Noname manuscript No.
(will be inserted by the editor)

An Infeasible-Start Framework for Convex Quadratic Optimization, with Application to Constraint-Reduced Interior-Point and Other Methods

M. Paul Laiu · André L. Tits

Received: date / Accepted: date

Abstract A framework is proposed for solving general convex quadratic programs (CQPs) from an infeasible starting point by invoking an existing *feasible-start* algorithm tailored for *inequality*-constrained CQPs. The central tool is an exact penalty function scheme equipped with a penalty-parameter updating rule. The feasible-start algorithm merely has to satisfy certain general requirements, and so is the updating rule. Under mild assumptions, the framework is proved to converge on CQPs with both inequality and equality constraints and, at a negligible additional cost per iteration, produces an infeasibility certificate, together with a feasible point for an (approximately) ℓ_1-least relaxed feasible problem, when the given problem does not have a feasible solution. The framework is applied to a feasible-start constraint-reduced interior-point algorithm previously proved to be highly performant on problems with many more inequality constraints than variables ("imbalanced"). Numerical com-

This manuscript has been authored, in part, by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (`http://energy.gov/downloads/doe-public-access-plan`).

M. P. Laiu
Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA
E-mail: laiump@ornl.gov
This author's research was sponsored by the Office of Advanced Scientific Computing Research and performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725.

A. L. Tits
Department of Electrical and Computer Engineering & Institute for Systems Research, University of Maryland College Park, MD 20742 USA,
E-mail: andre@umd.edu

parison with popular codes (OSQP, qpOASES, MOSEK) is reported on both randomly generated problems and support-vector machine classifier training problems. The results show that the former typically outperforms the latter on imbalanced problems. Finally, application of the proposed infeasible-start framework to other feasible-start algorithms is briefly considered, and is tested on a simplex iteration.

Keywords convex quadratic/linear programming · infeasible start · infeasibility certificate · constraint reduction · interior point · simplex algorithm

Mathematics Subject Classification (2010) 65K05 · 90C05 · 90C06 · 90C20 · 90C51

1 Introduction

Consider a convex quadratic program (CQP)

$$\operatorname*{minimize}_{\mathbf{x}\in\mathbb{R}^n} \ \mathbf{f}(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{c}^T\mathbf{x} \text{ s.t. } \ \mathbf{A}\mathbf{x} \geq \mathbf{b},\ \mathbf{C}\mathbf{x} = \mathbf{d}, \tag{P}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of optimization variables and $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}$ the objective function, with $\mathbf{c} \in \mathbb{R}^n$ and with $\mathbf{H} \in \mathbb{R}^{n\times n}$ symmetric positive semi-definite; and where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{C} \in \mathbb{R}^{p\times n}$ and $\mathbf{d} \in \mathbb{R}^p$, with $n \geq p$ and $m + p > 0$. Here and elsewhere all inequalities ($\geq$, $\leq$, $>$, $<$) are meant component-wise.

Most available algorithms for solving such problems belong either to the interior-point family or to the simplex-like family. While the most popular interior-point algorithms do not require that an initial feasible point be provided, simplex algorithms do: such feasible points, when not readily available, are typically obtained by solving an auxiliary linear optimization problem ("phase 1"). Like simplex algorithms, recently proposed "constraint-reduced" interior-point algorithms, the latest of which (see, e.g., [28]) were observed to often largely outperform other approaches when the problem at hand is severely "imbalanced" (i.e., with most inequality constraints being inactive at the solution; e.g., $m \gg n - p$), do require a primal-feasible initial point. While a two-phase approach could again be employed here, an important drawback of that approach is that, in the first phase, the objective function is altogether ignored, leading to likely computational waste.[1,2]

[1] Note however that, in the context of feasible-direction methods for general nonlinear optimization problems, methods that craftily combine the two phases have been known for decades; see [34, 36].

[2] One option, not considered in the present paper, may be to resort to homogeneous self-dual (HSD) embedding methods, first proposed in [44] for the solution of linear optimization problems and later extended in [43] to that of linear complementarity problems (which include convex quadratic programming as a special case). We are unaware of studies combining HSD with a simplex algorithm or a constraint-reduction method. Development, analysis, and numerical testing of such combinations may be of much interest.

Motivated by the above, the aim of the present paper is to propose an exact-penalty-function-based framework that "transforms" an available primal-feasible algorithm into one that accommodates infeasible starts. The central component of the proposed framework is an augmented version of (P) that involves a vector of relaxation variables and an exact penalty function. Exact penalty functions (i.e., penalty functions for which a threshold exists—but is unknown at the outset—such that, when the penalty parameter exceeds that threshold, solutions of the penalized problem also solve the original constrained problem) have been extensively used for many decades in nonlinear optimization, especially since the seminal work of A.R. Conn [14]; see, e.g., [1, 13, 15, 21].

While the adaptive selection of the penalty parameter is often heuristic, in some contexts, authors have proposed formal adaptation rules that guarantee that an appropriate value of the parameter will eventually be obtained and will be kept for the remainder of the solution process; this goes back several decades (e.g., [30] as well as, in the context of augmented Lagrangian, [17,35]) and also includes more recent work such as [11, 39]. Finally, in the past two decades, exact penalty functions have been used successfully in the solution of mathematical programs with complementary constraints (MPCC), e.g., [16,32] and references therein. Here again, an appropriate, finite value of the penalty parameter is reached after finitely many iterations.

While, originally, the intent of exact penalty functions was to turn a constrained optimization problem into an unconstrained one, this tool has also been used to eliminate equality constraints when only an inequality-constraint algorithm is available, specifically, by replacing in each scalar equality the "=" sign with ''≥" and penalizing positive deviations from equality; see [30, 39]. More recently, in [22, 23], exact penalty functions have been used for yet another purpose: allowing algorithms that require a feasible initial point to accept infeasible initial points. As pointed out above, this is the focus of the present work.

Use of penalty functions in the solution of linear or convex-quadratic optimization problems has been scarcer than their use in nonlinear optimization, for obvious reasons: powerful methods have long existed (starting with the original simplex method for linear optimization seven decades ago) for the solution of such problems and there was no perceived need to resort to such tool. Exceptions include the use of an exact penalty method for warmstarting interior-point methods [6] and the "big M" approach (where the penalty parameter is "large" but fixed) considered in [25, section 4.3]. Also, as mentioned above, such need does arise in the context of constraint-reduced interior-point methods. An exact penalty function scheme was thus used in [22, 23] in the context of a specific constraint-reduced algorithm for inequality-constrained linear [23], then convex quadratic [22], optimization.

In the present paper, a rather general framework is proposed, analyzed, and numerically tested, for the solution of a CQP, starting from a possibly primal-infeasible point, that invokes an iteration of a rather arbitrary user-provided feasible-start CQP solver, referred to below as "base iteration". The key contributions are as follows. First the approach introduced in [22] is generalized to apply to a general class of feasible-start base iterations (as opposed to, merely, a specific version of a constraint-reduced scheme), and to offer broad freedom in the choice of a penalty-parameter updating rule; the base iteration and the updating rule are merely required to satisfy certain general specifications. Second, the framework is then extended to solve problems that include equality constraints without destroying any existing sparsity. Third, it is shown how at a negligible additional cost per iteration, when the CQP is primal-infeasible, a certificate of infeasibility can be produced.[3] Fourth, promising numerical results are obtained, with the algorithm of [28] as the base iteration, in comparison with those obtained with popular schemes. Finally, application of the proposed framework to other base iterations is considered, and the case of a revised primal simplex iteration is briefly tested.

The paper is organized as follows. In section 2, the framework is outlined, and requirements to be satisfied by the base iteration and the penalty-parameter updating rule are introduced. Section 3 is devoted to the convergence analysis, under the assumption that the requirements specified in section 2 are satisfied. For sake of simplicity of exposition, sections 2 and 3 deal with purely inequality-constrained problems, i.e., $p = 0$. Extension to the general problem is dealt with in section 4. In section 5, issuance of an infeasibility certificate in cases when (P) is infeasible is investigated. Section 6 introduces a penalty-parameter updating rule that satisfies the required specifications, discusses implementation details, and reports numerical results with the base iteration from [28] on randomly generated problems and support-vector machine training problems with comparison to popular optimization solvers. Application of the framework to other base iterations is considered in section 7. Concluding remarks are given in section 8.

The notation is mostly standard. In particular, consistent with the interior-point literature, given a vector $\mathbf{v}$, the associated matrix $\mathrm{diag}(v_i)$ is denoted by the corresponding capital letter V. We use $\|\cdot\|$ to denote an arbitrary norm, possibly different in each instance that it is being used; of course, $\|\cdot\|_\infty$, $\|\cdot\|_1$, and $\|\cdot\|_2$ are specific. The matrix norms are the respective induced norms. The Matlab notation ($[A\ B; C\ D]$, $[\mathbf{u};\mathbf{v}]$) is used for block matrices and vector concatenation. Also, $\mathbf{1}$ designates the vector of all 1s, of appropriate dimension, and $[\mathbf{u}]_+$ clips to zero all negative entries of vector $\mathbf{u}$.

Before proceeding, we state here two assumptions on problem (P), which will be in force throughout—with the exception of section 5, as duly noted there. Recall (e.g., [33, Propositions 2.1–2.2], [7, Proposition 6.3.1]) that if the dual of a CQP is feasible then the CQP is bounded, and that if the CQP is feasible and bounded then its dual is feasible and its optimal solution set is nonempty and bounded. The dual of (P) is given by

$$\underset{\mathbf{x}\in\mathbb{R}^n,\boldsymbol{\pi}\in\mathbb{R}^m,\boldsymbol{\omega}\in\mathbb{R}^p}{\text{maximize}} \; -\frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}+\boldsymbol{\pi}^T\mathbf{b}+\boldsymbol{\omega}^T\mathbf{d} \text{ s.t. } \mathbf{H}\mathbf{x}+\mathbf{c}-\mathbf{A}^T\boldsymbol{\pi}-\mathbf{C}^T\boldsymbol{\omega}=\mathbf{0}, \boldsymbol{\pi}\geq\mathbf{0}.$$

[3] Earlier algorithms that include such features include, in the context of sequential quadratic programming (in nonlinear optimization), that of [10].

Assumption 1 (P) *is strictly feasible and so is its dual, and* (P)*'s (nonempty) optimal solution set is bounded.*[4,5]

Assumption 2 $\mathbf{C}$ *has full (row) rank and* $[\mathbf{H};\mathbf{A};\mathbf{C}]$ *has full (column) rank.*

2 A Framework for Accommodating Infeasible Starts

2.1 General Idea

Suppose a feasible-start base iteration is available toward solving (P) with $p = 0$ and suppose moreover that applying such iteration repeatedly on (P) produces a sequence of feasible iterates that enjoys certain additional properties (to be specified in section 2.3 below). It is suggested in [28], in the context of a "constraint-reduced" primal-dual interior-point method that requires an initial primal-feasible point, that an extension to handle problems for which a primal-feasible initial point is *not* available can be constructed by involving the following penalized relaxed primal–dual pair,[6] for which a primal-feasible point $(\mathbf{x},\mathbf{z})$ is readily available:

$$\underset{\mathbf{x}\in\mathbb{R}^n,\,\mathbf{z}\in\mathbb{R}^m}{\text{minimize}}\ \mathbf{f}(\mathbf{x})+\varphi\mathbf{1}^T\mathbf{z} \quad \text{s.t.}\ \ \mathbf{A}\mathbf{x}+\mathbf{z}\geq\mathbf{b},\ \mathbf{z}\geq\mathbf{0}\,, \tag{1}$$

$$\underset{\mathbf{x}\in\mathbb{R}^n,\,\boldsymbol{\pi}\in\mathbb{R}^m,\boldsymbol{\xi}\in\mathbb{R}^m}{\text{maximize}}\ \boldsymbol{\psi}(\mathbf{x},\boldsymbol{\pi}) \quad \text{s.t.}\ \ \mathbf{H}\mathbf{x}+\mathbf{c}-\mathbf{A}^T\boldsymbol{\pi}=\mathbf{0}\,, \tag{2}$$

$$\boldsymbol{\pi}+\boldsymbol{\xi}=\varphi\mathbf{1},\,(\boldsymbol{\pi},\boldsymbol{\xi})\geq\mathbf{0},$$

with $\boldsymbol{\psi}(\mathbf{x},\boldsymbol{\pi}):=-\frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}+\mathbf{b}^T\boldsymbol{\pi}$ and $\varphi>0$ a penalty parameter, equivalently,

$$\underset{\mathbb{x}\in\mathbb{R}^{n+m}}{\text{minimize}}\ \mathbb{f}_\varphi(\mathbb{x}):=\frac{1}{2}\mathbb{x}^T\mathbb{H}\mathbb{x}+\mathbb{c}_\varphi^T\mathbb{x} \qquad \text{s.t.}\ \ \mathbb{A}\mathbb{x}\geq\mathbb{b}\,, \tag{P$_\varphi$}$$

$$\underset{(\mathbb{x},\mathbb{\lambda})\in\mathbb{R}^{(n+m)+2m}}{\text{maximize}}\ \mathbb{\psi}(\mathbb{x},\mathbb{\lambda}):=-\frac{1}{2}\mathbb{x}^T\mathbb{H}\mathbb{x}+\mathbb{b}^T\mathbb{\lambda} \quad \text{s.t.}\ \ \mathbb{H}\mathbb{x}+\mathbb{c}_\varphi-\mathbb{A}^T\mathbb{\lambda}=\mathbf{0}, \tag{D$_\varphi$}$$

$$\mathbb{\lambda}\geq\mathbf{0}\,,$$

where $\mathbb{x}:=[\mathbf{x};\mathbf{z}]$, $\mathbb{\lambda}:=[\boldsymbol{\pi};\boldsymbol{\xi}]$, $\mathbb{c}_\varphi:=[\mathbf{c};\varphi\mathbf{1}]$, $\mathbb{b}:=[\mathbf{b};\mathbf{0}]$,

$$\mathbb{H}:=\begin{bmatrix}\mathbf{H}&\mathbf{0}\\\mathbf{0}&\mathbf{0}\end{bmatrix}, \quad \text{and} \quad \mathbb{A}:=\begin{bmatrix}\mathbf{A}&\mathbf{I}\\\mathbf{0}&\mathbf{I}\end{bmatrix}. \tag{3}$$

[4] Only Proposition 1 invokes boundedness of the primal optimal solution set. It is not clear at this point whether such assumption is necessary indeed. In any case, if it is, it of course can be achieved by imposing large bounds to the components of $\mathbf{x}$.

[5] Note that, while, for primal linear optimization problems in *standard form*, dual strict feasibility implies boundedness of the primal optimal solution set (see, e.g., [42, Theorem 2.3]), it does not for problem (P).

[6] An ℓ_∞ penalty function can be substituted for this ℓ_1 penalty function with minor adjustments: see [22, 23] for details.

Necessary and sufficient conditions for $(\mathbf{x},\mathbf{z},\boldsymbol{\pi},\boldsymbol{\xi})$ to solve (P_φ)–(D_φ) are

$$\mathbf{H}\mathbf{x}+\mathbf{c}-\mathbf{A}^T\boldsymbol{\pi}=\mathbf{0},\ \boldsymbol{\pi}+\boldsymbol{\xi}=\varphi\mathbf{1},\ \mathbf{S}\boldsymbol{\pi}=\mathbf{0},\ \mathbf{Z}\boldsymbol{\xi}=\mathbf{0},\ (\mathbf{s},\mathbf{z},\boldsymbol{\pi},\boldsymbol{\xi})\geq\mathbf{0}, \tag{4}$$

where $\mathbf{s}:=\mathbf{A}\mathbf{x}+\mathbf{z}-\mathbf{b}$; equivalently,

$$\mathbb{H}\mathbb{x}+\mathbb{c}_\varphi-\mathbb{A}^T\mathbb{\lambda}=\mathbf{0},\ \mathbb{S}\mathbb{\lambda}=\mathbf{0},\ (\mathbb{s},\mathbb{\lambda})\geq\mathbf{0}, \tag{5}$$

where $\mathbb{s}:=[\mathbf{s};\mathbf{z}]$.

The rationale for introducing the penalized relaxed problem (P_φ) is as follows. The penalty function penalizes positive values of the components of $\mathbf{z}$ while the $\mathbf{z}\geq\mathbf{0}$ constraints in (1) prevent negative values. Hence, since the ℓ_1 penalty function is known to be exact, if the optimal solution set of (P) is nonempty (implying that the optimal solution set of (P_φ) is nonempty for φ sufficiently large), for φ above a certain threshold, every solution of (P_φ) will be of the form $(\mathbf{x}^*,\mathbf{0})$, with $\mathbf{x}^*$ a solution of (P) (see Lemma 2 below). On the other hand, for given φ, (P_φ) (with feasible initial $(\mathbf{x},\mathbf{z})$) can be tackled by repeated application of the base iteration. The idea is then to (i) apply the base iteration to (P_φ), starting from a (readily available) feasible point $(\mathbf{x},\mathbf{z})$, and (ii) augment the base iteration with a penalty-parameter updating rule to bring φ above such threshold. One such rule was proposed in [22,23] (again in the context of a specific constraint-reduced algorithm).

Problem (P_φ) enjoys the following properties to be invoked in the analysis.

Lemma 1 *Given $\varphi>0$, (P_φ) is strictly feasible. Further, for $\varphi>0$ large enough, (P_φ) is bounded, i.e., has a nonempty optimal solution set. Finally, for $\varphi>0$, given any $\rho>0$ and $\alpha\in\mathbb{R}$, the set $\mathcal{S}:=\{(\mathbf{x},\mathbf{z})\in\mathcal{F}_\alpha:\|\mathbf{z}\|\leq\rho\}$ is bounded, where $\mathcal{F}_\alpha:=\{\mathbb{x}:\mathbf{z}\geq\mathbf{0},\mathbf{A}\mathbf{x}+\mathbf{z}\geq\mathbf{b},\mathbb{f}_\varphi(\mathbb{x})\leq\alpha\}$.*

Proof First, trivially, given any $\mathbf{x}$, there exists $\mathbf{z}$ with large enough components that $(\mathbf{x},\mathbf{z})$ is strictly feasible. Next, boundedness of (P_φ) for φ large enough follows from feasibility of (P_φ) and feasibility of (D_φ) for φ large enough, where the latter follows from Assumption 1, since the only difference between the dual of (P) and (D_φ) is the constraint $\boldsymbol{\pi}+\boldsymbol{\xi}=\varphi\mathbf{1}$, with $\boldsymbol{\xi}\geq\mathbf{0}$, in the latter. As for the third claim, proceeding by contradiction, suppose that $\mathcal{S}$ is unbounded. Then $\mathcal{S}$ must contain a nontrivial recession (translated) cone, i.e., (since $\mathbf{z}$ is "bounded in $\mathcal{S}$",) there exists a (φ-dependent) direction $\mathbf{v}\neq\mathbf{0}$ such that $\mathbf{H}\mathbf{v}=\mathbf{0}$, $\mathbf{c}^T\mathbf{v}\leq 0$, and $\mathbf{A}\mathbf{v}\geq\mathbf{0}$. If $\mathbf{c}^T\mathbf{v}=0$, this contradicts boundedness of the optimal solution set of (P) (Assumption 1). On the other hand, if $\mathbf{c}^T\mathbf{v}<0$, this contradicts the assumption that the optimal solution set of (P) is nonempty (again, Assumption 1).

2.2 Proposed Meta-Algorithm

To solve (P) from a potentially infeasible starting point $\mathbf{x}^0$ using some given feasible-start CQP solver, we propose an infeasible-start meta-algorithm (Meta-Algorithm IS) that repeatedly applies the base iteration—a single iteration taken from the given feasible-start CQP solver—to the penalized relaxed problem (P_φ), starting from $(\mathbf{x}^0, \mathbf{z}^0)$ with $\mathbf{z}^0$ selected such that $(\mathbf{x}^0, \mathbf{z}^0)$ is feasible to (P_φ), while monitoring φ. Specifically, to guarantee convergence of the iterates to the solution set of original problem (P), Meta-Algorithm IS runs a penalty-parameter updating rule, which increases the penalty parameter φ if deemed necessary. A pseudocode for Meta-Algorithm IS is given in Pseudocode 1, specialized to the case where the base iteration is taken from a primal–dual type solver which generates a sequence of dual iterates $\{\mathbb{\lambda}^k\}$ in addition to $\{\mathbb{x}^k\}$. When a purely primal base iteration is used, $\mathbb{\lambda}^k \geq \mathbf{0}$ is to be specifically constructed based on $\mathbb{x}^k$. For an instance of this, see Pseudocode 4 in section 7.2.

Pseudocode 1: Meta-Algorithm IS

Initialization: $\varphi_0 > 0$; $\mathbb{x}^0 := [\mathbf{x}^0; \mathbf{z}^0]$ feasible to (P_{φ_0}); $\mathbb{\lambda}^0 \geq \mathbf{0}$;
for *iteration* $k = 0, 1, 2, \ldots$ **do**
  if *user-provided stopping criterion is satisfied* **then**
    stop
  $\varphi_{k+1} \leftarrow$ `penalty_parameter_update`$(\varphi_k, \mathbb{x}^k, \mathbb{\lambda}^k)$;
  /* `penalty_parameter_update()` updates the penalty parameter φ_k to $\varphi_{k+1} (\geq \varphi_k)$ based on the current penalty parameter value φ_k, the primal iterate $\mathbb{x}^k = [\mathbf{x}^k; \mathbf{z}^k]$, and the dual iterate $\mathbb{\lambda}^k$. */
  $(\mathbb{x}^{\mathrm{BI}}, \mathbb{\lambda}^{\mathrm{BI}}) \leftarrow (\mathbb{x}^k, \mathbb{\lambda}^k)$;
  repeat
    $(\mathbb{x}^{\mathrm{BI}}, \mathbb{\lambda}^{\mathrm{BI}}) \leftarrow$ `base_iteration`$(\varphi_{k+1}, \mathbb{x}^{\mathrm{BI}}, \mathbb{\lambda}^{\mathrm{BI}})$;
  until $\mathbb{f}_{\varphi_{k+1}}(\mathbb{x}^{\mathrm{BI}}) \leq \mathbb{f}_{\varphi_{k+1}}(\mathbb{x}^k)$;
  $\mathbb{x}^{k+1} \leftarrow \mathbb{x}^{\mathrm{BI}}$, $\mathbb{\lambda}^{k+1} \leftarrow \mathbb{\lambda}^{\mathrm{BI}}$;
  /* `base_iteration()` is applied repeatedly to update the primal-dual iterate $(\mathbb{x}^{\mathrm{BI}}, \mathbb{\lambda}^{\mathrm{BI}})$ for ($\mathrm{P}_{\varphi_{k+1}}$)-($\mathrm{D}_{\varphi_{k+1}}$), starting from $(\mathbb{x}^k, \mathbb{\lambda}^k)$, until a (non-strictly) lower primal objective function value is achieved. The resulting $(\mathbb{x}^{\mathrm{BI}}, \mathbb{\lambda}^{\mathrm{BI}})$ is then accepted as the next iterate for Meta-Algorithm IS. */

If desired, the input and output of `base_iteration()` can be augmented with other variables of interest.

Remark 1 Regardless of whether or not the base iteration enforces monotone decrease of the objective function, the sequence "seen" by the penalty-parameter update does enjoy such property, i.e., upon entry into the penalty-parameter update, $\mathbf{f}(\mathbf{x}^k) + \varphi_k \mathbf{z}^k \leq \mathbf{f}(\mathbf{x}^{k-1}) + \varphi_k \mathbf{z}^{k-1}$ for all $k \geq 1$. Such monotone decrease is key to Lemma 4 below, on which the convergence analysis relies.

Up until this point, we have not discussed the details of any specific base iteration (`base_iteration()`), neither have we introduced any specific penalty-parameter updating rule (`penalty_parameter_update()`). Indeed, our goal is to propose an infeasible-start framework that works for any base iteration and penalty-parameter updating rule that satisfy certain rather general requirements.

In the remainder of section 2, we introduce requirements to be imposed on the base iteration (section 2.3) and on the penalty-parameter updating rule (section 2.5), and give one example base iteration that satisfies the former (section 2.4); other such examples are briefly discussed in section 7. An example penalty-parameter updating rule is given in section 6.1, where the rule is proved to satisfy the requirements in a more general setting, with equality constraints included. In section 3, we prove that, when both sets of requirements are satisfied, the penalty parameter φ_k is eventually constant, and the primal iteration $\mathbf{x}^k$ converges to the optimal solution set of (P).

2.3 Requirements for the Base Iteration

When the base iteration is applied repeatedly toward solving a CQP of the form

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) := \frac{1}{2}x^THx + c^Tx \ \text{ s.t. } \ Ax \geq b, \tag{6}$$

(with any stopping criterion turned off,) it must construct a primal sequence $\{x^\ell\}$ which, together with some "dual" sequence $\{\lambda^\ell\}$ (constructed by the base iteration), with $\lambda^\ell \geq \mathbf{0}$ for all ℓ, satisfies the following requirements of (i) feasibility, (ii) eventual descent (Requirement BI2 guarantees that $\hat{K} := \{\ell > 0 \colon f(x^\ell) \leq f(x^{\ell'}), \forall \ell' < \ell\}$ is an infinite index set), and (iii)—when the descending primal subsequence $\{x^\ell\}_{\ell\in\hat{K}}$ is bounded—convergence to the optimal solution set. (Note that the latter requires *asymptotically exact* dual variable estimates.)

Requirement BI1 *Given x^ℓ primal feasible (i.e., $Ax^\ell \geq b$), the base iteration produces $x^{\ell+1}$ primal feasible.*[7]

Requirement BI2 *Given $\ell_0 > 0$, there exists $\ell > \ell_0$ such that $f(x^\ell) \leq f(x^{\ell_0})$.*

Requirement BI3 *Either $\{x^\ell\}_{\ell\in\hat{K}}$ is unbounded or*

$$\max\{\|S^\ell\lambda^\ell\|, \|Hx^\ell + c - A^T\lambda^\ell\|\} \to 0 \ \ on \ \hat{K}$$

(or both), where $S^\ell := \operatorname{diag}(Ax^\ell - b)$.

Remark 2 When boundedness of $\{x^\ell\}$ is otherwise guaranteed, Requirement BI2 is redundant: it is implied by Requirements BI1 and BI3.

[7] Iterations that require a *strictly* primal-feasible x^ℓ are automatically catered to, subject to providing a strictly feasible initial $(\mathbb{x}, \mathbb{z})$ in the Initialization step of Meta-Algorithm IS. Due to the relaxation, such strictly feasible points for (P_φ) are readily available.

2.4 Example: An Infeasible-Start CR-MPC Algorithm

In [28], a constraint-reduced interior-point algorithm dubbed CR-MPC[8] is proposed to tackle CQPs for which a strictly primal-feasible initial point $\mathbf{x}$ is available and no equality constraints are present. For ease of reference, a brief introduction to Algorithm CR-MPC is included in Appendix A, with pseudocode that describes its main steps. Since CR-MPC is a strictly feasible primal–dual algorithm, Meta-Algorithm IS needs to be initialized with $\mathbb{x}^0$ strictly feasible to (P_{φ_0}) and $\mathbb{\lambda}^0 > \mathbf{0}$. CR-MPC does produce an appropriate $\mathbb{\lambda}$ sequence; specifically, the output $\mathbb{\lambda}^{\mathrm{BI}}$ in line 8 of Meta-Algorithm IS is assigned the value $[\tilde{\mathbb{\lambda}}^+]_+$, where $\tilde{\mathbb{\lambda}}^+$ is as generated in Step 5 of iteration CR-MPC (Pseudocode 5). Here we show that under Assumption 1, iteration CR-MPC satisfies the Requirements BI in section 2.3.

Because iteration CR-MPC is a primal-strictly-feasible iteration with monotone decrease of the objective function, Requirements BI1 and BI2 are trivially satisfied. As for Requirement BI3, it follows from parts (i) and (iv) of Theorem 1 of [28] that it is also satisfied by CR-MPC, provided that the Assumptions 1 and 2 of [28] are met by (P_φ). Assumption 1 of [28] requires that (P_φ) be strictly feasible, be bounded, and have a bounded optimal solution set. The first property is established by Lemma 1 above, under Assumption 1 of the present paper. The second and third ones are invoked only in Lemma 5 of [28] (see the sentence immediately preceding that lemma) in proving boundedness of the primal sequence. Since unboundedness makes Requirement BI3 satisfied, the second and third properties in Assumption 1 of [28] are not needed here. As for Assumption 2 of [28] (linear independence of the gradients of active constraints at stationary points), when applied to (P_φ), it amounts to requiring linear independence, for all $\mathbf{x} \in \mathbb{R}^n$, of $\{\mathbf{a}_i : \mathbf{a}_i^T\mathbf{x} \le b_i\}$. Accordingly, in order to cover Assumption 2 of [28], we append here a third assumption to our list; *it is in force in the present subsection only.*

Assumption 3 [9] *For all $\mathbf{x} \in \mathbb{R}^n$, $\{\mathbf{a}_i : \mathbf{a}_i^T\mathbf{x} \le b_i\}$ is a linearly independent set.*

(Note that iteration CR-MPC enforces descent of the objective function, so that in Meta-Algorithm IS, `base_iteration()` in line 8 will be called exactly once per iteration k.)

2.5 Requirements for the Penalty-parameter Update

The penalty-parameter update has a dual purpose. First, see to it that φ_k (rapidly) achieves a value sufficient for (P_{φ_k}) to have a nonempty optimal

[8] A constraint-reduced version of Mehrotra's Predictor Corrector.

[9] While the authors of [28] (who are also the authors of the present paper) were not able to do away with such linear-independence assumption in proving that Theorem 1 of that paper holds, intuition and extensive numerical testing suggest that Assumption 3 can be dropped.

solution set. Second, further see to it that such value is further increased till solutions to (P_{φ_k}) are solutions to the original problem, and eventually remain constant. Existence of a threshold insuring the latter is indeed guaranteed by the "exact" character of the penalty function in (P_{φ_k}).

In view of Lemma 2 below, the first three requirements below are natural.

Requirement PU1 $\{\varphi_k\}$ *is a positive, nondecreasing scalar sequence that either is eventually constant or grows without bound.*

Requirement PU2 *If* $\{\mathbf{z}^k\}$ *is unbounded, then* $\varphi_k \to \infty$*.*

Requirement PU3 *If* φ_k *is eventually constant and equal to* $\hat{\varphi}$*, and* $\max\{\|\mathbb{S}^k\mathbb{\lambda}^k\|, \|\mathbb{H}\mathbb{x}^k + \mathbb{c}_{\varphi_k} - \mathbb{A}^T\mathbb{\lambda}^k\|, |(\mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}^T\boldsymbol{\pi}^k)^T\mathbf{x}^k|\} \to 0$*, then* $\hat{\varphi} > \liminf \|\boldsymbol{\pi}^k\|_\infty$*.*

While the above requirements allow for φ_k to be increased freely, the last one, stated next, insures that, when the stated assumptions are satisfied, φ_k will eventually remain constant indeed. This is achieved by preventing situations where φ_k is increased prematurely, based only on Requirement PU3, with each increase of φ_k possibly triggering an initial increase of $\|\boldsymbol{\pi}^k\|_\infty$, in turn triggering a further increase of φ_k, resulting in a runaway phenomenon. To this effect, it is important to give a "chance" to the base iteration to recover from the disruption caused by an increase of φ_k, so $\|\boldsymbol{\pi}^k\|_\infty$ can settle to a reasonably low value; i.e., not to rush to increase it merely because it is again less than $\|\boldsymbol{\pi}^k\|_\infty$. Accordingly (since, for constant φ, Requirement BI3 implies convergence to a solution of (P_φ)), the requirement below allows φ_k to "track" $\|\boldsymbol{\pi}^k\|_\infty$ *only if* the iteration does not diverge away from optimality, as indicated by growing duality measure or growing dual infeasibility. Indeed, as it turns out, in addition to φ_k not being already much larger than $\|\boldsymbol{\pi}^k\|_\infty$, boundedness of distance to optimality, together with boundedness of a certain inner product with $\mathbf{x}^k$, is sufficient.

Requirement PU4 *If* $\{\mathbf{z}^k\}$ *is bounded but* $\varphi_k \to \infty$*, then there exists an infinite index set* K *such that the following quantities are bounded on* K*:*

$$\mathbb{S}^k\mathbb{\lambda}^k \quad (\text{i.e., } \mathbf{S}^k\boldsymbol{\pi}^k \text{ and } \mathbf{Z}^k\boldsymbol{\xi}^k); \tag{7a}$$

$$\mathbb{H}\mathbb{x}^k + \mathbb{c}_{\varphi_k} - \mathbb{A}^T\mathbb{\lambda}^k \quad (\text{i.e., } \mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}^T\boldsymbol{\pi}^k \text{ and } \boldsymbol{\pi}^k + \boldsymbol{\xi}^k - \varphi_k\mathbf{1}); \tag{7b}$$

$$(\mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}^T\boldsymbol{\pi}^k)^T\mathbf{x}^k; \tag{7c}$$

$$\varphi_k / \max\{1, \|\boldsymbol{\pi}^k\|\}\,. \tag{7d}$$

Here $\boldsymbol{\pi}^k$ designates the subvector of $\mathbb{\lambda}^k$ associated to the first m constraints in (1) (see line before (3)). An instance of a penalty-parameter updating rule that satisfies a more general version (where equality constraints are allowed) of the Requirements PU is given in section 6.1.

3 Convergence Analysis for the Framework

Like the previous section, this section focuses exclusively on the case of problems without equality constraints, i.e., $p = 0$. The general case is dealt with in section 4. The analysis in this section is strongly inspired from that in [22] (and indirectly that in [23]), in particular Lemmas 3.2 to 3.4 of [22], streamlined and generalized here by allowing for the classes of base iterations and penalty-parameter updating rules specified in the previous section, rather than being tailored to a specific base iteration and penalty-parameter updating rule. It invokes the dual of (P), which, when $p = 0$, is

$$\underset{\mathbf{x}\in\mathbb{R}^n,\boldsymbol{\pi}\in\mathbb{R}^m}{\text{maximize}}\ \boldsymbol{\psi}(\mathbf{x},\boldsymbol{\pi}) := -\frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}+\boldsymbol{\pi}^T\mathbf{b} \ \text{ s.t. } \ \mathbf{H}\mathbf{x}+\mathbf{c}-\mathbf{A}^T\boldsymbol{\pi} = \mathbf{0},\ \boldsymbol{\pi} \geq \mathbf{0}. \quad \text{(D)}$$

Of course, a key for the penalty-parameter updating approach to succeed is that φ be (eventually) large enough, which is ascertained next.

Lemma 2 *Suppose* $(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\pi}^*, \boldsymbol{\xi}^*)$ *solves* (P_φ)–(D_φ) *for some* $\varphi > \|\boldsymbol{\pi}^*\|_\infty$*. Then* $\mathbf{z}^* = \mathbf{0}$ *and* $(\mathbf{x}^*, \boldsymbol{\pi}^*)$ *solves* (P)–(D)*.*

Proof Since $\varphi > \|\boldsymbol{\pi}^*\|_\infty$, feasibility for (D_φ) implies that $\boldsymbol{\xi}^* = \varphi\mathbf{1} - \boldsymbol{\pi}^* > \mathbf{0}$. Complementary slackness ($\mathbf{Z}^*\boldsymbol{\xi}^* = \mathbf{0}$) then implies that $\mathbf{z}^* = \mathbf{0}$. Therefore $(\mathbf{x}^*, \boldsymbol{\pi}^*)$ is feasible, thus optimal, for (P)–(D).

Proposition 1 *Suppose* φ_k *is eventually constant. Let* $\hat{\varphi} := \lim_{k\to\infty} \varphi_k$*. Then (i) the optimal solution set of* $(\mathrm{P}_{\hat{\varphi}})$ *is nonempty and bounded, and (ii) as* $k \to \infty$*,* $\mathbf{z}^k \to \mathbf{0}$ *and* $\mathbf{x}^k$ *converges to the optimal solution set of* (P)*.*

Proof Since φ_k is eventually constant, Requirement PU2 implies that $\{\mathbf{z}^k\}$ is bounded. From the third claim in Lemma 1 and the facts that (i) $\{\mathbf{z}^k\}$ is bounded, (ii) $\{(\mathbf{x}^k, \mathbf{z}^k)\}$ is feasible for $(\mathrm{P}_{\hat{\varphi}})$ (Requirement BI1), and (iii) $\mathbf{f}(\mathbf{x}^k) + \hat{\varphi}\mathbf{1}^T\mathbf{z}^k$ monotonically decreases (Requirement BI2, Remark 1), it follows that $\{\mathbb{x}^k\}$ is bounded. Requirement BI3 then gives that $\max\{\|\mathbb{S}^k\mathbb{\lambda}^k\|, \|\mathbb{H}\mathbb{x}^k + \mathbb{c}_{\varphi_k} - \mathbb{A}^T\mathbb{\lambda}^k\|\} \to 0$, which implies that $\mathbb{x}^k$ converges to the optimal solution set of $(\mathrm{P}_{\hat{\varphi}})$, and hence that $(\mathrm{P}_{\hat{\varphi}})$ is bounded. Next, from boundedness of $\{\mathbb{x}^k\}$, we have (again invoking Requirement BI3) $|(\mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}^T\boldsymbol{\pi}^k)^T\mathbf{x}^k| \to 0$. Requirement PU3 then leads to $\hat{\varphi} > \|\boldsymbol{\pi}^k\|_\infty$ for k large enough. It follows from Lemma 2 and Assumption 1 that the optimal solution set of $(\mathrm{P}_{\hat{\varphi}})$ is bounded. Finally, from Lemma 2, $\mathbf{z}^* = \mathbf{0}$ and $(\mathbf{x}^k, \boldsymbol{\pi}^k)$ converges to the set of primal–dual solutions to (P)–(D). Also, because the key properties of (P) (as listed in Assumption 1) are shared by $(\mathrm{P}_{\hat{\varphi}})$, all specific additional convergence properties of the base iteration are preserved.

Remark 3 Furthermore, possible additional convergence properties (beyond Requirement BI3) of the specific base iteration under consideration (with a feasible initial point), such that rate of convergence, are preserved when the initial point is infeasible for (P). If the number of times the penalty parameter is increased is polynomial in the number of variables, then possible polynomial complexity of the base algorithm will be preserved as well. Investigating when this is the case is beyond the scope of the present paper.

Towards proving that φ_k is eventually constant indeed, the next two lemmas show that the sequence of relaxation variables $\{\mathbf{z}^k\}$ is bounded. The first one gives an upper bound on the magnitude of $\mathbf{z}$ when $(\mathbf{x}, \mathbf{z})$ is feasible for (P$_\varphi$) and φ is large enough. For use in the proofs here and in section 4, recall that, because $\mathbf{H} \succeq \mathbf{0}$,

$$\hat{\mathbf{x}}^T\mathbf{H}\hat{\mathbf{x}} + \mathbf{x}^T\mathbf{H}\mathbf{x} - 2(\hat{\mathbf{x}}^T\mathbf{H}\mathbf{x}) = (\hat{\mathbf{x}} - \mathbf{x})^T\mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}) \geq 0\,. \tag{8}$$

Lemma 3 *Let* $(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})$ *be feasible for* (D) *and* $(\mathbf{x}, \mathbf{z})$ *be feasible for* (P$_\varphi$), *and let* $\varphi > \|\hat{\boldsymbol{\pi}}\|_\infty$. *Then*

$$\|\mathbf{z}\|_\infty \leq \frac{\mathbf{f}(\mathbf{x}) + \varphi\mathbf{1}^T\mathbf{z} - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})}{\varphi - \|\hat{\boldsymbol{\pi}}\|_\infty}\,. \tag{9}$$

Proof Feasibility of $(\mathbf{x}, \mathbf{z})$ for (P$_\varphi$) implies that $\mathbf{A}\mathbf{x} + \mathbf{z} \geq \mathbf{b}$ so that, since $\hat{\boldsymbol{\pi}} \geq \mathbf{0}$ (feasible for (D)),

$$\hat{\boldsymbol{\pi}}^T\mathbf{A}\mathbf{x} + \hat{\boldsymbol{\pi}}^T\mathbf{z} \geq \mathbf{b}^T\hat{\boldsymbol{\pi}}. \tag{10}$$

Since feasibility of $(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})$ for (D) implies $\mathbf{H}\hat{\mathbf{x}} + \mathbf{c} = \mathbf{A}^T\hat{\boldsymbol{\pi}}$, it follows that

$$-\hat{\boldsymbol{\pi}}^T\mathbf{z} \leq (\mathbf{H}\hat{\mathbf{x}} + \mathbf{c})^T\mathbf{x} - \mathbf{b}^T\hat{\boldsymbol{\pi}} \leq \mathbf{f}(\mathbf{x}) - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})\,, \tag{11}$$

where we have used (8). Since $\varphi > \|\hat{\boldsymbol{\pi}}\|_\infty$, $\hat{\boldsymbol{\xi}} := \varphi\mathbf{1} - \hat{\boldsymbol{\pi}} > \mathbf{0}$. Adding $\varphi\mathbf{1}^T\mathbf{z}$ to both sides of (11) then yields

$$\hat{\boldsymbol{\xi}}^T\mathbf{z} \leq \mathbf{f}(\mathbf{x}) + \varphi\mathbf{1}^T\mathbf{z} - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}}). \tag{12}$$

Then, since $\mathbf{z} \geq \mathbf{0}$ (feasible for (P$_\varphi$)),

$$\hat{\xi}_i z_i \leq \hat{\boldsymbol{\xi}}^T\mathbf{z} \leq \mathbf{f}(\mathbf{x}) + \varphi\mathbf{1}^T\mathbf{z} - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}}), \tag{13}$$

yielding, for $i = 1, \ldots, m$,

$$z_i \leq \frac{\mathbf{f}(\mathbf{x}) + \varphi\mathbf{1}^T\mathbf{z} - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})}{\hat{\xi}_i} \leq \frac{\mathbf{f}(\mathbf{x}) + \varphi\mathbf{1}^T\mathbf{z} - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})}{\varphi - \|\hat{\boldsymbol{\pi}}\|_\infty}. \tag{14}$$

Since $\mathbf{z} \geq \mathbf{0}$, the claim follows.

Boundedness of $\{\mathbf{z}^k\}$ can now be proved.

Lemma 4 *Sequence* $\{\mathbf{z}^k\}$ *is bounded.*

Proof Proceeding by contradiction, suppose $\{\mathbf{z}^k\}$ is unbounded, so that, from Requirement PU2, $\varphi_k \to \infty$ as $k \to \infty$. Then, given any (D)–feasible $(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})$, there exists k_1 such that $\varphi_k > \|\hat{\boldsymbol{\pi}}\|_\infty$ for all $k \geq k_1$ and in view of Lemma 3,

$$\|\mathbf{z}^{k-1}\|_\infty \leq \nu_k := \frac{\mathbf{f}(\mathbf{x}^{k-1}) + \varphi_k\mathbf{1}^T\mathbf{z}^{k-1} - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})}{\varphi_k - \|\hat{\boldsymbol{\pi}}\|_\infty}, \quad k \geq k_1. \tag{15}$$

To conclude, we show that $\{\nu_k\}$ is bounded, specifically, that $\nu_{k+1} \leq \nu_k$ for all $k \geq k_1$, contradicting unboundedness of $\{\mathbf{z}^k\}$. Since (see Remark 1) $\mathbf{f}(\mathbf{x}^k) +$

$\varphi_k \mathbf{1}^T\mathbf{z}^k \leq \mathbf{f}(\mathbf{x}^{k-1}) + \varphi_k \mathbf{1}^T\mathbf{z}^{k-1}$ for all $k \geq 1$, it suffices to show that, for all $k \geq k_1$,

$$(\nu_{k+1} =)\ \frac{\mathbf{f}(\mathbf{x}^k) + \varphi_{k+1}\mathbf{1}^T\mathbf{z}^k - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})}{\varphi_{k+1} - \|\hat{\boldsymbol{\pi}}\|_\infty} \leq \frac{\mathbf{f}(\mathbf{x}^k) + \varphi_k\mathbf{1}^T\mathbf{z}^k - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})}{\varphi_k - \|\hat{\boldsymbol{\pi}}\|_\infty}. \quad (16)$$

To that effect, we show that, for all k, the function $g_k : \mathbb{R} \to \mathbb{R}$ defined by

$$g_k(\varphi) := \frac{\mathbf{f}(\mathbf{x}^k) + \varphi\mathbf{1}^T\mathbf{z}^k - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})}{\varphi - \|\hat{\boldsymbol{\pi}}\|_\infty} \quad (17)$$

has a nonpositive derivative when $\varphi > \|\hat{\boldsymbol{\pi}}\|_\infty$. Indeed,

$$g_k'(\varphi) = -\frac{\mathbf{f}(\mathbf{x}^k) + \|\hat{\boldsymbol{\pi}}\|_\infty\mathbf{1}^T\mathbf{z}^k - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})}{(\varphi - \|\hat{\boldsymbol{\pi}}\|_\infty)^2} \quad (18)$$

and

$$\begin{aligned}\mathbf{f}(\mathbf{x}^k) + \|\hat{\boldsymbol{\pi}}\|_\infty\mathbf{1}^T\mathbf{z}^k - \boldsymbol{\psi}(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}}) &= \frac{1}{2}\hat{\mathbf{x}}^T\mathbf{H}\hat{\mathbf{x}} - \mathbf{b}^T\hat{\boldsymbol{\pi}} + \mathbf{f}(\mathbf{x}^k) + \|\hat{\boldsymbol{\pi}}\|_\infty\mathbf{1}^T\mathbf{z}^k \\ &\geq -\hat{\boldsymbol{\pi}}^T\mathbf{z}^k + \|\hat{\boldsymbol{\pi}}\|_\infty\mathbf{1}^T\mathbf{z}^k \geq 0\,,\end{aligned} \quad (19)$$

where we have used the facts that, given any (P_φ)–feasible $(\mathbf{x}, \mathbf{z})$ (and since $(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})$ is (D)–feasible), recalling (8),

$$\mathbf{f}(\mathbf{x}) + \frac{1}{2}\hat{\mathbf{x}}^T\mathbf{H}\hat{\mathbf{x}} \geq \mathbf{f}(\mathbf{x}) - \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \hat{\mathbf{x}}^T\mathbf{H}\mathbf{x} = (\mathbf{c} + \mathbf{H}\hat{\mathbf{x}})^T\mathbf{x} = \hat{\boldsymbol{\pi}}^T\mathbf{A}\mathbf{x}\,, \quad (20)$$

and that, since $\hat{\boldsymbol{\pi}} \geq \mathbf{0}$ and $\mathbf{A}\mathbf{x} + \mathbf{z} \geq \mathbf{b}$,

$$\hat{\boldsymbol{\pi}}^T\mathbf{A}\mathbf{x} - \hat{\boldsymbol{\pi}}^T\mathbf{b} = -\hat{\boldsymbol{\pi}}^T(\mathbf{b} - \mathbf{A}\mathbf{x}) \geq -\hat{\boldsymbol{\pi}}^T\mathbf{z}\,. \quad (21)$$

Since $\varphi_{k+1} \geq \varphi_k$ (Requirement PU1), the proof is complete.

It remains to show that, under Requirement PU4, φ_k is eventually constant, so Proposition 1 applies. This is done in the next two lemmas and Theorem 1.

Lemma 5 *Suppose $\varphi_k \to \infty$ as $k \to \infty$. Then there exists an infinite index set K that satisfies the properties listed in Requirement PU4. Further, given any such K, (i) $\mathbf{z}^k \to \mathbf{0}$ on K and (ii) $\{\mathbf{x}^k\}$ is bounded on K.*

Proof Since $\varphi_k \to \infty$ as $k \to \infty$, boundedness of $\{\mathbf{z}^k\}$ (Lemma 4) and Requirement PU4 guarantee existence of an infinite index set K such that (7a)–(7c) are bounded on K. Let $(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}})$ be (P)–(D)–feasible so that (i) $\hat{\mathbf{s}} := \mathbf{A}\hat{\mathbf{x}} - \mathbf{b} \geq \mathbf{0}$ and since $\mathbf{A}\mathbf{x}^k + \mathbf{z}^k - \mathbf{b} = \mathbf{s}^k$ for all k,

$$\mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}^k) - \mathbf{z}^k - (\hat{\mathbf{s}} - \mathbf{s}^k) = \mathbf{0} \quad \forall k, \quad (22)$$

and (ii) $\mathbf{A}^T\hat{\boldsymbol{\pi}} = \mathbf{H}\hat{\mathbf{x}} + \mathbf{c}$ and $\hat{\boldsymbol{\pi}} \geq \mathbf{0}$. Next, (7b)–(7c) imply that $(\hat{\mathbf{x}} - \mathbf{x}^k)^T(\mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}^T\boldsymbol{\pi}^k)$ is bounded for $k \in K$, and adding $(\hat{\mathbf{x}} - \mathbf{x}^k)^T(\mathbf{A}^T\hat{\boldsymbol{\pi}} - H\hat{\mathbf{x}} - \mathbf{c}) = 0$ to it yields that

$$(\hat{\mathbf{x}} - \mathbf{x}^k)^T\mathbf{A}^T(\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}^k) - (\hat{\mathbf{x}} - \mathbf{x}^k)^T\mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}^k) \text{ is bounded for } k \in K. \quad (23)$$

Now we first show that, for some C,

$$(\hat{\mathbf{x}} - \mathbf{x}^k)^T \mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}^k) + \hat{\boldsymbol{\pi}}^T \mathbf{s}^k + (\varphi_k \mathbf{1} - \hat{\boldsymbol{\pi}})^T \mathbf{z}^k \le C \quad \forall k \in K\,. \tag{24}$$

From (22)–(23), we have, for some $\{\beta_k\}$ bounded on K,

$$(\hat{\mathbf{x}}-\mathbf{x}^k)^T \mathbf{H}(\hat{\mathbf{x}}-\mathbf{x}^k) = (\hat{\mathbf{x}}-\mathbf{x}^k)^T \mathbf{A}^T(\hat{\boldsymbol{\pi}}-\boldsymbol{\pi}^k)+\beta_k = (\hat{\mathbf{s}}-\mathbf{s}^k+\mathbf{z}^k)^T(\hat{\boldsymbol{\pi}}-\boldsymbol{\pi}^k)+\beta_k\,. \tag{25}$$

Reorganizing and adding $\varphi_k(\mathbf{z}^k)^T\mathbf{1}$ to both sides yields, for $k \in K$,

$$\begin{aligned}(\hat{\mathbf{x}} - \mathbf{x}^k)^T \mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}^k) &+ (\mathbf{s}^k)^T\hat{\boldsymbol{\pi}} + (\mathbf{z}^k)^T(\varphi_k\mathbf{1} - \hat{\boldsymbol{\pi}}) \\ &= \hat{\mathbf{s}}^T\hat{\boldsymbol{\pi}} - (\hat{\mathbf{s}} - \mathbf{s}^k + \mathbf{z}^k)^T\boldsymbol{\pi}^k + \varphi_k(\mathbf{z}^k)^T\mathbf{1} + \beta_k \\ &= \hat{\mathbf{s}}^T\hat{\boldsymbol{\pi}} - \hat{\mathbf{s}}^T\boldsymbol{\pi}^k + (\mathbf{s}^k)^T\boldsymbol{\pi}^k + (\mathbf{z}^k)^T(\varphi_k\mathbf{1} - \boldsymbol{\pi}^k) + \beta_k.\end{aligned} \tag{26}$$

Here the second term is nonpositive, and Requirement PU4 (7a)–(7b) implies that the third and fourth terms are bounded on K. Thus, the boundedness of $\{\beta_k\}$ on K yields (24). Next, note that each of the three terms on the left-hand side of (24) is bounded from below, so that all three are bounded on K. Indeed, the first and second terms are nonnegative since $\mathbf{H} \succeq \mathbf{0}$, $\hat{\boldsymbol{\pi}} \ge \mathbf{0}$, and $\mathbf{s}^k \ge \mathbf{0}$; and the third term is nonnegative for k large enough since $\mathbf{z}^k \ge \mathbf{0}$ and $\varphi_k \to \infty$. Since $\varphi_k \to \infty$, claim (i) follows from boundedness of the third term in the left-hand side of (24).

With (24) in hand, invoking strict dual feasibility (Assumption 1), assume without loss of generality that $\hat{\boldsymbol{\pi}} > \mathbf{0}$. Then boundedness on K of the second term in (24) implies boundedness of $\{\mathbf{s}^k\}$ on K. From boundedness on K of $\{\mathbf{z}^k\}$ and $\{\mathbf{s}^k\}$ and the definition of $\mathbf{s}^k$, it follows that $\{\mathbf{A}\mathbf{x}^k\}$ is bounded on K. Also, since $\mathbf{H} = \mathbf{H}^T \succeq \mathbf{0}$, boundedness on K of the first term in (24) implies boundedness of $\mathbf{H}\mathbf{x}^k$, again on K. Finally, boundedness on K of $\{\mathbf{A}\mathbf{x}^k\}$ and $\{\mathbf{H}\mathbf{x}^k\}$ together with the full-rank assumption on $[\mathbf{H}; \mathbf{A}]$ (Assumption 2) proves claim (ii).

Lemma 6 *Suppose $\varphi_k \to \infty$ as $k \to \infty$ and let K be as in Lemma 5, so that $\mathbf{z}^k \to \mathbf{z}^* = \mathbf{0}$ on K, $\{\mathbf{x}^k\}$ is bounded on K, and K has the properties guaranteed by Requirement PU4. Then, given any limit point $\mathbf{x}^*$ of $\{\mathbf{x}^k\}$ on K, there exists a nonzero $\overline{\boldsymbol{\pi}}^* \ge \mathbf{0}$, such that*

$$\mathbf{A}^T\overline{\boldsymbol{\pi}}^* = \mathbf{0}, \quad \mathbf{S}^*\overline{\boldsymbol{\pi}}^* = \mathbf{0}, \tag{27}$$

where $\mathbf{s}^* := \mathbf{A}\mathbf{x}^* + \mathbf{z}^* - \mathbf{b} = \mathbf{A}\mathbf{x}^* - \mathbf{b}$.

Proof First, we have from Requirement PU4 (7b) that $\varphi_k\mathbf{1} - (\boldsymbol{\pi}^k + \boldsymbol{\xi}^k)$ is bounded on K. Letting $\overline{\boldsymbol{\pi}}^k := \frac{1}{\varphi_k}\boldsymbol{\pi}^k$ and $\overline{\boldsymbol{\xi}}^k := \frac{1}{\varphi_k}\boldsymbol{\xi}^k$, we conclude that $\overline{\boldsymbol{\pi}}^k + \overline{\boldsymbol{\xi}}^k \to \mathbf{1}$ on K and in view of Requirement PU4 (7d), $\overline{\boldsymbol{\pi}}^k$ is bounded away from $\mathbf{0}$ on K. Further, since (see Meta-Algorithm IS in Pseudocode 1) $(\boldsymbol{\pi}^k, \boldsymbol{\xi}^k) = \mathbb{\lambda}^k$ has nonnegative components, $\overline{\boldsymbol{\pi}}^k$ and $\overline{\boldsymbol{\xi}}^k$ are bounded on K, hence have limit points on K, and every limit point $\overline{\boldsymbol{\pi}}^*$ of $\overline{\boldsymbol{\pi}}^k$ on K satisfies $\overline{\boldsymbol{\pi}}^* \ge \mathbf{0}$ and $\overline{\boldsymbol{\pi}}^* \ne \mathbf{0}$. Finally, since $\mathbf{z}^* = \mathbf{0}$ and $\{\mathbf{x}^k\}$ is bounded on K (Lemma 5 (ii)), boundedness of (7a)–(7b) in Requirement PU4 yields, by dividing through by φ_k,

$$\mathbf{A}^T\overline{\boldsymbol{\pi}}^* = \mathbf{0}, \quad \mathbf{S}^*\overline{\boldsymbol{\pi}}^* = \mathbf{0}\,. \tag{28}$$

Theorem 1 *(i) φ_k is eventually constant and (ii) as $k \to \infty$, $\mathbf{z}^k \to \mathbf{0}$ and $\mathbf{x}^k$ converges to the optimal solution set of* (P).

Proof To prove claim (i), proceeding by contradiction, suppose that $\varphi_k \to \infty$ and let K be as in Lemma 5. Then in view of Lemma 5, $\mathbf{z}^k \to \mathbf{z}^* := \mathbf{0}$ on K and $\{\mathbf{x}^k\}$ is bounded on K. Let $\mathbf{x}^*$ be a limit points of $\{\mathbf{x}^k\}$ on K. From Lemma 6, there exists $\overline{\boldsymbol{\pi}}^* \neq \mathbf{0}$, with $\overline{\boldsymbol{\pi}}^* \geq \mathbf{0}$, such that

$$\mathbf{A}^T\overline{\boldsymbol{\pi}}^* = \mathbf{0} \quad \text{and} \quad \mathbf{S}^*\overline{\boldsymbol{\pi}}^* = \mathbf{0}, \tag{29}$$

i.e., $\overline{\pi}_i^* = 0$ for all i such that $s_i^* > 0$, where $\mathbf{s}^* := \mathbf{A}\mathbf{x}^* - \mathbf{b}$. Next, let $\mathbf{A}_{\text{act}}$ be the submatrix of $\mathbf{A}$ associated with active constraints at $\mathbf{x}^*$ (i.e., the rows of $\mathbf{A}_{\text{act}}$ are all those rows of $\mathbf{A}$ with index i such that $s_i^* = 0$); and let $\overline{\boldsymbol{\pi}}^*_{\text{act}}$ be the corresponding subvector of $\overline{\boldsymbol{\pi}}^*$. Then $\mathbf{A}_{\text{act}}\mathbf{x}^* = \mathbf{b}_{\text{act}}$ and (29) imply that

$$\mathbf{A}_{\text{act}}^T\overline{\boldsymbol{\pi}}^*_{\text{act}} = \mathbf{0}. \tag{30}$$

Now, invoking Assumption 1, let $\hat{\mathbf{x}}$ be strictly feasible for (P), i.e., $\mathbf{A}\hat{\mathbf{x}} > \mathbf{b}$, in particular, $\mathbf{A}_{\text{act}}\hat{\mathbf{x}} > \mathbf{b}_{\text{act}}$. With $\mathbf{v} := \hat{\mathbf{x}} - \mathbf{x}^*$, by subtraction, we get $\mathbf{A}_{\text{act}}\mathbf{v} > \mathbf{0}$. Left-multiplying both sides of (30) by $\mathbf{v}^T$ yields $(\mathbf{A}_{\text{act}}\mathbf{v})^T\overline{\boldsymbol{\pi}}^*_{\text{act}} = 0$, a contradiction since (29) together with $\overline{\boldsymbol{\pi}}^* \neq \mathbf{0}$ and $\overline{\boldsymbol{\pi}}^* \geq \mathbf{0}$ implies that $\overline{\boldsymbol{\pi}}^*_{\text{act}} \neq \mathbf{0}$ and $\overline{\boldsymbol{\pi}}^*_{\text{act}} \geq \mathbf{0}$. This proves the first claim. The second claim follows from Proposition 1.

4 Problems with Equality Constraints

One standard approach for handling linear equality constraints within an inequality-constrained optimization framework is, after constructing an initial point that satisfies the equality constraints, to simply carry out the inequality-constrained optimization on the affine space defined by the equality constraints, rather than on $\mathbb{R}^n$. Search directions based on the inequality constraints are thus projected on that subspace. A drawback of such approach is that possible sparsity of the original inequality-constraint matrix is not inherited by the projected inequality-constraint matrix, especially when the equality-constraint matrix is dense. Further, unless special care is taken, the initial equality-feasible point may be far removed from the region of interest, as its construction does not take the objective function into account. An alternative approach, proposed in [30] in a nonlinear-programming (NLP) context, deals with one side of the (possibly nonlinear) equality constraints (e.g., the side that is satisfied by the initial point) as an *inequality* constraint, and uses an exact (and smooth) ℓ_1 penalty function to drive the iterates to feasibility. A refined version of this approach was later used in [39] in an interior-point NLP context.

Inspired by the latter, we now formulate each scalar (linear) equality as *two* inequality constraints, i.e, we equivalently express (P) as

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \ \mathbf{f}(\mathbf{x}) := \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{c}^T\mathbf{x} \text{ s.t. } \mathbf{A}\mathbf{x} \geq \mathbf{b},\ \mathbf{C}\mathbf{x} \geq \mathbf{d},\ -\mathbf{C}\mathbf{x} \geq -\mathbf{d}, \tag{$\tilde{\text{P}}$}$$

which, as we will demonstrate, can be handled within the same infeasible-start framework. The dual of $(\tilde{\mathrm{P}})$ is given by

$$\underset{\mathbf{x},\boldsymbol{\pi},\boldsymbol{\eta},\boldsymbol{\zeta}}{\text{maximize}}\, \boldsymbol{\psi}(\mathbf{x},\boldsymbol{\pi},\boldsymbol{\eta},\boldsymbol{\zeta}) \text{ s.t. } \mathbf{H}\mathbf{x}+\mathbf{c}-\mathbf{A}^T\boldsymbol{\pi}-\mathbf{C}^T(\boldsymbol{\eta}-\boldsymbol{\zeta}) = \mathbf{0}, (\boldsymbol{\pi},\boldsymbol{\eta},\boldsymbol{\zeta}) \geq \mathbf{0}, \quad (\tilde{\mathrm{D}})$$

where $\mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{\pi} \in \mathbb{R}^m$, $\boldsymbol{\eta} \in \mathbb{R}^p$, $\boldsymbol{\zeta} \in \mathbb{R}^p$, and

$$\boldsymbol{\psi}(\mathbf{x},\boldsymbol{\pi},\boldsymbol{\eta},\boldsymbol{\zeta}) := -\frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \boldsymbol{\pi}^T\mathbf{b} + (\boldsymbol{\eta}-\boldsymbol{\zeta})^T\mathbf{d}\,. \tag{31}$$

The corresponding augmented problem is

$$\underset{\mathbf{x},\mathbf{z},\mathbf{y}}{\text{minimize}}\ \mathbf{f}(\mathbf{x}) + \varphi\mathbf{1}^T[\mathbf{z};\mathbf{y}] \text{ s.t. } \mathbf{A}\mathbf{x}+\mathbf{z} \geq \mathbf{b},\ \mathbf{z} \geq \mathbf{0}, \quad (\tilde{\mathrm{P}}_\varphi)$$
$$\mathbf{C}\mathbf{x}+\mathbf{y} \geq \mathbf{d},\ \mathbf{C}\mathbf{x}-\mathbf{y} \leq \mathbf{d}$$

$$\underset{\mathbf{x},\boldsymbol{\pi},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\zeta}}{\text{maximize}}\, \boldsymbol{\psi}(\mathbf{x},\boldsymbol{\pi},\boldsymbol{\eta},\boldsymbol{\zeta}) \quad \text{s.t. } \mathbf{H}\mathbf{x}+\mathbf{c}-\mathbf{A}^T\boldsymbol{\pi}-\mathbf{C}^T(\boldsymbol{\eta}-\boldsymbol{\zeta}) = \mathbf{0}, \quad (\tilde{\mathrm{D}}_\varphi)$$
$$\boldsymbol{\pi}+\boldsymbol{\xi} = \varphi\mathbf{1}, \boldsymbol{\eta}+\boldsymbol{\zeta} = \varphi\mathbf{1}, (\boldsymbol{\pi},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\zeta}) \geq \mathbf{0}$$

with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^p$, $\boldsymbol{\pi} \in \mathbb{R}^m$, $\boldsymbol{\xi} \in \mathbb{R}^m$ $\boldsymbol{\eta} \in \mathbb{R}^p$, $\boldsymbol{\zeta} \in \mathbb{R}^p$. We will also make use of the slack variables

$$\mathbf{t}_+^k := \mathbf{C}\mathbf{x}^k + \mathbf{y}^k - \mathbf{d} \geq \mathbf{0},\ \mathbf{t}_-^k := -\mathbf{C}\mathbf{x}^k + \mathbf{y}^k + \mathbf{d} \geq \mathbf{0}; \tag{32}$$

note that $\mathbf{t}_+^k + \mathbf{t}_-^k = 2\mathbf{y}^k$.

Remark 4 Note the dissymmetry between the way original inequalities are augmented and the way inequalities issued from equalities are augmented in $(\tilde{\mathrm{P}}_\varphi)$: unlike $\mathbf{z} \geq \mathbf{0}$, $\mathbf{y} \geq \mathbf{0}$ is not included. While including $\mathbf{y} \geq \mathbf{0}$ (which is implied by $\mathbf{C}\mathbf{x}+\mathbf{y} \geq \mathbf{d}$ and $\mathbf{C}\mathbf{x}-\mathbf{y} \leq \mathbf{d}$) would have simplified (by exploiting the symmetry) the expression of requirements for the penalty-parameter update as well as the ensuing analysis, the three sets of constraints involving $\mathbf{y}$ would then form a structurally linearly dependent set (the difference of the first two is twice the third one), and because all three are active when $\mathbf{y} = \mathbf{0}$ (which is the case at the solution when φ is large enough) this may rule out some possible base iterations (such as, in theory, CR-MPC).

Substituting $[\mathbf{A};\mathbf{C};-\mathbf{C}]$ for $\mathbf{A}$, $(\mathbf{z},\mathbf{y})$ for $\mathbf{z}$ and $(\boldsymbol{\pi},\boldsymbol{\eta}-\boldsymbol{\zeta})$ for $\boldsymbol{\pi}$, we obtain the following revised list of requirements for the penalty-parameter update.

Requirement PU1′ *$\{\varphi_k\}$ is a positive, nondecreasing scalar sequence that either is eventually constant or grows without bound.*

Requirement PU2′ *If $\{(\mathbf{z}^k,\mathbf{y}^k)\}$ is unbounded, then $\varphi_k \to \infty$.*

Requirement PU3′ *If φ_k is eventually constant and equal to $\hat{\varphi}$, and $\|G_1^k\|$, $\|G_2^k\|$, and $|G_3^k|$ tend to zero, where*

$$G_1^k := (\mathbf{S}^k\boldsymbol{\pi}^k,\ \mathbf{Z}^k\boldsymbol{\xi}^k,\ \mathbf{T}_+^k\boldsymbol{\eta}^k,\ \mathbf{T}_-^k\boldsymbol{\zeta}^k), \tag{33a}$$

$$G_2^k := (\mathbf{H}\mathbf{x}^k+\mathbf{c}-\mathbf{A}^T\boldsymbol{\pi}^k-\mathbf{C}^T(\boldsymbol{\eta}^k-\boldsymbol{\zeta}^k), \boldsymbol{\pi}^k+\boldsymbol{\xi}^k-\varphi_k\mathbf{1}, \boldsymbol{\eta}^k+\boldsymbol{\zeta}^k-\varphi_k\mathbf{1}), \tag{33b}$$

$$G_3^k := (\mathbf{H}\mathbf{x}^k+\mathbf{c}-\mathbf{A}^T\boldsymbol{\pi}^k-\mathbf{C}^T(\boldsymbol{\eta}^k-\boldsymbol{\zeta}^k))^T\mathbf{x}^k, \tag{33c}$$

then $\hat{\varphi} > \liminf\|[\boldsymbol{\pi}^k;\boldsymbol{\eta}^k-\boldsymbol{\zeta}^k]\|_\infty$.

Requirement PU4′ *If $\varphi_k \to \infty$ and $\{(\mathbf{z}^k, \mathbf{y}^k)\}$ is bounded, then there exists an infinite index set K such that G_1^k, G_2^k, G_3^k, and $\varphi_k/\max\{1, \|[\boldsymbol{\pi}^k; \boldsymbol{\eta}^k - \boldsymbol{\zeta}^k]\|\}$ are bounded on K.*

With Requirements PU1′ to PU4′ substituted for Requirements PU1 to PU4, and Meta-Algorithm IS in Pseudocode 1 extended in the obvious way to account for the additional variable $\mathbf{y}^k$, with some adjustments, the convergence analysis in section 3 extends to cases when equality constraints are present, as we show next. The following (readily proved) extended version of Lemma 1 will be used.

Lemma 7 *Given $\varphi > 0$, $(\tilde{\mathrm{P}}_\varphi)$ is strictly feasible. Further, for $\varphi > 0$ large enough, $(\tilde{\mathrm{P}}_\varphi)$ is bounded, i.e., has a nonempty optimal solution set. Finally, for $\varphi > 0$, given any $\rho > 0$ and $\alpha \in \mathbb{R}$, the set $\mathcal{S} := \{(\mathbf{x}, \mathbf{z}, \mathbf{y}) \in \mathcal{F}_\alpha : \|[\mathbf{z}; \mathbf{y}]\| \leq \rho\}$ is bounded, where $\mathcal{F}_\alpha := \{(\mathbf{x}, \mathbf{z}, \mathbf{y}) : \mathbf{z} \geq \mathbf{0},\ \mathbf{y} \geq \mathbf{0},\ \mathbf{A}\mathbf{x} + \mathbf{z} \geq \mathbf{b},\ \mathbf{C}\mathbf{x} + \mathbf{y} \geq \mathbf{d},\ -\mathbf{C}\mathbf{x} + \mathbf{y} \geq -\mathbf{d},\ \mathbf{f}(\mathbf{x}) + \varphi \mathbf{1}^T[\mathbf{z}; \mathbf{y}] \leq \alpha\}$.*

Next (Lemma 8 and Proposition 2), just like in the special context of section 3 (see Lemma 2 and Proposition 1), if φ_k is eventually constant, convergence to the optimal solution set occurs.

Lemma 8 *Suppose $(\mathbf{x}^*, \mathbf{z}^*, \mathbf{y}^*, \boldsymbol{\pi}^*, \boldsymbol{\xi}^*, \boldsymbol{\eta}^*, \boldsymbol{\zeta}^*)$ solves $(\tilde{\mathrm{P}}_\varphi)$–$(\tilde{\mathrm{D}}_\varphi)$ for some $\varphi > \|[\boldsymbol{\pi}^*; \boldsymbol{\eta}^* - \boldsymbol{\zeta}^*]\|_\infty$. Then $(\mathbf{z}^*, \mathbf{y}^*) = (\mathbf{0}, \mathbf{0})$ and $(\mathbf{x}^*, \boldsymbol{\pi}^*, \boldsymbol{\eta}^*, \boldsymbol{\zeta}^*)$ is an optimal solution to $(\tilde{\mathrm{P}})$–$(\tilde{\mathrm{D}})$.*

Proof First, $\mathbf{z}^* = \mathbf{0}$ follows as in the proof of Lemma 2. Next, $\varphi > \|\boldsymbol{\eta}^* - \boldsymbol{\zeta}^*\|_\infty$ implies that $\varphi \mathbf{1} > \boldsymbol{\eta}^* - \boldsymbol{\zeta}^*$, and since $\boldsymbol{\eta}^*, \boldsymbol{\zeta}^* \geq \mathbf{0}$, $\varphi \mathbf{1} > \boldsymbol{\eta}^*$; similarly, $\varphi \mathbf{1} > \boldsymbol{\zeta}^*$. Since feasibility for Eq. $(\tilde{\mathrm{D}}_\varphi)$ implies $\boldsymbol{\eta}^* + \boldsymbol{\zeta}^* = \varphi \mathbf{1}$, it follows that $\boldsymbol{\eta}^*, \boldsymbol{\zeta}^* > \mathbf{0}$. Complementary slackness then implies that $\mathbf{C}\mathbf{x}^* - \mathbf{d} + \mathbf{y}^* = \mathbf{0}$ and $\mathbf{C}\mathbf{x}^* - \mathbf{d} - \mathbf{y}^* = \mathbf{0}$, hence that $\mathbf{y}^* = \mathbf{0}$. Therefore, $(\mathbf{x}^*, \boldsymbol{\pi}^*, \boldsymbol{\eta}^*, \boldsymbol{\zeta}^*)$ is feasible, thus optimal, for $(\tilde{\mathrm{P}})$–$(\tilde{\mathrm{D}})$.

Proposition 2 *Suppose φ_k is eventually constant. Then, $[\mathbf{z}^k; \mathbf{y}^k] \to \mathbf{0}$ as $k \to \infty$ and $\mathbf{x}^k$ converges to the optimal solution set of $(\tilde{\mathrm{P}})$.*

Proof The proof is identical to that of Proposition 1, subject to replacing throughout $\mathbf{z}^k$ with $[\mathbf{z}^k; \mathbf{y}^k; \mathbf{y}^k]$, $\mathbb{z}^k$ with $(\mathbf{x}^k, \mathbf{z}^k, \mathbf{y}^k)$, and $\mathbb{\lambda}^k$ with $(\boldsymbol{\pi}^k, \boldsymbol{\xi}^k, \boldsymbol{\eta}^k, \boldsymbol{\zeta}^k)$ and invoking Requirements PU2′ and PU3′ instead of PU2 and PU3, Lemma 8 instead of Lemma 2, and Lemma 7 instead of Lemma 1.

Similar to Lemma 3, the next lemma provides an upper bound on the magnitude of relaxation variables $\mathbf{z}$ and $\mathbf{y}$ when $(\mathbf{x}, \mathbf{z}, \mathbf{y})$ is feasible for $(\tilde{\mathrm{P}}_\varphi)$ and φ is large enough. This bound is then used in Lemma 10 to show boundedness of $\{(\mathbf{z}^k, \mathbf{y}^k)\}$. Note that, in contrast with Lemma 3, Lemma 9 assumes a lower bound on the penalty parameter φ that is more restrictive than the one in Lemma 8. This however does not interfere with the analysis, since Lemma 9 is only invoked in the proof of Lemma 10, in which φ is assumed (in a contradiction argument) to be unbounded.

Lemma 9 *Let* $(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}})$ *be feasible for* $(\tilde{\mathrm{D}})$ *and* $(\mathbf{x},\mathbf{z},\mathbf{y})$ *be feasible for* $(\tilde{\mathrm{P}}_\varphi)$*, and let* $\varphi > \|[\hat{\boldsymbol{\pi}};2\hat{\boldsymbol{\eta}};2\hat{\boldsymbol{\zeta}}]\|_\infty$*. Then*

$$\|[\mathbf{z};\mathbf{y}]\|_\infty \leq \frac{\mathbf{f}(\mathbf{x})+\varphi\mathbf{1}^T[\mathbf{z};\mathbf{y}]-\boldsymbol{\psi}(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}})}{\frac{1}{2}(\varphi-\|[\hat{\boldsymbol{\pi}};2\hat{\boldsymbol{\eta}};2\hat{\boldsymbol{\zeta}}]\|_\infty)}\,. \tag{34}$$

Proof Feasibility of $(\mathbf{x},\mathbf{z},\mathbf{y})$ for $(\tilde{\mathrm{P}}_\varphi)$ implies that $\mathbf{A}\mathbf{x}+\mathbf{z}\geq\mathbf{b}$, $\mathbf{C}\mathbf{x}+\mathbf{y}\geq\mathbf{d}$, and $-\mathbf{C}\mathbf{x}+\mathbf{y}\geq-\mathbf{d}$ so that, since $\hat{\boldsymbol{\pi}}\geq\mathbf{0}$, $\hat{\boldsymbol{\eta}}\geq\mathbf{0}$, and $\hat{\boldsymbol{\zeta}}\geq\mathbf{0}$ (feasible for $(\tilde{\mathrm{D}})$),

$$\hat{\boldsymbol{\pi}}^T\mathbf{A}\mathbf{x}+\hat{\boldsymbol{\pi}}^T\mathbf{z}\geq\mathbf{b}^T\hat{\boldsymbol{\pi}},\quad \hat{\boldsymbol{\eta}}^T\mathbf{C}\mathbf{x}+\hat{\boldsymbol{\eta}}^T\mathbf{y}\geq\hat{\boldsymbol{\eta}}^T\mathbf{d},\quad -\hat{\boldsymbol{\zeta}}^T\mathbf{C}\mathbf{x}+\hat{\boldsymbol{\zeta}}^T\mathbf{y}\geq-\hat{\boldsymbol{\zeta}}^T\mathbf{d}. \tag{35}$$

Since feasibility of $(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}})$ for $(\tilde{\mathrm{D}})$ implies $\mathbf{H}\hat{\mathbf{x}}+\mathbf{c}=\mathbf{A}^T\hat{\boldsymbol{\pi}}+\mathbf{C}^T(\hat{\boldsymbol{\eta}}-\hat{\boldsymbol{\zeta}})$, it follows that

$$-\hat{\boldsymbol{\pi}}^T\mathbf{z}-(\hat{\boldsymbol{\eta}}+\hat{\boldsymbol{\zeta}})^T\mathbf{y}\leq(\mathbf{H}\hat{\mathbf{x}}+\mathbf{c})^T\mathbf{x}-\mathbf{b}^T\hat{\boldsymbol{\pi}}-(\hat{\boldsymbol{\eta}}-\hat{\boldsymbol{\zeta}})^T\mathbf{d}\leq\mathbf{f}(\mathbf{x})-\boldsymbol{\psi}(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}}), \tag{36}$$

where we again used (8). Since $\varphi>\|[\hat{\boldsymbol{\pi}};2\hat{\boldsymbol{\eta}};2\hat{\boldsymbol{\zeta}}]\|_\infty$, we have $\hat{\boldsymbol{\xi}}:=\varphi\mathbf{1}-\hat{\boldsymbol{\pi}}>\mathbf{0}$, $\hat{\boldsymbol{\alpha}}_1:=\frac{1}{2}\varphi\mathbf{1}-\hat{\boldsymbol{\eta}}>\mathbf{0}$, and $\hat{\boldsymbol{\alpha}}_2:=\frac{1}{2}\varphi\mathbf{1}-\hat{\boldsymbol{\zeta}}>\mathbf{0}$. Adding $\varphi(\mathbf{1}^T\mathbf{z}+\mathbf{1}^T\mathbf{y})$ to both sides of (36) then yields

$$[\hat{\boldsymbol{\xi}};\hat{\boldsymbol{\alpha}}_1;\hat{\boldsymbol{\alpha}}_2]^T[\mathbf{z};\mathbf{y};\mathbf{y}]\leq\mathbf{f}(\mathbf{x})+\varphi\mathbf{1}^T[\mathbf{z};\mathbf{y}]-\boldsymbol{\psi}(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}}). \tag{37}$$

Then, since $(\mathbf{z},\mathbf{y})\geq\mathbf{0}$ (feasible for $(\tilde{\mathrm{P}}_\varphi)$)

$$[\hat{\boldsymbol{\xi}};\hat{\boldsymbol{\alpha}}_1;\hat{\boldsymbol{\alpha}}_2]_i[\mathbf{z};\mathbf{y};\mathbf{y}]_i\leq[\hat{\boldsymbol{\xi}};\hat{\boldsymbol{\alpha}}_1;\hat{\boldsymbol{\alpha}}_2]^T[\mathbf{z};\mathbf{y};\mathbf{y}]\leq\mathbf{f}(\mathbf{x})+\varphi\mathbf{1}^T[\mathbf{z};\mathbf{y}]-\boldsymbol{\psi}(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}}), \tag{38}$$

yielding, for $i=1,\ldots,m+2p$,

$$[\mathbf{z};\mathbf{y};\mathbf{y}]_i\leq\frac{\mathbf{f}(\mathbf{x})+\varphi\mathbf{1}^T[\mathbf{z};\mathbf{y}]-\boldsymbol{\psi}(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}})}{[\hat{\boldsymbol{\xi}};\hat{\boldsymbol{\alpha}}_1;\hat{\boldsymbol{\alpha}}_2]_i}\leq\frac{\mathbf{f}(\mathbf{x})+\varphi\mathbf{1}^T[\mathbf{z};\mathbf{y}]-\boldsymbol{\psi}(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}})}{\frac{1}{2}(\varphi-\|[\hat{\boldsymbol{\pi}};2\hat{\boldsymbol{\eta}};2\hat{\boldsymbol{\zeta}}]\|_\infty)}, \tag{39}$$

where the last inequality can be verified by noting (i) that $(\hat{\alpha}_1)_j\geq\frac{1}{2}\varphi-\|\hat{\boldsymbol{\eta}}\|_\infty$ and $(\hat{\alpha}_2)_j\geq\frac{1}{2}\varphi-\|\hat{\boldsymbol{\zeta}}\|_\infty$, (ii) that since $\varphi-\|\hat{\boldsymbol{\pi}}\|_\infty>0$, $\hat{\xi}_j\geq\varphi-\|\hat{\boldsymbol{\pi}}\|_\infty>\frac{1}{2}(\varphi-\|\hat{\boldsymbol{\pi}}\|_\infty)$, and (iii) that this implies that $[\hat{\boldsymbol{\xi}};\hat{\boldsymbol{\alpha}}_1;\hat{\boldsymbol{\alpha}}_2]_i\geq\min\{\frac{1}{2}(\varphi-\|\hat{\boldsymbol{\pi}}\|_\infty),\frac{1}{2}\varphi-\|\hat{\boldsymbol{\eta}}\|_\infty,\frac{1}{2}\varphi-\|\hat{\boldsymbol{\zeta}}\|_\infty\}$. Since $(\mathbf{z},\mathbf{y})\geq\mathbf{0}$, the claim follows.

Lemma 10 *Sequence* $\{(\mathbf{z}^k,\mathbf{y}^k)\}$ *is bounded.*

Proof The proof is identical to that of Lemma 4 (invoking Lemma 9 instead of Lemma 3), subject to replacing throughout $\mathbf{z}^k$ with $(\mathbf{z}^k,\mathbf{y}^k)$, $\boldsymbol{\psi}(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}})$ with $\boldsymbol{\psi}(\hat{\mathbf{x}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}})$, and $\|\hat{\boldsymbol{\pi}}\|_\infty$ with $\|[\hat{\boldsymbol{\pi}};2\hat{\boldsymbol{\eta}};2\hat{\boldsymbol{\zeta}}]\|_\infty$.

Still following section 3, it remains to show that, under Requirement PU4′, φ_k is eventually constant, so Proposition 2 applies. This is done in the next two lemmas and Theorem 2.

Lemma 11 *Suppose* $\varphi_k\to\infty$ *as* $k\to\infty$*. Then there exists an infinite index set* K *such that (i)* $(\mathbf{z}^k,\mathbf{y}^k)\to\mathbf{0}$ *on* K *and (ii)* $\{\mathbf{x}^k\}$ *is bounded on* K*.*

Proof The proof of Lemma 5 is adapted as follows, with Requirement PU4′ now being invoked instead of Requirement PU4. Let $(\hat{\mathbf{x}}, \hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\zeta}})$ be $(\tilde{\mathrm{P}})$–$(\tilde{\mathrm{D}})$–feasible, $\hat{\mathbf{s}} := \mathbf{A}\hat{\mathbf{x}} - \mathbf{b} (\geq \mathbf{0})$, and $\hat{\mathbf{t}}_+ := \hat{\mathbf{t}}_- := \mathbf{0}$. Then (22) becomes a set of three equations:

$$\mathbf{A}(\hat{\mathbf{x}} - \mathbf{x}^k) - \mathbf{z}^k - (\hat{\mathbf{s}} - \mathbf{s}^k) = \mathbf{0} \quad \forall k \tag{40a}$$

$$\mathbf{C}(\hat{\mathbf{x}} - \mathbf{x}^k) - \mathbf{y}^k - (\hat{\mathbf{t}}_+ - \mathbf{t}_+^k) = \mathbf{0} \quad \forall k \tag{40b}$$

$$-\mathbf{C}(\hat{\mathbf{x}} - \mathbf{x}^k) - \mathbf{y}^k - (\hat{\mathbf{t}}_- - \mathbf{t}_-^k) = \mathbf{0} \quad \forall k\,. \tag{40c}$$

Dual feasibility combined with (40) now yields (replacing (23)) boundedness on K of

$$(\hat{\mathbf{x}} - \mathbf{x}^k)^T(\mathbf{A}^T(\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}^k) + \mathbf{C}^T((\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}^k) - (\hat{\boldsymbol{\zeta}} - \boldsymbol{\zeta}^k))) - (\hat{\mathbf{x}} - \mathbf{x}^k)^T\mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}^k), \tag{41}$$

and it now follows that there exists $D > 0$ such that, for all $k \in K$,

$$\begin{aligned}(\hat{\mathbf{x}} - \mathbf{x}^k)^T\mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}^k) + \hat{\boldsymbol{\pi}}^T\mathbf{s}^k + \hat{\boldsymbol{\eta}}^T\mathbf{t}_+^k + \hat{\boldsymbol{\zeta}}^T\mathbf{t}_-^k \\ + (\varphi_k\mathbf{1} - \hat{\boldsymbol{\pi}})^T\mathbf{z}^k + (\varphi_k\mathbf{1} - (\hat{\boldsymbol{\eta}} + \hat{\boldsymbol{\zeta}}))^T\mathbf{y}^k \leq D\,,\end{aligned} \tag{42}$$

replacing (24). The proof concludes essentially like that of Lemma 5.

The details that lead to (42) are as follows. Equation (25) becomes

$$\begin{aligned}&(\hat{\mathbf{x}} - \mathbf{x}^k)^T\mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}^k) \\ &= (\hat{\mathbf{x}} - \mathbf{x}^k)^T(\mathbf{A}^T(\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}^k) + \mathbf{C}^T((\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}^k) - (\hat{\boldsymbol{\zeta}} - \boldsymbol{\zeta}^k))) + \beta_k \\ &= (\hat{\mathbf{s}} - \mathbf{s}^k + \mathbf{z}^k)^T(\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}^k) + (\hat{\mathbf{t}}_+ - \mathbf{t}_+^k + \mathbf{y}^k)(\hat{\boldsymbol{\eta}} - \boldsymbol{\eta}^k) \\ &\quad + (\hat{\mathbf{t}}_- - \mathbf{t}_-^k + \mathbf{y}^k)(\hat{\boldsymbol{\zeta}} - \boldsymbol{\zeta}^k) + \beta_k\,,\end{aligned} \tag{43}$$

where β_k is bounded on K and the second equality follows from (40). Upon adding $\varphi_k\mathbf{1}^T[\mathbf{z}^k; \mathbf{y}^k]$ to both sides and reorganizing, we get (since $\hat{\mathbf{t}}_+ = \hat{\mathbf{t}}_- = \mathbf{0}$)

$$\begin{aligned}&(\hat{\mathbf{x}} - \mathbf{x}^k)^T\mathbf{H}(\hat{\mathbf{x}} - \mathbf{x}^k) + \hat{\boldsymbol{\pi}}^T\mathbf{s}^k + \hat{\boldsymbol{\eta}}^T\mathbf{t}_+^k + \hat{\boldsymbol{\zeta}}^T\mathbf{t}_-^k \\ &\qquad + (\varphi_k\mathbf{1} - \hat{\boldsymbol{\pi}})^T\mathbf{z}^k + (\varphi_k\mathbf{1} - (\hat{\boldsymbol{\eta}} + \hat{\boldsymbol{\zeta}}))^T\mathbf{y}^k \\ &= \hat{\mathbf{s}}^T\hat{\boldsymbol{\pi}} - \hat{\mathbf{s}}^T\boldsymbol{\pi}^k + (\mathbf{s}^k)^T\boldsymbol{\pi}^k + (\mathbf{t}_+^k)^T\boldsymbol{\eta}^k + (\mathbf{t}_-^k)^T\boldsymbol{\zeta}^k \\ &\qquad + (\varphi_k\mathbf{1} - \boldsymbol{\pi}^k)^T\mathbf{z}^k + (\varphi_k\mathbf{1} - (\boldsymbol{\eta}^k + \boldsymbol{\zeta}^k))^T\mathbf{y}^k + \beta_k\,,\end{aligned} \tag{44}$$

and essentially the same analysis as is done in the proof of Lemma 5 applies here, concluding the proof.

In the remainder of this section, the difference $\boldsymbol{\eta} - \boldsymbol{\zeta}$ plays a key role, so we define $\boldsymbol{\omega} := \boldsymbol{\eta} - \boldsymbol{\zeta}$, and similarly, $\boldsymbol{\omega}^k := \boldsymbol{\eta}^k - \boldsymbol{\zeta}^k$, etc.

Lemma 12 *Suppose $\varphi_k \to \infty$ as $k \to \infty$ and let K be as in Lemma 11, so that $(\mathbf{z}^k, \mathbf{y}^k) \to (\mathbf{z}^*, \mathbf{y}^*) = (\mathbf{0}, \mathbf{0})$ on K, $\{\mathbf{x}^k\}$ is bounded on K, and K has the properties guaranteed by Requirement PU4′. Then, given any limit point $\mathbf{x}^*$ of*

$\{\mathbf{x}^k\}$ on K, there exist $\overline{\boldsymbol{\pi}}^* \in \mathbb{R}^m_+$ (if $m > 0$) and $\overline{\boldsymbol{\varpi}}^* \in \mathbb{R}^p$ (if $p > 0$), with $(\overline{\boldsymbol{\pi}}^*, \overline{\boldsymbol{\varpi}}^*) \neq \mathbf{0}$, such that

$$\mathbf{A}^T\overline{\boldsymbol{\pi}}^* + \mathbf{C}^T\overline{\boldsymbol{\varpi}}^* = \mathbf{0}, \quad \mathbf{S}^*\overline{\boldsymbol{\pi}}^* = \mathbf{0}, \tag{45}$$

where $\mathbf{s}^* := \mathbf{A}\mathbf{x}^* + \mathbf{z}^* - \mathbf{b} = \mathbf{A}\mathbf{x}^* - \mathbf{b}$.

Proof First, we have from Requirement PU4′ (G_2^k) that $\varphi_k\mathbf{1} - (\boldsymbol{\pi}^k + \boldsymbol{\xi}^k)$ and $\varphi_k\mathbf{1} - (\boldsymbol{\eta}^k + \boldsymbol{\zeta}^k)$ are bounded on K, which implies that $\overline{\boldsymbol{\pi}}^k + \overline{\boldsymbol{\xi}}^k \to \mathbf{1}$ on K and $\overline{\boldsymbol{\eta}}^k + \overline{\boldsymbol{\zeta}}^k \to \mathbf{1}$ on K, where, again $\overline{\boldsymbol{\pi}}^k := \boldsymbol{\pi}^k/\varphi_k$, $\overline{\boldsymbol{\xi}}^k := \boldsymbol{\xi}^k/\varphi_k$, $\overline{\boldsymbol{\eta}}^k := \boldsymbol{\eta}^k/\varphi_k$, and $\overline{\boldsymbol{\zeta}}^k := \boldsymbol{\zeta}^k/\varphi_k$. Additionally, in view of Requirement PU4′, $(\overline{\boldsymbol{\pi}}^k, \overline{\boldsymbol{\varpi}}^k := (\boldsymbol{\eta}^k - \boldsymbol{\zeta}^k)/\varphi_k)$ is bounded away from $\mathbf{0}$ on K, and since $\overline{\boldsymbol{\pi}}^k$, $\overline{\boldsymbol{\xi}}^k$, $\overline{\boldsymbol{\eta}}^k$, and $\overline{\boldsymbol{\zeta}}^k$ all have nonnegative components, they are all bounded on K, and so is $\overline{\boldsymbol{\varpi}}^k$. Hence all have limit points on K, and every limit points $(\overline{\boldsymbol{\pi}}^*, \overline{\boldsymbol{\varpi}}^*)$ of $(\overline{\boldsymbol{\pi}}^k, \overline{\boldsymbol{\varpi}}^k)$ satisfies $\overline{\boldsymbol{\pi}}^* \geq \mathbf{0}$ and $(\overline{\boldsymbol{\pi}}^*, \overline{\boldsymbol{\varpi}}^*) \neq \mathbf{0}$. Finally, since $(\mathbf{z}^*, \mathbf{y}^*) = (\mathbf{0}, \mathbf{0})$ and $\{\mathbf{x}^k\}$ is bounded on K (Lemma 11 (ii)), boundedness of G_1^k and G_2^k in Requirement PU4′ yields, by dividing through by φ_k,

$$\mathbf{A}^T\overline{\boldsymbol{\pi}}^* + \mathbf{C}^T\overline{\boldsymbol{\varpi}}^* = \mathbf{0}, \quad \mathbf{S}^*\overline{\boldsymbol{\pi}}^* = \mathbf{0}\,. \tag{46}$$

Theorem 2 *(i) φ_k is eventually constant and (ii) $(\mathbf{z}^k, \mathbf{y}^k) \to \mathbf{0}$ as $k \to \infty$ and $\mathbf{x}^k$ converges to the set of solutions of* $(\tilde{\mathrm{P}})$*.*

Proof Following the proof of Theorem 1, invoking Lemma 12, we note that for some $\overline{\boldsymbol{\pi}}^* \geq \mathbf{0}$ (if $m > 0$) and $\overline{\boldsymbol{\varpi}}^*$ (if $p > 0$), with $(\overline{\boldsymbol{\pi}}^*, \overline{\boldsymbol{\varpi}}^*) \neq \mathbf{0}$, we have

$$\mathbf{A}^T\overline{\boldsymbol{\pi}}^* + \mathbf{C}^T\overline{\boldsymbol{\varpi}}^* = \mathbf{0}, \quad \text{with} \quad \overline{\pi}_i^* = 0 \quad \forall i \in \{i\colon s_i^* > 0\}\,, \tag{47}$$

where $\mathbf{s}^* := \mathbf{A}\mathbf{x}^* - \mathbf{b}$. Next, let $\mathbf{A}_{\text{act}}$ be the submatrix of $\mathbf{A}$ associated with active constraints at $\mathbf{x}^*$; i.e., the rows of $\mathbf{A}_{\text{act}}$ are all those rows of $\mathbf{A}$ with index i such that $s_i^* = 0$. Then $\mathbf{A}_{\text{act}}\mathbf{x}^* = \mathbf{b}_{\text{act}}$, $\mathbf{C}\mathbf{x}^* = \mathbf{d}$, and (47) imply that

$$\mathbf{A}_{\text{act}}^T\overline{\boldsymbol{\pi}}_{\text{act}}^* + \mathbf{C}^T\overline{\boldsymbol{\varpi}}^* = \mathbf{0}. \tag{48}$$

Now, invoking Assumption 1, let $\hat{\mathbf{x}}$ be strictly feasible for (P), i.e., $\mathbf{C}\hat{\mathbf{x}} = \mathbf{d}$ and $\mathbf{A}\hat{\mathbf{x}} > \mathbf{b}$, in particular, $\mathbf{A}_{\text{act}}\hat{\mathbf{x}} > \mathbf{b}_{\text{act}}$. Proceeding as in the proof of Theorem 1, we conclude that $\overline{\boldsymbol{\pi}}_{\text{act}}^* = \mathbf{0}$ and, from (48), that $\mathbf{C}^T\overline{\boldsymbol{\varpi}}^* = \mathbf{0}$, a contradiction since $\mathbf{C}$ has full row rank (Assumption 2), proving the first claim. The second claim follows from Proposition 2.

5 Certificate of Infeasibility

The assumption (part of Assumption 1) that (P) has a (strictly) feasible point generally cannot be ascertained at the outset, and in case of infeasibility it is desirable that the sequences generated by the algorithm provide, preferably early on, a *certificate of infeasibility.* In this section it is shown that the proposed framework does provide such a certificate and that, in addition, it provides an initial feasible point for a nearby feasible problem.

Thus, in this section, Assumption 1 is replaced with the following less restrictive assumption (primal feasibility of course is not assumed), involving auxiliary problem

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\ \mathbf{f}(\mathbf{x}) \text{ s.t. } \mathbf{Ax} \geq \mathbf{b}',\ \mathbf{Cx} \geq \mathbf{d}'_-,\ \mathbf{Cx} \leq \mathbf{d}'_+\,, \qquad (\mathrm{P}')$$

a feasible relaxation of the infeasible (P), with some $\mathbf{b}' \leq \mathbf{b}$, $\mathbf{d}'_- \leq \mathbf{d}$, and $\mathbf{d}'_+ \geq \mathbf{d}$ selected in such a way that (P′) is indeed feasible. (Note that if *some* such choice of (P′) satisfies Assumption 1′, then *every* such choice does.)

Assumption 1′ (P′) *has a (nonempty) bounded optimal solution set.*

Note that Assumption 1′ implies feasibility of the dual of (P′), which is equivalent to feasibility of (D). Lemmas 8 and 10, invoked in the analysis below, were established without using the primal-feasibility *nor strict-dual-feasibility* portions of Assumption 1, so that the less restrictive Assumption 1′ is sufficient there. The following additional assumption is also invoked.

Assumption 4 [10] (P′) *has a bounded feasible set.*

The notation used below is as in section 4. In particular, $\boldsymbol{\omega} := \boldsymbol{\eta} - \boldsymbol{\zeta}$, $\boldsymbol{\omega}^k := \boldsymbol{\eta}^k - \boldsymbol{\zeta}^k$, etc. It is well known (Farkas's Lemma; e.g. [29, Proposition 6.4.3(iii)]) that a system of the form $\mathbf{Ax} \geq \mathbf{b},\ \mathbf{Cx} = \mathbf{d}$ is infeasible, i.e., has no solution, if and only if there exists $(\boldsymbol{\pi}, \boldsymbol{\omega})$ such that

$$\boldsymbol{\pi} \geq \mathbf{0}, \quad \mathbf{A}^T\boldsymbol{\pi} + \mathbf{C}^T\boldsymbol{\omega} = \mathbf{0}, \quad \mathbf{b}^T\boldsymbol{\pi} + \mathbf{d}^T\boldsymbol{\omega} > 0\,. \qquad (49)$$

Now, consider the reparameterization/rescaling of $(\tilde{\mathrm{P}}_\varphi)$ obtained by defining $\alpha := 1/\varphi$ and scaling the objective function by α, viz.

$$\begin{aligned} \underset{\mathbf{x},\mathbf{z},\mathbf{y}}{\text{minimize}}\ \alpha\mathbf{f}(\mathbf{x}) + \mathbf{1}^T[\mathbf{z};\mathbf{y}] \quad \text{s.t. } & \mathbf{Ax} + \mathbf{z} \geq \mathbf{b},\ \mathbf{z} \geq \mathbf{0}, \\ & \mathbf{Cx} + \mathbf{y} \geq \mathbf{d},\ -\mathbf{Cx} + \mathbf{y} \geq -\mathbf{d} \end{aligned} \qquad (\tilde{\mathrm{P}}_\alpha)$$

with $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^p$. The limit problem (with $\alpha = 0$) and its dual are

$$\begin{aligned} \underset{\mathbf{x},\mathbf{z},\mathbf{y}}{\text{minimize}}\ \mathbf{1}^T[\mathbf{z};\mathbf{y}] \quad \text{s.t. } & \mathbf{Ax} + \mathbf{z} \geq \mathbf{b},\ \mathbf{z} \geq \mathbf{0}, & (\tilde{\mathrm{P}}_0) \\ & \mathbf{Cx} + \mathbf{y} \geq \mathbf{d},\ -\mathbf{Cx} + \mathbf{y} \geq -\mathbf{d}\,, \\ \underset{\boldsymbol{\pi}\in\mathbb{R}^m,\boldsymbol{\omega}\in\mathbb{R}^p}{\text{maximize}}\ \mathbf{b}^T\boldsymbol{\pi} + \mathbf{d}^T\boldsymbol{\omega} \quad \text{s.t. } & \mathbf{A}^T\boldsymbol{\pi} + \mathbf{C}^T\boldsymbol{\omega} = \mathbf{0}, & (\tilde{\mathrm{D}}_0) \\ & \boldsymbol{\omega} \in [-1, 1],\ \boldsymbol{\pi} \in [0, 1]\,, \end{aligned}$$

and optimality conditions are given by

$$\begin{gathered} \mathbf{A}^T\boldsymbol{\pi} + \mathbf{C}^T(\boldsymbol{\eta} - \boldsymbol{\zeta}) = \mathbf{0},\ \boldsymbol{\pi} + \boldsymbol{\xi} = \mathbf{1},\ \boldsymbol{\eta} + \boldsymbol{\zeta} = \mathbf{1}, \\ \mathbf{S}\boldsymbol{\pi} = \mathbf{0},\ \mathbf{Z}\boldsymbol{\xi} = \mathbf{0},\ \mathbf{T}_+\boldsymbol{\eta} = \mathbf{0},\ \mathbf{T}_-\boldsymbol{\zeta} = \mathbf{0},\ (\mathbf{s}, \mathbf{t}_+, \mathbf{t}_-, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\zeta}) \geq \mathbf{0}. \end{gathered} \qquad (50)$$

[10] Numerical experimentation, including tests with small-size problems that do not satisfy Assumption 4, suggests that mere boundedness of the *optimal* solution set of (P′), as implied by Assumption 1′, is sufficient for the results to hold. A proof of this is elusive at this time though. In any case, boundedness of the feasible set of course can be achieved by imposing appropriately large bounds to the components of $\mathbf{x}$.

Remark 5 $(\tilde{\mathrm{P}}_0)$ is equivalently written as

$$\underset{\mathbf{x},\mathbf{z},\mathbf{y}}{\text{minimize}} \ \|\mathbf{z}\|_1+\|\mathbf{y}\|_1 \text{ s.t. } \ \mathbf{z}\geq\mathbf{0},\ z_i\geq b_i-(\mathbf{A}\mathbf{x})_i,\ y_i\geq|d_i-(\mathbf{C}\mathbf{x})_i|\ \forall i\,,$$

i.e., its solution minimizes a measure of the maximum constraint violation.

The analysis proceeds as follows.

Lemma 13 *If* (P) *is infeasible, then* $\alpha_k\to 0$ *as* $k\to\infty$*.*

Proof By contradiction. If α_k does not tend to 0, then it is eventually constant, say, equal to $\hat{\alpha}>0$. Thus, Requirement PU2′ implies that $\{(\mathbf{z}^k,\mathbf{y}^k)\}$ is bounded. Since $(\tilde{\mathrm{P}}_{\hat{\alpha}})$ is strictly feasible, Assumption 1′ and Lemma 7 imply boundedness of its constrained level sets, hence boundedness of $\{\mathbf{x}^k\}$, and Requirements BI3 and PU3′ imply convergence to the set of optimal solution and $\hat{\varphi}:=1/\hat{\alpha}>\liminf\|[\boldsymbol{\pi}^*;\boldsymbol{\omega}^*]\|_\infty$. Lemma 8 then implies that $(\mathbf{x}^*,\boldsymbol{\pi}^*,\boldsymbol{\eta}^*,\boldsymbol{\zeta}^*)$ solves (P)–(D), in contradiction with (P) being infeasible.

Lemma 14 *If* (P) *is infeasible, then there exists an infinite index set* K *such that, as* $k\to\infty$*,* $k\in K$*,* $(\overline{\boldsymbol{\pi}}^k,\overline{\boldsymbol{\omega}}^k)$*, with* $\overline{\boldsymbol{\pi}}^k:=\alpha_k\boldsymbol{\pi}^k$ *and* $\overline{\boldsymbol{\omega}}^k:=\alpha_k\boldsymbol{\omega}^k$*, tends to the optimal solution set of* $(\tilde{\mathrm{D}}_0)$ *and* $(\mathbf{z}^k,\mathbf{y}^k)$ *tends to the optimal solution set of* $(\tilde{\mathrm{P}}_0)$*.*

Proof Let $\overline{\boldsymbol{\xi}}^k:=\alpha_k\boldsymbol{\xi}^k$, $\overline{\boldsymbol{\eta}}^k:=\alpha_k\boldsymbol{\eta}^k$, and $\overline{\boldsymbol{\zeta}}^k:=\alpha_k\boldsymbol{\zeta}^k$. In view of Lemma 10 (boundedness of $\{(\mathbf{z}^k,\mathbf{y}^k)\}$) and Requirement PU4′, since $\alpha_k\to 0$ (Lemma 13), we have, together with $(\overline{\boldsymbol{\pi}}^k,\overline{\boldsymbol{\xi}}^k,\overline{\boldsymbol{\eta}}^k,\overline{\boldsymbol{\zeta}}^k)\geq\mathbf{0}$,

$$\mathbf{S}^k\overline{\boldsymbol{\pi}}^k\to\mathbf{0},\quad \mathbf{Z}^k\overline{\boldsymbol{\xi}}^k\to\mathbf{0},\quad \mathbf{T}_+^k\overline{\boldsymbol{\eta}}^k\to\mathbf{0},\quad \mathbf{T}_-^k\overline{\boldsymbol{\zeta}}^k\to\mathbf{0}, \tag{51}$$
$$\alpha_k(\mathbf{H}\mathbf{x}^k+\mathbf{c})-\mathbf{A}^T\overline{\boldsymbol{\pi}}^k-\mathbf{C}^T\overline{\boldsymbol{\omega}}^k\to\mathbf{0},\quad \overline{\boldsymbol{\pi}}^k+\overline{\boldsymbol{\xi}}^k-\mathbf{1}\to\mathbf{0},\quad \overline{\boldsymbol{\eta}}^k+\overline{\boldsymbol{\zeta}}^k-\mathbf{1}\to\mathbf{0},$$

as $k\to\infty$, $k\in K$, for some infinite index set K. In particular, $(\overline{\boldsymbol{\pi}}^k,\overline{\boldsymbol{\xi}}^k,\overline{\boldsymbol{\eta}}^k,\overline{\boldsymbol{\zeta}}^k)$ is bounded on K. In view of Assumption 4, $\{\mathbf{x}^k\}$ is also bounded, and it follows that for any limit point $(\hat{\mathbf{x}},\hat{\mathbf{z}},\hat{\mathbf{y}},\hat{\boldsymbol{\pi}},\hat{\boldsymbol{\xi}},\hat{\boldsymbol{\eta}},\hat{\boldsymbol{\zeta}})$ of $(\mathbf{x}^k,\mathbf{z}^k,\mathbf{y}^k,\overline{\boldsymbol{\pi}}^k,\overline{\boldsymbol{\xi}}^k,\overline{\boldsymbol{\eta}}^k,\overline{\boldsymbol{\zeta}}^k)$ on K,

$$\hat{\mathbf{S}}\hat{\boldsymbol{\pi}}=\mathbf{0},\ \hat{\mathbf{T}}_+\hat{\boldsymbol{\eta}}=\mathbf{0},\ \hat{\mathbf{T}}_-\hat{\boldsymbol{\zeta}}=\mathbf{0},\ \hat{\mathbf{Z}}\hat{\boldsymbol{\xi}}=\mathbf{0},\ \mathbf{A}^T\hat{\boldsymbol{\pi}}+\mathbf{C}^T\hat{\boldsymbol{\omega}}=\mathbf{0},\ \hat{\boldsymbol{\pi}}+\hat{\boldsymbol{\xi}}=\mathbf{1},\ \hat{\boldsymbol{\eta}}+\hat{\boldsymbol{\zeta}}=\mathbf{1}, \tag{52}$$

implying the claim.

Lemma 15 *If* (P) *is infeasible, every solution* $(\boldsymbol{\pi}^*,\boldsymbol{\omega}^*)$ *of* $(\tilde{\mathrm{D}}_0)$ *satisfies* $\boldsymbol{\pi}^*\geq\mathbf{0}$*,* $\mathbf{A}^T\boldsymbol{\pi}^*+\mathbf{C}^T\boldsymbol{\omega}^*=\mathbf{0}$*, and* $\mathbf{b}^T\boldsymbol{\pi}^*+\mathbf{d}^T\boldsymbol{\omega}^*>0$*.*

Proof Immediate consequence of strong duality, since the dual of $(\tilde{\mathrm{P}}_0)$ is $(\tilde{\mathrm{D}}_0)$.

Together, these three lemmas establish the following.

Theorem 3 *If* (P) *is infeasible, then given* $\epsilon>0$*, there exists* $\hat{k}$ *such that*

$$\|\mathbf{A}^T\overline{\boldsymbol{\pi}}^{\hat{k}}+\mathbf{C}^T\overline{\boldsymbol{\omega}}^{\hat{k}}\|\leq\epsilon,\quad \mathbf{b}^T\overline{\boldsymbol{\pi}}^{\hat{k}}+\mathbf{d}^T\overline{\boldsymbol{\omega}}^{\hat{k}}>0, \tag{53}$$

and

$$\mathbf{1}^T[\mathbf{z}^{\hat{k}};\mathbf{y}^{\hat{k}}]\leq\mathbf{1}^T[\hat{\mathbf{z}};\hat{\mathbf{y}}]+\epsilon, \tag{54}$$

where $\overline{\boldsymbol{\pi}}^{\hat{k}}\geq\mathbf{0}$ *and* $(\hat{\mathbf{z}},\hat{\mathbf{y}})$ *solves* $(\tilde{\mathrm{P}}_0)$*.*

Hence, if (P) is infeasible, Meta-Algorithm IS in Pseudocode 1 provides a certificate of (approximate) infeasibility, as well as an ϵ-ℓ_1-least relaxation of the constraints, replacing $\mathbf{b}$ with $\mathbf{b}' := \mathbf{b} - \mathbf{z}^{\hat{k}}$, and "spreading" $\mathbf{Cx} = \mathbf{d}$ to $-\Delta\mathbf{d}_- \leq \mathbf{Cx} - \mathbf{d} \leq \Delta\mathbf{d}_+$, with

$$(\Delta\mathbf{d}_\pm)_i := \begin{cases} (\mathbf{y}^k)_i\,, & \text{if } (\mathbf{Cx}^k - \mathbf{d})_i \lesseqgtr 0 \\ 0\,, & \text{otherwise} \end{cases}$$

that makes $\mathbf{x}^{\hat{k}}$ feasible for the relaxed problem.[11]

6 Implementation and Numerical Experiments

6.1 A Penalty-Parameter Updating Rule

The penalty-parameter updating rule used in our experiments is specified in Pseudocode 2; here $\sigma_1 > 0$, $\sigma_2 > 1$, and $\gamma_0 > 0$, $\gamma_1 > 0$, $\gamma_2 > 0$, $\gamma_3 > 0$ are prescribed, but γ_1 through γ_3 can be freely reduced with every increase of φ_k.

Pseudocode 2: Penalty-parameter updating rule
$\varphi^+ \leftarrow$ `penalty_parameter_update`$(\varphi, \mathbb{x}, \mathbb{\lambda})$

Input: penalty parameter φ, primal-dual iterate $(\mathbb{x}, \mathbb{\lambda})$
Output: updated penalty parameter φ^+
1 $[\mathbf{x}; \mathbf{z}; \mathbf{y}] \leftarrow \mathbb{x}$, $\quad [\boldsymbol{\pi}; \boldsymbol{\eta}; \boldsymbol{\zeta}] \leftarrow \mathbb{\lambda}$;
2 $\varphi^+ \leftarrow \varphi$;
3 **if** $\|(\mathbf{z}, \mathbf{y})\| > \gamma_0 \varphi$ **then**
4 $\quad \varphi^+ \leftarrow \frac{\sigma_2}{\gamma_0} \|(\mathbf{z}, \mathbf{y})\|$
5 Compute G_1, G_2, G_3 (defined in (33)) at $([\mathbf{x}; \mathbf{z}; \mathbf{y}], [\boldsymbol{\pi}; \boldsymbol{\eta}; \boldsymbol{\zeta}])$;
6 **if** $\varphi^+ \leq \|[\boldsymbol{\pi}; \boldsymbol{\eta} - \boldsymbol{\zeta}]\|_\infty + \sigma_1$, $\|G_1\| \leq \gamma_1$, $\|G_2\| \leq \gamma_2$, *and* $|G_3| \leq \gamma_3$ **then**
7 $\quad \varphi^+ \leftarrow \sigma_2(\|[\boldsymbol{\pi}; \boldsymbol{\eta} - \boldsymbol{\zeta}]\|_\infty + \sigma_1)$

This rule satisfies Requirements PU1′ to PU4′, as shown next.

Requirement PU1′: Clear, since $\varphi_0 > 0$ in Meta-Algorithm IS, and $\sigma_1 > 0$ and $\sigma_2 > 1$ here.

Requirement PU2′: Lines 3–4 in Pseudocode 2 implies that $\varphi_{k+1} \geq \frac{1}{\gamma_0} \|(\mathbf{z}^k, \mathbf{y}^k)\|$ for all k, proving the claim.

Requirement PU3′: Suppose $\varphi_k = \hat{\varphi}$ for all $k > \hat{k}$. Then, in view of lines 3–4 in Pseudocode 2, it must be the case that $\{(\mathbf{z}^k, \mathbf{y}^k)\}$ is bounded. Further, since G_1^k, G_2^k, and G_3^k all tend to zero (see Requirement PU3′), lines 6–7 in

[11] Alternatively, $\mathbf{x}^{\hat{k}}$ is also feasible for the adjusted (rather than relaxed) problem obtained by still replacing $\mathbf{b}$ with $\mathbf{b}'$ but then including instead the *equality* constraints $\mathbf{Cx} = \mathbf{d}'$, with $\mathbf{d}' = \mathbf{Cx}^{\hat{k}}$.

Pseudocode 2 above implies that $\hat{\varphi} > \|[\boldsymbol{\pi}^k; \boldsymbol{\eta}^k - \boldsymbol{\zeta}^k]\|_\infty + \sigma_1$ for $k > \hat{k}$, so the requirement is satisfied.
Requirement PU4′: Suppose $\varphi_k \to \infty$ and $\{(\mathbf{z}^k, \mathbf{y}^k)\}$ is bounded, so the condition in line 3 above cannot hold more than finitely many times. Then, since $\varphi_k \to \infty$, the conditions in line 3 must be satisfied on an infinite index set K, implying that G_1^k, G_2^k and G_3^k are all bounded on K, and $\varphi_k \leq \|[\boldsymbol{\pi}^k; \boldsymbol{\eta}^k - \boldsymbol{\zeta}^k]\|_\infty + \sigma_1$ on K so that $\|[\boldsymbol{\pi}^k; \boldsymbol{\eta}^k - \boldsymbol{\zeta}^k]\|_\infty \to \infty$ on K. Thus $\frac{\varphi_k}{\|[\boldsymbol{\pi}^k; \boldsymbol{\eta}^k - \boldsymbol{\zeta}^k]\|_\infty} \leq 1 + \frac{\sigma_1}{\|[\boldsymbol{\pi}^k; \boldsymbol{\eta}^k - \boldsymbol{\zeta}^k]\|_\infty}$ is bounded on K, proving the claim.

6.2 Implementation Details

All numerical tests were run with a Matlab implementation of Meta-Algorithm IS (Pseudocode 1 in section 2.2), base iteration (CR-MPC proposed in [28], see also Appendix A), and penalty-parameter updating rule (Pseudocode 2 in section 6.1) on a machine with AMD Opteron(tm) CPU Processor 6376 (2.3GHz) and Matlab R2019a in Linux platform.

Stopping criterion In the implementation, the stopping criterion for Meta-Algorithm IS was $\texttt{Err} \leq \texttt{tol}$ with the normalized error term[12]

$$\texttt{Err} := \frac{\left\| \left[\mathbf{H}\mathbf{x} + \mathbf{c} - \mathbf{A}^T \boldsymbol{\pi} - \mathbf{C}^T(\boldsymbol{\eta} - \boldsymbol{\zeta});\ \min\{\mathbf{s}, \boldsymbol{\pi}\};\ \min\{[\mathbf{t}_+; \mathbf{t}_-], [\boldsymbol{\eta}; \boldsymbol{\zeta}]\} \right] \right\|_2}{\max\{\|\mathbf{H}\|_\infty,\ \|\mathbf{c}\|_\infty,\ \|\mathbf{A}\|_\infty,\ \|\mathbf{C}\|_\infty\}}, \tag{55}$$

where $\mathbf{s} := \mathbf{A}\mathbf{x} - \mathbf{b}$, $\mathbf{t}_+ := \mathbf{C}\mathbf{x} - \mathbf{d}$, $\mathbf{t}_- := -(\mathbf{C}\mathbf{x} - \mathbf{d})$, and $\min\{\cdot, \cdot\}$ denotes component-wise minimum. When equality constraints are not present, `Err` is reduced by setting $\min\{[\mathbf{t}_+; \mathbf{t}_-], [\boldsymbol{\eta}; \boldsymbol{\zeta}]\} = \mathbf{0}$, and $\mathbf{C} = \mathbf{0}$. We used $\texttt{tol} = 10^{-8}$ in all numerical tests.

Initialization Meta-Algorithm IS requires that $(\mathbf{x}^0, \mathbf{z}^0, \mathbf{y}^0)$ be feasible for the augmented primal problem $(\tilde{\mathrm{P}}_\varphi)$, while the CR-MPC base iteration requires primal-strictly-feasible initial points, i.e., $\mathbf{z}^0 > -\min\{\mathbf{A}\mathbf{x}^0 - \mathbf{b}, \mathbf{0}\}$ and $\mathbf{y}^0 > \mathrm{abs}(\mathbf{C}\mathbf{x}^0 - \mathbf{d})$, with $\mathrm{abs}(\cdot)$ the component-wise absolute value. In our tests, given a (problem dependent) $\mathbf{x}^0$, we chose $\mathbf{z}^0 = c_{\mathbf{z}}\mathbf{1}$[13] and $\mathbf{y}^0 = c_{\mathbf{y}}\mathbf{1}$ with $c_{\mathbf{z}} = -\min\{\min\{\mathbf{A}\mathbf{x}^0 - \mathbf{b}\}, 0\} + 1$ and $c_{\mathbf{y}} = \max\{\mathrm{abs}(\mathbf{C}\mathbf{x}^0 - \mathbf{d})\} + 1$. For the initial dual variable and penalty parameter, we used $\mathbb{\lambda}^0 = \mathbf{1}$ and $\varphi_0 = 1$.

Base iteration The base iteration used in the tests is that of Algorithm CR-MPC proposed in [28] (see Appendix A of the present paper for a brief description), with the stopping criterion turned off, and with implementation details (including parameter values) essentially identical to those laid out in

[12] Approximate primal feasibility $(\mathbf{s}, \mathbf{t}_+, \mathbf{t}_-) \geq \mathbf{0}$ (or $\approx \mathbf{0}$) is implicitly taken into account in the last three terms in the numerator.

[13] A possibility would be to freeze $\mathbf{z}$ at zero when $\mathbf{x}^0$ is primal feasible and indeed, when a component z_i^k of $\mathbf{z}^k$ reaches zero at some iteration k, freeze that component to zero thereafter. This was not done in the tests reported here.

section 3.2 of that paper. A notable exception is that, here, in connection with relaxation variables $(\mathbf{z}, \mathbf{y})$, the constraints are structurally sparse, and this was specifically attended to in the solution of the Newton-KKT systems; thus, the associated CPU cost was only slightly higher than if there were only n, rather than $n + m + p$ variables. A few constraint selection rules were considered in [28] for Algorithm CR-MPC. Here we used Rule R (see [28, section 2.6]) with the same parameter values as in [28] but with two minor modifications: (i) we keep the slack threshold δ_k equal to its initial value $\bar{\delta}$ in the first five iteration, and (ii) we always include the sparse constraints $\mathbf{z} \geq \mathbf{0}$, $\mathbf{C}\mathbf{x}+\mathbf{y} \geq \mathbf{d}$, and $\mathbf{C}\mathbf{x} - \mathbf{y} \leq \mathbf{d}$ (all of which are eventually active) in the selected constraint set. In the numerical tests, (i) improved the robustness of Rule R to the choice of $(\mathbf{x}^0, \mathbf{z}^0, \mathbf{y}^0)$, while (ii) led to faster convergence with little additional cost per iteration.

Penalty-parameter update We implemented the penalty-parameter update in Meta-Algorithm IS following the rule given in Pseudocode 2 in section 6.1, with parameter values $\sigma_1 = 1$, $\sigma_2 = 10$, $\gamma_0 := \frac{\|[\mathbf{z}^0;\mathbf{y}^0]\|_\infty}{\varphi_0}$ and, for $i = 1, 2, 3$, $\gamma_i := \|G_i^0\|_2$. We chose $\|(\mathbf{z}, \mathbf{y})\| := \|[\mathbf{z}; \mathbf{y}]\|_\infty$ in line 3, and $\|G_i\| = \|G_i\|_2$ for $i = 1, 2$, in line 6. Importantly, at every increase of φ, the internal base iteration variables, e.g., regularization and constraint selection variables, were reset to the initial values specified in [28], since a new optimization problem (different objective function) is then dealt with.

Detection of infeasibility As discussed in section 5, the proposed framework provides an infeasible certificate whenever (P) is infeasible. Stopping criterion (55) was thus augmented with an alternative criterion (see (49)) which is declared satisfied when a "certificate" $(\hat{\boldsymbol{\pi}}^k, \hat{\boldsymbol{\omega}}^k)$ is produced such that

$$\mathbf{b}^T\hat{\boldsymbol{\pi}}^k + \mathbf{d}^T\hat{\boldsymbol{\omega}}^k > \sqrt{\epsilon_{\mathtt{m}}} \quad \text{and} \quad \frac{\|[\mathbf{A}^T\hat{\boldsymbol{\pi}} + \mathbf{C}^T\hat{\boldsymbol{\omega}};\ \min\{\hat{\boldsymbol{\pi}}, \mathbf{0}\}]\|_2}{\max\{\|\mathbf{A}\|_\infty, \|\mathbf{C}\|_\infty\}} \leq \mathtt{tol}_{\mathtt{infeas}}, \tag{56}$$

where $\epsilon_{\mathtt{m}}$ is the machine precision and $\mathtt{tol}_{\mathtt{infeas}}$ a tolerance parameter, in which case (P) is declared to be infeasible. We used $\mathtt{tol}_{\mathtt{infeas}} = 10^{-6}$ in the numerical tests.

An obvious choice for $(\hat{\boldsymbol{\pi}}^k, \hat{\boldsymbol{\omega}}^k)$ is to set it to the pair $(\bar{\boldsymbol{\pi}}^k, \bar{\boldsymbol{\omega}}^k)$ used in Theorem 3, which is defined as $(\bar{\boldsymbol{\pi}}^k, \bar{\boldsymbol{\omega}}^k) := (\boldsymbol{\pi}^k/\varphi_k; (\boldsymbol{\eta}^k - \boldsymbol{\zeta}^k)/\varphi_k)$ with $(\boldsymbol{\pi}^k, \boldsymbol{\eta}^k, \boldsymbol{\zeta}^k)$ the dual variables given by the base iteration. However, we found that for some infeasible problems, this choice requires many iterations to satisfy (56). In our implementation, we constructed $(\hat{\boldsymbol{\pi}}^k, \hat{\boldsymbol{\omega}}^k)$ by setting

$$[\hat{\boldsymbol{\pi}}_Q^k; \hat{\boldsymbol{\omega}}^k] := [\,[\mathbf{p}_{\boldsymbol{\pi}_Q}]_+; \mathbf{p}_{\boldsymbol{\omega}}]\,,\ [\mathbf{p}_{\boldsymbol{\pi}_Q}; \mathbf{p}_{\boldsymbol{\omega}}] := \mathrm{proj}_{\mathcal{N}([\mathbf{A}_Q^T, \mathbf{C}^T])}([\boldsymbol{\pi}_Q^k/\varphi_k; (\boldsymbol{\eta}^k - \boldsymbol{\zeta}^k)/\varphi_k])\,, \tag{57}$$

and $\hat{\boldsymbol{\pi}}_{Q^c} = \mathbf{0}$, where Q and Q^c denote the reduced constraint index set and its complement, both given by the CR-MPC base iteration, $\boldsymbol{\pi}_Q$ and $\mathbf{A}_Q$ denote the subvector and submatrix of $\boldsymbol{\pi}$ and $\mathbf{A}$ associated to the index set Q, respectively (see, e.g., [28], for details). In (57), $\mathrm{proj}_{\mathcal{N}([\mathbf{A}_Q^T, \mathbf{C}^T])}$ denotes the orthogonal projection operator onto the null space of $[\mathbf{A}_Q^T, \mathbf{C}^T]$. We note that, with this choice of $(\hat{\boldsymbol{\pi}}^k, \hat{\boldsymbol{\omega}}^k)$ substituted for $(\overline{\boldsymbol{\pi}}^k, \overline{\boldsymbol{\omega}}^k)$, Lemma 14 still holds, so that, on an infeasible problem, (56) will eventually be satisfied, and an infeasibility certificate will be produced. Finally, the computational cost of running the infeasibility test is negligible in comparison with the overall cost of an iteration.

6.3 Randomly Generated Problems

We first tested Meta-Algorithm IS with the CR-MPC base iteration on imbalanced ($m \gg n$) randomly generated problems both with and without equality constraints. We considered problems of the form (P) with sizes $m := 10\,000$, n ranging from 10 to 200, and $p = n/2$ or 0. We solved two sub-classes of problems: (i) strongly convex—$\mathbf{H}$ diagonal and positive definite, with random diagonal entries from uniform distribution $\mathcal{U}(0,1)$—and (ii) linear—$\mathbf{H} = \mathbf{0}$. For each sub-class, 20 randomly generated problems were solved for each problem size, and the results averaged over the 20 problems were reported. Consistent results were observed with $\mathbf{H} \neq \mathbf{0}$ but $\det(\mathbf{H})=0$; the specifics are not reported here. The entries of $\mathbf{A}$, $\mathbf{C}$, and $\mathbf{c}$ were taken from a standard normal distribution $\mathcal{N}(0,1)$; as for $\mathbf{b}$ and $\mathbf{d}$, see sections 6.3.1 and 6.3.2.

6.3.1 Comparison on Feasible Problems

To guarantee strict feasibility (Assumption 1), we generated $\mathbf{x}^{\text{feas}}$ and $\mathbf{s}^{\text{feas}}$ with i.i.d. entries taken from $\mathcal{N}(0,1)$ and uniform distribution $\mathcal{U}(1,2)$, respectively, and then set $\mathbf{b} := \mathbf{A}\mathbf{x}^{\text{feas}} - \mathbf{s}^{\text{feas}}$ and $\mathbf{d} := \mathbf{C}\mathbf{x}^{\text{feas}}$. *For feasible-start algorithms considered in the comparison, the starting point was* $\mathbf{x}^0 := \mathbf{x}^{\text{feas}}$, while for Meta-Algorithm IS, a starting point $\mathbf{x}^0$ was generated by repeatedly taking i.i.d. entries from $\mathcal{N}(0,1)$ until $\mathbf{x}^0$ became infeasible. All other codes used in the comparison do not allow the user to provide an initial point. For scaling purpose, we followed the heuristic proposed in [27] and (for all tested codes) used the normalized constraints $(\mathbf{D}_1\mathbf{A})\mathbf{x} \geq \mathbf{D}_1\mathbf{b}$ and $(\mathbf{D}_2\mathbf{C})\mathbf{x} = \mathbf{D}_2\mathbf{d}$, where $\mathbf{D}_1 = \operatorname{diag}(1/\|\mathbf{a}_i\|_2)$ and $\mathbf{D}_2 = \operatorname{diag}(1/\|\mathbf{c}_i\|_2)$ with $\mathbf{a}_i$ and $\mathbf{c}_i$ the i-th row of $\mathbf{A}$ and $\mathbf{C}$, respectively. The modified $\mathbf{A}$ and $\mathbf{C}$ matrices were also used in the stopping criteria (55) and (56).

Fig. 1 reports the iteration counts and computation time of the tested algorithms on the two sub-classes of problems with equality constraints. Here the proposed Meta-Algorithm IS with the CR-MPC base iteration (with "Rule R" for constraint selection), IS-CR-MPC, is compared to the same with constraint reduction turned off (IS-MPC*).[14] For both cases, the tolerance `tol` was set to 10^{-8}. Also, three popular QP solvers, MOSEK (ver. 9.1.9) [2, 3], OSQP [38], and qpOASES [18] are included in the comparison. The OSQP (Operator Splitting Quadratic Program) solver runs an alternating direction

[14] Here we denote the CR-MPC algorithm with constraint reduction turned off as MPC* (rather than MPC) to avoid confusion with the original MPC algorithm in [31].

method of multipliers (ADMM) [9], and qpOASES is an active set QP solver. We include them here to benchmark the performance of Meta-Algorithm IS in comparison to other types of solvers in addition to interior-point method. Here the MOSEK and OSQP solvers were accessed through their Matlab API, and the qpOASES solver was called through the CasADi interface [4]. In all reported tests, no starting points were supplied to these three solvers (i.e., the initial guesses were generated by the solvers from problem data), and all parameters for these solvers were set to the default values, with the exception that the convergence tolerances were set to 10^{-8} as well.[15] As seen in Fig. 1, on such imbalanced CQPs, in spite of the fact that, in terms of iteration count, IS-CR-MPC is inferior to MOSEK[16] , the total computation time recorded by IS-CR-MPC is 30% to three times lower than that recorded by the second best solver (qpOAESE or MOSEK).

Fig. 2 illustrates the results on problems with *no equality constraints*. For these tests, we also included the feasible-start CR-MPC algorithm of [28] and the same with constraint reduction turned off (MPC*) into the comparison, with convergence criterion given in [28] and tolerance 10^{-8}. In the linear ($\mathbf{H} = \mathbf{0}$) case, we included in the comparison a revised primal simplex (RPS) code with partial pricing (referred to in the legend of Figure 2 as "Two-phase RPS") used in [40][17] which solves the dual of (P) via a two-phase approach:[18] following standard practice, phase 1 starts from the readily available basic feasible solution (BFS) for the auxiliary feasibility problem to generate a BFS for solving the *original* problem (phase 2). A brief description and a pseudocode for the RPS method are included in Appendix B. We refer the reader to, e.g., [8] and references therein for more comprehensive discussion.

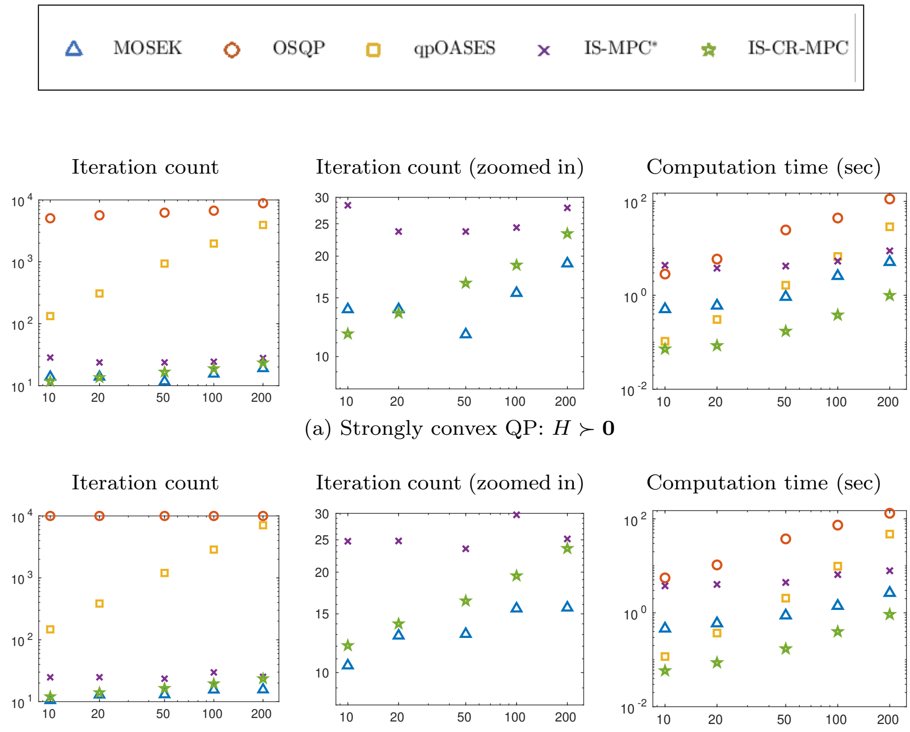
As shown in Fig. 2, the feasible-start MPC* and CR-MPC solvers required fewer iterations to converge than IS-CR-MPC, most likely due to the readily available feasible initial point (a "warm-start" of sorts). On the tested (imbalanced) problems, the constraint-reduced solvers generally outperformed other solvers in terms of computation time. The feasible-start CR-MPC algorithm is at most two times faster than IS-CR-MPC, which reflects the difference in iteration counts. In tests not reported here, we also observed that, when starting from the feasible $\mathbf{x}^{\text{feas}}$, IS-CR-MPC and CR-MPC give nearly identical performance, i.e., the overhead for allowing infeasible start is minor.

[15] To avoid biases due to different stopping criteria, we also experimented with tolerances set to 10^{-6} (while keeping $\texttt{tol} = 10^{-8}$ for all MPC versions) and observed results with slightly fewer iterations and nearly identical computation time.

[16] Here the iteration counts of OSQP and qpOASES are significantly higher than other solver, which is as expected. However, OSQP failed to converge within 10 000 iterations in several problem instances. We observed that OSQP converges on all problem instances when the tolerance is relaxed from 10^{-8} to 10^{-4}, but the computation time is still higher than other compared solvers.

[17] We used an implementation due to Luke Winternitz, who kindly made it available to us.

[18] Two-phase RPS is not included in Fig. 1b because when $p \neq 0$, the dual of (P) is not in standard dual form, a form it requires.



(a) Strongly convex QP: $H \succ \mathbf{0}$

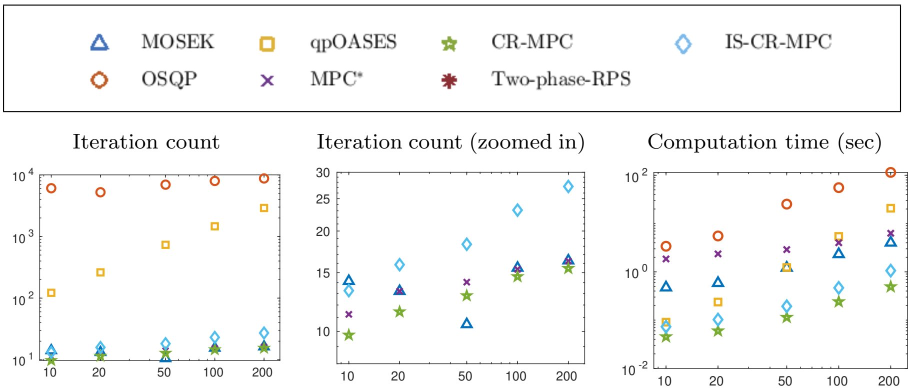
(b) Linear optimization: $H = \mathbf{0}$

Fig. 1: Randomly generated problems with $m = 10\,000$ inequality constraints and $p = \frac{n}{2}$ equality constraints. Results are pictured for two sub-classes of problems. In each figure, the x-axis is the number of variables (n) and the y-axis is the iteration count or total computation time, both averaged over the 20 problem instances and plotted in logarithmic scale. In the tests, IS-MPC* and IS-CR-MPC started from a randomly generated infeasible point, and MOSEK, OSQP, and qpOASES started from their own initial guesses.

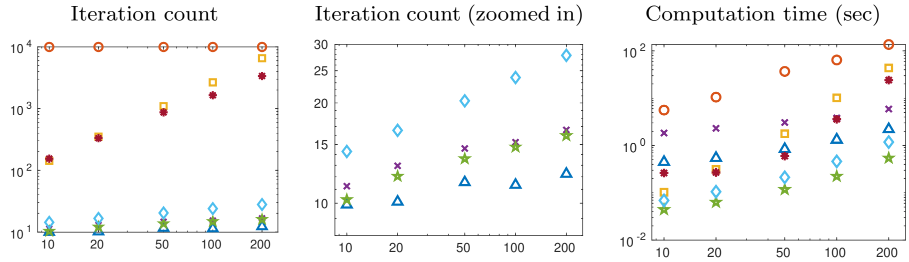
6.3.2 Infeasibility Detection Tests

Here, the entries of $\mathbf{A}, \mathbf{b}, \mathbf{C}, \mathbf{d}$, and $\mathbf{c}$ were first all generated from $\mathcal{N}(0,1)$ (i.i.d.). To guarantee infeasibility of the problem, the last inequality constraint $\mathbf{a}_m^T\mathbf{x} \geq b_m$ was then replaced by $-\mathbf{a}_i^T\mathbf{x} \geq -b_i + \delta$, for some index i randomly selected from $\{1,\ldots,m-1\}$ and $\delta > 0$ taken from $\mathcal{U}(0,1)$. The starting point $\mathbf{x}^0$ was generated by taking i.i.d. entries from $\mathcal{N}(0,1)$.

In Table 1, the averaged iteration counts and computation time over 20 problem instances are reported for IS-CR-MPC. As seen from the table, with the dual estimates generated by (57), the conditions in (56) were satisfied on all tested problem instances within about 10 iterations on average. These results suggest that the proposed Meta-Algorithm IS is capable of providing infeasibility certificates efficiently for infeasible problems. It also is worth noting that



(a) Strongly convex QP: $H \succ \mathbf{0}$



(b) Linear optimization: $H = \mathbf{0}$

Fig. 2: Randomly generated problems with $m = 10\,000$ inequality constraints and no equality constraints. Results are pictured for two sub-classes of problems. In each figure, the x-axis is the number of variables (n) and the y-axis is the iteration count or total computation time, both averaged over the 20 problem instances and plotted in logarithmic scale. In the tests, MPC* and CR-MPC started from a feasible point; IS-CR-MPC started from a randomly generated infeasible point; MOSEK, OSQP, and qpOASES started from their own initial guesses; Two-phase-RPS started from the readily available initial BFS for the auxiliary feasibility problem in phase 1.

no infeasibility certificates were issued in the tests reported in section 6.3.1. (i.e., there were no false positives).

6.4 Support-Vector Machine Training Problems

We tested IS-CR-MPC on CQPs arising in the training of support-vector machine (SVM) classifiers for pattern recognition tasks in high dimensions (see, e.g., [26] and references therein for relevant discussions). In the problems considered here, the training data set takes the form $(\mathbf{P}, \boldsymbol{\ell})$, where $\mathbf{P} \in \mathbb{R}^{\bar{m}\times\bar{n}}$, $\boldsymbol{\ell} \in \mathbb{R}^{\bar{m}}$ and, for $i = 1, \ldots, \bar{m}$, $\mathbf{p}_i$ (transpose of i-th row of $\mathbf{P}$) denotes a pattern that corresponds to a known label $\ell_i \in \{-1, 1\}$. The training process of

n		10		20		50		100		200	
p	$\mathbf{H}$	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
$\frac{n}{2}$	$\succ \mathbf{0}$	7.5	0.04	11.1	0.10	10.0	0.15	10.2	0.35	10.0	1.01
$\frac{n}{2}$	$= \mathbf{0}$	8.5	0.05	10.4	0.08	9.9	0.15	10.1	0.38	10.1	1.08
0	$\succ \mathbf{0}$	10.2	0.07	11.2	0.10	10.4	0.17	10.0	0.34	10.3	0.97
0	$= \mathbf{0}$	10.8	0.07	10.4	0.09	9.8	0.16	10.2	0.35	10.3	0.95

Table 1: Infeasibility detection results with IS-CR-MPC on randomly generated (infeasible) problems with $m = 10\,000$ inequality constraints. In each row, the iteration count and computation time (sec) averaged over 20 problem instances are reported for problems with $n = 10, \ldots, 200$ variables and $\frac{n}{2}$ or 0 equality constraints, in the strongly convex ($\mathbf{H} \succ \mathbf{0}$) or linear ($\mathbf{H} = \mathbf{0}$) sub-classes.

SVMs aims at finding an optimal separating hyperplane (when one exists) in the pattern space, that separates the "+" class patterns (with label $\ell_i = 1$) from the "−" class patterns (with label $\ell_i = -1$) and is equidistant from both classes. Specifically, the goal is to construct a hyperplane

$$\{\mathbf{p} \in \mathbb{R}^{\bar{n}} \colon \langle \mathbf{w}, \mathbf{p} \rangle - \beta = 0\}, \tag{58}$$

under inner product $\langle \cdot, \cdot \rangle$, such that the parameters $\mathbf{w} \in \mathbb{R}^{\bar{n}}$ and $\beta \in \mathbb{R}$ satisfy

$$\operatorname{sign}\{\langle \mathbf{w}, \mathbf{p}_i \rangle - \beta\} = \ell_i\,, \quad i = 1, \ldots, \bar{m}\,, \tag{59}$$

while maximizing the separation margin $\frac{2}{\|\mathbf{w}\|_2}$. When the Euclidean inner product is selected, this amounts to solving

$$\underset{\mathbf{w} \in \mathbb{R}^{\bar{n}}, \beta \in \mathbb{R}}{\text{minimize}} \;\; \frac{1}{2}\|\mathbf{w}\|_2^2 \quad \text{s.t.} \;\; \mathbf{L}(\mathbf{P}\mathbf{w} - \beta\mathbf{1}) \geq \mathbf{1}\,, \tag{60}$$

where $\mathbf{L} := \operatorname{diag}(\boldsymbol{\ell})$. By denoting $\mathbf{x} = [\mathbf{w}; \beta]$, this problem takes the form of (P) with $n = \bar{n} + 1$ optimization variables and $m = \bar{m}$ inequality constraints. Of course, when the given training data is not separable, (60) is infeasible. When this is known (e.g., an infeasibility certificate has been produced by IS-CR-MPC), a constraint-relaxation variable is introduced that allows misclassification, and the objective function is penalized accordingly, viz.

$$\underset{\mathbf{w} \in \mathbb{R}^{\bar{n}}, \beta \in \mathbb{R}, \nu \in \mathbb{R}}{\text{minimize}} \;\; \frac{1}{2}\|\mathbf{w}\|_2^2 + \tau\nu \quad \text{s.t.} \;\; \mathbf{L}(\mathbf{P}\mathbf{w} - \beta\mathbf{1}) + \nu\mathbf{1} \geq \mathbf{1}\,,\;\; \nu \geq 0\,, \tag{61}$$

where $\tau > 0$ is a constant penalty parameter on the relaxation variable $\nu \in \mathbb{R}$. In practice, τ is often determined by cross-validation on the training data. This relaxed problem still takes the form of (P), with $n = \bar{n} + 2$ optimization variables $\mathbf{x} = [\mathbf{w}; \beta; \nu]$ and $m = \bar{m} + 1$ inequality constraints.[19]

[19] Alternatively, following the suggestion made at the very end of section 5, an $(n+1)$-variable problem with feasible start could be solved.

We tested IS-CR-MPC on SVM training for four data sets—`MUSHROOM`, `ISOLET`, `WAVEFORM`, and `LETTER`—from the UCI machine learning repository [24]. As in [20], a lifted version of the data, in a higher-dimensional feature space, with increased likeliness of linear separation was used instead; see [20,26] for details. Such mapping results in `MUSHROOM` and `ISOLET` being separable; `WAVEFORM`, and `LETTER` are not, and the relaxed problem (61) was solved instead.[20] The numbers of features and patterns for the lifted version of each data set are listed in Table 2.

	`MUSHROOM`	`ISOLET`	`WAVEFORM`	`LETTER`
# of features ($\bar{n}$)	276	617	861	153
# of patterns ($\bar{m}$)	8124	7797	5000	20000
separable	Yes	Yes	No	No

Table 2: Problem specifications of the four tested data sets for SVM training.

Remark 6 For these problems, in which the number of patterns is much larger than the number of features, the efficiency of interior-point methods can be improved by approximating the Gram matrix $\mathbf{P}^T\mathbf{P}$ with low-rank matrices as considered in, e.g., [12, 19, 37]. In fact, Meta-Algorithm IS proposed in the present paper could invoke base iterations that use approximate low-rank Gram matrices, such as the constraint-reduced SVM solver tested in [26, section 5.2], which incorporates a low-rank Cholesky factorization with symmetric pivoting [19]. (The convergence of Meta-Algorithm IS is guaranteed as long as the base iteration satisfies Requirements BI1 to BI3 in section 2.3. We did not adopt a low-rank approximation in the CR-MPC base iteration tested here.)

The performance of MOSEK, OSQP, qpOASES, IS-MPC*, and IS-CR-MPC is reported in Fig. 3, where logarithmic scales are used. Here the starting point for the infeasible-start algorithms was $\mathbf{x}^0 = \mathbf{0}$. The results show that, on these imbalanced CQPs, IS-CR-MPC enjoys fastest convergence among the tested solvers. Indeed, compared to the next fastest (MOSEK in all four cases), the speedups for `MUSHROOM`,`ISOLET`, `WAVEFORM`, and `LETTER` were 1.3x, 2.0x, 1.1x, and 2.7x, respectively. The lower speed for `WAVEFORM` is readily explained by the fact that this data set is the most balanced one among the tested data sets.

7 Application to Other Base Iterations

As stressed throughout, the proposed framework applies to many existing (and future) primal-feasible iterations. In this section, for simplicity, we limit

[20] A Matlab-formatted version of these data sets was kindly made available to us by Jin Jung, first author of [26].

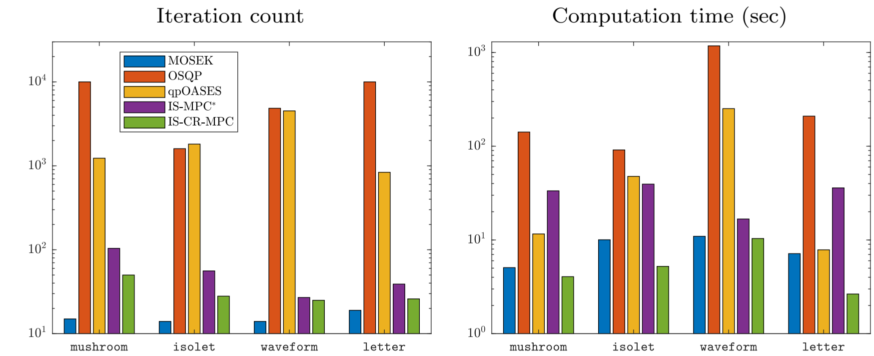


Fig. 3: Support-vector machine training problems. Numerical results of tested algorithms on the `MUSHROOM`, `ISOLET`, `WAVEFORM`, and `LETTER` data sets. In the figures, iteration counts and computation time are reported and plotted in logarithmic scale.

ourselves to base iterations that (i) cater only to linear optimization (LO), i.e., $\mathbf{H} = \mathbf{0}$, and (ii) deal with such problems in standard form,[21] i.e.,

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\ \mathbf{c}^T\mathbf{x} \ \text{ s.t. }\ \mathbf{Cx} = \mathbf{d}, \quad \mathbf{x} \geq \mathbf{0}\,; \tag{LO–P}$$

without loss of generality, we assume that $\mathbf{d} \geq \mathbf{0}$. Existing candidates would include Karmarkar-type interior point algorithms (e.g., [5]) and primal barrier-function methods (e.g., [41]). As per Requirement BI3, asymptotically exact estimates of dual variables should be provided by the base iteration, or be constructible at reasonable cost.

7.1 Framework for Handling Base Iterations that Require Standard Form

Within the proposed infeasible-start framework, the base iteration is applied toward solving the *penalized relaxed problem for* (LO–P), which reduces from $(\tilde{\mathrm{P}}_\varphi)$ to

$$\underset{\mathbf{x}\in\mathbb{R}^n,\mathbf{y}\in\mathbb{R}^p}{\text{minimize}}\ \mathbf{c}^T\mathbf{x} + \varphi\mathbf{1}^T\mathbf{y} \ \text{ s.t. }\ \mathbf{Cx}+\mathbf{y} \geq \mathbf{d}\,,\ -\mathbf{Cx}+\mathbf{y} \geq -\mathbf{d}\,,\ \mathbf{x} \geq \mathbf{0}\,, \tag{LO–$\tilde{\mathrm{P}}_\varphi$}$$

where the $\mathbf{z}$ variable has been done away with since points that satisfy $\mathbf{x} \geq \mathbf{0}$ are trivially available. Transposition from (LO–$\tilde{\mathrm{P}}_\varphi$) to standard form can be effected by wrapping the iteration within an interface (yielding a "wrapped base iteration") that introduces additional slack variables $\mathbf{t}_1$ and $\mathbf{t}_2$ and makes the constraint $\mathbf{y} \geq \mathbf{0}$ explicit, yielding (the subscript in $\mathrm{LO_s}$ stands for "standard

[21] Typical base iterations accept, some require, problems to be expressed in standard form.

form")

$$\underset{\mathbf{x}\in\mathbb{R}^n,\mathbf{y}\in\mathbb{R}^p,\mathbf{t}_1\in\mathbb{R}^p,\mathbf{t}_2\in\mathbb{R}^p}{\text{minimize}} \quad \mathbf{c}^T\mathbf{x} + \varphi\mathbf{1}^T\mathbf{y} \ \text{ s.t. } \ \begin{aligned} \mathbf{C}\mathbf{x} + \mathbf{y} - \mathbf{t}_1 &= \mathbf{d}\,, \\ \mathbf{C}\mathbf{x} - \mathbf{y} + \mathbf{t}_2 &= \mathbf{d}\,, \\ (\mathbf{x},\mathbf{y},\mathbf{t}_1,\mathbf{t}_2) &\geq \mathbf{0}\,. \end{aligned} \tag{LO$_\mathrm{s}$–$\tilde{\mathrm{P}}_\varphi$}$$

Upon return from the base iteration, the new estimate of the primal variables $(\mathbf{x},\mathbf{y})$ in (LO–$\tilde{\mathrm{P}}_\varphi$) is to be provided to `penalty_parameter_update()`,[22] as should appropriate nonnegative estimates of the associated dual variables $(\boldsymbol{\eta},\boldsymbol{\zeta},\boldsymbol{\pi})$ for the three sets of constraints in (LO–$\tilde{\mathrm{P}}_\varphi$). While the primal iterates for (LO–$\tilde{\mathrm{P}}_\varphi$) can be directly taken from the primal sequence $\{(\mathbf{x},\mathbf{y},\mathbf{t}_1,\mathbf{t}_2)\}$ for (LO$_\mathrm{s}$–$\tilde{\mathrm{P}}_\varphi$) generated by the base iteration, the nonnegative dual iterates for (LO–$\tilde{\mathrm{P}}_\varphi$) usually need to be estimated from the information provided by the base iteration. The wrapped base iteration would then be augmented with these processes. A sample wrapped base iteration is given in Pseudocode 3, which describes the necessary components of an interface for wrapping `SFLO_iteration()`, an iteration from a general standard-form linear-optimization primal solver, into an instance of `base_iteration()` in Meta-Algorithm IS (see Pseudocode 1) that aims to solve the corresponding inequality-form problem. A specific example of a wrapped base iteration based on a simplex solver is given in the next subsection.

In order to guarantee that sequences generated by Meta-Algorithm IS converge to the solution set of (LO–P) when the wrapped base iteration is used, the estimated primal sequence for (LO–$\tilde{\mathrm{P}}_\varphi$) needs to be feasible (Requirement BI1), to yield eventual descent (Requirement BI2) and, together with the estimated dual sequence, to satisfy Requirement BI3. Specifically, in the present context, Requirement BI3 requires that, if $\{(\mathbf{x}^\ell,\mathbf{y}^\ell)\}_{\ell\in\hat{K}}$ is bounded, then

$$\max\{\|\mathbf{T}_+^\ell\boldsymbol{\eta}^\ell\|, \|\mathbf{T}_-^\ell\boldsymbol{\zeta}^\ell\|, \|\mathbf{X}^\ell\boldsymbol{\pi}^\ell\|\} \to 0, \ \text{ on } \hat{K} \tag{62a}$$

and

$$\max\{\|\mathbf{c} - \mathbf{C}^T(\boldsymbol{\eta}^\ell - \boldsymbol{\zeta}^\ell) - \boldsymbol{\pi}^\ell\|, \|\boldsymbol{\eta}^\ell + \boldsymbol{\zeta}^\ell - \varphi\|\} \to 0, \ \text{ on } \hat{K}\,. \tag{62b}$$

Here $\{(\mathbf{x}^\ell,\mathbf{y}^\ell)\}$ and $\{(\boldsymbol{\eta}^\ell,\boldsymbol{\zeta}^\ell,\boldsymbol{\pi}^\ell)\}$ are respectively the estimated primal and dual sequences for (LO–$\tilde{\mathrm{P}}_\varphi$), $\mathbf{T}_+^\ell := \mathrm{diag}(\mathbf{C}\mathbf{x}^\ell + \mathbf{y}^\ell - \mathbf{d})$, $\mathbf{T}_-^\ell := \mathrm{diag}(-\mathbf{C}\mathbf{x}^\ell + \mathbf{y}^\ell + \mathbf{d})$, $\mathbf{X}^\ell := \mathrm{diag}(\mathbf{x}^\ell)$, and $\hat{K}$ collects the indices of primal iterates with descending objective function values, as defined in section 2.3.

In the remainder of section 7, we consider an instance of a wrapped base iteration based on the RPS (simplex) solver tested in section 6.3.1 ("Two-phase RPS" in Figure 2).

[22]Slightly modified versions of `penalty_parameter_update()`, of the Requirements PU' of section 4 and, accordingly, of the penalty-parameter updating rule of section 6.1 are to be used here. Specifically, all instances of $\mathbf{z}$ and the associated dual variable $\boldsymbol{\xi}$ must be removed.

Pseudocode 3: Wrapped Base Iteration
$(\mathbb{x}^{\text{out}}, \mathbb{\lambda}^{\text{out}}) \leftarrow$ `base_iteration`$(\varphi, \mathbb{x}^{\text{in}}, \mathbb{\lambda}^{\text{in}})$

Input: penalty parameter φ, primal-dual iterate $(\mathbb{x}^{\text{in}}, \mathbb{\lambda}^{\text{in}})$ for (LO–$\tilde{\text{P}}_\varphi$)
Output: primal-dual iterate $(\mathbb{x}^{\text{out}}, \mathbb{\lambda}^{\text{out}})$ for (LO–$\tilde{\text{P}}_\varphi$)

1 $(\mathbf{x}^{\text{in}}, \mathbf{y}^{\text{in}}, \mathbf{t}_1^{\text{in}}, \mathbf{t}_2^{\text{in}}) \leftarrow$ `iterate_construction`$(\mathbb{x}^{\text{in}}, \mathbb{\lambda}^{\text{in}})$;
/* construct primal iterate for (LO$_\text{s}$–$\tilde{\text{P}}_\varphi$) from the primal-dual iterate $(\mathbb{x}^{\text{in}}, \mathbb{\lambda}^{\text{in}})$ for (LO–$\tilde{\text{P}}_\varphi$). */

2 $([\mathbf{x}^{\text{out}}; \mathbf{y}^{\text{out}}; \mathbf{t}_1^{\text{out}}; \mathbf{t}_2^{\text{out}}], \texttt{info}^{\text{out}}) \leftarrow$ `SFLO_iteration`$(\varphi, [\mathbf{x}^{\text{in}}; \mathbf{y}^{\text{in}}; \mathbf{t}_1^{\text{in}}; \mathbf{t}_2^{\text{in}}])$;
/* `SFLO_iteration()` denotes the application of one iteration from a standard-form linear-optimization solver on (LO$_\text{s}$–$\tilde{\text{P}}_\varphi$), which updates the starting primal point $(\mathbf{x}^{\text{in}}, \mathbf{y}^{\text{in}}, \mathbf{t}_1^{\text{in}}, \mathbf{t}_2^{\text{in}})$ to $(\mathbf{x}^{\text{out}}, \mathbf{y}^{\text{out}}, \mathbf{t}_1^{\text{out}}, \mathbf{t}_2^{\text{out}})$. `SFLO_iteration()` could provide some additional information $\texttt{info}^{\text{out}}$, such as estimates of dual variables in (LO$_\text{s}$–$\tilde{\text{P}}_\varphi$). */

3 $(\mathbb{x}^{\text{out}}, \mathbb{\lambda}^{\text{out}}) \leftarrow$ `iterate_estimation`$(\mathbf{x}^{\text{out}}, \mathbf{y}^{\text{out}}, \mathbf{t}_1^{\text{out}}, \mathbf{t}_2^{\text{out}}, \texttt{info}^{\text{out}})$;
/* estimate feasible primal iterates and nonnegative dual iterates for (LO–$\tilde{\text{P}}_\varphi$) using the primal iterate for (LO$_\text{s}$–$\tilde{\text{P}}_\varphi$) as well as additional information from `SFLO_iteration()`. */

7.2 Application to a Revised Primal Simplex Iteration

Consider solving (LO–P) with infeasible start using Meta-Algorithm IS with a (wrapped) base iteration specific to RPS. A short description of the RPS method and a pseudocode for an RPS iteration are given in Appendix B. Meta-Algorithm IS allows for bypassing the typical two-phase approach for simplex methods when no initial basic feasible solution (BFS) for (LO–P) is readily available.

Specifically, in the context of Meta-Algorithm IS, the RPS method is applied to solve the penalized relaxed problem (LO$_\text{s}$–$\tilde{\text{P}}_\varphi$), for which an initial BFS $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{t}_1^0, \mathbf{t}_2^0) := (\mathbf{0}, \mathbf{d}, \mathbf{0}, 2\mathbf{d})$ is readily available (recall that $\mathbf{d} \geq \mathbf{0}$). As discussed in section 7.1, a wrapped base iteration as in Pseudocode 3 is needed to take care of the transformation from the inequality form problem (LO–$\tilde{\text{P}}_\varphi$) to the standard-form problem (LO$_\text{s}$–$\tilde{\text{P}}_\varphi$), and to produce estimates of the dual variables. The wrapped base iteration tailored for the RPS method, referred to as "Base Iteration RPS", is detailed in Pseudocode 4 (an instance of Pseudocode 3), in which steps 1–2 correspond to `iterate_construction()` (step 1 in Pseudocode 3), step 3 replaces the general standard-form linear-optimization iteration `SFLO_iteration()` with the specific `RPS_iteration()` given in Pseudocode 6 in Appendix B (setp 2 in Pseudocode 3), and steps 4–7 correspond to `iterate_estimation()` (step 3 in Pseudocode 3).

Pseudocode 4: Base Iteration RPS (Wrapped)

$(\mathbb{x}^{\text{out}}, \mathbb{\lambda}^{\text{out}}) \leftarrow$ `base_iteration`$(\varphi, \mathbb{x}^{\text{in}}, \mathbb{\lambda}^{\text{in}})$

Input: penalty parameter φ, primal-dual iterate $(\mathbb{x}^{\text{in}}, \mathbb{\lambda}^{\text{in}})$

Output: primal-dual iterate $(\mathbb{x}^{\text{out}}, \mathbb{\lambda}^{\text{out}})$

`// --- construct the primal iterate for` $(\text{LO}_\text{s}\text{–}\tilde{\text{P}}_\varphi)$ `---`

$[\mathbf{x}^{\text{in}}; \mathbf{y}^{\text{in}}] \leftarrow \mathbb{x}^{\text{in}}$;

$\mathbf{t}_1^{\text{in}} \leftarrow \mathbf{C}\mathbf{x}^{\text{in}} + \mathbf{y}^{\text{in}} - \mathbf{d}, \;\; \mathbf{t}_2^{\text{in}} \leftarrow -\mathbf{C}\mathbf{x}^{\text{in}} + \mathbf{y}^{\text{in}} + \mathbf{d}$;

`// --- take one RPS iterate on` $(\text{LO}_\text{s}\text{–}\tilde{\text{P}}_\varphi)$ `---`

$([\mathbf{x}^{\text{out}}; \mathbf{y}^{\text{out}}; \mathbf{t}_1^{\text{out}}; \mathbf{t}_2^{\text{out}}], [\boldsymbol{\lambda}_1^{\text{out}}; \boldsymbol{\lambda}_2^{\text{out}}]) \leftarrow$ `RPS_iteration`$(\varphi, [\mathbf{x}^{\text{in}}; \mathbf{y}^{\text{in}}; \mathbf{t}_1^{\text{in}}; \mathbf{t}_2^{\text{in}}])$;

`/* RPS_iteration() denotes the application of one RPS iteration to solve` $(\text{LO}_\text{s}\text{–}\tilde{\text{P}}_\varphi)$`, which updates the BFS` $(\mathbf{x}^{\text{in}}, \mathbf{y}^{\text{in}}, \mathbf{t}_1^{\text{in}}, \mathbf{t}_2^{\text{in}})$ `to another BFS` $(\mathbf{x}^{\text{out}}, \mathbf{y}^{\text{out}}, \mathbf{t}_1^{\text{out}}, \mathbf{t}_2^{\text{out}})$ `and estimates dual variables` $(\boldsymbol{\lambda}_1^{\text{out}}, \boldsymbol{\lambda}_2^{\text{out}})$ `associated to the two sets of equality constraints in` $(\text{LO}_\text{s}\text{–}\tilde{\text{P}}_\varphi)$`. */`

`// --- estimate the primal iterate for` $(\text{LO–}\tilde{\text{P}}_\varphi)$ `---`

$\mathbb{x}^{\text{out}} \leftarrow [\mathbf{x}^{\text{out}}; \mathbf{y}^{\text{out}}]$;

`// --- estimate the dual iterate for` $(\text{LO–}\tilde{\text{P}}_\varphi)$ `---`

$\boldsymbol{\omega} \leftarrow \boldsymbol{\lambda}_1^{\text{out}} + \boldsymbol{\lambda}_2^{\text{out}}$;

$\boldsymbol{\eta}^{\text{out}} = [\frac{1}{2}(\varphi\mathbf{1} + \boldsymbol{\omega})]_+, \;\; \boldsymbol{\zeta}^{\text{out}} = [\frac{1}{2}(\varphi\mathbf{1} - \boldsymbol{\omega})]_+, \;\; \boldsymbol{\pi}^{\text{out}} = [\mathbf{c} - \mathbf{C}^T\boldsymbol{\omega}]_+$;

$\mathbb{\lambda}^{\text{out}} \leftarrow [\boldsymbol{\eta}^{\text{out}}; \boldsymbol{\zeta}^{\text{out}}; \boldsymbol{\pi}^{\text{out}}]$;

The analysis in section 4 guarantees convergence of Meta-Algorithm IS to the solution set of (LO–P) when, e.g., the penalty-parameter updating rule of section 6.1 is used, provided that the base iteration satisfies Requirements BI1 to BI3. Note however that Requirements BI1 to BI3 assume that the base iteration generates infinite sequences, while simplex methods typically stop after finitely many iterations either with an exactly optimal solution for the problem at hand or with an indication that the problem is unbounded. This is catered to in the wrapper as follows. When an optimal solution to $(\text{LO–}\tilde{\text{P}}_\varphi)$ has been reached in `RPS_iteration()` (i.e., line 4 in Pseudocode 6 in Appendix B is reached), Base Iteration RPS in Pseudocode 4 returns $(\mathbb{x}^*, \mathbb{\lambda}^*)$, a primal-dual solution to $(\text{LO–}\tilde{\text{P}}_\varphi)$. Repeatedly applying Base Iteration RPS on the same problem starting from $(\mathbb{x}^*, \mathbb{\lambda}^*)$ then results in an infinite constant sequence $\{(\mathbb{x}^*, \mathbb{\lambda}^*)\}$ that can be used to verify Requirements BI1 to BI3. On the other hand, when unboundedness is detected in `RPS_iteration()` (line 9 in Pseudocode 6, Appendix B), Base Iteration RPS returns to Meta-Algorithm IS an infinite feasible sequence[23] with decreasing and unbounded cost value, while bypassing `RPS_iteration()` until φ is increased by `penalty_parameter_update()`. We do not show this in Pseudocode 4 to keep the presentation simple.

With the constructed infinite sequence, we now verify that Pseudocode 4 indeed satisfies Requirements BI1 to BI3, under the assumption that the primal sequence starts from initial point $\mathbb{x}^0 = [\mathbf{x}^0; \mathbf{y}^0]$ taken from a BFS $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{t}_1^0, \mathbf{t}_2^0)$ for (LO$_\mathrm{s}$–$\tilde{\mathrm{P}}_\varphi$), e.g., $\mathbb{x}^0 = [\mathbf{0}; \mathbf{d}]$ from the known BFS $(\mathbf{0}, \mathbf{d}, \mathbf{0}, 2\mathbf{d})$. Requirement BI1 is automatically satisfied (step 4 in Pseudocode 4). Requirement BI2 also is automatically satisfied since (i) `RPS_iteration()` is taken from a descent method, (ii) (LO–$\tilde{\mathrm{P}}_\varphi$) and (LO$_\mathrm{s}$–$\tilde{\mathrm{P}}_\varphi$) share the same objective function, and (iii) the primal iterate $\mathbb{x}^\ell$ for (LO–$\tilde{\mathrm{P}}_\varphi$) is either taken from the corresponding iterate $(\mathbf{x}^\ell, \mathbf{y}^\ell, \mathbf{t}_1^\ell, \mathbf{t}_2^\ell)$ for (LO$_\mathrm{s}$–$\tilde{\mathrm{P}}_\varphi$) (step 4 in Pseudocode 4), or constructed to drive the objective function value to $-\infty$. As for Requirement BI3, there are two cases. If the constructed infinite sequence is eventually constant at entry $(\mathbb{x}^*, \mathbb{\lambda}^*)$, then the entry is recovered (in steps 4–7 in Pseudocode 4) from the output $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{t}_1^*, \mathbf{t}_2^*, \boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*)$ of `RPS_iteration()`. Since $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{t}_1^*, \mathbf{t}_2^*, \boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*)$ satisfies the Karush-Kuhn-Tucker optimality conditions for (LO$_\mathrm{s}$–$\tilde{\mathrm{P}}_\varphi$), the estimated primal-dual pair $(\mathbb{x}^*, \mathbb{\lambda}^*)$ satisfies the KKT optimality conditions of (LO–$\tilde{\mathrm{P}}_\varphi$), thus (62) holds and Requirement BI3 is satisfied. On the other hand, when the problem is detected to be unbounded, the constructed infinite primal sequence is unbounded, which automatically satisfies Requirement BI3.

As a "proof of concept", we carried out a preliminary numerical test on standard-form linear-optimization problems (LO–P). In the test, the infeasible-start Meta-Algorithm IS with RPS base iteration (IS-RPS) was compared against the typical two-phase approach with the same RPS solver (Two-phase-RPS), which is included in the comparison to IS-CR-MPC in section 6.3.1 (Figure 2b). In IS-RPS, `base_iteration()` was implemented as specified in Pseudocode 4, and `penalty_parameter_update()` was slightly modified from the one proposed in section 6.1, as described in footnote 22. The two methods were tested on randomly generated problems of the form (LO–P), which were formulated by taking the dual of the randomly generated inequality-form linear optimization problems ($\mathbf{H} = \mathbf{0}$) considered in section 6.3.1 to guarantee a nonempty and bounded optimal solution set. All tested problems were pre-

[23] This unbounded feasible primal sequence can be constructed by taking $\{\mathbb{x}^\ell\}$ such that $\mathbb{x}_B^\ell := \hat{x}_B + t^\ell \Delta\hat{x}_B$ and $\mathbb{x}_{B^c}^\ell := t^\ell e_j$, where $\hat{x}$, $\Delta\hat{x}_B$, and B are respectively the primal BFS, direction, and basis at which unboundedness is detected in Iteration RPS (Pseudocode 6), e_j is a standard unit vector that takes value 1 at the entering column index j and 0 elsewhere, and $t^\ell \to \infty$ as $\ell \to \infty$. Here the direction $\Delta\hat{x}_B$ and the entering column index j would be appended to the output of `RPS_iteration()` as part of $\texttt{info}^{\mathrm{out}}$ (together with $\boldsymbol{\lambda}_1^{\mathrm{out}}$ and $\boldsymbol{\lambda}_2^{\mathrm{out}}$) when unboundedness is detected. In Meta-Algorithm IS, this unbounded sequence would trigger `penalty_parameter_update()` to increase the penalty parameter φ. After the increment of φ, the primal variable then needs to be reinitialized to the BFS $\hat{x}$ (at which unboundedness was detected) so that `RPS_iteration()` can be applied to solve the updated, possibly bounded problem.

processed so that $\mathbf{d} \geq \mathbf{0}$ (by flipping the signs on both sides of the equality constraints) as assumed at the top of section 7.1.

The test was performed on problems of size $n = 1000$ and $p = 5$, 10, 20, 50, 100, 200, 500, and 20 instances of randomly generated problems were tested for each size. The iteration counts averaged over the 20 problem instances are reported in Table 3, where the counts for Two-phase-RPS are sums of iteration counts in the two phases. From the results, we observe that IS-RPS required 8–24% fewer iterations than the Two-phase-RPS solver for the highly imbalanced cases ($p = 5$, 10, 20, 50). This shows a potential benefit in solving the relaxed penalized problem over solving the combination of feasibility problem and original optimization problem on these imbalanced problems. However, in order to translate the lower iteration count into less computation time, it will be necessary to mitigate the additional computation cost resulting from constraint augmentation in the penalized relaxed problem; this should be achievable by means of a careful implementation that utilizes the structure of the relaxed penalized problem. Finally, Table 3 also shows that the advantage of IS-RPS decreases as the problem becomes better balanced. In particular, IS-RPS requires more iterations to converge than Two-phase-RPS on problems with only twice as many variables than equality constraints ($p = 500$). This is due to the fact that the penalized relaxed problem ($\mathrm{LO_s}$–$\tilde{\mathrm{P}}_\varphi$) has $3p$ more optimization variables than the original problem (LO–P). While the effect of these additional variables is rather insignificant in the highly imbalanced case ($p \ll n$), the slowdown of convergence likely will become noticeable when the imbalance subsides. This issue could potentially be mitigated by reducing the dimension of ($\mathrm{LO_s}$–$\tilde{\mathrm{P}}_\varphi$), e.g., eliminating the $\mathbf{y}$ variable in ($\mathrm{LO_s}$–$\tilde{\mathrm{P}}_\varphi$) by setting $\mathbf{y} = \frac{\mathbf{t}_1+\mathbf{t}_2}{2}$. Investigation of this possibility is beyond the scope of the present paper.

Averaged iteration counts for problems with $n = 1000$ and various p

Method	Number of equality constraints (p)						
	5	10	20	50	100	200	500
Two-phase-RPS	35.0	72.2	159.6	365.3	713.2	1438.3	3047.9
IS-RPS	30.7	63.0	128.3	335.9	697.7	1406.9	3106.1

Table 3: Iteration counts of Two-phase-RPS and IS-RPS methods for solving randomly generated problems (LO–P) of size $n = 1000$ and various p. The reported iteration counts are averaged over the 20 problem instances.

8 Conclusion

An exact-penalty-based framework that allows for infeasible starts in solving CQPs (including linear optimization problems) with a feasible-start method was proposed and analyzed. With negligible additional computational cost per iteration, an infeasibility test is included that provides an infeasibility certificate when the problem at hand is indeed infeasible. The framework was tested on constrained-reduced MPC. Numerical results suggest that, on imbalanced CQPs, infeasible-start CR-MPC is significantly faster than OSQP, qpOASES, and MOSEK, in spite of their speed advantage of executing compiled code. It is also confirmed that constraint reduction is very powerful on such problems. Brief testing on a simplex iteration also shows promise.

Appendix

Two algorithms considered in this paper are presented here for the sake of completeness. Some notation used in the appendix is "local", i.e., only applies to this appendix; indeed, some of the same symbols are used in the main body of the paper for unrelated quantities.

A Feasible-Start Algorithm CR-MPC

In this appendix, we present a brief description of Algorithm CR-MPC in [28], as applied to a generic inequality-constrained CQP of the form (6). We refer the reader to [28] for the complete version of Algorithm CR-MPC in full detail.

For ease of reference, we restate the generic inequality-constrained CQP (6) here:

$$\underset{x}{\text{minimize}}\ f(x) := \frac{1}{2}x^T H x + c^T x \ \text{ s.t. } \ Ax \geq b. \tag{63}$$

Algorithm CR-MPC is a feasible-start, descent CQP variant of S. Mehrotra's predictor–corrector algorithm [31]; computational efficiency is enhanced on imbalanced problems by solving a mere *approximate* Newton-KKT system for search directions ("constraint reduction"). Specifically, the Newton-KKT system is approximated by including only a selected subset of constraints: an which aims to estimate of the active constraint set at a solution to (63). Also, an adaptive positive definite regularization W of the original Hessian H is substituted in that system.

Let Q denote the index set of the selected set of constraints, then the affine-scaling search direction is given by the solution of the approximate Newton-KKT system for (63)

$$\begin{bmatrix} W & -A_Q^T & \mathbf{0} \\ A_Q & \mathbf{0} & -I \\ \mathbf{0} & S_Q & \Lambda_Q \end{bmatrix} \begin{bmatrix} \Delta x^{\mathrm{a}} \\ \Delta \lambda_Q^{\mathrm{a}} \\ \Delta s_Q^{\mathrm{a}} \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + (A_Q)^T \lambda_Q \\ \mathbf{0} \\ -S_Q \lambda_Q \end{bmatrix}, \tag{64}$$

and the corrector direction is computed by solving

$$\begin{bmatrix} W & -A_Q^T & \mathbf{0} \\ A_Q & \mathbf{0} & -I \\ \mathbf{0} & S_Q & \Lambda_Q \end{bmatrix} \begin{bmatrix} \Delta x^{\mathrm{c}} \\ \Delta \lambda_Q^{\mathrm{c}} \\ \Delta s_Q^{\mathrm{c}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \sigma \mu_{(Q)} \mathbf{1} - \Delta S_Q^{\mathrm{a}} \Delta \lambda_Q^{\mathrm{a}} \end{bmatrix} \tag{65}$$

where $\mu_{(Q)} := \begin{cases} s_Q^T \lambda_Q / |Q|\,, & \text{if } Q \neq \emptyset \\ 0\,, & \text{otherwise} \end{cases}$ and $\sigma = (1 - \alpha^{\mathrm{a}})^3$ with α^{a} the maximum feasible step for the affine-scaling direction. A pseudocode for an iteration in Algorithm CR-MPC is given in Pseudocode 5, which can serve as `base_iteration()` in Meta-Algorithm IS with minimal modifications that tailor the code for penalized relaxed problems of the form (P_φ).

Pseudocode 5:
Iteration CR-MPC: A Constraint-Reduced variant of MPC iteration

Input: strictly feasible primal point x for (63), dual point $\lambda > \mathbf{0}$,
Output: updated strictly feasible primal point x^+, dual estimate $\tilde{\lambda}^+$

Set $s := Ax - b$. Select an index set Q that estimates active constraints at a solution. Compute an adaptively regularized Hessian $W \succ \mathbf{0}$ from original Hessian H;
Solve (64) for the affine-scaling direction $(\Delta x^{\rm a},\, \Delta\lambda_Q^{\rm a},\, \Delta s_Q^{\rm a})$;
Solve (65) for the corrector direction $(\Delta x^{\rm c},\, \Delta\lambda_Q^{\rm c},\, \Delta s_Q^{\rm c})$;
Compute a mixing variable γ based on the two directions and obtain the search direction by setting $(\Delta x,\, \Delta\lambda_Q) := (\Delta x^{\rm a},\, \Delta\lambda_Q^{\rm a}) + \gamma\,(\Delta x^{\rm c},\, \Delta\lambda_Q^{\rm c})$;
Set auxiliary dual variable $\tilde{\lambda}_Q^+ = \lambda_Q + \Delta\lambda_Q$, and $\tilde{\lambda}_{Q^{\rm c}}^+ = \mathbf{0}$;
Compute primal and dual steps $\alpha_{\rm p}$ and $\alpha_{\rm d}$ that maintain strict feasibility;
Update the primal and dual variables:

$$x^+ := x + \alpha_{\rm p}\Delta x\,, \quad \lambda_Q^+ := \lambda_Q + \alpha_{\rm d}\Delta\lambda_Q, \tag{66}$$

with $\lambda_{Q^{\rm c}}^+$ set to some well-centered estimation;

B Revised Primal Simplex (RPS) Algorithm

In this appendix, we give a brief description of the revised primal simplex method, which is considered in section 6.3.1 in the comparison to the CR-MPC method discussed in Appendix A, as well as in section 7 as a base iteration in the IS framework. We consider the RPS method that handles linear optimization problems in primal standard form

$$\underset{x}{\text{minimize}}\; c^T x \;\; \text{s.t.}\;\; Cx = d,\; x \geq \mathbf{0}. \tag{67}$$

The RPS method considered here starts from an initial primal BFS for (67) and, at each iteration, it reduces the objective function value by moving to an adjacent primal BFS with lower value. Specifically, each RPS iteration forms a descent search direction by adding an entering column to the basis associated to the BFS, and takes a step towards the descent direction until one of the BFS entries becomes zero, the index of which is then identified as the leaving column and is removed from the basis. Pseudocode 6 presents the main steps of an RPS iteration, which can be used as `iteration_RPS()` in the wrapped RPS base iteration in Pseudocode 4 by incorporating the penalty parameter φ into the input.

References

1. Anczak, T.: Exact penalty functions method for mathematical programming problems involving invex functions. European J. of Operational Research **198**, 29–36 (2009)
2. Andersen, E.D., Andersen, K.D.: The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In: High performance optimization, pp. 197–232. Springer (2000)
3. Andersen, E.D., Roos, C., Terlaky, T.: On implementing a primal-dual interior-point method for conic quadratic optimization. Mathematical Programming **95**(2), 249–277 (2003)
4. Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi – A software framework for nonlinear optimization and optimal control. Mathematical Programming Computation **11**(1), 1–36 (2019). DOI 10.1007/s12532-018-0139-4
5. Anstreicher, K.: A standard form variant, and safeguarded linesearch, for the modified Karmarkar algorithm. Mathematical Programming **47**, 337–351 (1990)

Pseudocode 6:
Iteration RPS: A revised primal simplex iteration

Input: primal BFS x for (67), basis index set $B := \{i : x_i > 0\}$
Output: updated primal BFS x^+, dual estimate λ

$\lambda \leftarrow (C_B^T)^{-1} c_B$; `// estimate dual variable`
$s_{B^c} \leftarrow c_{B^c} - C_{B^c}^T \lambda$; `// compute dual slack values`
if $s_{B^c} \geq \mathbf{0}$ **then**
  $x^+ \leftarrow x$; `// problem solved`
else
  choose j such that $(s_{B^c})_j < 0$; `// choose an entering column`
  $\Delta x_B \leftarrow C_B^{-1} C_{B^c} e_j$; `// compute the primal direction`
  if $\Delta x_B \geq \mathbf{0}$ **then**
    $x^+ \leftarrow x$; `// problem is unbounded`
  else
    $i \leftarrow \mathrm{argmin}\{\frac{(x_B)_k}{-(\Delta x_B)_k} : (\Delta x_B)_k < 0\}$; `//set the leaving column`
    $t^* \leftarrow \frac{(x_B)_i}{-(\Delta x_B)_i}$; `// set the step size`
    $x_B^+ \leftarrow x_B + t^* \Delta x_B$; `// update primal variable`
    $x_{B^c}^+ \leftarrow t^* e_j$;
    $B^+ \leftarrow (B \cup \{j\}) \setminus \{i\}$; `// update the basis`

6. Benson, H.Y., Shanno, D.F.: An exact primal-dual penalty method approach to warmstarting interior-point methods for linear programming. Computational Optimization and Applications **38**(3), 371–399 (2007). DOI 10.1007/s10589-007-9048-6. URL `https://doi.org/10.1007/s10589-007-9048-6`
7. Bertsekas, D., Nedic, A., Ozdaglar, A.: Convex Analysis and Optimization. Athena Scientific (2003)
8. Bertsimas, D., Tsitsiklis, J.: Introduction to Linear Optimization. Athena (1997)
9. Boyd, S., Parikh, N., Chu, E.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc (2011)
10. Byrd, R., Curtis, F., Nocedal, J.: Infeasibility detection and SQP methods for nonlinear optimization. SIAM J. Optimization **4**(5), 2281–2299 (2010)
11. Byrd, R.H., Nocedal, J., Waltz, R.A.: Steering exact penalty methods for nonlinear programming. Optimization Methods and Software **23**(2), 197–213 (2008). DOI 10.1080/10556780701394169. URL `https://doi.org/10.1080/10556780701394169`
12. Chapelle, O.: Training a support vector machine in the primal. Neural computation **19**(5), 1155–1178 (2007)
13. Coleman, T., Conn, A.: Nonlinear programming via an exact penalty method: asymptotic analysis. Math. Programming **24**, 123–136 (1982)
14. Conn, A.: Constrained optimization using a nondifferentiable penalty function. SIAM J. on Numerical Analysis **10**(4), 760–784 (1973)
15. Conn, A.R., Mongeau, M.: Discontinuous piecewise linear optimization. Mathematical Programming **80**(3), 315–380 (1998). DOI 10.1007/BF01581171. URL `https://doi.org/10.1007/BF01581171`
16. Coulibaly, Z., Orban, D.: An l1 elastic interior-point method for mathematical programs with complementarity constraints. SIAM J. Optimization **22**, 187–211 (2012)
17. Di Pillo, G., Facchinei, F., Grippo, L.: An RQP algorithm using a differentiable exact penalty function for inequality constrained problems. Math. Programming **25**, 49–68 (1992)
18. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. Mathematical Programming Computation **6**(4), 327–363 (2014)

19. Fine, S., Scheinberg, K.: Efficient SVM training using low-rank kernel representations. Journal of Machine Learning Research **2**(Dec), 243–264 (2001)
20. Gertz, E.M., Griffin, J.D.: Support vector machine classifiers for large data sets. Tech. Rep. ANL/MCS-TM-289, Argonne National Laboratory (2005). DOI 10.2172/881587
21. Hassan, H., Baharum, A.: Generalized logarithmic penalty function method for solving smooth nonlinear programming involving invex functions. Arab J. of Basic and Applied Sciences **26**, 202–214 (2019)
22. He, M.: Infeasible constraint reduction for linear and convex quadratic optimization. Ph.D. thesis, University of Maryland (2011). URL: http://hdl.handle.net/1903/12772
23. He, M.Y., Tits, A.L.: Infeasible constraint-reduced interior-point methods for linear optimization. Optim. Methods Softw. **27**(4-5), 801–825 (2012). DOI 10.1080/10556788.2011.589056. URL http://dx.doi.org/10.1080/10556788.2011.589056
24. Hettich, C.B.S., Merz, C.: UCI repository of machine learning databases (1998). URL http://www.ics.uci.edu/~mlearn/MLRepository.html
25. Hungerländer, P.: Algorithms for convex quadratic programming. Ph.D. thesis, Alpen-Adria-Universität Klagenfurt (2009)
26. Jung, J.H., O'Leary, D.P., Tits, A.L.: Adaptive constraint reduction for training support vector machines. Electron. T. Numer. Ana. **31**, 156–177 (2008)
27. Jung, J.H., O'Leary, D.P., Tits, A.L.: Adaptive constraint reduction for convex quadratic programming. Comput. Optim. Appl. **51**(1), 125 – 157 (2012)
28. Laiu, M., Tits, A.: A constraint-reduced MPC algorithm for convex quadratic programming, with a modified active set identification scheme. Computational Optimization and Applications **72**, 727–768 (2019). DOI DOI:10.1007/s10589-019-00058-0
29. Matušek, J., Gärtner, R.: Understanding and Using Linear Programming. Springer New York (2007)
30. Mayne, D.Q., Polak, E.: Feasible directions algorithms for optimization problems with equality and inequality constraints. Mathematical Programming **11**(1), 67–80 (1976). DOI 10.1007/BF01580371. URL https://doi.org/10.1007/BF01580371
31. Mehrotra, S.: On the implementation of a primal-dual interior point method. SIAM J. Optim. **2**(4), 575–601 (1992)
32. Melo, T., Matias, J., Monteiro, M.: A penalty methods for solving the MPCC problem. J. of Mathematical Analysis **7**(1), 91–101 (2016)
33. Monteiro, R., Adler, I.: Interior path following primal–dual algorithms. Part II: Convex quadratic programming. Math. Programming **44**, 43–66 (1989)
34. Polak, E., He, L.: Unified steerable phase I – phase II method of feasible directions for semi-infinite optimization. Journal of Optimization Theory and Applications **69**(1), 83–107 (1991). DOI 10.1007/BF00940462. URL https://doi.org/10.1007/BF00940462
35. Polak, E., Tits, A.L.: A globally convergent, implementable multiplier method with automatic penalty limitation. Applied Mathematics and Optimization **6**(1), 335–360 (1980). DOI 10.1007/BF01442901. URL https://doi.org/10.1007/BF01442901
36. Polak, E., Trahan, R., Mayne, D.Q.: Combined phase I – phase II methods of feasible directions. Mathematical Programming **17**(1), 61–73 (1979). DOI 10.1007/BF01588225. URL https://doi.org/10.1007/BF01588225
37. Schölkopf, B., Smola, A.J., Bach, F., et al.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press (2002)
38. Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: OSQP: an operator splitting solver for quadratic programs. Mathematical Programming Computation **12**(4), 637–672 (2020). DOI 10.1007/s12532-020-00179-2. URL https://doi.org/10.1007/s12532-020-00179-2
39. Tits, A., Wächter, A., Bakhtiari, S., Urban, T., Lawrence, C.: A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties. SIAM J. Optimiz. **14**(1), 173–199 (2003). DOI 10.1137/S1052623401392123. URL https://doi.org/10.1137/S1052623401392123
40. Winternitz, L.: Primal-dual interior-point algorithms for linear programming problems with many inequality constraints. Ph.D. thesis, University of Maryland (2010). URL: http://hdl.handle.net/1903/10400
41. Wright, S.: On the convergence of the Newton/log-barrier method. Math. Programming pp. 71–100 (2001)

42. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM (1997)
43. Ye, Y.: On homogeneous and self-dual algorithms for LCP. Math. Programming **76**, 211–221 (1996)
44. Ye, Y., Todd, M., Mizuno, S.: An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. Mathematics of Operations Research **19**, 53–67 (1994)