« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 1/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

Comput Optim Appl
DOI 10.1007/s10589-010-9389-4

A constraint-reduced variant of Mehrotra's predictor-corrector algorithm

Luke B. Winternitz · Stacey O. Nicholls · André L. Tits · Dianne P. O'Leary

Received: 13 January 2010

Abstract Consider linear programs in dual standard form with n constraints and m variables. When typical interior-point algorithms are used for the solution of such problems, updating the iterates, using direct methods for solving the linear systems and assuming a dense constraint matrix A, requires $\mathcal{O}(nm^2)$ operations per iteration. When $n \gg m$ it is often the case that at each iteration most of the constraints are not very relevant for the construction of a good update and could be ignored to achieve computational savings. This idea was considered in the 1990s by Dantzig and Ye,

This work was supported by NSF grant DMI0422931 and DoE grants DEFG0204ER25655 and DESC0002218. The work of the first author was supported by NASA under the Goddard Space Flight Center Study Fellowship Program. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation, those of the US Department of Energy, or those of NASA.

Electronic supplementary material The online version of this article (doi:10.1007/s10589-010-9389-4) contains supplementary material, which is available to authorized users.

L.B. Winternitz
NASA – Goddard Space Flight Center, Mail Stop 596, Greenbelt, MD 20771, USA
e-mail: Luke.B.Winternitz@nasa.gov

S.O. Nicholls
Applied Mathematics and Scientific Computing Program, University of Maryland, College Park, MD 20742, USA
e-mail: sonicholls@aacc.edu

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

D.P. O'Leary
Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA
e-mail: oleary@cs.umd.edu

Tone, Kaliski and Ye, den Hertog et al. and others. More recently, Tits et al. proposed a simple "constraint-reduction" scheme and proved global and local quadratic convergence for a dual-feasible primal-dual affine-scaling method modified according to that scheme. In the present work, similar convergence results are proved for a dual-feasible constraint-reduced variant of Mehrotra's predictor-corrector algorithm, under less restrictive nondegeneracy assumptions. These stronger results extend to primal-dual affine scaling as a limiting case. Promising numerical results are reported.

As a special case, our analysis applies to standard (unreduced) primal-dual affine scaling. While we do not prove polynomial complexity, our algorithm allows for much larger steps than in previous convergence analyses of such algorithms.

Keywords Linear programming · Linear optimization · Constraint reduction · Primal-dual interior-point methods · Mehrotra's predictor corrector

1 Introduction

Consider the primal and dual standard forms of linear programming (LP):

$$\begin{array}{rl} & \min c^{\mathrm{T}}x \\ \text{s.t.} & Ax = b, \\ & x \geq 0, \end{array} \quad \text{and} \quad \begin{array}{rl} & \max b^{\mathrm{T}}y \\ \text{s.t.} & A^{\mathrm{T}}y \leq c, \end{array} \tag{1}$$

where A is an $m \times n$ matrix with $n \gg m$, that is, the dual problem has many more inequality constraints than variables. We assume $b \neq 0$.[1] The dual problem can alternatively be written in the form (with slack variable s)

$$\begin{array}{rl} & \max b^{\mathrm{T}}y \\ \text{s.t.} & A^{\mathrm{T}}y + s = c, \\ & s \geq 0. \end{array} \tag{2}$$

Some of the most effective algorithms for solving LPs are the primal-dual interior-point methods (PDIPMs), which apply Newton's method, or variations thereof, to the perturbed Karush-Kuhn-Tucker (KKT) optimality conditions for the primal-dual pair (1):

$$\begin{aligned} A^{\mathrm{T}}y + s - c &= 0, \\ Ax - b &= 0, \\ Xs - \tau e &= 0, \\ (x, s) &\geq 0, \end{aligned} \tag{3}$$

[1] This assumption is benign, since if $b = 0$ the problem at hand is readily solved: any dual feasible point y^0 (assumed available for the algorithm analyzed in this paper) is dual optimal and (under our dual feasibility assumption) $x = 0$ is primal optimal.

« **COAP 10589** layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 3/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

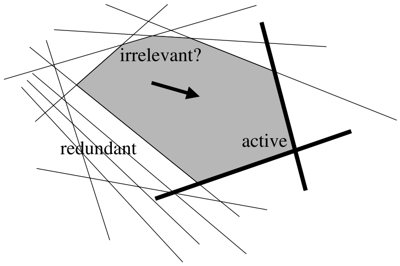

Fig. 1 A view of the y space when $m = 2$ and $n = 12$. The *arrow* indicates the direction of vector b. The two active constraints are critical and define the solution, while the others are redundant or perhaps not very relevant for the formation of good search directions

with $X = \operatorname{diag}(x)$, $S = \operatorname{diag}(s)$, e is the vector of all ones, and τ a positive scalar. As τ ranges over $(0, \infty)$, the unique (if it exists)[2] solution (x, y, s) to this system traces out the primal-dual "central path". Newton-type steps for system (3), which are well defined, e.g., when X and S are positive definite and A has full rank, are obtained by solving one or more linear systems of the form

$$\begin{pmatrix} 0 & A^{\mathrm{T}} & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} f \\ g \\ h \end{pmatrix}, \tag{4}$$

where f, g, and h are certain vectors of appropriate dimension. System (4) is often solved by first eliminating Δs, giving the "augmented system"

$$\begin{aligned} \begin{pmatrix} A & 0 \\ S & -XA^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} &= \begin{pmatrix} g \\ h - Xf \end{pmatrix}, \\ \Delta s &= f - A^{\mathrm{T}} \Delta y, \end{aligned} \tag{5}$$

or by further eliminating Δx, giving the "normal system"

$$\begin{aligned} AS^{-1}XA^{\mathrm{T}} \Delta y &= g - AS^{-1}(h - Xf), \\ \Delta s &= f - A^{\mathrm{T}} \Delta y, \\ \Delta x &= S^{-1}(h - X\Delta s). \end{aligned} \tag{6}$$

When $n \gg m$, a drawback of most interior-point methods is that the computational cost of determining a search direction is rather high. For example, in the context of PDIPMs, if we choose to solve (6) by a direct method and A is dense, the most expensive computation is forming the normal matrix $AS^{-1}XA^{\mathrm{T}}$, which costs $\mathcal{O}(nm^2)$ operations. This computation involves forming the sum

$$AS^{-1}XA^{\mathrm{T}} = \sum_{i=1}^{n} \frac{x_i}{s_i} a_i a_i^{\mathrm{T}}, \tag{7}$$

[2] System (3) has a unique solution for each $\tau > 0$ (equivalently, for *some* $\tau > 0$) if there exists (x, y, s) with $Ax = b$, $A^{\mathrm{T}} y + s = c$ and $(x, s) > 0$ [33, Thm. 2.8, p. 39].

« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 4/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

where a_i is the ith column of A and x_i and s_i are the ith components of x and s respectively, so that each term of the sum corresponds to a particular constraint in the dual problem. Note however that we expect most of the n constraints are redundant or not very relevant for the formation of a good search direction (see Fig. 1). In (7), if we were to select a small set of $q < n$ "important" constraints and compute only the corresponding partial sum, then the work would be reduced to $\mathcal{O}(qm^2)$ operations. Similar possibilities arise in other interior-point methods: by ignoring most of the constraints, we may hope that a "good" search direction can still be computed, at significantly reduced cost.[3] This observation is the basis of the present paper. In the sequel we refer to methods that attempt such computations as *constraint-reduced*.

Prior work investigating this question started at least as far back as Dantzig and Ye [4], who proposed a "build-up" variant of a dual affine-scaling algorithm. In their scheme, at each iteration, starting with a small working set of constraints, a dual affine-scaling step is computed. If this step is feasible with respect to the full constraint set, then it is taken. Otherwise, more constraints are added to the working set and the process is repeated. Convergence of this method was shown to follow from prior convergence results on the dual affine-scaling algorithm. At about the same time, Tone [30] developed an "active set" version of Ye's dual potential-reduction (DPR) algorithm [34]. There, starting with a small working set of constraints, a DPR-type search direction is computed. If a step along this direction gives a sufficient decrease of the potential function, then it is accepted. Otherwise, more constraints are added to the working set and the process is repeated. Convergence and complexity results are essentially inherited from the properties of the DPR algorithm. Kaliski and Ye [16] tailored Tone's algorithm to large scale transportation problems. By exploiting the structure of these problems, several significant enhancements to Tone's method were made, and some remarkable computational results were obtained.

A different approach was used by den Hertog et al. [5] who proposed a "build-up and down" path-following algorithm based on a dual logarithmic barrier method. Starting from an interior dual-feasible point, the central path corresponding to a small set of working constraints is followed until it becomes infeasible with respect to the full constraint set, whereupon the working set is appropriately updated and the process restarts from the previous iterate. The authors proved an $\mathcal{O}(\sqrt{q}\log\frac{1}{\varepsilon})$ iteration complexity bound for this algorithm, where q is the maximum size of the working constraint set during the iteration. Notably, this suggests that both the computational cost per iteration and the *iteration complexity* may be reduced. However, it appears that in this algorithm the only sure upper bound on q is n.

A common component of [4, 5, 30] is the backtracking that "adds constraints and tries again" when the step generated using the working constraint set fails to pass certain acceptability tests.[4] In contrast, no such backtracking is used in [29] where the authors considered constraint reduction for *primal-dual* algorithms. In particular, they proposed constraint-reduced versions of a primal-dual affine-scaling algorithm

[3] Such a step may even be *better*: see [5] (discussed two paragraphs below) and [6] for evidence of potential harm caused by redundant constraints.

[4] Kaliski and Ye [16] showed that this backtracking could be eliminated in their variant of Tone's algorithm for transportation problems.

(rPDAS) and of Mehrotra's Predictor-Corrector algorithm (rMPC). As in [4, 5, 30], at each iteration, rPDAS and rMPC use a small working set of constraints to generate a search direction, but this direction is not subjected to acceptability tests; it is simply taken. This has the advantage that the cost per iteration can be guaranteed to be cheaper than when the full constraint set is used; however it may preclude polynomial complexity results, as were obtained in [30] and [5]. Global and local quadratic convergence of rPDAS was proved (but convergence of rMPC was not analyzed) in [29], under nondegeneracy assumptions, using a nonlinear programming inspired line of argument [13, 22], and promising numerical results were reported for both rPDAS and rMPC.

Very recently, a "matrix-free" interior-point method was proposed by Gondzio, targeted at large-scale LPs, which appears to perform well, in particular, on large problems with many more inequality constraints than variables [11]. No "constraint reduction" is involved though; rather, the emphasis is on the use of suitably preconditioned iterative methods.

Finally, a wealth of non-interior-point methods can also be used toward the solution of problem (1) when $n \gg m$, including cutting-plane-based methods and column generation methods. The recent work of Mangasarian [17] deserves special attention as it is specifically targeted at handling linear programs with $n \gg m$ that arise in data mining and machine learning. Comparison with such methods is not pursued in the present work.

To our knowledge, aside from the analysis of rPDAS in [29], no attempts have been made to date at analyzing constraint-reduced versions of PDIPMs, the leading class of interior-points methods over the past decade. This observation applies in particular to the current "champion", Mehrotra's Predictor Corrector algorithm (MPC, [18]), which combines an adaptive choice of the perturbation parameter τ in (3), a second order correction to the Newton direction, and several ingenious heuristics which together have proven to be extremely effective in the solution of large scale problems. Investigations of the convergence properties of variants of MPC are reported in [2, 18, 25, 26, 35, 36].

The primary contribution of the present paper is theoretical. We provide a convergence analysis for a constraint-reduced variant of MPC that uses a minimally restrictive class of constraint selection rules. Furthermore, these constraint selection rules do not require "backtracking" or "minor iterations". We borrow from the line of analysis of [13, 22, 29] to analyze a proposed dual-feasible[5] constraint-reduced version of MPC, inspired from rMPC of [29], which we term rMPC*. We use a somewhat different, and perhaps more natural, perspective on the notion of constraint reduction than was put forth in [29] (see Remark 2.1 below). We prove global convergence under assumptions that are significantly milder than those invoked in [29]. We then prove q-quadratic local convergence under appropriate nondegeneracy assumptions. The proposed iteration and stronger convergence results apply, as a limiting case, to constrained-reduced primal-dual *affine scaling*, thus improving on the results of [29].

[5] In recent work, carried out while the present paper was under first review, an "infeasible" version of the algorithm discussed in the present paper was developed and analyzed. It involves an exact penalty function and features a scheme that adaptively adjusts the penalty parameter, guaranteeing that an appropriate value is eventually achieved. See [12].

As a further special case, they apply to standard (unreduced) primal-dual affine scaling. In that context, our conclusions (Theorem 3.8 and Remark 3.1) are weaker than those obtained in the work of Monteiro et al. [20] or, for a different type of affine scaling (closer to the spirit of Dikin's work [7]), in that of Jansen et al. [15]. In particular, we do not prove polynomial complexity. On the other hand, the specific algorithm we analyze has the advantage of allowing for much larger steps, of the order of one compared to steps no larger than $1/n$ (in [20]) or equal to $1/(15\sqrt{n})$ (in [15]), and convergence results on the dual sequence are obtained without an assumption of primal feasibility.

The notation in this paper is mostly standard. We use $\|\cdot\|$ to denote the 2-norm or its induced operator norm. Given a vector $x \in \mathbb{R}^n$, we let the corresponding capital letter X denote the diagonal $n \times n$ matrix with x on its main diagonal. We define $\mathbf{n} := \{1, 2, \ldots, n\}$ and given any index set $Q \subseteq \mathbf{n}$, we use A_Q to denote the $m \times |Q|$ (where $|Q|$ is the cardinality of Q) matrix obtained from A by deleting all columns a_i with $i \notin Q$. Similarly, we use x_Q and s_Q to denote the vectors of size $|Q|$ obtained from x and s by deleting all entries x_i and s_i with $i \notin Q$. We define e to be the column vector of ones, with length determined by context. For a vector v, $[v]_-$ is defined by $([v]_-)_i := \min\{v_i, 0\}$. Lowercase k always indicates an iteration count, and limits of the form $y^k \to y^*$ are meant as $k \to \infty$. Uppercase K generally refers to an infinite index set and the qualification "on K" is synonymous with "for $k \in K$". In particular, "$y^k \to y^*$ on K" means $y^k \to y^*$ as $k \to \infty$, $k \in K$. Further, we define the dual feasible, dual strictly feasible, and dual solution sets, respectively, as

$$F := \{y \in \mathbb{R}^m \mid A^{\mathrm{T}} y \le c\},$$

$$F^o := \{y \in \mathbb{R}^m \mid A^{\mathrm{T}} y < c\},$$

$$F^* := \{y \in F \mid b^{\mathrm{T}} y \ge b^{\mathrm{T}} w \text{ for all } w \in F\}.$$

We term a vector $y \in \mathbb{R}^m$ *stationary* if $y \in F^{\mathrm{s}}$, where

$$F^{\mathrm{s}} := \{y \in F \mid \exists x \in \mathbb{R}^n, \text{ s.t. } Ax = b,\ X(c - A^{\mathrm{T}} y) = 0\}. \tag{8}$$

Given $y \in F^{\mathrm{s}}$, every x satisfying the conditions of (8) is called *a multiplier associated to the stationary point* y. A stationary vector y belongs to F^* if and only if $x \ge 0$ for some multiplier x. The active set at $y \in F$ is

$$I(y) := \{i \in \mathbf{n} \mid a_i^{\mathrm{T}} y = c_i\}.$$

Finally we define

$$J(G, u, v) := \begin{pmatrix} 0 & G^{\mathrm{T}} & I \\ G & 0 & 0 \\ \operatorname{diag}(v) & 0 & \operatorname{diag}(u) \end{pmatrix} \tag{9}$$

and

$$J_{\mathrm{a}}(G, u, v) := \begin{pmatrix} G & 0 \\ \operatorname{diag}(v) & -\operatorname{diag}(u) G^{\mathrm{T}} \end{pmatrix} \tag{10}$$

« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 7/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

for any matrix G and vectors u and v of compatible dimensions (cf. systems (4) and (5)).

The first lemma is taken (almost) verbatim from [29, Lemma 1].

Lemma 1.1 *$J_a(A,x,s)$ is nonsingular if and only if $J(A,x,s)$ is. Further suppose $x \geq 0$ and $s \geq 0$. Then $J(A,x,s)$ is nonsingular if and only if (i) $x_i + s_i > 0$ for all i; (ii) $\{a_i : s_i = 0\}$ is linearly independent; and (iii) $\{a_i : x_i \neq 0\}$ spans $\mathbb{R}^m$.*

The rest of the paper is structured as follows. Section 2 contains the definition and discussion of algorithm rMPC*. Sections 3 and 4 contain the global analysis and a statement of the local convergence results, respectively. The proof of the local convergence results is given in an appendix available as Electronic Supplementary Material. Some numerical results are presented in Sect. 5, and conclusions are drawn in Sect. 6.

2 A constraint-reduced MPC algorithm

2.1 A convergent variant of MPC

Our proposed algorithm, rMPC*, is based on the implementation of MPC discussed in [33, Ch. 10], which we reproduce here for ease of reference.

Iteration MPC [18, 33].
Parameter. $\beta \in (0,1)$.
Data. $y \in \mathbb{R}^m$, $s \in \mathbb{R}^n$ with $s > 0$, $x \in \mathbb{R}^n$ with $x > 0$.

Step 1. Compute the affine scaling direction, i.e., solve

$$\begin{pmatrix} 0 & A^{\mathrm{T}} & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^{\mathrm{a}} \\ \Delta y^{\mathrm{a}} \\ \Delta s^{\mathrm{a}} \end{pmatrix} = \begin{pmatrix} c - A^{\mathrm{T}} y - s \\ b - Ax \\ -Xs \end{pmatrix} \tag{11}$$

for $(\Delta x^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s^{\mathrm{a}})$ and set

$$t_{\mathrm{p}}^{\mathrm{a}} := \arg\max\{t \in [0,1] \mid x + t\Delta x^{\mathrm{a}} \geq 0\}, \tag{12}$$

$$t_{\mathrm{d}}^{\mathrm{a}} := \arg\max\{t \in [0,1] \mid s + t\Delta s^{\mathrm{a}} \geq 0\}. \tag{13}$$

Step 2. Set $\mu := x^{\mathrm{T}} s/n$ and compute the "centering parameter"

$$\sigma := (\mu^{\mathrm{a}}/\mu)^3, \tag{14}$$

where $\mu^{\mathrm{a}} := (x + t_{\mathrm{p}}^{\mathrm{a}} \Delta x^{\mathrm{a}})^{\mathrm{T}} (s + t_{\mathrm{d}}^{\mathrm{a}} \Delta s^{\mathrm{a}})/n$.

Step 3. Compute the centering/corrector direction, i.e., solve

$$\begin{pmatrix} 0 & A^{\mathrm{T}} & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x^{\mathrm{c}} \\ \Delta y^{\mathrm{c}} \\ \Delta s^{\mathrm{c}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \sigma\mu e - \Delta X^{\mathrm{a}} \Delta s^{\mathrm{a}} \end{pmatrix} \tag{15}$$

for $(\Delta x^{\mathrm{c}}, \Delta y^{\mathrm{c}}, \Delta s^{\mathrm{c}})$.

Step 4. Form the total search direction

$$(\Delta x^{\mathrm{m}}, \Delta y^{\mathrm{m}}, \Delta s^{\mathrm{m}}) := (\Delta x^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s^{\mathrm{a}}) + (\Delta x^{\mathrm{c}}, \Delta y^{\mathrm{c}}, \Delta s^{\mathrm{c}}), \tag{16}$$

and set

$$\bar{t}_{\mathrm{p}}^{\mathrm{m}} := \arg\max\{t \in [0, 1] \mid x + t\Delta x^{\mathrm{m}} \geq 0\}, \tag{17}$$

$$\bar{t}_{\mathrm{d}}^{\mathrm{m}} := \arg\max\{t \in [0, 1] \mid s + t\Delta s^{\mathrm{m}} \geq 0\}. \tag{18}$$

Step 5. Update the variables: set

$$t_{\mathrm{p}}^{\mathrm{m}} := \beta \bar{t}_{\mathrm{p}}^{\mathrm{m}}, \ t_{\mathrm{d}}^{\mathrm{m}} := \beta \bar{t}_{\mathrm{d}}^{\mathrm{m}}, \tag{19}$$

and set

$$(x^+, y^+, s^+) := (x, y, s) + (t_{\mathrm{p}}^{\mathrm{m}} \Delta x^{\mathrm{m}}, \ t_{\mathrm{d}}^{\mathrm{m}} \Delta y^{\mathrm{m}}, \ t_{\mathrm{d}}^{\mathrm{m}} \Delta s^{\mathrm{m}}). \tag{20}$$

Algorithm MPC is of the "infeasible" type, in that it does not require the availability of a feasible initial point.[6] In contrast, the global convergence analysis for Algorithm rMPC⋆ (see Sect. 3 below) critically relies on the monotonic increase of the dual objective $b^{\mathrm{T}} y$ from iteration to iteration, and for this we do need a dual feasible initial point.

As stated, Iteration MPC has no known convergence guarantees. Previous approaches to providing such guarantees involve introducing certain safeguards or modifications [2, 18, 25, 26, 35, 36]. We do this here as well. Specifically, aside from the constraint-reduction mechanism (to be discussed in Sect. 2.2), Iteration rMPC⋆ proposed below has *four differences* from Iteration MPC, all motivated by the structure of the convergence analysis adapted from [13, 22, 28, 29]. These differences, which occur in Steps 2, 4, and 5, are discussed next. Our numerical experience (in particular, that reported in Sect. 5.4) suggests that they can affect positively or negatively the performance of the algorithms, but seldom to a dramatic extent.

The *first difference*, in Step 2, is the formula for the centering parameter σ. Instead of using (14), we set

$$\sigma := (1 - t^{\mathrm{a}})^{\lambda},$$

where $t^{\mathrm{a}} := \min\{t_{\mathrm{p}}^{\mathrm{a}}, \ t_{\mathrm{d}}^{\mathrm{a}}\}$ and $\lambda \geq 2$ is a scalar algorithm parameter. This formula agrees with (14) when $\lambda = 3$, (x, y, s) is primal and dual feasible, and $t^{\mathrm{a}} = t_{\mathrm{p}}^{\mathrm{a}} = t_{\mathrm{d}}^{\mathrm{a}}$. It simplifies our analysis.

The *second difference* is in Step 4, where we introduce a *mixing parameter* $\gamma \in (0, 1]$ and replace (16) with[7]

$$(\Delta x^{\mathrm{m}}, \Delta y^{\mathrm{m}}, \Delta s^{\mathrm{m}}) := (\Delta x^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s^{\mathrm{a}}) + \gamma(\Delta x^{\mathrm{c}}, \Delta y^{\mathrm{c}}, \Delta s^{\mathrm{c}}). \tag{21}$$

---

[6] In MPC and rMPC⋆ and most other PDIPMs, dual (resp. primal) feasibility of the initial iterate implies dual (primal) feasibility of all subsequent iterates.

[7] Such mixing is also recommended in [3], the aim there being to enhance the practical efficiency of MPC by allowing a larger stepsize.

Nominally we want $\gamma = 1$, but we reduce γ as needed to enforce three properties of our algorithm that are needed in the analysis.

- The first such property is the monotonic increase of $b^T y$ mentioned previously. While, given dual feasibility, it is readily verified that Δy^a is an ascent direction for $b^T y$ (i.e., $b^T \Delta y^a > 0$), this may not be the case for Δy^m, defined in (16). To enforce monotonicity we choose $\gamma \le \gamma_1$ where γ_1 is the largest number in $[0, 1]$ such that

$$b^T(\Delta y^a + \gamma_1 \Delta y^c) \ge \theta b^T \Delta y^a, \tag{22}$$

with $\theta \in (0, 1)$ an algorithm parameter. It is easily verified that γ_1 is given by

$$\gamma_1 = \begin{cases} 1 & \text{if } b^T \Delta y^c \ge 0, \\ \min\{1, (1-\theta)\frac{b^T \Delta y^a}{|b^T \Delta y^c|}\} & \text{else.} \end{cases} \tag{23}$$

- The second essential property addressed via the mixing parameter is that the centering-corrector component cannot be too large relative to the affine-scaling component. Specifically, we require

$$\|\gamma \Delta y^c\| \le \psi \|\Delta y^a\|, \qquad \|\gamma \Delta x^c\| \le \psi \|x + \Delta x^a\| \quad \text{and} \quad \gamma \sigma \mu \le \psi \|\Delta y^a\|,$$

where $\psi \ge 0$ is another algorithm parameter.[8] This property is enforced by requiring $\gamma \le \gamma_0$, where

$$\gamma_0 := \min\left\{\gamma_1, \psi \frac{\|\Delta y^a\|}{\|\Delta y^c\|}, \psi \frac{\|x + \Delta x^a\|}{\|\Delta x^c\|}, \psi \frac{\|\Delta y^a\|}{\sigma \mu}\right\}. \tag{24}$$

- The final property enforced by γ is that

$$\bar{t}_d^m \ge \zeta t_d^a, \tag{25}$$

where $\zeta \in (0, 1)$ is a third algorithm parameter and $\bar{t}_d^m$ depends on γ via (18) and (21). We could choose γ to be the largest number in $[0, \gamma_0]$ such that (25) holds, but this would seem to require a potentially expensive iterative procedure. Instead, rMPC* sets

$$\gamma := \begin{cases} \gamma_0 & \text{if } \bar{t}_{d,0}^m \ge \zeta t_d^a, \\ \gamma_0 \frac{(1-\zeta)\bar{t}_{d,0}^m}{(1-\zeta)\bar{t}_{d,0}^m + (\zeta t_d^a - \bar{t}_{d,0}^m)} & \text{else,} \end{cases} \tag{26}$$

where

$$\bar{t}_{d,0}^m := \arg\max\{t \in [0, 1] \mid s + t(\Delta s^a + \gamma_0 \Delta s^c) \ge 0\}. \tag{27}$$

Geometrically, if $\bar{t}_{d,0}^m \ge \zeta t_d^a$ then $\gamma = \gamma_0$, but otherwise $\gamma \in [0, \gamma_0)$ is selected in such a way that the search direction $\Delta s^m = \Delta s^a + \gamma \Delta s^c$ goes through the intersection of the line segment connecting $s + \zeta t_d^a \Delta s^a$ and $s + \zeta t_d^a(\Delta s^a + \gamma_0 \Delta s^c)$ with

---

[8] If $\psi = 0$, then rMPC* essentially becomes rPDAS, the constraint-reduced affine scaling algorithm analyzed in [29].

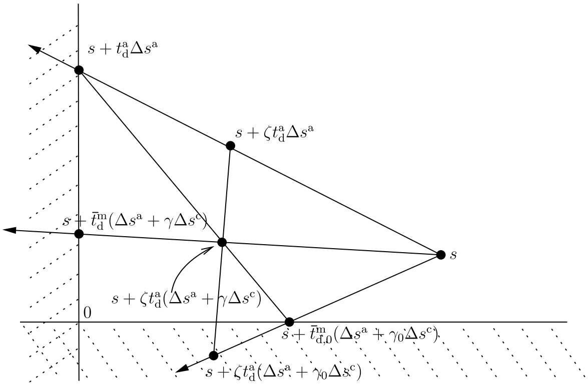

Fig. 2 Enforcing $\bar{t}_{\mathrm{d}}^{\mathrm{m}} \geq \zeta t_{\mathrm{d}}^{\mathrm{a}}$ with γ. The positive orthant here represents the feasible set $s \geq 0$ in two-dimensional slack space. The *top arrow* shows the step taken from some $s > 0$ along the affine scaling direction Δs^{a}. The *bottom arrow* is the step along the MPC direction with mixing parameter γ_0. In this picture, the damping factor $\bar{t}_{\mathrm{d},0}^{\mathrm{m}}$ is less than $\zeta t_{\mathrm{d}}^{\mathrm{a}}$, so we do not choose $\gamma = \gamma_0$. Rather, we take a step along the direction from s that passes through the intersection of two lines: the line consisting of points of the form $s + \zeta t_{\mathrm{d}}^{\mathrm{a}}(\Delta s^{\mathrm{a}} + \gamma\Delta s^{\mathrm{c}})$ with $\gamma \in [0, \gamma_0]$ and the feasible line connecting $s + t_{\mathrm{d}}^{\mathrm{a}}\Delta s^{\mathrm{a}}$ and $s + \bar{t}_{\mathrm{d},0}^{\mathrm{m}}(\Delta s^{\mathrm{a}} + \gamma_0\Delta s^{\mathrm{c}})$. The maximum feasible step along this direction has length $\bar{t}_{\mathrm{d}}^{\mathrm{m}} \geq \zeta t_{\mathrm{d}}^{\mathrm{a}}$

the feasible line segment connecting $s + t_{\mathrm{d}}^{\mathrm{a}}\Delta s^{\mathrm{a}}$ and $s + \bar{t}_{\mathrm{d},0}^{\mathrm{m}}(\Delta s^{\mathrm{a}} + \gamma_0\Delta s^{\mathrm{c}})$. See Fig. 2. Since the intersection point $s + \zeta t_{\mathrm{d}}^{\mathrm{a}}(\Delta s^{\mathrm{a}} + \gamma\Delta s^{\mathrm{c}})$ is feasible, (25) will hold. Overall we have

$$\gamma \in [0, \gamma_0] \subseteq [0, \gamma_1] \subseteq [0, 1]. \tag{28}$$

In spite of these three requirements on γ, it is typical that $\gamma = 1$ in practice, with appropriate choice of algorithm parameters, as in Sect. 5, except when aggressive constraint reduction is used—i.e., very few constraints are retained at each iteration.

The *remaining two differences* between rMPC* and MPC, aside from constraint reduction, are in Step 5. They are both taken from [29]. First, (19) is replaced by

$$t_{\mathrm{p}}^{\mathrm{m}} := \max\{\beta\bar{t}_{\mathrm{p}}^{\mathrm{m}},\ \bar{t}_{\mathrm{p}}^{\mathrm{m}} - \|\Delta y^{\mathrm{a}}\|\} \tag{29}$$

and similarly for $t_{\mathrm{d}}^{\mathrm{m}}$, to allow for local quadratic convergence. Second, the primal update is replaced by a componentwise clipped-from-below version of the primal update in (20). Namely, defining $\hat{x} := x + t_{\mathrm{p}}^{\mathrm{m}}\Delta x^{\mathrm{m}}$ and $\tilde{x}^{\mathrm{a}} := x + \Delta x^{\mathrm{a}}$, for all $i \in \mathbf{n}$, we update x_i to

$$x_i^+ := \max\{\hat{x}_i,\ \min\{\underline{\xi}^{\max}, \|\Delta y^{\mathrm{a}}\|^\nu + \|[\tilde{x}^{\mathrm{a}}]_-\|^\nu\}\}, \tag{30}$$

where $\nu \geq 2$ and $\underline{\xi}^{\max} > 0$ (small) are algorithm parameters.[9] The lower bound, $\min\{\underline{\xi}^{\max}, \|\Delta y^{\mathrm{a}}\|^\nu + \|[\tilde{x}^{\mathrm{a}}]_-\|^\nu\}$, ensures that, away from KKT points, the components of x remain bounded away from zero, which is crucial to the global convergence analysis, while allowing for local quadratic convergence. Parameter $\underline{\xi}^{\max}$,

[9]In [29], the primal update is also clipped from above by a large, user selected value, to insure boundedness of the primal sequence. We show in Lemma 3.4 below that such clipping is unnecessary.

a maximum value imposed on the lower bound, ensures boundedness (as the iteration counter goes to infinity) of the lower bound, which is needed in proving that the sequence of primal iterates remains bounded (Lemma 3.4 below). A reasonably low value of $\underline{\xi}^{\max}$ is also important in practice.

2.2 A constraint reduction mechanism

Given a working set of constraints Q and a dual-feasible point (x, y, s), we compute an MPC-type direction for the "reduced" primal-dual pair

$$\begin{array}{rl} & \min c_Q^{\mathrm{T}} x_Q \\ \text{s.t.} & A_Q x_Q = b, \\ & x_Q \geq 0, \end{array} \qquad \text{and} \qquad \begin{array}{rl} & \max b^{\mathrm{T}} y \\ \text{s.t.} & A_Q^{\mathrm{T}} y + s_Q = c_Q, \\ & s_Q \geq 0. \end{array} \tag{31}$$

To that effect, we first compute the "reduced" affine-scaling direction by solving

$$\begin{pmatrix} 0 & A_Q^{\mathrm{T}} & I_Q \\ A_Q & 0 & 0 \\ S_Q & 0 & X_Q \end{pmatrix} \begin{pmatrix} \Delta x_Q^{\mathrm{a}} \\ \Delta y^{\mathrm{a}} \\ \Delta s_Q^{\mathrm{a}} \end{pmatrix} = \begin{pmatrix} 0 \\ b - A_Q x_Q \\ -X_Q s_Q \end{pmatrix} \tag{32}$$

and then the "reduced" centering-corrector direction by solving

$$\begin{pmatrix} 0 & A_Q^{\mathrm{T}} & I_Q \\ A_Q & 0 & 0 \\ S_Q & 0 & X_Q \end{pmatrix} \begin{pmatrix} \Delta x_Q^{\mathrm{c}} \\ \Delta y^{\mathrm{c}} \\ \Delta s_Q^{\mathrm{c}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \sigma \mu_Q e - \Delta X_Q^{\mathrm{a}} \Delta s_Q^{\mathrm{a}} \end{pmatrix}, \tag{33}$$

where $\mu_Q := (x_Q)^{\mathrm{T}}(s_Q)/|Q|$. As discussed above, we combine these components using the mixing parameter γ to get our primal and dual search directions:

$$(\Delta x_Q^{\mathrm{m}}, \Delta y^{\mathrm{m}}, \Delta s_Q^{\mathrm{m}}) := (\Delta x_Q^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s_Q^{\mathrm{a}}) + \gamma(\Delta x_Q^{\mathrm{c}}, \Delta y^{\mathrm{c}}, \Delta s_Q^{\mathrm{c}}). \tag{34}$$

This leaves unspecified the search direction in the $\mathbf{n} \setminus Q$ components of Δx^{m} and Δs^{m}. However, in conjunction with an update of the form (20), maintaining dual feasibility from iteration to iteration requires that we set

$$\Delta s_{\mathbf{n}\setminus Q}^{\mathrm{a}} := -A_{\mathbf{n}\setminus Q}^{\mathrm{T}} \Delta y^{\mathrm{a}} \quad \text{and} \quad \Delta s_{\mathbf{n}\setminus Q}^{\mathrm{c}} := -A_{\mathbf{n}\setminus Q}^{\mathrm{T}} \Delta y^{\mathrm{c}}.$$

Thus, we augment (34) accordingly, yielding the search direction for x_Q, y, and s,

$$(\Delta x_Q^{\mathrm{m}}, \Delta y^{\mathrm{m}}, \Delta s^{\mathrm{m}}) = (\Delta x_Q^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s^{\mathrm{a}}) + \gamma(\Delta x_Q^{\mathrm{c}}, \Delta y^{\mathrm{c}}, \Delta s^{\mathrm{c}}). \tag{35}$$

As for $x_{\mathbf{n}\setminus Q}$, we do not update it by taking a step along a computed direction. Rather, inspired by an idea used in [30], we consider the update

$$x_i^+ := \frac{\mu_Q^+}{s_i^+} \quad i \in \mathbf{n} \setminus Q,$$

where $\mu_Q^+ := (x_Q^+)^T(s_Q^+)/|Q|$. This would make $(x_{\mathbf{n}\setminus Q}^+, s_{\mathbf{n}\setminus Q}^+)$ perfectly "centered". Indeed,

$$\mu^+ = \frac{(x^+)^T(s^+)}{n} = \frac{(x_Q^+)^T(s_Q^+) + (x_{\mathbf{n}\setminus Q}^+)^T(s_{\mathbf{n}\setminus Q}^+)}{n} = \frac{|Q|}{n}\mu_Q^+ + \sum_{i\in\mathbf{n}\setminus Q}\frac{x_i^+ s_i^+}{n}$$
$$= \frac{|Q|}{n}\mu_Q^+ + \frac{n-|Q|}{n}\mu_Q^+ = \mu_Q^+,$$

and hence $x_i^+ s_i^+ = \mu^+$ for all $i \in \mathbf{n} \setminus Q$. However, in order to ensure boundedness of the primal iterates, we use instead, for $i \in \mathbf{n} \setminus Q$,

$$\hat{x}_i := \frac{\mu_Q^+}{s_i^+}, \qquad x_i^+ := \min\{\hat{x}_i, \chi\}, \tag{36}$$

where $\chi > 0$ is a large parameter. This clipping is innocuous because, as proved in the ensuing analysis, under our stated assumptions, all the $n \setminus Q$ components of the vector x constructed by Iteration rMPC* will be small eventually, regardless of how Q may change from iteration to iteration. In practice, this upper bound will never be active if χ is chosen reasonably large.

Remark 2.1 A somewhat different approach to constraint-reduction, where the motivating idea of ignoring irrelevant constraints is less prominent, is used in [29]. There, instead of the reduced systems (32)–(33), full systems of equations of the form (4) are solved via the corresponding normal systems (6), only with the normal matrix $AS^{-1}XA^T$ replaced by the reduced normal matrix $A_Q S_Q^{-1} X_Q A_Q^T$. Possible benefits of the approach taken here in rMPC*are: (i) the [29] approach is essentially tied to the normal equations, whereas our approach is not, (ii) if we do solve the normal equations (64) (below) there is a (mild) computational savings over algorithm rMPC of [29], and (iii) our initial computational experiments suggest that rMPC* is at least as efficient as rMPC in practice (e.g., see Table 3 in Sect. 5.4).

2.2.1 Rules for selecting Q

Before formally stating Iteration rMPC*, we describe a general constraint selection rule under which our convergence analysis can be carried out. We use a rule related to the one used in [29] in that we require Q to contain some number of nearly active constraints at the current iterate y.[10] However, the rule here aims to allow the convergence analysis to be carried out under weaker assumptions on the problem data than those used in [29]. In particular, we explicitly require that the selection of Q ensures $\mathrm{rank}(A_Q) = m$, whereas, in [29], this rank condition is enforced indirectly through a

[10] Of course, nearness to activity can be measured in different ways. Here, the "activity" of a dual constraint refers to the magnitude of the slack value s_i associated to it. When the columns of A are normalized to unit 2-norm, the slack in a constraint is just the Euclidean distance to the constraint boundary. Also see Remark 2.3 below on invariance under scaling.

rather strong assumption on A. Also, the choice made here makes it possible to potentially[11] eliminate a strong linear independence assumption, namely, Assumption 3 of [29], equivalent to "nondegeneracy" of all "dual basic feasible solutions".

Before stating the rule, we define two terms used throughout the paper. For a natural number $M \geq 0$ and a real number $\epsilon > 0$, a set of "M most-active" and the set of "ϵ-active" constraints refer, respectively, to a set of constraints with the M smallest slack values (ties broken arbitrarily) and the set of all constraints with slack value no larger than ϵ.

Definition 2.1 Let $\epsilon \in (0, \infty]$, and let $M \in \mathbf{n}$. Then a set $Q \subseteq \mathbf{n}$ belongs to $\mathfrak{Q}_{\epsilon,M}(y)$ if and only if it contains (as a subset) all ϵ-active constraints at y among some set of M most-active constraints.

Rule 2.1 *At a dual feasible point y, for $\epsilon > 0$ and $M \in \mathbf{n}$ an upper bound on the number of constraints active at any dual feasible point, select Q from the set $\mathfrak{Q}_{\epsilon,M}(y)$ in such a way that A_Q has full row rank.*

To help clarify Rule 2.1, we now describe two extreme variants. First, if the problem is known to be nondegenerate in the sense that the set of vectors a_i associated to dual active constraints at any feasible point y is a linearly independent set, we may set $M = m$ and $\epsilon = \infty$. Then, a minimal Q will consist of m most active constraints, achieving "maximum" constraint reduction. On the other hand, if we have no prior knowledge of the problem, $M = n$ is the only sure choice, and in this case we may set ϵ equal to a small positive value to enact the constraint reduction.

Rule 2.1 leaves a lot of freedom in choosing the constraint set. In practice, we have had most success with specific rules that keep a small number, typically $2m$ or $3m$, most-active constraints and then add additional constraints based on heuristics suggested by prior knowledge of the problem structure.

The following two lemmas are immediate consequences of Rule 2.1.

Lemma 2.2 *Let $x > 0$, $s > 0$, and $Q \in \mathfrak{Q}_{\epsilon,M}(y)$ for some $y \in F$. Then $A_Q X_Q S_Q^{-1} A_Q^{\mathrm{T}}$ is positive definite.*

Lemma 2.3 *Given $y' \in F$, there exists $\rho > 0$ such that for every $Q \in \mathfrak{Q}_{\epsilon,M}(y)$ with $y \in B(y', \rho) \cap F$ we have $I(y') \subseteq Q$.*

Before specifying Iteration rMPC*, we state two basic assumptions that guarantee it is well defined.

Assumption 2.1 *A has full row rank.*

Assumption 2.2 *The dual strictly feasible set is nonempty.*

[11] Within the present effort, this unfortunately was not achieved: we re-introduced such assumption at the last step of the analysis; see Theorem 3.8.

All that is needed for the iteration to be well-defined is the existence of a dual strictly feasible point y, that $\mathfrak{Q}_{\epsilon,M}(y)$ be nonempty, and that the linear systems (32) and (33), of Steps 1 and 3, be solvable. Under Assumption 2.1, $\mathfrak{Q}_{\epsilon,M}(y)$ is always nonempty since it then contains $\mathbf{n}$. Solvability of the linear systems then follows from Lemma 1.1 and Rule 2.1.

Iteration rMPC*.

Parameters. $\beta \in (0,1)$, $\theta \in (0,1)$, $\psi \geq 0$, $\chi > 0$, $\zeta \in (0,1)$, $\lambda \geq 2$, $\nu \geq 2$, $\underline{\xi}^{\max} \in (0,\infty]$,[12] $\epsilon \in (0,\infty]$ and an upper bound $M \in \mathbf{n}$ on the number of constraints active at any dual feasible point.

Data. $y \in \mathbb{R}^m$, $s \in \mathbb{R}^n$ such that $A^{\mathrm{T}} y + s = c$ and $s > 0$, $x \in \mathbb{R}^n$ such that $x > 0$.

Step 1. Choose Q according to Rule 2.1, compute the affine scaling direction, i.e., solve (32) for $(\Delta x_Q^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s_Q^{\mathrm{a}})$, and set

$$\Delta s_{\mathbf{n}\setminus Q}^{\mathrm{a}} := -A_{\mathbf{n}\setminus Q}^{\mathrm{T}} \Delta y^{\mathrm{a}},$$

$$t_{\mathrm{p}}^{\mathrm{a}} := \arg\max\{t \in [0,1] \mid x_Q + t\Delta x_Q^{\mathrm{a}} \geq 0\}, \tag{37}$$

$$t_{\mathrm{d}}^{\mathrm{a}} := \arg\max\{t \in [0,1] \mid s + t\Delta s^{\mathrm{a}} \geq 0\}, \tag{38}$$

$$t^{\mathrm{a}} := \min\{t_{\mathrm{p}}^{\mathrm{a}}, t_{\mathrm{d}}^{\mathrm{a}}\}. \tag{39}$$

Step 2. Set $\mu_Q := \frac{(x_Q)^{\mathrm{T}}(s_Q)}{|Q|}$ and compute the centering parameter

$$\sigma := \left(1 - t^{\mathrm{a}}\right)^{\lambda}. \tag{40}$$

Step 3. Compute the centering/corrector direction, i.e., solve (33) for $(\Delta x_Q^{\mathrm{c}}, \Delta y^{\mathrm{c}}, \Delta s_Q^{\mathrm{c}})$ and set

$$\Delta s_{\mathbf{n}\setminus Q}^{\mathrm{c}} := -A_{\mathbf{n}\setminus Q}^{\mathrm{T}} \Delta y^{\mathrm{c}}.$$

Step 4. Form the total search direction

$$(\Delta x_Q^{\mathrm{m}}, \Delta y^{\mathrm{m}}, \Delta s^{\mathrm{m}}) := (\Delta x_Q^{\mathrm{a}}, \Delta y^{\mathrm{a}}, \Delta s^{\mathrm{a}}) + \gamma(\Delta x_Q^{\mathrm{c}}, \Delta y^{\mathrm{c}}, \Delta s^{\mathrm{c}}), \tag{41}$$

where γ is as in (26), with μ (in (24)) replaced by μ_Q. Set

$$\bar{t}_{\mathrm{p}}^{\mathrm{m}} := \arg\max\{t \in [0,1] \mid x_Q + t\Delta x_Q^{\mathrm{m}} \geq 0\}, \tag{42}$$

$$\bar{t}_{\mathrm{d}}^{\mathrm{m}} := \arg\max\{t \in [0,1] \mid s + t\Delta s^{\mathrm{m}} \geq 0\}. \tag{43}$$

Step 5. Update the variables: set

[12] The convergence analysis allows for $\underline{\xi}^{\max} = \infty$, i.e., for the simplified version of (49): $x_i^+ := \max\{\hat{x}_i, \phi\}$. However a finite, small value of $\underline{\xi}^{\max}$ seems to be beneficial in practice.

$$t_{\mathrm{p}}^{\mathrm{m}} := \max\{\beta \bar{t}_{\mathrm{p}}^{\mathrm{m}},\ \bar{t}_{\mathrm{p}}^{\mathrm{m}} - \|\Delta y^{\mathrm{a}}\|\}, \tag{44}$$

$$t_{\mathrm{d}}^{\mathrm{m}} := \max\{\beta \bar{t}_{\mathrm{d}}^{\mathrm{m}},\ \bar{t}_{\mathrm{d}}^{\mathrm{m}} - \|\Delta y^{\mathrm{a}}\|\}, \tag{45}$$

and set

$$(\hat{x}_Q, y^+, s^+) := (x_Q, y, s) + (t_{\mathrm{p}}^{\mathrm{m}} \Delta x_Q^{\mathrm{m}},\ t_{\mathrm{d}}^{\mathrm{m}} \Delta y^{\mathrm{m}},\ t_{\mathrm{d}}^{\mathrm{m}} \Delta s^{\mathrm{m}}). \tag{46}$$

Set

$$\tilde{x}_i^{\mathrm{a}} := \begin{cases} x_i + \Delta x_i^{\mathrm{a}} & i \in Q, \\ 0 & i \in \mathbf{n} \setminus Q, \end{cases} \tag{47}$$

$$\phi := \|\Delta y^{\mathrm{a}}\|^{\nu} + \|[\tilde{x}^{\mathrm{a}}]_-\|^{\nu}, \tag{48}$$

and for each $i \in Q$, set

$$x_i^+ := \max\{\hat{x}_i,\ \min\{\underline{\xi}^{\max}, \phi\}\}. \tag{49}$$

Set

$$\mu_Q^+ := \frac{(x_Q^+)^{\mathrm{T}}(s_Q^+)}{|Q|} \tag{50}$$

and, for each $i \in \mathbf{n} \setminus Q$, set

$$\hat{x}_i := \frac{\mu_Q^+}{s_i^+}, \tag{51}$$

$$x_i^+ := \min\{\hat{x}_i, \chi\}. \tag{52}$$

□

In the convergence analysis, we will also make use of the quantities $\tilde{x}^{\mathrm{m}}$, $\tilde{s}^{\mathrm{a}}$, and $\tilde{s}^{\mathrm{m}}$ defined by the expressions

$$\tilde{x}_i^{\mathrm{m}} := \begin{cases} x_i + \Delta x_i^{\mathrm{m}} & i \in Q, \\ 0 & i \in \mathbf{n} \setminus Q, \end{cases} \tag{53}$$

$$\tilde{s}^{\mathrm{a}} := s + \Delta s^{\mathrm{a}}, \tag{54}$$

$$\tilde{s}^{\mathrm{m}} := s + \Delta s^{\mathrm{m}}. \tag{55}$$

Remark 2.2 Just like Iteration MPC, Iteration rMPC⋆ uses separate step sizes for the primal and dual variables. It has been broadly acknowledged (e.g., p. 195 in [33]) that the use of separate sizes has computational advantages.

Remark 2.3 While rMPC⋆ as stated fails to retain the remarkable scaling invariance properties of MPC, invariance under diagonal scaling in the primal space and under Euclidean transformations and uniform diagonal scaling in the dual space can be readily recovered (without affecting the theoretical properties of the algorithm) by modifying iteration rMPC⋆ along lines similar to those discussed in Sect. 5 of [29].

In closing this section, we note a few immediate results to be used in the sequel. First, the following identities are valid for $j \in \{\mathrm{a}, \mathrm{m}\}$:

$$t_{\mathrm{p}}^{j} = \min\left\{1, \min\left\{\frac{x_i}{-\Delta x_i^{j}} \,\middle|\, i \in Q,\ \Delta x_i^{j} < 0\right\}\right\}, \tag{56}$$

$$t_{\mathrm{d}}^{j} = \min\left\{1, \min\left\{\frac{s_i}{-\Delta s_i^{j}} \,\middle|\, \Delta s_i^{j} < 0\right\}\right\}. \tag{57}$$

Next, the following are direct consequences of (32)–(33) and Steps 1 and 3 of Iteration rMPC*:

$$\Delta s^{j} = -A^{\mathrm{T}} \Delta y^{j} \quad \text{for } j \in \{\mathrm{a}, \mathrm{c}, \mathrm{m}\}, \tag{58}$$

and, for $i \in Q$,

$$s_i \Delta x_i^{\mathrm{a}} + x_i \Delta s_i^{\mathrm{a}} = -x_i s_i, \tag{59}$$

$$\frac{s_i}{-\Delta s_i^{\mathrm{a}}} = \frac{x_i}{\tilde{x}_i^{\mathrm{a}}} \quad \text{when } \Delta s_i^{\mathrm{a}} \neq 0 \quad \text{and} \quad \frac{x_i}{-\Delta x_i^{\mathrm{a}}} = \frac{s_i}{\tilde{s}_i^{\mathrm{a}}} \quad \text{when } \Delta x_i^{\mathrm{a}} \neq 0, \tag{60}$$

$$s_i \Delta x_i^{\mathrm{m}} + x_i \Delta s_i^{\mathrm{m}} = -x_i s_i + \gamma(\sigma \mu_Q - \Delta x_i^{\mathrm{a}} \Delta s_i^{\mathrm{a}}), \tag{61}$$

and, as a direct consequence of (58) and (46),

$$A^{\mathrm{T}} y^{+} + s^{+} = c. \tag{62}$$

Further, system (32) can alternatively be solved in augmented system form

$$\begin{pmatrix} A_Q & 0 \\ S_Q & -X_Q A_Q^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \Delta x_Q^{\mathrm{a}} \\ \Delta y^{\mathrm{a}} \end{pmatrix} = \begin{pmatrix} b - A_Q x_Q \\ -X_Q s_Q \end{pmatrix}, \qquad \Delta s_Q^{\mathrm{a}} = -A_Q^{\mathrm{T}} \Delta y^{\mathrm{a}}, \tag{63}$$

or in normal equations form

$$A_Q S_Q^{-1} X_Q A_Q^{\mathrm{T}} \Delta y^{\mathrm{a}} = b, \tag{64a}$$

$$\Delta s_Q^{\mathrm{a}} = -A_Q^{\mathrm{T}} \Delta y^{\mathrm{a}}, \tag{64b}$$

$$\Delta x_Q^{\mathrm{a}} = -x_Q - S_Q^{-1} X_Q \Delta s_Q^{\mathrm{a}}. \tag{64c}$$

Similarly, (33) can be solved in augmented system form

$$\begin{pmatrix} A_Q & 0 \\ S_Q & -X_Q A_Q^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \Delta x_Q^{\mathrm{c}} \\ \Delta y^{\mathrm{c}} \end{pmatrix} = \begin{pmatrix} 0 \\ \sigma \mu_Q e - \Delta X_Q^{\mathrm{a}} \Delta s_Q^{\mathrm{a}} \end{pmatrix}, \qquad \Delta s_Q^{\mathrm{c}} = -A_Q^{\mathrm{T}} \Delta y^{\mathrm{c}}, \tag{65}$$

or in normal equations form

$$A_Q S_Q^{-1} X_Q A_Q^{\mathrm{T}} \Delta y^{\mathrm{c}} = -A_Q S_Q^{-1} (\sigma \mu_Q - \Delta X_Q^{\mathrm{a}} \Delta s_Q^{\mathrm{a}}), \tag{66a}$$

$$\Delta s^{\mathrm{c}}_Q = -A^{\mathrm{T}}_Q \Delta y^{\mathrm{c}}, \tag{66b}$$

$$\Delta x^{\mathrm{c}}_Q = -S^{-1}_Q X_Q \Delta s^{\mathrm{c}}_Q + S^{-1}_Q (\sigma \mu_Q - \Delta X^{\mathrm{a}}_Q \Delta s^{\mathrm{a}}_Q). \tag{66c}$$

Finally, as an immediate consequence of the definition (41) of the rMPC* search direction in Step 4 of Iteration rMPC* and of the expressions (24) and (26) (in particular (24)), we have

$$\|\gamma \Delta y^{\mathrm{c}}\| \le \psi \|\Delta y^{\mathrm{a}}\|, \qquad \gamma \sigma \mu_Q \le \psi \|\Delta y^{\mathrm{a}}\|. \tag{67}$$

3 Global convergence analysis

The analysis given here is inspired from the line of argument used in [29] for the rPDAS algorithm, but we use less restrictive assumptions. The following proposition, which builds on [29, Prop. 3], shows that Iteration rMPC* can be repeated indefinitely and that the dual objective strictly increases.

Proposition 3.1 *Let $x > 0$, $s > 0$, and $Q \in \mathfrak{Q}_{\epsilon,M}(y)$ for some $y \in \mathbb{R}^m$. Then the following hold*: (*i*) $b^{\mathrm{T}}\Delta y^{\mathrm{a}} > 0$, (*ii*) $b^{\mathrm{T}}\Delta y^{\mathrm{m}} \ge \theta b^{\mathrm{T}} \Delta y^{\mathrm{a}}$, *and* (*iii*) $t^{\mathrm{m}}_{\mathrm{p}} > 0$, $t^{\mathrm{m}}_{\mathrm{d}} > 0$, $y^+ \in F^{\mathrm{o}}$, $s^+ = c - A^{\mathrm{T}} y^+ > 0$, *and* $x^+ > 0$.

Proof Claim (i) follows directly from Lemma 2.2, (64a) and $b \neq 0$, which imply

$$b^{\mathrm{T}}\Delta y^{\mathrm{a}} = b^{\mathrm{T}}(A_Q S^{-1}_Q X_Q A^{\mathrm{T}}_Q)^{-1} b > 0.$$

For claim (ii), if $b^{\mathrm{T}}\Delta y^{\mathrm{c}} \ge 0$, then, by claim (i),

$$b^{\mathrm{T}}\Delta y^{\mathrm{m}} = b^{\mathrm{T}}\Delta y^{\mathrm{a}} + \gamma b^{\mathrm{T}}\Delta y^{\mathrm{c}} \ge b^{\mathrm{T}}\Delta y^{\mathrm{a}} \ge \theta b^{\mathrm{T}}\Delta y^{\mathrm{a}},$$

and from Step 4 of Iteration rMPC*, if $b^{\mathrm{T}}\Delta y^{\mathrm{c}} < 0$ then, using (24) and (26) ($\gamma \le \gamma_1$), (23), and claim (i), we get

$$\begin{aligned} b^{\mathrm{T}}\Delta y^{\mathrm{m}} &\ge b^{\mathrm{T}}\Delta y^{\mathrm{a}} + \gamma_1 b^{\mathrm{T}}\Delta y^{\mathrm{c}} \ge b^{\mathrm{T}}\Delta y^{\mathrm{a}} + (1-\theta)\frac{b^{\mathrm{T}}\Delta y^{\mathrm{a}}}{|b^{\mathrm{T}}\Delta y^{\mathrm{c}}|} b^{\mathrm{T}}\Delta y^{\mathrm{c}} \\ &= b^{\mathrm{T}}\Delta y^{\mathrm{a}} - (1-\theta) b^{\mathrm{T}}\Delta y^{\mathrm{a}} = \theta b^{\mathrm{T}}\Delta y^{\mathrm{a}}. \end{aligned}$$

Finally, claim (iii) follows from Steps 4–5 of Iteration rMPC*. □

It follows from Proposition 3.1 that, under Assumption 2.1, Iteration rMPC* generates an infinite sequence of iterates with monotonically increasing dual objective value. From here on we attach an iteration index k to the iterates.

As a first step, we show that if the sequence $\{y^k\}$ remains bounded (which cannot be guaranteed under our limited assumptions), then it must converge and that it will be unbounded only if $b^{\mathrm{T}} y^k \to \infty$, making the feasible sequence $\{y^k\}$ maximizing. This result is taken from [12, Sect. 2]. Its proof is reproduced here for ease of reference. It makes use of the following lemma, a direct consequence of results in [24] (see also [23]).

Lemma 3.2 *Let A be a full row rank matrix and b be a vector of same dimension as a column of A. Then, (i) there exists $\rho > 0$ (depending only on A and b) such that, given any positive definite diagonal matrix D, the solution Δy to*

$$ADA^{\mathrm{T}}\Delta y = b, \tag{68}$$

satisfies

$$\|\Delta y\| \le \rho b^{\mathrm{T}}\Delta y;$$

and (ii), if a sequence $\{y^k\}$ is such that $\{b^{\mathrm{T}}y^k\}$ is bounded and, for some $\omega > 0$, satisfies

$$\|y^{k+1} - y^k\| \le \omega b^{\mathrm{T}}(y^{k+1} - y^k) \quad \forall k, \tag{69}$$

then $\{y^k\}$ converges.

Proof The first claim immediately follows from Theorem 5 in [24], noting (as in [23], Sect. 4) that, for some $\alpha > 0$, $\alpha\Delta y$ solves

$$\max\{b^{\mathrm{T}}u \mid u^{\mathrm{T}}ADA^{\mathrm{T}}u \le 1\}.$$

(See also Theorem 7 in [23].) The second claim is proved using the central argument of the proof of Theorem 9 in [24]:

$$\sum_{k=0}^{N-1}\|y^{k+1} - y^k\| \le \omega\sum_{k=0}^{N-1} b^{\mathrm{T}}(y^{k+1} - y^k) = \omega b^{\mathrm{T}}(y^N - y^0) \le 2\omega v \quad \forall N > 0,$$

where v is an upper bound to $\{b^{\mathrm{T}}y^k\}$, implying that $\{y^k\}$ is Cauchy, and thus converges. (See also Theorem 9 in [23].) □

Lemma 3.3 *Suppose Assumptions 2.1 and 2.2 hold. Then, if $\{y^k\}$ is bounded then $y^k \to y^*$ for some $y^* \in F$, and if it is not, then $b^{\mathrm{T}}y^k \to \infty$.*

Proof We first show that $\{y^k\}$ satisfies (69) for some $\omega > 0$. In view of (43), (45), and (46), it suffices to show that, for some $\omega > 0$,

$$\|\Delta y^{\mathrm{m},k}\| \le \omega b^{\mathrm{T}}\Delta y^{\mathrm{m},k} \quad \forall k.$$

Now, since $\Delta y^{\mathrm{a},k}$ solves (64a) and Q takes only finitely many values, the hypothesis of Lemma 3.2 (i) is validated for $\Delta y^{\mathrm{a},k}$, and thus, for some $\rho > 0$,

$$\|\Delta y^{\mathrm{a},k}\| \le \rho b^{\mathrm{T}}\Delta y^{\mathrm{a},k} \quad \forall k.$$

With this in hand, we obtain, for all k, using (41), (28), (24), and (22),

$$\begin{aligned}\|\Delta y^{\mathrm{m},k}\| &\le \|\Delta y^{\mathrm{a},k}\| + \gamma^k\|\Delta y^{\mathrm{c},k}\| \le (1+\psi)\|\Delta y^{\mathrm{a},k}\| \\ &\le (1+\psi)\rho b^{\mathrm{T}}\Delta y^{\mathrm{a},k} \le (1+\psi)\frac{\rho}{\theta}b^{\mathrm{T}}\Delta y^{\mathrm{m},k},\end{aligned}$$

so (69) holds with $\omega := (1+\psi)\frac{\rho}{\theta}$.

Now first suppose that $\{y^k\}$ is bounded. Then so is $\{b^{\mathrm{T}}y^k\}$ and, in view of Lemma 3.2 (ii) and the fact that $\{y^k\}$ is feasible, we have $y^k \to y^*$, for some $y^* \in F$. On the other hand, if $\{y^k\}$ is unbounded, then $\{b^{\mathrm{T}}y^k\}$ is also unbounded (since, in view of Lemma 3.2 (ii), having $\{b^{\mathrm{T}}y^k\}$ bounded together with (69) would lead to the contradiction that the unbounded sequence $\{y^k\}$ converges). □

We also have that the primal iterates remain bounded.

Lemma 3.4 *Suppose Assumption 2.1 holds. Then* $\{x^k\}$, $\{\tilde{x}^{\mathrm{a},k}\}$, *and* $\{\tilde{x}^{\mathrm{m},k}\}$ *are all bounded.*

Proof We first show that $\{\tilde{x}^{\mathrm{a},k}\}$ is bounded. Defining $D^k_{Q^k} := X^k_{Q^k}(S^k_{Q^k})^{-1}$ and using (64a)–(64b) we have $\Delta s^{\mathrm{a},k} = -A^{\mathrm{T}}_{Q^k}(A_{Q^k}D^k_{Q^k}A^{\mathrm{T}}_{Q^k})^{-1}b$, which, using definition (47) of $\tilde{x}^{\mathrm{a},k}_{Q^k}$, and (64c) gives

$$\tilde{x}^{\mathrm{a},k}_{Q^k} = D^k_{Q^k}A^{\mathrm{T}}_{Q^k}(A_{Q^k}D^k_{Q^k}A^{\mathrm{T}}_{Q^k})^{-1}b. \tag{70}$$

Sequences of the form $D^kA^{\mathrm{T}}(AD^kA^{\mathrm{T}})^{-1}$, with A full rank and D^k diagonal and positive definite for all k, are known to be bounded; a proof can be found in [8].[13] Hence $\|\tilde{x}^{\mathrm{a},k}\| = \|\tilde{x}^{\mathrm{a},k}_{Q^k}\| \le R$ with R independent of k (there are only finitely many choices of Q^k). Finally, boundedness of $\{\tilde{x}^{\mathrm{m},k}\}$ and $\{x^k\}$ is proved by induction as follows. Let R' be such that $\max\{\|x^k\|_\infty, (1+\psi)R, \chi, \underline{\xi}^{\max}\} < R'$, for some k. From (53), (41), (26) ($\gamma \le \gamma_0$), (24), and (47), we have

$$\begin{aligned}\|\tilde{x}^{\mathrm{m},k}\| &= \|\tilde{x}^{\mathrm{m},k}_{Q^k}\| = \|x^k_Q + \Delta x^{\mathrm{a},k}_{Q^k} + \gamma\Delta x^{\mathrm{c},k}_{Q^k}\| \\ &\le \|\tilde{x}^{\mathrm{a},k}_{Q^k}\| + \psi\|\tilde{x}^{\mathrm{a},k}_{Q^k}\| \le (1+\psi)R \le R',\end{aligned} \tag{71}$$

and since, as per (42), (44) and (46), $\hat{x}^k_{Q^k}$ is on the line segment between $x^k_{Q^k}$ and the full step $\tilde{x}^{\mathrm{m},k}_{Q^k}$, both of which are bounded in norm by R', we have $\|x^{k+1}_{Q^k}\|_\infty \le \max\{\|\hat{x}^k_{Q^k}\|_\infty, \underline{\xi}^{\max}\} \le R'$. On the other hand, the update (52) for the $\mathbf{n}\setminus Q^k$ components of x^{k+1}, ensures that

$$\|x^{k+1}_{\mathbf{n}\setminus Q^k}\|_\infty \le \chi \le R',$$

and the result follows by induction. □

The global convergence analysis essentially considers two possibilities: either $\Delta y^{\mathrm{a},k} \to 0$ or $\Delta y^{\mathrm{a},k} \not\to 0$. In the former case $y^k \to y^* \in F^{\mathrm{s}}$, which follows from the next lemma. In the latter case, Lemma 3.6 (the proof of which uses the full power of Lemma 3.5) and Lemma 3.7 show that $y^k \to y^* \in F^*$.

[13] An English version of the (algebraic) proof of [8] can be found in [31]; see also [24]. Stewart [27] obtained this result in the form of a bound on the norm of oblique projectors, and provided an independent, geometric proof. O'Leary [21] later proved that Stewart's bound is sharp.

Lemma 3.5 *For all k, $A\tilde{x}^{\mathrm{a},k} = b$ and $A\tilde{x}^{\mathrm{m},k} = b$. Further, if Assumption 2.1 holds and $\Delta y^{\mathrm{a},k} \to 0$ on an infinite index set K, then for all j, $\tilde{x}_j^{\mathrm{a},k} s_j^k \to 0$ and $\tilde{x}_j^{\mathrm{m},k} s_j^k \to 0$, both on K. If, in addition, $\{y^k\}$ is bounded, then $y^k \to y^* \in F^{\mathrm{s}}$ and all limit points of the bounded sequences $\{\tilde{x}^{\mathrm{a},k}\}_{k\in K}$ and $\{\tilde{x}^{\mathrm{m},k}\}_{k\in K}$ are multipliers associated to the stationary point y^*.*[14]

Proof The first claim is a direct consequence of the second block equations of (32) and (33), (41), and definitions (47), and (53). Next, we prove asymptotic complementarity of $\{(\tilde{x}^{\mathrm{a},k}, s^k)\}_{k\in K}$. Using the third block equation in (32) and, again, using (47) we have, for all k,

$$\tilde{x}_j^{\mathrm{a},k} s_j^k = -x_j^k \Delta s_j^{\mathrm{a},k}, \quad j \in Q^k, \tag{72}$$

$$\tilde{x}_j^{\mathrm{a},k} s_j^k = 0, \quad j \in \mathbf{n} \setminus Q^k. \tag{73}$$

Since x^k is bounded (Lemma 3.4), and $\Delta s^{\mathrm{a},k} = -A^{\mathrm{T}} \Delta y^{\mathrm{a},k} \to 0$ on K, this implies $\tilde{x}_j^{\mathrm{a},k} s_j^k \to 0$ on K for all j. We can similarly prove asymptotic complementarity of $\{(\tilde{x}^{\mathrm{m},k}, s^k)\}_{k\in K}$. Equations (61) and (53) yield, for all k,

$$s_j^k \tilde{x}_j^{\mathrm{m},k} = -x_j^k \Delta s_j^{\mathrm{m},k} + \gamma^k (\sigma^k \mu_{Q^k}^k - \Delta x_j^{\mathrm{a},k} \Delta s_j^{\mathrm{a},k}), \quad j \in Q^k, \tag{74}$$

$$s_j^k \tilde{x}_j^{\mathrm{m},k} = 0, \quad j \in \mathbf{n} \setminus Q^k. \tag{75}$$

Boundedness of $\{\tilde{x}^{\mathrm{a},k}\}_{k\in K}$ and $\{x^k\}$ (Lemma 3.4) implies boundedness of $\{\Delta x_{Q^k}^{\mathrm{a},k}\}_{k\in K}$ since $\Delta x_{Q^k}^{\mathrm{a},k} = \tilde{x}_{Q^k}^{\mathrm{a},k} - x_{Q^k}^k$. In addition, $\Delta y^{\mathrm{a},k} \to 0$ on K and (67) imply that $\gamma^k \Delta y^{\mathrm{c},k} \to 0$ on K and $\gamma^k \sigma^k \mu_{Q^k}^k \to 0$ on K. The former implies in turn that $\gamma^k \Delta s^{\mathrm{c},k} = -\gamma^k A^{\mathrm{T}} \Delta y^{\mathrm{c},k} \to 0$ on K by (58). Thus, in view of (41), $\{\Delta s^{\mathrm{m},k}\}_{k\in K}$ and the entire right-hand side of (74) converge to zero on K. Asymptotic complementarity then follows from boundedness of $\{\tilde{x}^{\mathrm{m},k}\}$ (Lemma 3.4). Finally, the last claim follows from the above and from Lemma 3.3. □

Recall the definition $\phi^k := \|\Delta y^{\mathrm{a},k}\|^\nu + \|[\tilde{x}^{\mathrm{a},k}]_-\|^\nu$ from (48). The next two lemmas outline some properties of this quantity, in particular, that small values of ϕ^k indicate nearness to dual optimal points.

Lemma 3.6 *Suppose Assumption 2.1 holds. If $\{y^k\}$ is bounded and $\liminf \phi^k = 0$, then $y^k \to y^* \in F^*$.*

Proof By definition (48) of ϕ^k, its convergence to zero on some infinite index set K implies that $\Delta y^{\mathrm{a},k} \to 0$ and $[\tilde{x}^{\mathrm{a},k}]_- \to 0$ on K. Lemma 3.5 and $\{[\tilde{x}^{\mathrm{a},k}]_-\}_{k\in K} \to 0$ thus imply that $y^k \to y^* \in F^*$. □

[14] Such "multipliers" are defined below (8).

« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 21/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

Lemma 3.7 *Suppose Assumptions 2.1 and 2.2 hold and* $\{y^k\}$ *is bounded. If* $\Delta y^{\mathrm{a},k} \not\to 0$, *then* $\liminf \phi^k = 0$. *Specifically, for any infinite index set* K *on which* $\inf_{k\in K} \|\Delta y^{\mathrm{a},k}\| > 0$, *we have* $\phi^{k-1} \to 0$ *on* K.

Proof In view of Lemma 3.3, $y^k \to y^*$ for some $y^* \in F$. Now we proceed by contradiction. Thus, suppose there exists an infinite set $K' \subseteq K$ on which $\|\Delta y^{\mathrm{a},k}\|$ and ϕ^{k-1} are both bounded away from zero. Let us also suppose, without loss of generality, that Q^k is constant on K', say equal to some fixed Q. Lemma 2.3 then guarantees that $I(y^*) \subseteq Q$. Note that, since the rule for selecting Q ensures that A_Q has full rank and, as per (32), $\Delta s^{\mathrm{a},k} = -A_Q^{\mathrm{T}} \Delta y^{\mathrm{a},k}$, we have that $\|\Delta s^{\mathrm{a},k}\|$ is also bounded away from zero on K'. Define $\delta_1 := \inf_{k\in K'} \|\Delta s^{\mathrm{a},k}\|^2 > 0$. In view of (49), the fact that ϕ^{k-1} is bounded away from zero for $k \in K'$ implies that $\delta_2 := \inf\{x_i^k \mid i \in Q^k,\ k \in K'\} > 0$. We now note that, by Step 5 of rMPC* and Proposition 3.1 (ii), for all $k \in K'$,

$$b^{\mathrm{T}} y^{k+1} = b^{\mathrm{T}}(y^k + t_{\mathrm{d}}^{\mathrm{m},k} \Delta y^{\mathrm{m},k}) \ge b^{\mathrm{T}} y^k + t_{\mathrm{d}}^{\mathrm{m},k} \theta b^{\mathrm{T}} \Delta y^{\mathrm{a},k}. \tag{76}$$

Also, from (64a) and (58), we have for all $k \in K'$,

$$b^{\mathrm{T}} \Delta y^{\mathrm{a},k} = (\Delta y^{\mathrm{a},k})^{\mathrm{T}} A_Q (S_Q^k)^{-1} X_Q^k A_Q^{\mathrm{T}} \Delta y^{\mathrm{a},k} = (\Delta s_Q^{\mathrm{a},k})^{\mathrm{T}} (S_Q^k)^{-1} X_Q^k \Delta s_Q^{\mathrm{a},k} \ge \frac{\delta_2}{R} \delta_1 > 0,$$

where R is an upper bound on $\{\|s^k\|_\infty\}_{k\in K'}$ (boundedness of s^k follows from (62)). In view of (76), establishing a positive lower bound on $t_{\mathrm{d}}^{\mathrm{m},k}$ for $k \in K'$ will contradict boundedness of $\{y^k\}$, thereby completing the proof.

By (45) and since Step 4 of Iteration rMPC* ensures (25), we have $t_{\mathrm{d}}^{\mathrm{m},k} \ge \beta \bar{t}_{\mathrm{d}}^{\mathrm{m},k} \ge \beta \zeta t_{\mathrm{d}}^{\mathrm{a},k} \ge 0$. Therefore, it suffices to bound $t_{\mathrm{d}}^{\mathrm{a},k}$ away from zero. From (38), either $t_{\mathrm{d}}^{\mathrm{a},k} = 1$ or, for some i_0 such that $\Delta s_{i_0}^{\mathrm{a},k} < 0$ (without loss of generality we assume such i_0 is independent of $k \in K'$) we have

$$t_{\mathrm{d}}^{\mathrm{a},k} = \frac{s_{i_0}^k}{-\Delta s_{i_0}^{\mathrm{a},k}}. \tag{77}$$

If $i_0 \in \mathbf{n} \setminus Q$ then it is a consequence of Rule 2.1 that $\{s_{i_0}^k\}_{k\in K'}$ is bounded away from zero. In this case, the desired positive lower bound for $t_{\mathrm{d}}^{\mathrm{a},k}$ follows if we can show that $\Delta s^{\mathrm{a},k}$ is bounded on K'. To see that the latter holds, we first note that, since A_Q^{T} is full column rank, it has left inverse $(A_Q A_Q^{\mathrm{T}})^{-1} A_Q$, so that (64b) implies

$$\Delta y^{\mathrm{a},k} = -(A_Q A_Q^{\mathrm{T}})^{-1} A_Q \Delta s_Q^{\mathrm{a},k},$$

and, using (58),

$$\Delta s^{\mathrm{a},k} = A^{\mathrm{T}} (A_Q A_Q^{\mathrm{T}})^{-1} A_Q \Delta s_Q^{\mathrm{a},k}. \tag{78}$$

Finally, using (64c) and (47) to write $\Delta s_Q^{\mathrm{a},k} = -(X_Q^k)^{-1} S_Q^k \tilde{x}_Q^{\mathrm{a},k}$, and substituting in (78) gives

$$\Delta s^{\mathrm{a},k} = -A^{\mathrm{T}} (A_Q A_Q^{\mathrm{T}})^{-1} A_Q (X_Q^k)^{-1} S_Q^k \tilde{x}_Q^{\mathrm{a},k},$$

« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 22/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

which is bounded on K' since $\delta_2 > 0$, s^k is bounded, and $\tilde{x}^{\mathrm{a},k}$ is bounded (by Lemma 3.4). On the other hand, if $i_0 \in Q$, using (77) and (60) we obtain $t_{\mathrm{d}}^{\mathrm{a},k} = x_{i_0}^k/\tilde{x}_{i_0}^{\mathrm{a},k}$, which is bounded away from zero on K' since x_Q^k is bounded away from zero on K' and $\tilde{x}_Q^{\mathrm{a},k}$ is bounded by Lemma 3.4. This completes the proof. □

Theorem 3.8 *Suppose Assumptions 2.1 and 2.2 hold. Then, if $\{y^k\}$ is unbounded, $b^{\mathrm{T}}y^k \to \infty$. On the other hand, if $\{y^k\}$ is bounded, then $y^k \to y^* \in F^{\mathrm{s}}$. Under the further assumption that, at every dual feasible point, the gradients of all active constraints are linearly independent,*[15] *it holds that if F^* is not empty, $\{y^k\}$ converges to some $y^* \in F^*$, while if F^* is empty, $b^{\mathrm{T}}y^k \to \infty$, so that, in both cases, $\{b^{\mathrm{T}}y^k\}$ converges to the optimal dual value.*

Proof The first claim is a direct consequence of Lemma 3.3. Concerning the second claim, under Assumptions 2.1 and 2.2, the hypothesis of either Lemma 3.5 or Lemma 3.7 must hold: $\{\Delta y^{\mathrm{a},k}\}$ either converges to zero or it does not. In the latter case, the second claim follows from Lemmas 3.7 and 3.6 since $F^* \subseteq F^{\mathrm{s}}$. In the former case, it follows from Lemma 3.5. To prove the last claim, it is sufficient to show that, under the stated linear independence assumption, it cannot be the case that $\{y^k\}$ converges to some $y^* \in F^{\mathrm{s}} \setminus F^*$. Indeed, the first two claims will then imply that either $y^k \to F^*$, which cannot occur when F^* is empty, or $b^{\mathrm{T}}y^k \to \infty$, which can only occur if F^* is empty, proving the claim. Now, proceeding by contradiction, suppose that $y^k \to y^* \in F^{\mathrm{s}} \setminus F^*$. It then follows from Lemma 3.7 that $\Delta y^{\mathrm{a},k} \to 0$, since, with $y^* \notin F^*$, Lemma 3.6 implies that $\liminf \phi^k > 0$. Lemma 3.5 then implies that $S^k\tilde{x}^{\mathrm{a},k} \to 0$, and $A\tilde{x}^{\mathrm{a},k} = b$. Define $J := \{j \in \mathbf{n} \mid \tilde{x}_j^{\mathrm{a},k} \not\to 0\}$. Since $S^k\tilde{x}^{\mathrm{a},k} \to 0$, and since $s^k = c - A^{\mathrm{T}}y^k \to c - A^{\mathrm{T}}y^*$, we have that $s_J^k \to 0$, i.e., $J \subseteq I(y^*)$. Thus, by Lemma 2.3, $J \subseteq I(y^*) \subseteq Q^k$ holds for all k sufficiently large. Then, using the second block equation of (32) and (47), we can write

$$b = A_{Q^k}\tilde{x}_{Q^k}^{\mathrm{a},k} = A\tilde{x}^{\mathrm{a},k} = A_J\tilde{x}_J^{\mathrm{a},k} + A_{\mathbf{n}\setminus J}\tilde{x}_{\mathbf{n}\setminus J}^{\mathrm{a},k}, \tag{79}$$

where, by definition of J, the second term in the right hand side converges to zero. Under the linear independence assumption, since $J \subseteq I(y^*)$, A_J must have linearly independent columns and a left inverse given by $(A_J^{\mathrm{T}}A_J)^{-1}A_J^{\mathrm{T}}$. Thus, using (79), we have $\tilde{x}_J^{\mathrm{a},k} \to (A_J^{\mathrm{T}}A_J)^{-1}A_J^{\mathrm{T}}b$. Define $\tilde{x}^*$ by $\tilde{x}_J^* := (A_J^{\mathrm{T}}A_J)^{-1}A_J^{\mathrm{T}}b$ and $\tilde{x}_{\mathbf{n}\setminus J}^* := 0$, so that $\tilde{x}^{\mathrm{a},k} \to \tilde{x}^*$. Since $y^* \notin F^*$, $\tilde{x}_{j_0}^* < 0$ for some $j_0 \in J$, and $\tilde{x}_{j_0}^{\mathrm{a},k} < 0$ holds for all k sufficiently large, which implies that $s_{j_0}^k \to 0$. However, from (60), for all k large enough,

$$\Delta s_{j_0}^{\mathrm{a},k} = -\frac{s_{j_0}}{x_{j_0}}\tilde{x}_{j_0}^{\mathrm{a},k} > 0,$$

so that, by (46), $s_{j_0}^{k+1} > s_{j_0}^k > 0$ holds for all k large enough, which contradicts $s_{j_0}^k \to 0$. □

[15] This additional assumption is equivalent to the assumption that "all dual basic feasible solutions are nondegenerate" commonly used in convergence analyses of affine scaling and simplex algorithms, e.g., [1].

« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 23/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

Whether $y^k \to y^* \in F^*$ is guaranteed (when F^* is nonempty) without the linear independence assumption is an open question.

Remark 3.1 While a fairly standard assumption, the linear independence condition used in Theorem 3.8 to prove convergence to a dual optimal point, admittedly, is rather strong, and may be difficult to verify a priori. We remark here that, in view of the monotonic increase of $b^{\mathrm{T}} y^k$ and of the finiteness of the set $\{b^{\mathrm{T}} y : y \in F^{\mathrm{s}} \setminus F^*\}$, convergence to a dual optimal point should occur *without* such assumption if the iterates are subject to perturbations (say, due to roundoff) assumed to be uniformly distributed over a small ball. Indeed, suppose that y^k converges to $y^* \in F^{\mathrm{s}} \setminus F^*$, say, with limit dual value equal to v. There exists $\alpha > 0$ such that, for every k large enough, the *computed* y^k will satisfy $b^T y^k > v$ with probability at least α, so that this will happen for *some* k with probability one, and monotonicity of $b^{\mathrm{T}} y^k$ would then rule out v as a limit value. Of course, again due to perturbations, $b^T y^k$ could drop below v again at some later iteration. This however can be addressed by the following simple modification of the algorithm. Whenever the *computed* y^{k+1} satisfies $b^T y^{k+1} < b^T y^k$, discard such y^{k+1}, compute $\Delta y^{\mathrm{p}}(y^k, Q)$ by solving an appropriate auxiliary problem, such as the small dimension LP

$$\max\{b^{\mathrm{T}} \Delta y^{\mathrm{p}} \mid A_{Q^k}^{\mathrm{T}} \Delta y^{\mathrm{p}} \le s_{Q^k} := c_{Q^k} - A_{Q^k}^{\mathrm{T}} y^k,\ \|\Delta y^{\mathrm{p}}\|_\infty \le 1\}, \tag{80}$$

where $Q \in \mathfrak{Q}_{\epsilon,M}(y^k)$, and redefine y^{k+1} to be the point produced by a long step (close to the largest feasible step) taken from y^k in direction $\Delta y^{\mathrm{p}}(y^k, Q)$. It is readily shown that the solution $\Delta y^{\mathrm{p}}(y^k, Q)$ provides a feasible step that gives uniform ascent near any $y^k \in F^{\mathrm{s}} \setminus F^*$. Note that, in "normal" operation, stopping criterion (81) (see Sect. 5 below) will be satisfied before any decrease in $b^T y^k$ due to roundoff is observed, and the suggested restoration step will never be used.

Finally, the following convergence properties of the primal sequence can be inferred whenever $\{y^k\}$ converges to $y^* \in F^*$, without further assumptions.

Proposition 3.9 *Suppose that Assumptions 2.1 and 2.2 hold and that $y^k \to y^* \in F^*$. Then, there exists an infinite index set K on which $\Delta y^{\mathrm{a},k} \to 0$ and $\{\tilde{x}^{\mathrm{a},k}\}_{k\in K}$ and $\{\tilde{x}^{\mathrm{m},k}\}_{k\in K}$ converge to the primal optimal set.*

Proof Suppose $\Delta y^{\mathrm{a},k} \not\to 0$. Then, by Lemma 3.7, we have $\liminf \phi^k = 0$, which implies $\liminf \Delta y^{\mathrm{a},k} = 0$, a contradiction. Thus, $\Delta y^{\mathrm{a},k} \to 0$ on some K, and the rest of the claim follows from Lemma 3.5. □

4 Local convergence

The next assumption is justified by Remark 3.1 at the end of the previous section.

Assumption 4.1 $y^k \to y^* \in F^*$.

Our final assumption, Assumption 4.2 below, supersedes Assumption 2.2. Under this additional assumption, the iteration sequence $\{z^k\} := \{(x^k, y^k)\}$ converges q-quadratically to the unique primal-dual solution $z^* := (x^*, y^*)$. (Uniqueness of x^* follows from Assumption 4.2.) The details of the analysis are deferred to the appendix, available as Electronic Supplementary Material.

Assumption 4.2 *The dual solution set is a singleton, i.e.,* $F^* = \{y^*\}$, *and* $\{a_i : i \in I(y^*)\}$ *is a linearly independent set.*

Theorem 4.1 *Suppose Assumptions 2.1, 4.1 and 4.2 hold. Then the iteration sequence* $\{z^k\}$ *converges locally q-quadratically, i.e.,* $z^k \to z^*$ *and there exists* $c^* > 0$ *such that, for all k large enough, we have*

$$\|z^{k+1} - z^*\| \le c^* \|z^k - z^*\|^2.$$

Further $\{(t_{\mathrm{p}}^{\mathrm{m}})^k\}$ *and* $\{(t_{\mathrm{d}}^{\mathrm{m}})^k\}$ *both converge to* 1. *Finally, for k large enough, the rank condition in Rule 2.1 is automatically satisfied.*

5 Numerical experiments

5.1 Implementation

Algorithm rMPC* was implemented in Matlab and run on an Intel(R) Pentium(R) Centrino Duo 1.73 GHz Laptop machine with 2 GB RAM, Linux kernel 2.6.31 and Matlab 7 (R14). To compute the search directions (35) we solved the normal equations (64) and (66), using Matlab's Cholesky factorization routine. Parameters for rMPC* were chosen as $\beta := 0.95$, $\theta := 0.1$, $\psi := 10^9$, $\zeta := 0.3$, $\lambda := 3$, $\nu := 3$, $\chi := 10^9$, and $\underline{\xi}^{\max} := 10^{-11}$, and for each problem discussed below, we assume that a small upper bound M on the number of active constraints is available, and so we always take $\epsilon := \infty$. The code was supplied with strictly feasible initial dual points (i.e., $y^0 \in F^{\mathrm{o}}$), and we set $x^0 := e$.

We used a stopping criterion adapted from [18, p. 592], based on normalized primal and dual infeasibilities and duality gap. Specifically, convergence was declared when

$$\texttt{termcrit} := \max\left\{\frac{\|c - A^{\mathrm{T}}y - s\|}{1+\|s\|}, \frac{\|b - Ax\|}{1+\|x\|}, \frac{\|[s]_-\|}{1+\|s\|}, \frac{\|[x]_-\|}{1+\|x\|}, \frac{|c^{\mathrm{T}}x - b^{\mathrm{T}}y|}{1+|b^{\mathrm{T}}y|}\right\} < \texttt{tol}, \tag{81}$$

where `tol` was set to 10^{-8}. (While the first, third, and fourth terms in the max are zero for rMPC*, they are useful for Sect. 5.4.)

Our analysis assumes Q is selected according to the *general* Rule 2.1, i.e., that $Q \in \mathfrak{Q}_{\epsilon,M}(y)$ at each iteration. To complete the description of a *specific* rule for constraint selection, we simply need to specify any additional constraints that are to be included in Q, particularly so that rank condition in Rule 2.1 holds. A simple way to deal with the rank condition is the following. At each iteration, set Q to be a set of M most active constraints, form the normal matrix and attempt to factor it. If the rank condition fails, then the standard Cholesky factorization will fail. At this point simply add the next M most active constraints to Q, and repeat the factorization attempt with $|Q| = 2M$. If it still fails, increase $|Q|$ to $4M$ by adding the next most active constraints, etc. (On the next iteration we revert to using only M constraints.) We refer to this technique as the "doubling" method.

One alternative to the doubling method is to augment the dual problem with bound constraints on the y variables, i.e., $-\pi e \le y \le \pi e$ for some scalar $\pi > 0$, and always include these constraints in Q in addition to the M most active ones. This ensures that the rank condition holds, while adding negligible additional work (since the associated constraint vectors are sparse). Furthermore, in practice, π can be chosen large enough so that these constraints are never active at the solution. Both the doubling method and this "bounding" method were used in our tests as indicated below.

A third possibility would be to use instead a pivoted Cholesky algorithm that will compute the factor of a nearby matrix [14], regardless of Q. If the rank condition fails, the factor can be (efficiently) updated by including additional constraints [10, Sect. 12.5], chosen according to slack value or otherwise, until the estimated condition number [10, p. 129] is acceptably small.

We refer to this rule that uses only the M most active constraints, doubling or bounding if needed, as the "Most Active Rule". While simple, the Most Active Rule does not always provide great performance on its own, and it may be desirable to keep additional constraints in Q to boost performance. In the sequel we describe some possible methods for selecting additional constraints, and in Sect. 5.3 below, we give a detailed example of how a heuristic may be developed and tailored to a particular class of problems. We emphasize, however, that the primary purpose of this paper is not to investigate constraint selection heuristics, but rather to provide a general convergence analysis under minimally restrictive selection rules.

5.2 Randomly generated problems

As a first test, following [29], we randomly generated a sequence of unbalanced (standard form) linear programs of size $n = m^2$ for increasing values of m, by taking $A,\ b,\ y_0 \sim \mathcal{N}(0,1)$ and then normalizing the columns of A. We set $s_0 \sim \mathcal{U}(0,1)$ (uniformly distributed in $(0,1)$) and $c := A^{\mathrm{T}} y_0 + s_0$ which guarantees that the initial iterate is strictly dual feasible. The iteration was initialized with this (s_0, y_0) and $x_0 := e$. This problem is called the "fully random" problem in [29], where a different x_0 is used.

On this problem class the columns of A are in general linear position (every $m \times m$ submatrix of A is nonsingular), so that any $M \ge m$ is valid, and the doubling (or bounding) technique is never needed. Here, it turns out, constraint reduction works extremely well with the simple Most Active Rule as long as M is chosen slightly larger than m. Figure 3 shows the average time and iterations to solve 100 instances each of the size $m = 25$ to $m = 200$ problem and 10 instances of the size $m = 400$ problem ($n = m^2$ in all cases) using the Most Active Rule. The points on the plots correspond to different runs on the same problem. The runs differ in the number of

« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 26/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

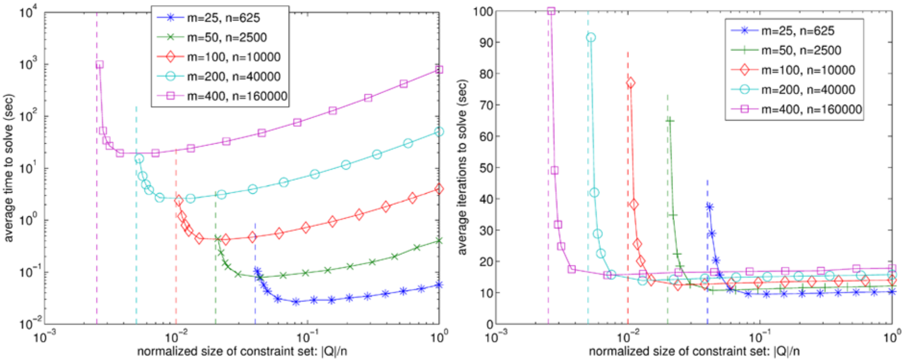

Fig. 3 Performance of constraint reduction on the random problems using the Most Active Rule. The plots show the average time and iterations to solve 100 instances each of the size $m = 25$ to $m = 200$ problem and 10 instances of the size $m = 400$ problem ($n = m^2$ in all cases). Each problem is solved using 14 different values of M ranging from m to n. The *dashed vertical asymptotes* correspond to choosing $M = m$, the lower theoretical and (apparently) practical limit on the size of the constraint set

constraints M that are retained in Q, which is indicated on the horizontal axis as a fraction of the full constraint set (i.e., M/n is plotted). Thus, the rightmost point corresponds to the experiment without constraint reduction, while the points on the extreme left correspond to the most drastic constraint reduction. In the left plot, the vertical axis indicates, for each value of the abscissa, total CPU time to successful termination, as returned by the Matlab function `cputime`, while the right plot shows the total number of iterations to successful termination. The vertical dotted lines correspond to choosing $M = m$, the lower limit for convergence to be guaranteed, and the plots show that this is also a practical limit, as the time and iterations grow rapidly as M is decreased toward m.

While the random problem has a large amount of redundancy in the constraint set, this may not always be the case, and in general we may not know a priori how many constraints should be kept. We also expect, intuitively, that fewer constraints will be needed as the algorithm nears the solution and the partition into active/inactive constraints becomes better resolved. Thus we would like to find rules that let the algorithm *adaptively* choose how many constraints it keeps at each iteration, i.e., that allow the *cardinality* of the working set to change from iteration to iteration. As an initial stride towards this end, we consider associating a scalar value v_i to each constraint for $i \in \mathbf{n}$. A large value of v_i indicates that we believe keeping constraint i in Q will improve the search direction and a small value means we believe it will not help or possibly will do harm. In addition to the M constraints selected according to the Most Active Rule, we add up to M' constraints that have $v_i \geq 1$, selecting them in order of largest value v_i first. We refer to this rule as the Adaptive Rule. We propose two specific variants of this rule. In the first variant, we set

$$v_i = \eta \min\{s_j\}/s_i,$$

i.e., add additional constraints that have a slack value smaller than a fixed multiple $\eta > 1$ of the minimum slack. In the second variant, we set

$$v_i = \eta \sqrt{\frac{x_i/s_i}{\max\{x_j/s_j\}}},$$

i.e., we add the ith constraint if $\sqrt{x_i/s_i}$ is within a fixed multiple $1/\eta$ of the maximum value of $\sqrt{x_j/s_j}$.[16] This rule combines information from both the primal and dual variables with regard to "activity" of this constraint. Note also that this v_i is the (scaled, square root of the) "coefficient" of the ith constraint in the normal matrix sum (7); thus we could interpret this rule as trying to keep the error between the reduced and unreduced normal matrix small. In view of (64a), we may expect that constraints with small values of $\sqrt{x_i/s_i}$ do not play much of a role in the construction of Δy^{m}.

Figure 4 shows the results of using the second variant of the Adaptive Rule on our random LP. We set $M = 2m$ and plot $(M + M')/n$ on the horizontal axis. Note that, when $\eta = 10$, the average (over an optimization run) time per iteration increases very slowly as the upper bound $M + M'$ on $|Q|$ increases, starting from the lower bound $M = 2m$. (Indeed, the right plot shows that the total number of iterations remains roughly constant.) This means that the average size of $|Q|$ (over a run) itself increases very slowly, i.e., that $|Q|$ departs little from its minimum value $2m$ in the course of a run. If η is increased to 1000, the average value of $|Q|$ increases, which means more variation of $|Q|$ in the course of a run (since $|Q|$ is close to M at the end of the runs: see below); this is the intended behavior. The general behavior of these rules is that in early iterations the v_i are spread out and, with large η, many will be larger than the threshold value of one. Thus, the iteration usually starts out using $M + M'$ constraints, the upper bound. As the solution is approached, all v_i's tend to zero except those corresponding to active constraints which go to infinity (the second variant needs strict complementarity for this), thus in later iterations only the M most active constraints (the lower bound) will be included in Q. We have observed that this transition from $M + M'$ to M constraints occurs rather abruptly usually over the course of just a few iterations; the choice of η serves to advance or delay this transition. In summary, the Adaptive Rule, like the Most Active Rule, keeps the number of iterations approximately constant over a wide range of choices of $M + M'$, but unlike the Most Active Rule, the time is also approximately constant remaining much less than that for MPC.

We could think of many variations of the Adaptive Rule. Here we have only considered rules that choose v as a function of the current iterate, whereas we expect that by allowing v to depend on the entire history of iterates and incorporating more prior knowledge concerning the problem structure, etc., better constraint reduction heuristics could be developed. We believe that designing good adaptive rules will be a key to successful and robust application of constraint reduction; we largely leave this for future work.

[16] The square root allows the use of similar magnitude η for both variants.

« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 28/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

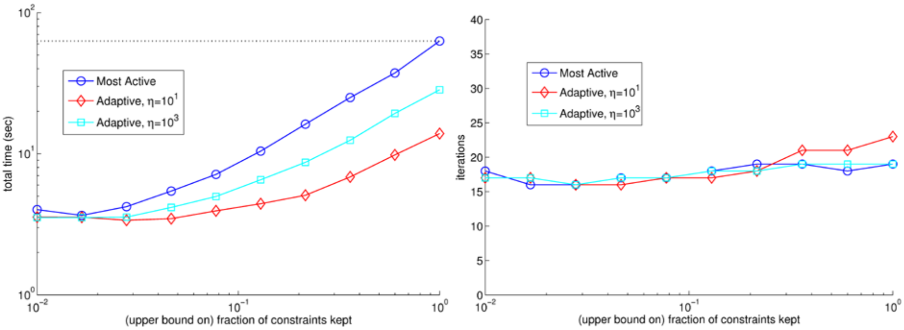

Fig. 4 Adaptive Rule, second variant with $M = 2m$ and $\eta = 10^1, 10^3, \infty$ (setting $\eta = \infty$ corresponds to the Most Active Rule) on the size $m = 200$, $n = 40000$ random problem, horizontal axis on log scale. Here the horizontal axis represents $M + M'$, the upper bound on the size of the constraint set. The *horizontal dotted black line in the left plot* marks the performance of the unreduced MPC algorithm

5.3 Discrete Chebyshev approximation problems

Here we investigate a "real-world" application, fitting a linear model to a target vector by minimizing the infinity norm of the residual, viz.

$$\min_u \|Hu - g\|_\infty,$$

where g is the target vector, H is the model matrix, and u is the vector of model parameters. This can be formulated as a linear program in standard dual form

$$\max\{-t \mid Hu - g \le te,\ -Hu + g \le te\}. \tag{82}$$

If H has dimension $p \times q$, then the "A matrix" of this LP has dimension $m \times n$ with $m = q + 1$ and $n = 2p$ so that, if $p \gg q$ (as is typical), then $n \gg m$. Dual strictly feasible points are readily available for this problem; we used the dual-feasible point $u_0 = 0$ and $t_0 = \|g\|_\infty + 1$ to initialize the algorithm.

As a specific test, we took $p = 20000$ equally spaced samples of the smooth function

$$g_0(t) = \sin(10t)\cos(25t^2), \quad t \in [0, 1] \tag{83}$$

and stacked them in the p-dimensional vector g. For the columns of H, we took the $q = 199$ lowest frequency elements of the discrete Fourier transform (DFT) basis. When converted to (82), this resulted in a $m \times n$ linear program with $m = 200$ and $n = 40000$. For this problem, we circumvented the rank condition of Rule 2.1 by adding the bound constraints $-10^3 \le y \le 10^3$ (for a total of 40400 constraints) and always including them in Q.

The initial results were poor: using the basic Most Active Rule with $M = 20m$ and an additional $3m$ randomly selected constraints, rMPC* required over 500 iterations to solve the problem to 10^{-8} accuracy. Numerical evidence suggests that there are two distinct issues here; the first causes slow convergence in the initial phase of the iteration, reducing `termcrit` (see (81)) to around 10^{-2}, and the second causes slow convergence in the later phase of the iteration, further reducing `termcrit` to 10^{-8}.

The first issue is that since, for fixed y, the slack "function" $c - A^{\mathrm{T}}y$ is "smooth"[17] with respect to its index, the most nearly active constraints are all clustered into a few groups of contiguous indices corresponding to the minimal modes of the slack function. Intuitively, this does not give a good description of the feasible set, and furthermore, since the columns of A are also smooth in the index, A_Q is likely to be rank deficient, or nearly so, when only the most active constraints are included in Q, i.e., for the Most Active Rule. This clustering appears to cause slow convergence in the initial phase. This problem can in large part be avoided by adding a small random sample of constraints to Q: vastly improved performance is gained, especially in the initial phase.

The second issue, which persists even after adding random constraints, is that Q is missing certain constraints that appear to be critical in the later phase, namely the local minimizers (strict or not) of the slack function $s(i) := c_i - a_i^{\mathrm{T}} y$. The omission of these constraints results in very slow convergence in the later phase of the iteration. For example, we ran rMPC* using $M = 3m$ and adding $10m$ random constraints and observed that `termcrit` was reduced below 10^{-2} in 90 iterations, but that another 247 iterations were needed to achieve `termcrit`$< 10^{-8}$. Strikingly, in 88% of these later iterations, the blocking constraint, i.e., the one which limited the line search, was a local minimizer of the slack function not included in Q. If we instead used $M = m$ and again $10m$ random constraints, this happened in nearly 100% of the later iterations.

In light of these observations, we devised a simple heuristic for this class of smooth Chebyshev problems: use a small M, a small number of random constraints and add the local minimizers of the slack function in Q (it is enough to keep those local minimizers with slack value less than, say, half of the maximum slack value). Note that in this case the size of the constraint set is not fixed a priori nor upper bounded—however since the target vector g and the basis elements have relatively low frequency content, adding the local minimizers generally added only a few (always fewer than m) extra constraints at each iteration.

Additional observations led to further refinement of this heuristic. First, we noted that the random constraints only seem to help in the early iterations and actually seem to slow convergence in the later iterations, so we considered gradually phasing them out as the iteration approached optimality. Second, we noted that in place of a random sample of constraints we could instead include all constraints from a regular grid of the form $\{i, i+j, i+2j, \ldots, i+(k-1)j\} \subseteq \mathbf{n}$ for some integers i, j, k with $i \in \{1, 2, \ldots, j\}$, and $jk = n$.

Table 1 displays the performance of these various rules on the discrete Chebyshev approximation problem discussed above. The left side of the table describes the rule used: columns MA, RND and GRD give the number of most active, random, and gridded constraints respectively, and columns LM and COOL indicate whether the local minimizers of the slack function are included and whether the random constraints are phased out or "cooled" as the iteration nears optimality. The right side

[17] This is due to the fact that the 20 kHz sampling frequency (if t has units of seconds) is much higher than the highest significant frequency present in the "signal" $\{g_0(t) \mid t \in [0, 1]\}$, whose periodogram is 60 dB below its peak value at 50 Hz.

Table 1 Results of various heuristics on the Chebyshev approximation problem

Rule description					Performance			
MA	RND	GRD	LM	COOL	cputime	it:10^{-2}	it:10^{-8}	avg. $\lvert Q^k\rvert$
n	0	0	No	–	105.8 s	13	31	40400.0
$13m$	0	0	No	–	382.9 s	758	947	2849.8
$3m$	$10m$	0	No	No	180.2 s	82	492	2570.0
$1m$	$10m$	0	Yes	No	14.5 s	17	41	2307.8
$1m$	$10m$	0	Yes	Yes	9.0 s	21	36	1027.4
$1m$	0	$2m$	Yes	–	9.5 s	26	41	745.7

gives the performance of the corresponding rule: the first column lists the CPU time needed to reduce `termcrit` below 10^{-8}, the next two columns give the number of iterations needed to reduce `termcrit` below 10^{-2} and 10^{-8} respectively, and the last column gives the average size of the constraint set during the iteration. The first row of the table describes the unreduced MPC and gives a baseline performance level. The second row again illustrates the failure of the Most Active Rule, while the third and fourth show that adding randomly selected constraints and the local minimizers of the slack function effectively deals with the issues described above. The fifth and sixth rows show enhancements of the specialized rule that achieve a 10-fold speed up over unreduced MPC.

Numerical experiments on other discrete Chebyshev approximation problems (arising from practical applications at NASA) are discussed in [32].

5.4 Comparison with other algorithms

Finally, we made a brief comparison of rMPC* vs. related algorithms (constraint-reduced or not) and summarize the results in Tables 2 and 3. We implemented the rMPC algorithm as described in [29], both using a feasible starting point (listed as `rmpc` in the tables) and using the typically infeasible starting point described in [18] (`rmpcinf`), as well as a variant of rPDAS (`rpdas`) from [29] obtained by setting $\psi = 0$ in rMPC*. We also implemented the "active-set" potential-reduction algorithm of [30] (`rdpr`), and a long-step variant `budas-ls` of the short-step build-up affine-scaling algorithm analyzed in [4] (we followed the suggestion ($\alpha > 1$) made at the end of Sect. 1 of that paper). Finally, we compared the performance of Zhang's `lipsol` (the interior-point solver used in MATLAB's `linprog` routine) as a benchmark for our test problems without using constraint reduction.

The algorithms were implemented in MATLAB, all using stopping criterion (81) with `tol` $= 10^{-8}$. For `rpdas` we used the same implementation and parameters as rMPC*, but with $\psi = 0$. In the implementation of `rmpc` and `rmpcinf`, as prescribed in [29], we used a stepsize rule that moves the iterates a fixed fraction 0.95 (parameter β in [29]) of the way to the boundary of the interior region $\{(x, s) | x > 0, s > 0\}$. In our implementation of `budas` we used parameters $\beta = 0.95$, and we replaced the finite termination scheme with our termination criterion (81). For `rdpr` we used $\alpha = 0.5$, $\delta^* = 0.05$, and $\rho = 5n$ (the latter attempts to get good practical performance although is not as good theoretically as $\rho = n + \nu\sqrt{n}$ for a constant $\nu > 1$). We also removed the finite termination scheme for `rdpr`, and since `rdpr` requires an upper bound on the dual optimal value, we used the optimal value obtained by rMPC* and added 10. Finally, again in `rdpr`, we used an Armijo line search (with slope parameter $\alpha_{\text{armijo}} = 0.1$ and step size parameter $\beta_{\text{armijo}} = 0.5$) in place of the exact minimizing line search.[18]

For test problems we chose an instance of the 200×40000 random problem described in Sect. 5.2 (`rand`), the Chebyshev approximation problem described in Sect. 5.3 (`cheb`), three problems from the original netlib collection [9] (`scsd1`, `scsd6`, and `scsd8`) with $n \gg m$, and two problems from the Mittelman collection [19] (`rail507` and `rail582`). (We note that the `scsd` problems are of less interest for applying constraint reduction because, as compared to our other test problems, they are of small dimension, less unbalanced, and very sparse which means the cost of forming the normal matrix is much less than $\mathcal{O}(nm^2)$. The rail problems are also sparse, but much larger.) We choose initial iterates for the (`cheb`) and (`rand`) as described in Sects. 5.2 and 5.3 and, for the `scsd` problems, we used a vector of zeros (dual strictly feasible) as the initial dual iterate and a vector of ones as the initial primal iterate. For each of these problems, we ran each algorithm, first using no reduction as a benchmark, then, for all but `lipsol`, using a common constraint reduction rule. (Note that all tested algorithms allow for a heuristic constraint selection rule.) The constraint selection rules used in the test were as follows. First, as before, we always set $\epsilon = \infty$. Then, for the random problem we used $M = 2m$ most active constraints and no additional constraints, for the Chebyshev problem we used the rule corresponding to the last row of Table 1, and for the `scsd` problems we used $M = 3m$ most active constraints. Finally, the remaining constraints were sorted by increasing slack value, and in the case of numerical issues solving the linear systems (in particular if the rank condition of Rule 2.1 failed), or if the step was not acceptable, i.e., infeasible in the case of `budas` or did not achieve required decrease in the potential function for `rdpr`, the constraint set was augmented with $|Q|$ additional constraints, where $|Q|$ refers to the original size of the constraint set, and the step was recomputed.

The results for the unreduced and reduced cases are shown in Tables 2 and 3, respectively. The columns of each table are, in order: the problem name (`prob`), the algorithm (`alg`), the final status (`status`), the total running time (`time`), the number of iterations (`iter`), and finally the maximum (`Mmax`) and average (`Mavg`) number of constraints used at each iteration. If any algorithm took more than 200 iterations, we declared the `status` to be a `fail`, and set the time and iteration counts to `Inf`.

In general, the best performance is obtained using rMPC*. In the unreduced case, the other MPC-type algorithms LIPSOL and rMPC[19] showed similar iteration counts

[18]In the comparison below, we use $\alpha_{\text{armijo}} = 0.1$ in the line search of `rdpr` in all cases except in the unreduced case on problem `rail582`, where we increased $\alpha_{\text{armijo}} = 0.4$. This was done because `rdpr` failed to achieve the required descent in its potential using $\alpha_{\text{armijo}} = 0.1$ on that problem, and thus halted.

[19]There are some differences between rMPC and rMPC* that could explain the variation in iteration counts, the most significant of which is probably in the modified choice of centering parameter σ in rMPC* (cf. (14) and (40)).

Table 2 Comparison of algorithms with no constraint reduction

Prob	Alg	Status	Time	Iter	M max	M avg
cheb	rmpc*	succ	97.74	29	40400	40400.0
cheb	rmpc	succ	132.59	41	40400	40400.0
cheb	rmpcinf	succ	139.83	41	40400	40400.0
cheb	pdas	fail	Inf	Inf	40400	40400.0
cheb	rdpr	succ	351.47	110	40400	40400.0
cheb	budas-ls	fail	Inf	Inf	40400	40400.0
cheb	lipsol	succ	998.72	53	40400	40400.0
rail507	rmpc*	succ	37.90	51	63516	63516.0
rail507	rmpc	succ	28.63	43	63516	63516.0
rail507	rmpcinf	succ	32.83	47	63516	63516.0
rail507	pdas	succ	35.47	58	63516	63516.0
rail507	rdpr	succ	99.15	116	63516	63516.0
rail507	budas-ls	succ	48.29	64	63516	63516.0
rail507	lipsol	succ	29.48	53	63516	63516.0
rail582	rmpc*	succ	42.69	58	56097	56097.0
rail582	rmpc	succ	32.89	49	56097	56097.0
rail582	rmpcinf	succ	36.29	52	56097	56097.0
rail582	pdas	succ	39.13	65	56097	56097.0
rail582	rdpr	succ	120.77	125	56097	56097.0
rail582	budas-ls	succ	44.02	53	56097	56097.0
rail582	lipsol	succ	29.50	55	56097	56097.0
rand	rmpc*	succ	60.40	18	40000	40000.0
rand	rmpc	succ	57.99	18	40000	40000.0
rand	rmpcinf	succ	58.37	17	40000	40000.0
rand	pdas	succ	69.28	22	40000	40000.0
rand	rdpr	succ	249.33	77	40000	40000.0
rand	budas-ls	succ	110.20	33	40000	40000.0
rand	lipsol	succ	377.42	19	40000	40000.0
scsd1	rmpc*	succ	0.11	10	760	760.0
scsd1	rmpc	succ	0.12	12	760	760.0
scsd1	rmpcinf	succ	0.10	11	760	760.0
scsd1	pdas	succ	0.08	9	760	760.0
scsd1	rdpr	succ	0.78	64	760	760.0
scsd1	budas-ls	succ	0.15	17	760	760.0
scsd1	lipsol	succ	0.08	10	760	760.0
scsd6	rmpc*	succ	0.21	12	1350	1350.0
scsd6	rmpc	succ	0.21	13	1350	1350.0
scsd6	rmpcinf	succ	0.23	13	1350	1350.0
scsd6	pdas	succ	0.21	14	1350	1350.0
scsd6	rdpr	succ	0.96	56	1350	1350.0
scsd6	budas-ls	succ	0.27	20	1350	1350.0
scsd6	lipsol	succ	0.13	12	1350	1350.0
scsd8	rmpc*	succ	0.55	10	2750	2750.0
scsd8	rmpc	succ	0.62	12	2750	2750.0
scsd8	rmpcinf	succ	0.71	13	2750	2750.0
scsd8	pdas	succ	0.70	14	2750	2750.0
scsd8	rdpr	succ	1.75	57	2750	2750.0
scsd8	budas-ls	succ	0.60	21	2750	2750.0
scsd8	lipsol	succ	0.22	11	2750	2750.0

Table 3 Comparison of algorithms with constraint reduction

Prob	Alg	Status	Time	Iter	M max	M avg
cheb	rmpc*	succ	14.07	50	1184	1128.5
cheb	rmpc	fail	Inf	Inf	1180	1168.3
cheb	rmpcinf	fail	Inf	Inf	1189	1019.8
cheb	pdas	fail	Inf	Inf	15636	2760.4
cheb	rdpr	fail	Inf	Inf	2771	1038.8
cheb	budas-ls	succ	16.62	57	3278	1667.1
cheb	lipsol	–	–	–	–	–
rail507	rmpc*	succ	6.98	24	2535	2535.0
rail507	rmpc	fail	Inf	Inf	2535	2535.0
rail507	rmpcinf	fail	Inf	Inf	2535	2535.0
rail507	pdas	succ	21.31	84	63516	3261.0
rail507	rdpr	fail	Inf	Inf	2535	2535.0
rail507	budas-ls	succ	10.78	31	63516	5892.3
rail507	lipsol	–	–	–	–	–
rail582	rmpc*	succ	9.62	27	56097	4879.9
rail582	rmpc	fail	Inf	Inf	2910	2910.0
rail582	rmpcinf	fail	Inf	Inf	2910	2910.0
rail582	pdas	succ	33.29	127	11640	2978.7
rail582	rdpr	fail	Inf	Inf	2910	2910.0
rail582	budas-ls	succ	13.65	32	46560	7365.9
rail582	lipsol	–	–	–	–	–
rand	rmpc*	succ	3.54	17	400	400.0
rand	rmpc	succ	12.94	34	400	400.0
rand	rmpcinf	fail	Inf	Inf	400	400.0
rand	pdas	succ	8.59	51	400	400.0
rand	rdpr	succ	21.23	63	400	400.0
rand	budas-ls	succ	3.70	19	3200	589.5
rand	lipsol	–	–	–	–	–
scsd1	rmpc*	succ	0.07	9	231	231.0
scsd1	rmpc	succ	0.11	13	231	231.0
scsd1	rmpcinf	succ	0.15	15	760	266.3
scsd1	pdas	succ	0.08	13	231	231.0
scsd1	rdpr	succ	0.53	71	760	264.5
scsd1	budas-ls	succ	0.07	16	231	231.0
scsd1	lipsol	–	–	–	–	–
scsd6	rmpc*	succ	0.14	11	441	441.0
scsd6	rmpc	succ	0.19	14	441	441.0
scsd6	rmpcinf	succ	0.22	18	441	441.0
scsd6	pdas	succ	0.17	15	441	441.0
scsd6	rdpr	succ	0.75	68	882	447.5
scsd6	budas-ls	succ	0.12	18	441	441.0
scsd6	lipsol	–	–	–	–	–
scsd8	rmpc*	succ	0.47	10	1191	1191.0
scsd8	rmpc	succ	0.61	12	1191	1191.0
scsd8	rmpcinf	succ	0.81	15	2750	1294.9
scsd8	pdas	succ	0.56	13	1191	1191.0
scsd8	rdpr	succ	2.38	51	2382	1284.4
scsd8	budas-ls	succ	0.70	18	1191	1191.0
scsd8	lipsol	–	–	–	–	–

to rMPC* both from the feasible and infeasible starting point. Note that LIPSOL was much slower in terms of running-time on the dense problems, likely because of overhead in its sparse linear algebra. The MPC-based algorithms were generally superior to the alternatives, which is to be expected as the primal-dual MPC is generally regarded as practically superior to PDAS and dual only algorithms. In the constraint reduced case rMPC* performed significantly better than rMPC (perhaps indicating practical value of the safeguards developed in this paper) and also outperformed rPDAS (`rpdas`) and the potential reduction algorithm (`rdpr`), while the long-step variant of the Dantzig-Ye algorithm (`budas-ls`)[20] was nearly on par with rMPC* under constraint reduction.

The improvement in running time for algorithms rMPC* and `budas-ls` when using constraint reduction versus no reduction is quite significant on all but the small, sparse `scsd` problems. Remarkably, for these two algorithms, on the `rail507` problem the number of iterations is also much smaller with constraint reduction than without.

6 Conclusions

We have proposed a variant of Mehrotra's Predictor-Corrector algorithm, rMPC*, designed to efficiently solve standard form linear programs (1) where A is $m \times n$ with $n \gg m$. Specifically, rMPC* uses MPC-like search directions computed for "constraint-reduced" versions of the problem; see (31). The cost of an iteration of rMPC* can be much less than that of an iteration of MPC; specifically, the high order work when solving the normal equations by direct methods with dense A is reduced from $\mathcal{O}(nm^2)$ (the cost of forming the normal matrix) to $\mathcal{O}(|Q|m^2)$ where the theory allows $|Q|$ to be as small as m in some cases. The primary contribution of this paper is the global and local quadratic convergence of the algorithm under a very general class of constraint selection rules. The analysis has similarities to that in [29] for the rPDAS algorithm, but the constraint selection rule here is more general and the nondegeneracy assumptions here are less restrictive. The results extend to constrained-reduced primal-dual affine scaling as a limit case, thus improving on the results of [29]. As a further special case, they apply to unreduced primal-dual affine scaling.

Using various constraint selection heuristics, we demonstrated the effectiveness of the proposed algorithm on a class of random problems where the performance was remarkably good. In fact on these problems it appears that we can use constraint reduction without penalty: the iteration counts were the same whether we used the entire constraint set or only the 1% most nearly active constraints, and computation times were reduced dramatically. We also observed remarkable numerical behavior of rMPC* on a class of discrete Chebyshev approximation problems after the introduction of a specialized rule for constraint selection tailored to this class of problems. Finally, in a brief comparison against other constraint reduced IPMs, rMPC* performed favorably.

[20] For the random problem, `budas-ls` had to use minor-cycles to increase the constraint set size to 3200, 800, and 800, respectively, in its first three iterations and used $2m = 400$ in the remaining iterations. On the Chebyshev approximation problem in 17 of the 57 iterations, minor cycle iterations were used that each effectively doubled the constraint size.

Acknowledgement The authors wish to thank Pierre-Antoine Absil for many helpful discussions.

References

1. Bertsimas, D., Tsitsiklis, J.: Introduction to Linear Optimization, Athena (1997)
2. Cartis, C.: Some disadvantages of a Mehrotra-type primal-dual corrector interior point algorithm for linear programming. Appl. Numer. Math. **59**, 1110–1119 (2009)
3. Colombo, M., Gondzio, J.: Further development of multiple centrality correctors for interior point methods. Comput. Optim. Appl. **41**(3), 277–305 (2008)
4. Dantzig, G., Ye, Y.: A build-up interior-point method for linear programming: affine scaling form. Working paper, Department of Management Science, University of Iowa, 1991
5. den Hertog, D., Roos, C., Terlaky, T.: Adding and deleting constraints in the path-following method for LP. In: Du, D.Z., Sun, J. (eds.) Advances in Optimization and Approximation, pp. 166–185. Kluwer Academic, Norwell (1994)
6. Deza, A., Nematollahi, E., Peyghami, R., Terlaky, T.: The central path visits all the vertices of the Klee-Minty cube. Optim. Methods Softw. **21**(5), 851–865 (2006)
7. Dikin, I.I.: Iterative solution of problems of linear and quadratic programming. Dokl. Akad. Nauk SSSR **174**, 747–748 (1967). English translation: Sov. Math. Dokl. **8**, 674–675 (1967)
8. Dikin, I.I.: O skhodimosti odnogo iteratsionnogo protsessa (On the convergence of an iterative process). Upr. Sist. **12**, 54–60 (1974), in Russian
9. Gay, D.M.: Electronic mail distribution of linear programming test problems. Math. Program. Soc. COAL Newslett. **13**, 10–12 (1985). http://www-fp.mcs.anl.gov/OTC/Guide/TestProblems/LPtest/
10. Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press, London (1996)
11. Gondzio, J.: Matrix-free interior point method. Comput. Optim. Appl. (2010). doi:10.1007/s10589-010-9361-3
12. He, M.Y., Tits, A.L.: Infeasible constraint-reduced interior-point methods for linear optimization (2010). Submitted for possible publication in Optim. Methods Softw., special issue in honor of Florian A. Potra's 60-th birthday. www.optimization-online.org/DB_HTML/2010/11/2822.html
13. Herskovits, J.N.: Développement d'une méthode numérique pour l'optimization non-linéaire. Ph.D. thesis, Université Paris IX—Dauphine, Paris, France, January 1982
14. Higham, N.J.: Analysis of the Cholesky decomposition of a semi-definite matrix. In: Cox, M.G., Hammarling, S.J. (eds.) Reliable Numerical Computation, pp. 161–185. Oxford University Press, London (1990)
15. Jansen, B., Roos, C., Terlaky, T.: A polynomial primal-dual Dikin-type algorithm for linear programming. Math. Oper. Res. **21**(2), 341–353 (1996)
16. Kaliski, J.A., Ye, Y.: A decomposition variant of the potential reduction algorithm for linear programming. Manag. Sci. **39**, 757–776 (1993)
17. Mangasarian, O.: A Newton method for linear programming. J. Optim. Theory Appl. **121**(1), 1–18 (2004)
18. Mehrotra, S.: On the implementation of a primal-dual interior point method. SIAM J. Optim. **2**, 575–601 (1992)
19. Mittelmann, H.: LP Models, Miscellaneous LP models collected by Hans D. Mittelmann: online at http://plato.asu.edu/ftp/lptestset/
20. Monteiro, R.D.C., Adler, I., Resende, M.G.C.: A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. Math. Oper. Res. **15**, 191–214 (1990)
21. O'Leary, D.P.: On bounds for scaled projections and pseudo-inverses. Linear Algebra Appl. **132**, 115–117 (1990)
22. Panier, E.R., Tits, A.L., Herskovits, J.N.: A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. SIAM J. Control Optim. **26**(4), 788–811 (1988)
23. Saigal, R.: On the primal-dual affine scaling method. Tech. report, Dept. of Industrial and Operational Engineering, The University of Michigan, 1994
24. Saigal, R.: A simple proof of a primal affine scaling method. Ann. Oper. Res. **62**, 303–324 (1996)
25. Salahi, M., Terlaky, T.: Postponing the choice of the barrier parameter in Mehrotra-type predictor-corrector algorithms. Eur. J. Oper. Res. **182**, 502–513 (2007)

« COAP 10589 layout: Small Extended v.1.2 file: coap9389.tex (Loreta) class: spr-small-v1.1 v.2010/12/15 Prn:2011/01/13; 15:45 p. 36/36»
« doctopic: OriginalPaper numbering style: ContentOnly reference style: mathphys»

26. Salahi, M., Peng, J., Terlaky, T.: On Mehrotra-type predictor-corrector algorithms. SIAM J. Optim. **18**(4), 1377–1397 (2007)
27. Stewart, G.W.: On scaled projections and pseudo-inverses. Linear Algebra Appl. **112**, 189–194 (1989)
28. Tits, A.L., Zhou, J.L.: A simple, quadratically convergent algorithm for linear and convex quadratic programming. In: Hager, W.W., Hearn, D.W., Pardalos, P.M. (eds.) Large Scale Optimization: State of the Art, pp. 411–427. Kluwer Academic, Norwell (1994)
29. Tits, A.L., Absil, P.A., Woessner, W.: Constraint reduction for linear programs with many constraints. SIAM J. Optim. **17**(1), 119–146 (2006)
30. Tone, K.: An active-set strategy in an interior point method for linear programming. Math. Program. **59**, 345–360 (1993)
31. Vanderbei, R.J., Lagarias, J.C.: I.I. Dikin's convergence result for the affine-scaling algorithm. In: Lagarias, J.C., Todd, M.J. (eds.) Mathematical Developments Arising from Linear Programming: Proceedings of a Joint Summer Research Conference (Bowdoin College, Brunswick, Maine, USA), pp. 109–119. American Mathematical Society, Providence (1988)
32. Winternitz, L.B.: Primal-dual interior-point algorithms for linear programming problems with many inequality constraints. Ph.D. thesis, Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, May 2010. http://drum.lib.umd.edu/handle/1903/10400
33. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM, Philadelphia (1997)
34. Ye, Y.: An $O(n^3L)$ potential reduction algorithm for linear programming. Math. Program. **50**(2), 239–258 (1991)
35. Zhang, Y., Zhang, D.: On polynomiality of the Mehrotra-type predictor-corrector interior point algorithms. Math. Program. **68**, 303–31 (1995)
36. Zhang, D., Zhang, Y.: A Mehrotra-type predictor-corrector algorithm with polynomiality and Q-subquadratic convergence. Ann. Oper. Res. **62**(1), 131–150 (1996)