Comput Optim Appl (2017) 67:317–360
DOI 10.1007/s10589-017-9895-8

Solving nearly-separable quadratic optimization problems as nonsmooth equations

Frank E. Curtis[1] · **Arvind U. Raghunathan**[2]

Received: 23 March 2016 / Published online: 4 February 2017

Abstract An algorithm for solving nearly-separable quadratic optimization problems (QPs) is presented. The approach is based on applying a semismooth Newton method to solve the implicit complementarity problem arising as the first-order stationarity conditions of such a QP. An important feature of the approach is that, as in dual decomposition methods, separability of the dual function of the QP can be exploited in the search direction computation. Global convergence of the method is promoted by enforcing decrease in component(s) of a Fischer–Burmeister formulation of the complementarity conditions, either via a merit function or through a filter mechanism. The results of numerical experiments when solving convex and nonconvex instances are provided to illustrate the efficacy of the method.

Keywords Quadratic optimization problems · Dual decomposition · Complementarity problems · Semismooth Newton methods · Fischer–Burmeister function

Mathematics Subject Classification 49M05 · 49M15 · 49M27 · 49M29 · 65K05 · 65K10 · 90C20 · 90C33

Frank E. Curtis: This author was partially supported by the U.S. Department of Energy, Office of Science, Applied Mathematics, Early Career Research Program under Award Number DE-SC0010615.

✉ Frank E. Curtis
frank.e.curtis@gmail.com

Arvind U. Raghunathan
raghunathan@merl.com

[1] Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

[2] Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

1 Introduction

We consider the numerical solution of the quadratic optimization problem

$$\min_{\boldsymbol{x}} \sum_{i=1}^{N} \left(\frac{1}{2} x_i^T Q_i x_i + q_i^T x_i \right) \tag{1.1a}$$

$$\text{s.t.} \sum_{i=1}^{N} A_i x_i \leq b \tag{1.1b}$$

$$x_i \in \mathcal{X}_i \quad \text{for all} \quad i \in \{1, \ldots, N\}, \tag{1.1c}$$

where $\boldsymbol{x} := (x_1, \ldots, x_N) \in \mathbb{R}^{nN}$, $b \in \mathbb{R}^m$, and, for all $i \in \{1, \ldots, N\}$, $x_i \in \mathbb{R}^n$, $Q_i \in \mathbb{R}^{n \times n}$ is symmetric, $q_i \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{m \times n}$, and $\mathcal{X}_i \subseteq \mathbb{R}^n$ is nonempty, closed, and convex. The set $\mathcal{X}_i$ may include affine equality and inequality constraints, such as bounds or polyhedral constraints. The critical feature of (1.1) is that it is separable, except for the coupling constraint (1.1b). (These coupling constraints might also include affine equalities, but for ease of exposition we consider the formulation (1.1) in our algorithm development.) Such problems arise in a number of important applications including network utility maximization, nonlinear network flow analysis, and two-stage stochastic optimization; see Sect. 6 for further details.

Several algorithmic approaches have been proposed that are able to exploit the nearly-separable structure present in (1.1). Broadly speaking, these algorithms can be characterized as being part of either of two classes:

- *Linear Algebra Decomposition-Based Algorithms* This refers to algorithms in which the problem structure is exploited at the level of the linear algebra in the step computations. Implementations of such techniques require access to the internal routines of the optimization algorithm. A prime example of such an approach is the Schur complement decomposition employed in interior point algorithms for convex [21] and nonconvex [52] optimization. On the other hand, employing such an approach in an active-set algorithm can be less effective.
- *Problem Decomposition-Based Algorithms* This refers to algorithms in which the optimization problem is decomposed into smaller problems by relaxing the coupling constraint (1.1b), along with which coordination of the subproblem solutions is performed to promote convergence to the solution of the original problem. The dual decomposition approach [2] that dualizes the coupling constraints is such an example. An attractive feature of such approaches is that they do not require access to internal routines of the optimization algorithm(s) for solving the subproblems. Hence, they are not restricted by the type of algorithms (e.g., active-set or interior point) used to solve the subproblems. The algorithms in this class typically rely on convex duality and, hence, have been restricted in their applicability to convex problems.

Our work falls among the latter class of algorithms. Hence, it is worthwhile to review briefly the dual decomposition approach [2], as we do next.

1.1 Dual decomposition

The dual decomposition approach (see [2]) solves (1.1) by dualizing the coupling constraint (1.1b). Defining Lagrange multipliers $\lambda \in \mathbb{R}^m$ for these constraints, one obtains from (1.1) the dual function $g : \mathbb{R}^m \to \mathbb{R}$ defined by

$$g(\lambda) = \min_{x} \sum_{i=1}^{N} \left(\frac{1}{2} x_i^T Q_i x_i + q_i^T x_i \right) + \lambda^T \left(\sum_{i=1}^{N} A_i x_i - b \right) \quad \text{s.t. } x_i \in \mathcal{X}_i \ \text{ for all } \ i \in \{1, \ldots, N\}. \tag{1.2}$$

We denote a solution of (1.2) as $\boldsymbol{x}(\lambda) = (x_1(\lambda), \ldots, x_N(\lambda))$ to emphasize its dependence on λ. One of the attractive features of this dual function is that the optimization problem on the right-hand side of (1.2) is separable in the x_i's. That is, the solution $\boldsymbol{x}(\lambda)$ to (1.2) can be obtained by solving quadratic optimization subproblems to obtain, for each $i \in \{1, \ldots, N\}$,

$$x_i(\lambda) \in \arg\min_{x_i} \ \tfrac{1}{2} x_i^T Q_i x_i + (q_i + A_i^T \lambda)^T x_i \quad \text{s.t. } x_i \in \mathcal{X}_i. \tag{1.2$_i$}$$

Consequently, for any $\lambda \in \mathbb{R}^m$, the dual function value is given by

$$g(\lambda) = \sum_{i=1}^{N} \left(\frac{1}{2} x_i(\lambda)^T Q_i x_i(\lambda) + (q_i + A_i^T \lambda)^T x_i(\lambda) \right) - \lambda^T b. \tag{1.3}$$

If $Q_i \succeq 0$ for all $i \in \{1, \ldots, N\}$, then, by convex duality, the optimal solution to (1.1) can be obtained by maximizing the dual function subject to nonnegativity constraints on the multipliers, as in

$$\max_{\lambda \geq 0} \ g(\lambda). \tag{1.4}$$

The dual function g is concave [3]. In addition, it might be nonsmooth; e.g., this might occur due to non-uniqueness of the solution $\boldsymbol{x}(\lambda)$ for a given λ. Moreover, the subproblem solution $\boldsymbol{x}(\lambda)$ might be nonsmooth due to inequality constraints implicit in the $\mathcal{X}_i$'s. In all such cases, (1.4) is nonsmooth, in which case nonsmooth optimization methods such as subgradient and bundle methods can be employed to solve it [46]. However, these algorithms often require many iterations to converge to a solution and suffer from poor local convergence properties. The typical iteration complexity of these methods to obtain a point with an objective value within $\epsilon > 0$ of the minimal value is $\mathcal{O}(\frac{1}{\epsilon^2})$ [46].

A variety of other dual decomposition methods for solving smooth convex problems have also been proposed. A number of first-order algorithms have combined the smoothing technique pioneered by Nesterov [36] and the fast gradient method [34] to obtain $\mathcal{O}(\frac{1}{\epsilon})$ iteration complexity. Necoara and Suykens [31] use proximal-center-based smoothing and the fast gradient method. Dinh, Savorgnan, and Diehl [10] also use a proximal-center-based smoothing technique, but use the excessive gap technique

of Nesterov [35] to obtain an identical iteration complexity. In numerical experiments, the excessive gap technique of [10] outperformed the algorithm in [31]. Dinh, Necoara, and Savorgnan [9] extended this algorithm to allow for inexact solutions of the subproblems. Dinh, Necoara, and Diehl [8] employ a logarithmic barrier to smooth the dual function and employ a gradient-based path-following algorithm. Necoara and Patrascu [30] propose a first-order dual-decomposition method that allows for the presence of a general conic constraint; Lagrangian relaxation is used to relax this constraint, after which a first-order dual method is applied. Second-order algorithms have also been investigated in the context of dual decomposition. Necoara and Suykens [33] employ a logarithmic barrier to smooth the dual function and use a standard interior-point algorithm [37] for self-concordant functions. Frasch, Sager, and Diehl [20] consider strictly convex quadratic optimization problems with coupling equality constrains. They propose a semismooth Newton method for maximizing the dual function.

It is worthwhile to note that all of the papers cited above (except [30]) only consider coupling equality constraints and recommend handling coupling inequalities by introducing slack variables. Such a transformation turns a strictly convex problem into a (not strictly) convex one. Further, all of these approaches do not consider nonconvex problems and are all based on maximizing the dual function.

1.2 Our approach

In this paper, we are interested in the development of problem decomposition-based algorithms that possess superlinear local convergence properties. In contrast to existing approaches that employ dual maximization as in (1.4), we consider solving the *implicit complementarity problem* (ICP) given by

$$0 \leq F(\lambda) \perp \lambda \geq 0, \quad \text{where} \quad F(\lambda) = b - \sum_{i=1}^{N} A_i x_i(\lambda). \tag{1.5}$$

Essentially, we want to find a Lagrange multiplier vector $\lambda \geq 0$ and an associated $\boldsymbol{x}(\lambda)$ such that the dualized constraint (1.1b) is satisfied. It is easy to show that if such a solution exists, then it satisfies the first-order stationarity conditions of (1.1). More importantly, this consequence does not rely on *convex* duality, which is an important observation when (1.1) is not convex. Consequently, our approach bridges between the problem decomposition-based class and methods for solving nonconvex optimization problems, where in nonconvex settings our definition of $\boldsymbol{x}(\lambda)$ may vary depending on whether the elements $x_i(\lambda)$ satisfy first and/or second order necessary and/or sufficient conditions corresponding to (1.2_i).

An importance consequence of solving the ICP is that we readily obtain the primal solution to problem (1.1) as $\boldsymbol{x}(\lambda^*)$ where λ^* solves (1.5). By contrast, dual gradient and subgradient approaches only recover the optimal primal solution as an ergodic average of the corresponding primal iterate sequence. The convergence rate of such an average is typically sublinear, which can be exceeding slow in practice. We have observed this in our numerical experiments. The ICP formulation is advocated for

applications where recovery of the optimal primal solution is important, not just the optimal dual objective value.

Readers familiar with the literature on nonlinear complementarity problems (NCPs) will recognize that (1.5) has the same form as a typical NCP. However, the literature on NCP typically assumes that the function F is continuously differentiable (C^1) or continuously differentiable with Lipschitz continuous derivatives (LC^1). Such an assumption is not guaranteed for (1.5) in general; the differentiability of F depends on the differentiability of $x_i(\cdot)$. We rely on the theory of sensitivity analysis of parametric nonlinear optimization problems to derive differentiability properties of $x_i(\cdot)$. Under appropriate regularity assumptions at a solution of (1.1), the function $x_i(\cdot)$ can at least be shown to be piecewise continuous (PC^1). It is known that PC^1 functions are semismooth [25,41]. Further, semismooth functions are closed under the composition operation [41]. We focus on these properties to show that a semismooth Newton approach for solving (1.5) can achieve a fast local convergence rate [25]. This work builds on and generalizes the preliminary work of the second author and a collaborator in [1].

1.3 Notation

We use $\mathbb{N} := \{0, 1, 2, \dots\}$ to denote the set of nonnegative integers, $\mathbb{R}$ to denote the set of real numbers (scalars), and $\mathbb{R}^n$ to denote the set of real $n \times 1$ vectors. Given a vector $x \in \mathbb{R}^n$, its i-th component is denoted by $x_{[i]}$, and, given a set of indices $\alpha \subseteq \{1, \dots, n\}$, we write $x_{[\alpha]}$ to denote the subvector of x formed from the $x_{[i]}$'s with $i \in \alpha$. Similarly, given a matrix $M \in \mathbb{R}^{n\times m}$, its (i, j) entry is denoted by $M_{[ij]}$, its ith row is denoted by $M_{[i\cdot]}$, and the transpose of its ith row is denoted by $M^T_{[i\cdot]}$. Moreover, given sets of indices $\alpha \subseteq \{1, \dots, n\}$ and $\beta \subseteq \{1, \dots, m\}$, we write $M_{[\alpha\beta]}$ to refer to the submatrix formed from the $M_{[ij]}$'s with $i \in \alpha$ and $j \in \beta$. For a directionally differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$, we denote by $f'(x; d)$ the directional derivative of f at x along d. If such an f is C^1, then we denote by $\nabla f(x) \in \mathbb{R}^{m\times n}$ the Jacobian of the function f at the point x.

We assume throughout the remainder of the paper that, for all $i \in \{1, \dots, N\}$,

$$\mathcal{X}_i := \left\{x \in \mathbb{R}^n : B_i x \le c_i\right\} \quad \text{for some} \quad B_i \in \mathbb{R}^{p_i\times n} \quad \text{and} \quad c_i \in \mathbb{R}^{p_i}.$$

We also assume that each of these sets is nonempty. For ease of exposition, let

$$\boldsymbol{Q} := \begin{bmatrix} Q_1 & & \\ & \ddots & \\ & & Q_N \end{bmatrix}, \quad \boldsymbol{A} := \begin{bmatrix} A_1 & \cdots & A_N \end{bmatrix}, \quad \text{and} \quad \boldsymbol{B} := \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_N \end{bmatrix},$$

along with $\boldsymbol{q} := \begin{bmatrix} q_1^T & \cdots & q_N^T \end{bmatrix}^T$ and $\boldsymbol{c} := \begin{bmatrix} c_1^T & \cdots & c_N^T \end{bmatrix}^T$.

1.4 Organization

The rest of the paper is organized as follows. To put into context opportunities for solving (1.1) by solving (1.5), Sect. 2 overviews relationships between solutions of these problems, then discusses superlinear local convergence opportunities when solving (1.5). A complete algorithm for solving strictly convex instances of (1.5) is presented in Sect. 3 (see Algorithm 1 on page 14), and an algorithm for solving convex or even indefinite instances is presented in Sect. 4 (see Algorithm 2 on page 16). An implementation and numerical results on some randomly generated problems are discussed in Sect. 5 while a discussion of important applications and corresponding numerical experiments are given in Sect. 6. Concluding remarks are provided in Sect. 7.

2 Solving QPs and ICPs as semismooth equations

Our proposed algorithms for solving problem (1.1) are based on the idea that, under certain assumptions, a solution can be obtained by solving (1.5). Understanding relationships between solutions of these problems requires knowledge of stationarity conditions, differentiability concepts, regularity properties of complementarity constraints, and parametric sensitivity analysis. For a summary of this background, see Appendix 1. In short, the conclusion of this investigation reveals that solving (1.5) is only guaranteed to yield a second-order KKT point of (1.1) when the corresponding subproblem solutions are second-order KKT points and, with respect to a critical cone defined by the subproblem solutions, the matrix $\boldsymbol{Q}$ is positive semidefinite. This is all guaranteed to hold when $\boldsymbol{Q} \succeq 0$, but might not all hold if $Q_i \not\succeq 0$ for some $i \in \{1, \ldots, N\}$. Overall, in nonconvex settings, solving (1.5) does not necessarily yield a second-order KKT point of (1.1), but it does in certain cases of interest even when $\boldsymbol{Q} \not\succeq 0$.

Let us now proceed with our preliminary ideas for solving (1.5), which in particular are based on solving reformulations using two semismooth operators, namely the Minimum and Fischer operators. With $\min\{\cdot, \cdot\}$ defined componentwise, the Minimum and Fischer operators in our context are given respectively by

$$\Phi^{\min}(\lambda) := \begin{bmatrix} \min\{\lambda_{[1]}, F_{[1]}(\lambda)\} \\ \vdots \\ \min\{\lambda_{[m]}, F_{[m]}(\lambda)\} \end{bmatrix} \quad \text{and} \quad \Phi^{\mathrm{FB}}(\lambda) = \begin{bmatrix} \phi(\lambda_{[1]}, F_{[1]}(\lambda)) \\ \vdots \\ \phi(\lambda_{[m]}, F_{[m]}(\lambda)) \end{bmatrix}, \tag{2.1}$$

where, given scalars a and b, the Fischer–Burmeister function [15] has the form

$$\phi(a, b) = \sqrt{a^2 + b^2} - a - b. \tag{2.2}$$

It is easily verified that this latter function satisfies

$$\phi(a, b) = 0 \iff \{a \geq 0,\ b \geq 0,\ \text{and } ab = 0\}. \tag{2.3}$$

The articles [11,29] discuss regularity properties and sophisticated implementations of semismooth Newton algorithms for complementarity problems using the Fischer–Burmeister function. However, our formulation here is different in the sense that, in our context, the complementarity components λ and $F(\lambda)$ are both functions of λ; hence, our formulation is somewhat more straightforward.

Using $\Phi \in \{\Phi^{\min}, \Phi^{\mathrm{FB}}\}$, the conditions in (1.5) can be generically posed as

$$\Phi(\lambda) = 0. \tag{2.4}$$

If $\Phi \in PC^1$, then Φ is semismooth [25]. Further, if Φ and Ψ are semismooth, then the composite $\Phi \circ \Psi$ is also semismooth by closedness of semismooth functions under the composition operation [41]. This is a crucial property for our purposes since, under suitable conditions, we can show that $x_i(\cdot) \in PC^1$; see Lemma 7.2.

Motivated by these facts, a *semismooth Newton* iteration for solving (2.4) is

$$\lambda^{k+1} \leftarrow \lambda^k - (H^k)^{-1}\Phi(\lambda^k) \ \text{ for } \ k \in \mathbb{N}, \tag{2.5}$$

where $H^k \in \partial_B \Phi(\lambda^k)$, i.e., the *B-subdifferential* of Φ at λ^k; see (7.11). In order for this iteration to be well-defined in the neighborhood of λ^* satisfying $\Phi(\lambda^*) = 0$, nonsingularity of H^k is required at λ^k near λ^*. Qi [40] formalized a condition under which this occurs by introducing the notion of *BD-regularity*, as we do now.

Definition 2.1 (BD-regularity) A solution $\lambda^* \in \mathbb{R}^m$ of (2.4) is *BD-regular* if all elements of $\partial_B \Phi(\lambda^*)$ are nonsingular.

Under BD-regularity, (2.5) converges locally superlinearly; see [40].

Theorem 2.1 *Suppose $\Phi(\cdot) : \mathbb{R}^m \to \mathbb{R}^m$ is semismooth and $\lambda^* \in \mathbb{R}^m$ is a BD-regular solution of (2.4). Then, for any λ^0 sufficiently close to λ^*, the iteration (2.5) is well-defined and generates a sequence $\{\lambda^k\}$ which converges to λ^* Q-superlinearly.*

BD-regularity for the Minimum and Fischer operators have been characterized in various situations when $F \in C^1$. The following is a result related to such a characterization; see [47, Props. 2.8 and 2.9]. The result uses notions of b- and R-regularity; see Definition 7.1.

Lemma 2.1 *Suppose $F \in C^1$ and λ^* solves (1.5). Then,*

1. *λ^* is a BD-regular solution of $\Phi^{\min}(\lambda) = 0$ if λ^* is a b-regular solution of (1.5).*
2. *λ^* is a BD-regular solution of $\Phi^{\mathrm{FB}}(\lambda) = 0$ if λ^* is an R-regular solution of (1.5).*

We now establish conditions that may hold at a solution to (1.1) that ensure that BD-regularity holds at a solution to (1.5). We begin by proving, in the following lemma, nonsingularity of some matrices which will be essential in showing b-regularity and R-regularity of a solution to (1.5). Here, and throughout the remainder of this section, we refer to certain constraint index sets at solutions of (1.1) and (1.2_i); see (α, β, γ) for (1.1) defined in (7.3) and $(\alpha_i, \beta_i, \gamma_i)$ defined for (1.2_i) defined in (7.4). We also refer to constraint qualifications and second-order sufficiency conditions for these problems, which we define in Appendix 1.

Lemma 2.2 *Suppose that x^* is a second-order KKT point of (1.1) in that there exist multipliers (λ^*, ξ^*) such that (7.1) and (7.7) hold. Moreover, suppose the following:*

- *$\beta_i(x_i^*, \xi_i^*) = \emptyset$ for all $i \in \{1, \ldots, N\}$,*
- *the (1.1)-LICQ holds at (x^*, λ^*, ξ^*) in that*

$$\begin{bmatrix} A_{1[\alpha\cup\beta]} & \cdots & A_{N[\alpha\cup\beta]} \\ B_{1[\alpha_1\cdot]} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_{N[\alpha_N\cdot]} \end{bmatrix}$$

has full row rank, and

- *the (1.2_i)-SSOSC holds at (x_i^*, ξ_i^*) for all $i \in \{1, \ldots, N\}$.*

Then, defining

$$\begin{aligned} \boldsymbol{B}_{\overline{\alpha}} &:= \begin{bmatrix} B_{1[\alpha_1\cdot]} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_{N[\alpha_N\cdot]} \end{bmatrix} \text{ and} \\ \boldsymbol{A}_{\sigma} &:= \begin{bmatrix} A_{1[\sigma\cdot]} \cdots A_{N[\sigma\cdot]} \end{bmatrix} \text{ for all } \alpha(x^*, \lambda^*) \subseteq \sigma \subseteq \alpha(x^*, \lambda^*) \cup \beta(x^*, \lambda^*), \end{aligned} \tag{2.6}$$

the following matrices are nonsingular:

(i) $\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T & \boldsymbol{A}_{\sigma}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 & 0 \\ \boldsymbol{A}_{\sigma} & 0 & 0 \end{bmatrix}$ *for all* $\alpha(x^*, \lambda^*) \subseteq \sigma \subseteq \alpha(x^*, \lambda^*) \cup \beta(x^*, \lambda^*)$,

(ii) $\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}$, *and*

(iii) $\begin{bmatrix} \boldsymbol{A}_{\sigma} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_{\sigma}^T \\ 0 \end{bmatrix}$ *for all* $\alpha(x^*, \lambda^*) \subseteq \sigma \subseteq \alpha(x^*, \lambda^*) \cup \beta(x^*, \lambda^*)$.

Proof For the sake of brevity, we suppress the dependence of α and β on (x^*, λ^*) in the following. Given σ satisfying $\alpha \subseteq \sigma \subseteq \alpha \cup \beta$, suppose that the matrix in (i) is singular. Then, there exists a vector $(u, v, w) \neq 0$ such that

$$\begin{aligned} \boldsymbol{Q}u + \boldsymbol{B}_{\overline{\alpha}}^T v + \boldsymbol{A}_{\sigma}^T w &= 0 \\ \boldsymbol{B}_{\overline{\alpha}} u &= 0 \\ \boldsymbol{A}_{\sigma} u &= 0. \end{aligned} \tag{2.7}$$

Left-multiplying the first equation by u^T, we have $u^T(\boldsymbol{Q}u + \boldsymbol{B}_{\overline{\alpha}}^T v + \boldsymbol{A}_{\sigma}^T w) = 0$ which, from the second and third equations, implies that $u^T \boldsymbol{Q} u = 0$. However, from our assumption that the (1.2_i)-SSOSC holds for all $i \in \{1, \ldots, N\}$, we have that $u^T \boldsymbol{Q} u > 0$ for all u such that $\boldsymbol{B}_{\overline{\alpha}} u = 0$. Hence, along with the second equation in (2.7), we may conclude that $u = 0$. Substituting $u = 0$ into the first equation of (2.7), we obtain that $\boldsymbol{B}_{\overline{\alpha}}^T v + \boldsymbol{A}_{\sigma}^T w = 0$. From our assumption that the LICQ for (1.1) holds, we have that $(\boldsymbol{B}_{\overline{\alpha}}^T \ \boldsymbol{A}_{\sigma}^T)$ has full column rank, from which we may conclude that

$(v, w) = 0$. Overall, we have shown that $(u, v, w) = 0$, which contradicts the fact that this was defined to be a nonzero vector. Hence, it follows that the matrix in (i) is nonsingular. The proof for the matrix in (ii) is similar.

Now, given σ satisfying $\alpha \subseteq \sigma \subseteq \alpha \cup \beta$, suppose that the matrix in statement (iii) is singular. If this were true, then there would exist $w \neq 0$ such that

$$\begin{bmatrix} \boldsymbol{A}_\sigma & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0, \quad \text{where} \quad \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_\sigma^T \\ 0 \end{bmatrix} w.$$

However, this means that we have constructed $(u, v, w) \neq 0$ satisfying (2.7). This is a contradiction to the nonsingularity of the matrix in (i); hence, we may conclude that the matrix in (iii) is nonsingular. □

We now establish an expression for a submatrix of the Jacobian of the function F under similar conditions.

Lemma 2.3 *Suppose that $\boldsymbol{x}^*$ is a second-order KKT point of (1.1) in that there exist multipliers $(\lambda^*, \boldsymbol{\xi}^*)$ such that (7.1) and (7.7) hold. Moreover, suppose that the assumptions of Lemma 2.2 hold. Then,*

$$\nabla F_{[\sigma\sigma]}(\lambda^*) = \begin{bmatrix} \boldsymbol{A}_\sigma & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_\sigma^T \\ 0 \end{bmatrix} \tag{2.8}$$
$$\text{for all } \alpha(\boldsymbol{x}^*, \lambda^*) \subseteq \sigma \subseteq \alpha(\boldsymbol{x}^*, \lambda^*) \cup \beta(\boldsymbol{x}^*, \lambda^*).$$

Proof Observe that the conditions of Lemma 2.2 imply that, for each $i \in \{1, \ldots, N\}$, the conditions of Lemma 7.1 hold. Thus, from Lemma 7.1, we have $x_i(\cdot) \in C^1$ as a function of λ in W_i and that $\nabla x_i(\lambda)$ is given by (7.15). Utilizing the separability of $\nabla \boldsymbol{x}_{[\sigma\sigma]}(\lambda)$ into the components $\nabla x_i(\lambda)$ for all $i \in \{1, \ldots, N\}$ and rearranging, one can verify (with $\boldsymbol{I}$ representing an identity matrix) that

$$\nabla \boldsymbol{x}_{[\sigma\sigma]}(\lambda) = \begin{bmatrix} \boldsymbol{I} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} -\boldsymbol{A}_\sigma^T \\ 0 \end{bmatrix}.$$

The fact that $\nabla F_{[\sigma\sigma]}(\lambda) = -\boldsymbol{A}_\sigma \nabla \boldsymbol{x}_{[\sigma\sigma]}(\lambda)$ then yields the desired result. □

We now state our main result on the BD-regularity of the operators of interest.

Theorem 2.2 *Suppose that $\boldsymbol{x}^*$ is a second-order KKT point of (1.1) in that there exist multipliers $(\lambda^*, \boldsymbol{\xi}^*)$ such that (7.1) and (7.7) hold. Moreover, suppose that the assumptions of Lemma 2.2 hold. Then:*

(i) λ^* is a BD-regular solution of $\Phi^{\min}(\lambda) = 0$ and the iteration (2.5) applied with $\Phi = \Phi^{\min}$ converges Q-superlinearly.
(ii) λ^* is a BD-regular solution of $\Phi^{\mathrm{FB}}(\lambda) = 0$ and the iteration (2.5) applied with $\Phi = \Phi^{\mathrm{FB}}$ converges Q-superlinearly.

Proof From Lemma 2.3, we have that for all σ satisfying $\alpha \subseteq \sigma \subseteq \alpha \cup \beta$, the Jacobian $\nabla F_{[\sigma\sigma]}$ is given by the matrix in Lemma 2.2(iii). Thus, by Lemmas 2.2 and 2.3, $\nabla F_{[\sigma\sigma]}$ is nonsingular, which establishes that λ^* is a b-regular solution of (1.5). Hence, by Lemma 2.1, λ^* is a BD-regular solution of $\Phi^{\min}(\lambda) = 0$, from which Theorem 2.1 establishes the convergence rate of (2.5).

We have from Lemmas 2.3 and 2.2(iii) that $\nabla F_{[\alpha\alpha]}$ is nonsingular. We will show in the following that $\nabla F_{[\beta\beta]}/\nabla F_{[\alpha\alpha]}(\lambda^*)$ is positive definite, which implies that it is a P-matrix since it is symmetric. This establishes that λ^* is an R-regular solution of (1.5). Hence, by Lemma 2.1, λ^* is a BD-regular solution of $\Phi^{\mathrm{FB}}(\lambda) = 0$, from which Theorem 2.1 establishes the convergence rate of (2.5).

To prove positive definiteness of $\nabla F_{[\beta\beta]}/\nabla F_{[\alpha\alpha]}(\lambda^*)$, observe that

$$\begin{aligned}
& 0 < w_\beta^T \left(\nabla F_{[\beta\beta]}/\nabla F_{[\alpha\alpha]}(\lambda^*)\right) w_\beta \ \text{ for all } \ w_\beta \in \mathbb{R}^{|\beta|} \setminus \{0\} \\
\iff\ & 0 < w_\beta^T \left(\nabla F_{[\beta\beta]}(\lambda^*) w_\beta + \nabla F_{[\beta\alpha]}(\lambda^*) w_\alpha\right) \\
& \quad \text{for all } \ w_\beta \in \mathbb{R}^{|\beta|} \setminus \{0\},\ w_\alpha \in \mathbb{R}^{|\alpha|} \ \text{ such that} \\
& \quad \nabla F_{[\alpha\alpha]}(\lambda^*) w_\alpha = -\nabla F_{[\alpha\beta]}(\lambda^*) w_\beta .
\end{aligned}$$

Using $\nabla F_{[\sigma\sigma]}(\lambda^*)$ from Lemma 2.3, this latter property holds if and only if

$$\begin{aligned}
& 0 < w_\beta^T \begin{bmatrix} \boldsymbol{A}_\beta & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_\alpha^T w_\alpha & \boldsymbol{A}_\beta^T w_\beta \\ 0 & 0 \end{bmatrix} \\
& \quad \equiv \begin{cases} \text{for all } \ w_\beta \in \mathbb{R}^{|\beta|} \setminus \{0\},\ w_\alpha \in \mathbb{R}^{|\alpha|} \ \text{ such that} \\ \begin{bmatrix} \boldsymbol{A}_\alpha & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_\alpha^T \\ 0 \end{bmatrix} w_\alpha = -\begin{bmatrix} \boldsymbol{A}_\alpha & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_\beta^T \\ 0 \end{bmatrix} w_\beta \end{cases} \\
& \quad \equiv \begin{cases} \text{for all } \ w_\beta \in \mathbb{R}^{|\beta|} \setminus \{0\},\ w_\alpha \in \mathbb{R}^{|\alpha|} \ \text{ such that} \\ \begin{bmatrix} \boldsymbol{A}_\alpha & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{A}_\alpha^T & \boldsymbol{A}_\beta^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_\alpha \\ w_\beta \end{bmatrix} = 0 \end{cases} \\
& \quad \equiv \begin{cases} \text{for all } \ w_\beta \in \mathbb{R}^{|\beta|} \setminus \{0\},\ w_\alpha \in \mathbb{R}^{|\alpha|},\ (u, v) \in \mathbb{R}^{N+|\overline{\alpha}|} \text{such that} \\ \begin{bmatrix} \boldsymbol{A}_\alpha & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0 \ \text{ where } \ \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \boldsymbol{A}_\alpha^T & \boldsymbol{A}_\beta^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} w_\alpha \\ w_\beta \end{bmatrix} \end{cases} \\
\iff\ & 0 < w_\beta^T \begin{bmatrix} \boldsymbol{A}_\beta & 0 \end{bmatrix} \begin{bmatrix} -u \\ -v \end{bmatrix} \\
& \quad \begin{cases} \text{for all } \ w_\beta \in \mathbb{R}^{|\beta|},\ (u, v, w_\alpha) \in \mathbb{R}^{N+|\overline{\alpha}|+|\alpha|} \ \text{ such that} \\ \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}_{\overline{\alpha}}^T & \boldsymbol{A}_\alpha^T \\ \boldsymbol{B}_{\overline{\alpha}} & 0 & 0 \\ \boldsymbol{A}_\alpha & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w_\alpha \end{bmatrix} = \begin{bmatrix} -\boldsymbol{A}_\beta^T \\ 0 \\ 0 \end{bmatrix} w_\beta \end{cases} \\
\iff\ & 0 < -w_\beta^T \boldsymbol{A}_\beta u = u^T \boldsymbol{Q} u \ \text{ for all } \ u \in \mathbb{R}^N \ \text{ with } \ \boldsymbol{B}_{\overline{\alpha}} u = 0 \ \text{ and } \ \boldsymbol{A}_\alpha u = 0.
\end{aligned}$$

Here, the last equality follows from the required properties of (u, v, w_α), namely, multiplying the first equation

$$\boldsymbol{Q}u + \boldsymbol{B}_{\overline{\alpha}}^T v + \boldsymbol{A}_\alpha^T w_\alpha = -\boldsymbol{A}_\beta^T w_\beta \tag{2.9}$$

by u^T and using that u lies in the null spaces of $\boldsymbol{B}_{\overline{\alpha}}$ and $\boldsymbol{A}_\alpha$, which follows from the second and third equations in the properties of (u, v, w_α). Since the (1.2_i)-SSOSC holds at $\lambda = \lambda^*$, we have that $u^T \boldsymbol{Q} u > 0$ for all such $u \neq 0$. To show that $u \neq 0$ whenever $w_\beta \neq 0$, suppose that the contrary $u = 0$ for some $w_\beta \neq 0$ holds. Then (2.9) reduces to $\boldsymbol{B}_{\overline{\alpha}}^T v + \boldsymbol{A}_\alpha^T w_\alpha = -\boldsymbol{A}_\beta^T w_\beta$ for $(v, w_\alpha, w_\beta) \neq 0$ contradicting the assumption of (1.1)-LICQ. Hence, $u \neq 0$ whenever $w_\beta \neq 0$. This proves the positive definiteness of $\nabla F_{[\beta\beta]} / \nabla F_{[\alpha\alpha]}(\lambda^*)$. □

The key assumptions in Theorem 2.2 are: (1.1)-LICQ, $\beta_i = \emptyset$, and (1.2_i)-SSOSC. In other words, the result requires that the subproblems satisfy strict complementarity and strong second order sufficient conditions, but the coupling constraints are allowed to have non-strict complementarity components. From the above results, it is clear that by employing either the Minimum operator or the Fischer operator, one may establish a fast local rate of convergence for (2.5).

In the rest of the paper, we will restrict our development to the Fischer operator formulation. We make this choice since the step computation from this formulation only requires solving a linear equation, whereas for the Minimum operator formulation the step computation requires solving an optimization problem.

3 An algorithm for solving strictly convex QPs

In this section, we describe a complete algorithm for solving (1.5) using the iteration (2.5) applied with the Fischer formulation (i.e., $\Phi = \Phi^{\mathrm{FB}}$) when $\boldsymbol{Q} \succ 0$. The positive definiteness of Q_i for all $i \in \{1, \ldots, N\}$ ensures that each subproblem (1.2_i) is solvable and has a unique solution $x_i(\lambda)$ for any $\lambda \in \mathbb{R}^m$.

3.1 Step computation

For all k, we compute a step $d\lambda^k$ as a solution of $\Phi^{\mathrm{FB}}(\lambda^k) + H^k d\lambda^k = 0$ where $H^k \in \partial_B \Phi^{\mathrm{FB}}(\lambda^k)$. We will assume that $x_i(\lambda) \in C^1$ in a neighborhood around λ^k. From Lemma 7.1, this will hold for (1.2_i) provided that the (1.2_i)-LICQ, $\beta_i = \emptyset$, and (1.2_i)-SSOSC hold. De Luca, Facchinei, and Kanzow [27, Sec. 7] show that an element of the subdifferential $\partial_B \Phi^{\mathrm{FB}}(\lambda^k)$ can be computed as

$$H^k = D_\lambda^k + D_F^k \nabla F(\lambda^k) = D_\lambda^k + D_F^k \boldsymbol{A} \nabla \boldsymbol{x}(\lambda^k), \tag{3.1}$$

where D_F^k and D_λ^k are diagonal matrices defined, respectively, by

$$D_{\lambda[jj]}^k = \begin{cases} \left(\dfrac{\lambda_{[j]}^k}{\|(\lambda_{[j]}^k, F_{[j]}^k)\|} - 1 \right) & \text{for all } \; j \notin \beta^k \\ \left(\dfrac{z_{[j]}}{\|(z_{[j]}, z^T \nabla F_{[j]}^k)\|} - 1 \right) & \text{for all } \; j \in \beta^k \end{cases}$$

$$\text{and } D^k_{F[jj]} = \begin{cases} \left(\dfrac{F^k_{[j]}}{\|(\lambda^k_{[j]}, F^k_{[j]})\|} - 1 \right) & \text{for all } j \notin \beta^k \\ \left(\dfrac{z^T \nabla F^k_{[j]}}{\|(z_{[j]}, z^T \nabla F^k_{[j]})\|} - 1 \right) & \text{for all } j \in \beta^k \end{cases}$$

with $\beta^k := \{j : \lambda^k_{[j]} = 0 = F_{[j]}(\lambda^k)\}$, $F^k_{[j]} := F_{[j]}(\lambda^k)$, $\nabla F^k_{[j]} := \nabla F^k_{[j]}(\lambda^k)$, and $z \in \mathbb{R}^m$ chosen such that $z_{[\beta^k]} \neq 0$. The choice $z_{[j]} = 1$ for $j \in \beta^k$ is recommended [27].

Using (3.1) and the expression for $\nabla x_i(\lambda^k)$ in (7.15), the step is obtained from

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{B}^T_{\bar{\alpha}^k} & \boldsymbol{A}^T \\ \boldsymbol{B}_{\bar{\alpha}^k} & 0 & 0 \\ -D^k_F \boldsymbol{A} & 0 & D^k_\lambda \end{bmatrix} \begin{bmatrix} d\boldsymbol{x} \\ d\boldsymbol{\xi}_{[\bar{\alpha}^k]} \\ d\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\Phi^{\mathrm{FB}}(\lambda^k) \end{bmatrix} \tag{3.2}$$

where $\boldsymbol{B}^k_{\bar{\alpha}^k}$ is defined as in (2.6) with $\alpha^k_i = \{j : B_{i[j\cdot]} x_i(\lambda^k) = c_{[j]}\}$ replacing α_i.

3.2 Promoting global convergence

In order to promote progress by the algorithm from any initial point towards obtaining a solution to (1.5), we propose two approaches. The first is to employ a merit function, such as has been widely used in the context of semismooth equations [12,23]. The second is to employ a multidimensional filter mechanism [17,18,22]. We note at the outset that global convergence guarantees based on these mechanisms are not straightforward to obtain, since in our context the function F is only semismooth, as opposed to being C^1 as is commonly assumed in order to prove such guarantees. Still, we claim that, under reasonable conditions, these mechanisms should offer such guarantees. We comment on this further along with our descriptions of each mechanism in the following two subsections.

3.2.1 Merit function globalization

Solving the semismooth equation $\Phi^{\mathrm{FB}}(\lambda) = 0$ is equivalent to solving the following unconstrained optimization problem, assuming its optimal value is zero:

$$\min_\lambda \ \Psi^{\mathrm{FB}}(\lambda), \quad \text{where } \Psi^{\mathrm{FB}}(\lambda) := \tfrac{1}{2}\|\Phi^{\mathrm{FB}}(\lambda)\|_2^2. \tag{3.3}$$

For the Fischer function, it has been shown that, if $F \in C^1$, then $\Psi^{\mathrm{FB}}(\lambda)$ is continuously differentiable, and that, if F is a P_0-function, then stationary points of $\Psi^{\mathrm{FB}}(\lambda)$ correspond to zeros of $\Phi^{\mathrm{FB}}(\lambda)$ [27]. However, in our case, F is itself only semismooth. Still, one can use the function Ψ^{FB} as a merit function. In particular, progress towards satisfaction of $\Phi^{\mathrm{FB}}(\lambda) = 0$ can be gauged by reductions in Ψ^{FB}. We propose that, once a direction $d\lambda^k$ has been computed, one should aim to choose a steplength by

determining the smallest $r \in \mathbb{N}$ such that, for some $\sigma \in (0, 1)$,

$$\Psi^{\mathrm{FB}}(\lambda^k + \rho^r d\lambda^k) \leq (1 - \sigma\rho^r)\Psi^{\mathrm{FB}}(\lambda^k). \tag{3.4}$$

In terms of guaranteeing global convergence, it is easily seen that if $\Psi^{\mathrm{FB}}(\lambda^k) > 0$ for all k, yet there exists an infinite subsequence of iterations in which the steplength is bounded away from zero, then (3.4) ensures that $\{\Psi^{\mathrm{FB}}(\lambda^k)\} \to 0$.

3.2.2 Multidimensional filter globalization

The notion of a filter was introduced by Fletcher and Leyffer [17] as an alternative to a merit function for ensuring global convergence of an algorithm for solving constrained optimization problems. In this context, the filter was defined as a two-dimensional object with entries corresponding to the objective function value and a measure of constraint violation. The filter monitors progress toward a solution of the optimization problem by requiring that each iterate offers an objective or constraint violation value that is sufficiently small compared to those at previous iterates. Subsequently, several studies [16,19,48,49] have introduced filter mechanisms into various algorithmic frameworks for constrained optimization.

Concurrently, Fletcher and Leyffer [18] and Gould et al. [22] extended the filter idea to algorithms for solving nonlinear equations. Kanzow and Petra [24] applied a multidimensional filter for the globalization of a projected trust region method for a semismooth least squares formulation of mixed complementarity problems. In particular, they observed that the filter allowed for full Newton steps to be accepted on most iterations. Our motivation for applying such a filter similarly stems from a desire to reduce the number of evaluations of F, since each of these calls is expensive, requiring the solution of several optimization problems, i.e., (1.2_i) for all $i \in \{1, \ldots, N\}$. Here, we adapt the multidimensional filter approach from [22] for the globalization of the semismooth formulation (2.4) of (1.5).

Following [22], we define the function $\theta : \mathbb{R}^m \to \mathbb{R}^q$ as

$$\theta(\lambda) := \begin{bmatrix} \|\Phi_{[p_1]}(\lambda)\| \\ \vdots \\ \|\Phi_{[p_q]}(\lambda)\| \end{bmatrix} \tag{3.5}$$

where Φ is as defined in (2.4) and $\{p_1, \ldots, p_q\}$ is a set of mutually exclusive and exhaustive subsets of $\{1, \ldots, m\}$. We say that a vector $\lambda \in \mathbb{R}^m$ dominates another vector $\lambda' \in \mathbb{R}^m$ if $\theta_{[p_i]}(\lambda) \leq \theta_{[p_i]}(\lambda')$ for all $i \in \{1, \ldots, q\}$. A *filter* at the k-th iteration is a subset

$$\mathcal{F}^k \subseteq \{\theta(\lambda^0), \theta(\lambda^1), \ldots, \theta(\lambda^k)\} \tag{3.6}$$

such that λ^j does not dominate λ^l for any $l \neq j$ such that $\{\theta(\lambda^j), \theta(\lambda^l)\} \subseteq \mathcal{F}^k$.

Given a filter $\mathcal{F}^k$ and a trial point λ, we say that λ is *acceptable* to the filter if $\theta_{[p_i]}(\lambda)$ is sufficiently smaller than $\theta_{[p_i]}(\lambda^l)$ for some $i \in \{1, \ldots, q\}$ for all $\lambda^l \in \mathcal{F}_k$. In particular, λ is acceptable to the filter $\mathcal{F}^k$ if there exists $\gamma_\theta > 0$ such that for each

$\theta(\lambda^l) \in \mathcal{F}^k$ there exists an index $i \in \{1, \ldots, q\}$ such that

$$\theta_{[i]}(\lambda) \leq \theta_{[i]}(\lambda^l) - \gamma_\theta \|\theta(\lambda)\|. \tag{3.7}$$

If such a λ is obtained, then one sets the new filter as

$$\mathcal{F}^{k+1} := (\mathcal{F}^k \setminus \mathcal{F}^{k,d}(\lambda)) \cup \{\theta(\lambda)\}, \tag{3.8}$$

where $\mathcal{F}^{k,d}(\lambda) \subseteq \mathcal{F}^k$ is the set of entries in $\mathcal{F}^k$ that are dominated by λ, i.e.,

$$\mathcal{F}^{k,d}(\lambda) := \{\theta(\lambda^l) \in \mathcal{F}^k \mid \theta_{[i]}(\lambda) \leq \theta_{[i]}(\lambda^l) \text{ for all } i \in \{1, \ldots, q\}\}.$$

Most methods for optimization and solving systems of equations that employ filters and enjoy global convergence guarantees are based on trust region methodologies. However, there have been a few examples of filter-line-search methods for such problems as well [39,49]. For such a method, if there exists an infinite subsequence of iterations in which the steplength is bounded away from zero, then (3.7) in combination with the filter update (3.8) ensures convergence of the method.

3.3 Line search algorithm

Using the globalization mechanisms defined in Sect. 3.2, we propose a line search algorithm that is applicable to merit function and filter based strategies; see Algorithm 1 below. For each k, the algorithm computes the Newton step $d\lambda_N^k$ (see Step 7) as defined in (2.5), provided that $(H^k)^{-1}$ exists. If so, then a backtracking line search is performed along the Newton direction $d\lambda_N^k$ to compute the smallest $r_N \in \mathbb{N}$ that yields acceptance for the globalization mechanism: (i) if a merit function is used, then r_N is chosen such that (3.4) holds, while (ii) if a filter is used, then r_N is chosen such that the trial point is acceptable to the filter $\mathcal{F}^k$ according to the criterion in (3.7). If the Newton step is not computable or the steplength for the Newton step to be accepted is smaller than a user-defined constant $\rho^{\text{small}} > 0$, then the algorithm resorts to a different search direction. In particular, the algorithm computes $d\lambda_G^k = -H^k \Phi(\lambda^k)/\|H^k \Phi(\lambda^k)\|$, which is an approximation for the steepest descent direction of the function $\frac{1}{2}\|\Phi(\lambda)\|^2$ at the point λ^k. Similar to the Newton step, a backtracking line search is used to determine the smallest $r_G \in \mathbb{N}$ such that the trial point $(\lambda^k + \rho^{r_G} d\lambda_G^k)$ is acceptable. Using the appropriate step and steplength, the iterate λ^{k+1} is then set, and, in the case that a filter strategy is used, the new filter $\mathcal{F}^{k+1}$ is obtained. The algorithm terminates when either the norm of the most recent iterate displacement or the residual of the semismooth equation is below a user-defined tolerance $\epsilon > 0$.

4 An algorithm for solving convex or indefinite QPs

If (1.1) is (not strictly) convex or indefinite, then the subproblem solutions for a given λ, namely $\{x_i(\lambda)\}$, might not be unique. Further, in such cases, it is possible

Algorithm 1: Semismooth Newton Method (SNM) for Strictly Convex QP

Let $\epsilon \in (0, 1)$ be a desired convergence tolerance and let λ^0 be an initial iterate.
Choose $\{\rho, \gamma_\theta, \rho^{\text{small}}\} \subset (0, 1)$.
Set $k = 0$ (and $\mathcal{F}^0 = \{\theta(\lambda^0)\}$, if using a filter).
repeat
  Compute $H^k \in \partial_B \Phi(\lambda^k)$.
  `/* Compute Newton Step */`
  if *the linear system* (2.5) *solvable* **then**
    Set $d\lambda_N^k = -(H^k)^{-1}\Phi(\lambda^k)$.
    Find the smallest $r_N \in \mathbb{N}$ such that $(\lambda^k + \rho^{r_N} d\lambda_N^k)$ is acceptable (or set $r_N = \infty$ if no such value is found such that $\rho^{r_N} \geq \rho^{\text{small}}$).
    Set $d\lambda^k = \rho^{r_N} d\lambda_N^k$.
  else
    Set $r_N = \infty$.
  end
  `/* Compute Gradient Step if Newton Step fails */`
  if $\rho^{r_N} < \rho^{small}$ **then**
    Set $d\lambda_G^k = -H^k\Phi(\lambda^k)/\|H^k\Phi(\lambda^k)\|$.
    Find smallest $r_G \in \mathbb{N}$ such that $(\lambda^k + \rho^{r_G} d\lambda_G^k)$ is acceptable.
    Set $d\lambda^k = \rho^{r_G} d\lambda_G^k$.
  end
  `/* Update iterate */`
  Set $\lambda^{k+1} = \lambda^k + d\lambda^k$.
  `/* – Filter strategy only – Update filter */`
  Update the filter using (3.8).
  Set $k = k + 1$.
until $\|\lambda^k - \lambda^{k-1}\|_\infty \leq \epsilon$ *or* $\|\Phi(\lambda^k)\|_\infty \leq \epsilon$

that (1.2_i) is unbounded. We handle this through the addition of a penalized proximal term to (1.2_i). Our resulting algorithm updates the penalty parameter and the proximal parameter dynamically, as described in the following subsections.

4.1 Proximal-point subproblems

Given a proximal point $\widehat{x_i}$ for all $i \in \{1, \ldots, N\}$ and a penalty parameter $\mu > 0$, we consider modifying the subproblem (1.2_i) as

$$x_i(\lambda) = \arg\min_{x_i} \tfrac{1}{2} x_i^T Q_i x_i + (q_i + A_i^T \lambda)^T x_i + \frac{\mu}{2}\|x_i - \widehat{x_i}\|_2^2 \quad \text{s.t. } x_i \in \mathcal{X}_i. \tag{$1.2_i'$}$$

For sufficiently large $\mu > 0$, the proximal term in the objective function of this subproblem ensures that ($1.2_i'$) is solvable and $x_i(\lambda)$ is unique.

4.2 Penalty parameter and proximal-point update

For a fixed penalty parameter μ and proximal-point $\widehat{\boldsymbol{x}}$, Algorithm 1 with subproblems defined by (1.2$_i'$) can be applied to obtain a point that solves (1.1) with the objective augmented by the proximal term $(\mu/2)\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|_2^2$. However, a solution to (1.1) is only attained when $\mu\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\| \to 0$. Since attaining this by having $\mu \to 0$ will likely result in numerical difficulties, we propose Algorithm 2 (see page 16) for gradually reducing the contribution from the penalized proximal-point term.

The strategy contained in Algorithm 2 can be motivated as follows. In iteration k, the algorithm has proximal parameters $(\mu^k, \widehat{\boldsymbol{x}}^k)$. Given this pair, a point λ^{k+1} that is acceptable for the globalization strategy is computed as described in Algorithm 1; see Step 5 of Algorithm 2. Given the proximal-point modifications of the subproblems, this means that the algorithm produces λ^{k+1} and associated $\{x_i^{k+1}\}$ satisfying the first order stationary conditions of (1.2$_i'$), namely,

$$Q_i x_i^{k+1} + q_i + A_i^T \lambda^{k+1} + \mu^k (x_i^{k+1} - \widehat{x}_i^k) = 0 \tag{4.1a}$$

$$0 \leq c_i - B_i x_i^{k+1} \perp \xi_i^{k+1} \geq 0. \tag{4.1b}$$

If there is reason to believe that $\{\mu^k \|\boldsymbol{x}^{k+1} - \widehat{\boldsymbol{x}}^k\|\} \to 0$ while $\{\Phi(\lambda^{k+1})\} \to 0$, then one might consider leaving the proximal parameters unchanged so that the algorithm may continue to iterate and behave as Algorithm 1. Otherwise, it is imperative that the parameters are updated so that $\{\mu^k \|\boldsymbol{x}^{k+1} - \widehat{\boldsymbol{x}}^k\|\} \to 0$. Our update strategy for these parameters is inspired by the Fiacco-McCormick-type barrier parameter update strategy [14] and the proximal point update strategy used in proximal-point algorithms [45]. We consider the satisfaction of two conditions in Step 6 in order to update the parameters.

(i) If $\mu^k \|\boldsymbol{x}^{k+1} - \widehat{\boldsymbol{x}}^k\|_\infty \leq \epsilon$, then the influence of the proximal term may be considered negligible and (4.1a)–(4.1b) imply that $\boldsymbol{x}^{k+1}$ is close to solving (1.2$_i$). Therefore, we choose not to update the proximal parameters when this holds, which is the first condition in Step 6.

(ii) If $\|\Phi(\lambda^{k+1})\|_\infty \gg 0$, then the algorithm has not yet neared a solution of (2.4). If the proximal terms were updated in such an iteration, then the algorithm runs the risk of updating these parameters too often based on information from points that are far from solutions of the semismooth equations. Therefore, we choose not to update the parameters at such points, and instead only update them when $\|\Phi(\lambda^{k+1})\|_\infty \leq \kappa^k \mu^k$, where the right-hand side represents a tolerance that converges to zero along with the proximal parameters $\{\mu^k\}$.

Thus, if the conditions in (i)–(ii) hold, then the algorithm updates the proximal point $\widehat{\boldsymbol{x}}$ and either μ^k or κ^k according to the following strategy.

– If the problem is convex, then the penalty parameter is decreased, but no smaller than μ^{small}; see Step 8. The lower bound of μ^{small} is imposed to ensure that the procedure for solving (1.2$_i'$) does not encounter numerical difficulties. This reduction in the penalty parameter has the desired effect of reducing the possible influence of the proximal term.

– If the problem is indefinite, then we choose $\mu^0 > -\min_{i\in\{1,\ldots,N\}}\{\sigma_{\min}(Q_i)\}$, where $\sigma_{\min}(Q_i)$ denotes the left-most eigenvalue of Q_i. This ensures that (1.2$_i'$) is strictly convex for all $i \in \{1,\ldots,N\}$, so that the subproblems are well-posed and have unique solutions. To maintain this feature, when (i)–(ii) hold, we do not reduce the penalty parameter, but rather reduce the factor κ, though not below κ^{small}. This ensures that the semismooth equations have to be satisfied to a tighter tolerance before the proximal parameters are updated again.

In either case, the decrease in μ or κ has the effect of tightening the tolerance to which the semismooth equations must be satisfied before the proximal parameters are updated. If the filter globalization strategy is employed, then the algorithm clears the current filter and adds the entry corresponding to the residual evaluated using the updated parameters; see Step 19. The algorithm continues until the proximal and residual terms are both below the required tolerance; see Step 22.

Algorithm 2: SNM for Convex and Indefinite QP

1 Let $\epsilon \in (0,1)$ be a desired convergence tolerance, $\mu^0 > 0$ an initial penalty parameter, κ^0 an initial convergence factor, λ^0 an initial iterate, and $\widehat{\boldsymbol{x}}^0$ an initial proximal point.
2 Choose $(\rho, \gamma_\theta, \rho^{\text{small}}, \mu^{\text{small}}, \delta_\mu, \kappa^{\text{small}}, \delta_\kappa) \in (0,1)$.
3 Set $k = 0$ (and $\mathcal{F}^0 = \{\theta(\lambda^0)\}$, if using a filter).
4 **repeat**
5 Compute λ^{k+1} as in Algorithm 1 with subproblems (1.2$_i'$) with $(\mu, \widehat{\boldsymbol{x}}) = (\mu^k, \widehat{\boldsymbol{x}}^k)$, letting $\{x_i^{k+1}\}$ denote the corresponding solutions of (1.2$_i'$).
6 **if** $\mu^k\|\boldsymbol{x}^{k+1} - \widehat{\boldsymbol{x}}^k\|_\infty > \epsilon$ *and* $\|\Phi(\lambda^{k+1})\|_\infty \le \kappa^k\mu^k$ **then**
`/* Reduce μ or κ, update x̂ */`
7 **if** (1.1) *is convex* **then**
8 Set $\mu^{k+1} = \max\{\mu^{\text{small}}, \delta_\mu\mu^k\}$.
9 **else**
10 Set $\kappa^{k+1} = \max\{\kappa^{\text{small}}, \delta_\kappa\kappa^k\}$.
11 **end**
12 Set $\widehat{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^{k+1}$.
13 Compute $(\lambda^{k+1}, \{x_i^{k+1}\})$ as in Step 5 with $(\mu, \widehat{\boldsymbol{x}}) = (\mu^{k+1}, \widehat{\boldsymbol{x}}^{k+1})$.
14 **else**
`/* Do not update proximal parameters */`
15 Set $\mu^{k+1} = \mu^k$, $\kappa^{k+1} = \kappa^k$, $\widehat{\boldsymbol{x}}^{k+1} = \widehat{\boldsymbol{x}}^k$.
16 **end**
`/* – Filter strategy only – Update filter */`
17 **if** ρ^{k+1} *or* $\widehat{\boldsymbol{x}}^{k+1}$ *has been updated* **then**
18 Set $\mathcal{F}^k = \emptyset$.
19 Update the filter using (3.8).
20 **end**
21 Set $k = k + 1$.
22 **until** $\mu^k\|\boldsymbol{x}^k - \widehat{\boldsymbol{x}}^k\|_\infty \le \epsilon$ *and* $\|\Phi(\lambda^k)\|_\infty \le \epsilon$

5 Implementation and numerical experiments

We implemented Algorithms 1 and 2 in MATLAB. We used the QP solver `MINQ` [38] developed by Neumaier. `MINQ` can handle both convex and indefinite bound constrained QPs. General equalities and inequalities can be handled in the case of strictly convex QPs since `MINQ` works on the dual QP, which is bound-constrained. The parameters in Algorithm 1 were set as $\rho = 0.9$, $\gamma_\theta = 10^{-4}$, and $\rho^{\text{small}} = 10^{-8}$. The additional parameters in Algorithm 2 were set as $\mu^{\text{small}} = \kappa^{\text{small}} = 10^{-4}$ and $\delta_\mu = \delta_\kappa = 0.5$. The initial proximal parameters were set as $\mu^0 = 10^{-1} + 10^{-4}$, $\kappa^0 = 1$, and $\widehat{\boldsymbol{x}} = 0$. In all our experiments, the initial multipliers were set as $\lambda^0 = 0$.

For comparison purposes, we also implemented in MATLAB a standard subgradient algorithm with normalized steps and diminishing stepsizes [46].

5.1 Structured random QP generator

We randomly generated QPs with the structure of problem (1.1) at least satisfying the desirable properties of (1.2_i)-LICQ, (1.2_i)-SSOSC and (1.2_i)-SC. The procedures for generating the QPs can be described as follows. In all cases, we simultaneously generated a primal-dual solution vector $(\boldsymbol{x}^*, \boldsymbol{\xi}^*)$ and problem data so that at least $(\boldsymbol{x}^*, \boldsymbol{\xi}^*)$ corresponds to a stationary point of the resulting problem.

For simplicity, the inequalities in (1.2_i) were simply chosen as the nonnegativity constraints $x_i \geq 0$. This implies the satisfaction of (1.2_i)-LICQ. Given $n_a < n$, a number of components of x_i^* that were desired to be at the bound of 0, the optimal solution $\boldsymbol{x}^*$ and bound multipliers $\boldsymbol{\xi}^*$ were chosen to satisfy (1.2_i)-SC by

$$x^*_{i[j]} = \begin{cases} 0 & \text{for all } j \leq n_a \\ 0.1 + \texttt{rand}(0,1) & \text{for all } j > n_a \end{cases}$$

$$\text{and } \xi^*_{i[j]} = \begin{cases} 0.1 + \texttt{rand}(0,1) & \text{for all } j \leq n_a \\ 0 & \text{for all } j > n_a, \end{cases}$$

for all $i \in \{1, \ldots, N\}$. Here, `rand`$(0, 1)$ denotes a uniform random number in $[0, 1]$. Letting Y_i and Z_i respectively denote bases for the range and null spaces of the active bounds, the Hessian for the subproblems were randomly chosen by

$$Q_i = \begin{cases} U_i \text{Diag}(u_i) U_i^T & \text{for strictly convex QPs} \\ U_i (Z_i \text{Diag}(u_{i,Z}) Z_i) U_i^T & \text{for convex QPs} \\ U_i (Y_i \text{Diag}(u_{i,Y}) Y_i^T + Z_i \text{Diag}(u_{i,Z}) Z_i^T) U_i^T & \text{for indefinite QPs,} \end{cases}$$

where $U_i \in \mathbb{R}^{n \times n}$ is a unitary matrix, $u_i \in \mathbb{R}^n$ had elements $u_{i[j]} = \texttt{rand}(0, 1)$, $u_{i,Z} \in \mathbb{R}^{n-n_a}$ had elements $u_{i,Z[j]} = \texttt{rand}(0, 1)$, and $u_{i,Y} \in \mathbb{R}^{n_a}$ had elements $u_{i,Y[j]} = \texttt{rand}(-10^{-1}, 0)$. Since Q_i was positive definite in the null space of bound constraints that were active at x_i^* and (1.2_i)-SC holds, (1.2_i)-SSOSC was satisfied.

For the indefinite QPs, we additionally included the subproblem upper bounds $x_{i[j]} \leq \|\boldsymbol{x}^*\|_\infty + 1$. This guaranteed the existence of minima for (1.2_i) for any λ due to compactness of the feasible set. The choice ensured that these upper bounds were inactive at $\boldsymbol{x}^*$ (though they may be active at other stationary points).

Let us now describe the generation of the coupling equality and/or inequality constraints. Suppose m_e $(\leq nN - n_a N)$ was the number equality constraints that were desired in the problem and suppose m_a $(\leq nN - n_a N - m_e)$ was the number coupling inequality constraints that were desired to be active at $\boldsymbol{x}^*$. The coupling equality and active inequality constraints were obtained as

$$\left[A_{[\cdot 1]} \cdots A_{[\cdot m_a]}\right]^T = ZV$$

where $Z \in \mathbb{R}^{nN \times (n-n_a)N}$ was block diagonal with Z_i on the diagonal and $V \in \mathbb{R}^{(n-n_a)N \times (m_e+m_a)}$ were $(m_e + m_a)$ columns of an $(n - n_a)N \times (n - n_a)N$ unitary matrix. The entries of the $(m - m_a)$ remaining rows of the matrix A for coupling inequality constraints were chosen at random without any structure. The right hand side of the inequality constraints were chosen as

$$b_{[j]} = \begin{cases} A_{[\cdot j]}x^* & \text{for all } j \leq m_e + m_a \\ A_{[\cdot j]}x^* + 1 & \text{for all } j > m_a \end{cases}$$

$$\text{and } \lambda^*_{[j]} = \begin{cases} \texttt{randn}(0, 1) & \text{for all } j \leq m_e \\ 0.1 + \texttt{rand}(0, 1) & \text{for all } m_e < j \leq m_e + m_a \\ 0 & \text{for all } j > m_a \end{cases}$$

where `randn`$(0, 1)$ denotes a random number drawn from normal distribution with mean 0 and standard deviataion 1. Thus, the multipiers for the equality constraints are unsrestricted in sign. Finally, the linear term in the QP was chosen by

$$q_i = -(Q_i x_i^* - v_i^* + A_i^T \lambda_i^*) \text{ for all } i \in \{1, \ldots, N\}.$$

5.2 Strictly convex QPs

We generated 100 strictly convex QP instances using the procedure outlined in Sect. 5.1 for different values of $\{n, N, m, n_a, m_a\}$. Three different globalization strategies—`merit` (merit function), `filter-2` (filter with 2 entries), and `filter-5` (filter with 5 entries)—were used to solve the problems. Table 1 summarizes the performance of Algorithm 1 on QPs with only coupling inequality constraints. One finds that the results are consistent for all problem sizes and globalization strategies. In all cases, the average numbers of iterations and function evaluations were roughly around 10 or fewer, despite the instances ranging in dimension from $nN = 100$ to 400. One also finds that the average residual for the semismooth equations at termination was below 10^{-6} while the average CPU time (in seconds) was consistently small.

We also generated 100 strictly convex QPs with both equality and inequality coupling constraints. Table 2 summarizes the performance of Algorithm 1 on these QPs,

where in all cases we chose $m_e = m$. One again finds consistent results, except in the fact that the `filter-5` strategy performed slightly worse for the smaller dimensional problems. This behavior may be attributed to a combination of having fewer degrees of freedom at the solution and the filter being too flexible as to allowing a step to be accepted as long as progress is made in a smaller number of entries of the residual. Still, however, the performance in terms of all measures is consistently good even for the larger problems.

In Tables 3 and 4, we provide the results obtained by running a subgradient algorithm (`subgrad`), the *proximal Jacobian ADMM* (i.e., alternating direction method of multipliers) method of [7] (`prox-admm`), and the *proximal center* algorithm of [32] (which uses a smoothing technique proposed by Nesterov) (`prox-cent`). For `prox-admm`, the penalty parameters ρ, γ, and τ_i in [7, Algorithm 4, Eq. (1.3)] are set to 0.1, 0.1, and $1.01\rho\|A_i\|^2$, respectively. These choices satisfy [7, Lemma 2.2] and were also employed by the authors in their experiments. The parameters are adjusted adaptively as suggested in [7, Algorithm 2.3]. The parameters in `prox-cent` are c and L_c [32, Algorithm 3.2], which are chosen as ϵ and $\frac{1}{\epsilon}\sum_{i=1}^{n}(\|A_i\|^2/\sigma_i)$ where σ_i is the smallest positive eigenvalue of $Q_i \succ 0$ and 1 otherwise. In all instances, `subgrad` and `prox-cent` did not obtain a solution with a residual below 10^{-6} within the maximum iteration limit of 5000 or the CPU time limit (in seconds) of 900. The reported final residuals illustrate that the algorithms made some progress, but with CPU times so much higher than Algorithm 1 one can conclude that the performance of these methods is inferior in our setting. At the same time, `prox-admm` successfully solved some problems, but it is not reliable for this set and still required many more iterations, function evaluations, and CPU seconds than Algorithm 1 even in successful cases. Table 3 provides the results for the case of only coupling inequalities (i.e., the same problems used to generate the results in Table 1). Table 4 provides the results for coupling equality and inequality constraints (i.e., the same problems used for Table 2).

5.3 Convex QPs

Following similar procedures as in Sect. 5.2, we generated 100 (not strictly) convex QPs using the procedure outlined in Sect. 5.1, each when only inequality coupling constraints were present and when equality and inequality coupling constraints were present. The results for only inequality coupling constraints are given in Table 5 and the results for equality and inequality coupling constraints are given in Table 6. The results are again consistently good for all problem sizes and globalization strategies. In terms of function evaluations, the filter strategies outperformed the merit function strategy when only inequality coupling constraints were present, though again the `filter-5` strategy lagged behind the other strategies in terms of both of these performance measures when equality and inequality coupling constraints were present. Still, however, all strategies solved 100% of the problems in a number of iterations significantly fewer than the problem dimension.

Tables 7 and 8 report the corresponding performace of `subgrad`, `prox-admm`, and `prox-cent`. For essentially all instances, these other methods did not obtain a solution with residual below 10^{-6} within the iteration and time limits.

Table 1 Algorithm 1: strictly convex QPs with coupling inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`merit`	10	10	20	2	5	100	7.0	9.6	2.7e−08	0.5
`filter-2`	10	10	20	2	5	100	8.5	13.6	3.5e−08	0.6
`filter-5`	10	10	20	2	5	100	9.1	12.8	3.6e−08	0.6
`merit`	20	20	20	5	5	100	5.9	7.0	2.3e−08	5.6
`filter-2`	20	20	20	5	5	100	6.0	7.0	2.3e−08	5.8
`filter-5`	20	20	20	5	5	100	6.0	7.0	2.3e−08	5.7
`merit`	10	10	20	5	10	100	6.4	8.1	7.4e−08	0.3
`filter-2`	10	10	20	5	10	100	6.5	7.6	7.4e−08	0.3
`filter-5`	10	10	20	5	10	100	6.6	7.7	7.4e−08	0.3
`merit`	20	20	20	10	10	100	5.3	6.3	1.3e−07	3.5
`filter-2`	20	20	20	10	10	100	5.3	6.3	1.3e−07	3.6
`filter-5`	20	20	20	10	10	100	5.3	6.3	1.3e−07	3.6

Table 2 Algorithm 1: strictly convex QPs with coupling equalities and inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`merit`	10	10	20	2	5	100	12.5	34.3	1.1e−08	1.6
`filter-2`	10	10	20	2	5	100	20.9	52.1	1.9e−08	2.5
`filter-5`	10	10	20	2	5	100	50.4	119.5	9.5e−09	5.7
`merit`	20	20	20	5	5	100	8.1	10.6	2.8e−08	8.7
`filter-2`	20	20	20	5	5	100	8.7	10.2	3.7e−08	8.6
`filter-5`	20	20	20	5	5	100	8.8	10.4	3.7e−08	8.8
`merit`	10	10	20	5	10	100	10.6	31.6	3.8e−08	1.1
`filter-2`	10	10	20	5	10	100	12.8	30.6	4.7e−08	1.1
`filter-5`	10	10	20	5	10	100	19.2	36.2	3.9e−08	1.4
`merit`	20	20	20	10	10	100	6.8	8.5	2.2e−08	4.8
`filter-2`	20	20	20	10	10	100	6.8	7.8	3.0e−08	4.4
`filter-5`	20	20	20	10	10	100	6.8	7.8	3.0e−08	4.5

Table 3 Other methods: strictly convex QPs with coupling inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`subgrad`	10	10	20	2	5	0	5002.0	10005.0	2.8e−02	258.6
`prox-admm`	10	10	20	2	5	18	4855.4	4856.4	4.1e−05	180.1
`prox-cent`	10	10	20	2	5	0	5002.0	10004.0	7.5e−03	362.0
`subgrad`	20	20	20	5	5	0	1012.7	2026.4	5.8e−02	900.5
`prox-admm`	20	20	20	5	5	0	1167.6	1168.6	1.6e−02	900.4
`prox-cent`	20	20	20	5	5	0	621.4	1242.8	1.1e+00	900.8
`subgrad`	10	10	20	5	10	0	5002.0	10005.0	3.0e−02	186.4
`prox-admm`	10	10	20	5	10	100	2344.5	2345.5	1.0e−06	65.0
`prox-cent`	10	10	20	5	10	0	5002.0	10004.0	5.2e−03	269.4
`subgrad`	20	20	20	10	10	0	1510.9	3022.7	4.8e−02	900.3
`prox-admm`	20	20	20	10	10	0	1691.0	1692.0	4.4e−03	900.3
`prox-cent`	20	20	20	10	10	0	907.0	1813.9	5.7e−01	900.5

Table 4 Other methods: strictly convex QPs with coupling equalities and inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`subgrad`	10	10	20	2	5	0	5002.0	10005.0	9.4e−02	269.2
`prox-admm`	10	10	20	2	5	97	3313.4	3314.4	1.1e−06	122.5
`prox-cent`	10	10	20	2	5	0	5002.0	10004.0	8.7e−03	360.1
`subgrad`	20	20	20	5	5	0	1011.6	2024.3	1.4e−01	900.5
`prox-admm`	20	20	20	5	5	0	1154.6	1155.6	3.1e−02	900.4
`prox-cent`	20	20	20	5	5	0	676.0	1351.9	2.3e+00	900.7
`subgrad`	10	10	20	5	10	0	5002.0	10005.0	5.5e−02	194.4
`prox-admm`	10	10	20	5	10	1	4991.8	4992.8	1.2e−04	138.2
`prox-cent`	10	10	20	5	10	0	5002.0	10004.0	7.2e−03	267.5
`subgrad`	20	20	20	10	10	0	1410.1	2821.3	8.4e−02	900.3
`prox-admm`	20	20	20	10	10	0	1617.5	1618.5	4.2e−03	900.3
`prox-cent`	20	20	20	10	10	0	927.4	1854.8	1.3e+00	900.5

Table 5 Algorithm 2: convex QPs with coupling inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`merit`	10	10	20	2	5	100	18.2	39.5	1.3e−07	2.1
`filter-2`	10	10	20	2	5	100	18.3	22.9	1.2e−07	1.5
`filter-5`	10	10	20	2	5	100	20.3	29.3	1.3e−07	1.8
`merit`	20	20	20	5	5	100	23.6	55.0	1.7e−07	54.0
`filter-2`	20	20	20	5	5	100	25.3	39.9	1.6e−07	43.0
`filter-5`	20	20	20	5	5	100	25.1	26.3	1.5e−07	32.0
`merit`	10	10	20	5	10	100	17.1	27.9	1.6e−07	1.3
`filter-2`	10	10	20	5	10	100	17.8	30.0	1.4e−07	1.3
`filter-5`	10	10	20	5	10	100	23.8	35.4	1.3e−07	1.6
`merit`	20	20	20	10	10	100	20.6	37.1	1.9e−07	27.2
`filter-2`	20	20	20	10	10	100	21.7	22.9	1.7e−07	19.7
`filter-5`	20	20	20	10	10	100	21.8	22.8	1.7e−07	19.7

Table 6 Algorithm 2: convex QPs with coupling equalities and inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`merit`	10	10	20	2	5	100	20.1	52.0	1.2e−07	2.7
`filter-2`	10	10	20	2	5	100	31.2	77.4	1.5e−07	3.9
`filter-5`	10	10	20	2	5	100	63.6	146.1	1.1e−07	7.4
`merit`	20	20	20	5	5	100	19.4	27.0	2.3e−07	29.8
`filter-2`	20	20	20	5	5	100	19.9	21.1	2.5e−07	25.2
`filter-5`	20	20	20	5	5	100	19.9	20.9	2.4e−07	25.0
`merit`	10	10	20	5	10	100	24.1	141.9	1.3e−07	4.9
`filter-2`	10	10	20	5	10	100	32.0	130.8	1.6e−07	4.7
`filter-5`	10	10	20	5	10	100	57.7	253.7	1.8e−07	8.9
`merit`	20	20	20	10	10	100	17.9	26.6	3.1e−07	20.0
`filter-2`	20	20	20	10	10	100	19.4	21.5	2.9e−07	17.7
`filter-5`	20	20	20	10	10	100	19.9	22.2	2.9e−07	17.9

Table 7 Other methods: convex QPs with coupling inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`subgrad`	10	10	20	2	5	0	5002.0	10005.0	4.5e−02	309.2
`prox-admm`	10	10	20	2	5	0	5002.0	5003.0	6.2e−04	182.7
`prox-cent`	10	10	20	2	5	0	5002.0	10004.0	4.4e+00	297.9
`subgrad`	20	20	20	5	5	0	986.4	1973.7	1.2e−01	900.5
`prox-admm`	20	20	20	5	5	0	1191.0	1192.0	2.0e−02	900.4
`prox-cent`	20	20	20	5	5	0	759.7	1519.4	2.8e+01	900.6
`subgrad`	10	10	20	5	10	0	5002.0	10005.0	7.2e−02	212.0
`prox-admm`	10	10	20	5	10	0	5002.0	5003.0	3.8e−04	138.0
`prox-cent`	10	10	20	5	10	0	4952.0	9904.0	6.2e+01	227.9
`subgrad`	20	20	20	10	10	0	1527.1	3055.1	1.3e−01	900.3
`prox-admm`	20	20	20	10	10	0	1718.3	1719.3	1.2e−02	900.3
`prox-cent`	20	20	20	10	10	0	1176.8	2353.6	2.2e+03	891.8

Table 8 Other methods: convex QPs with coupling equalities and inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`subgrad`	10	10	20	2	5	0	5002.0	10005.0	1.1e−01	278.4
`prox-admm`	10	10	20	2	5	0	5002.0	5003.0	8.7e−05	184.7
`prox-cent`	10	10	20	2	5	0	5002.0	10004.0	2.0e+01	279.6
`subgrad`	20	20	20	5	5	0	1098.8	2198.6	1.4e−01	900.4
`prox-admm`	20	20	20	5	5	0	1187.0	1188.0	8.0e−02	900.4
`prox-cent`	20	20	20	5	5	0	842.4	1684.8	8.8e+01	900.5
`subgrad`	10	10	20	5	10	0	5002.0	10005.0	1.1e−01	205.7
`prox-admm`	10	10	20	5	10	6	4993.5	4994.5	3.0e−05	138.3
`prox-cent`	10	10	20	5	10	0	4952.0	9904.0	5.4e+01	237.1
`subgrad`	20	20	20	10	10	0	1657.9	3316.8	1.3e−01	900.3
`prox-admm`	20	20	20	10	10	0	1704.8	1705.8	5.1e−02	900.3
`prox-cent`	20	20	20	10	10	0	1142.8	2285.7	1.5e+03	882.4

5.4 Indefinite QPs

Following similar procedures as in Sect. 5.2, we generated 100 indefinite QPs using the procedure outlined in Sect. 5.1, each when only coupling inequalities were present and when coupling equalities and inequalities were present. Recall that the procedure in Sect. 5.1 produces Hessians Q_i with eigenvalues in $(-10^{-1}, 1)$ and that, for such indefinite QPs, Algorithm 1 does not update the proximal parameter μ; in particular, we have $\mu^k = \mu^0 = 10^{-1} + 10^{-4}$ for all k. Thus, in these experiments, the subproblems are all convex despite the fact that the overall QP is indefinite.

The results for only inequality coupling constraints are given in Table 9 and the results for equality and inequality coupling constraints are given in Table 10. The results for all globalization strategies are good when only inequality coupling constraints are present, though there are some failures for the problems with larger (n_a, m_a). As for the problems with equality and inequality coupling constraints, we find that the `filter-5` strategy led to many failures, suggesting that the other globalization strategies are preferred for these problems. We remark that some additional instances are solved by the `merit` and `filter-2` methods when the time limit is set higher than 900 seconds. However, for our purposes here, we simply state the results obtained with this time limit.

We do not provide the results for `subgrad`, `prox-admm`, and `prox-cent` for these indefinite QPs since these methods are not designed to solve nonconvex problems. Results were obtained for `subgrad` for these problems, but since it did not solve any instances successfully and the final residual values were all consistently worse than those obtained using Algorithm 1, we do not bother stating the results here.

6 Applications

In this section, we describe three important real-world applications for our algorithms. In Sect. 6.1, we describe how the computation of market-clearing prices in electricity markets can lead to a strictly convex QP with coupling constraints that include equalities and inequalities. In Sect. 6.2, we show how nearly-separable QPs arise in the analysis of nonlinear network flow problems. This applicaiton gives rise to strictly convex problems (not necessarily QPs) with coupling equality constraints. Finally, two-stage stochastic optimization problems are described in Sect. 6.3. In particular, we present a newsvendor problem with uncertain demands and quadratic costs, leading to indefinite QPs with coupling inequality constraints.

6.1 Electricity markets

A prime application in which instances of problem (1.1) arise is in the establishment of competitive equilibria in electricity markets. For example, consider a market in which supply nodes (generation companies, or GenCos, denoted by $\mathcal{N}^G$) and demand nodes (Distribution System Operators, or DSOs, denoted by $\mathcal{N}^D$) are spread across a network (with edges or electrical lines denoted by $\mathcal{E}$). Over such a network, the goal of an independent system operator (ISO) is to determine prices that establish a competitive

Table 9 Algorithm 2: indefinite QPs with coupling inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`merit`	10	10	20	2	5	100	226.0	719.7	6.6e−08	29.3
`filter-2`	10	10	20	2	5	100	251.4	439.4	6.2e−08	20.5
`filter-5`	10	10	20	2	5	100	318.3	472.0	8.0e−08	22.9
`merit`	20	20	20	5	5	48	260.8	1247.7	2.5e−03	772.1
`filter-2`	20	20	20	5	5	81	354.8	860.1	2.1e−03	576.2
`filter-5`	20	20	20	5	5	97	448.1	486.9	4.0e−07	369.5
`merit`	10	10	20	5	10	100	229.5	658.1	6.1e−08	21.2
`filter-2`	10	10	20	5	10	100	262.8	380.5	4.9e−08	14.8
`filter-5`	10	10	20	5	10	94	355.1	7872.6	3.1e+01	73.2
`merit`	20	20	20	10	10	69	420.8	1830.8	7.7e−04	672.6
`filter-2`	20	20	20	10	10	99	600.3	656.0	4.0e−07	305.5
`filter-5`	20	20	20	10	10	98	622.2	679.9	2.1e−03	318.8

Table 10 Algorithm 2: indefinite QPs with coupling equalities and inequalities

	n	N	m	n_a	m_a	# solved	Avg. #Iters.	Avg. #Func.	Avg. Res.	Avg. CPU (s)
`merit`	10	10	20	2	5	94	213.3	5946.9	2.3e−02	88.0
`filter-2`	10	10	20	2	5	85	397.2	21662.1	9.2e−01	224.0
`filter-5`	10	10	20	2	5	0	654.3	142829.5	1.0e+02	901.0
`merit`	20	20	20	5	5	22	240.5	1329.5	4.2e−03	871.4
`filter-2`	20	20	20	5	5	94	562.3	621.6	4.7e−04	487.4
`filter-5`	20	20	20	5	5	57	808.2	10481.4	3.5e+01	635.3
`merit`	10	10	20	5	10	67	209.6	30916.5	1.2e−01	332.5
`filter-2`	10	10	20	5	10	57	368.9	52480.7	1.0e+00	472.5
`filter-5`	10	10	20	5	10	0	636.0	140952.9	1.8e+02	900.9
`merit`	20	20	20	10	10	42	384.7	1846.7	1.7e−03	788.4
`filter-2`	20	20	20	10	10	97	708.3	840.6	3.8e−04	447.1
`filter-5`	20	20	20	10	10	20	590.5	23888.0	9.3e+01	835.9

equilibrium, which can be done by formulating a social welfare maximization problem. If the GenCos assume quadratic power generation costs, the DSOs assume quadratic consumption costs, and a DC power flow model is used throughout the network, then one obtains a problem equivalent to an instance of (1.1) with the power flow equations playing the role of the coupling constraints. An added feature of such a problem is that, for privacy reasons, the individual market agents may be unwilling to share their specific cost function with the ISO. This necessitates the use of a distributed optimization framework, in which the ISO transmits price signals to the agents, who in turn only communicate their solutions and/or the sensitivities of their solutions to the price signals.

By the second fundamental theorem of welfare economics [28,50], a competitive equilibrium can be characterized by maximizing social welfare through

$$\min_{P} \sum_{i \in \mathcal{N}^G} c_i(P_i) - \sum_{i \in \mathcal{N}^D} u_i(-P_i) \tag{6.1a}$$

$$\text{s.t. } \mathbb{1}^T P = 0 \tag{6.1b}$$

$$-\overline{P} \leq AP \leq \overline{P} \tag{6.1c}$$

$$\underline{P}_i^G \leq P_i \leq \overline{P}_i^G, \quad \text{for all } \; i \in \mathcal{N}^G \tag{6.1d}$$

$$\underline{P}_i^D \leq -P_i \leq \overline{P}_i^D, \quad \text{for all } \; i \in \mathcal{N}^D, \tag{6.1e}$$

where c_i is a strictly convex function representing the cost of generation for GenCo $i \in \mathcal{N}^G$ and u_i is a strictly concave utility function for DSO $i \in \mathcal{N}^D$. The vector $\overline{P} \in \mathbb{R}^{|\mathcal{E}|}$ denotes the power limits on the lines in the network, $\mathbb{1} \in \mathbb{R}^{|\mathcal{N}|}$ is a vector of all ones, and A is the matrix of power distribution factors for the ISO's transmission network. The constraint (6.1b) imposes power balance between the GenCos and DSOs. The DC power flow model of the ISO network is modeled in (6.1c) through the power distribution factors [51].

Introducing multipliers $\xi \in \mathbb{R}$ for (6.1b) and $(\underline{\zeta}, \overline{\zeta}) \in \mathbb{R}_+^{|\mathcal{E}|} \times \mathbb{R}_+^{|\mathcal{E}|}$ for the lower and upper bound constraints in (6.1c), the coupling constraints (6.1b)–(6.1c) can be dualized. This results in the dual function

$$P(\boldsymbol{\nu}) \in \arg\min_{P} \sum_{i \in \mathcal{N}^G} c_i(P_i) - \sum_{i \in \mathcal{N}^D} u_i(-P_i) - \boldsymbol{\nu}^T P \tag{6.2a}$$

$$\text{s.t. } \underline{P}_i^G \leq P_i \leq \overline{P}_i^G, \quad \text{for all } \; i \in \mathcal{N}^G \tag{6.2b}$$

$$\underline{P}_i^D \leq -P_i \leq \overline{P}_i^D, \quad \text{for all } \; i \in \mathcal{N}^D, \tag{6.2c}$$

where the right-hand side optimization problems are separable in the GenCos and DSOs. The optimization problems are parameterized by $\boldsymbol{\nu} = \xi \mathbb{1} + A^T(\underline{\zeta} - \overline{\zeta})$, which has the economic interpretation as the *locational marginal price* or *nodal price* of electricity. Thus, (6.2) allows the GenCos and DSOs to maintain their privacy of their objective functions and interact with the ISO only through a price signal ν_i. The ISO, which is responsible for maintaining the balance of power supply and adherence of

Table 11 Results for the social welfare maximization problem (6.1)

Name	Semismooth Avg. #Iters.	Avg. #Fcn.	Avg. CPU (s)	Subgradient Avg. #Iters.	Avg. CPU (s)
case9	5.5	19.6	0.01	100000	1.8
case14	6.1	49.7	0.03	100000	2.1
case30	5.6	19.7	0.02	100000	3.1
case39	10.0	90.7	0.07	43262	1.6
case57	7.6	34.7	0.04	100000	4.4
case118	6.0	32.3	1.08	100000	89.6

line flow limits, can be viewed as solving the ICP

$$\boldsymbol{v} = \xi \mathbb{1} + A^T(\underline{\zeta} - \overline{\zeta}) \tag{6.3a}$$

$$\mathbb{1}^T P(\boldsymbol{v}) = 0 \tag{6.3b}$$

$$0 \leq \underline{\zeta} \perp AP(\boldsymbol{v}) + \overline{P} \geq 0 \tag{6.3c}$$

$$0 \leq \overline{\zeta} \perp -AP(\boldsymbol{v}) + \overline{P} \geq 0. \tag{6.3d}$$

If $\{(c_i, u_i)\}$ are quadratic, then we obtain subproblems of the form in (1.2_i). For such cases, the approach proposed in this paper has been pursued by the authors in [44], some of the results of which we provide now. The cost function for the GenCos is chosen as a strictly convex quadratic function, $c_i(P) = c_{1i}P + c_{2i}P^2$ where $c_{2i} > 0$. The values for the coefficients c_{1i} and c_{2i} are generated randomly. The utility function for the DSOs is chosen as a strictly concave quadratic function, $u_i(-P) = u_{i1}(-P) + u_{2i}P^2$ where $u_{2i} < 0$. The coefficient values u_{1i} and u_{2i} are generated randomly. The demands at the buses are allowed to vary between 80% and 120% of the nominal demand specified in the test cases available in MATPOWER [53]. Table 11 summarizes the performance statistics of Algorithm 1 versus a subgradient algorithm with diminishing stepsizes averaged over 10 different runs in which the cost and demands are varied. The subgradient algorithm hits the iteration limit of 100000 on most instances, whereas Algorithm 1 solves the problems in very few iterations with modest function evaluation counts. Further, Algorithm 1 is 2-3 orders of magnitude faster than the subgradient algorithm.

6.2 Nonlinear network flows

Networks are used for modeling the transportation of energy, materials, people, information, etc. between source and demand locations. The term *nonlinear networks* refers to the class of networks in which (i) flow through the network is driven by a potential and (ii) potential loss on an edge in the network is a nonlinear function of the flow rate through that edge. Nonlinear network analysis is the problem of determining potential and flows that result in a network for a particular choice of resistances on

each edge. Raghunathan [43] presented a strictly convex formulation for a nonlinear network analysis problem. Given a network with a set of nodes $\mathcal{N}$, set of source nodes $\mathcal{N}^{\text{src}}$ fixed at a potential π_i^{src}, set of edges $\mathcal{E}$, set of flow demands q_i^{dem} for $i \in \mathcal{N} \setminus \mathcal{N}^{\text{src}}$, set of resistance choices r_e for each edge $e \in \mathcal{E}$, potential loss function $\theta(\cdot; r) : \mathbb{R}_+ \to \mathbb{R}_+$ for a choice of resistance r, and with $\Theta(q; r) := \int_0^q \theta(q'; r) dq'$, the convex formulation from [43] can be expressed as

$$\min \sum_{e \in \mathcal{E}} (\Theta(q_e^+; r_e) + \Theta(q_e^-; r_e)) L_e - \sum_{i \in \mathcal{N}^{\text{src}}} \sum_{e=(i,j)} \pi_i^{\text{src}} (q_e^+ - q_e^-) \tag{6.4a}$$

$$\text{s.t.} \sum_{e=(j,i)} (q_e^+ - q_e^-) - \sum_{e=(i,j)} (q_e^+ - q_e^-) = q_i^{\text{dem}} \quad \text{for all} \quad i \in \mathcal{N} \setminus \mathcal{N}^{\text{src}} \tag{6.4b}$$

$$0 \leq (q_e^+, q_e^-) \quad \text{for all} \quad e \in \mathcal{E}. \tag{6.4c}$$

Typically, the potential loss function takes the form $\theta(q; r) = k_r q^a$ for some $k_r > 0$ and $a \geq 1$. For all $a \geq 1$, the function $\Theta(\cdot; r)$ is strictly convex and, hence, (6.4) is an instance of a strictly convex optimization problem.

Dualizing the equality constraint (6.4b) using multipliers λ_i for $i \in \mathcal{N} \setminus \mathcal{N}^{\text{src}}$, one obtain an implicit equality constraint system as

$$\sum_{e=(j,i)} (q_e^+(\lambda) - q_e^-(\lambda)) - \sum_{e=(i,j)} (q_e^+(\lambda) - q_e^-(\lambda)) = q_i^{\text{dem}} \quad \text{for all} \quad i \in \mathcal{N} \setminus \mathcal{N}^{\text{src}}$$

where $(q_e^+(\lambda), q_e^-(\lambda))$ for each $e = (i, j)$ solves

$$\min_{(q^+, q^-) \geq 0} (\Theta(q^+; r_e) + \Theta(q^-; r_e)) L_e + (\lambda_j - \lambda_i - b_e \pi_i^{\text{src}})(q^+ - q^-),$$

with $b_e = 1$ if $i \in \mathcal{N}^{\text{src}}$ and $b_e = 0$ otherwise.

If $\theta(q; r) = k_r q^2$, then the subproblems on the edges are quadratic optimization problems of the form (1.2_i). The semismooth equation approach presented in this paper can then be readily applied to obtain the superior convergence observed in Table 12. The instances in Table 12 correspond to the water network design problems available at http://or.dei.unibo.it/instances/water-network-design. The description of the data format can be found in [5]. Though the potential loss function is specified for $a = 1.857$, we also consider the case of $a = 1$. For each instance, the diameters for the pipes are randomly assigned from the available set of pipe diameters in the input file. The reported results on number of iterations and function evaluations are averaged over 10 different runs in which the diameter assignments are varied. The CPU times for Algorithm 1 are also provided, demonstrating that solver is very fast. Though the approach has been described here by means of a fine-grained decomposition by edges in the network, the approach can be readily extended to decomposition of a large network into sub-networks.

Table 12 Results for water network analysis problems

Name	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{N}^{\text{src}}\rvert$	$a = 1$			$a = 1.857$		
				Av. it.	Av. fcn.	Av. CPU (s)	Av. it.	Av. fcn.	Av. CPU (s)
`hanoi`	32	34	1	2.3	3.3	0.02	39.9	50.6	0.22
`foss_iron`	37	58	1	1.8	2.8	0.02	77.8	98.2	0.60
`foss_poly_0`	37	58	1	2.5	3.5	0.02	68.0	81.7	0.49
`pescara`	71	99	3	2.8	3.8	0.05	79.9	124.3	1.12
`modena`	272	317	4	2.7	3.7	0.49	83.4	129.7	11.77

6.3 Two-stage stochastic optimization problems

Decision-making under uncertainty pervades almost all areas of planning and operation. Such problems fall under the class of stochastic programming [4]. Two-stage stochastic programs are problems in which the decision-maker is required to make a *first-stage* decision *here-and-now*, agnostic to the realization of the uncertain parameters. Once the values of these uncertain parameters manifest themselves, the decision-maker has the opportunity in the *second-stage* to take corrective action. Such problems can generally be formulated as

$$\min_{(y,y^k)} \; f(y) + \sum_{k=1}^{N_r} p^k f^k(y^k) \tag{6.5a}$$

$$\text{s.t. } Ty + T^k y^k \le t^k \;\; \text{for all} \;\; k \in \{1, \ldots, N_r\} \tag{6.5b}$$

$$y \in \mathcal{Y} \;\; \text{and} \;\; y^k \in \mathcal{Y}^k \;\; \text{for all} \;\; k \in \{1, \ldots, N_r\}, \tag{6.5c}$$

where N_r denotes the number of uncertainty scenarios and p^k is the probability of scenario k. The variable y denotes the *first-stage* decisions while y^k denotes the second-stage decision variable for each scenario k. The sets $\mathcal{Y}$ and $\mathcal{Y}^k$ are typically polyhedral, defined by linear equalities and inequalities. The constraint (6.5b) couples the first-stage and the second-stages decisions. (For ease of exposition, we have restricted our attention to a formulation with inequality constraints only, though formulations with equality constraints can also be handled in a straightforward manner.) As before, dualization of (6.5b) results in an optimization problem

$$\min_{(y,y^k)} \; f(y) + \sum_{k=1}^{N_r} p^k f^k(y^k) + \sum_{k=1}^{N_r} (\lambda^k)(Ty + T^k y^k - t^k) \tag{6.6a}$$

$$\text{s.t. } y \in \mathcal{Y} \;\; \text{and} \;\; y^k \in \mathcal{Y}^k \;\; \text{for all} \;\; k \in \{1, \ldots, N_r\}, \tag{6.6b}$$

which is separable in the first-stage and second-stage variables. When the functions f and $\{f_i\}$ are quadratic, the subproblems are precisely of the form (1.2_i). The approach presented in this paper can be readily applied to the ICP

Table 13 Results for the newsvendor problem with quadratic costs and prices

N_r	Avg. #Iters.	Avg. # Fcn.	Avg. CPU (s)	Obj.	IPOPT Obj.
10	164.2	392.9	0.41	-2.24 ± 1.47	-2.24 ± 1.47
100	186.2	2469.2	3.21	-1.55 ± 0.38	-1.54 ± 0.38

$$0 \leq \lambda^k \perp Ty(\lambda^1, \ldots, \lambda^{N_r}) + T^k y^k(\lambda^k) \leq t^k \quad \text{for all} \quad k \in \{1, \ldots, N_r\}.$$

Examples of such instances include the stock-ordering problem that arises in a number applications such as the newsvendor problem [4], and contaminant source inversion in water networks [26].

Let us consider the newvendor problem with uncertain demand and quadratic costs and prices. The first-stage quantity is the order-quantity y whose cost is given as $f(y) = 0.01y^2 + y$ and the price for the quantity that is sold is $f^k(y^k) = -0.01(y^k)^2 - 1.2y^k$. Thus, the objective function (6.5a) is nonconvex. The coupling constraints (6.5b) are $y^k \leq y$ which limits the sale quantity in each scenario to be smaller than the order quantity. Further, $\mathcal{Y} = [0, 150]$ and $\mathcal{Y}^k = [0, d^k]$ where d^k is the demand in that scenario. The demand is assumed to be uniformly distributed in $(0, 100)$ and, hence, $p^k = 1/N_r$. Table 13 presents results for scenarios with $N_r = 10$ and $N_r = 100$. The proximal penalty parameter μ^0 is chosen as $0.1/N_r$, ensuring that the resulting problems are all sufficiently strictly convex. The algorithm successfully solved all 100 problems in each of the cases. The reported iterations and function evaluations are averaged over 100 runs in which the demands are varied. The objective values reported as the mean and standard deviation over the 100 runs. We also report the mean and standard deviation of the objective from running IPOPT [49] on the same instances.

7 Conclusion

We have presented algorithms for solving an interesting class of quadratic optimization problems (QPs). The distinguishing feature of the problems in this class are that they involve groups of variables that can be considered separately except for a set of linking constraints. The methods that we have presented exploit this nearly separability, but in a manner that still allows fast local convergence guarantees under reasonable assumptions. Our first algorithm is applicable when solving strictly convex QPs while our second algorithm incorporates proximal point methodologies to handle a lack of strict convexity or even indefiniteness. Numerical experiments with randomly generated problems illustrate that our approaches are generally very efficient and reliable, either with merit function or filter globalization techniques. We have also demonstrated superior performance of our methods compared to a classical subgradient approach when applied to solve real-world problems related to finding competitive equilibrium in electricity markets, nonlinear network analysis, and two-stage stochastic optimization.

Appendix: Stationarity, differentiability, regularity, and (1.1) versus (1.5)

In this appendix, we discuss stationarity conditions for the QPs (1.1) and (1.2_i), differentiability properties of the subproblem solutions $x_i(\cdot)$ with respect to λ, regularity properties of implicit complementarity problem (1.5), and relationships between solutions of (1.1) and (1.5).

Stationarity conditions for (1.1) and (1.2_i)

First-order stationarity conditions for the QPs (1.1) and (1.2_i) can be derived from standard theory on Karush-Kuhn-Tucker conditions; e.g., see [2]. A point $\boldsymbol{x}^*$ is a first-order KKT point of (1.1) if there exist multipliers $(\lambda^*, \boldsymbol{\xi}^*)$ such that

$$\begin{aligned} \boldsymbol{Q}\boldsymbol{x}^* + \boldsymbol{q} + \boldsymbol{A}^T\lambda^* + \boldsymbol{B}^T\boldsymbol{\xi}^* &= 0 \\ 0 \le b - \boldsymbol{A}\boldsymbol{x}^* \perp \lambda^* &\ge 0 \\ 0 \le \boldsymbol{c} - \boldsymbol{B}\boldsymbol{x}^* \perp \boldsymbol{\xi}^* &\ge 0. \end{aligned} \tag{7.1}$$

Similarly, a point x_i^* is a first-order KKT point of (1.2_i) if there exists multipliers ξ_i^* such that

$$\begin{aligned} Q_i x_i^* + q_i + A_i^T\lambda + B_i^T\xi_i^* &= 0 \\ 0 \le c_i - B_i x_i^* \perp \xi_i^* &\ge 0. \end{aligned} \tag{7.2}$$

Second-order KKT points are defined as those at which the first-order KKT conditions can be satisfied and at which a curvature condition on the Hessian holds over a critical cone defined by the corresponding active constraints. To formally state these second-order conditions, let us first define the following: Given a primal point $\boldsymbol{x} \in \mathbb{R}^{nN}$, a Lagrange multiplier vector for the coupling constraint $\lambda \in \mathbb{R}^m$, and Lagrange multipliers for the subproblem constraints $\boldsymbol{\xi} := (\xi_1, \ldots, \xi_N) \in \mathbb{R}^{pN}$, let subsets of the indices for the coupling constraints be

$$\begin{aligned} \alpha(\boldsymbol{x}, \lambda) &:= \{j : (\boldsymbol{A}\boldsymbol{x} - b)_{[j]} = 0 < \lambda_{[j]}\}, \\ \beta(\boldsymbol{x}, \lambda) &:= \{j : (\boldsymbol{A}\boldsymbol{x} - b)_{[j]} = 0 = \lambda_{[j]}\}, \\ \text{and } \gamma(\boldsymbol{x}, \lambda) &:= \{j : (\boldsymbol{A}\boldsymbol{x} - b)_{[j]} < 0 = \lambda_{[j]}\}, \end{aligned} \tag{7.3}$$

and, for all $i \in \{1, \ldots, N\}$, let subsets of the indices for the inequalities in $\mathcal{X}_i$ be

$$\begin{aligned} \alpha_i(x_i, \xi_i) &:= \{j : (B_i x_i - c_i)_{[j]} = 0 < \xi_{i[j]}\}, \\ \beta_i(x_i, \xi_i) &:= \{j : (B_i x_i - c_i)_{[j]} = 0 = \xi_{i[j]}\}, \\ \text{and } \gamma_i(x_i, \xi_i) &:= \{j : (B_i x_i - c_i)_{[j]} < 0 = \xi_{i[j]}\}. \end{aligned} \tag{7.4}$$

Using these definitions, given $(\boldsymbol{x}^*, \lambda^*, \boldsymbol{\xi}^*)$ satisfying (7.1), we define the critical cone

$$\mathcal{T}(\boldsymbol{x}^*, \lambda^*, \boldsymbol{\xi}^*) := \left\{ \boldsymbol{d} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix} \in \mathbb{R}^{nN} : \left\{ \begin{array}{c} \displaystyle\sum_{i=1}^{N} A_{i[\alpha\cdot]} d_i = 0, \ \sum_{i=1}^{N} A_{i[\beta\cdot]} d_i \leq 0, \\ \text{and } B_{i[\alpha_i\cdot]} d_i = 0, \ B_{i[\beta_i\cdot]} d_i \leq 0 \\ \text{for all } i \in \{1, \ldots, N\} \end{array} \right\} \right\}. \tag{7.5}$$

Similarly, corresponding to a point (x_i^*, ξ_i^*) satisfying (7.2), we define the critical cone

$$\mathcal{T}_i(x_i^*, \xi_i^*) := \left\{ d_i \in \mathbb{R}^n : B_{i[\alpha_i\cdot]} d_i = 0 \text{ and } B_{i[\beta_i\cdot]} d_i \leq 0 \right\}, \tag{7.6}$$

We may now state that $\boldsymbol{x}^*$ is a second-order KKT point of (1.1) if there exist multipliers $(\lambda^*, \boldsymbol{\xi}^*)$ such that (7.1) holds and

$$\boldsymbol{d}^T \boldsymbol{Q} \boldsymbol{d} \geq 0 \text{ for all } \boldsymbol{d} \in \mathcal{T}(\boldsymbol{x}^*, \boldsymbol{\xi}^*, \lambda^*), \tag{7.7}$$

and x_i^* is a second-order KKT point of (1.2_i) if there exists ξ_i^* such that (7.2) holds and

$$d_i^T Q_i d_i \geq 0 \text{ for all } d_i \in \mathcal{T}_i(x_i^*, \xi_i^*). \tag{7.8}$$

If these curvature conditions hold strictly for nonzero elements in the critical cone, then one can conclude that the corresponding point is a strict local minimizer. In particular, $\boldsymbol{x}^*$ is a strict local minimizer of (1.1) if there exist multipliers $(\lambda^*, \boldsymbol{\xi}^*)$ such that (7.1) holds and

$$\boldsymbol{d}^T \boldsymbol{Q} \boldsymbol{d} > 0 \text{ for all } \boldsymbol{d} \in \mathcal{T}(\boldsymbol{x}^*, \boldsymbol{\xi}^*, \lambda^*) \setminus \{0\}, \tag{7.9}$$

while x_i^* is a strict local minimizer of (1.2_i) if there exists ξ_i^* such that (7.2) holds and

$$d_i^T Q_i d_i > 0 \text{ for all } d_i \in \mathcal{T}_i(x_i^*, \xi_i^*) \setminus \{0\}. \tag{7.10}$$

Overall, one can observe that if (1.1) (resp. (1.2_i)) is convex in that $\boldsymbol{Q} \succeq 0$ (resp. $Q_i \succeq 0$), then any first-order KKT point is a second-order KKT point, from which convexity implies that it is a global minimizer of (1.1) (resp. (1.2_i)). Similarly, if (1.1) (resp. (1.2_i)) is strictly convex in that $\boldsymbol{Q} \succ 0$ (resp. $Q_i \succ 0$), then any first-order KKT point is a strict local minimizer, from which convexity implies that it is a strict global minimizer of (1.1) (resp. (1.2_i)).

Differentiability concepts of locally Lipschitz functions

Let $\Phi : \mathbb{R}^m \to \mathbb{R}^m$ be locally Lipschitz. Then, by Rademacher's theorem, Φ is differentiable almost everywhere in $\mathbb{R}^m$. If D_Φ denotes the set of $\lambda \in \mathbb{R}^m$ at which Φ is differentiable, then the *B-subdifferential* of Φ at λ can be defined (see [25]) as

$$\partial_B \Phi(\lambda) := \left\{ H \in \mathbb{R}^{m \times m} : H = \lim_{\lambda^k \in D_\Phi, \lambda^k \to \lambda} \nabla \Phi(\lambda^k) \right\}. \tag{7.11}$$

The convex hull of this set is the *generalized Jacobian* of Φ at λ [25], written as

$$\partial\Phi := \text{conv } \partial_B\Phi(\lambda).$$

Such a function is *semismooth* [25,41] at λ if it is directionally differentiable at λ and

$$H\delta - \Phi'(\lambda;\delta) = o(\|\delta\|) \text{ for any } \delta \to 0 \text{ and } H \in \partial\Phi(\lambda+\delta).$$

Further, Φ is *strongly semismooth* at λ if

$$H\delta - \Phi'(\lambda;\delta) = o(\|\delta\|^2) \text{ for any } \delta \to 0 \text{ and } H \in \partial\Phi(\lambda+\delta).$$

Regularity properties of solutions of (1.5)

A square matrix $M \in \mathbb{R}^{m\times m}$ is said to be a P-matrix if all of its principal minors are positive [42, Def. 3.3.1]. If $M \in \mathbb{R}^{m\times m}$ is a symmetric P-matrix, then M is positive definite. This property is of importance for matrices arising in the following definition, which relates to regularity of solutions for (1.5); see [47, Def. 2.1].

Definition 7.1 (b- and R-regular solutions of (1.5)) Suppose that $F \in C^1$. Let λ^* be a solution of (1.5) and let α, β, and γ be defined as in (7.3). Then, λ^* is a

- b-regular solution of (1.5) if $\nabla F_{[\sigma\sigma]}(\lambda^*)$ is nonsingular whenever $\alpha \subseteq \sigma \subseteq \alpha\cup\beta$;
- R-regular solution of (1.5) if $\nabla F_{[\alpha\alpha]}(\lambda^*)$ is nonsingular and its Schur complement w.r.t.

$$\begin{bmatrix} \nabla F_{[\alpha\alpha]}(\lambda^*) & \nabla F_{[\alpha\beta]}(\lambda^*) \\ \nabla F_{[\beta\alpha]}(\lambda^*) & \nabla F_{[\beta\beta]}(\lambda^*) \end{bmatrix},$$

namely

$$(\nabla F_{[\beta\beta]}/\nabla F_{[\alpha\alpha]})(\lambda^*) := \nabla F_{[\beta\beta]}(\lambda^*) - \nabla F_{[\beta\alpha]}(\lambda^*)(\nabla F_{[\alpha\alpha]}(\lambda^*))^{-1}\nabla F_{[\alpha\beta]}(\lambda^*) \tag{7.12}$$

is a P-matrix.

Differentiability properties of $x_i(\lambda)$

The differentiability properties of a first-order KKT point $x_i^*(\lambda)$ of (1.2_i) and its corresponding objective value as functions of λ follow from the theory of sensitivity analysis of parametric nonlinear optimization problems. The study of solution and objective sensitivity to variations in parameters in such problems has been studied extensively since the 1970's. Since the constraints of (1.1) and (1.2_i) are affine, we have that the conditions in Appendix 1 are necessary for stationarity with the last conditions sufficient for (local) optimality. However, for certain results below, we rely on additional regularity conditions for (1.2_i).

We begin by stating two types of regularity conditions on the constraints in (1.2_i).

Definition 7.2 ((1.2$_i$)-**LICQ**) If $B_{i[(\alpha_i \cup \beta_i)\cdot]}$ has full row rank (with (α_i, β_i) from (7.4)), then the Linear Independence Constraint Qualification (LICQ) for (1.2$_i$) holds at $x_i \in \mathcal{X}_i$.

Definition 7.3 ((1.2$_i$)-**SCQ**) If there exists $x_i \in \mathbb{R}^n$ such that $B_i x_i < c_i$, then the Slater Constraint Qualification (SCQ) holds for (1.2$_i$).

If a problem has a convex feasible region, as does (1.2$_i$), the SCQ is equivalent to the Mangasarian-Fromovitz Constraint Qualification (MFCQ) holding at every feasible point. If the LICQ holds at any feasible point, then the SCQ holds. For a local minimizer $x_i^*(\lambda)$ of (1.2$_i$), the LICQ implies uniqueness of the optimal Lagrange multiplier vector $\xi_i^*(\lambda)$; however, the SCQ only implies that the set of optimal multipliers at $x_i^*(\lambda)$, call it $\Xi_i(x_i^*(\lambda))$, is bounded.

Now, given a first-order KKT point of (1.2$_i$) and the critical cone

$$\mathcal{T}_i^{sc}(x_i^*, \xi_i^*) := \{d_i \in \mathbb{R}^n : B_{i[\alpha_i \cdot]} d_i = 0\} \supseteq \mathcal{T}_i(x_i^*, \xi_i^*) \tag{7.13}$$

with α_i defined in (7.4), we define the following.

Definition 7.4 ((1.2$_i$)-**SSOSC**) If (x_i^*, ξ_i^*) is a first-order KKT point of (1.2$_i$), then the strong second-order sufficient condition (SSOSC) for (1.2$_i$) holds at (x_i^*, ξ_i^*) if

$$d_i^T Q_i d_i > 0 \ \text{ for all } \ d_i \in \mathcal{T}_i^{sc}(x_i^*, \xi_i^*). \tag{7.14}$$

An early result on differentiability of $x_i(\cdot)$ for our purposes can be derived from the work of Fiacco [13, Thm. 2.1]. We state it as the following lemma.

Lemma 7.1 *Given $\bar{\lambda} \in \mathbb{R}^m$, suppose that $x_i^*(\bar{\lambda})$ is a strict local solution of (1.2$_i$) with $\lambda = \bar{\lambda}$ such that with the corresponding Lagrange multiplier $\xi_i^*(\bar{\lambda})$ the following hold:*

- *$\beta_i(x_i^*(\bar{\lambda}), \xi_i^*(\bar{\lambda})) = \emptyset$,*
- *the (1.2$_i$)-LICQ holds at $x_i^*(\bar{\lambda})$, and*
- *the (1.2$_i$)-SSOSC holds at $(x_i^*(\bar{\lambda}), \xi_i^*(\bar{\lambda}))$.*

Then, there exist open neighborhoods U_i, V_i, and W_i centered at x_i^*, ξ_i^*, and $\bar{\lambda}$, respectively, and functions $x_i(\cdot) : W_i \to U_i \subseteq \mathcal{X}_i$ and $\xi_i(\cdot) : W_i \to V_i$ such that:

(i) $x_i(\cdot)$ and $\xi_i(\cdot)$ are C^1 functions of $\lambda \in W_i$;
(ii) for each $\lambda \in W_i$, the point $x_i(\lambda)$ is a strict local solution of (1.2$_i$) with Lagrange multiplier $\xi_i(\lambda)$; and
(iii) for each $\lambda \in W_i$, the Jacobian of $(x_i(\cdot), \xi_i(\cdot))$ at λ is given by

$$\begin{bmatrix} Q_i & B_{i[\alpha_i \cdot]}^T \\ B_{i[\alpha_i \cdot]} & 0 \end{bmatrix} \begin{bmatrix} \nabla x_i(\lambda) \\ \nabla \xi_{i[\alpha_i]}(\lambda) \end{bmatrix} = \begin{bmatrix} -A_i^T \\ 0 \end{bmatrix}. \tag{7.15}$$

The following result, due to Ralph and Dempe [6, Thms. 1 & 2 and Cor. 4(2)], represents an extension of this result in which the strict complementarity requirement (i.e., $\beta_i = \emptyset$) is relaxed and the LICQ is replaced by the (weaker) MFCQ along with

the *Constant Rank Constraint Qualification* (CRCQ). The CRCQ holds automatically for (1.1), so we do not state it explicitly. Moreover, the MFCQ is equivalent to the SCQ for (1.1), so we refer to the latter.

Lemma 7.2 *Given $\bar{\lambda} \in \mathbb{R}^m$, suppose that $x_i^*(\bar{\lambda})$ is a strict local solution of (1.2$_i$) with $\lambda = \bar{\lambda}$ such that the following hold:*

- *the (1.2$_i$)-SCQ holds and*
- *the (1.2$_i$)-SSOSC holds for all $\xi_i \in \Xi_i(x_i^*(\bar{\lambda}))$.*

Then, for all $i \in \{1, \ldots, N\}$, there exist open neighborhoods U_i and W_i centered at $x_i^*(\bar{\lambda})$ and $\bar{\lambda}$, respectively, and a function $x_i(\cdot) : W_i \to U_i \subseteq \mathcal{X}_i$ such that:

(i) $x_i(\cdot)$ is a PC^1 function of $\lambda \in W_i$;
(ii) for each $\lambda \in W_i$, the point $x_i(\lambda)$ is a strict local solution of (1.2$_i$); and
(iii) the directional derivative $x_i'(\lambda; \cdot)$ is a piecewise linear function such that for each $\lambda \in W_i$, $d \in \mathbb{R}^m$, and $\xi_i \in \Xi_i(x_i(\lambda))$ the value $x_i'(\lambda; \delta)$ is given by the unique solution of

$$\min_{v} \ \tfrac{1}{2} v^T Q_i v + v^T A_i^T \delta \ \ s.t. \ \ B_{i[\alpha_i \cdot]} v = 0, \quad B_{i[\beta_i \cdot]} v \leq 0. \tag{7.16}$$

We remark that the conclusion of Lemma 7.2, namely, that $x_i(\cdot)$ is a PC^1 function of $\lambda \in W_i$, implies that $x_i(\cdot)$ is locally Lipschitz and directionally differentiable.

Relationship between the QP and the ICP formulations

A solution to (1.5) easily yields a first-order KKT point of (1.1). In particular, we have the following result whose proof follows simply by combining (7.2) with (1.5).

Lemma 7.3 *Suppose λ^* solves (1.5) with x_i^* denoting a first-order KKT point of (1.2$_i$) for $\lambda = \lambda^*$ for all $i \in \{1, \ldots, N\}$. Then, $\boldsymbol{x}^*$ is a first-order KKT point of (1.1).*

Similarly, a solution to (1.5) can also yield a second-order KKT point of (1.1). For this claim, we have the following result whose proof follows by combining (7.8) and the fact that the Cartesian product of the $\mathcal{T}_i$'s represents a superset of $\mathcal{T}$.

Lemma 7.4 *Suppose λ^* solves (1.5) with x_i^* denoting a second-order KKT point of (1.2$_i$) for $\lambda = \lambda^*$ for all $i \in \{1, \ldots, N\}$. Then, $\boldsymbol{x}^*$ is a second-order KKT point of (1.1).*

The reverse implications are the subjects of the following lemmas, the proofs of which are straightforward; hence, we state them without proof.

Lemma 7.5 *Suppose that $\boldsymbol{x}^*$ is a first-order KKT point of (1.1) in that there exist multipliers $(\lambda^*, \boldsymbol{\xi}^*)$ such that (7.1) holds. Then, the following hold:*

(i) for each $i \in \{1, \ldots, N\}$, x_i^* is a first-order KKT point for (1.2$_i$) for $\lambda = \lambda^*$, and
(ii) λ^* solves (1.5).

Lemma 7.6 *Suppose that $\boldsymbol{x}^*$ is a second-order KKT point of (1.1) in that there exist multipliers $(\boldsymbol{\xi}^*, \lambda^*)$ such that (7.1) and (7.7) hold. Moreover, suppose that*

$$\boldsymbol{d}^T \boldsymbol{Q} \boldsymbol{d} \geq 0 \text{ for all } \boldsymbol{d} = (d_1, \ldots, d_N) \in \prod_{i=1}^{N} \mathcal{T}_i(x_i^*, \xi_i^*). \tag{7.17}$$

Then, the following hold:

(i) for each $i \in \{1, \ldots, N\}$, x_i^* is a second-order KKT point for (1.2$_i$) for $\lambda = \lambda^*$, and
(ii) λ^* solves (1.5).

References

1. Bai, L., Raghunathan, A. U.: Semismooth equation approach to network utility maximization (NUM). In American Control Conference (ACC), pp. 4795–4801, (2013)
2. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
3. Bertsekas, D.P.: Convex Optimization Theory. Athena Scientific, Belmont (2009)
4. Birge, J.R., Louveaux, F.V.: Introduction to Stochastic Programming. Springer, New York (1997)
5. Bragalli, C., Ambrosio, C.D., Lee, J., Lodi, A., Toth, P.: On the optimal design of water distribution networks: a practical MINLP approach. Opt. Eng. **13**(2), 219–246 (2012)
6. Ralph, D., Dempe, S.: Directional derivatives of the solution of a parametric nonlinear program. Math. Prog. **70**, 159–172 (1995)
7. Deng, W., Lai, M.-J., Peng, Z., Yin, W.: Parallel Multi-Block ADMM with $o(1/k)$ Convergence. Technical report, arXiv:1312.3040 (2014)
8. Dinh, Q.T., Necoara, I., Diehl, M.: Path-following gradient-based decomposition algorithms for separable convex optimization. J. Global Optim. **59**(1), 59–80 (2014)
9. Dinh, Q.T., Necoara, I., Savorgnan, C., Diehl, M.: An inexact perturbed path-following method for lagrangian decomposition in large-scale separable convex optimization. SIAM J. Optim. **23**(1), 95–125 (2013)
10. Dinh, Q.T., Savorgnan, C., Diehl, M.: Combining lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. Comput. Optim. Appl. **55**(1), 75–111 (2013)
11. Facchinei, F., Fischer, A., Kanzow, C.: Regularity properties of a semismooth reformulation of variational inequalities. SIAM J. Optim. **8**, 850–869 (1998)
12. Facchinei, F., Soares, J.: A new merit function for nonlinear complementarity problems and a related algorithm. SIAM J. Optim. **7**, 225–247 (1997)
13. Fiacco, A.V.: Sensitivity analysis for nonlinear programming using penalty methods. Math. Program. **10**, 287–311 (1976)
14. Fiacco, A.V., McCormick, G.P.: Nonlinear programming: sequential unconstrained minimization techniques. Class. Appl. Math. Soc. Ind. Appl. Math. (1987)
15. Fischer, A.: A special Newton-type optimization method. Optimization **24**, 269–284 (1992)
16. Fletcher, R., Gould, N.I.M., Leyffer, S., Toint, PhL, Wächter, A.: Global convergence of trust-region SQP-filter algorithms for nonlinear programming. SIAM J. Opt. **13**, 635–659 (2002)
17. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Math. Program. **91**, 239–269 (2002)
18. Fletcher, R., Leyffer, S.: Filter-type algorithms for solving systems of algebraic equations and inequalities. In: High-Performance Algorithms and Software in Nonlinear Optimization, pp. 259–278. Kluwer, Dordrecht, The Netherlands (2003)
19. Fletcher, R., Leyffer, S., Toint, PhL: On the global convergence of a filter-SQP algorithm. SIAM J. Opt. **13**, 44–59 (2002)
20. Frasch, J.V., Sager, S., Diehl, M.: A parallel quadratic programming method for dynamic optimization problems. Math. Prog. Comput. **7**(3), 289–329 (2015)

21. Gondzio, J., Grothey, A.: Parallel interior-point solver for structured quadratic programs: Application to financial planning problems. Ann. Oper. Res. **152**(1), 319–339 (2007)
22. Gould, N.I.M., Leyffer, S., Toint, PhL: A multidimensional filter algorithm for nonlinear equations and nonlinear least squares. SIAM J. Opt. **15**, 17–38 (2004)
23. Jiang, H., Qi, L.: A new nonsmooth equations approach to nonlinear complementarity problems. SIAM J Control Opt. **35**, 178–193 (1997)
24. Kanzow, C., Petra, S.: Projected filter trust region methods for a semismooth least squares formulation of mixed complementarity problems. Opt. Methods Softw. **22**(5), 713–735 (2007)
25. Qi, L., Sun, J.: A nonsmooth version of Newton's method. Math. Progr. **58**, 353–368 (1993)
26. Laird, C.D., Biegler, L.T.: Large-scale nonlinear programming for multi-scenario optimization. In: Bock, H.G., Kostina, E., Phu, H.X., Ranacher, R. (eds.) Modeling, simulation and optimization of complex processes, pp. 323–336. Springer, Berlin (2008)
27. De Luca, T., Facchinei, F., Kanzow, C.: A semismooth equation approach to the solution of nonlinear complementarity problems. Math. Progr. **75**, 407–439 (1996)
28. Mas-Colell, A., Whinston, M.D., Green, J.R.: Microeconomic Theory. Oxford University Press, Oxford (1991)
29. Munson, T.S., Facchinei, F., Ferris, M., Fischer, A., Kanzow, C.: The Semismooth algorithm for large scale complementarity problems. J. Comput. **13**, 294–311 (2001)
30. Necoara, I., Patrascu, A.: Iteration complexity analysis of dual first-order methods for conic convex programming. Opt. Methods Softw. **31**(3), 645–678 (2016)
31. Necoara, I., Suykens, J.A.K.: Application of a smoothing technique to decomposition in convex optimization. IEEE Trans. Autom. Control **53**(11), 2674–2679 (2008)
32. Necoara, I., Suykens, J.A.K.: Application of a smoothing technique to decomposition in convex optimization. IEEE Trans. Autom. Control **53**(11), 2674–2679 (2008)
33. Necoara, I., Suykens, J.A.K.: Interior-point lagrangian decomposition method for separable convex optimization. J. Optim. Theory Appl. **143**(3), 567–588 (2009)
34. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer, Boston (2004)
35. Nesterov, Y.: Excessive gap technique in nonsmooth convex minimization. SIAM J Opt. **16**(1), 235–249 (2005)
36. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Progr. (A) **103**(1), 127–152 (2005)
37. Nesterov, Y., Nemirovskii, A.: Interior Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics (SIAM Studies in Applied Mathematics), Philadelphia (1994)
38. Neumaier, A.: MINQ—General Definite and Bound Constrained Indefinite Quadratic Programming (1998)
39. Nie, Pu-yan, Lai, Ming yong, Zhu, Shu jin, Zhang, Pei ai: A line search filter approach for the system of nonlinear equations. Comput. Math. Appl. **55**(9), 2134–2141 (2008)
40. Qi, L.: Convergence analysis of some algorithms for solving nonsmooth equations. Math. Oper. Res. **51**, 101–131 (1991)
41. Mifflin, R.: Semismooth and semiconvex functions in constrained optimization. SIAM J. Control Opt. **15**(6), 959–972 (1977)
42. Cottle, R.W., Pang, J.-S., Stone, R.E.: The Linear Complementarity Problem. Classics in Applied Mathematics. SIAM, Philadelphia (2009)
43. Raghunathan, A.U.: Global optimization of nonlinear network design. SIAM J. Optim. **23**(1), 268–295 (2013)
44. Raghunathan, A. U., Curtis, F. E., Takaguchi, Y., Hashimoto, H.: Accelerating convergence to competitive equilibrium in electricity markets. In: IEEE Power & Energy Society General Meeting, PESGM2016-000221 (2016)
45. Rockafellar, R.T.: Monotone Operators and the Proximal Point Algorithm. SIAM J. Control Optim. **14**(5), 877–898 (1976)
46. Shor, N.Z.: Minimization Methods for Non-differentiable Functions. Springer, New York (1985)
47. De Luca, T., Facchinei, F., Kanzow, C.: A theoretical and numerical comparison of some semismooth algorithms for complementarity problems. Comput. Opt. Appl. **16**(2), 173–205 (2000)
48. Ulbrich, M., Ulbrich, S., Vicente, L.N.: A globally convergent primal-dual interior-point filter method for nonlinear programming. Math. Progr. **100**(2), 379–410 (2004)
49. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)

50. Wang, G., Negrete-Pincetic, M., Kowli, A., Shafieepoorfard, E., Meyn, S., Shanbhag, U.V.: Dynamic competitive equilibria in electricity markets. In: Chakrabortty, Aranya, Ili, Marija D. (eds.) Control and Optimization Methods for Electric Smart Grids, volume 3 of Power Electronics and Power Systems, pp. 35–62. Springer, New York (2012)
51. Wood, A.J., Wollenberg, B.F.: Power Generation Operation and Control, 2nd edn. Wiley, Hoboken (1996)
52. Zavala, V.M., Laird, C.D., Biegler, L.T.: Interior-point decomposition approaches for parallel solution of large-scale nonlinear parameter estimation problems. Chem. Eng. Sci. **63**(19), 4834–4845 (2008)
53. Zimmerman, R.D., Murillo-Sanchez, C.E., Thomas, R.J.: MATPOWER: steady-state operations, planning and analysis tools for power systems research and education. IEEE Trans. Power Syst. **26**(1), 12–19 (2011)