4.9}$$

which follows from Assumption 2(ii). Hence, (4.9) implies $\lim_{k\to\infty} \|z^{k+1} - z^k\| = 0$. □

We have shown that either the algorithm stops at a critical point in a finite number of iterations, or generates an infinite sequence of iterates such that $\lim_{k\to\infty} \|z^{k+1} - z^k\| = 0$ holds. Next, we establish the boundedness of the penalty parameter γ_k, governed by the update rule (3.5).

Lemma 4.3. *Suppose Assumption 2 holds and let (γ_k) be an infinite sequence of penalty parameters generated by Algorithm 1, then there exists an index k_0 such that $\gamma_k = \gamma_{k_0}$ for all $k \geq k_0$.*

Proof. For contradiction, assume that the positive sequence (γ_k) is unbounded, i.e., $\lim_{k\to\infty} \gamma_k = \infty$. Let $(z^k) \to z^*$ be the sequence of iterates generated by Algorithm 1. Then, there exists an infinite subsequence $(\gamma_{k_j})_j$ with

$$\gamma_{k_j} < \delta_2 \|z^{k_j+1} - z^{k_j}\|^{-1} \quad \text{and} \quad \theta(z^{k_j+1}) > \varepsilon. \tag{4.10}$$

Let z^{k_j+1} be the solution to (3.4) in iteration k_j. Then, by (3.4), this point satisfies

$$\begin{aligned} &0 \in \partial G^{k_j}(z^{k_j+1}) - w^{k_j} + B^{k_j}(z^{k_j+1} - z^{k_j}) + \mathcal{N}_C(z^{k_j+1}) \\ \iff &0 \in \partial g(z^{k_j+1}) - \partial h(z^{k_j}) + \gamma_{k_j}[\partial\phi(z^{k_j+1}) - \partial\psi(z^{k_j})] \\ &\quad + B^{k_j}(z^{k_j+1} - z^{k_j}) + \mathcal{N}_C(z^{k_j+1}), \end{aligned}$$

where $w^{k_j} \in \partial H^{k_j}(z^{k_j})$. Division by γ_{k_j} gives

$$\begin{aligned} 0 \in &\frac{1}{\gamma_{k_j}} \left[\partial g(z^{k_j+1}) - \partial h(z^{k_j}) + B^k(z^{k_j+1} - z^{k_j}) + \mathcal{N}_C(z^{k_j+1})\right] \\ &+ \partial\phi(z^{k_j+1}) - \partial\psi(z^{k_j}). \end{aligned}$$

Under Assumption 2(i)–(ii), taking $j \to \infty$ and noting that $\lim_{k\to\infty} \gamma_k = \infty$, we have

$$0 \in \partial\phi(z^*) - \partial\psi(z^*) \iff 0 \in \partial\theta(z^*),$$

which means z^* is a stationary point of θ. By definition, $\theta(z^*) \geq 0$ holds and $\theta(z^*) > 0$ cannot hold since this would lead to an immediate contradiction with Assumption 2(iii) because $0 \in \partial\theta(z^*)$ implies $\theta(z^*) = 0$.

Hence, we have $\theta(z^*) = 0$. Since $(\theta(z^k))$ is non-increasing by Lemma 3.1 and continuous by Assumption 2(ii), it follows that $\theta(z^k) \to 0$ as $k \to \infty$. Therefore, there exists an index k_0 such that $\theta(z^{k_j+1}) \leq \varepsilon$ for all $k_j \geq k_0$. Consequently, the second condition in (4.10) cannot be satisfied for sufficiently large k_j, leading to the desired contradiction. Hence, there exists an index k_0, such that $\gamma_k = \gamma_{k_0}$ for all $k \geq k_0$. □

We aim to show that the infinite sequence of iterates (z^k) approaches a critical point z^* in the limit. First, we will need the following result regarding the subdifferential of the limiting functions. We state the result for a general N-dimensional Euclidean space so that it does not rely on any of the specifics of Algorithm 1.

Lemma 4.4. *Let $(z^k) \subset \mathbb{R}^N$ be a bounded sequence of points that converge to a point $z^* \in \mathbb{R}^N$, i.e., $\lim_{k\to\infty} z^k = z^*$. Furthermore, let (γ_k) be a sequence in $\mathbb{R}_+$ that converges to $\gamma_* > 0$. Let $f_1, f_2 : \mathbb{R}^N \to \mathbb{R}$ be convex and lower semi-continuous functions. If $w^k \in \partial(f_1 + \gamma_k f_2)(z^k)$ and $\lim_{k\to\infty} w^k = w^*$, then $w^* \in \partial(f_1 + \gamma_* f_2)(z^*)$.*

Proof. This result follows from the subdifferential sum rule (Rockafellar and Wets 2009, Corollary 10.9) and the closedness of the graph of the subdifferential operator (Rockafellar and Wets 2009, Theorem 8.6). □

We now use the preceding results to prove our main convergence theorem. Our convergence analysis adopts the standard notion used in the DC literature: if there exists a cluster point of the sequence of iterates (z^k), i.e., a limit of some convergent subsequence generated by Algorithm 1, then this cluster point is a critical point of Problem (1.2). Moreover, the stopping criterion of Algorithm 1 is eventually met by the infinite sequence of iterates, since the limit of the difference of successive iterates tends to zero, i.e., $\lim_{k\to\infty} \|z^{k+1} - z^k\| = 0$. This result is distinct from stronger convergence notions which, under more restrictive assumptions, guarantee the existence of a cluster point and ensure that the entire sequence (z^k) converges to a cluster point z^*.

Theorem 4.5. *Suppose Assumption 2 holds and let (z^k) be a sequence of iterates generated by Algorithm 1 with limit point z^* and the corresponding sequence (γ_k) of penalty parameters. Then, one of the two following statements holds.*

(i) *The algorithm stops in a finite number of iterations at a critical point z^* of (1.2) that satisfies $\theta(z^*) \leq \varepsilon$.*
(ii) *(z^k) is an infinite sequence with $\lim_{k\to\infty} \|z^{k+1} - z^k\| = 0$ and any cluster point z^*, if it exists, is a critical point of (1.2) satisfying $\theta(z^*) \leq \varepsilon$.*

Proof. If the algorithm terminates in a finite number of iterations at a point z^*, Lemma 4.1 shows that z^* is a critical point of (1.2), proving Part (i).

For Part (ii), let z^* be a cluster point of the infinite sequence (z^k) generated by Algorithm 1 and let w^* be a cluster point of the sequence (w^k) with $w^k \in \partial(h + \gamma_k \psi)(z^k)$. Moreover, let γ_* be a cluster point of the sequence of non-decreasing penalty parameters (γ_k), which is finite by Lemma 4.3. Under Assumption 2(ii), h and ψ are convex and continuous, so Lemma 4.4 implies $w^* \in \partial(h + \gamma_* \psi)(z^*)$. We now argue that $w^* \in \partial(g + \gamma_* \phi + \chi_C)(z^*)$ holds. By (3.4), z^{k+1} satisfies

$$0 \in \partial \tilde{G}^k(z^{k+1}) - w^k + B^k(z^{k+1} - z^k) \implies w^k \in \partial \tilde{G}^k(z^{k+1}) + B^k(z^{k+1} - z^k),$$

where $\tilde{G}^k(z) = g(z) + \gamma_k \phi(z) + \chi_C(z)$. By Lemma 4.2, $\lim_{k\to\infty} \|z^{k+1} - z^k\| = 0$. Thus, we have $w^k \in \partial(g + \gamma_k \phi + \chi_C)(z^k)$. Lemma 4.4 then implies $w^* \in \partial(g + \gamma_* \phi + \chi_C)(z^*)$. Hence, $\partial G^*(z^*) \cap \partial H^*(z^*) \neq \emptyset$, and therefore z^* is a critical point of (1.2).

Lemma 4.3 implies that there exists an index k_0 such that $\gamma_k = \gamma_{k_0}$ for all $k \geq k_0$, i.e., the sequence (γ_k) of penalty parameters is bounded. By design, the bottom case in the update rule (3.5) of Algorithm 1 must be holding for all $k \geq k_0$, i.e., either

$$\gamma_k \geq \delta_2 \|z^{k+1} - z^k\|^{-1} \quad \text{or} \quad \theta(z^{k+1}) \leq \varepsilon \quad \forall k \geq k_0.$$

Lemma 4.2 implies that $\|z^{k+1} - z^k\|^{-1} \to \infty$ as $k \to \infty$, hence $\gamma_k < \delta_2 \|z^{k+1} - z^k\|^{-1}$ holds for all sufficiently large k. Thus, we must have that $\lim_{k\to\infty} \theta(z^k) = \theta(z^*) \leq \varepsilon$. Hence, the cluster point z^* satisfies $\theta(z^*) \leq \varepsilon$. □

The above results establish convergence to a critical point of the penalized DC program (1.2) that is ε-feasible for the original nonconvex program (1.1) for any $\varepsilon > 0$. As noted at the beginning of the section, in the case that H is smooth, we have convergence to a d-stationary point, i.e., a point satisfying (2.4).

5. The Penalty Problem for MPCCs

We proceed to specialize the discussion of the proposed SPDCA penalty method to MPCCs of the form (1.3) with the associated penalty problem (1.5). In this section, we introduce a smooth MPCC penalty function and examine the correspondence between the MPCC-DC (1.3) and its penalty counterpart (1.5). We then propose a new penalty reformulation for general MPCCs in which the bilinear complementarity constraints $x_i y_i = 0$ are expressed as the difference of smooth convex quadratic functions. First, let us define the set $D := \{(x, y): x_i y_i = 0, i = 1, \ldots, n\}$ so that we may express the feasible set to (1.3) as $C \cap D$.

Furthermore, we make the following assumption.

Assumption 3. *The set C is nonempty, convex and compact, and there exists a point $(x, y) \in C$ such that $x_i y_i > 0$ for some $i = 1, \ldots, n$. Furthermore, the feasible region $C \cap D$ of problem (1.3) is nonempty.*

Consider the following bilinear ℓ_p-norm penalty functions for the constraint (1.3b)

$$\theta_p(x, y) = \|x \circ y\|_p, \tag{5.1}$$

where $\circ$ represents the Hadamard product. In particular, we focus on the ℓ_1–norm variant of (5.1), given by

$$\theta_1(x, y) = x^\top y,$$

on $C \cap D$. In the specialization to MPCC (1.3), the nonnegativity of x and y permits the omission of the absolute values in these definitions. Consequently, the penalty function θ_p is smooth in the ℓ_1-norm case. Clearly, when $\varphi_i(x, y) = 0$ for all $i = 1, \ldots, n$, we have $\theta_p(x, y) = 0$ and $F(x, y) = f(x, y)$ for all $1 \le p \le \infty$. It follows from the Lipschitz property of the ℓ_p-norm over a compact set that the function θ_p is Lipschitz on the feasible set $C \cap D$ (Horn and Johnson 1990). Moreover, $\theta_1(x, y)$ satisfies Assumption 2(iii) on the non-negative orthant. Considering the gradients $\nabla_x \theta_1(x, y) = y$ and $\nabla_y \theta_1(x, y) = x$, the only stationary point occurs at the origin, where the penalty value is exactly zero.

5.1. Equivalence of Stationary Points.

We establish the equivalence of strongly stationary (S-stationary) points of (1.3) and KKT points of (1.5) in the case of the ℓ_1-norm penalty. This correspondence is essential for claiming that a stationary point of the penalty problem (1.5) is also a stationary point to the MPCC (1.3). Note that there is an extensive literature on penalty methods for MPCCs, which explores the correspondence of global, local, and stationary solutions for MPCCs and penalty problems such as (1.5); see, e.g., Liu et al. (2008), Lucidi and Rinaldi (2010), Mangasarian and Pang (1997), Scholtes and Stöhr (1999), Schwartz (2011), Ye et al. (1997), and Zhi-Quan et al. (1996). As such, the equivalence results for MPCCs presented here are not novel, but rather situate these established findings within the proposed penalty framework and clarify their relevance.

Our goal is to establish the correspondence between the strong stationarity condition in Definition 2.3 applied to the MPCC-DC (1.3) and the KKT conditions (2.7) applied to the penalty problem (1.5) with the ℓ_1-norm penalty function. The following result formalizes the connection between the penalty problem and the MPCC-DC stationarity conditions.

Theorem 5.1. *Let $(\bar{x}, \bar{y}) \in C \cap D$ be a feasible point to the MPCC-DC (1.3). Then, $(\bar{x}, \bar{y})$ is a strongly stationary point to the MPCC-DC (1.3) if and only if it is a KKT point, i.e., stationary point, of the penalty problem (1.5).*

Proof. First, let $(\bar{x}, \bar{y}) \in C \cap D$ be an S-stationary point to MPCC-DC satisfying Definition 2.3. Define $\mu_i = \gamma$ for all $i \in \mathcal{I}$, where $\gamma > 0$ is the penalty parameter. Then, by the gradient definitions of θ_1, we obtain

$$\gamma \nabla_x \theta_1^{(i)}(\bar{x}, \bar{y}) = \mu_i \bar{y}_i = \mu_i \nabla_x \varphi_i(\bar{x}, \bar{y}), \quad \gamma \nabla_y \theta_1^{(i)}(\bar{x}, \bar{y}) = \mu_i \bar{x}_i = \mu_i \nabla_y \varphi_i(\bar{x}, \bar{y}).$$

Therefore, the stationary conditions (2.6a) and (2.6b) directly yield the stationary conditions (2.7a) and (2.7b). The conditions (2.6c) are analogous to (2.7c). Now, consider the multipliers ξ_i and ζ_i of the nonnegativity constraints. By (2.6d) and (2.6e), $\xi_i = 0$ holds if $\bar{x}_i > 0$ and $\zeta_i = 0$ holds if $\bar{y} > 0$. Therefore, conditions (2.7d) and (2.7e) are satisfied for all $i \in \mathcal{I}_x(\bar{x}, \bar{y}) \cup \mathcal{I}_y(\bar{x}, \bar{y})$. By the strong stationary condition $\xi_i, \zeta_i \geq 0$ for all $i \in \mathcal{I}(\bar{x}, \bar{y})$, we also have that (2.7d) and (2.7e) hold for all $i \in \mathcal{I}(\bar{x}, \bar{y})$.

For the other implication, let now $(\bar{x}, \bar{y}) \in C \cap D$ be a stationary point of the penalty problem (1.5). Take $\mu_i = \gamma$ for all $i \in \mathcal{I}$ as before. Then, using the gradients of the penalty function we have

$$\sum_{i=1}^{n} \mu_i \nabla_x \varphi_i(\bar{x}, \bar{y}) = \gamma \sum_{i=1}^{n} \nabla_x \theta_1^{(i)}(\bar{x}, \bar{y}) = \gamma \nabla_x \theta_1(\bar{x}, \bar{y}),$$

and analogously for the derivatives with respect to y. As a result, (2.7a) and (2.7b) directly imply the stationary conditions (2.6a) and (2.6b). As before, the conditions (2.7c) are analogous to (2.6c). The complementarity conditions for the nonnegativity constraints (2.7d) and (2.7e) give the complementarity conditions (2.6d) and (2.6e). Lastly, by dual feasibility, $\xi_i, \zeta_i \geq 0$ holds for all $i \in \mathcal{I}$, which gives the desired sign restriction for strong stationarity. □

As a result, the KKT points of the penalty problem (1.5) correspond to S-stationary points of the MPCC-DC. Therefore, Theorem 4.5 implies that Algorithm 1 converges to an S-stationary of the MPCC-DC (1.3) when applied to the penalty formulation (1.5).

6. DC Reformulation of the MPCC Penalty Problem

In this section, we present a DC reformulation of the MPCC-DC (1.3). To re-express the bilinear terms $x_i y_i$, $i = 1, \ldots, n$, we apply the following theorem from Gabriel et al. (2025), where indices are omitted for the ease of notation. Each of the complementarity penalty functions proposed in this section are smooth and have smooth DC decompositions.

Theorem 6.1. *Let $y, x \in \mathbb{R}$ be given and let $u, v \in \mathbb{R}$ be defined by*

$$\begin{pmatrix} y \\ x \end{pmatrix} = N \begin{pmatrix} u \\ v \end{pmatrix} \quad \text{with} \quad N = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

satisfying (i) $ad - bc \neq 0$, (ii) $ad + bc = 0$, and (iii) $ac \geq 0$. Then, N is invertible and there exist constants $\alpha, \beta > 0$ such that $yx = \alpha u^2 - \beta v^2$.

Let $\tilde{x} := (x, y)$ denote the vector of original variables in the optimization problem and let $\tilde{y} := (u, v)$ denote the vector of auxiliary variables. The penalized problem (1.5) is equivalent to

$$\begin{aligned} \min_{\tilde{x}, \tilde{y}} \quad & F_\gamma(\tilde{x}, \tilde{y}) := f(x, y) + \gamma \hat{\theta}_p(u, v) \\ \text{s.t.} \quad & (x, y) \in C, \quad \begin{pmatrix} y_i \\ x_i \end{pmatrix} = N_i \begin{pmatrix} u_i \\ v_i \end{pmatrix}, \quad i = 1, \ldots, n, \end{aligned}$$

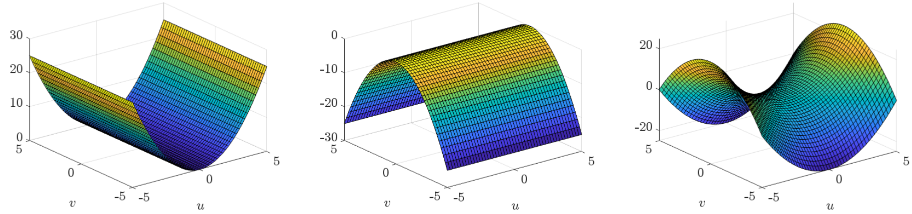


FIGURE 2. DC decomposition of $\hat{\theta}_1(u,v) = \alpha u^2 - \beta v^2$ with $\alpha = \beta = 1$. The DC function θ_1 (right) is decomposed into convex part $\hat{\phi}_1(u,v) = \alpha u^2$ (left) and concave part $-\hat{\psi}_1(u,v) = -\beta v^2$ (center).

where $\hat{\theta}_p$ is a penalty function of the form[1]

$$\hat{\theta}_p(u,v) = \left\| \left(\alpha \circ u^{\circ 2} - \beta \circ v^{\circ 2}\right) \right\|_p, \tag{6.1}$$

with DC decomposition $\hat{\theta}_p = \hat{\phi}_p - \hat{\psi}_p$ using convex functions $\hat{\phi}_p, \hat{\psi}_p : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. As before, the nonnegativity of x, y permits the omission of absolute values. Equivalence follows from the fact that $x_i y_i = \alpha_i u_i^2 - \beta_i v_i^2$ by Theorem 6.1. Since the composition of a convex function with a DC function is again a DC function (Horst and Thoai 1999, Theorem 2.3), the penalty term (6.1) is a DC function.

Next, we present a DC decomposition of the penalty term θ_p for $p = 1$. The DC decomposition of $\hat{\theta}_1(u,v)$ is straightforward:

$$\hat{\theta}_1(u,v) = \sum_{i=1}^{n} \alpha_i u_i^2 - \beta_i v_i^2. \tag{6.2}$$

The DC decomposition of the penalty function $\hat{\theta}_1 = \hat{\phi}_1 - \hat{\psi}_1$ is defined by

$$\hat{\phi}_1(u,v) = \sum_{i=1}^{n} \alpha_i u_i^2 \quad \text{and} \quad \hat{\psi}_1(u,v) = \sum_{i=1}^{n} \beta_i v_i^2,$$

where $\hat{\psi}_1$ and $\hat{\phi}_1$ are convex with respect to u, v, noting that $\alpha_i, \beta_i > 0$ for all $i = 1, \dots, n$. The DC decomposition of the penalty term $\hat{\theta}_1$ is visualized in Figure 2.

Applying a linear approximation to the nonconvex objective terms using standard DCA techniques results in a convex quadratic subproblem solved at iteration k. Given previous iterates $\tilde{x}^k \in \mathbb{R}^{2n}$ and $\tilde{y} \in \mathbb{R}^{2m}$, this convex subproblem is given by

$$\begin{aligned} \min_{\tilde{x},\tilde{y},w} \quad & \left[g(\tilde{x}) + \gamma \sum_{i=1}^{n} \alpha_i u_i^2 \right] - \left[\nabla_{\tilde{x}} h(\tilde{x}^k)^\top (\tilde{x} - \tilde{x}^k) + \gamma \sum_{i=1}^{n} 2\beta_i v_i^k (v_i - v_i^k) \right] \\ \text{s.t.} \quad & (x,y) \in C, \quad \begin{pmatrix} y_i \\ x_i \end{pmatrix} = N_i \begin{pmatrix} u_i \\ v_i \end{pmatrix}, \quad i = 1, \dots, n. \end{aligned} \tag{6.3}$$

Note that in this case, $\hat{\psi}_1$ is smooth, so $H = h + \gamma \hat{\psi}_1$ is differentiable as long as h is also smooth. Therefore, DC-criticality implies directional stationary, and Theorem 4.5 implies that Algorithm 1 applied to (6.3) converges to a d-stationary point of (6.3). Furthermore, since the added constraints define an invertible linear transformation between (x,y) and (u,v), i.e., it is a bijective linear transformation, the feasible set of (6.3) is linearly isomorphic to the original set C; therefore, the tangent and linearized cones are preserved, and ACQ continues to hold (Rockafellar

[1] We use $w^{\circ 2}$ to denote the Hadamard (element-wise) square of a vector $w \in \mathbb{R}^n$, i.e., $w^{\circ 2} = w \circ w$.

and Wets 2009, Chapter 6). Hence, for $p = 1$, a d-stationary point of (6.3) is also a strongly stationary point to the MPCC-DC (1.3).

7. Numerical Experiments

In this section we describe three classes of MPCCs to which we apply the proposed SPDCA. All experiments were conducted on a workstation equipped with an Apple Silicon M2 Max processor with 12 CPU cores running at 3.49 GHz and 64 GB RAM. The code used for all computational studies in this work is implemented in Python 3.11.8, with optimization problems formulated using Pyomo 6.9.1 (Hart et al. 2017). The SPDCA subproblems as well as all mixed-integer programs are solved with Gurobi 12.0.0. All code, data, and result files used in the numerical experiments are available in an open-source GitHub repository.[2]

7.1. Linear Bilevel Problems.

We consider the linear bilevel problem

$$\begin{aligned} \min_{x,y} \quad & c_x^\top x + c_y^\top y \\ \text{s.t.} \quad & Ax + By \geq a, \\ & y \in \arg\min_{\bar{y}} \left\{ d^\top \bar{y} \colon Cx + D\bar{y} \geq b \right\} \end{aligned}$$

in its optimistic version, where the leader in the upper level both optimizes over the upper-level variables x and over the lower-level variables y that are optimal for the follower's problem. Using the KKT conditions of the lower-level problem, we obtain the single-level reformulation

$$\min_{x,y,\lambda} \quad c_x^\top x + c_y^\top y \tag{7.1a}$$

$$\text{s.t.} \quad Ax + By \geq a, \ Cx + Dy \geq b, \tag{7.1b}$$

$$D^\top \lambda = d, \ \lambda \geq 0, \tag{7.1c}$$

$$\lambda_i (Cx + Dy - b)_i = 0, \quad i = 1, \dots, m_\ell, \tag{7.1d}$$

where the KKT complementarity conditions in (7.1d) are nonconvex and bilinear constraints. Hence, the number of complementarity constraints, which we refer to as the complementarity dimension, is the number m_ℓ of lower-level constraints. In the context of (1.3), the complementarity dimension m_ℓ takes the role of n. Introducing a slack variable $s_i = (Cx + Dy - b)_i$ for each $i = 1, \dots, m_\ell$, we can arrive at the following DC program with linear constraints

$$\min_{x,y,\lambda,s,u,v} \quad F(z) := c_x^\top x + c_y^\top y + \gamma \theta_p(u, v) \tag{7.2a}$$

$$\text{s.t.} \quad Ax + By \geq a, \ Cx + Dy \geq b, \tag{7.2b}$$

$$D^\top \lambda = d, \ \lambda \geq 0, \ s \geq 0, \tag{7.2c}$$

$$(Cx + Dy - b)_i - s_i = 0, \quad i = 1, \dots, m_\ell, \tag{7.2d}$$

$$\begin{pmatrix} \lambda_i \\ s_i \end{pmatrix} = N_i \begin{pmatrix} u_i \\ v_i \end{pmatrix}, \quad i = 1, \dots, m_\ell, \tag{7.2e}$$

where the penalty function $\theta_p(u, v)$ is given by (6.1).

Note that in our implementation, we substitute the variables λ_i and s_i via (7.2e) to reduce the total number of variables. Hence, our decision variables are x, y, u, and v, meaning that the introduction of auxiliary variables u and v do not significantly increase the problem size.

Table 1. Summary of test instances

Class	Reference	Total	Hard	Easy	$\lvert\mathcal{I}\rvert$	min m_ℓ	max m_ℓ
OR	Chu and Beasley (1998)	60	23	14	23	301	1504
IMKP	Khuri et al. (1995)	42	1	32	9	124	272
KP	Fischetti et al. (2018)	450	0	385	65	65	301
BCPINS	Tang et al. (2016)	38	5	3	30	44	65
PLUSBCPINS	Fischetti et al. (2018)	80	52	0	28	108	190

7.1.1. *BOBILib Test Instances and the Big-M Approach.* To analyze the performance of SPDCA when applied to linear bilevel problems, we use test instances from BOBILib (Thürauf et al. 2026). This test library contains mixed-integer bilevel optimization problems for which the integrality conditions can be relaxed to obtain linear bilevel problem instances as it is done as well in Kleinert et al. (2021a,b) and Kleinert and Schmidt (2021). For our comparison, we use the big-M mixed-integer reformulation of the complementarity constraints as a baseline (Fortuny-Amat and McCarl 1981). This reformulation modifies the KKT reformulation (7.2) as follows. Auxiliary binary variables $w \in \{0,1\}^{m_\ell}$ are introduced and (7.2e) is replaced with

$$\lambda_i \le M w_i, \;\; s_i \le M(1 - w_i), \quad i = 1, \dots, m_\ell,$$

and the penalty term $\gamma\theta_p(u, v)$ is removed from the objective (7.2a). Unless specified otherwise, we use $M = 10^4$. The reduced test set $\mathcal{I}$ has 155 instances, which are summarized in Table 1. We exclude 434 instances (referred to as “Easy”) that can be solved to global optimality using this big-M reformulation in less than one second. In addition, we exclude 81 instances (referred to as “Hard”) that cannot be solved in less than one hour. Although additional instance classes are available in BOBILib, they are omitted from our study because all instances in those classes fall entirely into either “Easy” or “Hard” categories.

Our experiments aim to show that SPDCA is an efficient local optimization scheme that finds high-quality solutions quickly. We note that the big-M reformulation solved with an MILP solver is a global solution method, while the proposed SPDCA method is a local one. Hence, when evaluating SPDCA, we analyze the quality of the solution in addition to the runtime of the algorithm. The quality of a solution is measured by the relative optimality gap is given by $|\omega - \underline{\omega}|/\underline{\omega}$, where $\underline{\omega}$ denotes the best lower bound for the optimal value computed by the big-M approach and ω is the objective determined by either SPDCA or the big-M approach. In this way, we compare both the big-M and SPDCA approaches to the same lower bound. For the discussion on linear bilevel problems, we define a “high-quality solution” to be one with a relative optimality gap less than 5 %. Ideally, SPDCA converges to a high-quality solution in a short amount of time. To test this claim, we first compare a number of SPDCA variants, and then take the best performing SPDCA method and compare its performance with the big-M approach.

7.1.2. *Internal DCA Comparison.* We also implement the ℓ_∞-norm DC penalty reformulation by Gabriel et al. (2025), which yields a DC program with DC constraints. This penalty method utilizes the penalty function

$$\hat{\theta}_\infty(u, v) = \left\| \alpha \circ u^{\circ 2} - \beta \circ v^{\circ 2} \right\|_\infty = \max_{i=1,\dots,n} \left\{ \alpha_i u_i^2 - \beta_i v_i^2 \right\}. \qquad (7.3)$$

[2] https://github.com/dominicflocco/spdca

Method	HQ sols.	Nr. Iterations		Runtime (sec.)		Rel. Opt. Gap (%)	
	(%)	Mean	Median	Mean	Median	Mean	Median
SPL-N-ℓ_1	**69.03**	**88.60**	**23.50**	1.13	0.13	9.75	**0.35**
SPL-N-ℓ_∞	40.00	281.40	20.00	24.78	0.32	12.55	1.18
SPL-R-ℓ_1	18.71	147.80	11.00	2.03	0.06	24.12	11.29
SPL-R-ℓ_∞	21.94	314.30	20.00	29.26	0.32	15.49	8.33
SPL-E-ℓ_1	54.19	**88.60**	**23.50**	**1.11**	**0.12**	17.04	1.47
SPL-E-ℓ_∞	49.68	304.50	30.00	26.01	0.47	**9.42**	0.70
big-M	92.26	–	–	0.30	0.07	3.59	0.01

TABLE 2. Performance of SPDCA methods and the big-M approach on linear bilevel problems. Percent of high-quality (HQ) solutions is the percentage of instances for which the method converged to a solution with a relative optimality gap less than or equal to 5 %. Reported runtimes are the time to the high-quality solution, and relative optimality gaps are best obtained within a 30 minutes time limit. Boldface items represent the best SPDCA method in each category.

We can reformulate the resulting min-max penalty objective using an auxiliary variable $\eta \geq 0$, leading to the DC program with DC constraints

$$\min_{\tilde{x},\tilde{y},\eta} \quad f(x,y) + \gamma\eta \tag{7.4a}$$

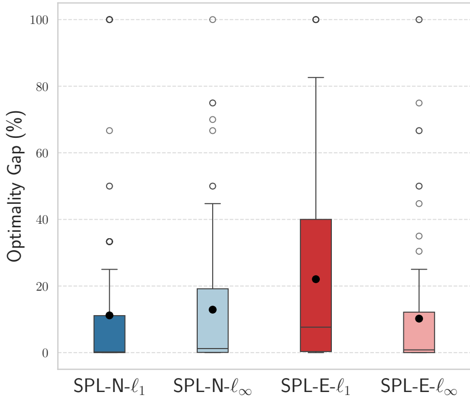
$$\tilde{x} \in C, \quad \begin{pmatrix} y_i \\ x_i \end{pmatrix} = N_i \begin{pmatrix} u_i \\ v_i \end{pmatrix}, \quad i = 1,\dots,n, \tag{7.4b}$$

$$\alpha_i u_i^2 - \beta_i v_i^2 \leq \eta, \quad i = 1,\dots,n, \tag{7.4c}$$

where (7.4c) are DC constraints. This problem may be solved using the DCA-BL method by Gabriel et al. (2025), for which we modify its implementation to incorporate the same penalty update rule and proximal term as SPDCA (Algorithm 1).

We test SPDCA with both the ℓ_1 and ℓ_∞-norm penalty function in (6.2) and (7.3), respectively, using varying starting points. As noted by Ben-Tal and Tetruashvili (2025), the choice of the starting point can have a significant impact on the performance of DCA methods. Therefore, we test each SPDCA scheme using three different starting points: a vector of all zeros "N" (for null vector), the vector of ones "E", and a relaxed solution "R", i.e., a solution to (7.1) with (7.1d) removed. We consider the vector of ones as a nonzero starting point that is offset from the origin, which is motivated by similar choices in the DC-complementarity literature (Jara-Moroni et al. 2018). This results in six variants: successive proximal linear DCA with ℓ_1-norm penalty (6.2) and each of the three starting points, referred to as SPL-N-ℓ_1, SPL-E-ℓ_1, and SPL-R-ℓ_1; and successive proximal linear DCA with ℓ_∞-norm penalty (7.3) and each of the three starting points, referred to as SPL-N-ℓ_∞, SPL-E-ℓ_∞, and SPL-R-ℓ_∞.

We use each variant to solve the test instances in $\mathcal{I}$ and compare both the relative optimality gap and the runtime to a high-quality solution. Each variant has the same adaptive proximal linear scheme. Specifically, the proximal term is governed by the positive definite matrix $B^k = \tau_k I$. We set $\tau_0 = L/2$, where $L = \gamma_0/2$ is the Lipschitz constant of the gradient of the objective function $F^0(z) = c_x^\top x + c_y^\top y + \gamma_0\theta(u,v)$ in the DC reformulation (7.2). Then, at each iteration k, the proximal term is updated according to $\tau_{k+1} = \max\{\alpha\tau_k, \tau_{\min}\}$, where $\alpha \in [0,1]$ and $\tau_{\min}$ are input parameters.



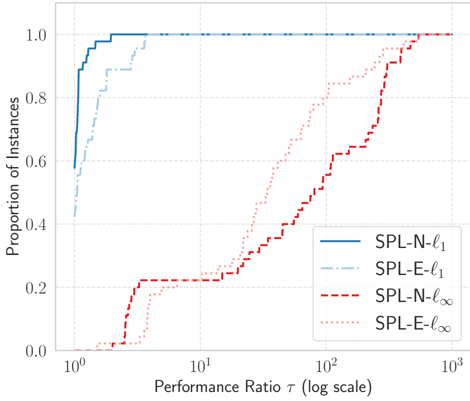


FIGURE 3. Comparison of SPDCA methods on linear bilevel optimization test instances. (Left) Box plots of optimality gap, where the black dot represents the mean and the horizontal line represents the median. (Right) Performance profiles of runtime.

For experimentation, we use $\alpha = 0.9$ and $\tau_{\min} = 10^{-6}$. For the penalty parameter updates, we use $\gamma_0 = 1$, $\delta_1 = 10$, and $\delta_2 = 1$, together with a feasibility tolerance of $\varepsilon = 10^{-6}$. The time limit for each DCA method is set to 30 minutes.

Table 2 shows the performance of each SPDCA method on the 155 linear bilevel test instances. First, we observe that SPL-N-ℓ_1 and SPL-E-ℓ_1 produce the largest number of high-quality solutions out of the SPDCA methods, returning a high-quality solution for 69.03 % and 54.19 % of the instances, respectively. Methods that use the starting point strategy "R", i.e., SPL-R-ℓ_1 and SPL-R-ℓ_∞ obtained a high-quality solution for only 18.71 % and 21.94 % of the instances. Due to the low success rates of methods using the starting point strategy "R", we focus our internal comparison on methods using the vector of zeros or ones as a starting point, i.e., SPL-N-ℓ_1, SPL-N-ℓ_∞, SPL-E-ℓ_1, and SPL-E-ℓ_∞.

Figure 3 (right) shows the performance of these four SPDCA methods on the 102 (65.8 %) of test instances for which all four methods obtained a feasible point within a tolerance ε, i.e., satisfying $\lambda_i s_i < \varepsilon$ for all $i = 1, \ldots, n$. As shown in the left plot, SPL-E-ℓ_∞ achieves the best mean optimality gap of 9.42 %. However, SPL-N-ℓ_1 and SPL-N-ℓ_∞ are not much worse, having mean optimality gaps of 9.75 %.[3] Moreover, SPL-N-ℓ_1 reports the best median optimality gap among these instances of 0.35 %, while SPL-N-ℓ_∞ and SPL-E-ℓ_∞ are slightly higher with medians of 1.18 % and 0.70 %. Overall, all four SPDCA methods attain a median relative optimality gap below 2%, highlighting their strong performance on the majority of instances. This suggests that these methods are able to produce high-quality stationary points in a rather short amount of time. Moreover, the discrepancy between the mean and median optimality gaps suggests that most instances are close to a global optimum, while a few low-quality solutions with large gaps strongly influence the mean, which is visualized in the box plot in Figure 3 (left).

Next, we compare the runtimes of the SPDCA methods using performance profiles, as introduced by Dolan and Moré (2002); see Figure 3 (right). [4] There is a clear

[3]Note that the mean and median optimality gaps shown in Figure 3 differ from those reported in Table 2, since the considered instances are different. For each method, Table 2 aggregates over all instances for which this method produced a feasible solution, while Figure 2 considers only instances for which SPL-N-ℓ_1, SPL-N-ℓ_∞, SPL-E-ℓ_1, and SPL-E-ℓ_∞ all produced a feasible solution.

[4]The performance profile provides a cumulative distribution function that plots, for each method, the percentage of instances for which the ratio $r_{i,s}$ is within a factor $\tau \geq 1$ of the best ratio. The

performance gap between ℓ_1- and ℓ_∞-norm penalty methods, shown by the blue and red profiles, respectively. This suggests that the ℓ_1-norm penalty function achieves the best performance as measured by total runtime. Referring back to Table 2, we observe this is due to more iterations required for the ℓ_∞-norm penalty methods. Specifically, SPL-N-ℓ_∞ obtains a high-quality solution in 281.4 iterations on average, compared to 88.6 for SPL-N-ℓ_1. Overall, the SPL-N-ℓ_1 method has the best balance between speed and solution quality on the linear bilevel test instances, obtaining a high-quality solution in 69.03 % of instances with a median runtime of 0.13 seconds and median optimality gap of 0.35 %.

7.1.3. *Benchmark Comparison.* Next, we compare the performance of SPDCA with the mixed-integer linear big-M formulation for the set of linear bilevel optimization test problems. Recall that the big-M formulation solved with Gurobi leads to a global solution approach, while we are only guaranteed to obtain stationary point using SPDCA. From the results in Table 2, the big-M approach demonstrates strong performance on the linear bilevel instances, particularly in terms of runtime to high-quality solutions and relative optimality gap within a fixed time limit. The SPL-N-ℓ_1 method achieves a high-quality solution in nearly 70 % of the instances, with mean and median runtime of 1.13 and 0.13 seconds, respectively. The big-M approach reaches a high-quality solution for 92.26 % of all instances with shorter mean and median runtimes of 0.30 and 0.07 seconds, respectively. However, when comparing relative optimality gaps for a fixed runtime (30 minutes), SPDCA performed comparably to big-M, with a mean and median relative gap of 9.75 % and 0.35 %, respectively, compared to 3.59 % and 0.01 % for big-M.

7.2. **Quadratic Bilevel Problems.** We now consider more general bilevel problems with a quadratic DC upper-level objective function and a convex-quadratic lower-level objective function:

$$\min_{x,y} \quad \frac{1}{2}x^\top Q_x^u x + \frac{1}{2}y^\top Q_y^u y + c_x^\top x + c_y^\top y \tag{7.5a}$$

$$\text{s.t.} \quad Ax + By \geq a, \tag{7.5b}$$

$$y \in \arg\min_{\bar{y}} \left\{ \frac{1}{2}\bar{y}^\top Q_y^\ell \bar{y} + d^\top \bar{y} \colon Cx + D\bar{y} \geq b \right\}. \tag{7.5c}$$

Here, $Q_x^u \in \mathbb{R}^{n_x \times n_x}$ and $Q_y^u \in \mathbb{R}^{n_y \times n_y}$ are symmetric, but not necessarily positive semi-definite matrices, and $Q_y^\ell \in \mathbb{R}^{n_y \times n_y}$ is a symmetric, positive semi-definite matrix. In this setting, the KKT conditions characterize globally optimal solutions to the lower-level problem (7.5c). The KKT reformulation is given by

$$\min_{x,y,\lambda} \quad \frac{1}{2}x^\top Q_x^u x + \frac{1}{2}y^\top Q_y^u y + c_x^\top x + c_y^\top y \tag{7.6a}$$

$$\text{s.t.} \quad Ax + By \geq a, \ Cx + Dy \geq b, \tag{7.6b}$$

$$Q_y^\ell y + d - D^\top \lambda = 0, \ \lambda \geq 0, \tag{7.6c}$$

$$\lambda_i(Cx + Dy - b)_i = 0, \quad i = 1, \dots, m_\ell, \tag{7.6d}$$

where $\lambda \in \mathbb{R}^{m_\ell}$ is the associated Lagrange multiplier vector of the linear lower-level constraints $Cx + Dy \geq b$.

To recast the upper-level objective (7.5a) as a DC function we use an eigenvalue perturbation approach. To this end, we define $\rho_x := -\lambda_{\min}(Q_x^u)$ and

ratio $r_{i,s}$ is defined by $r_{i,s} = t_{i,s}/\min\{t_{i,s} : s \in S\}$, where $t_{i,s}$ is the performance measure, e.g., time, of method $s \in S$ on instance $i \in \mathcal{I}$.

$\rho_y := -\lambda_{\min}(Q_y^u)$ to be the negative minimum eigenvalues of Q_x^u and Q_y^u, respectively. If Q_x^u and Q_y^u are indefinite, it holds that $\rho_x, \rho_y > 0$. We can then write the DC decomposition of (7.5a) as

$$\frac{1}{2}x^\top(Q_x^u + \rho_x I)x + \frac{1}{2}y^\top(Q_y^u + \rho_y I)y + c_x^\top x + c_y^\top y - \frac{1}{2}(\rho_x x^\top x + \rho_y y^\top y),$$

where $Q_x^u + \rho_x I \succeq 0$ and $Q_y^u + \rho_y I \succeq 0$ holds by construction.

7.2.1. *Numerical Results for Quadratic Bilevel Problems.* We now explore the numerical properties of the SPDCA algorithm when applied to quadratic bilevel instances of the form (7.5). Specifically, we take the 155 linear bilevel test instances from the BOBILib (Thürauf et al. 2026) described in Section 7.1.1 and generate random quadratic matrices Q_x^u, Q_y^u, and Q_y^ℓ using the code accompanying the paper by Kleinert and Schmidt (2021). For this test set, we also include the 81 instances that were classified as "Hard" in Table 1, which leads to complementarity dimensions up to $m_\ell = 1529$ for a total of 236 instances. As before, the time limit for each method is set to 30 minutes. To analyze the performance of SPDCA on a different classes of quadratic bilevel problems, we generate four variations of the Hessian matrices, leading to a total of 944 test instances. These variations determine the (non-)convexity of the upper-level objective and the curvature of the lower-level objective. In all four cases, the upper-level objective (7.5a) is quadratic, but may be convex or nonconvex. The four instances are denoted as follows.

(i) CQ-L: Convex quadratic upper-level objective ($Q_x^u, Q_y^u \succeq 0$) and linear lower-level objective ($Q_y^\ell = 0$).
(ii) CQ-CQ: Convex quadratic upper-level objective ($Q_x^u, Q_y^u \succeq 0$) and convex quadratic lower-level objective ($Q_y^\ell \succeq 0$).
(iii) NCQ-L: Nonconvex quadratic upper-level objective (Q_x^u, Q_y^u indefinite) and linear lower-level objective ($Q_y^\ell = 0$).
(iv) NCQ-CQ: Nonconvex quadratic upper-level objective (Q_x^u, Q_y^u indefinite) and convex quadratic lower-level objective ($Q_y^\ell \succeq 0$).

Note that we always ensure that the lower level-problem is convex so that the KKT conditions are necessary and sufficient. Hence, the reformulation (7.6) is valid for all cases. For each of the 236 test instances in these four test cases, we solve the resulting quadratic bilevel optimization problem with the best performing SPDCA method from Section 7.1 (SPL-N-ℓ_1) and Gurobi for solving the Big-M reformulation. Due to the difficulty of the problem instances, neither SPL-N-ℓ_1 nor the big-M approach achieved an optimality gap of 5 % for a majority of the instances. Hence, we simply compare the best objective values of both methods for these instances.

Table 3 reports the performance of SPL-N-ℓ_1 across the four quadratic problem classes. SPDCA obtains a feasible solution for over 80 % of the instances of all classes. In comparison, the big-M approach achieves feasibility for all CQ-L instances and 85.81 % for the CQ-CQ instances, but its performance drops substantially for problems with a nonconvex upper-level objective: only 68.18 % of NCQ-L and 53.38 % of NCQ-CQ instances are solved within the 30-minute time limit. These results provide evidence of SPDCA's better ability to obtain feasible solutions in nonconvex settings.

Next, we examine computational time to convergence, reported in Table 3 and visualized in Figure 4 (top row). For convex instances (CQ-L and CQ-CQ), the big-M approach achieves the fastest mean and median runtimes, though the two methods are rather comparable on CQ-CQ. For nonconvex upper-level problems, picture is different. For NCQ-L, the big-M approach has mean and median runtimes of 824.68 and 451.68 seconds, whereas SPDCA converges in 366.31 and 11.16 seconds. A similar trend appears for NCQ-CQ. The performance profiles show that SPDCA

UL Obj.	LL Obj.	Method	Feas. (%)	Rel. Gap (%) Mean	Rel. Gap (%) Median	Runtime (sec.) Mean	Runtime (sec.) Median	Nr. Iterations Mean	Nr. Iterations Median	Time per Iter. Mean	Time per Iter. Median
CQ	L	big-M	100.00	9.25	0.01	309.74	15.14	–	–	–	–
		SPL-N-ℓ_1	87.74	31.56	2.49	595.07	191.83	778	344	1.43	0.57
	CQ	big-M	85.81	15.55	3.33	821.45	55.12	–	–	–	–
		SPL-N-ℓ_1	81.94	20.75	7.67	873.61	87.72	1122	280	0.72	0.24
NCQ	L	big-M	68.18	24.90	0.06	824.68	451.68	–	–	–	–
		SPL-N-ℓ_1	89.61	57.77	50.15	366.31	11.16	1024	187	0.28	0.04
	CQ	big-M	53.38	13.68	0.01	636.98	95.19	–	–	–	–
		SPL-N-ℓ_1	91.22	58.53	56.04	468.78	1.17	2441	36	0.12	0.03

TABLE 3. Performance of SPL-N-ℓ_1 and the big-M approach on quadratic bilevel optimization test instances with varying upper-level (UL) and lower-level (LL) objective function properties. Percent of feasible solutions is the percentage of instances for which the method obtained a feasible solution. Reported runtimes and SPDCA iterations are summarized for all instances in which both methods obtained a feasible solution within the time limit of 30 minutes.

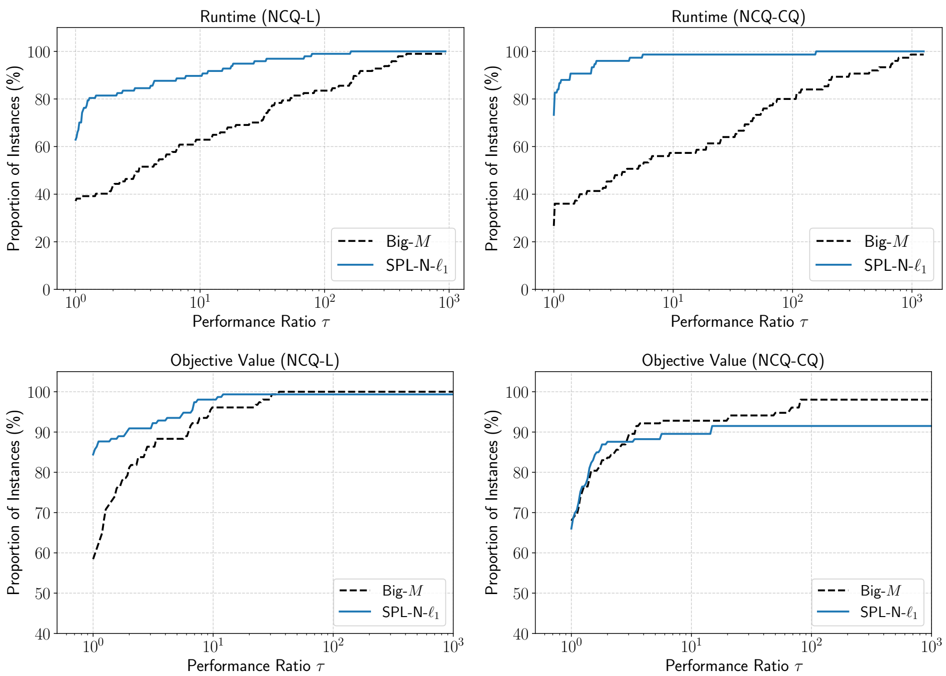


FIGURE 4. Performance plots of runtimes (top row) and objective values (bottom row) for SPL-N-ℓ_1 and the big-M approach on quadratic bilevel optimization test instances with nonconvex quadratic upper-level objective functions and linear lower-level objectives (left column) as well as convex quadratic lower-level objectives (right column).

achieves the fastest runtime in approximately 78 % of instances for both nonconvex classes (shown by the intercept at $\tau = 1$ in Figure 4), demonstrating its efficiency on challenging problems. Interestingly, SPDCA attains faster solve times on the nonconvex upper-level problems, a phenomenon that is not yet fully understood and is left for future investigation.

Moreover, we observe an interesting influence of the lower-level curvature on the SPDCA runtimes. In the KKT-based reformulation (7.6), a nonzero Hessian matrix in the lower-level objective introduces a linear term linking lower-level decisions y to lower-level dual variables λ. While each subproblem remains linear in both cases, instances with a convex quadratic lower-level lead to faster subproblem solves but require more total iterations overall. Consequently, total runtimes are smaller for linear lower-level instances (CQ-L, NCQ-L) than for convex-quadratic lower-level instances (CQ-CQ, NCQ-CQ).

When comparing solution quality between SPDCA and the big-M formulation, the relative optimality gaps reported in Table 3 indicate that big-M attains higher-quality solutions when given sufficient computational time. This outcome is expected, as big-M is a global optimization approach, whereas SPDCA is a local method with no global optimality guarantees. Nevertheless, SPDCA achieves competitive mean and median optimality gaps on the CQ-L and CQ-CQ instances. The disparity in solution quality becomes more pronounced for instances with a nonconvex quadratic upper-level objective, where the big-M approach is able to further improve solution quality over extended runtimes.

To further illustrate this trade-off between solution quality and the ability to compute feasible solutions, the bottom row of Figure 4 presents objective-value based performance profiles for both the big-M approach and SPDCA for a fixed time limit for problem classes with a nonconvex upper-level objective. Since performance profiles require a complete set of performance measures across all instances, we assign a dominated objective value for instances in which a solver failed to return a feasible solution, so that such failures are reflected as poor performance. SPDCA achieves the best objective performance in 84.42 % and 64.86 % of instances for the NCQ-L and NCQ-CQ problem classes, respectively.[5] These results indicate that, although SPDCA is a local method, it produces feasible solutions more consistently than the big-M approach on MPCC-DC instances with a nonconvex upper-level objective function. Furthermore, SPDCA frequently computes feasible solutions while maintaining significantly lower mean and median runtimes.

8. Conclusion

We develop a successive, proximal DC penalty method for solving general DC programs with DC constraints, termed SPDCA. The method uses a DC penalty function to measure constraint violation, yielding an equivalent DC penalty reformulation with convex constraints. This reformulation is solved via a proximal DCA with successive DC decomposition, driven by an adaptive penalty parameter update that balances the original objective with the penalization of equality constraint violations. We establish global convergence to a stationary point of the original DC program, extending existing results to settings involving successive DC decomposition. Notably, the analysis requires weaker constraint qualifications on the original problem than typically assumed in the literature; however, without MFCQ it remains open whether the penalty parameter γ_k stays bounded as $\varepsilon \to 0$, as in Lemma 4.3. Lastly, the method applies to any strongly convex proximal term, enabling both linear and

[5] Here, performance captures both the quality of objective value, as measured by relative optimality gap, and the ability of each method to compute feasible points. For instances for which one method converged to a feasible point, while the other did not, the method that computed a feasible point is considered the best performing. This is implemented via the aforementioned objective penalty.

quasi-Newton DCA variants. The generality of SPDCA makes it applicable to a broad class of DC optimization problems with DC equality constraints.

We further specialize SPDCA to MPCCs. Building on existing DCA approaches to complementarity problems, we incorporate a general ℓ_p-norm penalty function, leading to an approach capable of handling bilevel problems with a DC quadratic upper-level objective and convex quadratic lower-level objective. Central to this specialization is a DC penalty reformulation that expresses bilinear complementarity constraints as differences of smooth and convex quadratic functions. We also establish a correspondence between S-stationary points of the MPCC and critical points of the DC penalty reformulation, linking the DC and MPCC literature. Numerical experiments on MPCCs arising from bilevel optimization show that SPDCA reliably computes feasible solutions and achieves a good solution quality, while often requiring significantly less runtime than classic mixed-integer formulations, particularly on instances with quadratic upper-level DC objectives.

Acknowledgments

S. Gabriel was supported by the National Science Foundation Grant CMMI 2113891. S. Gabriel and T. Boomsma were supported by the Independent Research Fund Denmark, Project #0217-00009B. S. Gabriel and D. Flocco were supported by a grant from Petrobras #4324713. M. Lejeune acknowledges the support of the National Science Foundation through Grants DMS 2318519, RISE 2220626, and CMMI 2533372.

Data Availability Statement

The data used for the experiments are available in the following code repository: https://github.com/dominicflocco/spdca.

References

An, L. T. H. and P. D. Tao (2005). "The DC (Difference of Convex Functions) Programming and DCA Revisited with DC Models of Real World Nonconvex Optimization Problems." In: *Annals of Operations Research* 133.1, pp. 23–46. DOI: 10.1007/s10479-004-5022-1.

Bazaraa, M. S., H. D. Sherali, and C. M. Shetty (2006). *Nonlinear programming: theory and algorithms*. John Wiley & Sons.

Ben-Tal, A. and L. Tetruashvili (2025). *New Algorithms for maximizing the difference of convex functions*. URL: https://optimization-online.org/?p=29806.

Boyd, S. P. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.

Chu, P. C. and J. E. Beasley (1998). "A genetic algorithm for the multidimensional knapsack problem." In: *Journal of Heuristics* 4.1, pp. 63–86. DOI: 10.1023/A:1009642405419.

Constante-Flores, G., A. J. Conejo, and S. Constante-Flores (2022). "Solving certain complementarity problems in power markets via convex programming." In: *TOP* 30.3, pp. 465–491. DOI: 10.1007/s11750-022-00627-3.

Cottle, R. W., J.-S. Pang, and R. E. Stone (2009). *The Linear Complementarity Problem*. SIAM. DOI: 10.1137/1.9780898719000.

Dempe, S. (2002). *Foundations of Bilevel Programming*. Nonconvex Optimization and Its Applications. Springer New York, NY. DOI: 10.1007/b101970.

Dolan, E. D. and J. J. Moré (2002). "Benchmarking optimization software with performance profiles." In: *Mathematical Programming* 91.2, pp. 201–213. DOI: 10.1007/s101070100263.

Fischetti, M., M. Monaci, and M. Sinnl (2018). "A dynamic reformulation heuristic for generalized interdiction problems." In: *European Journal of Operational Research* 267.1, pp. 40–51. DOI: 10.1016/j.ejor.2017.11.043.

Flegel, M. L. and C. Kanzow (2005). "Abadie-type constraint qualification for mathematical programs with equilibrium constraints." In: *Journal of Optimization Theory and Applications* 124.3, pp. 595–614. DOI: 10.1007/s10957-004-1176-x.

Fortuny-Amat, J. and B. McCarl (1981). "A Representation and Economic Interpretation of a Two-Level Programming Problem." In: *The Journal of the Operational Research Society* 32.9, pp. 783–792. DOI: 10.2307/2581394.

Francisco, F. and J.-S. Pang (2003). *Finite-dimensional variational inequalities and complementarity problems*. New York, NY: Springer New York.

Gabriel, S. A., A. J. Conejo, J. D. Fuller, B. F. Hobbs, and C. Ruiz (2012). *Complementarity Modeling in Energy Markets*. Vol. 180. Springer Science & Business Media.

Gabriel, S. A., D. Flocco, T. K. Boomsma, M. Schmidt, and M. A. Lejeune (2025). "A Heuristic for Complementarity Problems Using Difference of Convex Functions." In: *INFORMS Journal on Computing*. DOI: 10.1287/ijoc.2024.0822.

Gabriel, S. A., R. García-Bertrand, P. Sahakij, and A. J. Conejo (2006). "A practical approach to approximate bilinear functions in mathematical programming problems by using Schur's decomposition and SOS type 2 variables." In: *Journal of the Operational Research Society* 57.8, pp. 995–1004. DOI: 10.1057/palgrave.jors.2602052.

Hanson, M. A. (1981). "On sufficiency of the Kuhn-Tucker conditions." In: *Journal of Mathematical Analysis and Applications* 80.2, pp. 545–550. DOI: 10.1016/0022-247X(81)90123-2.

Hart, W. E., C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola (2017). *Pyomo – Optimization Modeling in Python*. 2nd. Vol. 67. Springer Optimization and Its Applications. Springer. DOI: 10.1007/978-3-319-58821-6.

Hoai An, L. T., P. D. Tao, N. Nguyen Canh, and N. Van Thoai (2009). "DC programming techniques for solving a class of nonlinear bilevel programs." In: *Journal of Global Optimization* 44, pp. 313–337. DOI: 10.1007/s10898-008-9325-7.

Horn, R. A. and C. R. Johnson (1990). *Matrix Analysis*. Cambridge University Press.

Horst, R. and N. V. Thoai (1999). "DC programming: Overview." In: *Journal of Optimization Theory and Applications* 103.1, pp. 1–43. DOI: 10.1023/A:1021765131316.

Hu, X. M. and D. Ralph (2004). "Convergence of a Penalty Method for Mathematical Programming with Complementarity Constraints." In: *Journal of Optimization Theory and Applications* 123.2, pp. 365–390. DOI: 10.1007/s10957-004-5154-0.

Jara-Moroni, F., J.-S. Pang, and A. Wächter (2018). "A study of the difference-of-convex approach for solving linear programs with complementarity constraints." In: *Mathematical Programming* 169.1, pp. 221–254. DOI: 10.1007/s10107-017-1208-6.

Khuri, S., T. Baeck, and J. Heitkoetter (1995). *SAC94 Suite: Collection of Multiple Knapsack Problems*. Accessed: 2025-07-26. URL: http://www.cs.cmu.edu/Groups/AI/areas/genetic/ga/test/sac/0.html.

Kim, Y., S. Leyffer, and T. Munson (2020). "MPEC Methods for Bilevel Optimization Problems." In: *Bilevel Optimization: Advances and Next Challenges*. Ed. by S. Dempe and A. Zemkoho. Vol. 161. Springer Optimization and Its Applications.

Cham: Springer International Publishing, pp. 335–360. DOI: 10.1007/978-3-030-52119-6_12.
Kleinert, T., V. Grimm, and M. Schmidt (2021a). "Outer approximation for global optimization of mixed-integer quadratic bilevel problems." In: *Mathematical Programming.* Series B 188.1, pp. 489–524. DOI: 10.1007/s10107-020-01601-2.
Kleinert, T., M. Labbé, F. Plein, and M. Schmidt (2021b). "Closing the gap in linear bilevel optimization: a new valid primal-dual inequality." In: *Optimization Letters* 15.4, pp. 1027–1040. DOI: 10.1007/s11590-020-01660-6.
Kleinert, T. and M. Schmidt (2021). "Computing feasible points of bilevel problems with a penalty alternating direction method." In: *INFORMS Journal on Computing* 33.1, pp. 198–215. DOI: 10.1287/ijoc.2019.0945.
Le Thi, H. A. (2020). "DC programming and DCA for supply chain and production management: state-of-the-art models and methods." In: *International Journal of Production Research* 58.20, pp. 6078–6114. DOI: 10.1080/00207543.2019.1657245.
Le Thi, H. A., V. T. Ho, and D. T. Pham (2019). "A unified DC programming framework and efficient DCA based approaches for large scale batch reinforcement learning." In: *Journal of Global Optimization* 73.2, pp. 279–310. DOI: 10.1007/s10898-018-0698-y.
Le Thi, H. A., V. N. Huynh, and T. P. Dinh (2014). "DC programming and DCA for general DC programs." In: *Advanced Computational Methods for Knowledge Engineering: Proceedings of the 2nd International Conference on Computer Science, Applied Mathematics and Applications (ICCSAMA 2014).* Springer, pp. 15–35. DOI: 10.1007/978-3-319-06569-4_2.
Le Thi, H. A., T. M. T. Nguyen, and D. T. Pham (2023). "On solving difference of convex functions programs with linear complementarity constraints." In: *Computational Optimization and Applications* 84.3, pp. 833–867. DOI: 10.1007/s10589-023-00487-y.
Le Thi, H. A. and D. T. Pham (2001). "A continuous approach for globally solving linearly constrained quadratic programs." In: *Optimization* 50.1–2, pp. 93–120. DOI: 10.1080/02331930108844555.
– (2011). "On solving linear complementarity problems by DC programming and DCA." In: *Computational Optimization and Applications* 50.3, pp. 507–524. DOI: 10.1007/s10589-011-9398-y.
– (2023). "Open issues and recent advances in DC programming and DCA." In: *Journal of Global Optimization* 86.1, pp. 1–35. DOI: 10.1007/s10898-023-01272-1.
– (2018). "DC programming and DCA: thirty years of developments." In: *Mathematical Programming* 169.1, pp. 5–68. DOI: 10.1007/s10107-018-1235-y.
Le Thi, H. A., T. Pham Dinh, and H. V. Ngai (2012a). "Exact penalty and error bounds in DC programming." In: *Journal of Global Optimization* 52.3, pp. 509–535. DOI: 10.1007/s10898-011-9765-3.
Le Thi, H. A., D. Q. Tran, and D. T. Pham (2012b). "A DC programming approach for a class of bilevel programming problems and its application in Portfolio Selection." In: *Numerical Algebra, Control & Optimization* 2.1, pp. 167–185. DOI: 10.3934/naco.2012.2.167.
Liu, G., J. Ye, and J. Zhu (2008). "Partial Exact Penalty for Mathematical Programs with Equilibrium Constraints." In: *Set-Valued Analysis* 16.5, pp. 785–804. DOI: 10.1007/s11228-008-0095-7.
Liu, R., Z. Liu, W. Yao, S. Zeng, and J. Zhang (2024). "Moreau Envelope for Nonconvex Bi-Level Optimization: A Single-loop and Hessian-free Solution Strategy." In: *Proceedings of the 41st International Conference on Machine Learning.*

Vol. 235. Proceedings of Machine Learning Research. PMLR, pp. 31451–31478. URL: https://proceedings.mlr.press/v235/liu24ap.html.

Lucidi, S. and F. Rinaldi (2010). "Exact Penalty Functions for Nonlinear Integer Programming Problems." In: *Journal of Optimization Theory and Applications* 145.3, pp. 479–488. DOI: 10.1007/s10957-010-9700-7.

Mangasarian, O. L. and J.-S. Pang (1997). "Exact penalty for mathematical programs with linear complementarity constraints." In: *Optimization* 42.1, pp. 1–8. DOI: 10.1080/02331939708844356.

Oliveira, W. de (2020). "The ABC of DC Programming." In: *Set-Valued and Variational Analysis* 28.4, pp. 679–706. DOI: 10.1007/s11228-020-00566-w.

Pang, J.-S., M. Razaviyayn, and A. Alvarado (2017). "Computing B-stationary points of nonsmooth DC programs." In: *Mathematics of Operations Research* 42.1, pp. 95–118. DOI: 10.1287/moor.2016.0795.

Pham, D. T. and H. A. Le Thi (1997). "Convex analysis approach to D.C. programming: Theory, algorithms and applications." In: *Acta Mathematica Vietnamica* 22.1, pp. 289–355. URL: http://acta.math.ac.vn/index.php/AMV/article/view/234.

Pham, D. T., H. A. Le Thi, and F. Akoa (2008). "Combining DCA (DC Algorithms) and interior point techniques for large-scale nonconvex quadratic programming." In: *Optimization Methods and Software* 23.4, pp. 609–629. DOI: 10.1080/10556780802263990.

Rakotomamonjy, A., R. Flamary, and G. Gasso (2016). "DC Proximal Newton for Nonconvex Optimization Problems." In: *IEEE Transactions on Neural Networks and Learning Systems* 27.3, pp. 636–647. DOI: 10.1109/TNNLS.2015.2418224.

Ralph, D. and S. J. Wright (2004). "Some properties of regularization and penalization schemes for MPECs." In: *Optimization Methods and Software* 19.5, pp. 527–556. DOI: 10.1080/10556780410001691361.

Rockafellar, R. T. (1970). *Convex Analysis.* Princeton, NJ: Princeton University Press.

Rockafellar, R. T. and R. J.-B. Wets (2009). *Variational analysis.* Vol. 317. Springer Science & Business Media.

Scholtes, S. and M. Stöhr (1999). "Exact Penalization of Mathematical Programs with Equilibrium Constraints." In: *SIAM Journal on Control and Optimization* 37.2, pp. 617–652. DOI: 10.1137/S0363012996306121.

Schwartz, A. (2011). "Mathematical Programs with Complementarity Contraints: Theory, Methods, and Applications." PhD thesis. Julius-Maximilians-Universität Würzburg. URL: https://opus.bibliothek.uni-wuerzburg.de/frontdoor/index/index/docId/4977.

Siddiqui, S. and S. A. Gabriel (2013). "An SOS1-based approach for solving MPECs with a natural gas market application." In: *Networks and Spatial Economics* 13.2, pp. 205–227. DOI: 10.1007/s11067-012-9178-y.

Souza, J. C. O., P. R. Oliveira, and A. Soubeyran (2016). "Global convergence of a proximal linearized algorithm for difference of convex functions." In: *Optimization Letters* 10.7, pp. 1529–1539. DOI: 10.1007/s11590-015-0969-1.

Sun, W.-y., R. J. B. Sampaio, and M. A. B. Candido (2003). "Proximal point algorithm for minimization of DC function." In: *Journal of Computational Mathematics* 21.4, pp. 451–462. URL: https://www.jstor.org/stable/43693086.

Tang, Y., J.-P. P. Richard, and J. C. Smith (2016). "A class of algorithms for mixed-integer bilevel min–max optimization." In: *Journal of Global Optimization* 66.2, pp. 225–262. DOI: 10.1007/s10898-015-0274-7.

Thürauf, J., T. Kleinert, I. Ljubić, T. Ralphs, and M. Schmidt (2026). "BOBILib: Bilevel Optimization (Benchmark) Instance Library." In: *Mathematical Programming Computation*. URL: https://optimization-online.org/?p=27063.

Ye, J. J., D. L. Zhu, and Q. J. Zhu (1997). "Exact Penalization and Necessary Optimality Conditions for Generalized Bilevel Programming Problems." In: *SIAM Journal on Optimization* 7.2, pp. 481–507. DOI: 10.1137/S1052623493257344.

Zangwill, W. I. (1967). "Non-Linear Programming via Penalty Functions." In: *Management Science* 13.5, pp. 344–358. DOI: 10.1287/mnsc.13.5.344.

Zhi-Quan, L., J.-S. Pang, and D. Ralph (1996). *Mathematical programs with equilibrium constraints*. Cambridge University Press.

(D. Flocco) University of Maryland, Department of Mathematics, College Park, Maryland 20742, USA
Email address: dflocco@umd.edu

(S. A. Gabriel) (a) University of Maryland, College Park, Maryland 20742, USA, (b) Norwegian University of Science and Technology, Trondheim, Norway, (c) Aalto University, Espoo, Finland
Email address: sgabriel@umd.edu

(T. K. Boomsma) University of Copenhagen, Denmark
Email address: trine@math.ku.dk

(M. Schmidt) Trier University, Department of Mathematics, Universitätsring 15, 54296 Trier Germany
Email address: martin.schmidt@uni-trier.de

(M. A. Lejeune) George Washington University, Washington, DC, USA
Email address: mlejeune@gwu.edu