OPTIMIZATION METHODS AND APPLICATIONS
Published by KLUWER 1999

1 A HOMOGENIZED CUTTING PLANE METHOD TO SOLVE THE CONVEX FEASIBILITY PROBLEM

E. D. Andersen, J. E. Mitchell*, C. Roos, and T. Terlaky.
Faculty of ITS, TWI/SSOR,
TU Delft,
2628 CD Delft,
The Netherlands.

Abstract: We present a cutting plane algorithm for the feasibility problem that uses a homogenized self-dual approach to regain an approximate center when adding a cut. The algorithm requires a fully polynomial number of Newton steps. One novelty in the analysis of the algorithm is the use of a powerful proximity measure which is widely used in interior point methods but not previously used in the analysis of cutting plane methods.

Moreover, a practical implementation of a variant of the homogenized cutting plane for solution of LPs is presented. Computational results with this implementation show that it is possible to solve a problem having several thousand constraints and about one million variables on a standard PC in a moderate amount of time.

Key words: Interior-point, Homogeneous, cutting plane, set-partitioning.

*Permanent address: Math Sciences, RPI, Troy NY 12180, USA. Supported in part by fellowships from NWO and TU Delft

1 INTRODUCTION

We are trying to find a point y in a convex set $C \subseteq \Re^m$. We assume that if C is nonempty then it contains a ball of radius ϵ, and we assume that $C \subseteq B := \{y \in \Re^m : -e \leq y \leq e\}$, where e denotes a vector of ones. The set C is defined by a *separating oracle:* given a point in $\Re^m$, the oracle will either state that the point is in C, or it will return a hyperplane that separates the point and C.

We will solve the feasibility problem by solving a sequence of linear programming relaxations. The initial relaxation will correspond to the box B, and this relaxation will be extended by adding separating hyperplanes returned by the oracle. Thus, we will always have the primal-dual pair of linear programming problems:

$$\begin{array}{lrcll} \min & c^T x & & & \\ \text{subject to} & Ax & = & 0 & \qquad (P) \\ & x & \geq & 0 & \end{array}$$

and

$$\begin{array}{lrcccl} \max & 0 & & & & \\ \text{subject to} & A^T y & + & s & = & c \qquad (D) \\ & & & s & \geq & 0, \end{array}$$

where A is an $m \times n$ matrix and c, x, y and s are dimensioned appropriately. We assume $n \geq 2m$ and the constraints $A^T y \leq c$ contain all the constraints of the box B. Any point $y \in C$ will satisfy $A^T y \leq c$ in the current relaxation (D). We assume that the feasible region for (D) is defined by an oracle. Thus, we are going to solve this problem using a column generation method, where we add constraints to (D). We will assume that we always know feasible points $x > 0$ and (y, s) with $s > 0$ for the current relaxation — this assumption will be justified later. We will assume throughout, without loss of generality, that every column of A has norm 1.

Recent papers on interior point cutting plane algorithms include Goffin *et al.* (1996), who described a fully polynomial algorithm which added one constraint at a time, right through the current analytic center. This work has been extended in Ye (1997) and Goffin and Vial (1998), who considered adding many cuts simultaneously; in Ye (1997) the analytic center was recovered using a primal approach, while a primal-dual approach requiring the solution of a nonlinear programming problem was proposed in Goffin and Vial (1998). A simpler primal-dual updating approach with multiple cuts was proposed in Ramaswamy and Mitchell (1994), at the cost of weakening the cuts. Nesterov and Vial (1997) have described a homogeneous analytic center cutting plane algorithm with a very good complexity bound; this method is not a primal-dual

method. Atkinson and Vaidya (1995) described a polynomial cutting plane algorithm that required that unimportant constraints be dropped; this was extended to a long-step method for optimization problems by Mitchell and Ramaswamy (1993). Other work on interior point cutting plane methods includes that of den Hertog *et al.* (1995).

We propose to solve the original linear programming problem by solving the homogenized self-dual linear programming problem

$$
\begin{array}{lrrrrrrcll}
\min & & & & & & \bar{d}\phi & & & \\
\text{subject to} & & Ax & & & & & = & 0 & \\
& -A^T y & & -s & +c\tau & & & = & 0 & (HSDF) \\
& & -c^T x & & & -\kappa & +\bar{d}\phi & = & 0 & \\
& & & & -\bar{d}\tau & & & = & -\bar{d} & \\
& & & x, s, \tau, \kappa & \geq & 0, & & & &
\end{array}
$$

where $\bar{d}$ is a positive constant to be specified later. One novel aspect of our analysis is the use of a different proximity measure than those used in other interior point cutting plane algorithms.

In §2, we give some preliminaries on barrier functions and proximity measures. Our algorithm is described in §3. A method that requires $O(1)$ Newton steps to recover the analytic center after the addition of a cutting plane is described in §4. We show that the algorithm is fully polynomial in §5. In §6 we discuss a practical implementation of the homogenized cutting plane method for solution of LPs. In the following §7 we present our computational results for solving some large-scale set-partioning and set-covering problems. Finally, we offer our conclusions in §8.

Notation:

Given an n-vector x, we use X to denote the diagonal matrix with $X_{ii} = x_i$. The matrix S is defined similarly. As stated earlier, we use e to denote a vector of ones of an appropriate dimension.

2 BARRIER FUNCTIONS AND PROXIMITY MEASURES

The primal-dual logarithmic barrier function we use is

$$\Psi(x, s, \tau, \kappa) := x^T s + \tau\kappa - (n+1) - \sum_{i=1}^{n} \ln(x_i s_i) - \ln(\tau\kappa). \qquad (2.1)$$

Note that we will work throughout with a barrier parameter $\mu = 1$, so this parameter is omitted from our definitions of $\Psi(x, s, \tau, \kappa)$ and $\delta(x, s, \tau, \kappa)$.

It can be shown that $\Psi \geq 0$, with equality only at the analytic center. The analytic center is thus the point that minimizes Ψ, and at the analytic center we have $x_i s_i = 1$ for each i and $\tau\kappa = 1$.

One proximity measure is

$$\delta(x, s, \tau, \kappa) := 0.5 \parallel u - u^{-1} \parallel, \tag{2.2}$$

where $u, u^{-1} \in \Re^{n+1}$, with components indexed from 0 to n, and

$$u_i := \begin{cases} \sqrt{\tau\kappa} & \text{if } i = 0 \\ \sqrt{x_i s_i} & \text{otherwise} \end{cases} \tag{2.3}$$

$$(u^{-1})_i := \frac{1}{u_i}. \tag{2.4}$$

Further, we define the n-vectors $\bar{u}$ and $\bar{u}^{-1} \in \Re^n$:

$$\bar{u}_i = u_i, \quad (\bar{u}^{-1})_i = \frac{1}{u_i} \quad \text{for } i = 1, \ldots, n. \tag{2.5}$$

We will call a feasible point (x, s, τ, κ) *approximately centered* if it satisfies the inequality $\delta(x, s, \tau, \kappa) < \theta$ for some constant θ to be specified later. We will have $0 < \theta < 0.5$.

Define

$$\psi(t) := t - \ln(1 + t) \quad \text{for } t > -1 \tag{2.6}$$

$$\rho(\delta) := \delta + \sqrt{1 + \delta^2}. \tag{2.7}$$

Lemma II.67 in Roos *et al.* (1997) allows us to relate $\Psi(x, s, \tau, \kappa)$ and $\delta(x, s, \tau, \kappa)$:

Lemma 2.1 *Let $\delta := \delta(x, s, \tau, \kappa)$. We have*

$$\psi(-2\delta/\rho(\delta)) \leq \Psi(x, s, \tau, \kappa) \leq \psi(2\delta\rho(\delta)).$$

We also have the following lemma concerning the performance of Newton's method for this algorithm when μ and ϕ are both fixed at one.

Lemma 2.2 *Let $\delta := \delta(x, s, \tau, \kappa)$.*

1. *(Theorem II.49, Roos* et al. *(1997).) If $\delta \leq \frac{1}{\sqrt{2}}$ then the primal-dual Newton step is feasible and $\delta(x^+, s^+, \tau^+, \kappa^+) \leq \delta^2$.*

2. *(Lemma II.70, Roos* et al. *(1997).) If $\delta \geq \frac{1}{\sqrt{2}}$ then the barrier function Ψ is decreased by at least $\frac{1}{6}$ if a damped primal-dual Newton step with a specified step length is taken.*

The global convergence analysis in §5 will use the *dual logarithmic barrier function:*

$$\Psi_D(s) := \sum_{i=1}^{n} \ln(s_i). \tag{2.8}$$

The more usual proximity measure is

$$\| Xs - e \| = \| \bar{U}(\bar{u} - \bar{u}^{-1}) \| .$$

The eigenvalues of $\bar{U}$ are the components of the vector $\bar{u}$; it follows that if $\bar{u}_i < 1 + \zeta$ for each i then $\| Xs - e \| \leq (1 + \zeta) \| \bar{u} - \bar{u}^{-1} \|$.

3 A COLUMN GENERATION METHOD

3.1 Overview:

Assume that the initial dual constraints are given by $-e \leq y \leq e$, where e denotes the vector of ones of the appropriate dimension. Thus, initially, $n = 2m$. The vector $\hat{y} = 0$ is then feasible in (D), and the vector $\hat{x} = e$ is feasible in (P). It should be noted that with this assumption, the dual feasible region is bounded, so if κ is nonzero at the optimal complementary pair then there exists a primal ray, and the primal problem has unbounded optimal value since it is feasible.

In order to satisfy the constraints of $(HSDF)$, we will use $\tau^0 = 1$ and $\kappa^0 = 1$ initially; we can then start with $y^0 = 0$ and $x^0 = e$. We can set $\bar{d} := n+1$. Since we are using a column generation method, $\bar{d}$ will change whenever the number of dual constraints changes. Using $s^0 = e$ this gives $c^T x^0 = n = (x^0)^T s^0$. With $\phi = 1$, we then have an analytic center, with $x_i s_i = 1$ for each component i and $\tau\kappa = 1$ also.

The algorithm can be summarized as follows:

1. Initialize with $\phi = 1$.
2. If the problem has too many dual constraints, STOP with infeasibility.
3. Find an approximate center for the current value of ϕ and with $\mu = 1$.
4. Call the separation oracle at the point $\frac{y}{\tau}$.
5. If cuts are found, add them through the current dual iterate, obtain a new feasible iterate with $x > 0$ and $s > 0$, and return to step 2.
6. Otherwise, STOP with feasibility.

Clearly, we need to define what we mean by "too many dual constraints" in Step 2; this will be discussed in §5. Step 5 is the subject of §4. Notice that ϕ is never changed in this algorithm. This is because we are solving the feasibility problem, and so there is no need to decrease the duality gap by decreasing ϕ.

3.2 A direction for restarting

Our old iterate (x^k, y^k, s^k) is on the boundary of the feasible region in both the primal and the dual spaces. Let the new primal problem be

$$\begin{array}{lrcll} \min & c^T x & + & c_{n+1}x_{n+1} & \\ \text{subject to} & Ax & + & a_{n+1}x_{n+1} = 0 & (P^+) \\ & & & x, x_{n+1} \geq 0 & \end{array}$$

where a_{n+1} is the new primal column, and x_{n+1} is the new primal scalar variable with objective function coefficient $c_{n+1} := a_{n+1}^T y^k$. Note that we add the new constraint in a central position, that is, right through the current dual iterate y^k. The new dual problem is

$$\begin{array}{lrcll} \max & 0 & & & \\ \text{subject to} & A^T y & + & s = c & (D^+) \\ & a_{n+1}^T y & + & s_{n+1} = c_{n+1} & \\ & & & s, s_{n+1} \geq 0, & \end{array}$$

where s_{n+1} is the slack for the additional constraint.

Since we add the constraint right through the current iterate, we can get a primal-dual feasible iterate by taking $x_{n+1} = s_{n+1} = 0$. Unfortunately, this is not an interior point, so we need to take a direction to obtain positive iterates. We can use directions d_x and d_y which move us to the points in the corresponding Dikin ellipsoids that maximize the new variables:

$$d_x = -D_P^2 A^T (A D_P^2 A^T)^{-1} a_{n+1} \tag{3.1}$$

$$d_y = -(A D_D^2 A^T)^{-1} a_{n+1}, \tag{3.2}$$

where D_P and D_D are diagonal matrices. We could use, for example, $D_P = X$, $D_D = S^{-1}$, or $D_P = D_D = D_{PD} := X^{0.5} S^{-0.5}$. We now restrict attention to the primal-dual scaling $D_P = D_D = D_{PD}$. This considerably helps the analysis in §4. Moving in these directions is guaranteed to give an interior point, provided we don't take too large a step. So we update for some positive scalars α_P and α_D:

$$x^+ := x + \alpha_P d_x \tag{3.3}$$

$$x_{n+1} := \alpha_P \tag{3.4}$$
$$y^+ := y + \alpha_D d_y \tag{3.5}$$
$$s^+ := s - \alpha_D A^T d_y \tag{3.6}$$
$$s_{n+1} := \alpha_D a_{n+1}^T (A D_{PD}^2 A^T)^{-1} a_{n+1}. \tag{3.7}$$

Using (3.6) and (3.2), we define

$$d_s := -A^T d_y = A^T (A D_{PD}^2 A^T)^{-1} a_{n+1}. \tag{3.8}$$

Using the orthogonality of $(d_x, 1)$ and $(d_s, d_{s_{n+1}})$, the reader easily verifies the following lemma.

Lemma 3.1 *We have* $d_x^T d_s = -a_{n+1}^T (A D_{PD}^2 A^T)^{-1} a_{n+1}$.

Thus, the quantity

$$\Delta_{PD} := \sqrt{a_{n+1}^T (A D_{PD}^2 A^T)^{-1} a_{n+1}} \tag{3.9}$$

will have an effect on the performance of the algorithm, both through affecting $d_x^T d_s$ and through determining the value of s_{n+1}. Methods that back off the additional constraint usually back it off by a quantity proportional to Δ_{PD}, or to the equivalent quantity using a different scaling matrix. (See, for example, Atkinson and Vaidya (1995) and Goffin *et al.* (1993).)

3.3 Bounds on the step lengths

The algorithm has found an approximate analytic center for $\phi = 1$, $\mu = 1$ when it adds a constraint. Therefore, the current iterates satisfy

$$0.5 \parallel u - u^{-1} \parallel \leq \theta \tag{3.10}$$

for some positive scalar $\theta < 0.5$, where u and u^{-1} are defined in equation (2.3).

Lemma 3.2 *If the iterates are approximately centered, then*

$$(1 - 2\theta) \leq x_i s_i \leq 1 + 2\theta(2\theta + 1) \tag{3.11}$$

for each component i. Further, the duality gap is bounded:

$$n - 2\theta(2\theta + 1) \leq x^T s \leq n + 2\theta.$$

Proof: We have

$$\begin{aligned} 4\theta^2 &\geq (u-u^{-1})^T(u-u^{-1}) \\ &= u^Tu + (u^{-1})^T(u^{-1}) - 2(n+1) \\ &= (\sqrt{\tau\kappa} - \frac{1}{\sqrt{\tau\kappa}})^2 + \sum_{i=1}^{n}(x_is_i + \frac{1}{x_is_i} - 2). \end{aligned}$$

Since each term in the summand is nonnegative, each term must be less than the left hand side. This implies the following, multiplying by the positive term x_is_i.

$$(x_is_i)^2 - (2+4\theta^2)x_is_i + 1 \leq 0$$

The roots of this quadratic are

$$1 + 2\theta^2 \pm \sqrt{(1+2\theta^2)^2 - 1} = 1 + 2\theta^2 \pm 2\theta\sqrt{\theta^2+1},$$

so x_is_i must lie between these two roots. We can simplify these expressions slightly by observing that $\sqrt{\theta^2+1} \leq \theta + 1$. We thus get the bounds:

$$1 - 2\theta \leq x_is_i \leq 1 + 2\theta(2\theta+1).$$

To bound the duality gap, notice that $\kappa\tau$ must also satisfy these inequalities. Since $\tau = 1$, we obtain a restriction on κ. From the definition of $(HSDF)$, we obtain

$$c^Tx = -\kappa + \bar{d}\phi = n + 1 - \kappa,$$

since $\bar{d} = n+1$ and $\phi = 1$. The result follows. □

If $\alpha_P > 0$ then $x_{n+1} > 0$. To ensure that the other components of x remain positive, it is necessary to keep $x + \alpha_P d_x > 0$. We need positivity of the dual slacks as well, so we need $s + \alpha_D ds > 0$.

Lemma 3.3 *The following step length choices ensure positive iterates.*

1. *If* $0 < \alpha_P < \frac{\sqrt{1-2\theta}}{\Delta_{PD}}$ *then* $x + \alpha_P dx > 0$.
2. *If* $0 < \alpha_D < \frac{\sqrt{1-2\theta}}{\Delta_{PD}}$ *then* $s + \alpha_D ds > 0$.

Proof: To prove part 1, we have

$$\begin{aligned} x + \alpha_P d_x &= Xe - \alpha_P D_{PD}^2 A^T(AD_{PD}^2A^T)^{-1}a_{n+1} \qquad \text{from (3.1)} \\ &= D_{PD}(X^{0.5}S^{0.5}e - \alpha_P D_{PD}A^T(AD_{PD}^2A^T)^{-1}a_{n+1}) \\ &\geq D_{PD}(\sqrt{1-2\theta} - \alpha_P\Delta_{PD})e \end{aligned}$$

where the last line follows from Lemma 3.2 and equation (3.9). Similarly, to show part 2, we have

$$\begin{aligned} s + \alpha_D d_s &= Se + \alpha_D A^T (AD_{PD}^2 A^T)^{-1} a_{n+1} \qquad \text{from (3.8)} \\ &= D_{PD}^{-1}(X^{0.5}S^{0.5}e + \alpha_P D_{PD} A^T (AD_{PD}^2 A^T)^{-1} a_{n+1}) \\ &\geq D_{PD}(\sqrt{1-2\theta} - \alpha_P \Delta_{PD})e. \end{aligned}$$

The result follows. □

3.4 Adding multiple cuts

If we add more than one constraint, we can still get a new primal interior point fairly easily as in Mitchell and Todd (1992); Ye (1997):

1. Add together the additional columns to give a new column $\bar{a}$ and a corresponding variable $\bar{x}_{n+1}$.

2. Generate the direction d_x given above with $\bar{a}$ taking the place of a_{n+1} and $\bar{x}_{n+1}$ taking the place of x_{n+1}.

3. Disaggregate this additional variable, and set each of the new variables equal to the value of $\bar{x}_{n+1}$.

Unfortunately, it is not so easy to restart with the dual problem when adding multiple constraints. If only two constraints are added, then we can take d_y to be the average of the directions that would be calculated if only one of the constraints was added. If more than two constraints are added, one way to find a direction is by solving a small linear programming feasibility problem:

> Let $\bar{A}$ consist of the additional columns of A. Find a direction $\eta \geq e$ satisfying $\bar{A}^T (AD_D^2 A^T)^{-1} \bar{A}\eta \geq e$.

Notice that this is an LP in p variables, with p constraints, where p is the number of additional variables. Solving it with an interior point algorithm should result in a well-scaled direction if all the slacks are approximately equal. An appropriate direction is $d_y = -\bar{A}\eta$. Such a method is proposed in Ramaswamy and Mitchell (1994). A related, more complicated approach that requires approximately solving a nonlinear programming problem in p variables is proposed in Goffin and Vial (1998). The approach proposed in Goffin and Vial (1998) allows the recovery of an approximate analytic center for the modified problem in $O(p \ln(p))$ Newton steps.

4 CHOOSING A RESTART POINT

If the step lengths are chosen small enough, then the new x and s will still be approximately centered for the original n components. For the last component, we show that it is possible to bound an appropriate quantity, and that the distance from optimality will then be bounded by a constant. We can keep $\tau^+ = 1$. The constant $\bar{d}$ is increased by one, since the number of dual constraints is increased. We then set

$$\kappa^+ = \bar{d} - c^T x^+ - c_{n+1} x_{n+1} = \bar{d} - (x^+)^T s^+ - x_{n+1} s_{n+1}. \tag{4.1}$$

We will see that this will enable us to regain centrality in $O(1)$ iterations. The proof takes the following form:

- After taking our restart step to an interior point, the proximity measure is bounded by a constant. We prove this in §4.1.
- This implies, from Lemma 2.1 that the primal-dual logarithmic barrier function is also bounded by a constant.
- From Lemma 2.2, the logarithmic barrier function decreases by a constant as long as δ is large. Once δ is small, it will converge quadratically to zero. Therefore, a new approximate center can be found in $O(1)$ iterations. We discuss this further in §4.2.

4.1 Updating the measure of centrality

We had an approximately centered iterate before adding the constraint, so the proximity measure $\delta(x, s, \tau, \kappa) \leq \theta$, say. We define

$$\bar{\delta} := \| \bar{u} - \bar{u}^{-1} \| \tag{4.2}$$

for this approximate center. Note that

$$\delta(x, s, \tau, \kappa)^2 = 0.5 \left[\bar{\delta}^2 + (\sqrt{\tau\kappa} - \frac{1}{\sqrt{\tau\kappa}})^2 \right]. \tag{4.3}$$

After updating, the measure of centrality becomes

$$\begin{aligned} (\delta^+)^2 &:= (\delta((x^+, x_{n+1}), (s^+, s_{n+1}), \tau^+, \kappa^+))^2 \\ &= 0.5 \Bigg[\| \bar{u}^+ - (\bar{u}^+)^{-1} \|^2 + (x_{n+1} s_{n+1} + \frac{1}{x_{n+1} s_{n+1}} - 2) \\ &\qquad + (\kappa^+ + \frac{1}{\kappa^+} - 2) \Bigg], \end{aligned} \tag{4.4}$$

where $\bar{u}^+$ and $(\bar{u}^+)^{-1}$ are defined in the natural way using equations (2.3–2.5).

It will prove useful to use the same step length in the primal and dual problems, particularly in Lemma 4.2. Therefore, we take

$$\alpha_P = \alpha_D = \alpha = \frac{\beta}{\Delta_{PD}} \tag{4.5}$$

for an absolute constant β to be specified later, with $0 < \beta < \sqrt{1-2\theta}$.

Lemma 4.1 *After taking the specified step, the term in equation (4.4) corresponding to the new dual constraint satisfies*

$$x_{n+1}s_{n+1} + \frac{1}{x_{n+1}s_{n+1}} - 2 = \beta^2 + \frac{1}{\beta^2} - 2.$$

Proof: From equations (3.4), (3.7), and (4.5), we have

$$x_{n+1}s_{n+1} = \alpha_P \alpha_D \Delta_{PD}^2 = \beta^2.$$

The result follows. □

Let us define the vector

$$v := D_{PD}A^T(AD_{PD}^2A^T)^{-1}a_{n+1}. \tag{4.6}$$

Notice that

$$v^T v = \Delta_{PD}^2 \tag{4.7}$$

and that

$$d_x = -D_{PD}v \text{ and } d_s = D_{PD}^{-1}v. \tag{4.8}$$

This observation enables us to simplify the expression for $\| \bar{u}^+ - (\bar{u}^+)^{-1} \|$ considerably, as shown in the next two lemmas.

Lemma 4.2 *The ith component of the complementary slackness is reduced to* $x_i^+ s_i^+ = x_i s_i - \alpha^2 v_i^2$ *for* $i = 1, \ldots, n$.

Proof: We have

$$\begin{aligned} x_i^+ s_i^+ &= (x_i + \alpha(d_x)_i)(s_i + \alpha(d_s)_i) \\ &= x_i s_i + \alpha x_i^{0.5} s_i^{0.5} v_i - \alpha x_i^{0.5} s_i^{0.5} v_i - \alpha^2 v_i^2 \\ &= x_i s_i - \alpha^2 v_i^2. \end{aligned}$$

□

Lemma 4.3 *We have the following bound provided* $0 < \beta < \sqrt{1-2\theta}$*:*

$$\| \bar{u}^+ - (\bar{u}^+)^{-1} \|^2 \leq \bar{\delta}^2 + \gamma,$$

where

$$\gamma := \beta^2 \left(\frac{1}{(1-2\theta)(1-2\theta-\beta^2)} - 1 \right).$$

Proof: We have

$$\begin{aligned}
\| \bar{u}^+ - (\bar{u}^+)^{-1} \|^2 &= \sum_{i=1}^{n} \left[\sqrt{x_i s_i - \alpha^2 v_i^2} - \frac{1}{\sqrt{x_i s_i - \alpha^2 v_i^2}} \right]^2 \\
&= \sum_{i=1}^{n} \left[x_i s_i - \alpha^2 v_i^2 + \frac{1}{x_i s_i - \alpha^2 v_i^2} - 2 \right] \\
&= \sum_{i=1}^{n} (x_i s_i + \frac{1}{x_i s_i} - 2) - \alpha^2 \| v \|^2 \\
&\qquad + \sum_{i=1}^{n} \left[\frac{1}{x_i s_i - \alpha^2 v_i^2} - \frac{1}{x_i s_i} \right] \\
&= \bar{\delta}^2 - \alpha^2 \Delta_{PD} + \sum_{i=1}^{n} \frac{\alpha^2 v_i^2}{(x_i s_i - \alpha^2 v_i^2)(x_i s_i)} \qquad \text{by (4.7)} \\
&\leq \bar{\delta}^2 - \beta^2 + \frac{\beta^2}{(1-2\theta)(1-2\theta-\alpha^2\Delta_{PD})} \\
&\qquad \text{by (4.5), (4.7) and Lemma 3.2.}
\end{aligned}$$

The result follows from a further application of (4.5). □

We can also bound the term in the proximity measure due to $\tau^+ = 1$ and κ^+.

Lemma 4.4 *If* $0 < \beta < \sqrt{1-2\theta}$ *then*

$$\kappa^+ + \frac{1}{\kappa^+} - 2 \leq 1 + (\kappa + \frac{1}{\kappa} - 2).$$

Proof: Now,

$$\begin{aligned}
\kappa^+ &= n + 2 - (x^+)^T s^+ - x_{n+1} s_{n+1} \qquad \text{from (4.1)} \\
&= n + 2 - (x^T s - \beta^2) - \beta^2 \qquad \text{from Lemmas 4.2 and 4.1} \\
&= n + 2 - x^T s \\
&= 1 + \kappa \qquad \text{from } (HSDF) \text{ without the new constraint.}
\end{aligned}$$

Therefore,

$$\kappa^+ + \frac{1}{\kappa^+} = 1 + \kappa + \frac{1}{1+\kappa} < 1 + \kappa + \frac{1}{\kappa}.$$

The result follows. □

It is now possible to bound δ^+.

Theorem 4.1 *If* $\beta = \sqrt{0.5(1-2\theta)}$ *then* $(\delta^+)^2 \leq \theta^2 + 0.5(-1 + \frac{3}{1-2\theta})$.

Proof: From equation (4.4) and Lemmas 4.1, 4.3, and 4.4, we have

$$\begin{aligned}
(\delta^+)^2 &= 0.5\left[\| \bar{u}^+ - (\bar{u}^+)^{-1} \|^2 + (x_{n+1}s_{n+1} + \frac{1}{x_{n+1}s_{n+1}} - 2) + (\kappa^+ + \frac{1}{\kappa^+} - 2)\right] \\
&\leq 0.5\left[\bar{\delta}^2 + \gamma + \beta^2 + \frac{1}{\beta^2} - 2 + 1 + \kappa + \frac{1}{\kappa} - 2\right] \\
&= (\delta(x,s,\tau,\kappa))^2 + 0.5\left[\gamma + \beta^2 + \frac{1}{\beta^2} - 1\right] \qquad \text{from (4.3).}
\end{aligned}$$

Now,

$$\begin{aligned}
\gamma + \beta^2 + \frac{1}{\beta^2} &= \beta^2\left[\frac{1}{(1-2\theta)(1-2\theta-\beta^2)} - 1\right] + \beta^2 + \frac{1}{\beta^2} \\
&= \frac{\beta^2}{(1-2\theta)(1-2\theta-\beta^2)} + \frac{1}{\beta^2} \\
&= \frac{3}{1-2\theta}
\end{aligned}$$

with the specified choice of β. The result follows. □

4.2 Obtaining a new approximate analytic center

From Theorem 4.1 and Lemma 2.1, the logarithmic barrier function value at the restart point is bounded above by a constant:

$$\Psi^+ := \Psi((x^+, x_{n+1}), (s^+, s_{n+1}), \tau^+, \kappa^+) \leq \psi(2\delta^+\rho(\delta^+)).$$

Notice that if we have an *exact* analytic center with $\theta = 0$ then the proximity measure for the new iterate is no larger than 1, and $\Psi^+ \leq 3.066$. The bound on the proximity measure is achieved if just one component of v is nonzero, in which case the new iterate has $x_{n+1}s_{n+1} = 0.5$, $x_i s_i = 0.5$ for exactly one other component, and $x_i s_i = 1$ for the remaining components. One possible choice for an approximate analytic center is $\theta = 0.25$, which gives $\delta^+ \leq 1.601$ and $\Psi^+ \leq 8.672$.

By using damped Newton steps while $\delta > \frac{1}{\sqrt{2}}$ and full Newton steps thereafter, we can then retrieve an approximate center in $O(1)$ steps.

5 GLOBAL CONVERGENCE

The analysis of Goffin *et al.* (1996) can be used directly to analyze the performance of this algorithm. In Goffin *et al.* (1996), upper and lower bounds are placed on the value of the dual logarithmic barrier function evaluated at the exact analytic center. After a fully polynomial number of iterations, the bounds cross. The lower bound uses the assumption that C contains a ball of radius ϵ. Thus, if the bounds cross then this assumption must be violated, so the algorithm can terminate with infeasibility.

From Lemma 3.2 and the discussion in §2, if an iterate is approximately centered for an appropriate θ in our framework, then it is also approximately centered for an appropriate centering parameter with the standard proximity measure. We add cuts in exactly the same manner as in Goffin *et al.* (1996), so the dual analytic center will behave identically. Thus, we require the same fully polynomial number of cutting planes as in Goffin *et al.* (1996). This gives the following theorem:

Theorem 5.1 *The upper bound on the number of constraints that need to be added before the algorithm can be terminated with infeasibility is* $O(\frac{m^2}{\epsilon^2})$*, and the number of Newton steps required has the same complexity.*

6 AN IMPLEMENTATION

The purpose of this section is to present a practical implementation of the homogenized cutting plane method. The practical implementation differs from the theoretical in several ways, because it has been adapted to solve linear programs directly and not only the feasibility problem.

Hence, the problem of study is the primal LP

$$\begin{array}{lllll} \min & c^T x & & \\ \text{subject to} & Ax & = & b, \\ & x & \geq & 0 \end{array} \tag{6.1}$$

and the dual problem

$$\begin{array}{ll} \max & b^T y \\ \text{subject to} & A^T y \leq c, \end{array} \tag{6.2}$$

where A is an $m \times n$ matrix. We assume that c, b, x, and y are appropriately dimensioned.

In general we will assume that $n \gg m$ which for example is the case for LP relaxations of set-partitioning and set-covering problems occurring in practice, see Bixby *et al.* (1992) for an example. In that case it is impractical to solve (6.2) directly or may be impossible due to memory requirements.

The homogeneous and self-dual linear programming formulation (see Roos *et al.* (1997)) of the linear program (6.1) is

$$\begin{array}{lrrrrrrl} \min & & & & & \bar{g}\phi & & \\ \text{subject to} & & Ax & -b\tau & -\bar{b}\phi & = & 0, \\ & -A^Ty & & +c\tau & +\bar{c}\phi & \geq & 0, \\ & b^Ty & -c^Tx & & +\bar{d}\phi & \geq & 0, \\ & \bar{b}^Ty & -\bar{c}^Tx & -\bar{d}\tau & & = & -\bar{g}, \\ & & x \geq 0, & \tau \geq 0, & & & \end{array} \tag{6.3}$$

where

$$\begin{aligned} \bar{b} &:= Ax^0 - b\tau^0, \\ \bar{c} &:= -c\tau^0 + A^Ty^0 + s^0, \\ \bar{d} &:= c^Tx^0 - b^Ty^0 + \kappa^0, \\ \bar{g} &:= (x^0)^Ts^0 + \tau^0\kappa^0. \end{aligned}$$

One obvious choice is

$$(y^0, x^0, \tau^0, s^0, \kappa^0, \phi^0) = (0, e, 1, e, 1, 1), \tag{6.4}$$

where e is a vector of appropriate dimension containing all ones and subsequently it is the choice we will use. We have the following lemma:

Lemma 6.1 *The problem (6.3) always has an optimal solution with zero objective value. Let $(y^*, x^*, \tau^*, \phi^*)$ be any maximally complementary solution of (6.3), then $\tau^* > 0$ if and only if* (6.1) *has an optimal solution. Moreover, x^*/τ^* is an optimal solution to* (6.1).

The central path for the homogeneous model (6.3) is defined by

$$\begin{array}{rrrrrcl} & Ax & -b\tau & -\bar{b}\phi & & = & 0, \\ -A^Ty & & +c\tau & +\bar{c}\phi & -s & = & 0, \\ b^Ty & -c^Tx & & +\bar{d}\phi & -\kappa & = & 0, \\ \bar{b}^Ty & -\bar{c}^Tx & -\bar{d}\tau & & & = & -\bar{g}, \\ & & & & Xs & = & \rho\frac{\bar{g}}{n+1}e, \\ & & & & \tau\kappa & = & \rho\frac{\bar{g}}{n+1}, \\ & x \geq 0, & \tau \geq 0, & & s, \kappa \geq 0, & & \end{array} \tag{6.5}$$

where ρ is an additional parameter. Note we have added the slack variables s and κ. It can be proved that this system of linear and nonlinear equations has an unique solution for all positive values of ρ. Moreover, the solution to this system converge to a maximally complementary solution of (6.3) for ρ converging to zero.

Now let $(y, x, \tau, s, \kappa, \phi)$ be any solution to (6.5) for a given ρ, then we have that

$$\begin{aligned} 0 &= y^T(Ax - b\tau - \bar{b}\phi) + x^T(-A^T y + c\tau + \bar{c}\phi - s) \\ &\quad +\tau(b^T y - c^T x + \bar{d}\phi - \kappa) + \phi(\bar{b}^T y - \bar{c}^T x - \bar{d}\tau + \bar{g}) \\ &= -x^T s - \tau\kappa + \phi\bar{g} \\ &= -e^T X s - \tau\kappa + \phi\bar{g} \\ &= -\rho\bar{g} + \phi\bar{g} \\ &= (\phi - \rho)\bar{g} \end{aligned} \tag{6.6}$$

which implies that $\phi = \rho$ for all solutions on the central path. Moreover, it is easy to verify that

$$-x^T s - \tau\kappa = \phi(\bar{b}^T y - \bar{c}^T x - \bar{d}\tau) \tag{6.7}$$

which implies

$$\begin{aligned} \bar{b}^T y - \bar{c}^T x - \bar{d}\tau &= -\tfrac{1}{\phi}(x^T s + \tau\kappa) \\ &= -\tfrac{\rho\bar{g}}{\phi} \\ &= -\bar{g}. \end{aligned} \tag{6.8}$$

Hence, for any solution on the central path, the fourth equation in (6.5) is redundant. Therefore, the central path equations (6.5) can be reduced to

$$\begin{array}{rrrrcl} Ax & -b\tau & & = & \phi\bar{b}, \\ -A^T y & +c\tau & -s & = & -\phi\bar{c}, \\ b^T y & -c^T x & -\kappa & = & -\phi\bar{d}, \\ & & Xs & = & \phi\frac{\bar{g}}{n+1}e, \\ & & \tau\kappa & = & \phi\frac{\bar{g}}{n+1}, \\ x \geq 0, & \tau \geq 0, & s, \kappa \geq 0, & & \end{array} \tag{6.9}$$

where ϕ is now seen as a parameter which should be reduced to zero. It should be noted that the elimination of the redundant equation in the central path definition also leads to more efficient computation of the search direction to be presented later.

Given our assumptions it will be costly to solve (6.9) using the usual path following method, because n is large. Therefore, a cutting plane (or a column generation) method is devised. Now let

$$\mathcal{J} \subseteq \{1, \ldots, n\}$$

and define the complementary set

$$\bar{\mathcal{J}} := \{1, \ldots, n\} \setminus \mathcal{J}. \tag{6.10}$$

If all the variables in $\bar{\mathcal{J}}$ are fixed i.e.

$$x_{\bar{\mathcal{J}}} = \phi \frac{\bar{g}}{n+1} \tag{6.11}$$

and hence the dual variables $s_{\bar{\mathcal{J}}}$ are free variables, then a reduced central path problem

$$\begin{array}{cccccl} & A_{\mathcal{J}} x_{\mathcal{J}} & -b\tau & & = & \phi(A_{\mathcal{J}} x^0_{\mathcal{J}} - b\tau^0), \\ -A^T_{\mathcal{J}} y & & +c_{\mathcal{J}}\tau & -s_{\mathcal{J}} & = & -\phi(-c_{\mathcal{J}}\tau^0 + A^T_{\mathcal{J}} y^0 + s^0_{\mathcal{J}}), \\ b^T y & -(c_{\mathcal{J}})^T x_{\mathcal{J}} & & -\kappa & = & -\phi(b^T y^0 - c^T_{\mathcal{J}} x^0_{\mathcal{J}} + \kappa^0), \\ & & & X_{\mathcal{J}} s_{\mathcal{J}} & = & \phi \frac{\bar{g}}{n+1} e, \\ & & & \tau\kappa & = & \phi \frac{\bar{g}}{n+1} e, \\ & x_{\mathcal{J}} \geq 0, & \tau \geq 0, & s_{\mathcal{J}}, \kappa \geq 0, & & \end{array} \tag{6.12}$$

is obtained. Note that this problem only requires information about the variables in $\mathcal{J}$. Moreover, given that $|\mathcal{J}|$ is not too large, then an approximate solution to this problem can be computed cheaply. By an approximate solution we mean a solution $(y, x_{\mathcal{J}}, \tau, s_{\mathcal{J}}, \kappa)$ that satisfies

$$\begin{array}{cccccl} & A_{\mathcal{J}} x_{\mathcal{J}} & -b\tau & & = & \phi(A_{\mathcal{J}} x^0_{\mathcal{J}} - b\tau^0), \\ -A^T_{\mathcal{J}} y & & +c_{\mathcal{J}}\tau & -s_{\mathcal{J}} & = & -\phi(-c_{\mathcal{J}}\tau^0 + A^T_{\mathcal{J}} y^0 + s^0_{\mathcal{J}}), \\ b^T y & -(c_{\mathcal{J}})^T x_{\mathcal{J}} & & -\kappa & = & -\phi(b^T y^0 - c^T_{\mathcal{J}} x^0_{\mathcal{J}} + \kappa^0), \\ & x_{\mathcal{J}} \geq 0, & \tau \geq 0, & s_{\mathcal{J}}, \kappa \geq 0, & & \end{array} \tag{6.13}$$

and belongs to the set

$$\mathcal{N}(\mathcal{J}, \phi, \theta) := \left\{ (y, x_{\mathcal{J}}, \tau, s_{\mathcal{J}}, \kappa) : \begin{array}{l} \min_{j \in \mathcal{J}} (x_j s_j, \tau\kappa) \geq \theta\phi \frac{\bar{g}}{n+1}, \\ x^T_{\mathcal{J}} s_{\mathcal{J}} + \tau\kappa = \phi \frac{\bar{g}}{n+1} |\mathcal{J}| \end{array} \right\}$$

for some $\theta \in (0, 1)$. We call such a solution an approximate analytic center solution. Note, we do not use the proximity measure suggested in Section 2, but rather a relaxed variant. Hence, the practical algorithm does not follow the central path as closely as the theoretical one.

If $(\hat{y}, \hat{x}_{\mathcal{J}}, \hat{\tau}, \hat{s}_{\mathcal{J}})$ is an approximate analytic center solution, then it can be extended to a solution of the full problem by defining $\hat{x}_{\bar{\mathcal{J}}}$ by (6.11) and

$$\hat{s}_{\bar{\mathcal{J}}} := c_{\bar{\mathcal{J}}}\hat{\tau} - A^T \hat{y} + \phi(-c_{\bar{\mathcal{J}}}\tau^0 + A^T_{\bar{\mathcal{J}}} y^0 + s^0_{\bar{\mathcal{J}}}).$$

Clearly, nothing guarantees that $\hat{s}_{\bar{\mathcal{J}}} \geq 0$ i.e. the set

$$\bar{\mathcal{J}}^- := \{ j \in \mathcal{J} : \ \hat{s}_j < 0 \}$$

is nonempty. However, if $\bar{\mathcal{J}}^-$ is nonempty, then some of the variables in $\bar{\mathcal{J}}^-$ should be moved to $\mathcal{J}$. This corresponds to generating variables in the primal

and cutting planes in the dual. On the other hand if $\bar{\mathcal{J}}$ is empty, then ϕ is reduced. In both cases a new approximate analytic center solution is computed to the new relaxations. This process is continued until ϕ is reduced sufficiently, because then an approximate solution to (6.5) has been computed. This method stated formally leads to the algorithm:

Algorithm 6.1

1. ***Input:*** $\mathcal{J}^0 \subseteq \{1, ..., n\},\ \beta \in (0, 1),\ \varepsilon > 0$
2. $\phi^0 := 1,\ (x^0, \tau^0, y^0, s^0, \kappa^0) := (e, 1, 0, e, 1)$
3. $k := 0$
4. ***while*** $\phi^k > \beta\varepsilon$
5. $(x^{k+1}, \tau^{k+1}, y^{k+1}, s^{k+1}, \kappa^{k+1}) :=$ ***center***$(\mathcal{J}^k, \phi^k, (x^k, \tau^k, y^k, s^k, \kappa^k))$.
6. $\bar{\mathcal{J}}^- := \{j \notin \mathcal{J}^k : s_j^{k+1} < 0\}$
7. ***if*** $\bar{\mathcal{J}}^- \neq \emptyset$
8. ***Choose*** $\mathcal{J}^{k+1}$ ***such that*** $\mathcal{J}^k \subset \mathcal{J}^{k+1} \subseteq \mathcal{J}^k \cup \bar{\mathcal{J}}^-$
9. $\phi^{k+1} := \phi^k$
10. ***else***
11. $\mathcal{J}^{k+1} := \mathcal{J}^k$
12. $\phi^{k+1} := \beta\phi^k$
13. ***end if***
14. $k := k + 1$
15. ***end while***

The procedure "center" used in step 5 means that the approximate analytic center is computed to the reduced center problem defined by $\mathcal{J}^k$. Also note that whenever ϕ is reduced in step 12, then the value of the "inactive" variables is reduced.

Algorithm 6.1 converges in a number iterations bounded by $O(n\frac{\ln(\epsilon)}{\ln(\beta)})$ because in each iteration either ϕ is reduced by the factor β or the set $\mathcal{J}$ is extended. Moreover, the size of $\mathcal{J}$ is bounded by n and ϕ is bounded below by the termination tolerance ε.

6.1 Recomputing the approximate analytic center

One important issue not addressed in Algorithm 6.1 is how to compute the approximate center solution in step 5. Clearly, whenever the approximate analytic center is recomputed after ϕ has been reduced or $\mathcal{J}$ has been extended, then the previous approximate analytic center should serve as a warm-start.

In both cases we are seeking an approximate analytic center solution satisfying

$$\begin{array}{ccccc} A_{\mathcal{J}}x_{\mathcal{J}} & -b\tau & & = & \phi(A_{\mathcal{J}}x^0_{\mathcal{J}} - b\tau^0), \\ -A^T_{\mathcal{J}}y & +c_{\mathcal{J}}\tau & -s_{\mathcal{J}} & = & \phi(-c_{\mathcal{J}}\tau^0 + A^T_{\mathcal{J}}y^0 + s^0_{\mathcal{J}}), \\ b^Ty \quad -(c_{\mathcal{J}})^Tx_{\mathcal{J}} & & -\kappa & = & \phi(b^Ty^0 - c^T_{\mathcal{J}}x^0_{\mathcal{J}} + \kappa^0). \end{array} \tag{6.14}$$

Moreover, the complementarity gap should satisfy

$$(x_{\mathcal{J}})^Ts_{\mathcal{J}} + \tau\kappa = \phi((x^0_{\mathcal{J}})^Ts^0_{\mathcal{J}} + \tau^0\kappa^0). \tag{6.15}$$

and the new iterate should be close to the central path i.e.

$$\min_{j\in\mathcal{J}}(x_js_j, \tau\kappa) \geq \theta\phi\frac{\bar{g}}{n+1} \tag{6.16}$$

for $\phi = \phi^{k+1}$ and the given $\theta \in (0,1)$. The equations (6.14) and (6.15) define a target for the residuals and the complementarity gap respectively, whereas the equation (6.16) is the centering condition.

In the case ϕ is reduced, then a solution satisfying (6.14) and (6.16) is known for $\phi = \phi^k$. Let this solution be denoted as $(y', x'_{\mathcal{J}}, \tau', s'_{\mathcal{J}}, \kappa')$ and let ϕ' denote the corresponding ϕ. Hence, we want to compute a new approximate analytic center for a smaller ϕ and for that purpose we use the search direction

$$\begin{array}{ccccc} A_{\mathcal{J}}d_{x_{\mathcal{J}}} & -bd_\tau & & = & \eta\phi'(A_{\mathcal{J}}x^0_{\mathcal{J}} - b\tau^0), \\ -A^T_{\mathcal{J}}d_y & +c_{\mathcal{J}}d_\tau & -d_{s_{\mathcal{J}}} & = & -\eta\phi'(-c_{\mathcal{J}}\tau^0 + A^T_{\mathcal{J}}y^0 + s^0_{\mathcal{J}}), \\ b^Td_y \quad -(c_{\mathcal{J}})^Td_{x_{\mathcal{J}}} & & -d_\kappa & = & -\eta\phi'(b^Ty^0 - c^T_{\mathcal{J}}x^0_{\mathcal{J}} + \kappa^0), \\ S'_{\mathcal{J}}d_{x_{\mathcal{J}}} & & +X'_{\mathcal{J}}d_{s_{\mathcal{J}}} & = & -X'_{\mathcal{J}}s'_{\mathcal{J}} + \gamma\phi'\frac{\bar{g}}{n+1}e, \\ & \kappa d_\tau & +\tau d_\kappa & = & -\tau'\kappa' + \gamma\phi'\frac{\bar{g}}{n+1}e, \end{array} \tag{6.17}$$

suggested in Xu *et al* (1996). Here, $\gamma, \eta \in [0,1]$ are two parameters to be chosen later. After the search direction has been computed, then a step-size α is chosen and the current solution is updated as follows

$$\begin{bmatrix} x^+_{\mathcal{J}} \\ \tau^+ \\ y^+ \\ s^+_{\mathcal{J}} \\ \kappa^+ \end{bmatrix} = \begin{bmatrix} x'_{\mathcal{J}} \\ \tau' \\ y' \\ s'_{\mathcal{J}} \\ \kappa' \end{bmatrix} + \alpha \begin{bmatrix} d_{x_{\mathcal{J}}} \\ d_\tau \\ d_y \\ d_{s_{\mathcal{J}}} \\ d_\kappa \end{bmatrix}. \tag{6.18}$$

It is easy to verify that

$$\begin{array}{ccccc} A_{\mathcal{J}}x^+_{\mathcal{J}} & -b\tau^+ & & = & (1-\alpha\eta)\phi'(A_{\mathcal{J}}x^0_{\mathcal{J}} - b\tau^0), \\ -A^T_{\mathcal{J}}y^+ & +c_{\mathcal{J}}\tau^+ & -s^+_{\mathcal{J}} & = & -(1-\alpha\eta)\phi'(-c_{\mathcal{J}}\tau^0 + A^T_{\mathcal{J}}y^0 + s^0_{\mathcal{J}}), \\ b^Ty^+ \quad -(c_{\mathcal{J}})^Tx^+_{\mathcal{J}} & & -\kappa^+ & = & -(1-\alpha\eta)\phi'(b^Ty^0 - c^T_{\mathcal{J}}x^0_{\mathcal{J}} + \kappa^0), \end{array}$$

and

$$(x_{\mathcal{J}}^{+})^T s_{\mathcal{J}}^{+} + \tau^{+}\kappa^{+} = (1 - \alpha(1-\gamma) + \alpha^2\eta(1-\eta-\gamma))\phi'((x_{\mathcal{J}}^{0})^T s_{\mathcal{J}}^{0} + \tau^{0}\kappa^{0}).$$

Given $\alpha > 0$ and the choice $\eta = 1-\gamma > 0$, then $(x_{\mathcal{J}}^{+}, \tau^{+}, y, s_{\mathcal{J}}^{+}, \kappa^{+})$ is a solution to (6.14) and (6.15) for

$$\phi = (1 - \alpha(1-\gamma))\phi' = (1-\alpha\eta)\phi' =< \phi'.$$

This demonstrates that the updated solution is closer to the target solution. In general it might not be possible to reach the target in one step, because the step size α should be chosen such that $(y^{+}, x^{+}, \tau^{+}, s^{+}, \kappa^{+})$ is not too far from the central path. In our implementation we use a fairly small γ and let intermediate iterates move far from the central path. Therefore, whenever a solution satisfying (6.14) for the target ϕ has been computed, then several (centering) steps using $\gamma = 1$ and $\eta = 0$ may be required.

Note that the search direction given by (6.17) is a highly flexible direction. For example if a solution is known satisfying (6.14), then by choosing $\eta = 0$ and γ either smaller or greater than 1, then the complementarity gap can be reduced or increased respectively. Moreover, the new solution keeps on satisfying the three first equations in (6.14).

In summary, whenever ϕ is reduced in Algorithm 6.1, then it is easy to perform an efficient warmstart. However, in the case the set $\mathcal{J}$ is extended then the situation is more difficult, because the current solution is not feasible i.e. some of the elements in $s_{\mathcal{J}}$ are negative. In that case we define an intermediate solution

$$\begin{array}{rcl} \phi' & := & \phi, \\ y' & := & y, \\ x'_{\mathcal{J}} & := & x_{\mathcal{J}}, \\ x'_{\mathcal{J}^-} & := & \phi' x^0_{\mathcal{J}^-}, \\ \tau' & := & \tau, \\ s'_{\mathcal{J}'} & := & s_{\mathcal{J}}, \\ s'_{\mathcal{J}^-} & := & \frac{\bar{g}}{n+1}(X^0_{\mathcal{J}^-})^{-1}e, \\ \kappa' & := & \kappa, \end{array} \tag{6.19}$$

where $(y, x_{\mathcal{J}}, \tau, s_{\mathcal{J}}, \kappa)$ was the previous approximate analytic center and the search direction

$$\begin{array}{cccccl}
 & A_{\mathcal{J}} d_{x_{\mathcal{J}}} & -b d_\tau & & = & \phi^k (A_{\mathcal{J}} x^0_{\mathcal{J}} - b\tau^0) \\
 & & & & & -(A_{\mathcal{J}} x'_{\mathcal{J}} - b\tau'), \\
-A^T_{\mathcal{J}} d_y & & +c_{\mathcal{J}} d_\tau & -d_{s_{\mathcal{J}}} & = & \phi^k (-c_{\mathcal{J}} \tau^0 + A^T_{\mathcal{J}} y^0 + s^0_{\mathcal{J}}) \\
 & & & & & -(-c_{\mathcal{J}} \tau' + A^T_{\mathcal{J}} y' + s'_{\mathcal{J}}), \\
b^T d_y & -(c_{\mathcal{J}})^T d_{x_{\mathcal{J}}} & & -d_\kappa & = & \phi^k (b^T y^0 - c^T_{\mathcal{J}} x^0_{\mathcal{J}} + \kappa^0) \\
 & & & & & -(b^T y' - c^T_{\mathcal{J}} x'_{\mathcal{J}} + \kappa'), \\
 & S'_{\mathcal{J}} d_{x_{\mathcal{J}}} & & +X'_{\mathcal{J}} d_{s_{\mathcal{J}}} & = & -X'_{\mathcal{J}} s'_{\mathcal{J}} + \phi' \frac{g^0}{n+1} e, \\
 & & \kappa' d_\tau & +\tau' d_\kappa & = & -\tau' \kappa' + \phi' \frac{g^0}{n+1} e,
\end{array} \tag{6.20}$$

and perform the update (6.18). In this case it can be verified that

$$\begin{array}{cccccl}
 & A_{\mathcal{J}} x^+_{\mathcal{J}} & -b\tau^+ & & = & (1-\alpha)(A x'_{\mathcal{J}} - b\tau') \\
 & & & & & +\alpha\phi' (A_{\mathcal{J}} x^0_{\mathcal{J}} - b\tau^0), \\
-A^T_{\mathcal{J}} y^+ & & +c_{\mathcal{J}} \tau^+ & -s^+_{\mathcal{J}} & = & (1-\alpha)(-c_{\mathcal{J}} \tau' + A^T_{\mathcal{J}} y' + s'_{\mathcal{J}}) \\
 & & & & & +\alpha\phi' (-c_{\mathcal{J}} \tau^0 + A^T_{\mathcal{J}} y^0 + s^0_{\mathcal{J}}), \\
b^T y^+ & -(c_{\mathcal{J}})^T x^+_{\mathcal{J}} & & -\kappa^+ & = & (1-\alpha)(b^T y' - c^T_{\mathcal{J}} x'_{\mathcal{J}} + \kappa') \\
 & & & & & +\alpha\phi' (b^T y^0 - c^T_{\mathcal{J}} x^0_{\mathcal{J}} + \kappa^0),
\end{array}$$

Note if $\alpha = 1$, then the updated solution satisfies (6.14) exactly. Moreover, for any positive step-size α, then the solution is moved closer to satisfying (6.14). Unfortunately, there is no control over the complementary gap and it may decrease or increase. Therefore, α is chosen such that the complementarity gap is only allowed to be within the range $[0.5, 2.0]$ times the target gap (6.15). If the step-size has to be cutback, because the complementarity gap was moving outside the range, then the search direction (6.17) is used repeatedly with $\eta = 0$ and $\gamma > 1$ ($\gamma < 1$) to increase (decrease) the complementarity gap until the target value of the complementarity gap is reach. Ultimately this procedure generates a solution satisfying (6.14). However, when this procedure is terminated then the complementarity gap might be too small or too large and the solution might not be centered. However, this can be corrected using the search direction (6.17) with $\eta = 0$ and γ chosen appropriately.

7 COMPUTATIONAL RESULTS

In this section we report computational results for the implementation discussed in the previous section. The test problems we are using are LP relaxations of real world set-partitioning and set-covering problems obtained from the OR-Library, see `http://mscmga.ms.ic.ac.uk/info.html`. The characteristic of

these problems is that $n \gg m$, all elements in the matrix A are either zero or one, and all coefficients in the right-hand side are one. All constraints for the set-covering type problems are of the greater-than-equal type, whereas all the constraints for the set-partitioning type problems are of the equality type. Therefore, we add a full set of artificial variables, i.e. slack variables with lower and upper bounds zero, to the set-partitioning problems. Hence, the initial set $\mathcal{J}^0$ consists of slack variables which corresponds to the identity matrix. In our implementation we use $\beta = 0.1$ and $\theta = 0.1$, meaning that ϕ is reduced fairly fast and the central path is followed loosely. In each iteration we add multiple variables as follows. Assume that the variables in the set $\mathcal{J}^-$ has been sorted in increasing order after index, then each kth variable is chosen, where k is determined such that at most $m/4$ variables are chosen. The procedure for computing the search direction is a slight modification of those presented in Andersen and Andersen (1997). Finally, the algorithm is terminated when a primal and dual solution has been computed to the original problem that does not violate any inequality by more than 10^{-8} and 8 figures in the corresponding primal and dual objective value are identical.

The computational test is performed on a 300mhz Pentium II PC using Windows NT and having 128 megabytes of memory.

Name	Con-	Variables		Iterations		Time	Primal	Relative
	straints	full	relaxation	outer	inner		objective	gap
sppaa01	823	8904	2774	42	121	77.17	5.553544e+004	1.48e-010
sppaa02	531	5198	1592	29	71	15.27	3.049400e+004	6.42e-011
sppaa03	825	8627	2485	37	83	46.69	4.961636e+004	1.13e-010
sppaa04	426	7195	1486	36	93	12.66	2.587761e+004	1.45e-010
sppaa05	801	8308	2443	38	97	54.69	5.373593e+004	9.94e-011
sppaa06	646	7292	2087	33	86	29.51	2.697719e+004	1.74e-011
sppkl01	55	7479	350	21	48	0.48	1.084000e+003	2.19e-010
sppkl02	71	36699	506	26	61	1.15	2.152500e+002	3.69e-010
sppnw01	135	51975	586	38	76	2.55	1.148520e+005	3.50e-010
sppnw02	145	87879	742	39	80	3.91	1.054440e+005	9.24e-010
sppnw03	59	43749	392	28	58	1.31	2.444700e+004	4.55e-010
sppnw04	36	87482	397	28	67	1.94	1.631067e+004	-9.62e-012
sppnw05	71	288507	522	35	77	6.76	1.328780e+005	3.04e-012
sppnw06	50	6774	274	22	37	0.42	7.640000e+003	8.78e-010
sppnw07	36	5172	221	20	34	0.28	5.476000e+003	-2.49e-010
sppnw08	24	434	109	20	35	0.11	3.589400e+004	-2.21e-010
sppnw09	40	3103	226	23	45	0.25	6.776000e+004	1.62e-010
sppnw10	24	853	133	20	37	0.13	6.827100e+004	1.37e-013
sppnw11	39	8820	259	22	45	0.35	1.162545e+005	2.39e-011
sppnw12	27	626	134	20	34	0.12	1.411800e+004	1.07e-010
sppnw13	51	16043	354	27	57	0.61	5.013200e+004	9.07e-011
sppnw14	73	123409	514	35	78	3.35	6.184400e+004	5.79e-011
sppnw15	31	467	150	21	36	0.18	6.774300e+004	1.72e-011
sppnw16	139	148633	625	29	55	4.41	1.181590e+006	1.54e-011
sppnw17	61	118607	414	30	71	2.96	1.087575e+004	2.85e-010
sppnw18	124	10757	584	32	70	1.37	3.388643e+005	2.00e-011
sppnw19	40	2879	207	20	35	0.23	1.089800e+004	-5.31e-010
sppnw20	22	685	113	19	34	0.10	1.662600e+004	-2.09e-010
sppnw21	25	577	108	17	26	0.09	7.380000e+003	2.74e-011
sppnw22	23	619	123	18	32	0.10	6.942000e+003	-6.78e-011
sppnw23	19	711	127	22	38	0.11	1.231700e+004	1.41e-010
sppnw24	19	1366	140	18	27	0.09	5.843000e+003	-4.27e-010
sppnw25	20	1217	141	19	32	0.11	5.852000e+003	5.62e-011
sppnw26	23	771	165	20	33	0.11	6.743000e+003	1.54e-011
sppnw27	22	1355	165	20	36	0.13	9.877500e+003	3.72e-010

Table 1.1: Computation using the homogenized cutting plane algorithm.

Name	Con-	Variables		Iterations		Time	Primal	Relative
	straints	full	relaxation	outer	inner		objective	gap
sppnw28	18	1210	111	20	33	0.11	8.169000e+003	1.97e-010
sppnw29	18	2540	194	22	43	0.15	4.185333e+003	3.02e-010
sppnw30	26	2653	168	20	34	0.16	3.726800e+003	-4.14e-010
sppnw31	26	2662	173	19	32	0.15	7.980000e+003	2.22e-010
sppnw32	19	294	107	20	38	0.08	1.457000e+004	1.60e-010
sppnw33	23	3068	162	19	30	0.15	6.484000e+003	3.50e-011
sppnw34	20	899	125	18	24	0.08	1.045350e+004	5.43e-011
sppnw35	23	1709	139	18	28	0.11	7.206000e+003	3.98e-010
sppnw36	20	1783	158	21	38	0.13	7.260000e+003	-7.80e-011
sppnw37	19	770	105	18	28	0.09	9.961500e+003	-8.63e-011
sppnw38	23	1220	162	19	42	0.16	5.552000e+003	-6.17e-011
sppnw39	25	677	143	18	28	0.11	9.868500e+003	7.15e-010
sppnw40	19	404	90	19	30	0.07	1.065825e+004	-3.27e-011
sppnw41	17	197	80	18	28	0.06	1.097250e+004	-3.87e-010
sppnw42	23	1079	141	18	31	0.11	7.485000e+003	-4.26e-011
sppnw43	18	1072	151	20	36	0.10	8.897000e+003	-7.00e-011
sppus01	145	1053137	768	39	90	40.33	9.963067e+003	1.30e-010
sppus03	77	85552	405	26	52	2.95	5.338000e+003	5.16e-011
sppus04	163	28016	609	29	54	2.54	1.773167e+004	4.98e-011
rail2536	2536	1081841	11100	42	183	2346.76	6.883992e+002	3.09e-010
rail2586	2586	920683	12663	42	223	1217.80	9.359218e+002	1.54e-010
rail4284	4284	1092610	18206	43	201	14529.23	1.054054e+003	2.48e-010
rail507	507	63009	2489	35	144	26.08	1.721456e+002	4.23e-010
rail516	516	47311	1963	29	100	14.10	1.820000e+002	1.86e-010
rail582	582	55515	2525	35	146	27.93	2.097122e+002	1.54e-010
Sum			74585	1551	3661			

Table 1.1: Computation using the homogenized cutting plane algorithm.

In Table 1.1 the test problems and the computational results are presented. All problems having a name starting with "spp" are of the set-partitioning type and the remaining problems are of the set-covering type.

Beyond the name of the test problems the table first shows the number of constraints (m). Next the number of variables in the full problem (n) and the final relaxation is shown.

The column with the heading Iterations shows the number of times ϕ is reduced and the number of times the search direction is computed. It can be observed that for most of the problems few outer iterations are required. Furthermore, approximately 2 to 3 inner iterations are needed per outer iteration. Although for the large rail problems 4 to 5 inner iterations are needed per outer iteration. This indicates that our warm start technique works well. It should also be noted that relatively few variables out of the total number of variables in the full problem are included in the final relaxation. One exception is the set of problems with names starting with "sppa". However, these problems have relatively few variables compared to the number of constraints and in fact the final relaxation contains a number of variables which is only a small multiple of the number of constraints. In computational experiments not reported here it was observed that if the number of variables added per iterations was limited to about 50, then the number of outer iterations grew by around a factor of 2 for a lot of the problems. Moreover, contrary to expectations, the total number of generated variables was not reduced significantly.

Table 1.2 Results for solving some of the problems using MOSEK v1.0b.

Name	Iterations	Time		Primal
		full	cut	objective
sppnw01	15	23.2	2.55	1.148520e+005
sppnw02	15	40.2	3.91	1.054441e+005
sppnw04	20	28.1	1.94	1.631067e+004
sppnw05	29	170.5	6.76	1.328780e+005
sppnw14	23	60.2	3.35	6.184400e+004
sppnw16	23	106.1	4.41	1.181590e+006
sppnw17	24	58.7	2.96	1.087575e+004
sppus03	15	31.3	2.95	5.338000e+003
rail507	23	44.7	26.86	1.721456e+002
rail516	15	25.8	14.10	1.820000e+002
rail582	22	43.8	27.93	2.097122e+002

The column with the heading Time shows the number of CPU seconds spent to solve the problems. Except for the rail problems these numbers are small. The reason the solution time is large for the rail problems is that for those problems it is expensive to compute the search direction which is caused by the relatively large number of constraints and fill-in in the Cholesky factorization, see Andersen and Andersen (1997) for details about the latter. Finally, it should be noted that the solution time for rail4284 is exceptionally large due to memory swapping caused by needing more than 128 megabytes of memory to solve the problem.

The last two columns shows the optimal primal objective value and the relative gap. In all cases a solution satisfying the termination tolerances was computed.

One obvious question is how the homogenized cutting plane algorithm compares to solving the full problem directly. To answer that question we have made Table 1.2 which reports results for solving the problems having more than 50000 variables using MOSEK v1.0b interior-point optimizer (Andersen and Andersen (1997)). Note that most of the linear algebra routines are shared by MOSEK and the homogenized cutting plane algorithm so the comparison should be fair. Moreover, the largest problems are excluded from Table 1.2, because those problems could not be solved using only 128 megabytes of memory. Table 1.2 presents the number of interior-point iterations and the time spent

solving the problems for solving the full problem and using MOSEK (full) and the cutting plane approach (cut). Finally, the optimal primal objective value reported by MOSEK is shown. A comparison of the timing results shows that for the chosen problems the homogenized cutting-plane algorithm is significantly better than the full approach with respect to running time. Indeed for the problems sppnw05 and sppnw16 the cutting plane approach is approximately 20 to 25 times faster than the regular approach. Hence, we can conclude at least for LP relaxations of set-partitioning and set-covering problems having many more variables than constraints, then the homogenized cutting plane approach is an attractive alternative to solving the full problem. Moreover, it should be emphasized that the problems with about one million variables can only be solved using the cutting plane approach due to the large memory consumption of the full approach.

In summary, the suggested implementation is capable of solving large LP relaxations of set-covering and set-partitioning problems in a low number of outer and inner iterations. Moreover, for problems having many more variables than constraints, then the approach is much more efficient than solving the full problem directly. However, it seems to be important for efficiency reasons that possibly many variables in each iteration can be generated.

Finally, it should be mentioned that several possibilities exist for improving the efficiency of the implementation. For example a more sophisticated choice of the initial set $\mathcal{J}^0$ is likely to reduce the number of inner and outer iterations.

8 CONCLUSIONS

We have presented a cutting plane algorithm using a self-dual homogenized formulation. The analysis of the algorithm uses a different proximity measure than that in, for example, Goffin *et al.* (1996). The algorithm has the same complexity as that in Goffin *et al.* (1996).

Furthermore, we present an implementation of the homogenized cutting plane algorithm adapted to solve LP problems having many more variables than constraints. The implementation is applied to the LP relaxation of real-world set-covering and set-partitioning problems. The computational results show that the discussed implementation solves the test problems in a low number of inner and outer iterations. Moreover, the results show that for problems having many more variables than constraints, the cutting plane approach saves a lot of computation time and space compared to solving the problems directly.

References

E. D. Andersen and K. D. Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In J. B. G. Frenk, C. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization Techniques, Proceedings of the HPOPT-II conference*, 1997, forthcoming.

D. S. Atkinson and P. M. Vaidya. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69:1–43, 1995.

R. E. Bixby, J. W. Gregory, I. J. Lustig, R. E. Marsten, and D. F. Shanno. Very large-scale linear programming: A case study in combining interior point and simplex methods. *Oper. Res.*, 40(5):885–897, 1992.

J.-L. Goffin, Z.-Q. Luo, and Y. Ye. On the complexity of a column generation algorithm for convex or quasiconvex problems. In *Large Scale Optimization: The State of the Art*. Kluwer Academic Publishers, 1993.

J.-L. Goffin, Z.-Q. Luo, and Y. Ye. Complexity analysis of an interior cutting plane method for convex feasibility problems. *SIAM Journal on Optimization*, 6:638–652, 1996.

J.-L. Goffin and J.-P. Vial. Multiple cuts in the analytic center cutting plane method. Technical Report Logilab Technical Report 98.10, Logilab, Management Studies, University of Geneva, Geneva, Switzerland, June 1998.

D. den Hertog, J. Kaliski, C. Roos, and T. Terlaky. A logarithmic barrier cutting plane method for convex programming problems. *Annals of Operations Research*, 58:69–98, 1995.

J. E. Mitchell and S. Ramaswamy. A long-step, cutting plane algorithm for linear and convex programming. Technical Report 37–93–387, DSES, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, August 1993. Substantially revised: April 28, 1998.

J. E. Mitchell and M. J. Todd. Solving combinatorial optimization problems using Karmarkar's algorithm. *Mathematical Programming*, 56:245–284, 1992.

Y. E. Nesterov and J. Ph. Vial. Homogeneous analytic center cutting plane methods for convex problems and variational inequalities. Technical report, Department of Management Studies, University of Geneva, Switzerland, July 1997.

S. Ramaswamy and J. E. Mitchell. On updating the analytic center after the addition of multiple cuts. Technical Report 37–94–423, DSES, Rensselaer Polytechnic Institute, Troy, NY 12180, October 1994. Substantially revised: August, 1998.

C. Roos, T. Terlaky, and J.-Ph. Vial. *Theory and Algorithms for Linear Optimization: An Interior Point Approach*. John Wiley, Chichester, 1997.

X. Xu, P. -F. Hung, and Y. Ye. A simplified homogeneous and self-dual linear programming algorithm and its implementation. *Annals of Operations Research*, 62:151–171, 1996.

Y. Ye. Complexity analysis of the analytic center cutting plane method that uses multiple cuts. *Mathematical Programming*, 78:85–104, 1997.