RE-OPTIMIZATION WITH THE PRIMAL-DUAL INTERIOR POINT METHOD*

JACEK GONDZIO † AND ANDREAS GROTHEY ‡

JULY 27, 2001, REVISED JUNE 5, 2002 AND JULY 31, 2002

Abstract. Re-optimization techniques for an interior point method applied to solve a sequence of linear programming problems are discussed. Conditions are given for problem perturbations that can be absorbed in merely one Newton step. The analysis is performed for both short-step and long-step feasible path-following method. A practical procedure is then derived for an infeasible path-following method. It is applied in the context of *crash* start for several large-scale structured linear programs. Numerical results with OOPS, a new object-oriented parallel solver, demonstrate the efficiency of the approach. For large structured linear programs crash start leads to about 40% reduction of the iterations number and translates into 25% reduction of the solution time. The crash procedure parallelizes well and speed-ups between 3.1-3.8 on 4 processors are achieved.

Key words. interior point methods, warm-start, crash start

AMS subject classifications 90C51, 90C06

1. Introduction. A number of optimization algorithms require solving a sequence of linear programs. This is a common situation, for example, in the cutting plane methods [13], decomposition [6, 4], branch-and-bound and branch-and-cut approaches for mixed integer optimization [24] and many others. The problems in such a sequence are often similar, i.e. the following instance is only a minor perturbation of the earlier one. Hence the optimal solution of the earlier problem (or, more generally, a close-to-optimality solution of it with some desirable properties) should be a good starting point for the following problem.

Interior point methods are reputed to have difficulties when they are (naively) applied in this context. Indeed, the efficiency of practical interior point algorithm critically depends on the ability of the algorithm to stay close to the central path. It is no surprise that after the problem has been perturbed the optimal solution of the earlier problem (that necessarily must have been very close to the boundary of the feasible region) is a very bad starting point for a new problem with a different optimal partition. The difficulty of re-optimization may be decreased by the choice of a suitable non-optimal point lying in the neighborhood of the central path [9] and re-optimizing from it. Such an approach has an intuitive justification: a suitable non-optimal point is not yet too close to the boundary of the feasible region hence it is able to absorb larger perturbations to the problem.

This approach has already been used in two different classes of re-optimization problems.

In the first one, the size of the problem increases: a set of new constraints or a set of new variables is added to the previous linear program. This is for example the case in a cutting plane method (or a column generation method) widely used in combinatorial optimization or branch-and-cut approaches to integer programming. The advantages of the application of interior point methods in this context have

*Supported by the Engineering and Physical Sciences Research Council of UK, EPSRC grant GR/M68169. Accepted for publication in **SIAM Journal on Optimization.**

†School of Mathematics, University of Edinburgh, King's Buildings, EH9 3JZ, Scotland (gondzio@maths.ed.ac.uk).

‡School of Mathematics, University of Edinburgh, King's Buildings, EH9 3JZ, Scotland (agr@maths.ed.ac.uk).

been recognized by Mitchell and Todd [17, 19] who were the first to try practical re-optimization procedures in the primal projective algorithm. An implementation of the warm starting technique for the infeasible primal-dual method applied to solve restricted master problems in the cutting plane scheme was described in [9]. The reader interested in the use of interior point methods to solve combinatorial optimization problems should consult [18] and the references therein.

In the second important class of problems that need re-optimization the size of the problem does not change but the parameters such as the coefficient matrix, the right-hand-side, the objective function and/or the variable bounds do change. This is the case for example when subproblems in Dantzig-Wolfe decomposition [6] are solved (the objective of the linear program changes), when subproblems in Benders decomposition [4] are solved (the right-hand-side vector of the linear program changes), or when a variable has its bound tightened in the branch-and-bound technique (this again results in the perturbation of the right-hand-side vector). Preliminary results of applying an interior point-based re-optimization procedure for solving subproblems in the decomposition of large-scale structured linear programs have been reported in [11] where a straightforward extension of the method of [9] has been used.

A similar re-optimization strategy to the one proposed here has been analysed recently by Yıldırım and Wright [27]. In their approach, all intermediate iterates (approximate μ-centers) are stored. Once the perturbation to the problem is known one of the earlier stored μ-centers is chosen. This point is supposed to be sufficiently far away from the optimal solution to absorb the perturbation of the problem and to restore feasibility in only one Newton step. Yıldırım and Wright derive bounds on the size of absorbable perturbations. Their bounds depend on the problem size and the problem condition number. Two different condition numbers are used, the one of Nunez and Freund [20] and the one that follows from Dikin [7]. Unfortunately, none of these numbers can be easily computed. Therefore it is not obvious how to use these results in practice.

The approach proposed in this paper is different. We introduce a new relative measure of perturbations. The perturbations in the primal and dual spaces are compared with the primal and dual slack variables corresponding to a given starting point, respectively. We derive bounds on the largest possible perturbations that can be absorbed by a given approximate μ-center without significantly affecting its proximity measures hence allowing easy continuation of the path-following algorithm. We discuss the two cases of short-step and long-step methods including a recent result on $\mathcal{O}(\sqrt{n}\log n\log(1/\epsilon))$ complexity of the latter due to Peng, Roos and Terlaky [21].

We are aware of the gap between the theory and practice. The former provides complexity estimates and necessarily relies on the worst-case analysis. The approach proposed in this paper can be used in practice. Our relative measure of perturbations needs very little effort to be computed. It can thus be determined for a list of candidate starting points and used to choose the most suitable one. We also discuss how our theoretical developments made for feasible path-following method can be applied in the infeasible method known to be the most efficient interior point method in practice [14, 3].

One of the difficulties in the implementation of interior point methods is the choice of the starting point, cf. [3] and the references therein. Most implementations of interior point methods use some variation of Mehrotra's starting point [16]. Although for the self-dual embedding, which is believed to be less sensitive, any starting point is acceptable, there is still an issue of finding a good initial point [2]. In this paper

we will only consider the standard primal-dual interior point method.

Unlike the simplex method that can take advantage of an advanced starting basis [5, 12, 15], the interior point method is known to be unable of doing so. It is common to consider warm-starting interior point methods from an approximate μ-center. However, re-optimization techniques are also of interest if an advanced starting point is known not from a previous solve of a similar problem, but for example by a crash procedure. The issues involved are the same but the situation is more challenging since the advanced starting points are generally not μ-centers (or even close to one).

In the second part of this paper we study this problem in the context of interior point method applied to solve very large structured linear programs and we provide evidence that advanced *crash* starting points can be constructed for them. Our crashing procedure relies on decomposition but in general constructs infeasible and not necessarily well centered starting points. We use the re-optimization procedure presented in this paper to start the primal-dual algorithm from such points. Numerical results obtained with OOPS, the object-oriented parallel solver [10] confirm that our crash routine can save up to 30-40% of iterations compared with the standard starting point.

The paper is organized as follows. In Section 2 we briefly state the problem, recall some known facts about the worst-case complexity of the path-following methods and introduce the notation used throughout the paper. In Section 3 we derive bounds on the largest perturbations (primal and dual infeasibilities) that can be absorbed by a well-centered point in one Newton step. We show that the proximity measure of the updated point is worsened only slightly so the path-following method may continue from this point without affecting the worst-case complexity result. In Section 4 we translate our findings into computational practice. In particular, we discuss how to deal with large perturbations of the problem by gradually taking them into account in the following iterations. We also relax the constraint of maintaining close proximity to the central path: instead, we rely on the use of the multiple centrality corrections technique [8]. In Section 5 we formulate desired properties of good candidates for the starting point in interior point method and discuss how such points can be obtained through the use of decomposition techniques for large structured linear programs. In Section 6 we illustrate our findings with computational results for a number of structured linear programs. One class of problems originates from network optimization [10] and the other is a well-studied multistage stochastic programming formulation of the asset liability management problem [28]. In Section 7 we give our conclusions.

2. Preliminaries. We consider a primal-dual path-following method for linear programming. The theory for this class of methods has been discussed in detail in the excellent book of Wright [25]. In our developments we shall refer to several results that can be found in this book. In this section we shall deal with the *feasible* method.

Consider a pair of linear programs: the primal

$$\begin{aligned} &\text{minimize} && c_0^T x \\ &\text{subject to} && A_0 x = b_0, \\ & && x \geq 0, \end{aligned} \tag{2.1}$$

where $c_0, x \in \mathcal{R}^n, b_0 \in \mathcal{R}^m$ and $A_0 \in \mathcal{R}^{m \times n}$ has full row rank and its dual

$$\begin{aligned} &\text{maximize} && b_0^T y \\ &\text{subject to} && A_0^T y + s = c_0, \\ & && s \geq 0, \end{aligned} \tag{2.2}$$

where $y \in \mathcal{R}^m$ and $s \in \mathcal{R}^n$. We assume that the feasible sets of the primal and dual problems (2.1) and (2.2) have nonempty interiors

$$\mathcal{F}^0 = \{(x, y, s) | A_0 x = b_0,\ A_0^T y + s = c_0,\ (x, s) > 0\} \neq \emptyset.$$

Hence for any $\mu > 0$, there exists a uniquely defined point $(x(\mu), y(\mu), s(\mu))$, $x(\mu) > 0, s(\mu) > 0$ that satisfies the following first-order optimality conditions for the associated barrier problem

$$\begin{aligned} A_0 x &= b, \\ A_0^T y + s &= c, \\ XSe &= \mu e, \end{aligned} \tag{2.3}$$

where X and S are diagonal matrices with the elements x_j and s_j, respectively, $e \in \mathcal{R}^n$ is the n-vector of all ones and $\mu > 0$ is the barrier parameter. Such a point is called a μ-center.

We assume that a feasible path-following algorithm is used so all its iterates are primal and dual feasible. However, they are not necessarily perfectly centered. We shall consider two neighborhoods of the central path appropriate for the short-step and the long-step algorithms, respectively. The short-step algorithm keeps all its iterates in

$$N_2(\theta) = \{(x, y, s) \in \mathcal{F}^0 \,|\, \|XSe - \mu e\|_2 \leq \theta\mu\}, \tag{2.4}$$

where $0 \leq \theta < 1$. For the long-step algorithm we shall use the following neighborhood

$$N_\infty(\gamma_l, \gamma_u) = \{(x, y, s) \in \mathcal{F}^0 \,|\, \gamma_l\, \mu \leq x_j s_j \leq \gamma_u\, \mu \ \ \forall j\}, \tag{2.5}$$

where $0 < \gamma_l \leq 1 \leq \gamma_u$. The reader should notice that our definition of $N_\infty(\gamma_l, \gamma_u)$ neighborhood is a slight modification of the usual $N_{-\infty}(\gamma)$ neighborhood of [25].

2.1. Complexity Bounds for the Path-following Algorithms. Below we remind the reader the current best complexity results for linear optimization with the path-following algorithm. We recall them in a form that explicitly uses the parameter κ associated with the quality of the initial solution. Assume we seek an ϵ-optimal solution of a linear program and an initial well-centered feasible point is given such that $\mu^0 = (1/\epsilon)^\kappa$. The short-step method finds the ϵ-optimal solution in at most $\mathcal{O}((\kappa + 1)\sqrt{n}\log(1/\epsilon))$ iterations. The classical long-step method finds the ϵ-optimal solution in at most $\mathcal{O}((\kappa+1)n\log(1/\epsilon))$ iterations. Peng, Roos and Terlaky [21] have recently given a new result for a large-update method. Their method performs only $\mathcal{O}((\kappa + 1)\log(1/\epsilon))$ updates of the barrier parameter. However, it requires many so-called inner iterations ($\mathcal{O}(\sqrt{n})$) to restore centrality after the barrier update giving the overall complexity bound of $\mathcal{O}((\kappa + 1)\sqrt{n}\log n\, \log(1/\epsilon))$. It is not obvious that this method should be viewed as a long-step algorithm, but we link our results to it as well.

2.2. Re-optimization Problem. Assume that an approximate μ-center has been found for the primal-dual pair (2.1)-(2.2) and the linear optimization problem has changed. Namely, all its data A_0, b_0 and c_0 has been replaced with the new values A, b and c (where again A is assumed to have full row rank). Unlike in the method of [9] we assume that the size of the linear problem has not changed. We can thus use

the approximate μ-center as an iterate for the new problem. In the new first order conditions all three equations may possibly be violated. Let us define the residuals

$$\begin{bmatrix} \xi_b \\ \xi_c \\ \xi_\mu \end{bmatrix} = \begin{bmatrix} b - Ax \\ c - A^T y - s \\ \mu e - XSe \end{bmatrix}. \tag{2.6}$$

All entries of ξ_μ are of order $\mathcal{O}(\mu)$ because the point (x, y, s) is an approximate μ-center of (2.1)-(2.2). However, primal and dual infeasibilities ξ_b and ξ_c can be arbitrarily large.

We shall express these infeasibilities in the scaling related to the current primal-dual point. In the following section we shall prove sufficient conditions that such scaled perturbations have to satisfy to make re-optimization possible. For a current approximate μ-center (x, y, s) and primal perturbation ξ_b and dual perturbation ξ_c we define the relative residual vectors:

$$\tilde{\xi}_b = X^{-1}A^T(AA^T)^{-1}\xi_b \quad \text{and} \quad \tilde{\xi}_c = S^{-1}\xi_c, \tag{2.7}$$

where X^{-1} and S^{-1} is the usual notation for diagonal $n \times n$ matrices built of elements x_j^{-1} and s_j^{-1}, respectively.

Before we go any further the reader should be warned that one cannot expect to eliminate terms that depend on the problem dimension from the complexity bounds. Instead, following [27], we will concentrate on terms that depend on the quality of the initial point, namely on the parameter κ present in every bound. The aim of our warm-starting procedure is to find a new point $(\bar{x}, \bar{y}, \bar{s})$ that is primal and dual feasible for the perturbed problem and corresponds to a new barrier parameter $\bar{\mu}$ with the value close to μ.

3. Absorbing Primal and Dual Infeasibilities. Assume an approximate μ-center has been found for (2.1)-(2.2) and it is used to compute the Newton direction for the new linear program in which both primal and dual feasibility is violated. Consider the following Newton equation system

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_b \\ \xi_c \\ 0 \end{bmatrix}. \tag{3.1}$$

We draw the reader's attention to the fact that the Newton direction attempts to correct only primal and dual infeasibilities but the complementarity products $x_j s_j$ are not re-centered (although they will obviously change, possibly worsen, once the step is made). A few manipulations give the following Newton direction for dual variables

$$\Delta y = (AXS^{-1}A^T)^{-1}(AXS^{-1}\xi_c + \xi_b), \tag{3.2}$$

and the following Newton direction for primal variables and dual slacks

$$\Delta x = (XS^{-1}A^T(AXS^{-1}A^T)^{-1}AXS^{-1} - XS^{-1})\,\xi_c + XS^{-1}A^T(AXS^{-1}A^T)^{-1}\xi_b, \tag{3.3}$$

$$\Delta s = (I - A^T(AXS^{-1}A^T)^{-1}AXS^{-1})\,\xi_c - A^T(AXS^{-1}A^T)^{-1}\xi_b. \tag{3.4}$$

In our analysis we shall rely on bounds of norms of the following matrix

$$Q = I - S^{-1}A^T(AXS^{-1}A^T)^{-1}AX.$$

LEMMA 3.1. *If* $(x,y,s) \in N_2(\theta)$, *then* $\|Q\|_2 \leq (\frac{1+\theta}{1-\theta})^{1/2}$.

Proof. Since $(x,y,s) \in N_2(\theta)$, we have $(1-\theta)\mu \leq x_j s_j \leq (1+\theta)\mu$ for all $j = 1,2,...,n$. Further we have $\frac{1}{(x_j s_j)^{1/2}} \leq \frac{1}{(1-\theta)^{1/2}\mu^{1/2}}$ and $(x_j s_j)^{1/2} \leq (1+\theta)^{1/2}\mu^{1/2}$. Since

$$Q = X^{-1/2}S^{-1/2}(I - X^{1/2}S^{-1/2}A^T(AXS^{-1}A^T)^{-1}AX^{1/2}S^{-1/2})X^{1/2}S^{1/2},$$

and the matrix in outer parenthesis is an orthogonal projection on the null space of $AX^{1/2}S^{-1/2}$, hence its 2-norm is equal to 1, we can write

$$\|Q\|_2 \leq \|X^{-1/2}S^{-1/2}\|_2 \cdot 1 \cdot \|X^{1/2}S^{1/2}\|_2 \leq \left(\frac{1+\theta}{1-\theta}\right)^{1/2},$$

which completes the proof. □

LEMMA 3.2. *If* $(x,y,s) \in N_\infty(\gamma_l, \gamma_u)$, *then* $\|Q\|_2 \leq (\frac{\gamma_u}{\gamma_l})^{1/2}$.

The proof is omitted because it is very similar to the proof of Lemma 3.1.

Let us observe that for both neighborhoods, the bound on the norm of Q depends only on the constants that define the proximity of the point to the central path. Below we shall also use the ∞-norm of Q. We shall rely on the simple relation $\|Q\|_\infty \leq \sqrt{n}\,\|Q\|_2$ that holds for any square $n \times n$ matrix.

Our re-optimization procedure is divided into two steps. In the first step the total infeasibility is absorbed by making a full step in the Newton direction. In the next step the good quality of the proximity to the central path has to be restored. The second step is needed by the short-step algorithm and by the Peng et al's [21] large-update algorithm but it may be omitted in the large-step path-following algorithm.

We analyse independently the cases of perturbations in the primal and dual spaces and use techniques that are similar to those applied by Yıldırım and Todd [26] to analyse the sensitivity of interior point solutions subject to perturbations in vectors b and c (cf. Propositions 1 and 2 in [26]). One of the key features of the approach presented in this paper is that the primal perturbation ξ_b is used only in the primal direction Δx while the dual perturbation ξ_c is used only in the dual direction $(\Delta y, \Delta s)$. More precisely, our feasibility restoration directions are obtained by substituting $\xi_c = 0$ in (3.3) and $\xi_b = 0$ in (3.2) and (3.4).

Decoupling the primal and dual directions gives the algorithm added flexibility to recover from infeasibilities, particularly in situations where primal and dual infeasibilities differ substantially as is the case when subproblems in Benders or Dantzig-Wolfe decomposition are solved [11] or in our application to crash start (Section 5). Note that decoupling the steps results in only a slight increase in computational cost, since both steps can be obtained using the same Cholesky factors.

3.1. Restoring Dual Feasibility. After setting $\xi_b = 0$ (i.e. ignoring primal infeasibility), from (3.4) and (2.7) we obtain

$$S^{-1}\Delta s = (I - S^{-1}A^T(AXS^{-1}A^T)^{-1}AX)\,S^{-1}\xi_c = Q\,\tilde{\xi}_c. \tag{3.5}$$

We shall now analyse two cases: the short-step and the long-step path-following algorithm.

LEMMA 3.3. *Let* $(x,y,s) \in N_2(\theta)$ *and* $\beta < \sqrt{n}$. *If* $\|\tilde{\xi}_c\|_2 \leq \frac{\beta}{\sqrt{n}} \cdot (\frac{1-\theta}{1+\theta})^{1/2}$, *then the full Newton step in the dual space is feasible and it absorbs the total infeasibility* ξ_c. *For* $\theta = 0.25, \beta = 0.1$ *and* $\sqrt{n} \geq 100$ *the new point* $(\bar{x}, \bar{y}, \bar{s}) = (x, y+\Delta y, s+\Delta s) \in N_2(0.5)$.

Proof. From Lemma 3.1, equation (3.5) and the assumption of this lemma, we find that $\|S^{-1}\Delta s\|_2 \leq \beta/\sqrt{n} < 1$. Therefore $|\Delta s_j/s_j| \leq \beta/\sqrt{n} < 1$ and hence the full Newton step in the dual space is feasible. After this step, the dual feasibility is restored.
It remains to prove the result about the proximity of the new point to the new $\bar{\mu}$-center. We prove first that the new barrier parameter after a full step in the Newton direction in the dual space does not change significantly compared with the previous one. This new barrier parameter is defined as follows

$$n\bar{\mu} = \sum_j (x_j s_j + x_j s_j \frac{\Delta s_j}{s_j}).$$

The inequality $\|S^{-1}\Delta s\|_2 \leq \beta/\sqrt{n}$ implies that $-\beta/\sqrt{n} \leq \frac{\Delta s_j}{s_j} \leq \beta/\sqrt{n}$ and

$$\frac{-\beta}{\sqrt{n}}\, x_j s_j \leq x_j s_j \frac{\Delta s_j}{s_j} \leq \frac{\beta}{\sqrt{n}}\, x_j s_j$$

hence

$$\frac{-\beta}{\sqrt{n}}\, n\mu \leq \sum_j x_j s_j \frac{\Delta s_j}{s_j} \leq \frac{\beta}{\sqrt{n}}\, n\mu$$

so the new barrier parameter $\bar{\mu}$ satisfies

$$n\mu(1-\beta/\sqrt{n}) \leq n\bar{\mu} \leq n\mu(1+\beta/\sqrt{n}).$$

We need to evaluate the proximity of the new primal-dual pair to the new $\bar{\mu}$-center. First observe that

$$\|\bar{X}\bar{S}e - \bar{\mu}e\|_2 \leq \|\bar{X}\bar{S}e - XSe\|_2 + \|XSe - \mu e\|_2 + \|\mu e - \bar{\mu}e\|_2.$$

Each of these three terms can be bounded from above:

$$\|\bar{X}\bar{S}e - XSe\|_2 = \left(\sum_j (x_j s_j \frac{\Delta s_j}{s_j})^2\right)^{1/2} \leq \left(n\,(1+\theta)^2\,\mu^2\,\frac{\beta^2}{n}\right)^{1/2} = (1+\theta)\,\beta\,\mu,$$

$$\|XSe - \mu e\|_2 \leq \theta\mu,$$

$$\|\mu e - \bar{\mu}e\|_2 \leq \left(n\,\frac{\beta^2}{n}\,\mu^2\right)^{1/2} = \beta\mu.$$

Therefore

$$\|\bar{X}\bar{S}e - \bar{\mu}e\|_2 \leq ((1+\theta)\beta + \theta + \beta)\mu.$$

For $\theta = 0.25$ and $\beta = 0.1$ we have $(1+\theta)\beta+\theta+\beta = 0.475$. Also since $\mu(1-\beta/\sqrt{n}) \leq \bar{\mu} \leq \mu(1+\beta/\sqrt{n})$ for $\sqrt{n} \geq 100$, we have $0.99\mu \leq \bar{\mu} \leq 1.01\mu$ and

$$\|\bar{X}\bar{S}e - \bar{\mu}e\|_2 \leq 0.475\mu \leq \frac{0.475}{0.99}\bar{\mu} \leq 0.5\bar{\mu},$$

which completes the proof. □

The choice of constants in Lemma 3.3 has ensured that the new point $(\bar{x}, \bar{y}, \bar{s})$ belongs to a slightly larger neighborhood of the central path, $N_2(0.5)$. It suffices to

make only one pure centering step to get back to the original smaller neighborhood $N_2(0.25)$. This completes the analysis for the short-step method.

The analysis in the case of long-step algorithm differs slightly because we assume the current μ-center belongs to a large neighborhood $N_\infty(0.5, 2.0)$ and after absorbing infeasibility the new point belongs to a larger neighborhood $N_\infty(0.25, 2.5)$. We do not need any re-centering step in this case (we just accept a slightly larger neighborhood, cf. [25]).

LEMMA 3.4. *Let $(x, y, s) \in N_\infty(\gamma_l, \gamma_u)$ and $\beta < 1$. If $\|\tilde{\xi}_c\|_\infty \leq \beta/\|Q\|_\infty$, then the full Newton step is feasible and it absorbs the total infeasibility ξ_c. For $\gamma_l = 0.5, \gamma_u = 2.0$ and $\beta = 0.1$ the new point $(\bar{x}, \bar{y}, \bar{s}) = (x, y + \Delta y, s + \Delta s) \in N_\infty(0.25, 2.5)$.*

Proof. From equation (3.5) and the assumption of this lemma, we find that $\|S^{-1}\Delta s\|_\infty \leq \beta < 1$ and from $S\Delta x + X\Delta s = 0$ also that $\|X^{-1}\Delta x\|_\infty = \|S^{-1}\Delta s\|_\infty < 1$. Hence the full Newton step is feasible. After this step, feasibility is restored. Similarly to the proof of Lemma 3.3, we bound the new barrier parameter $\bar{\mu}$ and analyse the proximity of the new iterate to the central path. From the inequality $\|S^{-1}\Delta s\|_\infty \leq \beta$ we get

$$-\beta\, n\mu \leq \sum_j x_j s_j \frac{\Delta s_j}{s_j} \leq \beta\, n\mu$$

so the new barrier parameter $\bar{\mu}$ satisfies

$$n\mu(1-\beta) \leq n\bar{\mu} \leq n\mu(1+\beta).$$

We still need to prove that component-wise the centrality has not been worsened too much. To prove the claim, we need to show that

$$0.25\bar{\mu} \leq \bar{x}_j \bar{s}_j \leq 2.5\bar{\mu}.$$

Indeed, we find that

$$\bar{x}_j\bar{s}_j = x_j s_j + x_j s_j \frac{\Delta s_j}{s_j} \geq (\gamma_l - \gamma_u\beta)\mu \geq \frac{\gamma_l - \gamma_u\beta}{1+\beta}\,\bar{\mu},$$
$$\bar{x}_j\bar{s}_j = x_j s_j + x_j s_j \frac{\Delta s_j}{s_j} \leq (\gamma_u + \gamma_u\beta)\mu \leq \frac{\gamma_u + \gamma_u\beta}{1-\beta}\,\bar{\mu}.$$

For $\gamma_l = 0.5$, $\gamma_u = 2.0$ and $\beta = 0.1$ we obtain the required result. □

Note that the first part of Lemma 3.4 also follows from Proposition 2 in [26].

The case of the large-update method [21] requires one more comment. This algorithm uses a different proximity measure

$$\Psi(v) = \sum_j \psi(v_j), \quad \text{where} \quad \psi(v_j) = \frac{v_j^2 - 1}{2} + \frac{v_j^{1-q} - 1}{q-1},$$

in which $v_j = \sqrt{\frac{x_j s_j}{\mu}}$, $j = 1, 2, ..., n$ and $q \geq 1$ is an additional parameter. From Lemma 5.3 in [21] we have

$$\Psi(v) \leq \frac{1}{2}\,\|v^{-q} - v\|_2^2.$$

We leave it to the reader to prove the following bridge result.

LEMMA 3.5. *Let two constants γ_l and γ_u such that $0 < \gamma_l \leq 1 \leq \gamma_u$ be given. If $(x, y, s) \in N_\infty(\gamma_l, \gamma_u)$, then $\Psi(v) \leq C\, n$, where C is a constant independent of n.*

As a consequence of Lemma 3.5, we can apply Lemma 2.7 from [21] to conclude that after at most $\mathcal{O}(n^{\frac{q+1}{2q}})$ re-centering iterations the proximity measure of the new point, $\Psi(v')$ will be reduced from $\mathcal{O}(n)$ to $\mathcal{O}(1)$ thus allowing the continuation of the large-update method.

Summing up, for three different variants of the path-following algorithm we have given conditions on the perturbation in the dual space that can be easily absorbed in one Newton step without affecting the key properties of a given algorithm.

3.2. Restoring Primal Feasibility. The case of restoring primal feasibility can be dealt with in a very similar way to that of Section 3.1.

After setting $\xi_c = 0$ (i.e. ignoring dual infeasibility) from (3.3) and (2.7) we obtain

$$\begin{aligned} X^{-1}\Delta x &= S^{-1}A^T(AXS^{-1}A^T)^{-1}\xi_b \\ &= S^{-1}A^T(AXS^{-1}A^T)^{-1}AXX^{-1}A^T(AA^T)^{-1}\xi_b \\ &= (I - Q)\,\tilde{\xi}_b. \end{aligned} \tag{3.6}$$

It is worth noting that we use a relative primal residual $\tilde{\xi}_b = X^{-1}A^T(AA^T)^{-1}\xi_b \in \mathcal{R}^n$ instead of the real perturbation vector $\xi_b \in \mathcal{R}^m$. This has a common sense justification that any primal infeasibility has to be absorbed through changes of the primal variables.

Let us observe that computing the direction Δx could involve through (3.2) and (3.4) also the computation of directions in the dual space $(\Delta y, \Delta s)$ but the latter are skipped without being used. Although we do apply the primal-dual framework to compute feasibility restoration directions, we use the primal perturbation only in the primal feasibility restoration direction (3.6) and the dual perturbation only in the dual feasibility restoration direction (3.5).

We omit the detailed analysis of the case of primal perturbations. We formulate two lemmas analogous to Lemmas 3.3 and 3.4 but skip their proofs because they are almost identical to those of Lemmas 3.3 and 3.4. For these omitted proofs the reader should observe that the bound on the relative step in the primal space (3.6) involves the matrix $I - Q$ and by using Lemma 3.1 we have

$$\|I - Q\|_2 \leq \|Q\|_2 + 1 \leq \left(\frac{1+\theta}{1-\theta}\right)^{1/2} + 1.$$

LEMMA 3.6. *Let $(x, y, s) \in N_2(\theta)$ and $\beta < \sqrt{n}$. If $\|\tilde{\xi}_b\|_2 \leq \frac{\beta}{\sqrt{n}}/((\frac{1+\theta}{1-\theta})^{1/2} + 1)$, then the full Newton step in the primal space is feasible and it absorbs the total infeasibility ξ_b. For $\theta = 0.25, \beta = 0.1$ and $\sqrt{n} \geq 100$ the new point $(\bar{x}, \bar{y}, \bar{s}) = (x + \Delta x, y, s) \in N_2(0.5)$.*

LEMMA 3.7. *Let $(x, y, s) \in N_\infty(\gamma_l, \gamma_u)$ and $\beta < 1$. If $\|\tilde{\xi}_b\|_\infty \leq \beta/(\|Q\|_\infty + 1)$, then the full Newton step is feasible and it absorbs the total infeasibility ξ_b. For $\gamma_l = 0.5$, $\gamma_u = 2.0$ and $\beta = 0.1$ the new point $(\bar{x}, \bar{y}, \bar{s}) = (x + \Delta x, y, s) \in N_\infty(0.25, 2.5)$.*

The case of large update method [21] is covered by the use of Lemmas 3.7 and 3.5.

For ease of the presentation we have split our analysis into two independent cases for the primal and the dual spaces. It is possible to perform the analysis in the presence of perturbations in both spaces at the same time. Then, however, to evaluate the new complementarity products: $\bar{x}_j\bar{s}_j = (x_j + \Delta x_j)(s_j + \Delta s_j)$, we would have to

consider the products $\Delta x_j \Delta s_j$. Having bounds on $\|X^{-1}\Delta x\|_\infty$ and $\|S^{-1}\Delta s\|_\infty$ allows bounding these products with terms proportional to $x_j s_j$. Hence the component-wise changes to the complementarity products can also be bounded by terms proportional to $x_j s_j$ which themselves are bounded by terms proportional to μ. Therefore very similar analysis holds when both infeasibilities have to be absorbed at the same time, but the proofs become longer.

We have left the discussion of the simultaneous treatment of both primal and dual perturbations to the following section in which also some other issues of implementation of our re-optimization technique are addressed.

4. From Theory to Practice. There still exists an important gap between the theory and the computational practice of interior point methods. The theory for feasible path-following algorithms is more elegant and provides better complexity bounds than that for the infeasible algorithms (cf. [25], Chapters 5 and 6). On the other hand, the implementations of interior point methods use infeasible algorithms. In these approaches the primal and dual feasibility is expected to be attained together with optimality although, in practice, the infeasibilities are often reduced much earlier than the duality gap gets decreased below the optimality tolerance.

The infeasible primal-dual algorithm follows the central path, i.e., in subsequent iterations it makes (one) damped step in the Newton direction towards the solution of the first order optimality conditions for the barrier problem and reduces the barrier parameter. The iterates of this algorithm stay in a large neighborhood of the central path (cf. [25], page 109). In our implementation of the infeasible primal-dual method (applied to (2.1)-(2.2)), this neighborhood is defined by the following inequalities

$$\begin{aligned} \|A_0 x - b_0\| &\leq \epsilon_p(\mu)(||r_b^0|| + 1), \\ \|A_0^T y + s - c_0\| &\leq \epsilon_d(\mu)(||r_c^0|| + 1), \\ \gamma_l \mu \leq x_j s_j &\leq \gamma_u \mu, \qquad i = 1, 2, \ldots, n. \end{aligned} \tag{4.1}$$

The two vectors $r_b^0 = A_0 x^0 - b_0$ and $r_c^0 = A_0^T y^0 + s^0 - c_0$ are the violations of the primal and dual constraints at the initial point (x^0, y^0, s^0), respectively. The relative feasibility tolerances $\epsilon_p(\mu)$ and $\epsilon_d(\mu)$ decrease with μ and reach zero at $\mu = 0$. The parameters γ_l and γ_u control the discrepancy between the largest and the smallest complementarity products.

If we expect that re-optimizations will be done, then we impose stronger requirements on the reduction of the primal and dual infeasibilities by a fast reduction of the feasibility tolerances. This occasionally requires an additional re-centering step in which we preserve the duality gap but reduce the infeasibilities. Such a step uses multiple centrality correctors [8] and is expected to reduce the discrepancy between the largest and the smallest complementarity products.

Following [9], if we solve a given linear optimization problem (2.1)-(2.2) and expect that re-optimizations will be done later, then we modify the infeasible primal-dual algorithm and ask for a nearly optimal μ-center to be saved for future re-optimization. That is, we find a point (x, y, s) that satisfies:

$$\begin{aligned} \|A_0 x - b_0\| &\approx 0, \\ \|A_0^T y + s - c_0\| &\approx 0, \\ \gamma_l \mu \leq x_j s_j &\leq \gamma_u \mu, \qquad i = 1, 2, \ldots, n. \end{aligned} \tag{4.2}$$

for some small barrier parameter μ, which guarantees near-optimality.

Let us observe that an exact μ-center satisfies both primal and dual feasibility constraints and the parameter μ controls its distance to optimality since the duality gap at this point is

$$c_0^T x - b_0^T y = c_0^T x - x^T A_0^T y = x^T (c_0 - A_0^T y) = x^T s = n\mu. \tag{4.3}$$

In the usual applications, the path-following algorithm terminates when the duality gap drops below a predetermined relative optimality tolerance ϵ, i.e., when

$$|c_0^T x - b_0^T y| \leq \epsilon(|c_0^T x| + 1), \tag{4.4}$$

with ϵ usually taking the values 10^{-6} or 10^{-8}. However, instead of using (4.4) as the stopping criterion, we can use (4.2). Here the algorithm stops at an approximate μ-center corresponding to the predetermined barrier parameter μ (thereby controlling the required distance to optimality). Once we obtain a rough estimate of the optimal objective value $\tilde{z} = c_0^T x$, e.g. when (4.4) is already satisfied for $\epsilon_0 = 10^{-1}$, we limit the decrease of the barrier to

$$\mu = \hat{\epsilon}\frac{|\tilde{z}|}{n}. \tag{4.5}$$

From (4.2), (4.3) and (4.5) we see that an approximate μ-center corresponding to such a μ is a nearly optimal solution with the relative precision $\hat{\epsilon}$.

The choice of the tolerance $\hat{\epsilon}$ (and, in consequence, the parameter μ) depends on how significant changes to the problem are expected. If we expect violent modifications of the problem, then a larger value of $\hat{\epsilon}$, say 10^{-1} or 10^{-2} is suggested. For expected small perturbations to the problem, we suggest a closer to optimality point with $\hat{\epsilon}$ equal to 10^{-3} or 10^{-4}. It is also possible to store several candidate points and delay the decision on which of them should be used in re-optimization to the time when primal and dual perturbations ξ_b and ξ_c have become known.

From now on we assume that an approximate μ-center (x, y, s) that satisfies (4.2) is stored and the data in the linear program changes from A_0, b_0 and c_0 to the new one A, b and c. Naturally, we cannot expect that the feasible point for an earlier problem is feasible for the new one. We accept the possible violation of both primal and dual feasibility constraints in the new problem:

$$\xi_b = b - Ax \neq 0 \quad \text{and} \quad \xi_c = c - A^T y - s \neq 0$$

and compute the relative perturbation vectors $\tilde{\xi}_b$ and $\tilde{\xi}_c$ from (2.7). Following the theoretical developments of Section 3, since we work with the long-step (infeasible) path-following algorithm, we could use the ∞-norms of the relative perturbation vectors and verify if they satisfy the assumptions of Lemmas 3.4 and 3.7. If we knew $\|Q\|_\infty$, we could check if $\|\tilde{\xi}_c\|_\infty \leq \beta/\|Q\|_\infty$ and $\|\tilde{\xi}_b\|_\infty \leq \beta/(\|Q\|_\infty + 1)$. Since we do not know $\|Q\|_\infty$ we could replace it with an upper bound $\sqrt{n}\,(\frac{\gamma_u}{\gamma_l})^{1/2}$ or with what we expect to be its reasonable estimate $|Q|$. Regardless of whether the real norm $\|Q\|_\infty$ or only its estimate $|Q|$ are used, this would still allow us to predict how successful the feasibility restoration direction can be. In particular, if the conditions of Lemmas 3.4 and 3.7 were not satisfied, we could use for warm starting another μ-center which is further from optimality. Such a point could possibly absorb larger perturbations in the primal and dual spaces. If we had a whole history of iterates stored as suggested

in [27] we could backtrack to an approximate μ-center corresponding to a barrier parameter that is sufficiently large to absorb the perturbations ξ_b and ξ_c.

In the practical algorithm, we compute $S^{-1}\Delta s$ from (3.5) and $X^{-1}\Delta x$ from (3.6) and then perform the ratio tests for the stepsizes in the primal and dual spaces

$$\hat{\alpha}_P := \max\{\alpha > 0 : \; x + \alpha\Delta x \geq 0\}, \tag{4.6}$$

$$\hat{\alpha}_D := \max\{\alpha > 0 : \; s + \alpha\Delta s \geq 0\}. \tag{4.7}$$

Obviously, $\hat{\alpha}_P \geq 1/\|X^{-1}\Delta x\|_\infty > 0$ and $\hat{\alpha}_D \geq 1/\|S^{-1}\Delta s\|_\infty > 0$. If the stepsizes $\hat{\alpha}_P$ and $\hat{\alpha}_D$ are small, say, fall below a prescribed tolerance: $\hat{\alpha}_P \leq \alpha_{min}$ or $\hat{\alpha}_D \leq \alpha_{min}$, then we spread the absorption of primal and dual perturbations across a few following iterations. To achieve this we scale infeasibilities and use

$$\xi_b^{'} = \delta_P\,\xi_b \quad \text{and} \quad \xi_c^{'} = \delta_D\,\xi_c, \tag{4.8}$$

where $\delta_P, \delta_D \in (0,1)$ specify the fraction of infeasibilities that we expect could be absorbed in a single Newton step. We could obviously set $\delta_P = \hat{\alpha}_P$ and $\delta_D = \hat{\alpha}_D$, however, we found that this is sometimes too pessimistic. Therefore we choose $\delta > 1$ and define

$$\delta_P = \delta\,\hat{\alpha}_P \quad \text{and} \quad \delta_D = \delta\,\hat{\alpha}_D. \tag{4.9}$$

Although the analysis deals independently with the infeasibilities in two spaces, our practical re-optimization procedure takes them into account at the same time. We use the primal-dual framework (i.e. Newton equation system (3.1)) and ignore primal infeasibility in the dual step and dual infeasibility in the primal step. We thus have to solve two systems of equations like (3.1) with $\xi_c = 0$ and with $\xi_b = 0$, respectively. (Both these systems use the same factorization, of course.) For the first few iterations of the re-optimization algorithm the step in the primal space results from ξ_b, the step in the dual space results from ξ_c, and only the re-centering steps use both directions at the same time.

It is important to mention that we combine the step in which feasibility perturbations are absorbed with the use of multiple centrality correctors [8]. Hence after the step has been made the proximity of the new point to the central path is not necessarily worsened compared with that of the previous iterate. This is an important feature of our approach because we expect that not all the perturbations can be absorbed in this single interior point iteration; instead, few more iterations may be needed to restore feasibility in the perturbed problem. Therefore we need the intermediate points be as well centered as possible to be good candidates for absorbing the remaining feasibility perturbations.

Summing up, if large perturbations have to be dealt with, our practical re-optimization procedure absorbs them gradually making slow progress towards optimality in a new problem at the same time. One could interpret this as passing through a family of problems with data changing gradually from A_0, b_0, c_0 to A, b, c.

Let us summarize our findings in the re-optimization algorithm below. We assume that (x, y, s) is an approximate μ-center for (2.1) and (2.2) and that in the new linear program this point produces infeasibilities ξ_b and ξ_c given by (2.6). The procedure uses the following parameters: $\gamma_l = 0.5$ and $\gamma_u = 2.0$ define the neighborhood of the central path (2.5), $\delta = 2.0$ determines how much of the perturbations we expect to absorb in one primal-dual iteration (cf. (4.9)), $\alpha_{min} = 0.1$ is a threshold for acceptable stepsizes in the primal and dual spaces. We draw reader's attention to the fact that

our procedure does not choose how much backtracking is needed because we use only one μ-center saved for future warm starting. The procedure could obviously be enhanced by a simple test based on Lemmas 3.4 and 3.7 to choose a suitable starting point from the list of candidates if such a list were available.

Re-optimization with the Primal-Dual Method

Input

(x, y, s): approximate μ-center (4.2);

Parameters

γ_l, γ_u: relative threshold values for outlier complementarity products;
δ: parameter in (4.8) and (4.9);
α_{min}: the minimum acceptable stepsize;

Initialize

Δx: primal feasibility restoring direction (3.6);
$\Delta y, \Delta s$: dual feasibility restoring direction (3.2) (with $\xi_b = 0$) and (3.5);
$\hat{\alpha}_P, \hat{\alpha}_D$: stepsizes (4.6) and (4.7) in the primal and dual spaces;

Absorb Infeasibility

while ($\hat{\alpha}_P \leq \alpha_{min}$ or $\hat{\alpha}_D \leq \alpha_{min}$) **do**
if $(\hat{\alpha}_P \leq \alpha_{min})$ **then**
scale primal direction:
$\Delta x := \delta \hat{\alpha}_P \Delta x$;

endif
if $(\hat{\alpha}_D \leq \alpha_{min})$ **then**
scale dual direction:
$\Delta y := \delta \hat{\alpha}_D \Delta y$,
$\Delta s := \delta \hat{\alpha}_D \Delta s$;

endif
define the predictor direction $\Delta_p = (\Delta x, \Delta y, \Delta s)$;
$\Delta =$ `Re-center`(Δ_p);
`MakeStep` (Δ);
at the new point, recompute:
$(\Delta x, \Delta y, \Delta s)$ from (3.6), (3.2) (with $\xi_b = 0$) and (3.5);
$\hat{\alpha}_P, \hat{\alpha}_D$ from (4.6) and (4.7);

end-while

In our implementation $\gamma_l = 0.5$, $\gamma_u = 2.0$, $\delta = 2.0$, and $\alpha_{min} = 0.1$.

Two procedures in this algorithm need further comments. In the `Re-center`(Δ_p) procedure, the direction $\Delta_p = (\Delta x, \Delta y, \Delta s)$ is used as a predictor direction for the multiple centrality correctors technique [8]. Centrality correctors usually alter Δ_p and replace it with a new direction Δ. Centrality correctors aim at two goals: firstly to increase the stepsizes from $1/\delta = 0.5$ to larger values $\alpha_P, \alpha_D \in (0.5, 1)$ and secondly to improve the centrality of the new iterate, i.e. to decrease the spread between the

largest and the smallest complementarity products in it. In the procedure `MakeStep` the maximum feasible stepsizes in the primal and dual spaces are determined along direction Δ, the variables are updated from (x, y, s) to $(\bar{x}, \bar{y}, \bar{s})$ and infeasibilities ξ_b and ξ_c are recomputed.

The re-optimization procedure terminates when the significant part of initial perturbations in the primal and dual spaces have already been absorbed and for the remaining infeasibilities ξ_b and ξ_c the stepsizes in both spaces exceed the threshold α_{min}. We have found that at this point there is no more need for special feasibility restoration steps. The usual infeasible primal-dual method can be used to terminate the optimization.

5. Crash Start of Interior Point Method. In this paper, we report results on the application of the re-optimization technique to the problem of finding a good advanced starting point. In the next section we will present a decomposition based technique which aims to find an approximate μ-center quickly without using an interior point method for the whole system. This point is then used to start our re-optimization method from. Since this advanced solution is not the result of solving a similar linear program we cannot assume that we have a well centered point to begin with. In other words, in addition to possible large perturbations of the primal and dual feasibility, also ξ_μ in (2.6) may be large.

Guided by the theoretical results of Section 3, we realize that if some of the primal or dual slack variables are very close to zero, then the relative primal and dual perturbations (2.7) may become huge and this would inevitably lead to very small stepsizes $\hat{\alpha}_P$ and $\hat{\alpha}_D$ in the re-optimization procedure (cf. (4.6) and (4.7)). Therefore, when constructing a candidate for a starting point we shall bound these variables away from zero. Additionally, since the theory indicates through Lemma 3.2 that the ratio between the largest and the smallest complementarity products (bounded by γ_u/γ_l) contributes to the increase of $\|Q\|$, we shall pay particular attention to limiting the spread of complementarity products even at the expense of increasing primal and dual infeasibilities ξ_b and ξ_c.

Summing up, we shall look for a candidate initial point (x^*, y^*, s^*) that satisfies the following requirements:

1. $\exists \mu^* : \gamma_l \mu^* \leq x_j^* s_j^* \leq \gamma_u \mu^*$ for some $0 < \gamma_l \leq 1 \leq \gamma_u$ with γ_u/γ_l small;
2. μ^* is small;
3. primal and dual infeasibilities $\xi_b = b_0 - A_0 x^*$ and $\xi_c = c_0 - A_0^T y^* - s^*$ are small.

One could design different heuristics or more rigorous algorithms that would generate a good candidate point (x^*, y^*, s^*) along these lines. We want to apply our *crash* procedure to the solution of large structured linear programs. Hence we expect that the key to its success lies in exploiting the structure of the problem. Our approach uses one iteration of decomposition method to guess the initial point. We describe our heuristic in detail in the following sections.

Many real-life linear programs display some particular block structures. The structure usually results from system dynamics, uncertainty, spatial distribution or other factors that lead to the creation of huge problems made up of small, nearly identical parts which have to be coordinated through time, uncertainty, space or other dimensions. Moreover, modeling very complicated real-life optimization problems

often requires nested embedding of structures. The presence of special structure in the problem should be exploited by an interior point method. It can simplify and/or accelerate the execution of linear algebra operations [10].

The problem structure may also be used to find an advanced starting point. We shall illustrate this idea on two well-known classes of specially structured problems: the primal block-angular and the dual block-angular ones. Although for ease of the presentation we shall restrict our discussion to those two classes, we shall apply a similar approach also to more complicated linear programs that display nested block structures which combine those two.

Let us remind that the constraint matrix of the linear program with the primal block-angular structure has the following form

$$A = \begin{pmatrix} A_1 & & & & \\ & A_2 & & & \\ & & \ddots & & \\ & & & A_n & \\ B_1 & B_2 & \cdots & B_n & B_0 \end{pmatrix}, \tag{5.1}$$

where $A_i \in \mathcal{R}^{m_i \times n_i}, i = 1, ..., n$ and $B_i \in \mathcal{R}^{m_0 \times n_i}, i = 0, ..., n$. Matrix A has then $M = m_0 + \sum_{i=1}^{n} m_i$ rows and $N = \sum_{i=0}^{n} n_i$ columns. The constraint matrix of the linear program with the dual block-angular structure has the following form

$$A = \begin{pmatrix} A_1 & & & C_1 \\ & A_2 & & C_2 \\ & & \ddots & \vdots \\ & & A_n & C_n \end{pmatrix}, \tag{5.2}$$

where $A_i \in \mathcal{R}^{m_i \times n_i}, i = 1, ..., n$ and $C_i \in \mathcal{R}^{m_i \times k}, i = 1, ..., n$. Matrix A has then $M = \sum_{i=1}^{n} m_i$ rows and $N = k + \sum_{i=1}^{n} n_i$ columns. In the following sections we will use the partitioning of objective and variable vectors $c = [c_1, \ldots, c_n, c_0], x = [x_1, \ldots, x_n, x_0]$ and likewise the vector of right-hand sides $b = [b_1, \ldots, b_n, b_0]$ in (5.1) and $b = [b_1, \ldots, b_n]$ in (5.2). Dual variables y for constraints and s for the non-negativity constraints on x are partitioned as b and x, respectively.

5.1. Decomposition Based Starting Point. A problem whose constraint matrix is of the forms (5.1), (5.2) lends itself obviously to a decomposition approach. In our experience such a strategy is efficient at finding a near optimal point fast, however might take a long time to converge to within a specified tolerance. The idea is therefore to construct a starting point for the interior point method from information obtained after one iteration of a decomposition scheme applied to the original problem. One major aim will be to construct a point which is as close to primal and dual feasibility as possible. Assume we apply Dantzig-Wolfe decomposition [6] to (5.1). After dualising the coupling constraint the problem decomposes, and each of the subproblems could be solved independently. We could then combine the subproblem solutions to obtain an advanced starting point for the interior point method. The difficulty with this approach is that while the resulting point is dual feasible in the complete problem, there would be a considerable violation of primal feasibility in the (earlier ignored) coupling constraint. Similarly applying Benders decomposition [4] to

(5.2) and combining subproblem solutions would yield a point which is primal feasible, while violating dual feasibility (corresponding to ignored linking variables). Our idea therefore is to combine the two decomposition approaches. Problems (5.1) and (5.2) are extended into forms that allow to apply both Dantzig-Wolfe and Benders decomposition. We will first apply Dantzig-Wolfe decomposition; from its solution values of complicating variables to use in Benders decomposition can be derived. The Benders subproblem is then solved and from the solutions of both sets of subproblems we will construct a point which is close to both primal and dual feasibility.

A scheme to solve optimization problems by iterating between Dantzig-Wolfe and Benders subproblems has been suggested as Cross-Decomposition by van Roy [22] and Vlahos [23]. We will revise the Cross-Decomposition algorithm applied to problems of form (5.1) and (5.2). Note however that Cross-Decomposition requires strong assumptions about boundedness and feasibility of the resulting subproblems, which are not satisfied in our case. We will therefore suggest modifications to the algorithm which make it suited for our application.

5.2. Cross-Decomposition for Primal Block-Angular Structure. Assume in the next two sections that the system matrix A is of form (5.1). The problem could be solved by Dantzig-Wolfe decomposition. However, introducing extra variables $h_i, i = 0, \ldots, n$ and constraints $B_i x_i - h_i = 0, i = 0, \ldots, n$ leads to the augmented problem

$$\min_{\substack{x_0,\ldots,x_n \geq 0 \\ h_0,\ldots,h_n}} \sum_{i=0}^{n} c_i^T x_i \quad \text{s.t.} \quad \begin{array}{rl} A_i x_i & = b_i, \quad i = 1, \ldots, n \\ B_i x_i - h_i & = 0, \quad i = 0, \ldots, n \\ \sum_{i=0}^{n} h_i & = b_0 \end{array} \tag{5.3}$$

and enables us to apply Benders decomposition using $h = (h_0, \ldots, h_n)$ as complicating variables. The Cross-Decomposition scheme applied to (5.3) would proceed as follows: A guess of the multiplier $\hat{y}_0$ on the $\sum h_i = b_0$ constraint is obtained. With this the Dantzig-Wolfe subproblem

$$v_D(\hat{y}_0) = \min_{x_0,\ldots,x_n \geq 0} \sum_{i=0}^{n} (c_i - B_i^T \hat{y}_0)^T x_i \quad \text{s.t.} \quad A_i x_i = b_i, \quad i = 1, \ldots, n \tag{5.4}$$

is solved. From its solution $(x_0^*, \ldots, x_n^*)$ values $\hat{h}_i = B_i x_i^*$ of the Benders complicating variables are obtained, and with these the Benders subproblem

$$v_P(\hat{h}) = \min_{x_0,\ldots,x_n \geq 0} \sum_{i=0}^{n} c_i^T x_i \quad \text{s.t.} \quad \begin{array}{rl} A_i x_i & = b_i, \quad i = 1, \ldots, n \\ B_i x_i & = \hat{h}_i, \quad i = 0, \ldots, n. \end{array} \tag{5.5}$$

is solved. Multipliers $y_{0,i}^*$ on the $B_i x_i = \hat{h}_i$ constraints are obtained and averaged $\hat{y}_0 = (\sum_i y_{0,i}^*)/(n+1)$, which is then used again in (5.4). Note that problems (5.4) and (5.5) separate into n and $n+1$ smaller problems, respectively. However, in (5.4) the subproblems might be unbounded and in (5.5) the subproblems might be infeasible. We will now show how this procedure can be used to construct an advanced starting point.

5.3. Crash Start for Primal Block-Angular Problem. We are aiming for a point that is both near to primal and dual feasibility and close to the central path.

Using the particular system matrix (5.1) we therefore aim to satisfy (compare (2.3)):

$$c_i - A_i^T y_i - B_i^T y_0 - s_i = 0 \tag{5.6}$$
$$c_0 - B_0^T y_0 - s_0 = 0 \tag{5.7}$$
$$A_i x_i = b_i \tag{5.8}$$
$$\sum\nolimits_{i=0}^{n} B_i x_i = b_0 \tag{5.9}$$
$$S_i X_i e = \mu e \tag{5.10}$$
$$S_0 X_0 e = \mu e \tag{5.11}$$
$$s_i, x_i, s_0, x_0 \geq 0. \tag{5.12}$$

Let us now assume that an estimate $\hat{y}_0$ of the complicating constraint multiplier is available (we have used $\hat{y}_0 = e$ in the tests). With this the Dantzig-Wolfe subproblem (5.4) is solved. At the solution (denoted by superscripts $^{(1)}$) the following KKT conditions hold

$$c_i - A_i^T y_i^{(1)} - B_i^T \hat{y}_0 - s_i^{(1)} = 0, \quad i = 1, \ldots, n \tag{5.13}$$
$$A_i x_i^{(1)} = b_i, \quad i = 1, \ldots, n. \tag{5.14}$$

Note that as long as $c_i - B_i^T \hat{y}_0 \geq 0$, problem (5.4) is bounded, a condition which will be satisfied in our test problems. From this estimates $h_i^{(1)} = B_i x_i^{(1)}$ of the Benders complicating variables are obtained. Projecting them on the $\sum h_i = b_0$ constraint by $(\hat{h}_i)_j = (h_i^{(1)})_j (b_0)_j / (\sum h_i^{(1)})_j$ the Benders subproblem (5.5) is solved with $\hat{h}_i$ as complicating variables. Note that this subproblem is not necessarily feasible, so (5.5) is replaced by a penalized version

$$v_P(\hat{h}) = \min_{\substack{x_0,\ldots,x_n \geq 0 \\ p_i^+, p_i^- \geq 0}} \sum_{i=0}^{n} c_i^T x_i + \gamma e^T (p_i^+ + p_i^-) \quad \text{s.t.} \quad \begin{aligned} A_i x_i &= b_i, & i &= 1, \ldots, n \\ B_i x_i - \hat{h}_i &= p_i^+ - p_i^-, & i &= 0, \ldots, n, \end{aligned} \tag{5.15}$$

whose solution (denoted by superscripts $^{(2)}$) satisfies the KKT conditions

$$c_i - A_i^T y_i^{(2)} - B_i^T y_{0,i}^{(2)} - s_i^{(2)} = 0 \tag{5.16}$$
$$A_i x_i^{(2)} = b_i \tag{5.17}$$
$$B_i x_i^{(2)} - \hat{h}_i = p_i^+ - p_i^-. \tag{5.18}$$

After solving (5.4) and (5.15), we can accumulate a guess solution (x^*, y^*, s^*) to the LP problem of form (5.1) as follows

$$\begin{aligned} x_i^* &= x_i^{(2)} \\ y_i^* &= y_i^{(1)} \\ s_i^* &= s_i^{(1)} \\ x_0^* &= \max\{B_0^{-1}(b_0 - \sum_{i=1}^{n} B_i x_i^{(2)}), 0\} \\ y_0^* &= \hat{y}_0 \\ s_0^* &= c_0 - B_0^T \hat{y}_0, \end{aligned}$$

where it is assumed that B_0^{-1} is easy to compute (such as when B_0 is diagonal). With these choices dual feasibility is ensured by (5.13) and the definition of s_0^*. $A_i x_i^* = b_i$ holds by (5.17), and $\sum_{i=1}^0 B_i x_i^* - b_0 = \sum_{i=1}^n (p_i^+ - p_i^-)$ which should be small due to our choice of objective in (5.15). So the guess is close to primal feasibility. A good spread of complementarity products is achieved as follows.

All subproblems are solved by an interior point method. Rather than ensuring convergence of the subproblem we are aiming for a point on the central path for a relatively small μ. In order to do so we choose a target value $\hat{\mu}$ ($\hat{\mu} = 0.01$ has been used in the tests) and obtain an estimate $\tilde{z}_i \approx c_i^T x_i$ of the optimal objective value for each subproblem. The subproblems are solved using

$$\epsilon_i = \hat{\mu} n_i / |\tilde{z}_i| \tag{5.19}$$

in (4.4) as stopping criterion (see the discussion leading to (4.5)) and two additional re-centering steps using multiple centrality correctors are performed. Further we set $(x_0^*)_j = \max\{1, (x_0^*)_j\}, (s_0^*)_j = \max\{\hat{\mu}, (s_0^*)_j\}$. This ensures that

$$X_i^* S_i^* e \approx \hat{\mu} e \qquad\qquad X_0^* S_0^* e \geq \hat{\mu} e \tag{5.20}$$

so that the point (x^*, y^*, s^*) should be reasonably well centered for the application of the interior point method.

5.4. Cross-Decomposition for Dual Block-Angular Structure. In the case where the system matrix A is of form (5.2) we proceed similarly. We will start by stating the Cross-Decomposition for this case. In order to apply the Dantzig-Wolfe scheme we need to introduce additional variables $x_{0,i}, i = 1, \ldots, n$ and constraints $x_{0,i} - x_0 = 0, i = 1, \ldots, n$ to arrive at the augmented problem

$$\min_{\substack{x_0, x_i, x_{0,i} \geq 0 \\ i=1,\ldots,n}} \sum_{i=0}^{n} c_i^T x_i \quad \text{s.t.} \quad \begin{aligned} A_i x_i + C_i x_{0,i} &= b_i, \qquad i = 1, \ldots, n \\ x_{0,i} - x_0 &= 0, \qquad i = 1, \ldots, n. \end{aligned} \tag{5.21}$$

The Cross-Decomposition starts by relaxing the $x_{0,i} - x_0 = 0$ constraints. Initial multipliers $\hat{\lambda} = (\hat{\lambda}_1, \ldots, \hat{\lambda}_n)$ are guessed and the subproblem

$$v_D(\hat{\lambda}) = \min_{\substack{x_0, x_i, x_{0,i} \geq 0 \\ i=1,\ldots,n}} \sum_{i=0}^{n} c_i^T x_i + \sum_{i=1}^{n} \hat{\lambda}_i^T (x_{0,i} - x_0) \quad \text{s.t.} \quad A_i x_i + C_i x_{0,i} = b_i, \quad i = 1, \ldots, n \tag{5.22}$$

is solved. The optimal value of x_0 is obtained and used as complicating variable $\hat{x}_0$ in the Benders subproblem

$$v_P(\hat{x}_0) = \min_{\substack{x_i, x_{0,i} \geq 0 \\ i=1,\ldots,n}} \sum_{i=1}^{n} c_i^T x_i + c_0^T \hat{x}_0 \quad \text{s.t.} \quad \begin{aligned} A_i x_i + C_i x_{0,i} &= b_i \\ x_{0,i} &= \hat{x}_0. \end{aligned} \tag{5.23}$$

Multipliers λ_i^* on the $x_{0,i} = \hat{x}_0$ constraints are obtained and used as new $\hat{\lambda}_i$ in the next iteration of (5.22). Note that again (5.23) and (5.22) separate into smaller subproblems. Further (5.22) might be unbounded, just as (5.23) might be infeasible. We will now derive how we modify this scheme to construct an advanced starting point which is close to primal and dual feasibility.

5.5. Crash Start for Dual Block-Angular Problem. From the KKT conditions of the augmented system (5.21), $\sum_{i=1}^{n} \lambda_i = c_0$ needs to be satisfied at the solution. We therefore restrict ourselves to such choices of $\hat{\lambda}$ and use for instance $\hat{\lambda}_i = c_0/n$ as starting guess. With this choice subproblem (5.22) simplifies to

$$v_D(\hat{\lambda}) = \min_{\substack{x_i, x_{0,i} \geq 0 \\ i=1,\ldots,n}} \sum_{i=1}^{n} (c_i^T x_i + \hat{\lambda}_i^T x_{0,i}) \quad \text{s.t} \quad A_i x_i + C_i x_{0,i} = b_i, \quad i = 1, \ldots, n \tag{5.24}$$

and separates into n smaller subproblems. The Benders complicating variables $\hat{x}_0$ could be obtained by $\hat{x}_0 = \sum x_{0,i}/n$. In our test problems however the C_i are the negatives of projection matrices, so that a large value of $\hat{x}_0$ is likely to lead to a feasible Benders subproblem. Therefore we have used $(\hat{x}_0)_j = \max_j\{(x_{0,i})_j\}$. Problem (5.23) however might still be infeasible, so again it is replaced by a penalized version

$$v_P(\hat{x}_0) = \min_{\substack{x_1,\ldots,x_n \geq 0 \\ p_i^+, p_i^- \geq 0}} \sum_{i=1}^{n} (c_i^T x_i + \gamma e^T (p_i^+ + p_i^-)) \quad \text{s.t.} \quad A_i x_i + p_i^+ - p_i^- = b_i - C_i \hat{x}_0, \tag{5.25}$$

where also the $x_{0,i}$ have been substituted out. If new estimates $\hat{\lambda}_i$ are needed, they could be obtained as $\hat{\lambda}_i = -C_i^T y_i^* - s_i^*$, as can be motivated by the KKT conditions for the augmented system.

The estimate for an advanced solution is again obtained by combining the solutions of (5.24) and (5.25). We are aiming to find a point close to the central path for an LP with system matrix (5.2), so we aim to satisfy

$$c_i - A_i^T y_i - s_i = 0 \tag{5.26}$$

$$c_0 - \sum_{i=1}^{n} C_i^T y_i - s_0 = 0 \tag{5.27}$$

$$A_i x_i + C_i x_0 = b_i \tag{5.28}$$

$$S_i X_i e = \mu e \tag{5.29}$$

$$S_0 X_0 e = \mu e \tag{5.30}$$

$$s_i, x_i, s_0, x_0 \geq 0. \tag{5.31}$$

The solutions to the Lagrangian subproblem (5.24) (denoted by superscripts $^{(1)}$) and the Benders subproblem (5.25) (superscripts $^{(2)}$) satisfy respectively

$$A_i^T y_i^{(1)} + s_i^{(1)} = c_i \tag{5.32}$$

$$C_i^T y_i^{(1)} + s_{0,i}^{(1)} = \hat{\lambda}_i \tag{5.33}$$

$$A_i x_i^{(1)} + C_i x_{0,i}^{(1)} = b_i \tag{5.34}$$

$$A_i^T y_i^{(2)} + s_i^{(2)} = c_i \tag{5.35}$$

$$A_i x_i^{(2)} + C_i \hat{x}_0 + p_i^+ - p_i^- = b_i. \tag{5.36}$$

We combine these solutions by choosing

$$\begin{aligned} x_i^* &= x_i^{(2)} \\ y_i^* &= y_i^{(1)} \\ s_i^* &= s_i^{(1)} \\ x_0^* &= \hat{x}_0 \\ s_0^* &= \sum s_{0,i}^{(1)}. \end{aligned}$$

With these choices equation (5.26) is satisfied by (5.32); (5.27) follows from (5.33) together with the definition of s_0^* and the fact that $c_0 = \sum \hat{\lambda}_i$. Together the advanced solution is dual feasible. Further we have that the residual of equation (5.28) is $p_i^+ - p_i^-$ which should be small.

To obtain a fairly well centered point we apply the same heuristic as in the last section. We solve subproblems only to a relative accuracy of ϵ_i given by (5.19) and set $(x_0^*)_j = \max\{1, (x_0^*)_j\}, (s_0^*)_j = \max\{\hat{\mu}, (s_0^*)_j\}$.

6. Numerical Results. We have tested our approach in the context of OOPS, the object-oriented parallel solver [10]. We have implemented the re-optimization procedure and our crash procedure for constructing an advanced starting point as described in Section 5. We have applied the *crash* starting in the solution of several different classes of structured linear programs.

Our computational results demonstrate the performance of the crash procedure and give an insight into the practical behavior of the re-optimization strategy. All classes of problems solved in this paper are well documented in the literature so we restrict their description to an explanation of the associated block structures. These problems originate from network optimization [1] and multistage stochastic programming applied to asset liability management [28].

Multicommodity Network flow problems (MCNF) are of primal block-angular structure; all other problems display a nested dual block-angular structure: Multicommodity Survivable Network Design (MSND) and Joint Optimal Synthesis of Base and Spare Capacity (JOSBP) problems have primal block-angular subblocks. Detailed formulations of all these problems can be found in [10]. The Asset Liability Management (ALM) problems have dual block-angular subblocks.

We have used the algorithm to generate an advanced starting point as described in Section 5 with a few minor variations. First note that not all the slack variables p_i^+, p_i^- in (5.15) and (5.25) are necessary since some slacks might be easily picked up by the problem variables. We have removed these slacks from the subproblems as far as possible. Further for the ALM problems the complicating variable costs c_0 are zero and the default choice of $\hat{\lambda}$ ($\hat{\lambda}_i = c_0/n = 0$) would lead to unbounded Lagrangian Relaxation subproblems. We have therefore used a different choice of $\hat{\lambda}$, still satisfying $\sum_i \hat{\lambda}_i = c_0 = 0$, that guarantees bounded subproblems.

In Table 6.1 we report problem statistics. The problems are grouped by categories. For each problem we give its size in numbers of rows and columns and the number of diagonal blocks. In Table 6.2 we report the results of our method. The column following the problem name contains the number of iterations to reach optimal solution from the default starting point (which is based on Mehrotra [16]). The final block states the results for our algorithm. Its first column reports the number of iterations of the interior point method starting from the crash point and using the re-optimization. The following columns give some useful numbers: $\|\tilde{\xi}_b\|_\infty$ and $\|\tilde{\xi}_c\|_\infty$ at the start of the re-optimization, γ_u/γ_l as a measure of initial centrality of the generated point, and `itf`: the number of steps needed to reach $\max\{\|\xi_c\|_\infty, \|\xi_b\|_\infty\} \leq 0.01$ together with the initial stepsizes in primal and dual spaces as an indication of how fast primal and dual feasibility is regained.

It can be seen that in all cases the interior point method needed less iterations to converge to a solution from our advanced starting point than from the default starting

Problem	Rows	Columns	Blks
MCNF RN	14232	72996	119
MCNF R2	3621	16600	60
MCNF R4	6981	20850	70
MCNF R6	8715	51300	86
MCNF R13	88613	460000	288
MCNF R14	159602	637600	398
MSND PB1	22213	72514	81
MSND PB2	59021	207901	102
MSND PB3	54657	188266	109
MSND PB4	83561	294735	123
MSND PB5	242570	886178	179
JOSBP T1	1021	2400	26
JOSBP T2	3414	7266	43
JOSBP T3	13053	26860	80
JOSBP P1	3241	6970	42
JOSBP P2	6492	13978	59
JOSBP P3	14221	32760	92
ALM P1	66666	166665	101
ALM P2	666666	1666665	101
ALM P3	1222221	3333330	101

TABLE 6.1
Problem Statistics.

	default	advanced starting point						
Problem	Iters	Iters	$\|\tilde{\xi}_b\|_\infty$	$\|\tilde{\xi}_c\|_\infty$	$(\gamma_u/\gamma_l)^{\frac{1}{2}}$	itf	α_P	α_D
MCNF RN	31	22	7.1e+3	6.9e-1	2236	22	3.0e-2	7.9e-2
MCNF R2	20	12	4.0e+3	4.8e-4	875	12	1.8e-2	7.2e-2
MCNF R4	18	10	2.7e+4	1.1e-3	504	8	1.6e-1	2.8e-1
MCNF R6	20	12	7.4e+4	3.4e-3	1240	11	1.3e-2	7.5e-2
MCNF R13	37	26	3.9e+5	5.0e-3	3435	26	8.5e-3	3.3e-2
MCNF R14	49	38	3.3e+5	3.0e-3	3332	37	1.1e-2	1.4e-2
MSND PB1	25	20	2.81	0.97	269	10	8.3e-2	5.8e-2
MSND PB2	33	20	1.94	0.96	369	14	3.1e-2	6.6e-2
MSND PB3	29	17	1.40	0.97	359	12	1.0e-1	1.6e-2
MSND PB4	36	22	1.66	0.97	462	18	1.2e-1	1.8e-3
MSND PB5	51	29	1.59	0.97	589	20	2.6e-2	4.6e-3
JOSBP T1	15	12	14.3	0.99	74	8	3.6e-2	3.0e-1
JOSBP T2	22	19	13.6	0.97	108	18	9.3e-3	1.3e-2
JOSBP T3	28	22	1.23	1.00	32	13	1.3e-2	1.2e-2
JOSBP P1	25	19	1.17	1.00	42	13	1.6e-2	4.6e-2
JOSBP P2	27	22	0.96	1.00	41	13	1.1e-2	4.2e-2
JOSBP P3	40	37	0.70	1.00	34	7	1.0e-2	3.5e-2
ALM P1	21	12	2.7e+3	0.96	741	4	3.5e-1	3.9e-1
ALM P2	44	22	4.6e+3	0.97	2280	5	3.6e-3	2.7e-1
ALM P3	66	30	8.4e+4	0.98	2626	6	2.6e-2	2.5e-1

TABLE 6.2
Solution Statistics.

Prob	default start					advanced start				
	1 proc	2 procs		4 procs		1 proc	2 procs		4 procs	
	Time	Time	s-up	Time	s-up	Time	Time	s-up	Time	s-up
R13	1855	952	1.95	468	3.96	1427	734	1.94	377	3.79
R14	3560	1782	1.99	892	3.99	2730	1387	1.97	715	3.82
PB4	644	373	1.73	185	3.38	625	361	1.73	180	3.47
PB5	3255	1859	1.75	1003	3.25	2662	1491	1.78	850	3.13
P2	6070	3124	1.94	1674	3.62	3593	1813	1.98	1067	3.36
P3	29752	15143	1.96	7804	3.81	16039	8350	1.92	5118	3.13

TABLE 6.3
Speed-ups for parallel implementation.

point. On average about 33% of iterations could be saved. For the six largest of our test problems an average of 40% of iterations was saved and this translated into a 25% decrease in CPU time as can be seen in Table 6.3.

For all problem classes the advanced starting point is dual feasible. However for the dual block-angular problems s_0^* might have to be bounded away from zero, resulting in a scaled dual infeasibility of ≈ 1. For classes MSND and JOSPB, our choice for x_0^* is expected to lead to a solution of (5.25) with zero slacks, resulting in a primal feasible advanced starting point; for the other classes however primal feasibility is more difficult to achieve. These observations are reflected in the results.

It should be noted that for most problems the measure of centrality γ_u/γ_l is fairly large. This results as expected from the earlier discussion in small initial stepsizes α_P, α_D and hence a large number of iterations to regain feasibility. However, encouragingly, even in these adverse circumstances, the re-optimization strategy obtains good results.

Our implementation, OOPS, of the primal-dual method is a parallel solver, and the crash start has been implemented in parallel in the obvious way by distributing the Benders and Dantzig-Wolfe subproblems amongst processors. The program has been run on 4-processors Sun Enterprise 6500 parallel computer. Table 6.3 states the CPU times (first values) and obtained speed-ups (second values) on 2 and 4 processors for the largest of our test problems. Speed-ups between 3.1-3.8 were obtained. These are consistent with the speed-ups obtained for OOPS without crash start.

7. Conclusions. We have discussed the strategy that exploits a well centered solution of one linear program to generate a good starting point for a perturbed problem. We have given bounds on the size of perturbations of the primal and dual feasibility that can be absorbed in merely one Newton step. Our bounds critically depend on the size of perturbations measured in the scaled space of primal-dual solutions and can easily be computed in practical situations that require warm starting. They can thus be used to facilitate a choice of one well centered point from the list of candidates if such a list is available.

We have performed the analysis for the *feasible* path-following methods. We have shown that the measure of proximity to the central path appropriate for each of these methods will not be corrupted by the feasibility restoration step (if the perturbations are small, of course). We have then translated our findings into a computational practice of *infeasible* path-following method. The practical re-optimization strategy

spreads the process of restoring feasibility into few subsequent iterations. If the perturbations are large then only a fraction of them is absorbed in a single iteration.

Finally, we have applied the re-optimization technique to allow the start of an interior point method from almost an arbitrary point. We have provided numerical results which confirm that this strategy works well for large structured linear programs and that it can be efficiently implemented in parallel.

Acknowledgement. We are grateful to the anonymous referee for his comments.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*, Prentice-Hall, New York, 1993.

[2] E. D. Andersen and K. D. Andersen, *The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm*, in High Performance Optimization, H. Frenk, K. Roos, T. Terlaky, and S. Zhang, eds., Kluwer Academic Publishers, 2000, pp. 197–232.

[3] E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu, *Implementation of interior point methods for large scale linear programming*, in Interior Point Methods in Mathematical Programming, T. Terlaky, ed., Kluwer Academic Publishers, 1996, pp. 189–252.

[4] J. F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, Numerische Mathematik, 4 (1962), pp. 238–252.

[5] R. E. Bixby, *Implementing the simplex method: the initial basis*, ORSA Journal on Computing, 4 (1992), pp. 267–284.

[6] G. B. Dantzig and P. Wolfe, *The decomposition algorithm for linear programming*, Econometrica, 29 (1961), pp. 767–778.

[7] I. I. Dikin, *On the speed of an iterative process*, Upravlaemye Sistemy, 12 (1974), pp. 54–60.

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

[9] ———, *Warm start of the primal-dual method applied in the cutting plane scheme*, Mathematical Programming, 83 (1998), pp. 125–143.

[10] J. Gondzio and R. Sarkissian, *Parallel interior point solver for structured linear programs*, Tech. Report MS-00-025, Department of Mathematics and Statistics, University of Edinburgh, King's Buildings, Edinburgh EH9 3JZ, UK, December 2000, revised in February 2002.

[11] J. Gondzio and J.-P. Vial, *Warm start and ε-subgradients in cutting plane scheme for block-angular linear programs*, Computational Optimization and Applications, 14 (1999), pp. 17–36.

[12] N. I. M. Gould and J. K. Reid, *New crash procedures for large-scale systems of linear constraints*, Mathematical Programming, 45 (1989), pp. 475–501.

[13] J. E. Kelley, *The cutting plane method for solving convex programs*, Journal of the SIAM, 8 (1960), pp. 703–712.

[14] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Interior point methods for linear programming: computational state of the art*, ORSA Journal on Computing, 6 (1994), pp. 1–14.

[15] I. Maros and G. Mitra, *Startegies for creating advanced bases for large-scale linear programming problems*, INFORMS Journal on Computing, 10 (Spring 1998), pp. 248–260.

[16] S. Mehrotra, *On the implementation of a primal–dual interior point method*, SIAM Journal on Optimization, 2 (1992), pp. 575–601.

[17] J. E. Mitchell, *Karmakar's Algorithm and Combinatorial Optimization Problems*, PhD thesis, Cornell University, 1988.

[18] ———, *Computational experience with an interior point cutting plane algorithm*, SIAM Journal on Optimization, 10 (2000), pp. 1212–1227.

[19] J. E. Mitchell and M. J. Todd, *Solving combinatorial optimization problems using Karmarkar's algorithm*, Mathematical Programming, 56 (1992), pp. 245–284.

[20] M. A. Nunez and R. M. Freund, *Condition measures and properties of the central trajectory*, Mathematical Programming, 83 (1998), pp. 1–28.

[21] J. Peng, C. Roos, and T. Terlaky, *A new and efficient large-update interior-point method for linear optimization*, Journal of Computational Technologies, 6 (2001), pp. 61–80.

[22] T. J. Van Roy, *Cross decomposition for mixed integer programming*, Mathematical Programming, 25 (1983), pp. 46–63.

[23] K. Vlahos, *Generalised cross decomposition application to electricity capacity planning*, tech. report, London Business School, 1991. presented at TIMS XXIX conference.
[24] L. A. Wolsey, *Integer Programming*, John Wiley & Sons, New York, 1998.
[25] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.
[26] E. A. Yildirim and M. J. Todd, *Sensitivity analysis in linear programming and semidefinite programming using interior-point methods*, Mathematical Programming, 90 (2001), pp. 229–261.
[27] E. A. Yildirim and S. J. Wright, *Warm-start strategies in interior-point methods for linear programming*, SIAM Journal on Optimization, 12 (2002), pp. 782–810.
[28] W. T. Ziemba and J. M. Mulvey, *Worldwide Asset and Liability Modeling*, Publications of the Newton Institute, Cambridge University Press, Cambridge, 1998.