SOLUTION OF MONOTONE COMPLEMENTARITY AND GENERAL CONVEX PROGRAMMING PROBLEMS USING A MODIFIED POTENTIAL REDUCTION INTERIOR POINT METHOD*

KUO-LING HUANG AND SANJAY MEHROTRA†

Abstract. We present a homogeneous algorithm equipped with a modified potential function for the monotone complementarity problem. We show that this potential function is reduced by at least a constant amount if a scaled Lipschitz condition is satisfied. A practical algorithm based on this potential function is implemented in a software package named `iOptimize`. The implementation in `iOptimize` maintains global linear polynomial-time convergence properties while achieving practical performance. When compared with a mature software package `MOSEK` (barrier solver version `6.0.0.106`), `iOptimize` solves convex quadratic programming problems, convex quadratically constrained quadratic programming problems, and general convex programming problems in fewer iterations. Moreover, several problems for which `MOSEK` fails are solved to optimality. We also find that `iOptimize` seems to detect infeasibility more reliably than general nonlinear solvers `Ipopt` (version `3.9.2`) and `Knitro` (version `8.0`).

Key words. quadratic programs, quadratically constrained quadratic programs, convex programs, homogeneous algorithms, interior point methods

AMS subject classifications. 90C06, 90C20, 90C25, 90C30, 90C33, 90C51, 49M15

1. Introduction.

In this paper we consider a monotone complementarity problem (MCP) of the following form

$$s = f(x) \tag{1.1}$$

$$0 \le s \perp x \ge 0, \tag{1.2}$$

where $x, s \in R^n$, $f(x)$ is a continuous monotone mapping from $R^n_+ := \{x \in R^n \mid x \ge 0\}$ to R^n, and the notation $0 \le s \perp x \ge 0$ means that $(x, s) \ge 0$ and that $x^T s = 0$. The requirement $x^T s = 0$ is called complementarity condition. Then, for every $x^1, x^2 \in R^n_+$, we have

$$(x^1 - x^2)^T (f(x^1) - f(x^2)) \ge 0. \tag{1.3}$$

Let $\nabla f(x)$ denote the Jacobian matrix of $f(x)$. If $\nabla f(x)$ is positive semidefinite for all $x > 0$, that is,

$$h^T \nabla f(x) h \ge 0, \forall x > 0, h \in R^n, \tag{1.4}$$

then $f(x)$ is a continuous monotone mapping.

$(x, s) \subset R^{2n}_+$ is called a *feasible* solution of MCP if (1.1) is satisfied. It is called an *optimal or complementary* solution if both (1.1) and (1.2) are satisfied. If MCP has a complementary solution (x^*, s^*), then it has a maximal complementary solution where the number of positive components of (x^*, s^*) is maximal.

Given an initial point $(x^0, s^0) \subset R^{2n}_{++} := \{x \in R^n \mid x > 0\ \}$, we would like to compute a bounded sequence $\langle x^k, s^k \rangle \subset R^{2n}_{++}, k = 0, 1, \ldots$, such that

$$\lim_{k\to\infty} s^k - f(x^k) \to 0, \quad \text{and} \quad \lim_{k\to\infty} (x^k)^T s^k \to 0. \tag{1.5}$$

*The research of both authors was partially supported by grant DOE-SP0011568 and ONR N00014-09-10518, N00014-210051

†Dept. of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 60208, `Email: mehrotra@iems.northwestern.edu`

Let $X := diag(x_1, \dots, x_n)$, and let

$$\nu_f(\cdot) : (0,1) \mapsto (1,\infty)$$

be a monotone increasing function. We say that f satisfies a scaled Lipschitz condition (SLC) if

$$\|X(f(x+h) - f(x) - \nabla f(x)h)\|_1 \le \nu_f(\beta) h^T \nabla f(x) h, \tag{1.6}$$

whenever

$$h \in R^n, x \in R^n_{++}, \|X^{-1}h\|_\infty \le \beta < 1,$$

where $\beta \in (0,1)$ is a constant. The analysis in this paper assumes that f satisfies SLC. Such a condition for MCP was used by Potra and Ye [31] to develop an interior point method (IPM) for MCP. In their analysis Potra and Ye assumed that the set of strictly feasible points $\{x \in R^n \mid x > 0, f(x) > 0\}$ is nonempty, and a feasible starting point $x^0 > 0$ and $s^0 = f(x^0) > 0$ is available. Under these assumptions Potra and Ye [31] developed an IPM that achieves a constant reduction of the primal-dual potential function

$$\phi_\rho(x,s) = \rho \log x^T s - \sum_{j=1}^{n} \log x_j s_j, \tag{1.7}$$

for $n + \sqrt{n} \le \rho \le 2n$.

This type of potential function was introduced by Tanabe [35] and Todd and Ye [36] for LP and further extended for linear complementarity problems by Kojima *et al.* [22]. It satisfies the following inequality

$$(\rho - n)\log x^T s \le \phi_\rho(x,s) - n \log n,$$

and hence it follows that

$$x^T s \le \epsilon \text{ whenever } \phi_\rho(x,s) \le n\log n - (\rho - n)|\log \epsilon|.$$

Therefore, starting from a feasible point, if this potential function is reduced by at least a positive constant at each interior point iteration, then we will have $x^T s \le \epsilon$ after at most $\mathcal{O}(\phi_\rho(x^0, s^0) - n\log n + (\rho - n)\,|\log \epsilon|)$ iterations, where ϵ is the complementarity error.

However, in practice an initial feasible point x^0 for optimization problems may not be available. To handle this issue, Ye *et al.* [39] developed a homogeneous IPM for linear programming (LP) that can start from any (infeasible) interior point and achieves the best known convergence complexity without introducing a big M constant. Moreover, it detects the infeasibility of problems unambiguously. Xu *et al.* [38] developed a simplified (and equivalent) version of the homogeneous IPM, and showed that with similar computational efforts, their method can be more reliable than conventional primal-dual IPM while solving infeasible problems in `Netlib` [8] LP benchmark library.

Andersen *et al.* [10] implemented a homogeneous IPM for solving second order conic programming problems based on the work of Nesterov and Todd [28]. Luo [23], Sturm [34] and Zhang [40] extended the homogeneous IPM to general conic programming, including semidefinite programming. Recently, Skajaa *et al.* [33] implemented a homogeneous IPM for solving nonsymmetric conic programming problems.

Andersen and Ye [11, 12] extended the homogeneous IPM to the general MCP. They developed a homogeneous IPM that converges to a maximal complementary solution in at most $\mathcal{O}(\sqrt{n}\log 1/\epsilon)$ iterations if f satisfies SLC. Although the global linear polynomial time convergence is provided in their analysis, exact search directions and restrictive step size computations are required, and hence their algorithm is not practical.

Motivated by the apparent gap between the theoretical complexity results and long-step practical implementations in IPM, Mehrotra and Huang [26] recently introduced a homogeneous IPM equipped with a continuously differentiable potential function for LP. Their method ensures a global linear polynomial time convergence while providing the flexibility to integrate heuristics for generating the search directions and the step size computations. More importantly, they showed that it is possible to implement a potential reduction IPM that uses similar number of iterations as an algorithm that ignores global linear polynomial time convergence.

In this paper the Mehrotra-Huang potential function is extended to the context of MCP. Our goal is to develop a stable interior point solver that ensures the global linear polynomial time convergence without compromising the computational efficiency. Two guiding principles are used in our implementation: (1) ensure a reduction in the potential function; (2) use the Mehrotra "corrector" direction to improve the reduction in the potential function. Our algorithm is implemented in a software package named `iOptimize`. `iOptimize` is developed based on a variant of Mehrotra's predictor-corrector framework [25]. Computational results on the standard test problems [2, 5, 6, 24] show that `iOptimize` appears to perform more robustly and solves problems with fewer average iterations than a mature commercial solver `MOSEK` (barrier solver version `6.0.0.106`; `MOSEK` for short) [7]. In particular, all problems in test sets [2, 5, 6, 24] were solved by `iOptimize` using default settings, while `MOSEK` failed on several problems. Moreover, the average number of iterations used by `iOptimize` are fewer than those taken by `MOSEK` for convex quadratic programming (QP) problems, convex quadratically constrained quadratic programming (QCQP) problems, and general convex programming (CP) problems in the test sets. We further created some infeasible problems to check the robustness of our approach. While `iOptimize` solved all these problems (except one, where the failure was due to an `AMPL` error message), `MOSEK` failed on several problems. We also found that `iOptimize` seems to detect infeasibility more reliably than general nonlinear solvers `Ipopt` (version `3.9.2`) [3] and `Knitro` (version `8.0`) [4].

The rest of this paper is organized as follows. In Section 2 we outline the known results for the homogeneous formulation for MCP. In Section 3 we describe our potential function and show that our potential reduction algorithm converges to a maximal complementary solution in polynomial time. In Section 4 we provide additional implementation details of `iOptimize`. Computational results are presented and discussed in Section 5. Concluding remarks are made in Section 6.

2. Homogeneous formulation for the monotone complementarity problem. In this section we summarize the main results of Andersen and Ye's [11, 12, Theorem 1] homogeneous IPM for MCP. Andersen and Ye consider an augmented homogeneous model related to MCP (HMCP):

$$s = \tau f(x/\tau), \tag{2.1}$$

$$\kappa = -x^T f(x/\tau), \tag{2.2}$$

$$0 \le (s,\kappa) \perp (x,\tau) \ge 0. \tag{2.3}$$

The right-hand side in (2.1) is closely related to the recession function in convex analysis of Rockafellar [32, Section 8]. Given a maximal complementary solution $(x^*, \tau^*, s^*, \kappa^*)$ that satisfies (2.1) and (2.2), suppose $\tau^* > 0$ and hence $\kappa^* = 0$. Then from (2.1) and (2.2) we have

$$s^*/\tau^* = f(x^*/\tau^*) \quad \text{and} \quad \kappa^*/\tau^* = -(x^*/\tau^*)^T (s^*/\tau^*) = 0.$$

This implies that $(x^*/\tau^*, s^*/\tau^*)$ is a complementary solution for MCP. On the other hand, if $\kappa^* > 0$ and hence $\tau^* = 0$, then the complementarity of MCP at (x^*, s^*) is not zero. In this case MCP is said to be infeasible, and x^* is a direction of a separating hyperplane that separates 0 and the set $\{s - f(x) \in R^n \mid (x,s) > 0\}$.

Let

$$F(x,\tau) := \begin{pmatrix} \tau f(x/\tau) \\ -x^T f(x/\tau) \end{pmatrix} : R^{n+1}_{++} \to R^{n+1}.$$

Andersen and Ye [12] showed the following theorem.

THEOREM 2.1. *Consider the MCP and the associated HMCP.*

1. *If ∇f is positive semidefinite in R^n_+, then ∇F is also positive semidefinite in R^{n+1}_{++}.*
2. *F is a continuous homogeneous function in R^{n+1}_{++} with degree 1 and for any $(x,\tau) \in R^{n+1}_{++}$, $(x;\tau)^T F(x,\tau) = 0$, and $(x,\tau)^T \nabla F(x,\tau) = -F(x,\tau)^T$.*
3. *If f is a continuous monotone mapping from R^n_+ to R^n, then F is a continuous monotone mapping from R^{n+1}_{++} to R^{n+1}.*
4. *If f is scaled Lipschitz with ν_f, then F is also scaled Lipschitz; that is, it satisfies condition (1.6) with*
$$\nu_F(\beta) = \left(1 + \frac{2\nu_f(2\beta/(1-\beta))}{1-\beta}\right)\left(\frac{1}{1-\beta}\right),$$
where $\beta < 1/3$.
5. *There exists a bounded sequence $\langle x^k, \tau^k, s^k, \kappa^k \rangle \subset R^{2n+2}_{++}$, $k = 0, 1, \ldots$ that solves HMCP in a limit.*
6. *Let $(x^*, \tau^*, s^*, \kappa^*)$ be a maximal complementary solution or ray for HMCP. MCP has a solution if and only if $\tau^* > 0$. In this case, $(x^*/\tau^*, s^*/\tau^*)$ is a complementary solution for MCP.*
7. *MCP is (strongly) infeasible if and only if $\kappa^* > 0$. In this case, $(x^*/\kappa^*, s^*/\kappa^*)$ is a certificate (ray) to prove (strong) infeasibility.*

We note that if f is an affine function, then there is a strictly complementary solution for HMCP. On the other hand, if f is a monotone function, then the maximal complementary solution $(x^*, \tau^*, s^*, \kappa^*)$ may not be a strictly complementary solution. However, in this case Andersen and Ye [12, proof for Theorem 1] showed that $\tau^* + \kappa^* > 0$. This implies (x^*, s^*) must either generate a finite solution for HMCP, or give a certificate that proves the infeasibility of MCP.

2.1. The path following homogeneous algorithm. For simplicity in the following we let $\bar{n} := n+1$, $x := (x;\tau) \in R^{\bar{n}}_+$ and $s := (s;\kappa) \in R^{\bar{n}}_+$. Let

$$r^k := s^k - F(x^k),$$

and let $(x^0, s^0) = e$ be the starting point. Let $\mathcal{C}(\vartheta)$ denote a continuous trajectory such that

$$\mathcal{C}(\vartheta) := \left\{(x^k, s^k) \mid s^k - F(x^k) = \vartheta r^0, X^k s^k = \vartheta e, 0 < \vartheta \leq 1\right\}.$$

Note that such a trajectory always exists (Güler [19]). Moreover, Andersen and Ye [12, Theorem 2] showed that this continuous trajectory is bounded and any limit point is a maximal complementary solution for HMCP. At iteration k with iterate $(x^k, s^k) > 0$, the algorithm computes the search direction (d_x, d_s) by solving the following system of linear equations:

$$d_s - \nabla F(x^k) d_x = -\eta r^k, \tag{2.4}$$

$$X^k d_s + S^k d_x = \gamma \mu^k e - X^k s^k. \tag{2.5}$$

Here η and γ are parameters between 0 and 1, and $\mu^k := \frac{(x^k)^T s^k}{\bar{n}}$. For a step size $\alpha > 0$, let the new iterate be

$$x^+ := x^k + \alpha d_x > 0 \quad \text{and}$$

$$s^+ := s^k + \alpha d_s + (F(x^+) - F(x^k) - \alpha \nabla F(x^k) d_x)) = F(x^+) + (1 - \alpha\eta) r^k > 0. \tag{2.6}$$

Andersen and Ye [12] showed the following lemmas.

LEMMA 2.2. *The direction (d_x, d_s) satisfies*

$$d_x^T d_s = d_x^T \nabla F(x^k) d_x + \eta(1 - \gamma - \eta)\bar{n}\mu^k.$$

LEMMA 2.3. *Let $r^+ = s^+ - F(x^+)$, and consider the new iterate (x^+, s^+) given by (2.4)–(2.6). Then,*

1. *$r^+ = (1 - \alpha\eta) r^k$.*
2. *$(x^+)^T s^+ = (x^k)^T s^k (1 - \alpha(1 - \gamma)) + \alpha^2 \eta(1 - \eta - \gamma)\bar{n}\mu^k$.*

Note that from Lemma 2.3, for $\eta = 1 - \gamma$ the infeasibility residual and the complementarity are reduced at the same rate. Based on these lemmas, Andersen and Ye [12] showed that with suitable chosen parameters α and β, their homogeneous algorithm will generate a sequence $\langle x^k, s^k \rangle > 0$ such that $\langle x^k, s^k \rangle \leq \|X^K s^k - \mu^k e\| \leq \vartheta \mu^k$, where $\vartheta \in (0, 1]$ is a constant parameter related to $\nu_F(\beta)$. Moreover, they showed that both $\|r^k\|$ and $(x^k)^T s^k$ converge to zero at a global rate $\gamma = 1 - \vartheta/\sqrt{\bar{n}}$. As a result, if f satisfies the SLC, then the homogeneous algorithm converges in $\mathcal{O}(\sqrt{n}\log(1/\epsilon))$ iterations. However, it requires exact search directions and uses short step computations, and thus is not a practical algorithm.

3. A practical potential reduction homogeneous algorithm. We consider the following potential function for the HMCP.

$$\Phi_\rho(x, s) := (\rho/2)\log\left\{(x^T s)^2 + \theta\|r\|^2\right\} - \sum_{j=1}^{\bar{n}} \log x_j s_j, \tag{3.1}$$

where θ is a constant parameter with positive value and $\rho \geq \bar{n} + \sqrt{\bar{n}}$. A potential function of this form was first introduced by Goldfarb and Mehrotra [17], and was extended to the homogeneous LP model by Mehrotra and Huang [26]. It is similar to

(1.7) except that the residual vector r is considered in the function. Without loss of generality we choose $\theta = 1$ in (3.1).

We need to analyze the worst case decrease in (3.1) at each iteration of our potential reduction IPM. Let

$$\begin{aligned} &S^k := diag(s^k_1, \ldots, s^k_{\bar{n}}), \quad D := (X^k S^k)^{0.5}, \quad D_{\min} := \min_{j=1,\ldots,\bar{n}} \left(x^k_j s^k_j\right)^{0.5}, \\ &p := \tfrac{\bar{n}}{\rho}\mu e - D^2 e = \tfrac{{x^k}^T s^k}{\rho} e - D^2 e, \quad \text{and} \\ &\alpha := \beta \tfrac{D_{\min}}{\|D^{-1}p\|}, \end{aligned} \tag{3.2}$$

where $\beta \in (0, 1)$ is a positive constant that will be determined later. Suppose (d_x, d_s) is computed by solving the linear system of equations (2.4) and (2.5) with $\eta = 1 - \gamma$ and $\gamma = \frac{\bar{n}}{\rho}$. Let

$$\begin{aligned} p_x &:= \alpha d_x, \\ p_s &:= \alpha d_s + (F(x^+) - F(x^k) - \alpha \nabla F(x^k) d_x), \\ z &:= X^k (F(x^+) - F(x^k) - \nabla F(x^k) p_x), \quad \text{and} \\ q &:= z/\alpha, \end{aligned} \tag{3.3}$$

and let $x^+ := x^k + p_x > 0$, and $s^+ := s^k + p_s > 0$. We have

$$p = \gamma \mu e - D^2 e = X^k d_s + S^k d_x, \tag{3.4}$$

and

$$X^k p_s + S^k p_x - \alpha p = z = \alpha q. \tag{3.5}$$

In Section 3.1 we show that if f satisfies SLC (and consequently so does F), then at iteration k the theoretical direction (d_x, d_s) with an appropriate step size α results in

$$\Phi_\rho(x^+, s^+) - \Phi_\rho(x^k, s^k) \le \zeta,$$

where ζ is a negative constant. Let $(\tilde{x}^+, \tilde{s}^+)$ be an iterate computed by using any favored search direction with inexact step sizes in the primal and dual spaces. Starting from an initial solution $(x^0, s^0) > 0$, our potential reduction IPM is given in Algorithm 1. This algorithm is designed to have the freedom in choosing any favored search direction with inexact and different step sizes in the primal and dual spaces.

In Section 3.2 we show that if $\Phi_\rho(x^k, s^k)$ is reduced by at least a constant amount at each iteration and $\Phi_{\bar{n}}(x^k, s^k)$ (potential function 3.1 with parameter $\rho = \bar{n}$) is upper bounded by a constant amount $\iota \bar{n} \log \bar{n}$ ($\iota > 0$ is a constant that will be given later), and iterates are maintained while satisfying Assumption 3.1 given below, then Algorithm 1 generates a maximal complementary solution of desired precision in polynomial time.

Assumption 3.1.
(i) $\Omega(1) \le \frac{(x^0)^T s^0 \left|r^k_j\right|}{(x^k)^T s^k \left|r^0_j\right|} \le \mathcal{O}(1), \; j = 1, \ldots, \bar{n}.$
(ii) $\Omega(1) \le \frac{(\tilde{x}^+)^T \tilde{s}^+}{(x^+)^T s^+} \le \mathcal{O}(1).$

Assumption 3.1 (i) requires that the overall decrease in the infeasibility to complementarity ratio is maintained in Algorithm 1. Assumption 3.1 (ii) requires that the complementarity at the iterate $(\tilde{x}^+, \tilde{s}^+)$ generated using a favored direction and

Algorithm 1 Practical potential reduction IPM

Input: An initial solution $(x^0, s^0) > 0$, constants $\zeta < 0$ and $\iota > 0$.
Output: An (asymptotically) maximal complementary solution (x^k, s^k).

Let $k = 0$.
while Termination criteria is not met **do**
  Compute $(\tilde{x}^+, \tilde{s}^+)$ using any favored search direction and step size for x^k and s^k.
  if $\Phi_\rho(\tilde{x}^+, \tilde{s}^+) - \Phi_\rho(x^k, s^k) \le \zeta$, $\Phi_{\bar{n}}(x^k, s^k) \le \iota \bar{n} \log \bar{n}$, and if Assumption 3.1 is satisfied **then**
    Set $(x^{k+1}, s^{k+1}) := (\tilde{x}^+, \tilde{s}^+)$.
  else
    Compute (x^+, s^+) using the theoretical direction (d_x, d_s) with an appropriate step size α.
    Set $(x^{k+1}, s^{k+1}) := (x^+, s^+)$.
  end if
  Set $k := k + 1$.
end while

the theoretical iterate (x^+, s^+) are of the same order. Note that if $\eta = 1 - \gamma$, then from Lemma 2.3 we have $(x^+)^T s^+ = (1 - \alpha\eta)(x^k)^T s^k$. As a result if only theoretical direction is chosen in the algorithm, Assumption 3.1 is satisfied with constant equal to 1, and the complementarity is decreased monotonically. Convergence properties for an algorithm that has additional flexibility of choosing more aggressive directions and step size are still ensured under Assumption 3.1.

3.1. Potential function reduction using the theoretical direction. The following analysis follows some of the steps in Kojima *et al.* [22] and Potra and Ye [31, Sections 2 and 3]. We drop the index k for an iteration to simplify the presentation.

PROPOSITION 3.2 (Kojuma *et al.* [22, Lemma 2.3]). *Let $u, v,$ and $w \in R^{\bar{n}}$ satisfy $u + v = w$ and $u^T v \ge 0$. Then*

$$\|u\| \le \|w\|, \|v\| \le \|w\|, \text{ and } u^T v \le \frac{1}{2}\|w\|^2.$$

LEMMA 3.3. *Let p_x be defined as in (3.3). Then, $\|X^{-1} p_x\|_\infty \le \beta$.*

Proof. We use the technique in Kojima *et al.* [22] for the proof. We first rewrite equation (3.4) as

$$\frac{D^{-1} X d_s}{\|D^{-1} p\|} + \frac{D^{-1} S d_x}{\|D^{-1} p\|} = \frac{D^{-1} p}{\|D^{-1} p\|}. \tag{3.6}$$

From Lemma 2.2, we have

$$\frac{(D^{-1} X d_s)^T D^{-1} S d_x}{\|D^{-1} p\| \|D^{-1} p\|} = \frac{d_x^T d_s}{\|D^{-1} p\|^2} \ge 0. \tag{3.7}$$

Now using Proposition 3.2 in (3.6) and (3.7), we have

$$\frac{\|D^{-1} S d_x\|}{\|D^{-1} p\|} \le \frac{\|D^{-1} p\|}{\|D^{-1} p\|} = 1. \tag{3.8}$$

Using (3.3) and (3.8), we deduce the following relation

$$\frac{\|X^{-1}d_x\|}{\|D^{-1}p\|} = \frac{\|D^{-2}Sd_x\|}{\|D^{-1}p\|} \le D_{\min}^{-1}. \tag{3.9}$$

Finally, using (3.2) and (3.9), we have

$$\|X^{-1}p_x\|_\infty = \alpha\|X^{-1}d_x\|_\infty \le \beta D_{\min}\frac{\|X^{-1}d_x\|}{\|D^{-1}p\|} \le \beta.$$

□

Potra and Ye [31] showed the following lemma.

LEMMA 3.4 (Potra and Ye [31, Lemma 3.2]). *If d_x and d_s are the solution of the linear system (3.5), and if condition (1.6) and Lemma (3.3) are satisfied, then*

$$\|z\| \le \xi\beta^2 D_{\min}^2,$$

where $\xi = 0.25\nu_F(\beta)$ and z is defined in (3.3).

From the above lemma and (3.2) we have

$$\|D^{-1}q\| \le \|q\|D_{\min}^{-1} = \frac{\|z\|}{\alpha}D_{\min}^{-1} \le \frac{\xi\beta^2}{\alpha}D_{\min} = \xi\beta\|D^{-1}p\|. \tag{3.10}$$

Kojima *et al.* showed the following lemma.

LEMMA 3.5 (Kojima *et al.* [22, Lemma 2.5]). *Suppose the search direction (d_x, d_s) is computed by solving the linear system of equations (2.4) and (2.5) with $\eta = 1-\gamma$ and $\gamma = \frac{\bar n}{\rho}$ ($\rho \ge \bar n + \sqrt{\bar n}$), then*

$$\frac{\rho}{x^Ts}\|D^{-1}p\| = \left\|D^{-1}e - \frac{\rho}{x^Ts}De\right\| \ge \frac{\sqrt{3}}{2}D_{\min}^{-1}.$$

In the following we define two real valued functions g_1 and g_2, and establish results that upper bound them. These two functions and the related upper bounds will be used later in our analysis. Let

$$g_1 := -\alpha\rho\eta + e^T\left(X^{-1}p_x + S^{-1}p_s\right) \quad \text{and} \tag{3.11}$$

$$g_2 := \frac{\|X^{-1}p_x\|^2 + \|S^{-1}p_s\|^2}{2(1-\beta)}. \tag{3.12}$$

LEMMA 3.6. *Let g_1 be defined as in (3.11). With a suitably chosen parameter $\beta \in (0,1)$, suppose the search direction (d_x, d_s) is computed by solving the linear system of equations (2.4) and (2.5) with $\eta = 1-\gamma$ and $\gamma = \frac{\bar n}{\rho}$ ($\rho \ge \bar n + \sqrt{\bar n}$). Let the step size α be defined as in (3.2). If condition (1.6) is satisfied, then*

$$g_1 \le -\frac{\sqrt{3}}{2}(1-\xi\beta)\beta + \xi\beta^2\frac{\rho}{\sqrt{\bar n}}.$$

Proof. From (2.5) and $\gamma = 1-\eta$, we have

$$-\alpha\rho\eta = \alpha\rho\frac{x^Td_s + s^Td_x}{\bar n\mu}. \tag{3.13}$$

An upper bound of g_1 is obtained in the following set of inequalities.

$$\begin{aligned}
g_1 &= \alpha\rho\left(\frac{x^T d_s + s^T d_x}{x^T s}\right) - e^T\left(X^{-1}p_x + S^{-1}p_s\right)\\
&= \alpha\frac{\rho}{x^T s}e^T(Xd_s + Sd_x) - e^T D^{-2}(Xp_s + Sp_x)\\
&= \alpha\frac{\rho}{x^T s}e^T\left(p - \frac{x^T s}{\rho}D^{-2}(p+q)\right) \quad \text{(using (3.4) and (3.5))}\\
&= \alpha\frac{\rho}{x^T s}e^T\left(\left(D - \frac{x^T s}{\rho}D^{-1}\right)D^{-1}p - \frac{x^T s}{\rho}D^{-2}q\right)\\
&= -\alpha\frac{\rho}{x^T s}\left(p^T D^{-1}D^{-1}p\right) - \alpha\frac{\rho}{x^T s}\left(D^{-1}p + De\right)^T D^{-1}q \quad \text{(using (3.2))}\\
&= -\alpha\frac{\rho}{x^T s}\left(p^T D^{-1}D^{-1}(p+q)\right) - \alpha\frac{\rho}{x^T s}e^T q\\
&= -\alpha\frac{\rho}{x^T s}\left(\|D^{-1}p\|^2 + p^T D^{-1}D^{-1}q\right) - \frac{\rho}{x^T s}e^T z \quad \text{(using (3.3))}\\
&\le -\alpha\frac{\rho}{x^T s}\left(\|D^{-1}p\|^2 - \|D^{-1}p\|\|D^{-1}q\|\right) + \frac{\rho}{x^T s}\|z\|_1\\
&\le -\alpha\frac{\rho}{x^T s}\left(\|D^{-1}p\|^2 - \xi\beta\|D^{-1}p\|^2\right) + \frac{\rho}{x^T s}\sqrt{n}\|z\| \quad \text{(using (3.10))}\\
&\le -\alpha\frac{\rho}{x^T s}(1-\xi\beta)\|D^{-1}p\|^2 + \frac{\rho}{x^T s}\sqrt{n}\xi\beta^2 D_{\min}^2 \quad \text{(using Lemma 3.4)}\\
&\le -\frac{\sqrt{3}}{2}(1-\xi\beta)\alpha D_{\min}^{-1}\|D^{-1}p\| + \frac{\rho}{\sqrt{n}}\xi\beta^2 \quad \text{(using Lemma 3.5)}\\
&= -\frac{\sqrt{3}}{2}(1-\xi\beta)\beta + \frac{\rho}{\sqrt{n}}\xi\beta^2. \quad \text{(using (3.2))}
\end{aligned}$$

□

LEMMA 3.7. *Let g_2 be defined as in (3.12). Under the hypothesis of Lemma 3.7, we have*

$$g_2 \le \frac{\beta^2(1+\xi\beta)^2}{2(1-\beta)}.$$

Proof. The following proof is based on techniques in Potra and Ye [31]. We have

$$\begin{aligned}
2(1-\beta)g_2 &= \|X^{-1}p_x\|^2 + \|S^{-1}p_s\|^2\\
&= \|D^{-2}Xp_s\|^2 + \|D^{-2}Sp_x\|^2\\
&\le D_{\min}^{-2}\left(\|D^{-1}Xp_s\|^2 + \|D^{-1}Sp_x\|^2\right)\\
&= D_{\min}^{-2}\left(\|D^{-1}(Xp_s + Sp_x)\|^2 - 2p_x^T p_s\right)\\
&= D_{\min}^{-2}\left(\alpha^2\|D^{-1}(p+q)\|^2 - 2p_x^T p_s\right) \quad \text{(using (3.5))}\\
&\le D_{\min}^{-2}\left(\alpha^2(\|D^{-1}p\| + \|D^{-1}q\|)^2 - 2p_x^T p_s\right)\\
&\le D_{\min}^{-2}\left(\alpha^2\ (1+\xi\beta)^2|D^{-1}p\|^2 - 2p_x^T p_s\right) \quad \text{(using (3.10))}\\
&= \left(\alpha D_{\min}^{-1}\|D^{-1}p\|\right)^2(1+\xi\beta)^2 - 2D_{\min}^{-2}p_x^T p_s\\
&= \beta^2(1+\xi\beta)^2 - 2D_{\min}^{-2}p_x^T p_s. \quad \text{(using (3.2))}
\end{aligned} \tag{3.14}$$

The upper bound on $p_x^T p_s$ is obtained as follows.

$$p_x^T p_s = p_x^T(\alpha d_s + (F(x^+) - F(x) - \nabla F(x)p_x)) \quad \text{(using (3.3))} \tag{3.15}$$

$$
\begin{aligned}
&= (x^+ - x)^T (F(x^+) - F(x)) + \alpha^2 (d_x^T d_s - d_x^T \nabla F(x) d_x) \quad \text{(using (3.3))}\\
&= (x^+ - x)^T (F(x^+) - F(x)) + \alpha^2 \eta (1 - \eta - \gamma) \bar{n}\mu \quad \text{(using Lemma 2.2)}\\
&\geq 0 \quad \text{(from the monotonicity of } F\text{).}
\end{aligned}
$$

The result follows from using (3.14) and (3.15). □

The following proposition is frequently used in the IPM analysis.

PROPOSITION 3.8 (Karmarker [21]).

1. *If* $1-\delta > 0$, *then* $\log\{1-\delta\} \leq -\delta$.
2. *If* $\delta \in R^{\bar{n}}$ *satisfies* $\|\delta\|_\infty \leq \beta < 1$, *then* $\sum_{j=1}^{\bar{n}} \log\{1-\delta_j\} \geq -e^T\delta - \frac{\|\delta\|^2}{2(1-\beta)}$.

Now we are ready to show a bound on the difference between the potential function values $\Phi_\rho(x^+, s^+)$ and $\Phi_\rho(x,s)$. The next theorem bounds the improvement in the potential function from taking the theoretical direction with a controllable step size.

THEOREM 3.9. *Let* $\xi = 0.25\nu_F(\beta) \geq 0.25$. *Suppose the search direction* (d_x, d_s) *is computed by solving the linear system of equations (2.4) and (2.5) with* $\eta = 1-\gamma$ *and* $\gamma = \frac{\bar{n}}{\rho}$ $(\rho \geq \bar{n} + \sqrt{\bar{n}})$. *Let the step size* α *be defined as in (3.2). If condition (1.6) is satisfied, and* $\beta \in (0, 0.303)$ *is chosen such that*

$$
\xi\beta \leq \frac{\sqrt{3}}{4(\sqrt{3} + 2\rho/\sqrt{\bar{n}})}, \quad \textit{and} \tag{3.16}
$$

$$
\beta\sqrt{\xi}(1+\xi\beta) \leq \frac{\sqrt{3}}{2\left(\sqrt{3} + 2\rho/\sqrt{\bar{n}}\right)^{1/2}}. \tag{3.17}
$$

Then, (x^+, s^+) *is always strictly positive. Moreover,*

$$
\Phi_\rho(x^+, s^+) - \Phi_\rho(x,s) \leq -0.278\frac{\beta}{1-\beta}.
$$

Proof. We first show that (x^+, s^+) is strictly positive. From Lemma 3.7 and (3.17), we have

$$
\max\left\{\|X^{-1}p_x\|_\infty, \|S^{-1}p_s\|_\infty\right\} \leq \beta(1+\xi\beta) \leq \frac{\sqrt{3}}{2\left(\xi\left(\sqrt{3} + 2\rho/\sqrt{\bar{n}}\right)\right)^{1/2}} < 1.
$$

Hence, (p_x, p_s) is a valid direction. We note that from Lemma 3.3, we have

$$
\|X^{-1}p_x\|_\infty \leq \beta \leq \frac{\sqrt{3}}{4\xi(\sqrt{3} + 2\rho/\sqrt{\bar{n}})} \leq \frac{\sqrt{3}}{2\left(\xi\left(\sqrt{3} + 2\rho/\sqrt{\bar{n}}\right)\right)^{1/2}}.
$$

Therefore, the upper bound from Lemma 3.3 is still valid (and tighter). Moreover, $\beta < 0.303$ implies that the condition in Theorem 2.1 [4] is satisfied. Now,

$$
\begin{aligned}
&\Phi_\rho(x^+, s^+) - \Phi_\rho(x,s)\\
= \;& (\rho/2)\log\left\{(1-\alpha\eta)^2\left((\bar{n}\mu)^2 + \|r\|^2\right)\right\} - \sum_{j=1}^{\bar{n}} \log\left\{(x_j + (p_x)_j)(s_j + (p_s)_j)\right\}\\
&- (\rho/2)\log\left\{(\bar{n}\mu)^2 + \|r\|^2\right\} + \sum_{j=1}^{\bar{n}} \log\{x_j s_j\} \text{ (using Lemma 2.3)}
\end{aligned}
$$

$$
\begin{aligned}
&= \rho \log \{1 - \alpha\eta\} - \sum_{j=1}^{\bar{n}} \log \left\{1 + \frac{(p_x)_j}{x_j}\right\}\left\{1 + \frac{(p_s)_j}{s_j}\right\} \\
&\leq -\alpha\rho\eta - e^T\left(X^{-1}p_x + S^{-1}p_s\right) - \frac{\|X^{-1}p_x\|^2 + \|S^{-1}p_s\|^2}{2(1-\beta)} \text{ (using Proposition 3.8)} \\
&= g_1 + g_2.
\end{aligned}
$$

From Lemmas 3.6 and 3.7, we write

$$
\begin{aligned}
\Phi_\rho(x^+, s^+) - \Phi_\rho(x, s) &\leq -\frac{\sqrt{3}}{2}(1 - \xi\beta)\beta + \xi\beta^2\rho/\sqrt{\bar{n}} + \frac{\beta^2(1+\xi\beta)^2}{2(1-\beta)} \\
&= \frac{\beta^2(1+\xi\beta)^2}{2(1-\beta)} + \beta\left(-\frac{\sqrt{3}}{2} + \xi\beta\left(\frac{\sqrt{3}}{2} + \frac{\rho}{\sqrt{\bar{n}}}\right)\right) \\
&= \frac{\beta^2(1+\xi\beta)^2}{2(1-\beta)} - \frac{\sqrt{3}}{2}\beta\left(1 - \xi\beta\left(\frac{\sqrt{3} + 2(\rho/\sqrt{\bar{n}})}{\sqrt{3}}\right)\right) =: \zeta.
\end{aligned}
$$

It follows that

$$
-2\frac{1-\beta}{\beta}\zeta = -\beta(1+\xi\beta)^2 + \sqrt{3}(1-\beta)\left(1 - \xi\beta\left(\frac{\sqrt{3} + 2(\rho/\sqrt{\bar{n}})}{\sqrt{3}}\right)\right).
$$

Numerically, it is easy to verify that

$$
\zeta \leq -0.278\frac{\beta}{1-\beta}.
$$

□

Since $\nu_F(\beta) : (0, 1) \mapsto (1, \infty)$ is a monotone increasing function depending on $\beta \in (0, 0.303)$, we can always choose a sufficiently small β that satisfies (3.16) and (3.17). As a result, a reduction in the potential function ($\zeta < -0.278\frac{\beta}{1-\beta}$) is guaranteed. However, choosing β is dependent on $\nu_F(\beta)$, finding a maximal β that satisfies (3.16) and (3.17) requires the knowledge of the scaled Lipschitz constant $\nu_F(\beta)$, which may not be available in practice for an arbitrary convex function. Problems satisfying the scaled Lipschitz condition are discussed in [27, 30, 41]. Nevertheless, the analysis suggests that for an appropriate step size, the potential function value will reduce as long as the scaled Lipschitz constant is bounded. Consequently, the potential function can be used as a guide to solve monotone complementarity and convex optimization problems satisfying this property. If the condition is not satisfied, then lack of sufficient reduction in the potential function indicates that the algorithm proposed here is not appropriate for the problem being solved. Finally, we note that Theorem 3.9 is similar to the result by Potra and Ye [31], but with a difference that it allows us to choose a larger ρ, when β is sufficiently small.

Remark. Note that by performing simple block elimination on the system of linear equations (2.4) and (2.5), we obtain the following alternative formulation:

$$
X(\alpha\nabla F(x)d_x - \alpha\eta r) + \alpha S d_x - \alpha p = 0. \tag{3.18}
$$

Hence, d_x is a direction obtained by applying a single Newton step to the equation

$$
z(h) := X(F(x + \alpha h) - F(x) - \alpha\eta r) + \alpha S h - \alpha p = 0, \tag{3.19}
$$

with starting point $h = 0$. If F is linear (for example, LP formulated as a HMCP model), then the linear system (3.19) can be solved exactly. In this case Theorem 3.9 with $\xi = 0$ reduces to the result of Mehrotra and Huang [26].

Suppose F is nonlinear and does not satisfy SLC. Given positive constants ξ and β that satisfy conditions (3.16) and (3.17). If one can compute an approximate solution δ_x such that

$$\|z(\delta_x)\| = \|X(F(x+\alpha\delta_x) - F(x) - \alpha\eta r) + \alpha S\delta_x - \alpha p\| \le \xi\beta^2 \min_{j=1,\ldots,\bar{n}}(x_j s_j), \tag{3.20}$$

then the result of Theorem 3.9 is still satisfied. However, finding a direction δ_x may require many Newton steps, and an explicit bound on the number of Newton steps is unknown.

3.2. Convergence of the practical potential reduction homogeneous algorithm. The following theorem shows that if potential function (3.1) can be reduced by at least a constant amount at each iteration, then the Algorithm 1 generates an ϵ−complementary solution in polynomial time. Our analysis follows some of the steps in Kojima *et al.* [22] and Porta and Ye [29].

THEOREM 3.10. *Let $\epsilon > 0$. Starting from a solution $(x^0, s^0) > 0$, if the value of potential function (3.1) can be reduced by at least a constant amount and if Assumption 3.1 is satisfied at each iteration of Algorithm 1, then after at most $\mathcal{O}(\Phi_\rho(x^0, s^0) - \bar{n}\log\bar{n} + (\rho - \bar{n})|\log\epsilon|)$ iterations, we will have a solution such that*

$$x^T s \le \epsilon \quad \textit{and} \quad \frac{|r_j|}{|r_j^0|} \le \mathcal{O}(1)\frac{\epsilon}{(x^0)^T s^0},\ j = 1,\ldots,\bar{n}. \tag{3.21}$$

Proof. Using Assumption 3.1, we have

$$(\rho/2)\log\left\{(x^T s)^2\left(1 + \left(\Omega(1)\frac{\|r^0\|}{(x^0)^T s^0}\right)^2\right)\right\} - \sum_{j=1}^{\bar{n}} \log x_j s_j \le \Phi_\rho(x, s),$$

which implies

$$\begin{aligned}(\rho - \bar{n})\log x^T s - \sum_{j=1}^{\bar{n}} \log\left\{\frac{\bar{n}x_j s_j}{x^T s}\right\} + (\rho/2)\log\left\{1 + \left(\Omega(1)\frac{\|r^0\|}{(x^0)^T s^0}\right)^2\right\} \\ \le \Phi_\rho(x, s) - \bar{n}\log\bar{n}.\end{aligned} \tag{3.22}$$

Since the second and the third terms of (3.22) are always nonnegative, it follows that

$$(\rho - \bar{n})\log x^T s \le \Phi_\rho(x, s) - \bar{n}\log\bar{n}.$$

Therefore, $\Phi_\rho(x, s) \le \bar{n}\log\bar{n} - (\rho - \bar{n})|\log\epsilon|$, then we have

$$x^T s \le \epsilon.$$

Moreover, Assumption 3.1 implies that

$$\frac{|r_j|}{|r_j^0|} \le \mathcal{O}(1)\frac{x^T s}{(x^0)^T s^0} \le \mathcal{O}(1)\frac{\epsilon}{(x^0)^T s^0},\ j = 1,\ldots,\bar{n}.$$

□

We now show that any limit point of a sequence $\langle x^k, s^k\rangle$ generated by the potential reduction algorithm of Section 3.2 is a maximal complementary solution. Our analysis here follows some of the steps in Güler and Ye [20, Sections 2 and 4] and Potra and Ye [31, Section 4].

LEMMA 3.11. *Given an initial positive point (x^0, s^0). Recall that $(\tilde{x}^+, \tilde{s}^+)$ is a tentative point computed using any favored search direction using an inexact and different step size in the primal and dual spaces, and it satisfies Assumption 3.1; whereas (x^+, s^+) is computed using the theoretical direction (d_x, d_s) and a fixed step size α. Let*

$$C_1 := \frac{\rho-\bar{n}}{2}\log\left\{\frac{\left((x^0)^T s^0\right)^2 + \mathcal{O}(1)^2\left\|r^0\right\|^2}{\left((x^0)^T s^0\right)^2 + \Omega(1)^2\left\|r^0\right\|^2}\right\} \geq 0, \quad \textit{and}$$
$$C_2 := C_1 + \max\left\{0, -(\rho-\bar{n})\log\Omega(1)\right\} \geq 0,$$

then,

$$\Phi_{\bar{n}}(x^+, s^+) - \Phi_{\bar{n}}(x^k, s^k) \leq \zeta - (\rho-\bar{n})\log\left\{1 - D_{\min}^2\frac{\sqrt{\bar{n}}}{(x^k)^T s^k}\right\}, \quad \textit{and} \tag{3.23}$$

$$\Phi_{\bar{n}}(\tilde{x}^+, \tilde{s}^+) - \Phi_{\bar{n}}(x^k, s^k) \leq \zeta - (\rho-\bar{n})\log\left\{1 - D_{\min}^2\frac{\sqrt{\bar{n}}}{(x^k)^T s^k}\right\} + C_2, \tag{3.24}$$

where $\Phi_{\bar{n}}(x^k, s^k)$ is defined as in (3.1) with $\rho = \bar{n}$.

Proof. To simplify the presentation we drop the index k. Using Assumption 3.1, we have

$$\left\|\tilde{r}^+\right\|^2 \geq \left(\Omega(1)\left\|r^0\right\|\frac{(\tilde{x}^+)^T\tilde{s}^+}{(x^0)^T s^0}\right)^2, \quad \|r\|^2 \leq \left(\mathcal{O}(1)\left\|r^0\right\|\frac{x^T s}{(x^0)^T s^0}\right)^2, \quad \text{and} \tag{3.25}$$

$$(\tilde{x}^+)^T\tilde{s}^+ \geq \Omega(1)(x^+)^T s^+. \tag{3.26}$$

First we bound the difference between the values of $\Phi_{\bar{n}}(\tilde{x}^+, \tilde{s}^+)$ and $\Phi_{\bar{n}}(x, s)$. Let $\tilde{r}^+ = \tilde{s}^+ - F(\tilde{x}^+)$.

$$\begin{aligned}
&\Phi_{\bar{n}}(\tilde{x}^+, \tilde{s}^+) - \Phi_{\bar{n}}(x, s)\\
&= \Phi_\rho(\tilde{x}^+, \tilde{s}^+) - \Phi_\rho(x, s) - \Phi_\rho(\tilde{x}^+, \tilde{s}^+) + \Phi_\rho(x, s) + \Phi_{\bar{n}}(\tilde{x}^+, \tilde{s}^+) - \Phi_{\bar{n}}(x, s)\\
&\leq \zeta - \frac{\rho-\bar{n}}{2}\log\left\{((\tilde{x}^+)^T\tilde{s}^+)^2 + \left\|\tilde{r}^+\right\|^2\right\}\\
&\qquad + \frac{\rho-\bar{n}}{2}\log\left\{(x^T s)^2 + \|r\|^2\right\} \text{ (using Theorem 3.9)}\\
&\leq \zeta - \frac{\rho-\bar{n}}{2}\log\left\{((\tilde{x}^+)^T\tilde{s}^+)^2 + \left(\Omega(1)\left\|r^0\right\|\frac{(\tilde{x}^+)^T\tilde{s}^+}{(x^0)^T s^0}\right)^2\right\}\\
&\qquad + \frac{\rho-\bar{n}}{2}\log\left\{(x^T s)^2 + \left(\mathcal{O}(1)\left\|r^0\right\|\frac{x^T s}{(x^0)^T s^0}\right)^2\right\} \text{ (using (3.25))}\\
&= \zeta - (\rho-\bar{n})\log\left\{\frac{(\tilde{x}^+)^T\tilde{s}^+}{x^T s}\right\} + C_1\\
&\leq \zeta - (\rho-\bar{n})\log\left\{\frac{(x^+)^T s^+}{x^T s}\right\} - (\rho-\bar{n})\log\Omega(1) + C_1 \text{ (using (3.26))}\\
&\leq \zeta - (\rho-\bar{n})\log\left\{\frac{(x^+)^T s^+}{x^T s}\right\} + C_2
\end{aligned}$$

$$
\begin{aligned}
&\le \zeta - (\rho - \bar{n}) \log \{1 - \alpha\eta\} + C_2 \text{ (using Lemma 2.3)}\\
&= \zeta - (\rho - \bar{n}) \log \left\{ 1 + \alpha \frac{x^T d_s + s^T d_s}{x^T s} \right\} + C_2 \text{ (using (3.13))}\\
&= \zeta - (\rho - \bar{n}) \log \left\{ 1 + \alpha \frac{p^T e}{x^T s} \right\} + C_2 \text{ (using (3.4))}.
\end{aligned}
$$

Now,

$$
\begin{aligned}
\alpha \frac{p^T e}{x^T s} &= \frac{\alpha \left(\left(x^T s / \rho \right) e^T e - e^T D^2 e \right)}{x^T s} \text{ (using (3.2))}\\
&= \alpha \left(\frac{\bar{n}}{\rho} - 1 \right)\\
&= -\beta \frac{D_{\min}}{\|D^{-1} p\|} \left(\frac{\rho - \bar{n}}{\rho} \right) \text{ (using (3.2))}\\
&\ge -\beta D_{\min}^2 \frac{2(\rho - \bar{n})}{\sqrt{3} x^T s} \text{ (using Lemma 3.5)}\\
&\ge -D_{\min}^2 \frac{\sqrt{\bar{n}}(\rho - \bar{n})}{2 x^T s \xi (\sqrt{3\bar{n}} + 2\rho)} \text{ (using (3.16))}\\
&\ge -D_{\min}^2 \frac{2\rho\sqrt{\bar{n}}}{x^T s (\sqrt{3\bar{n}} + 2\rho)} \text{ (because } \xi = 0.25\nu_F \beta \ge 0.25)\\
&\ge -D_{\min}^2 \frac{\sqrt{\bar{n}}}{x^T s}.
\end{aligned}
$$

Therefore,

$$
\Phi_{\bar{n}}(\tilde{x}^+, \tilde{s}^+) - \Phi_{\bar{n}}(x, s) \le \zeta - (\rho - \bar{n}) \log \left\{ 1 - D_{\min}^2 \frac{\sqrt{\bar{n}}}{x^T s} \right\} + C_2.
$$

The difference between the values of $\Phi_{\bar{n}}(x^+, s^+)$ and $\Phi_{\bar{n}}(x, s)$ can be constructed in a similar way. □

LEMMA 3.12. *Given an initial positive point (x^0, s^0). Suppose a sequence $\langle x^k, s^k \rangle$ is generated by using Algorithm 1. There is a constant M_1 independent of k such that*

$$
\Phi_{\bar{n}}(x^k, s^k) \le M_1 \quad \textit{for all } k.
$$

Proof. Let $C_3 := (\bar{n}/2) \log \left\{ 1 + \left(\mathcal{O}(1) \frac{\|r^0\|}{(x^0)^T s^0} \right)^2 \right\} > 0$. Using (3.25), we have

$$
\begin{aligned}
\Phi_{\bar{n}}(x, s) &= (\bar{n}/2) \log \left\{ (x^T s)^2 + \|r\|^2 \right\} - \sum_{j=1}^{\bar{n}} \log x_j s_j\\
&\le (\bar{n}/2) \log \left\{ (x^T s)^2 + \left(\mathcal{O}(1) \left\| r^0 \right\| \frac{x^T s}{(x^0)^T s^0} \right)^2 \right\} - \sum_{j=1}^{\bar{n}} \log x_j s_j \text{ (using (3.25))}\\
&= \bar{n} \log x^T s - \sum_{j=1}^{\bar{n}} \log x_j s_j + C_3. \qquad (3.27)
\end{aligned}
$$

In the following we analyze the behavior of $\bar{n}\log x^T s - \sum_{j=1}^{\bar{n}} \log x_j s_j$. Let $\iota > 0$ be sufficiently large so that

$$\zeta - (\rho - \bar{n}) \log \left\{ 1 - \frac{1}{\bar{n}^{\iota - 0.5}} \right\} < 0. \tag{3.28}$$

Now consider the case where $\Phi_{\bar{n}}(x,s) \geq \bar{n}\log x^T s - \sum_{j=1}^{\bar{n}} \log x_j s_j \geq \iota \bar{n} \log \bar{n}$. In this case we have,

$$\begin{aligned}
&\bar{n}\log x^T s + \bar{n}\log D_{\min}^{-2} \geq \iota\bar{n}\log\bar{n}, \\
\Rightarrow\ &\log\left\{x^T s D_{\min}^{-2}\right\} \geq \iota\log\bar{n}, \\
\Rightarrow\ &\Phi_{\bar{n}}(x^+, s^+) - \Phi_{\bar{n}}(x,s) = \zeta - (\rho - \bar{n})\log\left\{1 - D_{\min}^2 \frac{\sqrt{\bar{n}}}{x^T s}\right\} \quad \text{(using (3.23))} \\
&\qquad\qquad \leq \zeta - (\rho - \bar{n})\log\left\{1 - \frac{1}{\bar{n}^{\iota - 0.5}}\right\} < 0.
\end{aligned}$$

The above relations suggests that using the theoretical direction and a fixed step size can decrease the value of $\Phi_{\bar{n}}(x,s)$ if this value is larger than $\iota\bar{n}\log\bar{n}$. Next we consider the case where $\Phi_{\bar{n}}(x,s) \leq \iota\bar{n}\log\bar{n}$. We show that the value of $\Phi_{\bar{n}}(x,s)$ in this case is upper bounded by a constant amount independent of k, and the iterate is computed using any favored search direction with inexact and different step sizes in the primal and dual spaces.
Using (3.24) and (3.27), we have

$$\begin{aligned}
\Phi_{\bar{n}}(\tilde{x}^+, \tilde{s}^+) &\leq \Phi_{\bar{n}}(x,s) \leq \iota\bar{n}\log\bar{n} + C_3 \\
&\leq \zeta - (\rho - \bar{n})\log\left\{1 - D_{\min}^2 \frac{\sqrt{\bar{n}}}{x^T s}\right\} + \iota\bar{n}\log\bar{n} + C_2 + C_3 \\
&\leq \zeta - (\rho - \bar{n})\log\left\{1 - \frac{1}{\sqrt{\bar{n}}}\right\} + \iota\bar{n}\log\bar{n} + C_2 + C_3.
\end{aligned}$$

Consequently, it follows that Lemma 3.12 holds with

$$M_1 := \max\left\{\zeta - (\rho - \bar{n})\log\left\{1 - \frac{1}{\sqrt{\bar{n}}}\right\} + \iota\bar{n}\log\bar{n} + C_2 + C_3, \Phi_{\bar{n}}(x^0, s^0)\right\}.$$

□

Lemma 3.13. *Let M_1 be a fixed constant, and*

$$\Phi_{\bar{n}}(x^k, s^k) \leq M_1 \quad \textit{for all } k.$$

Under the hypothesis of Lemma 3.12. Then, there exists a fixed M_2 (independent of k) such that

$$\frac{\min\left\{x_j^k s_j^k\right\}}{(x^k)^T s^k} \geq M_2.$$

Proof. As usual, the index k is dropped for simplification. Using the result in Lemma 3.12, we have

$$M_1 \geq \Phi_{\bar{n}}(x,s) = (\bar{n}/2)\log\left\{(x^T s)^2 + \|r\|^2\right\} - \sum_{j=1}^{\bar{n}} \log x_j s_j \geq \bar{n}\log x^T s - \sum_{j=1}^{\bar{n}} \log x_j s_j.$$

Hence, it follows that

$$\begin{aligned}
\log x^T s - \log x_i s_i &\le M_1 - (\bar{n}-1)\log x^T s + \sum_{j\neq i} \log x_j s_j \\
&\le M_1 - (\bar{n}-1)\log\left\{x^T s - x_i s_i\right\} + \sum_{j\neq i} \log x_j s_j \\
&\le M_1 - (\bar{n}-1)\log\left\{\bar{n}-1\right\} =: -\log M_2.
\end{aligned}$$

□

Now let

$$\begin{aligned}
\sigma(x) &:= \left\{j \in \{1,2,\ldots,\bar{n}\} : x_j \ge s_j\right\} \quad \text{and} \\
\sigma(s) &:= \left\{j \in \{1,2,\ldots,\bar{n}\} : s_j \ge x_j\right\}.
\end{aligned}$$

Combining Lemmas 3.12 and 3.13, the following theorem shows that the sequence $\langle x^k, s^k\rangle$ generates a maximal complementary solution for HMCP in the limit.

THEOREM 3.14. *Under the hypothesis of Lemma 3.12, the limit point of the sequence $\langle x^k, s^k\rangle$ is a maximal complementary solution for HMCP.*

Proof. Let (x^*, s^*) be any maximal complementary solution for HMCP such that

$$s^* = F(x^*) \quad \text{and} \quad (x^*)^T s^* = 0.$$

Let (x^∞, s^∞) be a limit point of the sequence $\langle x^k, s^k\rangle$. In the following we show that (x^∞, s^∞) is a maximal complementary solution, that is, $\sigma(x^\infty) = \sigma(x^*)$ and $\sigma(s^\infty) = \sigma(s^*)$. Let $|r|^T := (|r_1|, \ldots, |r_n|)$ and $C_4 := \mathcal{O}(1)\frac{|r^0|^T x^*}{(x^0)^T s^0} \ge 0$. Using Assumption 3.1, we have

$$\begin{aligned}
r^T x^* &\le |r|^T x^* \\
&\le O(1)(x^T s)\frac{|r^0|^T x^*}{(x^0)^T s^0} \\
&= C_4 x^T s. \qquad (3.29)
\end{aligned}$$

From the monotonicity of $F(x)$, for any positive solution $(x, s) > 0$, we have

$$\begin{aligned}
&(x - x^*)^T (F(x) - F(x^*)) \ge 0, \\
\Rightarrow\ & -x^T s^* - (s - r)^T x^* \ge 0, \\
\Rightarrow\ & -x^T s^* + s^T x^* \le r^T x^* \le C_4 x^T s, \ \text{(using (3.29))} \\
\Rightarrow\ & \sum_{j\in\sigma(x^*)} x_j^* s_j + \sum_{j\in\sigma(s^*)} s_j^* x_j \le C_4 x^T s.
\end{aligned}$$

As a result, if $j \in \sigma(x^*)$, then

$$C_4 x^T s \ge s_j x_j^* \ge (s_j x_j)\frac{x_j^*}{x_j} \ge \min\left\{x_j s_j\right\}\frac{x_j^*}{x_j}.$$

Using Lemma 3.13, the above relation implies

$$x_j \ge \frac{\min\left\{x_j s_j\right\}}{C_4 x^T s} x_j^* \ge \frac{M_2}{C_4} x_j^*.$$

Hence, we have $x_j^\infty \ge (M_2/C_4)x_j^* > 0$, consequently $\sigma(x^*) = \sigma(x^\infty)$. The proof of $\sigma(s^*) = \sigma(s^\infty)$ can be constructed in a similar way. □

4. Implementation details. In this section we present the implementation details of a potential reduction homogeneous algorithm for the MCP using the potential function given in Section 3. We describe the details in the context of solving a general convex optimization problem (CP).

4.1. Homogeneous formulation for convex programming. Consider a primal CP

$$\begin{aligned} \min\ & c(x) \\ \text{s.t. } & a_i(x) \geq 0, i = 1, \ldots, \hat{m}, \\ & a_i(x) = 0, i = \hat{m}+1, \ldots, m, \\ & \hat{x} \geq 0, \text{ and } \tilde{x} \text{ is free,} \end{aligned} \tag{4.1}$$

where $x \in R^n$ represents the primal variables. We let $\hat{n} + \tilde{n} = n$ and let $x = (\hat{x}; \tilde{x}) \in R^{\hat{n}} \times R^{\tilde{n}}$, where $\hat{x}$ and $\tilde{x}$ respectively represent the "normal" and "free" primal variables. Here, the function $c(\cdot) : R^n \to R$ is convex, the component functions $a_i(\cdot) : R^n \to R, i = 1, \ldots, \hat{m}$ are concave, and the component functions $a_i(\cdot) : R^n \to R, i = \hat{m}+1, \ldots, m$ are affine.

We let $m = \hat{m} + \tilde{m}$ and $y = (\hat{y}; \tilde{y}) \in R^{\hat{m}} \times R^{\tilde{m}}$, where $\hat{y}$ and $\tilde{y}$ respectively represent the "normal" and "free" dual variables. Let $s \in R^{\hat{n}}$ represent the vector of dual slacks. Let $\omega := (x; y; \tau; z; s; \kappa)$. In the following we use the notation

$$\begin{aligned} \mathring{\Omega}_x &:= \left\{ x = (\hat{x}; \tilde{x}) : \hat{x} \in R^{\hat{n}}_{++}, \tilde{x} \in R^{\tilde{n}} \right\}, \\ \mathring{\Omega}_y &:= \left\{ y = (\hat{y}; \tilde{y}) : \hat{y} \in R^{\hat{m}}_{++}, \tilde{y} \in R^{\tilde{m}} \right\}, \\ \mathring{\Omega} &:= \left\{ \omega : \omega \in \mathring{\Omega}_x \times \mathring{\Omega}_y \times R^{\hat{m}+\hat{n}+2}_{++} \right\}. \end{aligned}$$

Moreover, we assume that functions $c(\cdot)$ and $a_i(\cdot), i = 1, \ldots, \hat{m}$, are at least twice differentiable on the set $\mathring{\Omega}_x$, and the matrix

$$\begin{bmatrix} \begin{bmatrix} D_z & 0 \\ 0 & 0 \end{bmatrix} & \nabla a(x) \\ \nabla a(x)^T & -\nabla_x^2 \mathcal{L}(x, y) - \begin{bmatrix} D_x & 0 \\ 0 & 0 \end{bmatrix} \end{bmatrix}$$

is nonsingular for any positive semidefinite matrices $D_x \in R^{\hat{n} \times \hat{n}}$ and $D_z \in R^{\hat{m} \times \hat{m}}$.

Here

$$\mathcal{L}(x, y) := c(x) - y^T a(x)$$

denotes the Lagrangian function, and we use the notation

$$\nabla_x \mathcal{L}(x, y) := \left(\frac{\partial \mathcal{L}(x, y)}{\partial x_1}, \ldots, \frac{\partial \mathcal{L}(x, y)}{\partial x_n} \right)$$

to denote the gradient vector of $\mathcal{L}(x, y)$ associated with x, and use

$$\nabla_x^2 \mathcal{L}(x, y) := \begin{bmatrix} \frac{\partial^2 \mathcal{L}(x,y)}{\partial x_1^2} & \cdots & \frac{\partial^2 \mathcal{L}(x,y)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathcal{L}(x,y)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 \mathcal{L}(x,y)}{\partial x_n^2} \end{bmatrix}$$

to denote the Hessian matrix of $\mathcal{L}(x, y)$ associated with x.

A dual problem associated with (4.1) is

$$\begin{aligned} \min\ & \mathcal{L}(x,y) - \hat{x}^T s = \mathcal{L}(x,y) - \nabla_x \mathcal{L}(x,y)x \\ \text{s.t.}\ & \nabla_x \mathcal{L}(x,y)^T = (s;0), \\ & \hat{x}, \hat{y}, s \geq 0, \text{ and } \tilde{x}, \tilde{y} \text{ are free.} \end{aligned} \tag{4.2}$$

Using (4.1) and (4.2), one can construct optimality conditions to (4.1) as

$$\begin{aligned} & \nabla_x \mathcal{L}(x,y)^T = (s;0), \\ & a(x) = (z;0), \\ & 0 \leq (s, \hat{y}) \perp (\hat{x}, z) \geq 0, \\ & \tilde{x}, \tilde{y} \text{ are free.} \end{aligned} \tag{4.3}$$

Here $z \in R^{\hat{m}}$ represents the vector of surplus variables corresponding to the concave functions $a_i(\cdot), i = 1, \ldots, \hat{m}$.

Problem (4.3) is a monotone complementarity problem with equality constraints and free variables. Following Andersen and Ye [11], one can formulate a homogenized monotone complementarity problem for (4.3) as follows:

$$\begin{aligned} & \tau \nabla_x \mathcal{L}(x/\tau, y/\tau)^T = (s;0), \\ & \tau a(x/\tau) = (z;0), \\ & -x^T \nabla_x \mathcal{L}(x/\tau, y/\tau)^T - y^T a(x/\tau) = \kappa, \\ & 0 \leq (s, \hat{y}, \kappa) \perp (\hat{x}, z, \tau) \geq 0, \\ & \tilde{x}, \tilde{y} \text{ are free,} \end{aligned} \tag{4.4}$$

where τ and κ are introduced homogeneous variables.

Given a point $\omega \in \mathring{\Omega}$, let us define the residual vectors as

$$\begin{aligned} rv_d &:= \tau g - (s;0), \\ rv_p &:= \tau a(x/\tau) - (z;0), \\ rv_g &:= -\kappa - x^T g - y^T a(x/\tau), \end{aligned} \tag{4.5}$$

and

$$\begin{aligned} J &:= \nabla a(x/\tau), \\ g &:= \nabla_x \mathcal{L}(x/\tau, y/\tau)^T = \nabla c(x/\tau)^T - J^T (y/\tau). \end{aligned}$$

Let

$$(r_p; r_d; r_g) := \eta r := -\eta(rv_p; rv_d; rv_g).$$

Let $\hat{X} := diag(\hat{x})$ and $\hat{Y} := diag(\hat{y})$. Starting from ω, the (perturbed-)Newton method solves the following system of linear equations to compute direction $d_\omega := (d_x; d_z; d_\tau; d_y; d_s; d_\kappa)$:

$$\begin{bmatrix} H & v_2 & -J^T \\ v_1^T & H_g & -v_3^T \\ J & v_3 & 0 \end{bmatrix} \begin{pmatrix} d_x \\ d_\tau \\ d_y \end{pmatrix} - \begin{pmatrix} (d_s;0) \\ d_\kappa \\ (d_z;0) \end{pmatrix} = \begin{pmatrix} r_d \\ r_g \\ r_p \end{pmatrix}, \tag{4.6}$$

and

$$
\begin{aligned}
\hat{X} d_s + S d_{\hat{x}} = r_{\hat{x}s} &:= \gamma\mu(\omega)e - \hat{X}s, \\
Z d_{\hat{y}} + \hat{Y} d_z = r_{z\hat{y}} &:= \gamma\mu(\omega)e - Z\hat{y}, \\
\tau d_\kappa + \kappa d_\tau = r_{\tau\kappa} &:= \gamma\mu(\omega) - \tau\kappa,
\end{aligned} \tag{4.7}
$$

where

$$
\begin{aligned}
\mu(\omega) &:= ((\hat{x}; z; \tau)^T (s; \hat{y}; \kappa))/(\hat{n} + \hat{m} + 1), \\
H &:= \nabla_x^2 \mathcal{L}(x/\tau, y/\tau), \\
v_1 &:= -\nabla c(x/\tau)^T - H(x/\tau), \\
v_2 &:= \nabla c(x/\tau)^T - H(x/\tau), \\
v_3 &:= a(x/\tau) - J(x/\tau), \\
H_g &:= (x/\tau)^T H(x/\tau).
\end{aligned}
$$

We can verify that

$$
\begin{bmatrix} H & v_2 & -J^T \\ v_1^T & H_g & -v_3^T \\ J & v_3 & 0 \end{bmatrix}
$$

is a positive semidefinite matrix. Hence, the complementarity problem (4.4) is monotone.

The coefficient matrix in the system of linear equations (4.6) and (4.7) in general is sparse. By performing simple block elimination on the linear system, we obtain the following alternative formulation, which is an augmented system form [14] with an additional row and column:

$$
\begin{bmatrix} D_z & J & v_3 \\ -J^T & D_x & v_2 \\ -v_3^T & v_1^T & D_g \end{bmatrix}
\begin{pmatrix} d_y \\ d_x \\ d_\tau \end{pmatrix}
=
\begin{pmatrix} r_p + (\hat{Y}^{-1} r_{z\hat{y}}; 0) \\ r_d + (\hat{X}^{-1} r_{\hat{x}s}; 0) \\ r_g + \tau^{-1} r_{\tau\kappa} \end{pmatrix}
:=
\begin{pmatrix} \bar{r}_p \\ \bar{r}_d \\ \bar{r}_g \end{pmatrix}, \tag{4.8}
$$

where

$$
D_x := H + \begin{bmatrix} \hat{X}^{-1}S & 0 \\ 0 & 0 \end{bmatrix}, \quad D_z := \begin{bmatrix} Z\hat{Y}^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \quad \text{and } D_g := H_g + \tau^{-1}\kappa.
$$

Now define two vectors $\bar{u} := (v_3; -v_2)$ and $\bar{v} := (-v_3; v_1)$, and a matrix $M := \begin{bmatrix} D_z & J \\ J^T & -D_x \end{bmatrix}$. Solving (4.8) is equivalent to solving the following two augmented systems:

$$
\begin{aligned}
&Mq = (\bar{r}_p; \bar{r}_d) \quad \text{and} \\
&Mp = \bar{u}.
\end{aligned} \tag{4.9}
$$

Therefore, we have obatined the following components of the step direction:

$$
\begin{aligned}
d_\tau &= \frac{\bar{r}_g - \bar{v}^T q}{D_g - \bar{v}^T p} \quad \text{and} \\
(d_y; d_x) &= q - p d_\tau.
\end{aligned} \tag{4.10}
$$

The remaining components d_s, d_z, d_κ of the step direction can be easily recoverd.

M in (4.9) is a symmetric and indefinite matrix. To solve this system we can use the Bunch-Parlett symmetric matrix factorization method [14] or the quasi-definite method [37]. The Bunch-Parlett method is known to avoid numerical instability associated with free variables. In the current test results `iOptimize` uses the sparse symmetric indefinite matrix factorization library `MA57` [16].

4.2. Primal dual updates. Andersen and Ye [11] consider two types of updates: a linear update and a nonlinear update. Let $\omega = (x; y; \tau; z; s; \kappa) \in \mathring{\Omega}$ be a feasible solution and $d_\omega = (d_x; d_y; d_\tau; d_z; d_s; d_\kappa)$ be a search direction. The linear update computes a new solution ω^+ as the following:

$$\omega^+ := \omega + \alpha d_\omega,$$

where $\alpha \geq 0$ is a given step size. The nonlinear update is a direct result of (2.6). As shown in Lemma 2.3, with $\gamma = 1 - \eta$ the nonlinear update of the dual variables tend to reduce the infeasibilities and the complementarity gap at the same rate, which is theoretically desirable.

Andersen and Ye [11] propose a merit function to measure the progress in their algorithm. If the merit function is reduced to zero, then a complementary solution is obtained. They show that for a suitably choice of γ, the (perturbed-) Newton direction is a descent direction for the merit function.

In our implementation we use the potential function (3.1) to evaluate the progress in the algorithm. Given a search direction and a step size, we compute two trial solutions using the linear and nonlinear updates. We accept the trial solution that reduces the potential function more. We give a pseudo code for updating a solution, and discuss the step size computations in `iOptimize` in the next section.

4.3. Step size computations. To find a controllable step size, Andersen and Ye employ a simple backtracking line search method equipped with safeguards. They require their merit function to be reduced by checking the Goldstein-Armijo rule.

To avoid line search computations, some standard implementations have used a certain fixed distance (step factor) to the boundary. However, a fixed factor $>$ 0.99 may be overly aggressive in the earlier phase of IPM. Mehrotra [25] proposed a heuristic to compute the step factor adaptively. His method adaptively allows for larger (and smaller) step factor values. In our implementation we employ this heuristic to compute step sizes in the primal and dual spaces, and use them to update the point with linear or nonlinear update. This heuristic does not guarantee the reduction of the potential function value (assuming $\nabla_x \mathcal{L}(x, y)$ satisfies a SLC). In such a case, we choose the theoretical descent direction from solving (4.6) and (4.7) with suitable parameters η and γ that guarantees the reduction of the potential function value, and employ a golden-section line search to find a controllable step size (See Section 4.5 for a detailed strategy).

Procedure *computeStepFactor* gives a pseudo code for computing primal (σ_p) and dual (σ_d) step factors, Procedure *computeStepSize* for computing the step size (α) in the primal and dual spaces, and Procedure *updateSolution* for updating a given point ω along a search direction d_ω using linear and nonlinear updates. In procedure *computeStepFactor*, two parameters, θ_l (lower bound) and θ_u (upper bound), are used to safeguard against very tiny or overly aggressive step factors.

4.4. Predictor-corrector technique. The computation of a search direction from (4.6) and (4.7) involves a matrix factorization step and subsequent solves with the factored matrix. The cost of the solves in general is much cheaper, and thus

Procedure 2 $(\sigma_p, \sigma_d) = \mathit{computeStepFactor}(\omega, d_\omega)$

Input: Current point $\omega = (x; y; \tau; z; s; \kappa) \in \mathring{\Omega}$, search direction $d_\omega = (d_x; d_y; d_\tau; d_z; d_s; d_\kappa)$ parameters θ_l and θ_u.

Output: Primal step factor σ_p and dual step factor σ_d.

1: Find the blocking variables:

$$\begin{aligned}
lx &:= \arg\min_{j=1,\dots,\hat{n}} \{-x_j/(d_x)_j \;\text{ if }\; (d_x)_j < 0\},\\
ls &:= \arg\min_{j=1,\dots,\hat{n}} \{-s_j/(d_s)_j \;\text{ if }\; (d_s)_j < 0\},\\
lz &:= \arg\min_{i=1,\dots,\hat{m}} \{-z_i/(d_z)_i \;\text{ if }\; (d_z)_i < 0\},\\
ly &:= \arg\min_{i=1,\dots,\hat{m}} \{-y_i/(d_y)_i \;\text{ if }\; (d_y)_i < 0\},\\
l\tau &:= -\tau/d_\tau \;\text{ if }\; d_\tau < 0,\\
l\kappa &:= -\kappa/d_\kappa \;\text{ if }\; d_\kappa < 0.
\end{aligned}$$

2: Compute an estimate of the duality gap:

$$\begin{aligned}
\mu_x &:= \sum_{j=1}^{\hat{n}} (x_j + (d_x)_j)(s_j + (d_s)_j),\\
\mu_z &:= \sum_{i=1}^{\hat{m}} (z_i + (d_z)_i)(y_i + (d_y)_i),\\
\mu_\tau &:= (\tau + d_\tau)(\kappa + d_\kappa),\\
\mu &:= ((\mu_x + \mu_z + \mu_\tau)(1 - \theta_l)) \,/\, (\hat{n} + \hat{m} + 1).
\end{aligned}$$

3: Compute the step factors for all the variables:

$$\begin{aligned}
\sigma_x &:= \left(\frac{\mu}{s_{lx} + (d_s)_{lx}} - x_{lx}\right)\left(\frac{1}{(d_x)_{lx}}\right),\\
\sigma_s &:= \left(\frac{\mu}{x_{ls} + (d_x)_{ls}} - s_{ls}\right)\left(\frac{1}{(d_s)_{ls}}\right),\\
\sigma_z &:= \left(\frac{\mu}{y_{lz} + (d_y)_{lz}} - z_{lz}\right)\left(\frac{1}{(d_z)_{lz}}\right),\\
\sigma_y &:= \left(\frac{\mu}{z_{ly} + (d_z)_{ly}} - y_{ly}\right)\left(\frac{1}{(d_y)_{ly}}\right),\\
\sigma_\tau &:= \left(\frac{\mu}{\kappa_{l\tau} + (d_\kappa)_{l\tau}} - \tau_{l\tau}\right)\left(\frac{1}{(d_\tau)_{l\tau}}\right),\\
\sigma_\kappa &:= \left(\frac{\mu}{\tau_{l\kappa} + (d_\tau)_{l\kappa}} - \kappa_{l\kappa}\right)\left(\frac{1}{(d_\kappa)_{l\kappa}}\right).
\end{aligned}$$

4: Compute primal and dual step factors:

$$\begin{aligned}
\sigma_p &:= \max\{\theta_l, \min\{\sigma_x, \sigma_z, \sigma_\tau, \theta_u\}\},\\
\sigma_d &:= \max\{\theta_l, \min\{\sigma_s, \sigma_y, \sigma_\kappa, \theta_u\}\}.
\end{aligned}$$

Procedure 3 $\alpha = \textit{computeStepSize}(\omega, d_\omega)$

Input: Current point $\omega = (x; z; \tau; y; s; \kappa) \in \mathring{\Omega}$ and search direction $d_\omega = (d_x; d_z; d_\tau; d_y; d_s; d_\kappa)$.
Output: Primal and dual step size α.
1: Compute the step sizes to the boundary

$$\begin{aligned}
\alpha_x &:= \min_{j=1,\ldots,\hat{n}} \left\{-x_j/(d_x)_j \;\text{ if }\; (d_x)_j < 0\right\},\\
\alpha_s &:= \min_{j=1,\ldots,\hat{n}} \left\{-s_j/(d_s)_j \;\text{ if }\; (d_s)_j < 0\right\},\\
\alpha_z &:= \min_{i=1,\ldots,\hat{m}} \left\{-z_i/(d_z)_i \;\text{ if }\; (d_z)_i < 0\right\},\\
\alpha_y &:= \min_{i=1,\ldots,\hat{m}} \left\{-y_i/(d_y)_i \;\text{ if }\; (d_y)_i < 0\right\},\\
\alpha_\tau &:= -\tau/d_\tau \;\text{ if }\; d_\tau < 0,\\
\alpha_\kappa &:= -\kappa/d_\kappa \;\text{ if }\; d_\kappa < 0.
\end{aligned}$$

2: Call Procedure $\textit{computeStepFactor}(\sigma_p, \sigma_d, \omega, d_\omega)$ to compute the primal step factor (σ_p) and the dual step factor (σ_d).
3: Compute primal and dual step sizes:

$$\begin{aligned}
\alpha_p &:= \sigma_p \times \min\left\{\alpha_x, \alpha_z, \alpha_\tau\right\},\\
\alpha_d &:= \sigma_d \times \min\left\{\alpha_s, \alpha_y, \alpha_\kappa\right\},
\end{aligned}$$

4: Set $\alpha := \min\left\{\alpha_p, \alpha_d\right\}$.

Procedure 4 $\omega^+ = \textit{updateSolution}(\omega, d_\omega, \alpha)$

Input: Current point $\omega = (x; y; \tau; z; s; \kappa) \in \mathring{\Omega}$, search direction $d_\omega = (d_x; d_y; d_\tau; d_z; d_s; d_\kappa)$, primal and dual step size α.
Output: Updated point $\omega^+ \in \mathring{\Omega}$.
1: Compute the primal and dual updates:
$\omega^l := \omega + \alpha(d_x; d_y; d_\tau; d_z; d_s; d_\kappa)$.
2: Compute the nonlinear update:

$$\begin{aligned}
x^n &:= x + \alpha d_x,\\
y^n &:= y + \alpha d_x,\\
\tau^n &:= \tau + \alpha d_\tau,\\
z_i^n &:= \left((1-\alpha\eta)(-rv_p)_i + \tau^n a_i(x^n/\tau^n)\right),\; i = 1, \ldots, \hat{m},\\
s_j^n &:= \left((1-\alpha\eta)(-rv_d)_j + \tau^n \nabla_x \mathcal{L}_j(x^n/\tau^n, y^n/\tau^n)^T\right),\; j = 1, \ldots, \hat{n},\\
\kappa^n &:= \left((1-\alpha\eta)(-rv_g) - (x^n)^T \nabla_x \mathcal{L}(x^n/\tau^n, y^n/\tau^n)^T - (y^n)^T a(x^n/\tau^n)\right).
\end{aligned}$$

3: Check if it is beneficial to take linear update:
4: **if** $\Phi_\rho(\omega^n) \geq \Phi_\rho(\omega^l)$ **then**
5: Set $\omega^+ := \omega^l$.
6: **else**
7: Set $\omega^+ := \omega^n$.
8: **end if**

it leads to the idea of reusing the factorization of the linear system to compute a better search direction. Several approaches have been developed based on this idea. For example, Mehrotra corrector [25] and Gondzio's higher-order corrector [18] are proper choices. The presented empirical results are based on using Mehrotra corrector only. We now describe an adaption of this technique for our context.

Let $\omega^k \in \mathring{\Omega}$ be the current point, we first compute an affine (pure Newton) direction by solving (4.6) and (4.7) with $\eta = 1$ and $\gamma = 0$, that is,

$$\begin{aligned} &r_p := -rv_u^k,\ r_{\hat{x}s} := -\hat{X}^k s^k,\\ &r_d := -rv_p^k,\ r_{z\hat{y}} := -Z^k \hat{y}^k,\\ &r_g := -rv_d^k,\ r_{\tau\kappa} := -\tau^k \kappa^k. \end{aligned} \tag{4.11}$$

The resulting direction is denoted by $d_\omega^a = (d_x^a; d_z^a; d_\tau^a; d_y^a; d_s^a; d_\kappa^a)$. Next we compute a step size α^a by calling Procedure $\alpha^a =$*computeStepSize*(ω^k, d_ω^a) where ω^k and d_ω^a are the input, and α^a is the output, and then compute a tentative point ω^a by calling Procedure $\omega^k =$ *updateSolution*$(\omega^a, d_\omega^a, \alpha^a)$ where ω^k, d_ω, and α^a are the input and ω^a is the output.

Mehrotra suggested to determine the barrier parameters γ^a and η^a by using the predicted reduction in the complementarity gap $\mu(\omega^a)$ along the affine direction. The idea behind this heuristic is to reduce the centrality parameter γ^a more when the IPM can make good progress towards optimality in the affine direction. In our implementation we compute the barrier parameters γ^a and η^a as follows:

$$\begin{aligned} &\gamma^a = \min\left\{0.99, \max\left\{\left(\tfrac{\mu(\omega^a)}{\mu(\omega^k)}\right)^3, 0.01\right\}\right\},\\ &\eta^a = 1-\gamma^a. \end{aligned} \tag{4.12}$$

The new complementarity gap is given by

$$\begin{aligned} (\hat{X}^k + D_{\hat{x}}^a)(s^k + d_s^a) &= D_{\hat{x}}^a d_s^a,\\ (Z^k + D_z^a)(\hat{y}^k + d_{\hat{y}}^a) &= D_z^a d_{\hat{y}}^a,\\ (\tau^k + d_\kappa^a)(\kappa^k + d_\tau^a) &= d_\tau^a d_\kappa^a, \end{aligned}$$

where $D_{\hat{x}}^a := diag(d_{\hat{x}}^a)$ and $D_z^a := diag(d_z^a)$. Since the new complementarity product ideally should be equal to $\mu(\omega^a)e$, the new search direction compensates for the error and thus is computed with the following right-hand side:

$$\begin{aligned} &r_p := -\eta^a rv_p,\ r_{\hat{x}s} := \gamma^a\mu(\omega^a)e - \hat{X}^k s^k - D_{\hat{x}}^a d_s^a,\\ &r_d := -\eta^a rv_d,\ r_{z\hat{y}} := \gamma^a\mu(\omega^a)e - Z^k\hat{y}^k - D_z^a d_{\hat{y}}^a,\\ &r_g := -\eta^a rv_g,\ r_{\tau\kappa} := \gamma^a\mu(\omega^a) - \tau^k\kappa^k - d_\tau^a d_\kappa^a. \end{aligned}$$

The resulting direction here is called "Mehrotra direction" and is denoted by d_ω^m.

4.5. Search guided by the potential function. The implementation logic in `iOptimize` that combines the predictor and corrector search directions at each iteration is now given. At the core of our implementation is to ensure sufficient reduction in the potential function at each iteration, while using Mehrotra direction to further improve reduction in the potential function.

At each iteration k we first check if $\Phi_{\hat{n}+\hat{m}+1}(\omega^k) \le \iota(\hat{n}+\hat{m}+1)\log(\hat{n}+\hat{m}+1)$. This safeguard is used to make sure that our algorithm generates a maximal complementary solution. If it is not satisfied, then we directly jump to the theoretical direction computation phase given later; otherwise we compute a Mehrotra direction d_ω^m and use it to generate a tentative step size ω^m by calling Procedure $\alpha^m = \mathit{computeStepSize}(\omega^k, d_\omega^m)$, where ω^k and d_ω^m are the input, and α^m is the output. Then we compute a tentative point ω_m by calling Procedure $\omega^m = \mathit{updateSolution}(d_\omega^m, \omega^k, \alpha^m)$, where ω^k, d_ω^m, and α^m are the input and ω^m is the output. Though d_ω^m generally enhances the convergence, it does not guarantee reduction of the potential function value, or maintain the overall decrease in the infeasibility to complementarity ratio (Assumption 3.1). Hence we determine the quality of the Mehrotra direction by evaluating the potential function value at ω^m and check if ω^m satisfies Assumption 3.1. We note that checking 3.1(ii) requires the knowledge of the scaled Lipschitz constant, which may not be available in practice; hence, we simply check if $\mu(\omega^m) < \mu(\omega^k)$. If $\Phi_\rho(\omega^m) < \Phi_\rho(\omega^k)$ and Assumption 3.1 is satisfied, we set $\omega^{k+1} := \omega^m$; otherwise we compute a theoretical direction d_ω^t from (4.6) and (4.7), and iteratively employs a golden-section linear search method [13] to find a controllable step size α^t, and use it to compute a new tentative point ω^t by calling Procedure $\omega^t = \mathit{updateSolution}(\omega^k, d_\omega^t, \alpha)$, where ω^k, d_ω^t, and α^t are the input and ω^t is the output. We terminate the line search if the potential function achieves sufficient decrease.

Note that if the scaled Lipschitz constant is large, then many function evaluations may be needed in the golden section line search. To avoid this we simply take the step after 4 iterations of golden section method, and monitor the progress of the potential function over multiple interior point iterates. We ensure that the potential function is eventually reduced over these multiple iterates.

A flowchart of the implementation of search direction computations is given in Figure 4.1.

4.6. Starting point and termination criteria. Andersen and Ye [11] show that the HMCP IPM achieves consistently good performance even for a trivial starting point

$$(\hat{x}^0; \tilde{x}^0; \hat{y}^0; \tilde{y}^0; \tau^0; z^0; s^0; \kappa^0) = (e; 0; e; 0; 1; e; e; 1). \tag{4.13}$$

This trivial starting point is used in the results reported in this paper. However, it might be possible to further improve the computational results by implementing a (better) non-trivial starting point.

Another important issue is the termination criteria. We choose the termination criteria used by Andersen and Ye [11] with slight modifications. Define the primal and dual objective values by

$$\texttt{PriObj} := c(x/\tau), \quad \text{and} \quad \texttt{DualObj} := \mathcal{L}(x/\tau, y/\tau) - \nabla_x \mathcal{L}(x/\tau, y/\tau)(x/\tau).$$

Let rv_p^k, rv_d^k, and rv_g^k be the vectors of the residuals corresponding to the kth iterate. We terminate the algorithm whenever one of the following criteria is satisfied:

- For QP, an (approximate) optimal solution is obtained if

$$\begin{aligned}
\frac{\|rv_p^k\|}{\tau\{1+\|b\|\}} &< 10^{-8},\\
\frac{\|rv_d^k\|}{\tau\{1+\|c\|\}} &< 10^{-8},\\
\min\left\{\frac{|\texttt{DualObj}-\texttt{PriObj}|}{1+|\texttt{DualObj}|}, \frac{|rv_g^k|}{1+|rv_g^0|}\right\} &< 10^{-8}.
\end{aligned}$$

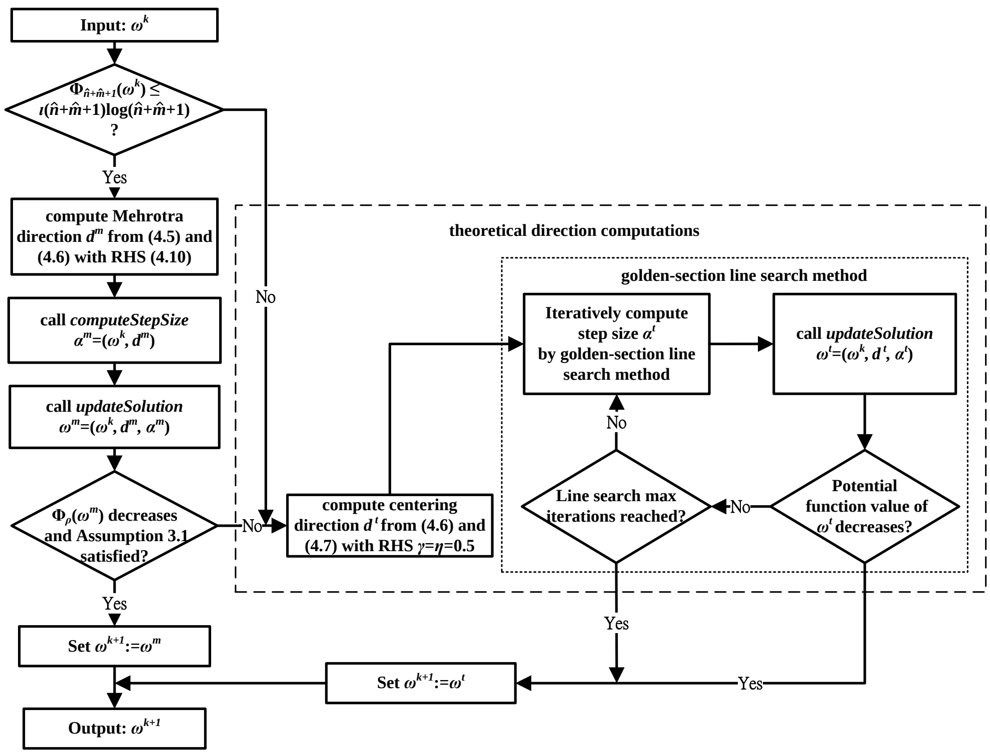


Fig. 4.1: Implementation of search direction computations

- For QCQP and CP, an (approximate) optimal solution is obtained if

$$\frac{\|rv_p^k\|}{\max\left\{1, \|rv_p^0\|\right\}} < 10^{-8},$$
$$\frac{\|rv_d^k\|}{\max\left\{1, \|rv_d^0\|\right\}} < 10^{-8},$$
$$\frac{|rv_g^k|}{\max\left\{1, |rv_g^0|\right\}} < 10^{-6}.$$

- For QP, QCQP, and CP, the problem is (near) infeasible if

$$\mu(\omega^k) < \mu(\omega^0)10^{-8} \quad \text{and} \quad \frac{\tau^k}{\min\{1, \kappa^k\}} < \frac{\tau^0}{\kappa^0}10^{-12}.$$

We note that Andersen [9] developed an infeasibility certificate for the homogeneous monotone complementarity problems, that provide further information about whether the primal or dual problem is infeasible. This feature, however, has not been implemented in the current version of `iOptimize`.

5. Computational results. `iOptimize` was compiled with the Visual Studio 2010 compiler. All computations were performed on a 3.2 GHz Intel Dual-Core CPU machine with 4GB RAM. The program is run on one CPU only. An `AMPL` [1] interface

Parameter	Value	Explanation of the parameter
θ	$\hat{n}+\hat{m}+1$	Parameter in the potential function (3.1)
ρ	100	Parameter in the potential function (3.1)
(θ_l, θ_u)	(0.999, 1.0)	Parameters for QP in Procedure *computeStepFactor*
(θ_l, θ_u)	(0.9, 1.0)	Parameters for QCQP and CP in Procedure *computeStepFactor*
ι	$(\hat{n}+\hat{m}+1)^2$	Parameter in (3.28)
$(\Theta(1), \Omega(1))$	$(10^{-2}, 10^2)$	Parameters in Assumption 3.1

Table 5.1: The default parameter settings in `iOptimize`

is implemented in `iOptimize` to read the `AMPL` nonlinear models and the corresponding first and second-derivatives. All problems are solved using the same default parameter settings, which are shown in Table 5.1.

To provide a clearer summary of the results, we use performance profiles [15]. Consider a set A of n_a algorithms, a set P of n_p problems, and a performance measure $m_{a,p}$, e.g., node count or computation time. We compare the performance on problem p by algorithm a with the best performance by any algorithm on this problem using the following performance ratio

$$r_{p,a} = \frac{m_{p,a}}{\min\{m_{p,a} \mid a \in A\}}.$$

We therefore obtain an overall assessment of the performance of the algorithm by defining the following value

$$\rho_a(\tau) = \frac{1}{n_p} size\{p \in P \mid r_{p,a} \leq \tau\}.$$

This represents the probability for algorithm a that the performance ratio $r_{p,a}$ is within a factor τ of the best possible ratio. The function $\rho_a(\cdot)$ represents the distribution function for the performance ratio.

5.1. Computational results on feasible problems. We solve the convex QP problems from Maros and Mézáros's QP test set [24], the convex QCQP problems from Mittelmann's QCQP test set [5] and Vanderbei's `AMPL` nonlinear `Cute` and `Non-Cute` set [2], the general convex CP problems from Vanderbei's `AMPL` nonlinear test set [2], and from Leyffer's mixed-integer nonlinear test set [6] while ignoring the integrality requirements.

Tables A.1–A.4 give the computational results that are obtained with the default parameter settings. Each row in these tables contains the profiles of the problems before and after presolving, the primal and the dual objectives, the number of iterations (Avg Iters), the computation times (Presolve Time and Solver Time), the proportion of the effort in evaluating the potential function (Prop Φ (%)), and the average number of search directions used (Avg Dirs) (here two means `iOptimize` uses pure Newton direction and Mehrotra corrector, whereas one means `iOptimize` uses the theoretical direction only). We note that the QCQP problems in Mittelmann's test set are equivalent to those of Maros and Mézáros's QP test set, as each QP problem is reformulated as a QCQP problem with an additional quadratic constraint and a free variable.

Tables A.5–A.8 provide a comparison between `MOSEK` homogeneous and self-dual interior point optimizer and `iOptimize` under their default settings. We focus on the number of iterations needed to achieve the desired accuracy by both solvers.

Table A.5 gives the computational results for solving 127 QP problems. The average number of iterations used by `iOptimize` (15.92 iters) and `MOSEK` (16.22 iters) are comparable. All problems can be solved within 50 iterations by both solvers. Note that for three problems (`liswet4`, `powell20`, and `qpilotno`) `MOSEK` terminates with `NEAR_OPTIMAL` status. This implies `MOSEK` cannot compute a solution that has the prescribed accuracy.

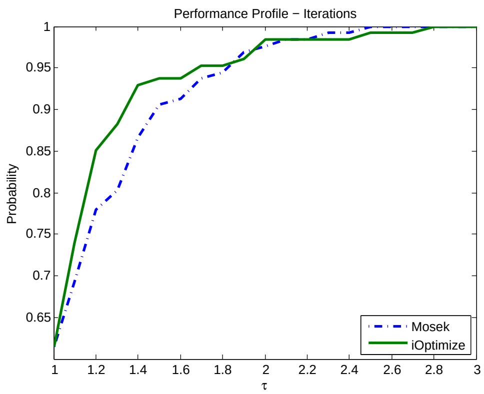
Table A.6 gives the computational results for solving Mittelmann's QCQP test set problems. Here two different average performance results are provided. "Avg1" provides the average performance on problems for which `MOSEK` terminate with `OPTIMAL`, `NEAR_OPTIMAL` or `UNKNOWN` status. For problems `QQ-aug2dqp` and `QQ-powell20`, `MOSEK` is not able to converge to the optimal solution within 400 iterations. We exclude both problems in "Avg1". For problems `QQ-laser`, `QQ-qforplan`, and `QQ-qseba`, `MOSEK` requires more than 50 interior point iterations to successfully solve. We recompute the average performance statistics again in "Avg2" by further excluding these three examples. Considering average performances in these experiments, `iOptimize` (Avg1=16.43 iters, Avg2=16.26 iters) is significantly better than `MOSEK` (Avg1=20.91 iters, Avg2=18.34 iters). Note that for `QQ-liswet1`, `QQ-liswet7`–`QQ-liswet12`, `MOSEK` terminates with `UNKNOWN` status. This may happen when `MOSEK` converges slowly, and hence the primal, dual, or gap residuals may not attain the prescribed accuracy. On the other hand, `iOptimize` is able to terminate with `OPTIMAL` status for all the test problems within 40 iterations.

Table A.7 contains the computational results for solving 11 QCQP problems in the Vanderbei's `AMPL` nonlinear test set. Comparing the size to those in Mittelmann's QCQP test set, the problems here are relatively small, but may include a quadratic objective term and more than one quadratic constraint. Considering the average iterations required for solving the problems, both solvers generate similar computational results (`MOSEK` Avg=11.36 iters versus `iOptimize` Avg=11.64 iters).

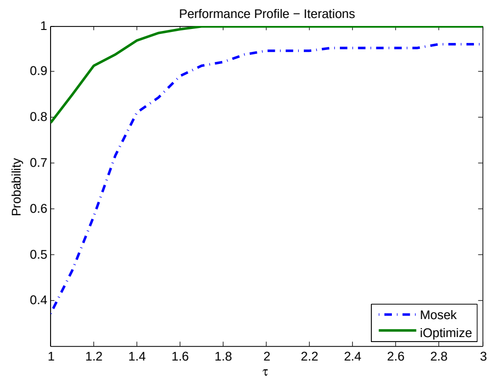
Table A.8 contains the computational results for solving 46 general CP problems. Thirty five of the CP problems are selected from Vanderbei's `AMPL` nonlinear `Cute` and `Non-Cute` set, and 11 of the CP problems are selected from Leyffer's mixed-integer nonlinear test set. Two different average performance statistics are provided. "Avg1" provides the average performance on all the instances, except problem `dallass`, for which `MOSEK` fails to converge to an optimal solution within 400 iterations. For problems `dallasm`, `dallasl`, and `elena-s383`, `MOSEK` respectively requires 128, 74, and 85 iterations before terminating with `OPTIMAL` status. Thus we recompute the average statistics again in "Avg2" by additionally excluding these three problems. Comparing "Avg1", the average number of iterations used by `iOptimize` (14.00 iters) is better than those used by `MOSEK` (18.46 iters). By excluding these problems for which `MOSEK` requires more than 50 iterations to solve ("Avg2"), the performance comparison between both solvers becomes comparable, though `iOptimize` (12.69 iters) still performs slightly better than `MOSEK` (13.07 iters). Note that for problems `antenna-antenna2`, `firfilter-fir_convex`, and `wbv-lowpass2`, `MOSEK` terminates with `NEAR_OPTIMAL` status.

The average number of directions used by `iOptimize` is ≈ 1.93, i.e., Mehrotra direction was not used in about 7% of the iterations. It implies that the average proportion of the computations for the theoretical direction with a fixed step size is not significant. In these experiments we observed that Assumption 3.1 and the requirement $\Phi_{\hat{n}+\hat{m}+1}(\omega^k) \leq \iota(\hat{n}+\hat{m}+1)\log(\hat{n}+\hat{m}+1)$ is always satisfied over all the iterates, implying that `iOptimize` uses the theoretical direction only in the case where Mehrotra direction fails to reduce the potential function value. We also note that the average proportion of the efforts used for the potential function evaluations is 3.1%, suggesting that the computational demands for evaluating the potential function is also not significant, while a potential reduction algorithm is implemented that guarantees global linear polynomial time convergence.

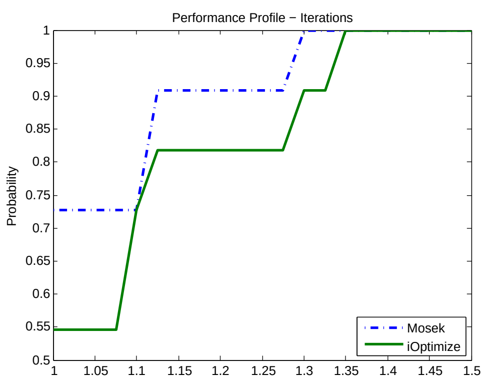
Figure 5.1 shows the performance profiles for feasible problems comparing the iteration performances of `iOptimize` and `MOSEK`.



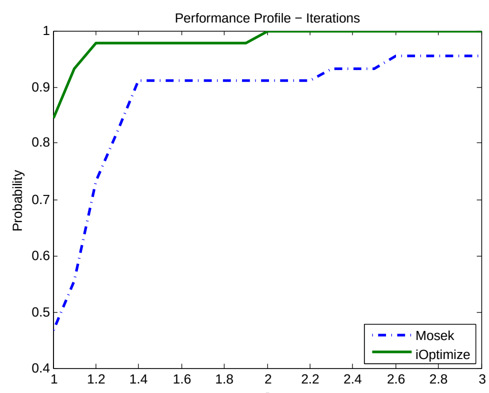
(a) Maros and Mézáros's QP feasible problems



(b) Mittelmann's QCQP feasible problems



(c) Vanderbei's QCQP feasible problems



(d) General CP feasible problems

Fig. 5.1: Performance profiles of `iOptimize` and `MOSEK` on feasible problems

5.2. Computational results on infeasible problems. In this section we test the performance of `iOptimize` for solving the QCQP and CP infeasible problems. With the lack of the infeasible test set in the public domain, we create infeasible test problems by adding invalid objective constraints to the feasible problems.

Let x^* be an optimal solution of problem (4.1). We create a corresponding infeasible problem that has the form:

$$\begin{aligned} \min\ & c(x) \\ \text{s.t. } & a_i(x) \geq 0, i = 1, \ldots, m_n, \\ & a_i(x) = 0, i = \hat{m} + 1, \ldots, m, \end{aligned} \tag{5.1}$$

$$c(x) \leq c(x^*) - \lambda,$$
$$\hat{x} \geq 0, \text{ and } \tilde{m} \text{ is free,}$$

where λ is a constant with positive value. We choose

$$\lambda := \begin{cases} 2|c(x^*)|, & \text{if } |c(x^*)| \geq 100, \\ 100, & \text{if } |c(x^*)| < 100. \end{cases}$$

Hence, the problems are made significantly infeasible.

Using this construction we create 127 infeasible QCQP problems from Maros and Mézáros's QP test set [24], 11 QCQP infeasible problems from Vanderbei's `AMPL` nonlinear `Cute` and `Non-Cute` set [2], 35 CP infeasible problems from Vanderbei's `AMPL` nonlinear `Cute` and `Non-Cute` set [2], and 11 CP infeasible problems from Leyffer's mixed-integer nonlinear test set [6] without integrality requirements.

Tables A.9–A.11 give the computational results that are obtained with the default parameter settings. The number of iterations (Avg Iters), the computation times (Presolve Time and Solver Time), the average number of search directions used (Avg Dirs), and the proportion of the effort in evaluating the potential function (Prop Φ (%)) are given. Tables A.12–A.14 provide a comparison between `MOSEK` and `iOptimize` under their default settings.

We focus on the number of iterations needed to detect infeasibility by both solvers. Table A.12 contains the computational results for solving 127 QCQP infeasible problems created from Maros and Mézáros's QP test set. For 38 problems `MOSEK` fails to detect infeasibility within 400 iterations. For problems `iQQ-qscagr25` and `iQQ-qstandat`, `MOSEK` terminates with `UNKNOWN` status. We compute the average iterations in "Avg1" by excluding these 38 examples. Comparing "Avg1", `iOptimize` (28.15 iters) requires approximately 30% fewer iterations than `MOSEK` (42.20 iters). This is because `MOSEK` requires more than 60 iterations to detect the infeasibility in 12 problems. We further exclude these 12 problems and recompute the average iterations in "Avg2". Comparing "Avg2", the performance of `iOptimize` (28.15 iters) is comparable to that of `MOSEK` (27.39 iters). However, `iOptimize` is able to correctly detect infeasibility for each the problem within 50 interior point iterations.

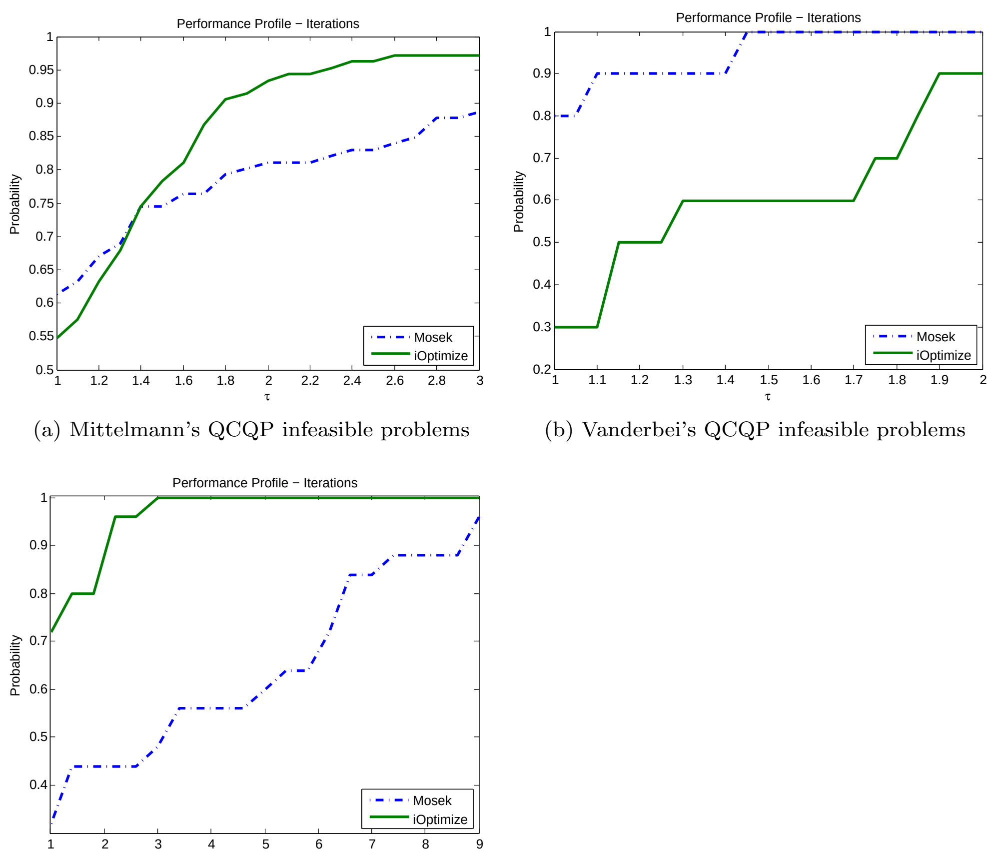
Table A.13 has the computational results for solving the 11 QCQP infeasible problems created from Vanderbei's `AMPL` nonlinear test set. We provide the average statistics by excluding problem `ipolak4`, which `MOSEK` fails to solve within 400 iterations. Though the comparison shows that `iOptimize` (24.60 iters) requires approximately 20% more iterations than `MOSEK` (20.30 iters), `iOptimize` correctly solves all these infeasible problems.

Table A.14 has the computational results for solving 46 CP infeasible problems created from Vanderbei's `AMPL` nonlinear `Cute` and `Non-Cute` set, and Leyffer's mixed-integer nonlinear test set. For problems `idallasl`, `idallasm`, and `idallass`, `MOSEK` receives function evaluate error from `AMPL`. `iOptimize` receives the same error while solving problem `idallasl`. In seven problems `MOSEK` fails to declare the infeasible status within 400 iterations. We compute the average statistics "Avg2" by excluding these 10 problems. Comparing "Avg1", `MOSEK` (66.33 iters) requires significantly more iterations than `iOptimize` (19.44 iters). This happens because `MOSEK` requires more than 60 iterations to detect infeasibility in 13 problems whereas `iOptimize` needs this in only one problem. We further exclude these 10 problems and recompute the average statistics in "Avg2". Comparing "Avg2", `iOptimize` (16.04 iters) requires 60% more iteration than `MOSEK` (11.74 iters). This is because `MOSEK` is able to detect infeasibility during the preprocessing phase in 11 cases; on the other hand, `iOptimize` preprocessor does not detect any infeasibility. This highlights the importance of advanced preprocessing implementations, which is not the focus of research described in the current paper.

Note that for problem `antenna-iantenna2`, `iOptimize` requires 186 iterations to solve, which is significantly larger than the average iterations (28.07 iters). With a fixed step factor parameter settings $(\theta_l, \theta_u) = (0.9, 0.9)$, `iOptimize` can solve this problem in 46 iterations.

Observe that the average number of directions used for this problem is around 1.39, which is significantly smaller than that used for solving the feasible problems. This suggests the importance of using the theoretical direction and potential function to decide the usefulness of a direction.

Figure 5.2 shows the performance profiles for the infeasible problems comparing the iteration performances of `iOptimize` and `MOSEK`. Note that problems which `iOptimize` or `MOSEK` fails to solve or solves them during the preprocessing phase are not considered.



(a) Mittelmann's QCQP infeasible problems

(b) Vanderbei's QCQP infeasible problems

(c) General CP infeasible problems

Fig. 5.2: Performance profiles of `iOptimize` and `MOSEK` on infeasible problems

5.3. Computational comparison with general nonlinear solvers. In this section we compare `iOptimize` with general nonlinear solver `Knitro` and `Ipopt` on CP feasible and infeasible problems. The computational results on QP and QCQP problems are not provided because these problems are stored in `MOSEK` QPS format, which `Ipopt` and `Knitro` do not support. Note that `Knitro` provides three different algorithms for solving the nonlinear problems. In our comparison, we choose the direct primal-dual IPM. For `Ipopt`, we use `MA57` library for the sparse symmetric indefinite matrix factorization, which is the same in `iOptimize`. Default parameter settings are used for both solvers expect that maximum iterations allowed is set at 400.

We first focus on the computational results for solving the CP feasible problems. The last two columns of Table A.8 contain the number of iterations used by `Ipopt` and `Knitro`. Considering the average number of iterations used for solving these problems, `iOptimize` (12.77 iters) ranked first, `Knitro` (18.95 iters) ranked second, whereas `Ipopt` (27.84 iters) ranked last. For problems `dallasm` and `polak3`, both `Knitro` and `Ipopt` received a function evaluation error from `AMPL`. `Ipopt` failed to solve `dallass` to optimality in 400 iterations. Observe that `Ipopt` requires more than 50 iterations for three problems and `Knitro` requires more than 50 iterations for five problems, whereas `iOptimize` solves all feasible CP problems within 50 iterations.

Now we focus on the CP infeasible problems. Computational results from `Ipopt` and `Knitro` are reported in the last two columns of Table A.14. `Knitro` failed to solve 13 problems within 400 iterations, whereas `Ipopt` fails in three problems. `Knitro` received an `AMPL` function evaluation error in three problems and `Ipopt` received the same error in four problems. Note that both `Knitro` and `Ipopt` terminate with a near optimal status in one problem. In three problems `Ipopt` reported error in feasibility restoration phase. Overall, `Ipopt` successfully detects the infeasibility in 35 problems and `Knitro` in 29 problems.

6. Concluding remarks. In this paper we have extended Mehrotra-Huang potential function [26] to the context of MCP, and shown that our potential reduction interior point method generates a maximal complementary solution with desired accuracy in polynomial time if a scaled Lipschitz condition is satisfied. However, the main emphasis of the analysis is not to improve the best known complexity, but to make sure the interior point method maintains the global linear polynomial time convergence properties while achieving practical performance. We have implemented an interior point solver in a software package named `iOptimize` that does not ignore the theory of potential function in the context of interior point methods, while adding only a small additional computational burden. Computational comparisons on feasible and infeasible test problems show that, in terms of iteration counts and the number of problems solved, `iOptimize` appears to perform more efficiently and robustly than a mature commercial solver `MOSEK`. We also find that `iOptimize` seems to detect infeasibility more reliably than general nonlinear solvers `Ipopt` and `Knitro`. It might be possible to further improve the computational results by implementing a better preprocessor, a non-trivial starting point, or better step size computations. However, that was not the focus of the research reported here.

At the core of the solver is the matrix factorization method, for which `iOptimize` uses Bunch-Parlett factorization [14]. The method used in `MOSEK` is not known to us. However, we point out that numerical instability may result if quasi-definite system approach is used [11, 37]. Bunch-Parlett factorization in general is numerically stabler, but requires additional computational effort.

We also implemented a version of Gondzio's higher-order correction strategy [18] in iOptimize. Our implementation of this strategy is a direct extension of the approach in Mehrotra and Huang [26]. Computational results show that iOptimize using Mehrotra direction and up to two additional Gondzio's correctors for the QP problems saves only approximately 9% iterations. However, we did not see any improvement on QCQP and CP problems. The practical value of using other correction strategies is a topic of future research.

We note that the computational performance for a problem can be improved possibly with different parameter settings. For example, with parameters $(\theta_l, \theta_u) = (0.99, 1)$, the average iterations for solving the QCQP and CP feasible problems is reduced further by about 6%. However, this number is increased by about 9% while solving the QCQP and CP infeasible problems. In fact, for antenna-iantenna_vareps, iOptimize under default setting ($(\theta_l, \theta_u) = (0.9, 1)$) needs only 47 iterations to correctly detect infeasibility whereas with parameters $(\theta_l, \theta_u) = (0.99, 1)$ it will require 454 iterations. A very aggressive step to the boundary is not desirable for infeasible and general convex programming problems.

Finally, we note that a small but difficult two-variable example can be found at http://mosek.com/fileadmin/talks/func_versus_conic.pdf, (page 29). With the default setting iOptimize uses 24 iterations to solve this example to optimality whereas MOSEK with default setting uses 124 iterations.

REFERENCES

[1] AMPL*: A modeling language for mathematical programming.* www.ampl.com.

[2] AMPL *nonlinear test problems.* http://www.orfe.princeton.edu/~rvdb/ampl/nlmodels/.

[3] COIN-OR Ipopt *interior point optimizer.* http://www.coin-or.org/ipopt/.

[4] Knitro*: A aolver for nonlinear optimization.* http://www.ziena.com/knitro.html.

[5] *Mittelmann's convex QCQP test problems.* http://plato.asu.edu/ftp/ampl_files/qpdata_ampl/.

[6] *Mixed integer nonlinear programming (minlp) test problems.* http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP.

[7] MOSEK*: A high performance solftware for large-scale optimization problems.* http://www.mosek.com/.

[8] Netlib. http://www.netlib.org/lp/index.html.

[9] E.D. Andersen, *On primal and dual infeasibility certificates in a homogeneous model for convex optimization*, SIAM Journal On Optimization, 11 (2000), pp. 380–388.

[10] E.D. Andersen, C. Roos, and T. Terlaky, *On implementing a primal-dual interior-point method for conic quadratic optimization*, Mathematical Programming, 95 (2003), pp. 249–277.

[11] E.D. Andersen and Y. Ye, *A computational study of the homogeneous algorithm for large-scale convex optimization*, Computational Optimization and Applications, 10 (1998), pp. 243–269.

[12] ———, *On a homogeneous algorithm for the monotone complementarity problem*, Mathematical Programming, 84 (1999), pp. 375–399.

[13] M.S. Bazarra, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory And Algorithms*, John Wiley & Sons, New York, 2 ed., 1993.

[14] J.R. Bunch and B.N. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 8 (1971), pp. 639–655.

[15] E. Dolan and J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213.

[16] I.S. Duff, *MA57–a code for the solution of sparse symmetric definite and indefinie systems*, ACM Transactions on Mathematical Software, 30 (2004), pp. 118–144.

[17] D. Goldfarb and S. Mehrotra, *A relaxed variant of Karmarkar's algorithm*, Mathematical Programming, 40 (1988), pp. 285–315.

[18] J. Gondzio, *Multiple centrality corrections in a primal-dual method for linear programming*, Computational Optimization and Applications, 6 (1996), pp. 137–156.

[19] O. Güler, *Existence of interior points and interior paths in nonlinear monotone complementarity problems*, Mathematics of Operations Research, 18 (1993), pp. 148–162.

[20] O. Güler and Y. Ye, *Convergence behavior of interior-point algorithms*, mathematical Programming, 60 (1993), pp. 215–228.
[21] N. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
[22] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementary problems*, Mathematical Programming, 50 (1991), pp. 331–342.
[23] Z.-Q. Luo, J.F. Sturm, and S. Zhang, *Conic convex programming and seld-dual embedding*, Optimization Method and Software, 14 (2000), pp. 169–218.
[24] I. Maros and C. Mézáros, *A repository of convex quadratic programming problems*, Optimization Methods and Software, 11&12 (1999), pp. 671–681.
[25] S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization, 2 (1992), pp. 575–601.
[26] S. Mehrotra and K.-L. Huang, *Computational experience with a modified potential reduction algorithm for linear programming*, Optimization Method and Software, 27 (2012), pp. 865–891.
[27] M.D.C. Monteiro and I. Adler, *An extension of Karmarkar type algorithm to a class of convex separable programming problems with global rate of convergence*, Mathematical of Operations Research, 15 (1990), pp. 408–422.
[28] Y.E. Nesterov and M.J. Todd, *Self-scaled barriers and interior-point methods for convex programming*, Mathematics of Operations Research, 22 (1997), pp. 1–42.
[29] F. Porta and Y. Ye, *Interior-point methods for nonlinear complementarity problem*, Journal of Optimization Theory and Applications, 68 (1996), pp. 617–642.
[30] F.A. Potra and Y. Ye, *A quadratically convergent polynomial algorithm for solving entropy optimization problems*, SIAM Jorunal on Optimization, 3 (1993), pp. 843–860.
[31] ———, *Interior point methods for nonlinear complementarity problems*, Journal of Optimization Theory and Application, 88 (1996), pp. 617–647.
[32] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 1996.
[33] A. Skajaa, J.B. Jørgensen, and P.C. Hansen, *On implementing a homogeneous interior-point algorithm for nonsymmetric conic optimization*, `http://www.imm.dtu.dk/English/Service/Phonebook.aspx?lg=showcommon&id=f30a529b-4c6b-4642-a67b-cca4f74d0a3a`, (2011).
[34] J.F. Sturm, *Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones*, Optimization Method and Software, 11–12 (1999), pp. 625–653.
[35] K. Tanabe, *Centered newton method for mathematical programming*, System Modelling and Optimization, 113 (1988), pp. 197–206.
[36] M.J. Todd and Y. Ye, *A centered projective algorithm for linear programming*, Mathematics of Operations Research, 15 (1990), pp. 508–529.
[37] R.J. Vanderbei, *Symmetric quasi-definite matrices*, SIAM Journal on Optimization, 5 (1995), pp. 100–113.
[38] X. Xu, P. Hung, and Y. Ye, *A simplified homogeneous and self-dual linear programming algorithm and its implementation*, Annals of Operations Research, 62 (1996), pp. 151–171.
[39] Y. Ye, M.J. Todd, and S. Mizuno, *An $O(\sqrt{n}L)$ -iteration homogeneous and self-dual linear programming algorithm*, Mathematics of Operations Research, 19 (1994), pp. 53–67.
[40] S. Zhang, *A new self-dual embedding method for convex programming*, Journal of Global Optimization, 29 (2004), pp. 479–496.
[41] J. Zhu, *A path following alrogirhm for a class of convex programming problems*, Method and Models of Operations Research, 36 (1992), pp. 359–377.

Appendix. A.

Problem	Before Presolving Rows	Cols	After Presolving Rows	Cols	Primal Objective	Dual Objective	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
aug2d	10000	20200	9604	19800	1.68741175E+06	1.68741175E+06	0.03	3.9	12	1.33	4.39
aug2dc	10000	20200	10000	20200	1.81836807E+06	1.81836807E+06	0.02	4.07	12	1.33	4.48
aug2dcqp	10000	20200	10000	20200	6.49813474E+06	6.49813473E+06	0.02	2.4	13	2	6.81
aug2dqp	10000	20200	10000	20200	6.23701204E+06	6.23701200E+06	0.01	2.64	14	2	7.5
aug3d	1000	3873	512	2673	5.54067726E+02	5.54067726E+02	0.01	0.27	8	1.5	3.89
aug3dc	1000	3873	1000	3873	7.71262439E+02	7.71262439E+02	0.01	0.38	8	1.5	4.38
aug3dcqp	1000	3873	1000	3873	9.93362148E+02	9.93362145E+02	0.01	0.36	11	2	4.84
aug3dqp	1000	3873	1000	3873	6.75237672E+02	6.75237670E+02	0.01	0.39	13	2	5.59
cont-050	2401	2597	2401	2597	-4.56385090E+00	-4.56385090E+00	0.01	0.65	11	2	3.49
cont-100	9801	10197	9801	10197	-4.64439787E+00	-4.64439787E+00	0.01	4.55	11	2	2.11
cont-101	10098	10197	10098	10197	1.95527325E-01	1.95527325E-01	0.01	4.06	9	2	1.95
cont-200	39601	40397	39601	40397	-4.68487592E+00	-4.68487592E+00	0.04	43.34	11	2	0.99
cont-201	40198	40397	40198	40397	1.92483373E-01	1.92483373E-01	0.04	42.09	10	1.9	1
cont-300	90298	90597	90298	90597	1.91512286E-01	1.91512286E-01	0.08	247.59	13	1.92	0.47
cvxqp1_m	500	1000	500	1000	1.08751157E+06	1.08751157E+06	0	0.41	10	2	2.25
cvxqp1_s	50	100	50	100	1.15907181E+04	1.15907181E+04	0	0.08	8	2	1.32
cvxqp2_m	250	1000	250	1000	8.20155431E+05	8.20155431E+05	0	0.23	10	2	2.76
cvxqp2_s	25	100	25	100	8.12094048E+03	8.12094048E+03	0	0.1	10	1.9	1.08
cvxqp3_m	750	1000	750	1000	1.36282874E+06	1.36282874E+06	0	0.65	12	2	1.41
cvxqp3_s	75	100	75	100	1.19434322E+04	1.19434322E+04	0	0.09	10	2	0
dpklo1	77	133	77	133	3.70096217E-01	3.70096217E-01	0	0.04	3	2	0
dtoc3	9998	14999	9997	14996	2.35262481E+02	2.35262481E+02	0.02	0.93	4	2	8.75
dual1	1	85	1	85	3.50129664E-02	3.50129651E-02	0	0.1	12	2	2.06
dual2	1	96	1	96	3.37336761E-02	3.37336761E-02	0	0.08	9	2	0
dual3	1	111	1	111	1.35755838E-01	1.35755833E-01	0	0.09	9	2	1.25
dual4	1	75	1	75	7.46090842E-01	7.46090840E-01	0	0.08	10	2	1.32
dualc1	215	223	13	21	6.15525083E+03	6.15525083E+03	0	0.26	25	1.48	0
dualc2	229	235	9	15	3.55130769E+03	3.55130769E+03	0	0.21	20	1.55	0
dualc5	278	285	1	8	4.27232327E+02	4.27232327E+02	0.01	0.12	13	1.62	0
dualc8	503	510	15	22	1.83093588E+04	1.83093588E+04	0.01	0.2	19	1.63	0.5
genhs28	8	10	8	10	9.27173692E-01	9.27173694E-01	0.01	0.03	3	2	0
gouldqp2	349	699	349	699	1.84275300E-04	1.84273273E-04	0	0.16	13	2	3.25
gouldqp3	349	699	349	699	2.06278397E+00	2.06278397E+00	0.01	0.14	11	2	4.58
hs118	17	32	17	32	6.64820450E+02	6.64820450E+02	0.01	0.09	10	2	0
hs21	1	3	1	3	-9.99599996E+01	-9.99600001E+01	0.01	0.08	9	2	0
hs268	5	10	5	10	2.58580312E-05	-5.06859105E-05	0	0.1	12	2	0
hs35	1	4	1	4	1.11111111E-01	1.11111111E-01	0	0.05	5	2	0
hs35mod	1	4	1	3	2.50000007E-01	2.49999989E-01	0	0.09	11	2	0
hs51	3	5	3	5	0.00000000E+00	9.97371075E-11	0.01	0.03	3	2	0
hs52	3	5	3	5	5.32664756E+00	5.32664757E+00	0	0.04	3	2	0
hs53	3	5	3	5	4.09302326E+00	4.09302302E+00	0.01	0.04	4	2	0
hs76	3	7	3	7	-4.68181818E+00	-4.68181818E+00	0	0.05	5	2	0
ksip	1001	1021	1001	1021	5.75797941E-01	5.75797941E-01	0.01	0.38	16	2	3.45

Problem	Before Presolving Rows	Cols	After Presolving Rows	Cols	Primal Objective	Dual Objective	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
laser	1000	2002	1000	2002	2.40960135E+06	2.40960134E+06	0.01	0.21	10	2	4.04
liswet1	10000	20002	10000	20002	3.61224021E+01	3.61224021E+01	0.03	3.21	19	1.89	7.53
liswet2	10000	20002	10000	20002	2.49980788E+01	2.49980759E+01	0.02	2.68	17	2	7.32
liswet3	10000	20002	10000	20002	2.50012273E+01	2.50012181E+01	0.02	2.55	16	2	7.39
liswet4	10000	20002	10000	20002	2.50001164E+01	2.50001107E+01	0.02	2.76	18	2	8.06
liswet5	10000	20002	10000	20002	2.50343102E+01	2.50341783E+01	0.02	2.54	16	2	7.91
liswet6	10000	20002	10000	20002	2.49957529E+01	2.49957445E+01	0.02	3	18	2	7.82
liswet7	10000	20002	10000	20002	4.98840891E+02	4.98840891E+02	0.02	3.71	19	1.89	7.31
liswet8	10000	20002	10000	20002	7.14470067E+02	7.14470059E+02	0.02	5.47	37	1.78	7.76
liswet9	10000	20002	10000	20002	1.96325126E+03	1.96325126E+03	0.02	5.12	31	1.65	7.18
liswet10	10000	20002	10000	20002	4.94858035E+01	4.94857840E+01	0.02	3.18	21	1.95	7.6
liswet11	10000	20002	10000	20002	4.95239659E+01	4.95239569E+01	0.02	3.23	22	1.95	7.82
liswet12	10000	20002	10000	20002	1.73692743E+03	1.73692743E+03	0.02	4.89	32	1.75	7.6
lotschd	7	12	7	12	2.39841590E+03	2.39841589E+03	0	0.05	9	2	0
mosarqp1	700	3200	700	3200	-9.52875441E+02	-9.52875445E+02	0	0.21	9	2	5.5
mosarqp2	600	1500	600	1500	-1.59748211E+03	-1.59748212E+03	0	0.14	9	2	7.35
powell20	10000	20000	10000	20000	5.20895828E+10	5.20895828E+10	0.02	1.46	8	1.88	7.39
primal1	85	410	85	410	-3.50129646E-02	-3.50129670E-02	0	0.11	9	2	2.97
primal2	96	745	96	745	-3.37336714E-02	-3.37336763E-02	0	0.1	7	2	1.04
primal3	111	856	111	856	-1.35755836E-01	-1.35755837E-01	0	0.21	9	2	4.71
primal4	75	1564	75	1564	-7.46090837E-01	-7.46090843E-01	0	0.2	9	2	5.08
primalc1	9	239	9	239	-6.15525083E+03	-6.15525083E+03	0	0.14	18	1.83	1.49
primalc2	7	238	7	237	-3.55130769E+03	-3.55130769E+03	0	0.2	25	1.68	3.96
primalc5	8	295	8	206	-4.27232327E+02	-4.27232327E+02	0	0.07	10	1.9	1.43
primalc8	8	528	8	528	-1.83094298E+04	-1.83094298E+04	0	0.15	12	1.83	3.36
q25fv47	820	1876	790	1846	1.37444479E+07	1.37444478E+07	0.02	1.55	27	2	4.48
qadlittle	56	138	55	137	4.80318859E+05	4.80318859E+05	0	0.07	11	2	0
qafiro	27	51	27	51	-1.59078179E+00	-1.59078179E+00	0	0.08	14	1.86	0
qbandm	305	472	252	419	1.63523420E+04	1.63523420E+04	0	0.17	20	2	0.6
qbeaconf	173	295	107	219	1.64712060E+05	1.64712060E+05	0	0.1	14	2	1.96
qbore3d	233	334	123	225	3.10020080E+03	3.10020080E+03	0	0.17	19	1.79	2.45
qbrandy	220	303	136	246	2.83751149E+04	2.83751149E+04	0	0.13	17	2	0
qcapri	271	482	243	438	6.67932932E+07	6.67932932E+07	0	0.37	36	1.72	2.72
qe226	223	472	214	463	2.12653433E+02	2.12653433E+02	0	0.17	16	2	2.48
qetamacr	400	816	334	669	8.67603696E+04	8.67603693E+04	0	0.49	32	2	2.29
qfffff80	524	1028	322	826	8.73147464E+05	8.73147456E+05	0	0.39	25	1.92	3.37
qforplan	161	492	122	450	7.45663147E+09	7.45663144E+09	0	0.39	31	1.71	2.86
qgfrdxpn	616	1160	590	1134	1.00790585E+11	1.00790585E+11	0	0.39	30	1.9	3.39
qgrow15	300	645	300	645	-1.01693640E+08	-1.01693640E+08	0	0.28	22	2	2.21
qgrow22	440	946	440	946	-1.49628953E+08	-1.49628953E+08	0	0.43	26	2	3.77
qgrow7	140	301	140	301	-4.27987139E+07	-4.27987139E+07	0	0.18	21	2	3.91
qisrael	174	316	174	316	2.53478378E+07	2.53478376E+07	0	0.28	28	1.75	1.83
qpcblend	74	114	74	114	-7.84254164E-03	-7.84254440E-03	0	0.11	17	2	1.94

Problem	Before Presolving Rows	Before Presolving Cols	After Presolving Rows	After Presolving Cols	Primal Objective	Dual Objective	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
qpcboei1	351	726	343	718	1.15039140E+07	1.15039140E+07	0	0.27	22	2	2.26
qpcboei2	166	305	126	265	8.17196226E+06	8.17196221E+06	0	0.22	26	1.88	1.83
qpcstair	356	614	356	532	6.20438749E+06	6.20438744E+06	0	0.32	25	1.76	1.92
qpilotno	975	2446	866	2151	4.72858690E+06	4.72858690E+06	0.01	1.84	42	1.86	2.8
qrecipe	91	204	76	141	-2.66615999E+02	-2.66616001E+02	0	0.12	19	2	0.88
qptest	2	4	2	4	4.37187500E+00	4.37187500E+00	0	0.03	6	2	0
qsc205	205	317	203	315	-5.81395349E-03	-5.81395349E-03	0	0.12	17	2	1.69
qscagr25	471	671	469	669	2.01737938E+08	2.01737938E+08	0	0.22	22	1.95	4.13
qscagr7	129	185	127	183	2.68659486E+07	2.68659485E+07	0	0.14	22	1.82	2.13
qscfxm1	330	600	311	581	1.68826916E+07	1.68826916E+07	0	0.29	25	2	2.79
qscfxm2	660	1200	622	1162	2.77761616E+07	2.77761615E+07	0	0.49	33	2	3.56
qscfxm3	990	1800	933	1743	3.08163545E+07	3.08163545E+07	0	0.73	35	1.94	3.37
qscorpio	388	466	355	436	1.88050955E+03	1.88050955E+03	0	0.15	14	1.93	2.8
qscrs8	490	1275	445	1230	9.04560015E+02	9.04560013E+02	0.01	0.41	24	1.96	2.7
qscsd1	77	760	77	760	8.66666669E+00	8.66666667E+00	0	0.15	9	2	2.05
qscsd6	147	1350	147	1350	5.08082139E+01	5.08082139E+01	0	0.29	14	2	2.2
qscsd8	397	2750	397	2750	9.40763578E+02	9.40763570E+02	0.01	0.38	11	2	3.76
qsctap1	300	660	284	644	1.41586111E+03	1.41586111E+03	0.01	0.33	17	2	1.88
qsctap2	1090	2500	1033	2443	1.73502651E+03	1.73502649E+03	0.01	0.51	13	2	2.45
qsctap3	1480	3340	1408	3268	1.43875468E+03	1.43875468E+03	0.01	0.64	15	2	3.95
qseba	515	1036	514	1033	8.14818004E+07	8.14818004E+07	0.01	0.64	26	1.92	1.75
qshare1b	117	253	112	248	7.20078318E+05	7.20078318E+05	0	0.47	29	1.76	1.96
qshare2b	96	162	96	162	1.17036917E+04	1.17036917E+04	0	0.25	17	2	0.41
qshell	536	1777	487	1475	1.57263685E+12	1.57263684E+12	0.01	0.98	38	1.79	1.76
qship04l	402	2166	323	2104	2.42001553E+06	2.42001553E+06	0.01	0.38	14	2	2.93
qship04s	402	1506	235	1356	2.42499367E+06	2.42499367E+06	0.01	0.31	14	2	4.26
qship08l	778	4363	638	4247	2.37604062E+06	2.37604062E+06	0.01	1.12	14	2	4.74
qship08s	778	2467	366	2079	2.38572885E+06	2.38572885E+06	0.01	0.41	14	1.93	3.31
qship12l	1151	5533	784	5226	3.01887658E+06	3.01887658E+06	0.02	2.06	16	2	4.04
qship12s	1151	2869	412	2190	3.05696225E+06	3.05696225E+06	0.01	0.73	17	2	3.13
qsierra	1227	2735	1212	2705	2.37504582E+07	2.37504582E+07	0	0.54	20	2	5.53
qstair	356	614	356	532	7.98545276E+06	7.98545275E+06	0	0.36	23	1.87	2.29
qstandat	359	1274	346	855	6.41183839E+03	6.41183839E+03	0	0.21	13	2	1.96
s268	5	10	5	10	2.58580312E-05	-5.06859105E-05	0	0.06	12	2	0
stadat1	3999	6000	3999	6000	-2.85268639E+07	-2.85268640E+07	0.01	1.03	12	2	4.5
stadat2	3999	6000	3999	6000	-3.26266649E+01	-3.26266649E+01	0.01	1.32	18	2	5.61
stadat3	7999	12000	7999	12000	-3.57794529E+01	-3.57794530E+01	0.01	2.5	19	2	7.01
tame	1	2	1	2	0.00000000E+00	-4.19289224E-10	0	0.04	3	2	0
ubh1	12000	18009	12000	17997	1.11600082E+00	1.11600082E+00	0.03	1.19	3	2	10.74
yao	2000	4002	2000	4000	1.97704256E+02	1.97704256E+02	0.01	1.11	23	1.57	4.52
zecevic2	2	4	2	4	-4.12499999E+00	-4.12500002E+00	0	0.04	5	2	0
Avg							0.01	3.44	15.92	1.92	3.03

Table A.1: `iOptimize` performance on Maros and Meszaros's QP test problems

Problem	Before Presolving Rows	Cols	After Presolving Rows	Cols	Primal Objective	Dual Objective	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
QQ-aug2d	10001	20201	10001	20201	1.68741182E+06	1.68741175E+06	0.01	3.75	8	1.75	4.14
QQ-aug2dc	10001	20201	10001	20201	1.81836815E+06	1.81836807E+06	0.02	3.47	8	1.75	4.7
QQ-aug2dcqp	10001	20201	10001	20201	6.49813474E+06	6.49813474E+06	0.75	6.79	19	1.68	5.16
QQ-aug2dqp	10001	20201	10001	20201	6.23701203E+06	6.23701202E+06	0.73	6.7	21	1.71	5.38
QQ-aug3d	1001	3874	1001	3874	5.54067753E+02	5.54067816E+02	0.01	0.56	5	2	2.03
QQ-aug3dc	1001	3874	1001	3874	7.71262488E+02	7.71262604E+02	0.01	0.21	5	2	6.38
QQ-aug3dcqp	1001	3874	1001	3874	9.93362146E+02	9.93362144E+02	0.04	0.44	11	2	5.74
QQ-aug3dqp	1001	3874	1001	3874	6.75237669E+02	6.75237664E+02	0.03	0.43	12	2	6.31
QQ-cont-050	2402	2598	2402	2598	-4.56385143E+00	-4.56385137E+00	0.01	0.75	11	2	3.51
QQ-cont-100	9802	10198	9802	10198	-4.64439991E+00	-4.64439983E+00	0.01	5.59	12	2	2.2
QQ-cont-101	10099	10198	10099	10198	1.95527204E-01	1.95527225E-01	0.01	5.21	11	2	1.97
QQ-cont-200	39602	40398	39602	40398	-4.68487611E+00	-4.68487609E+00	0.05	49.43	13	2	1.32
QQ-cont-201	40199	40398	40199	40398	1.92483103E-01	1.92483155E-01	0.04	57.03	12	2	0.84
QQ-cont-300	90299	90598	90299	90598	1.91512063E-01	1.91512092E-01	0.1	261.3	12	2	0.43
QQ-cvxqp1_m	501	1001	501	1001	1.08751109E+06	1.08751098E+06	0.01	0.38	10	2	2.16
QQ-cvxqp1_s	51	101	51	101	1.15907177E+04	1.15907177E+04	0	0.05	8	2	4.17
QQ-cvxqp2_m	251	1001	251	1001	8.20155390E+05	8.20155375E+05	0	0.22	10	2	2.86
QQ-cvxqp2_s	26	101	26	101	8.12093991E+03	8.12093970E+03	0	0.05	9	2	0
QQ-cvxqp3_m	751	1001	751	1001	1.36282857E+06	1.36282854E+06	0.01	0.78	13	2	1.57
QQ-cvxqp3_s	76	101	76	101	1.19434322E+04	1.19434317E+04	0	0.06	9	2	0
QQ-dpklo1	78	134	78	134	3.70096112E-01	3.70096109E-01	0	0.04	5	2	0
QQ-dtoc3	9999	15000	9999	14998	2.35262505E+02	2.35262481E+02	0.31	2.22	7	1.71	5.97
QQ-dual1	2	86	2	86	3.50125784E-02	3.50123964E-02	0	0.07	13	2	1.47
QQ-dual2	2	97	2	97	3.37335240E-02	3.37334725E-02	0	0.07	12	2	1.45
QQ-dual3	2	112	2	112	1.35755782E-01	1.35755743E-01	0	0.08	12	2	1.35
QQ-dual4	2	76	2	76	7.46090320E-01	7.46089988E-01	0	0.06	10	2	1.92
QQ-dualc1	216	224	216	224	6.15524425E+03	6.15524190E+03	0	0.14	21	2	3.62
QQ-dualc2	230	236	230	236	3.55130765E+03	3.55130761E+03	0	0.12	17	2	2.68
QQ-dualc5	279	286	279	286	4.27232103E+02	4.27231914E+02	0	0.07	9	2	4.76
QQ-dualc8	504	511	504	511	1.83093532E+04	1.83093473E+04	0	0.13	13	2	6.56
QQ-genhs28	9	11	9	11	9.27173702E-01	9.27173694E-01	0	0.03	5	2	0
QQ-gouldqp2	350	700	350	700	1.84274493E-04	1.84273733E-04	0	0.15	15	2	4.76
QQ-gouldqp3	350	700	350	700	2.06278287E+00	2.06278325E+00	0	0.12	10	2	4.24
QQ-hs118	18	33	18	33	6.64820445E+02	6.64820439E+02	0	0.05	11	2	0
QQ-hs21	2	4	2	4	-9.99600032E+01	-9.99600019E+01	0	0.05	10	2	0
QQ-hs268	6	11	6	11	-2.12146135E-07	-7.69541657E-07	0	0.07	15	2	0
QQ-hs35	2	5	2	5	1.11111110E-01	1.11111107E-01	0	0.03	6	2	0
QQ-hs35mod	2	5	2	5	2.49999954E-01	2.49999859E-01	0	0.06	12	2	0
QQ-hs51	4	6	4	6	4.24138339E-08	3.60237236E-08	0	0.02	5	2	4.76
QQ-hs52	4	6	4	6	5.32664757E+00	5.32664757E+00	0	0.02	5	2	0
QQ-hs53	4	6	4	6	4.09302193E+00	4.09302286E+00	0	0.03	6	2	0
QQ-hs76	4	8	4	8	-4.68181818E+00	-4.68181818E+00	0	0.03	6	2	0
QQ-ksip	1002	1022	1002	1022	5.75797941E-01	5.75797940E-01	0.01	0.29	13	2	3.42

Problem	Before Presolving Rows	Before Presolving Cols	After Presolving Rows	After Presolving Cols	Primal Objective	Dual Objective	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
QQ-laser	1001	2003	1001	2003	2.40960146E+06	2.40960145E+06	0	0.6	19	1.53	4.79
QQ-liswet1	10001	20003	10001	20003	3.61224031E+01	3.61224021E+01	0.02	4.87	21	1.95	5.63
QQ-liswet2	10001	20003	10001	20003	2.49980889E+01	2.49980663E+01	0.02	4.23	17	2	5.29
QQ-liswet3	10001	20003	10001	20003	2.50012248E+01	2.50012132E+01	0.02	3.94	17	2	6.13
QQ-liswet4	10001	20003	10001	20003	2.50001177E+01	2.50001054E+01	0.02	4.26	18	2	5.67
QQ-liswet5	10001	20003	10001	20003	2.50342522E+01	2.50342269E+01	0.02	4.44	20	2	5.95
QQ-liswet6	10001	20003	10001	20003	2.49957486E+01	2.49957422E+01	0.02	4.22	19	2	6.16
QQ-liswet7	10001	20003	10001	20003	4.98841004E+02	4.98840891E+02	0.02	5.85	22	1.73	5.4
QQ-liswet8	10001	20003	10001	20003	7.14470246E+02	7.14470012E+02	0.02	7.68	30	1.7	5.48
QQ-liswet9	10001	20003	10001	20003	1.96325241E+03	1.96325122E+03	0.02	10.43	36	1.67	5
QQ-liswet10	10001	20003	10001	20003	4.94858015E+01	4.94857829E+01	0.02	6.31	29	2	5.84
QQ-liswet11	10001	20003	10001	20003	4.95239698E+01	4.95239560E+01	0.02	5.3	23	1.96	6.63
QQ-liswet12	10001	20003	10001	20003	1.73692791E+03	1.73692742E+03	0.02	9.62	38	1.68	5.43
QQ-lotschd	8	13	8	13	2.39841591E+03	2.39841578E+03	0	0.03	9	2	0
QQ-mosarqp1	701	3201	701	3201	-9.52875445E+02	-9.52875470E+02	0	0.27	10	2	6.3
QQ-mosarqp2	601	1501	601	1501	-1.59748212E+03	-1.59748217E+03	0	0.15	9	2	5.56
QQ-powell20	10001	20001	10001	20001	5.20895810E+10	5.20895828E+10	0.02	7.8	19	1.37	3.72
QQ-primal1	86	411	86	411	-3.50129655E-02	-3.50129677E-02	0	0.1	10	2	1.04
QQ-primal2	97	746	97	746	-3.37336759E-02	-3.37336761E-02	0	0.13	9	2	3.42
QQ-primal3	112	857	112	857	-1.35755836E-01	-1.35755837E-01	0	0.26	11	2	4.96
QQ-primal4	76	1565	76	1565	-7.46090838E-01	-7.46090843E-01	0	0.22	9	2	4.46
QQ-primalc1	10	240	10	240	-6.15525075E+03	-6.15525083E+03	0	0.11	19	2	0.97
QQ-primalc2	8	239	8	239	-3.55130754E+03	-3.55130761E+03	0	0.09	17	2	0
QQ-primalc5	9	296	9	296	-4.27232136E+02	-4.27232268E+02	0	0.07	10	2	3.13
QQ-primalc8	9	529	9	529	-1.83094193E+04	-1.83094258E+04	0	0.11	12	2	1.83
QQ-q25fv47	821	1877	800	1856	1.37444358E+07	1.37444267E+07	0.02	1.74	28	1.93	3.16
QQ-qadlittle	57	139	56	138	4.80318851E+05	4.80318850E+05	0	0.05	12	2	0
QQ-qafiro	28	52	28	52	-1.59078178E+00	-1.59078188E+00	0	0.04	14	2	4.65
QQ-qbandm	306	473	278	445	1.63523409E+04	1.63523397E+04	0	0.18	21	2	2.27
QQ-qbeaconf	174	296	141	257	1.64711933E+05	1.64711773E+05	0	0.09	13	2	3.53
QQ-qbore3d	234	335	200	303	3.10020162E+03	3.10020194E+03	0	0.18	19	2	3.39
QQ-qbrandy	221	304	141	251	2.83751074E+04	2.83751029E+04	0	0.16	17	2	2.61
QQ-qcapri	272	483	262	457	6.67932279E+07	6.67932452E+07	0.01	0.3	32	1.69	3.36
QQ-qe226	224	473	223	472	2.12653431E+02	2.12653417E+02	0	0.15	17	2	1.39
QQ-qetamacr	401	817	401	814	8.67601160E+04	8.67601476E+04	0	0.59	33	2	3.97
QQ-qfffff80	525	1029	491	995	8.73147438E+05	8.73146951E+05	0	0.49	26	2	4.02
QQ-qforplan	162	493	136	467	7.45661267E+09	7.45660721E+09	0.01	0.27	23	2	3.04
QQ-qgfrdxpn	617	1161	591	1135	1.00790508E+11	1.00790507E+11	0.01	0.3	24	2	4.36
QQ-qgrow15	301	646	301	646	-1.01693609E+08	-1.01693652E+08	0	0.26	21	2	2.76
QQ-qgrow22	441	947	441	947	-1.49628886E+08	-1.49628968E+08	0	0.41	24	2	3.28
QQ-qgrow7	141	302	141	302	-4.27987148E+07	-4.27987170E+07	0	0.15	21	2	4.7
QQ-qisrael	175	317	175	317	2.53478001E+07	2.53477766E+07	0	0.28	29	1.83	2.89
QQ-qpcblend	75	115	75	115	-7.84259484E-03	-7.84280090E-03	0	0.08	16	2	5.41

Problem	Before Presolving Rows	Cols	After Presolving Rows	Cols	Primal Objective	Dual Objective	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
QQ-qpcboei1	352	727	349	724	1.15039007E+07	1.15039019E+07	0.01	0.34	24	1.96	5.18
QQ-qpcboei2	167	306	141	280	8.17195892E+06	8.17195821E+06	0	0.25	29	1.97	4.18
QQ-qpcstair	357	615	357	615	6.20438137E+06	6.20437660E+06	0.01	0.31	21	2	3.99
QQ-qpilotno	976	2447	955	2248	4.72823223E+06	4.72827122E+06	0.04	1.23	30	2	3.07
QQ-qrecipe	92	205	84	149	-2.66616053E+02	-2.66616081E+02	0	0.09	18	2	2.27
QQ-qptest	3	5	3	5	4.37187497E+00	4.37187483E+00	0	0.02	7	2	0
QQ-qsc205	206	318	204	316	-5.81401694E-03	-5.81421806E-03	0	0.1	17	2	2.13
QQ-qscagr25	472	672	472	672	2.01737737E+08	2.01737657E+08	0	0.36	29	1.66	3.13
QQ-qscagr7	130	186	129	185	2.68659112E+07	2.68658869E+07	0	0.09	19	2	3.33
QQ-qscfxm1	331	601	323	593	1.68826874E+07	1.68826848E+07	0	0.35	30	1.77	3.8
QQ-qscfxm2	661	1201	646	1186	2.77761534E+07	2.77761498E+07	0	0.69	36	1.81	4.15
QQ-qscfxm3	991	1801	969	1779	3.08163509E+07	3.08163491E+07	0	0.97	37	1.84	4.3
QQ-qscorpio	389	467	373	456	1.88050938E+03	1.88050937E+03	0	0.16	16	2	2.65
QQ-qscrs8	491	1276	491	1276	9.04559758E+02	9.04559689E+02	0	0.38	24	2	2.96
QQ-qscsd1	78	761	78	761	8.66666691E+00	8.66666672E+00	0	0.12	11	2	3.6
QQ-qscsd6	148	1351	148	1351	5.08082149E+01	5.08082137E+01	0	0.2	15	2	6.91
QQ-qscsd8	398	2751	398	2751	9.40763587E+02	9.40763560E+02	0	0.33	12	2	4.03
QQ-qsctap1	301	661	301	661	1.41586111E+03	1.41586111E+03	0	0.19	20	2	3.17
QQ-qsctap2	1091	2501	1091	2501	1.73502650E+03	1.73502649E+03	0	0.38	13	2	4.79
QQ-qsctap3	1481	3341	1481	3341	1.43875468E+03	1.43875468E+03	0	0.52	14	2	4.59
QQ-qseba	516	1037	516	1037	8.14817931E+07	8.14817942E+07	0.02	0.43	28	1.86	4.47
QQ-qshare1b	118	254	113	249	7.20078322E+05	7.20078214E+05	0	0.15	26	2	1.4
QQ-qshare2b	97	163	97	163	1.17036837E+04	1.17036812E+04	0	0.09	19	2	1.15
QQ-qshell	537	1778	537	1775	1.57263633E+12	1.57263551E+12	0.02	1.01	30	2	4.25
QQ-qship04l	403	2167	357	2163	2.42001501E+06	2.42001490E+06	0	0.32	15	2	3.27
QQ-qship04s	403	1507	269	1415	2.42499308E+06	2.42499280E+06	0	0.22	14	2	2.9
QQ-qship08l	779	4364	695	4346	2.37604021E+06	2.37604015E+06	0.02	1.53	15	2	3.9
QQ-qship08s	779	2468	487	2242	2.38572832E+06	2.38572819E+06	0.01	0.59	15	2	5.05
QQ-qship12l	1152	5534	921	5412	3.01887662E+06	3.01887624E+06	0.03	3.32	19	2	2.88
QQ-qship12s	1152	2870	688	2515	3.05696198E+06	3.05696007E+06	0.01	0.84	19	2	4.73
QQ-qsierra	1228	2736	1228	2721	2.37504456E+07	2.37504474E+07	0	0.64	20	2	5.06
QQ-qstair	357	615	357	546	7.98544934E+06	7.98544810E+06	0.01	0.33	24	1.92	2.51
QQ-qstandat	360	1275	359	1192	6.41183858E+03	6.41183826E+03	0.01	0.29	17	2	4.2
QQ-s268	6	11	6	11	-2.12146135E-07	-7.69541657E-07	0	0.04	15	2	0
QQ-stadat1	4000	6001	4000	6001	-2.85268640E+07	-2.85268640E+07	0.01	1.7	19	1.53	5.02
QQ-stadat2	4000	6001	4000	6001	-3.26266629E+01	-3.26266666E+01	0.01	1.01	18	2	7.01
QQ-stadat3	8000	12001	8000	12001	-3.57794523E+01	-3.57794533E+01	0.01	2.18	18	2	6.65
QQ-tame	2	3	2	3	-3.27685749E-10	-6.53153282E-10	0	0.02	6	2	0
QQ-ubh1	12001	18010	12001	17998	1.11600080E+00	1.11600080E+00	0.47	1.46	5	2	11.09
QQ-yao	2001	4003	2001	4003	1.97704313E+02	1.97704256E+02	0.02	1.01	23	1.87	5.14
QQ-zecevic2	3	5	3	5	-4.12500000E+00	-4.12500003E+00	0	0.02	7	2	0
Avg							0.03	4.08	16.49	1.95	3.37

Table A.2: `iOptimize` performance on Mittelmann's QCQP test problems

Problem	Before Presolving Rows	Cols	After Presolving Rows	Cols	Primal Objective	Dual Objective	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
airport	42	84	42	84	4.79527017E+04	4.79526915E+04	0	0.17	28	1.29	1.86
polak4	3	3	3	3	2.43646932E-08	1.65838221E-07	0	0.04	13	2	0
makela1	2	4	2	4	-1.41421356E+00	-1.41421356E+00	0	0.03	8	2	4.35
makela2	3	3	3	3	7.20000270E+00	7.20000353E+00	0	0.02	7	2	5.26
makela3	20	21	20	21	2.31888348E-11	0.00000000E+00	0	0.03	8	2	0
gigomez1	3	5	3	5	-2.99999998E+00	-2.99999992E+00	0	0.03	9	1.67	0
hanging	180	288	180	288	-6.20176047E+02	-6.20176047E+02	0	0.16	10	2	3.45
madsschj	158	81	158	81	-7.97283702E+02	-7.97283702E+02	0	0.14	11	1.64	2.27
rosenmmx	4	5	4	5	-4.39999956E+01	-4.39999945E+01	0	0.03	8	2	0
optreward	2	8	2	8	-1.09999913E+00	-1.09999913E+00	0	0.03	10	2	0
optprloc	30	35	30	35	-1.64197690E+01	-1.64197857E+01	0	0.06	16	1.81	0
Avg							0.00	0.07	11.64	1.86	1.56

Table A.3: `iOptimize` performance of QCQP on `Cute` test problem problems

Problem	Before Presolving Rows	Before Presolving Cols	After Presolving Rows	After Presolving Cols	Primal Objective	Dual Objective	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
bigbank	814	1773	814	1773	-4.20569612E+06	-4.20569667E+06	0.03	7.15	25	1.6	6.25
gridnete	3844	7565	3844	7565	2.06554692E+02	2.06554690E+02	0.05	3.1	7	2	7.66
gridnetf	3844	7565	3844	7565	2.42108991E+02	2.42108973E+02	0.04	27.23	15	2	8.17
gridneth	36	61	36	61	3.96262686E+01	3.96262685E+01	0.03	0.25	6	2	0.43
gridneti	36	61	36	61	4.02474712E+01	4.02474424E+01	0.03	0.39	8	2	0
dallasl	598	837	598	837	-2.02604132E+05	-2.02610489E+05	0.03	2.4	29	1.55	4.65
dallasm	119	164	119	164	-4.81981888E+04	-4.81982189E+04	0.03	2.08	49	1.92	1.07
dallass‡	29	44	29	44	-3.23932257E+04	-3.23932858E+04	0.03	2.04	29	1.69	0.3
polak1	2	3	2	3	2.71828182E+00	2.71828183E+00	0.03	0.21	6	2	0
polak2	2	11	2	11	5.45981499E+01	5.45981501E+01	0.03	0.21	6	2	0.52
polak3	10	12	10	12	5.93300335E+00	5.93300335E+00	0.03	0.43	9	2	0.24
polak5	2	3	2	3	4.99999999E+01	5.00000001E+01	0.03	0.29	6	2	0
smbank	64	117	64	117	-7.12929200E+06	-7.12929378E+06	0.03	1.85	30	1.37	0.65
cantilvr	1	5	1	5	1.33995635E+00	1.33995641E+00	0.03	0.61	12	2	0
cb2	3	3	3	3	1.95222455E+00	1.95222408E+00	0.03	0.29	7	2	0.36
cb3	3	3	3	3	2.00000001E+00	2.00000004E+00	0.03	0.28	7	2	0
chaconn1	3	3	3	3	1.95222455E+00	1.95222408E+00	0.03	0.3	7	2	0
chaconn2	3	3	3	3	2.00000001E+00	2.00000004E+00	0.03	0.28	7	2	0
dipigri	4	7	4	7	6.80630082E+02	6.80629944E+02	0.03	0.39	9	2	0
gpp	498	499	498	499	1.44009302E+04	1.44009293E+04	0.03	2.28	10	2	7.7
hong	1	4	1	4	1.34730701E+00	1.34730660E+00	0.03	0.66	16	2	0.31
loadbal	31	51	31	51	4.52851044E-01	4.52850995E-01	0.03	0.81	21	2	0.25
svanberg	5000	5000	5000	5000	8.36142276E+03	8.36142275E+03	0.11	114.33	16	2	11.24
antenna-antenna2	166	49	166	49	1.57986028E+00	1.57986028E+00	0.03	7.53	15	2	8.3
antenna-antenna_vareps	166	50	166	50	1.22309383E+00	1.22309383E+00	0.03	12.12	18	1.72	6.95
braess-trafequil	361	722	361	722	5.26883657E+01	5.26883559E+01	0.03	1.25	17	2	1.46
braess-trafequil2	437	798	437	798	5.26883639E+01	5.26883616E+01	0.03	1.86	14	2	7.73
braess-trafequilsf	309	856	309	856	5.15910455E+01	5.15910406E+01	0.02	1.81	17	2	0.76
braess-trafequil2sf	385	932	385	932	5.15910445E+01	5.15910425E+01	0.03	2.33	13	2	7.58
elena-chemeq	12	38	12	38	-1.91087024E+03	-1.91087054E+03	0.01	0.95	19	1.63	0.21
elena-s383	1	14	1	14	7.28593644E+05	7.28593346E+05	0.03	1.28	19	1.42	0.08
firfilter-fir_convex	243	163	243	163	1.04648765E+00	1.04648766E+00	0.03	0.83	14	2	6.99
markowitz-growthopt	1	8	1	8	-1.17084823E-01	-1.17084822E-01	0.04	0.36	9	2	0
wbv-antenna2	1197	1180	1197	1180	1.09359976E+00	1.09359978E+00	0.01	85.46	14	2	9.16
wbv-lowpass2	200	219	200	219	1.05905136E+00	1.05905055E+00	0	8.32	25	2	6.99
batch	73	106	73	106	2.59180347E+05	2.59175929E+05	0.01	0.38	13	2	1.37
stockcycle	97	481	97	481	1.17916213E+05	1.17915981E+05	0	0.36	11	2	1.44
synthes1	6	10	6	10	7.59284403E-01	7.59284359E-01	0	0.19	7	2	0.53
synthes2	14	21	14	21	-5.54416573E-01	-5.54416857E-01	0	0.21	8	2	0
synthes3	23	34	23	34	1.50821846E+01	1.50821817E+01	0.01	0.21	8	2	0
trimloss2	24	53	24	53	7.18306638E-01	7.18306802E-01	0.02	0.26	11	2	0.79
trimloss4	64	145	64	145	1.70933108E+00	1.70933115E+00	0.01	0.28	11	2	2.51
trimloss5	90	216	90	216	1.17887002E+00	1.17887282E+00	0	0.37	13	2	4.02
trimloss6	120	287	120	287	1.30565671E+00	1.30565766E+00	0.01	0.35	13	2	5.19
trimloss7	154	436	154	436	5.93497411E-01	5.93498455E-01	0	0.66	15	2	7.83
trimloss12	384	1028	384	1028	2.31187222E+00	2.31187205E+00	0	6.81	18	2	8.85
Avg							0.02	6.55	14.09	1.94	2.94

Table A.4: `iOptimize` performance on `Cute` CP test problems

Problem	MOSEK	iOptimize	Problem	MOSEK	iOptimize	Problem	MOSEK	iOptimize
aug2d	6	12	laser	12	10	qpcboei1	20	22
aug2dc	6	12	liswet1	26	19	qpcboei2	21	26
aug2dcqp	13	13	liswet2	32	17	qpcstair	22	25
aug2dqp	15	14	liswet3	21	16	qpilotno	35§	42
aug3d	6	8	liswet4	26§	18	qrecipe	16	19
aug3dc	6	8	liswet5	25	16	qptest	6	6
aug3dcqp	11	11	liswet6	24	18	qsc205	16	17
aug3dqp	12	13	liswet7	31	19	qscagr25	20	22
cont-050	11	11	liswet8	38	37	qscagr7	19	22
cont-100	12	11	liswet9	37	31	qscfxm1	25	25
cont-101	13	9	liswet10	36	21	qscfxm2	30	33
cont-200	12	11	liswet11	31	22	qscfxm3	30	35
cont-201	15	10	liswet12	30	32	qscorpio	13	14
cont-300	16	13	lotschd	10	9	qscrs8	21	24
cvxqp1_m	10	10	mosarqp1	9	9	qscsd1	10	9
cvxqp1_s	8	8	mosarqp2	9	9	qscsd6	13	14
cvxqp2_m	10	10	powell20	10§	8	qscsd8	12	11
cvxqp2_s	9	10	primal1	10	9	qsctap1	18	17
cvxqp3_m	12	12	primal2	8	7	qsctap2	13	13
cvxqp3_s	10	10	primal3	9	9	qsctap3	14	15
dpklo1	5	3	primal4	10	9	qseba	16	26
dtoc3	5	4	primalc1	18	18	qshare1b	23	29
dual1	14	12	primalc2	15	25	qshare2b	17	17
dual2	12	9	primalc5	8	10	qshell	33	38
dual3	12	9	primalc8	10	12	qship04l	14	14
dual4	12	10	q25fv47	28	27	qship04s	13	14
dualc1	10	25	qadlittle	11	11	qship08l	13	14
dualc2	10	20	qafiro	12	14	qship08s	13	14
dualc5	7	13	qbandm	19	20	qship12l	17	16
dualc8	7	19	qbeaconf	14	14	qship12s	17	17
genhs28	4	3	qbore3d	13	19	qsierra	18	20
gouldqp2	14	13	qbrandy	14	17	qstair	22	23
gouldqp3	10	11	qcapri	26	36	qstandat	15	13
hs118	10	10	qe226	14	16	s268	22	12
hs21	9	9	qetamacr	27	32	stadat1	25	12
hs268	22	12	qffffff80	22	25	stadat2	41	18
hs35	5	5	qforplan	23	31	stadat3	46	19
hs35mod	11	11	qgfrdxpn	23	30	tame	3	3
hs51	5	3	qgrow15	22	22	ubh1	6	3
hs52	4	3	qgrow22	25	26	yao	23	23
hs53	6	4	qgrow7	23	21	zecevic2	5	5
hs76	5	5	qisrael	21	28			
ksip	21	16	qpcblend	19	17	Avg	16.22	15.92

[1] §: Near optimal solution found.

Table A.5: Comparison with MOSEK on Maros and Meszaros's QP test problems

Problem	MOSEK	iOptimize	Problem	MOSEK	iOptimize	Problem	MOSEK	iOptimize
QQ-aug2d	11	8	QQ-laser	**135**	19	QQ-qpcboei1	24	24
QQ-aug2dc	9	8	QQ-liswet1	41?	21	QQ-qpcboei2	29	29
QQ-aug2dcqp	18	19	QQ-liswet2	21	17	QQ-qpcstair	24	21
QQ-aug2dqp	†	21	QQ-liswet3	17	17	QQ-qpilotno	39	30
QQ-aug3d	6	5	QQ-liswet4	17	18	QQ-qrecipe	16	18
QQ-aug3dc	7	5	QQ-liswet5	15	20	QQ-qptest	7	7
QQ-aug3dcqp	17	11	QQ-liswet6	17	19	QQ-qsc205	17	17
QQ-aug3dqp	22	12	QQ-liswet7	34?	22	QQ-qscagr25	20	29
QQ-cont-050	14	11	QQ-liswet8	49?	30	QQ-qscagr7	18	19
QQ-cont-100	15	12	QQ-liswet9	26?	36	QQ-qscfxm1	43	30
QQ-cont-101	12	11	QQ-liswet10	18?	29	QQ-qscfxm2	39	36
QQ-cont-200	19	13	QQ-liswet11	27?	23	QQ-qscfxm3	26	37
QQ-cont-201	14	12	QQ-liswet12	24?	38	QQ-qscorpio	17	16
QQ-cont-300	15	12	QQ-lotschd	15	9	QQ-qscrs8	24	24
QQ-cvxqp1_m	15	10	QQ-mosarqp1	15	10	QQ-qscsd1	13	11
QQ-cvxqp1_s	15	8	QQ-mosarqp2	14	9	QQ-qscsd6	17	15
QQ-cvxqp2_m	18	10	QQ-powell20	†	19	QQ-qscsd8	16	12
QQ-cvxqp2_s	12	9	QQ-primal1	11	10	QQ-qsctap1	19	20
QQ-cvxqp3_m	18	13	QQ-primal2	9	9	QQ-qsctap2	18	13
QQ-cvxqp3_s	14	9	QQ-primal3	10	11	QQ-qsctap3	18	14
QQ-dpklo1	6	5	QQ-primal4	11	9	QQ-qseba	**62**	28
QQ-dtoc3	11	7	QQ-primalc1	20	19	QQ-qshare1b	27	26
QQ-dual1	14	13	QQ-primalc2	18	17	QQ-qshare2b	19	19
QQ-dual2	13	12	QQ-primalc5	11	10	QQ-qshell	26	30
QQ-dual3	14	12	QQ-primalc8	15	12	QQ-qship04l	21	15
QQ-dual4	13	10	QQ-q25fv47	39	28	QQ-qship04s	18	14
QQ-dualc1	17	21	QQ-qadlittle	19	12	QQ-qship08l	20	15
QQ-dualc2	14	17	QQ-qafiro	16	14	QQ-qship08s	21	15
QQ-dualc5	11	9	QQ-qbandm	21	21	QQ-qship12l	24	19
QQ-dualc8	18	13	QQ-qbeaconf	12	13	QQ-qship12s	24	19
QQ-genhs28	5	5	QQ-qbore3d	14	19	QQ-qsierra	24	20
QQ-gouldqp2	13	15	QQ-qbrandy	17	17	QQ-qstair	31	24
QQ-gouldqp3	13	10	QQ-qcapri	26	32	QQ-qstandat	16	17
QQ-hs118	11	11	QQ-qe226	17	17	QQ-s268	13	15
QQ-hs21	13	10	QQ-qetamacr	24	33	QQ-stadat1	53	19
QQ-hs268	13	15	QQ-qfffff80	30	26	QQ-stadat2	18	18
QQ-hs35	6	6	QQ-qforplan	**180**§	23	QQ-stadat3	18	18
QQ-hs35mod	13	12	QQ-qgfrdxpn	25	24	QQ-tame	5	6
QQ-hs51	5	5	QQ-qgrow15	21	21	QQ-ubh1	20	5
QQ-hs52	6	5	QQ-qgrow22	24	24	QQ-yao	27§	23
QQ-hs53	10	6	QQ-qgrow7	21	21	QQ-zecevic2	6	7
QQ-hs76	8	6	QQ-qisrael	28	29	Avg1	20.91	16.43
QQ-ksip	15	13	QQ-qpcblend	20	16	Avg2	18.34	16.26

[1] †: Max number of iterations reached.
[2] §: Near optimal solution found.
[3] ?: Terminated with unknown status.

Table A.6: Comparison with MOSEK on Mittelmann's QCQP test problems

Problem	MOSEK	iOptimize
airport	26	28
polak4	10	13
makela1	8	8
makela2	7	7
makela3	6	8
gigomez1	8	9
hanging	13	10
madsschj	10	11
rosenmmx	9	8
optreward	10	10
optprloc	18	16
Avg	11.36	11.64

Table A.7: Comparison with MOSEK on Cute QCQP test problems

Problem	MOSEK	iOptimize	Ipopt	Knitro
bigbank	20	21	24	19
gridnete	9	7	5	4
gridnetf	16	13	27	17
gridneth	8	6	6	3
gridneti	10	8	7	6
dallasl	**74**	33	**249**	**245**
dallasm	**128**	31	‡	‡
dallass	†	31	†	**135**
polak1	6	5	4	2
polak2	6	5	4	1
polak3	10	8	‡	‡
polak5	6	5	24	12
smbank	14	27	12	11
cantilvr	12	12	5	3
cb2	8	8	4	3
cb3	7	7	4	4
chaconn1	8	8	4	3
chaconn2	7	7	4	4
dipigri	14	10	6	2
gpp	12	10	22	11
hong	13	15	11	7
loadbal	22	21	13	7
svanberg	14	14	30	16
antenna\antenna2	21§	16	**178**	**58**
antenna\antenna_vareps	19	18	**106**	**61**
braess\trafequil	15	16	18	22
braess\trafequil2	11	13	26	13
braess\trafequilsf	14	15	18	24
braess\trafequil2sf	12	12	25	12
elena\chemeq	23	20	37	16
elena\s383	**85**	21	18	8
firfilter\fir_convex	33§	13	27	11
markowitz\growthopt	8	7	8	7
wbv\antenna2	15	13	31	16
wbv\lowpass2	30§	22	31	**66**
batch	13	12	13	16
stockcycle	11	10	23	12
synthes1	6	6	6	6
synthes2	8	8	10	7
synthes3	8	8	9	8
trimloss2	10	10	13	7
trimloss4	11	12	16	8
trimloss5	13	13	20	10
trimloss6	12	12	23	11
trimloss7	13	13	35	14
trimloss12	19	17	41	22
Avg1	18.46	14.00	–	–
Avg2	13.07	12.69	–	–

[1] †: Max number of iterations reached.
[2] §: Near optimal solution found.
[3] ‡: Function evaluate error from AMPL.

Table A.8: Comparison with MOSEK, Ipopt, and Knitro on CP test problems

Problem	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
iQQ-aug2d	0.01	7.14	16	1.69	4.78
iQQ-aug2dc	0.01	6.8	16	1.69	4.34
iQQ-aug2dcqp	0.78	13.44	29	1.34	4.5
iQQ-aug2dqp	0.73	13.4	29	1.34	4.44
iQQ-aug3d	0	4.72	35	1.09	2.43
iQQ-aug3dc	0	1.06	16	1.5	5.13
iQQ-aug3dcqp	0.04	2.59	33	1.09	4.1
iQQ-aug3dqp	0.03	2.68	37	1.05	4.4
iQQ-cont-050	0	0.76	9	2	4.72
iQQ-cont-100	0.01	6.7	14	1.93	2.17
iQQ-cont-101	0	3.33	9	2	2.35
iQQ-cont-200	0.03	47.74	14	2	1.71
iQQ-cont-201	0.03	26.08	9	2	2.09
iQQ-cont-300	0.05	104.79	9	2	1.57
iQQ-cvxqp1_m	0	2.19	26	1.38	1.38
iQQ-cvxqp1_s	0	2.24	35	1.2	0.45
iQQ-cvxqp2_m	0	2.14	29	1.34	0.9
iQQ-cvxqp2_s	0	2.41	37	1.19	0.92
iQQ-cvxqp3_m	0	2.85	30	1.23	1.8
iQQ-cvxqp3_s	0	1.22	35	1.17	0.98
iQQ-dpklo1	0	0.36	13	2	0.83
iQQ-dtoc3	0.32	4.21	15	1.87	5.9
iQQ-dual1	0	1.14	31	1.32	0.88
iQQ-dual2	0	0.52	16	1.69	0
iQQ-dual3	0	0.63	18	1.67	1.61
iQQ-dual4	0	1.23	33	1.33	0
iQQ-dualc1	0	1.06	35	1.66	0
iQQ-dualc2	0	1.19	35	1.57	0.59
iQQ-dualc5	0	1.65	44	1.32	1.04
iQQ-dualc8	0	1.06	34	1.68	1.9
iQQ-genhs28	0	0.37	13	2	0
iQQ-gouldqp2	0	0.27	10	2	0
iQQ-gouldqp3	0	0.56	16	1.63	1.82
iQQ-hs118	0	0.27	11	2	0
iQQ-hs21	0.02	1.2	35	1.11	0.08
iQQ-hs268	0	1.52	45	1.11	0
iQQ-hs35	0	0.52	16	1.75	0
iQQ-hs35mod	0	0.54	16	1.75	0.19
iQQ-hs51	0	1.07	27	1.19	0
iQQ-hs52	0	0.73	26	1.31	0
iQQ-hs53	0	1.41	39	1.13	0
iQQ-hs76	0	0.36	13	2	0
iQQ-ksip	0	1.73	42	1.05	3.3

Problem	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
iQQ-laser	0	0.4	13	2	5.22
iQQ-liswet1	0.01	16.43	42	1.14	4.19
iQQ-liswet2	0.02	17.23	43	1.12	4.9
iQQ-liswet3	0.01	15	38	1.16	5.15
iQQ-liswet4	0.01	15.72	40	1.13	5.16
iQQ-liswet5	0.01	18.09	47	1.17	4.49
iQQ-liswet6	0.02	16.79	43	1.14	4.64
iQQ-liswet7	0.01	15.81	40	1.13	5.12
iQQ-liswet8	0.01	14.03	33	1.15	4.37
iQQ-liswet9	0.01	15.69	37	1.14	4.69
iQQ-liswet10	0.02	16.5	42	1.14	4.97
iQQ-liswet11	0.01	16.33	42	1.14	4.9
iQQ-liswet12	0.01	15.73	38	1.13	4.62
iQQ-lotschd	0	0.3	12	2	0
iQQ-mosarqp1	0	2.04	33	1.09	4.55
iQQ-mosarqp2	0	1.41	38	1.08	3.57
iQQ-powell20	0.02	6.5	16	1.63	3.5
iQQ-primal1	0.01	1.28	43	1.09	1.41
iQQ-primal2	0	1.31	37	1.14	2.47
iQQ-primal3	0	1.58	41	1.1	2.24
iQQ-primal4	0.01	1.56	36	1.17	4.44
iQQ-primalc1	0	1.37	42	1.26	1.03
iQQ-primalc2	0	1.32	36	1.33	1.52
iQQ-primalc5	0	0.97	23	1.39	0.93
iQQ-primalc8	0	1.1	32	1.28	1.01
iQQ-q25fv47	0.01	1.4	15	2	6.42
iQQ-qadlittle	0	0.34	14	2	0.59
iQQ-qafiro	0	1.37	39	1.15	0.07
iQQ-qbandm	0	0.34	14	1.86	3.02
iQQ-qbeaconf	0	0.41	14	1.93	0.5
iQQ-qbore3d	0	0.51	18	1.89	0
iQQ-qbrandy	0	0.48	16	1.88	0.41
iQQ-qcapri	0	0.62	22	1.91	1.63
iQQ-qe226	0	1.41	36	1.22	1.85
iQQ-qetamacr	0	1.09	34	1.21	3.17
iQQ-qffff80	0	0.57	20	1.8	3.57
iQQ-qforplan	0.01	0.25	11	2	1.63
iQQ-qgfrdxpn	0.01	0.47	19	2	2.42
iQQ-qgrow15	0	0.44	16	1.94	1.15
iQQ-qgrow22	0.01	0.54	18	1.89	0
iQQ-qgrow7	0	0.56	18	1.89	0.54
iQQ-qisrael	0	0.47	19	2	1.3
iQQ-qpcblend	0	1.5	38	1.16	0.27

Problem	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
iQQ-qpcboei1	0.01	0.31	12	2	6.06
iQQ-qpcboei2	0	0.44	17	1.94	4.65
iQQ-qpcstair	0.01	0.64	20	1.75	1.27
iQQ-qpilotno	0.04	0.65	16	2	3.84
iQQ-qrecipe	0	0.75	20	1.6	0.13
iQQ-qptest	0	0.56	16	1.81	0.18
iQQ-qsc205	0	1.51	41	1.24	1.35
iQQ-qscagr25	0	0.32	14	2	0.32
iQQ-qscagr7	0	0.33	14	2	1.22
iQQ-qscfxm1	0	0.4	15	1.93	1.79
iQQ-qscfxm2	0	0.42	16	1.94	2.44
iQQ-qscfxm3	0	0.54	16	1.94	4.71
iQQ-qscorpio	0	0.95	25	1.24	1.91
iQQ-qscrs8	0	0.63	22	1.73	6.45
iQQ-qscsd1	0	0.84	20	1.6	1.31
iQQ-qscsd6	0	0.89	22	1.64	2.07
iQQ-qscsd8	0	2.02	37	1.11	5.11
iQQ-qsctap1	0	1.07	36	1.19	1.14
iQQ-qsctap2	0	2.26	40	1.13	3.58
iQQ-qsctap3	0	2.26	32	1.22	3.65
iQQ-qseba	0.02	0.57	20	1.85	5.36
iQQ-qshare1b	0	0.44	17	1.94	2.33
iQQ-qshare2b	0	0.47	13	1.69	2.13
iQQ-qshell	0.02	0.82	17	2	6.4
iQQ-qship04l	0	0.31	11	2	0
iQQ-qship04s	0.01	0.25	11	2	6.88
iQQ-qship08l	0.01	1.63	16	1.94	4.56
iQQ-qship08s	0	0.72	17	1.88	3.73
iQQ-qship12l	0.01	2.96	17	1.71	4.74
iQQ-qship12s	0.01	0.99	18	1.83	2.98
iQQ-qsierra	0	0.64	17	2	5
iQQ-qstair	0	0.39	17	2	1.05
iQQ-qstandat	0.01	1.37	39	1.18	2.05
iQQ-s268	0	1.56	45	1.11	0.06
iQQ-stadat1	0.01	3.13	26	1.23	4.19
iQQ-stadat2	0	4.39	35	1.14	4.42
iQQ-stadat3	0.01	6.18	25	1.2	4.25
iQQ-tame	0	0.23	10	2	0
iQQ-ubh1	0.46	5.73	26	1.96	6.21
iQQ-yao	0.01	3.17	38	1.05	4.85
iQQ-zecevic2	0	0.23	9	2	0
Avg	0.02	4.40	25.30	1.56	2.48

Table A.9: `iOptimize` performance on Maros and Meszaros's QCQP infeasible test problems

Problem	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
iairport	0.01	0.6	19	1.63	0.67
ipolak4	0	0.81	25	1.28	0.12
imakela1	0	1.11	32	1.13	0
imakela2	0	1.13	32	1.13	0
imakela3	0	0.32	14	2	0
igigomez1	0	1.02	26	1.19	0
ihanging	0	1.17	30	1.13	4.27
imadsschj	0.06	0.53	17	1.59	0.76
irosenmmx	0	1	35	1.14	0
ioptreward	0	0.64	20	1.55	0
ioptprloc	0	0.92	21	1.52	0.11
Avg	0.01	0.84	24.64	1.39	0.54

Table A.10: `iOptimize` performance on `Cute` QCQP infeasible test problems

Problem	Presolve Time	Solver Time	Avg Iters	Avg Dirs	Prop Φ (%)
ibigbank	0	17.12	27	1.33	6.06
igridnete	0	81.51	14	1.71	6.79
igridnetf	0	377.58	36	1.08	5.67
igridneth	0.02	0.78	23	1.39	0.26
igridneti	0	1.23	40	1.07	1.15
idallasl‡					
idallasm	0.01	1.96	47	1.4	2.26
idallass	0	1.58	45	1.22	0.51
ipolak1	0	0.41	16	1.81	0
ipolak2	0	0.32	13	2	0
ipolak3	0	0.77	27	1.3	0.52
ipolak5	0	0.41	13	2	0.25
ismbank	0	1.13	31	1.13	1.7
icantilvr	0	0.29	12	2	0.35
icb2	0	0.94	32	1.19	0
icb3	0.01	1.25	31	1.16	0.24
ichaconn1	0	0.95	32	1.19	0
ichaconn2	0	1.26	31	1.16	0.08
idipigri	0	1.29	32	1.19	0.08
igpp	0	34.24	38	1.11	5.34
ihong	0	0.92	22	1.45	0.11
iloadbal	0	0.37	14	2	0.27
isvanberg	0.08	632.99	27	1.22	6.65
antenna-iantenna2	0	68.79	186	1.65	13.31
antenna-iantenna_vareps	0	48.02	47	1.13	5.83
braess-itrafequil	0	4.77	38	1.37	3.1
braess-itrafequil2	0	1.87	9	2	7.34
braess-itrafequilsf	0	9.23	46	1.26	2.71
braess-itrafequil2sf	0	2.56	9	2	7.16
elena-ichemeq	0	0.06	14	2	4.92
elena-is383	0	0.06	14	1.93	1.64
firfilter-ifir_convex	0	0.75	13	1.92	6.03
markowitz-igrowthopt	0	0.08	16	1.69	0
wbv-iantenna2	0.06	119.88	17	2	7.97
wbv-ilowpass2	0	4.54	13	1.92	7.35
ibatch	0	0.15	22	1.36	3.47
istockcycle	0	0.88	35	1.14	6.29
isynthes1	0	0.09	25	1.36	0
isynthes2	0	0.09	23	1.35	1.15
isynthes3	0	0.18	40	1.1	3.91
itrimloss2	0	0.1	15	1.93	0
itrimloss4	0	0.15	16	1.94	7.43
itrimloss5	0	0.22	15	2	6.07
itrimloss6	0	0.37	15	2	7.32
itrimloss7	0	0.84	16	2	6.58
itrimloss12	0.01	7.79	16	2	7.41
Avg	0.00	31.79	28.07	1.56	3.45

[1] ‡: Function evaluate error from `AMPL`.

Table A.11: `iOptimize` performance on `Cute` CP infeasible test problems

Problem	MOSEK	iOptimize	Problem	MOSEK	iOptimize	Problem	MOSEK	iOptimize
iQQ-aug2d	†	16	iQQ-laser	29	13	iQQ-qpcboei1	†	12
iQQ-aug2dc	†	16	iQQ-liswet1	31	42	iQQ-qpcboei2	†	17
iQQ-aug2dcqp	†	29	iQQ-liswet2	30	43	iQQ-qpcstair	†	20
iQQ-aug2dqp	24	29	iQQ-liswet3	30	38	iQQ-qpilotno	†	16
iQQ-aug3d	21	35	iQQ-liswet4	30	40	iQQ-qrecipe	12	20
iQQ-aug3dc	16	16	iQQ-liswet5	21	47	iQQ-qptest	15	16
iQQ-aug3dcqp	**121**	33	iQQ-liswet6	30	43	iQQ-qsc205	16	41
iQQ-aug3dqp	**95**	37	iQQ-liswet7	24	40	iQQ-qscagr25	383?	14
iQQ-cont-050	†	9	iQQ-liswet8	26	33	iQQ-qscagr7	38	14
iQQ-cont-100	†	14	iQQ-liswet9	24	37	iQQ-qscfxm1	†	15
iQQ-cont-101	†	9	iQQ-liswet10	30	42	iQQ-qscfxm2	†	16
iQQ-cont-200	†	14	iQQ-liswet11	30	42	iQQ-qscfxm3	†	16
iQQ-cont-201	**70**	9	iQQ-liswet12	24	38	iQQ-qscorpio	31	25
iQQ-cont-300	†	9	iQQ-lotschd	26	12	iQQ-qscrs8	52	22
iQQ-cvxqp1_m	†	26	iQQ-mosarqp1	**100**	33	iQQ-qscsd1	28	20
iQQ-cvxqp1_s	21	35	iQQ-mosarqp2	**66**	38	iQQ-qscsd6	25	22
iQQ-cvxqp2_m	**97**	29	iQQ-powell20	†	16	iQQ-qscsd8	30	37
iQQ-cvxqp2_s	18	37	iQQ-primal1	31	43	iQQ-qsctap1	38	36
iQQ-cvxqp3_m	†	30	iQQ-primal2	32	37	iQQ-qsctap2	†	40
iQQ-cvxqp3_s	22	35	iQQ-primal3	33	41	iQQ-qsctap3	**87**	32
iQQ-dpklo1	†	13	iQQ-primal4	34	36	iQQ-qseba	28	20
iQQ-dtoc3	24	15	iQQ-primalc1	36	42	iQQ-qshare1b	51	17
iQQ-dual1	10	31	iQQ-primalc2	19	36	iQQ-qshare2b	**175**	13
iQQ-dual2	9	16	iQQ-primalc5	22	23	iQQ-qshell	20	17
iQQ-dual3	9	18	iQQ-primalc8	24	32	iQQ-qship04l	15	11
iQQ-dual4	9	33	iQQ-q25fv47	†	15	iQQ-qship04s	15	11
iQQ-dualc1	18	35	iQQ-qadlittle	†	14	iQQ-qship08l	†	16
iQQ-dualc2	15	35	iQQ-qafiro	**70**	39	iQQ-qship08s	†	17
iQQ-dualc5	10	44	iQQ-qbandm	37	14	iQQ-qship12l	†	17
iQQ-dualc8	30	34	iQQ-qbeaconf	26	14	iQQ-qship12s	†	18
iQQ-genhs28	†	13	iQQ-qbore3d	10	18	iQQ-qsierra	†	17
iQQ-gouldqp2	10	10	iQQ-qbrandy	†	16	iQQ-qstair	353?	17
iQQ-gouldqp3	14	16	iQQ-qcapri	†	22	iQQ-qstandat	**340**	39
iQQ-hs118	**232**	11	iQQ-qe226	**281**	36	iQQ-s268	26	45
iQQ-hs21	40	35	iQQ-qetamacr	†	34	iQQ-stadat1	16	26
iQQ-hs268	26	45	iQQ-qfffff80	†	20	iQQ-stadat2	†	35
iQQ-hs35	**72**	16	iQQ-qforplan	21	11	iQQ-stadat3	†	25
iQQ-hs35mod	25	16	iQQ-qgfrdxpn	21	19	iQQ-tame	6	10
iQQ-hs51	29	27	iQQ-qgrow15	22	16	iQQ-ubh1	†	26
iQQ-hs52	22	26	iQQ-qgrow22	25	18	iQQ-yao	27	38
iQQ-hs53	27	39	iQQ-qgrow7	22	18	iQQ-zecevic2	8	9
iQQ-hs76	**69**	13	iQQ-qisrael	†	19	Avg1	42.20	28.15
iQQ-ksip	30	42	iQQ-qpcblend	**105**	38	Acg2	27.39	28.15

[1] †: Max number of iterations reached.
[2] ?: Terminated with unknown status.

Table A.12: Comparison with MOSEK on Maros and Meszaros's QCQP infeasible test problems

Problem	MOSEK	iOptimize
iairport	10	19
ipolak4	†	25
imakela1	28	32
imakela2	29	32
imakela3	8	14
igigomez1	26	26
ihanging	32	30
imadsschj	24	17
irosenmmx	27	35
ioptreward	11	20
ioptprloc	8	21
Avg	20.30	24.60

[1] †: Max number of iterations reached.

Table A.13: Comparison with `MOSEK` on `Cute` QCQP infeasible test problems

Problem	MOSEK	iOptimize	Ipopt	Knitro
ibigbank	**196**	27	48	42
igridnete	**134**	14	21	†
igridnetf	†	36	42	†
igridneth	**202**	23	25	18
igridneti	†	40	40	**133**§
idallasl	‡	‡	50§	‡
idallasm	‡	47	‡	0
idallass	‡	45	‡	0
ipolak1	50	16	‡	‡
ipolak2	7	13	‡	‡
ipolak3	**178**	27	⋄	†
ipolak5	7	13	37	†
ismbank	**165**	31	34	37
icantilvr	0	12	25	0
icb2	**196**	32	⋄	18
icb3	**196**	31	†	8
ichaconn1	**196**	32	⋄	6
ichaconn2	**196**	31	10	17
idipigri	**157**	32	**188**	†
igpp	†	38	**65**	†
ihong	11	22	16	†
iloadbal	**123**	14	22	†
isvanberg	†	27	48	53
antenna\iantenna2	†	**186**	†	**156**
antenna\iantenna_vareps	†	47	†	†
braess\itrafequil	43	38	36	**126**
braess\itrafequil2	7	9	**82**	50
braess\itrafequilsf	47	46	34	**103**
braess\itrafequil2sf	†	9	54	**91**
elena\ichemeq	14	14	**102**	16
elena\is383	7	14	**67**	†
firfilter\ifir_convex	0	13	**99**	0
markowitz\igrowthopt	15	16	22	†
wbv\iantenna2	0	17	32	0
wbv\ilowpass2	0	13	32	0
ibatch	0	22	32	**158**
istockcycle	**117**	35	**72**	†
isynthes1	9	25	35	35
isynthes2	**62**	23	48	46
isynthes3	53	40	**94**	†
itrimloss2	0	15	52	0
itrimloss4	0	16	40	0
itrimloss5	0	15	**60**	0
itrimloss6	0	15	55	0
itrimloss7	0	16	**61**	0
itrimloss12	0	16	**73**	0
Avg1	66.33	19.44	–	–
Avg2	11.74	16.04	–	–

[1] †: Max number of iterations reached.
[2] ‡: Function evaluate error from AMPL.
[3] §: Solver terminates with a near optimal status.
[4] ⋄: Ipopt fails in restoration phase.

Table A.14: Comparison with MOSEK, Ipopt, and Knitro on CP infeasible test problems