A Long-Step, Cutting Plane Algorithm for Linear and Convex Programming[1]

John E. Mitchell[2]
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
and
Srinivasan Ramaswamy
Information Services Division
Research and Development
United Airlines
1200 E. Algonquin Road
Elk Grove Village, IL 60007

DSES Technical Report No. 37-93-387

August 18, 1993.
Last revised: April 24, 2000.

Abstract

A cutting plane method for linear programming is described. This method is an extension of Atkinson and Vaidya's algorithm, and uses the central trajectory. The logarithmic barrier function is used explicitly, motivated partly by the successful implementation of such algorithms. This makes it possible to maintain primal and dual iterates, thus allowing termination at will, instead of having to solve to completion. This algorithm has the same complexity ($O(nL^2)$ iterations) as Atkinson and Vaidya's algorithm, but improves upon it in that it is a 'long-step' version, while theirs is a 'short-step' one in some sense. For this reason, this algorithm is computationally much more promising as well. This algorithm can be of use in solving combinatorial optimization problems with large numbers of constraints, such as the Traveling Salesman Problem.

Key words: Cutting plane, path following, analytic center, linear programming, convex programming.

[1]AMS 1991 subject classification. Primary 90C05. Secondary 90C25, 65K05.

[2]Research partially supported by ONR Grant numbers N00014–90–J–1714 and N00014-94-1-0391 and also by a grant from the Dutch NWO and Delft University of Technology for the 1997–98 academic year, while visiting ITS/TWI/SSOR at Delft University of Technology.

1 Introduction

The problem of interest is to solve

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \geq b \qquad\qquad (P) \end{array}$$

where x and c are n-vectors, b is an m-vector, and A is an $m \times n$ matrix. We are interested in instances where m is far larger than n, and our algorithm can handle problems with infinitely many constraints, provided we have an oracle for finding constraints as necessary. Its dual is:

$$\begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y = c \qquad\qquad (D) \\ & y \geq 0 \end{array}$$

where b and y are m-vectors.

Since $m \gg n$, it would be preferable to use a cutting plane approach to solve (P), wherein only a promising set of constraints or hyperplanes are kept, with hyperplanes being added/deleted at each iteration. Such an approach has been provided by Atkinson and Vaidya [1]. These authors discuss a cutting plane algorithm for finding a feasible point in a convex set. At each iteration, they maintain a polytope that is guaranteed to contain the convex set, and use the analytic center of this polytope as the test point. If this point is not contained in the convex set, an oracle returns a hyperplane that separates the test point and the convex set. This hyperplane is used to update the polytope. Further, hyperplanes currently in the polytope that are deemed 'unimportant' according to some criteria are dropped.

Clearly, such an algorithm can be very useful in solving problems like (P). Much like the ellipsoid algorithm (see [9]), Atkinson and Vaidya's algorithm can be applied to linear programming problems by cutting on a violated constraint when the test point is infeasible, and cutting on the objective function when the test point is feasible but not optimal. (We explicitly include such an algorithm in the Appendix). This would then somewhat resemble Renegar's algorithm (see [24]). However, this is somewhat like a 'short step' method — when cutting on the objective function, the cut is placed such that the next iterate is in a small ellipsoid around the current point. This suggests that the progress in objective function value cannot be too much.

Our interest is in a long step cutting plane algorithm that uses the barrier function explicitly, partly because of the successful implementation of such algorithms for linear programming. Such algorithms are not only polynomial in complexity, but also exhibit superlinear [34] or quadratic [33] convergence asymptotically. Also, in

extending Atkinson and Vaidya's work through the use of such barrier functions, some of their proofs become simplified and we show the links between their work and barrier function methods. Another point is that by making use of the standard logarithmic barrier function framework, we avoid having to increase the number of constraints whenever we need to drive the objective function value down — instead of cutting on the objective function, we just reduce the value of a scalar parameter. Finally, since in such an algorithm we would maintain primal and dual variables, it allows for early termination when the sub-optimality is deemed to be within allowable limits. Thus, what we visualize is essentially a long step path-following algorithm that uses only some of the constraints, and whenever infeasibility is encountered an Atkinson-Vaidya-like scheme is used until the current iterate again becomes feasible.

Recently, there has been some work on cutting plane methods, especially in the context of the convex feasibility problem. Most of these recent methods are "central cutting" plane methods, where the test point is in the center of the polytope that approximates the convex set at the current iteration. The most commonly used test point is the analytic center of the polytope (see eg., [1, 31, 5, 15, 32]), since this is fairly easy to compute. Other test points that have been used include the center of the largest inscribed sphere [3] or ellipsoid [28] and the volumetric center [30]. The analytic center is preferred because of computational reasons, and also because of its nice properties (see eg., [27]).

Goffin *et al.* [5] described a fully polynomial column generation algorithm, which does not drop any columns. This algorithm will find an optimal solution within ϵ in time polynomial in ϵ, but not in time polynomial in $\ln(\epsilon)$. This method has been extended to add many cuts at once, and to show that cuts do not need to be weakened but can be added right through the analytic center [15, 32, 6]. It should be noted that none of these papers deal explicitly with optimization problems; instead, they present results for the convex feasibility problem.

In particular, the recent work of Goffin and Vial [6] shows that the analytic center can be recovered in $O(p\log(p))$ Newton steps if p constraints are added through the current iterate. Our analysis could be modified to consider adding multiple cuts through the current iterate. The affected lemmas are Lemma 5 and Lemma 11; we note the effect of multiple cuts immediately after these lemmas. If no more than p cuts are added at any one time, then it should be possible to extend the results of [6] to show that the total number of Newton steps in our algorithm is multiplied by a factor of at most $O(p\log(p))$.

In this paper, we present a long step algorithm (i.e., one where the factor by which the barrier parameter is reduced is a constant and does not depend on the

problem size) that traces the central trajectory of the current polytope. When an iterate is infeasible in the 'overall problem', constraints are added until feasibility is regained. Additionally, constraints are dropped whenever possible according to some criterion. We prove a complexity of $O(nL^2)$ iterations for this algorithm. Going by computational experience discussed in the literature, we would certainly expect the long-step version to perform better in practice. This method has been extended to the case where many cuts are added at once right through the analytic center [22], and a similar method has been proposed using the volumetric center [23]. Luo *et al.* [16] have extended the fully polynomial algorithm to a long-step method with exponential complexity, although it does not require that constraints be dropped.

It is important to note the work of D. den Hertog *et al.* ([12, 13, 11]) and Kaliski et al. ([14]). These papers discuss an algorithmic framework very similar to ours – the "combination" of long steps in a barrier parameter with the addition and deletion of cuts to yield a long step cutting plane algorithm. Indeed, the authors in [11] describe an algorithm like this, and this algorithm is applied to semi-infinite programming in [14]. However, these papers only prove complexity that is polynomial in the total number of cuts that become active, and is not necessarily polynomial in the dimension of the space. Indeed, in their conclusions, the authors of [11] mention that proving possible polynomial complexity of their algorithm remains open. In [14], the authors point out that results from [5] suggest that their algorithm is fully polynomial. Thus, while our algorithm in this paper has a similar flavor to those just described, we accomplish one other objective — we prove that our algorithm has complexity polynomial in the dimension of the space, while also having the computationally appealing characteristics of a long step logarithmic barrier algorithm.

Good computational results have been obtained with interior point cutting plane methods. Recently, they have been used to solve stochastic programming problems [2], multicommodity network flow problems [4], and integer programming problems [18], as well as other forms of convex optimization problems [8, 7]. For some classes of linear ordering problems, a cutting plane scheme that combines an interior point method and a simplex method has been shown to be up to ten times faster than one that just uses either of the methods individually [20]. For some max cut problems, an interior point cutting plane method has been shown to considerably outperform a simplex cutting plane method [19].

In the following section, we introduce some preliminaries and notation. In section 3, the complete algorithm is described. Local convergence of the algorithm is discussed in section 4, where we show that a new approximate analytic center can be recovered in one Newton step when a constraint is added or dropped as speci-

fied in the algorithm. (We note that in the case where a cut is neither added nor dropped, the new approximate center cannot be recovered in one Newton step, but we develop a bound for the number of Newton steps.) The overall number of Newton steps required by the algorithm is bounded in section 5, where the maximum number of constraints generated by the algorithm is also bounded. We offer our conclusions in the final section.

2 Preliminaries

In order to simplify the exposition later, we will use $\hat{m}$ from now on to denote the number of constraints in the original, full problem. Similarly, we use $\hat{A}$ and $\hat{b}$ to denote the full constraint matrix and right hand side. The dimension of the vector x will remain n throughout the algorithm, and the objective vector c will remain unchanged. The original problem of interest can be written:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & \hat{A}x & \geq \hat{b}, \qquad (\hat{P}) \end{array}$$

where $\hat{A}$ is an $\hat{m} \times n$ matrix. Its dual is:

$$\begin{array}{lll} \max & \hat{b}^T y & \\ \text{s.t.} & \hat{A}^T y = c & \qquad (\hat{D}) \\ & y \geq 0. & \end{array}$$

We point out here that we do not actually require $\hat{m}$ to be finite — we will never work with the formulations $(\hat{P})$ or $(\hat{D})$ directly. We will just approximate them by a finite number of linear constraints. We further assume, without loss of generality, that the two-norms of c and the rows of $\hat{A}$ are each equal to 1. We also assume that there exists a problem dependent constant L such that

1. The set of optimal solutions to $(\hat{P})$ is guaranteed to be contained in the n-dimensional hypercube of half-width 2^L given by $\{x \in \Re^{\hat{n}} : |x_i| \leq 2^L\}$.

2. The set of feasible solutions to $(\hat{P})$ contains a full dimensional ball of radius 2^{-L}.

3. *Optimality criterion:* It suffices to find a solution to $(\hat{P})$ to within an accuracy 2^{-L}.

At any given iteration, we operate with a relaxation of $(\hat{P})$. The algorithm is initialized with the relaxation

$$\begin{array}{lrcll} \min & c^T x & & & \\ \text{s.t.} & Ix & \geq & (-2^L)e_n & (P_0) \\ & -Ix & \geq & (-2^L)e_n & \\ & c^T x & \geq & -2^L\sqrt{n} & \end{array}$$

where, by our assumptions, the feasible set of this relaxation contains $\mathcal{X}$, the set of optimal solutions to $(\hat{P})$. The vector of dimension n whose components are all equal to one is denoted by e_n. The first $2n$ hyperplanes included in this initial relaxation will be referred to as the box hyperplanes, and serve to ensure that the polytope we have at any iteration is always bounded, and thus has a unique μ-center for any $\mu > 0$. (We use the term μ-center without defining it for now — it will be defined shortly.) The final constraint does not affect the feasible region of the initial relaxation. As the algorithm obtains better lower bounds on the optimal value of the full problem, the right hand side of this lower bound constraint is updated.

At iteration k, we would have something like :

$$\begin{array}{lrcll} \min & c^T x & & & \\ \text{s.t.} & Ix & \geq & (-2^L)e_n & (P_k) \\ & -Ix & \geq & (-2^L)e_n & \\ & c^T x & \geq & l_k & \\ & \bar{A}_k x & \geq & \bar{b}_k & \end{array}$$

where $\bar{A}_k \in \Re^{\bar{m}_k \times n}$, and l_k is some lower bound on the optimal objective function value. This can be written more simply as

$$\begin{array}{lrcll} \min & c^T x & & & \\ \text{s.t.} & A_k x & \geq & b_k & (P_k) \end{array}$$

with $A_k \in \Re^{m_k \times n}$, so $m_k = 2n + 1 + \bar{m}_k$. We refer to the feasible region of (P_k) as Q_k. Throughout most of the discussion, we may omit the subscripts k since that causes no ambiguity.

We also need to define the so-called logarithmic barrier function :

$$f(x, \mu) := \frac{c^T x}{\mu} - \sum_{i=1}^{m} \ln s_i \tag{1}$$

where s_i is equal to $a_i^T x - b_i$, and a_i^T is the i^{th} row of A. We use s_i without explicitly noting its functional dependence on x.

It is easily seen that the gradient of the barrier function is given by

$$\nabla f(x,\mu) = \frac{c}{\mu} - \sum_{i=1}^{m} \frac{a_i}{s_i} = \frac{c}{\mu} - A^T S^{-1} e \tag{2}$$

and the Hessian by

$$\nabla^2 f(x,\mu) = \sum_{i=1}^{m} \frac{a_i a_i^T}{s_i^2} = A^T S^{-2} A. \tag{3}$$

For a given value of μ , $x(\mu)$ denotes the unique minimizer of this barrier function. We refer to this unique point when we use the term *(exact) μ-center*.

A related function is defined as:

$$F(x) \ := \quad - \ \sum_{i=1}^{m} \ln \ s_i. \tag{4}$$

We also define quantities that are analogous to the so-called *variational quantities* used by Atkinson and Vaidya [1], and also studied in den Hertog *et al.* [12, 13]:

$$\sigma_j \ := \quad \frac{a_j^T (\nabla^2 f(x,\mu))^{-1} a_j}{s_j^2} \tag{5}$$

for j from 1 to m.

In our analysis, we refer to the polytope Q^*. Before defining Q^*, it is necessary to define some more notation. The algorithm moves from one approximate μ-center to another. If the approximate μ-center is feasible in $(\hat{P})$ then we may decide to reduce the barrier parameter. In this situation, we denote the current feasible approximate μ-center as x_{prev}, we let μ_{prev} denote the barrier parameter before it is reduced, and we let m_{prev} denote the number of constraints before the barrier parameter is reduced. Note that x_{prev} need not be the current iterate — it is actually the last iterate at which no cuts were returned by the oracle. The polytope Q^* is to be understood as follows — if we have the current point $\tilde{x}$, then

$$Q^* := Q \cap \{x \in \Re^n : c^T x \leq \zeta\} \tag{6}$$

where ζ is equal to $\max(c^T \tilde{x}, c^T x_{prev}) + 1.25 m_{prev} \mu_{prev}$. Thus, this polytope is certain to contain the optimal set to the problem of interest and the current point. This property is important, and enables us to decide termination criteria based on the volume and width of this polytope. This property would hold without the $1.25 m_{prev} \mu_{prev}$ term in the definition of ζ; this extra term is used in Lemma 10, where we relate the width of Q^* to the width of Q. It is important to note that the polytope Q^* is used only in the theoretical analysis — we do not explicitly add upper-bound constraints that restrict subsequent iterates to be in the polytope Q^*, although we could.

Note that the algorithm can be terminated before Q^* becomes too small, in the same way that the ellipsoid method can be terminated. In particular, if Q^* can be contained in an ellipsoid of volume no more than that of a ball of radius 2^{-L}, then we could round to an optimal solution using simultaneous diophantine approximation, as described in Grötschel *et al.* [9]. We use this observation in the proof of Theorem 1. An alternative method for rounding to an optimal solution is described in Megiddo [17].

2.1 Notation

We list some of our notation.

- m: the number of constraints in the current relaxation.
- $\bar{m}$: the number of constraints in the current relaxation, excluding the box constraints and the lower bound constraint.
- $\hat{m}$: the number of constraints in the complete formulation.
- m_{prev}: the number of constraints in the relaxation immediately before the barrier parameter μ was last reduced.
- μ: the barrier parameter.
- μ_{prev}: the value of the barrier parameter immediately before it was last reduced.
- x: the primal iterate.
- x_{prev}: the iterate immediately before the barrier parameter was last reduced. Note that this iterate must be feasible in the full problem $(\hat{P})$, and it is the best known feasible point.
- $f(x, \mu)$: barrier function, see equation (1).
- $F(x)$: another barrier function, see equation (4).
- σ_j: a variational quantity, see equation (5).
- Q^*: a polytope defined in equation (6).
- ζ: a parameter used in the definition of Q^*.
- ν: a quantity related to the number of cutting planes. We show $\bar{m} \leq \nu n L$ in Theorem 1, where it is shown that it suffices to take $\nu = 4093$.

- π: another quantity related to the number of cutting planes. We have $\pi = \nu + 3$ and show $m \leq \pi n L$ in Theorem 1.
- $\Phi(x)$: a measure of centrality defined in Lemma 1.
- $\delta(x, \mu)$: the Roos-Vial measure of centrality, also defined in Lemma 1.
- $E(M, z, r)$: an ellipsoid defined for any positive definite matrix M in equation (9).
- width(h): the maximum variation of $h^T x$ over Q^*.
- $\Delta(x, \mu, l)$: a potential function defined in equation (10).
- Λ: the growth in Δ at certain iterations. See Lemma 11.
- $\Theta(x, \mu, l)$: a potential function defined in equation (12).

3 The Algorithm

For the purpose of initializing the algorithm, we assume that the origin is feasible in problem $(\hat{P})$. This is reasonable since if we know any feasible point in $(\hat{P})$, we can modify the problem to make the origin feasible. The algorithm may well produce infeasible iterates, but we always keep a record x_{prev} of the best feasible point found so far.

The algorithm is shown in figure 1. It should be mentioned that where a lower case letter denotes a vector, the corresponding upper case letter denotes a diagonal matrix with this vector along the diagonal.

The basic idea of the algorithm is this: at any iteration, if a slack variable has become large (according to some criterion), and the associated variational quantity is small (as defined in (5)), that hyperplane is then dropped. Indeed, these quantities measure the effect of dropping the cut in some sense. If the slack variables stay small, and the current point is infeasible in $(\hat{P})$, then there is a violated constraint, and the polytope is updated with this constraint. (We assume that there is an 'oracle' that can find this violated constraint.) If the current point is feasible in $(\hat{P})$, then the lower bound is updated and the barrier parameter is reduced. A sufficient condition for termination is that we enter Subcase 2.2 of the algorithm with $1.25m\mu < 2^{-L}$, as we argue in Corollary 1.

Atkinson and Vaidya also include two criteria for termination in their algorithm that would imply that the feasible set is empty. They show that if the number

Figure 1: **The Algorithm**

Step 0 : Initialization

Set $\mu = 2^L$; $k = 0$; $A = \begin{bmatrix} I_n \\ -I_n \\ c^T \end{bmatrix}$; $b = \begin{bmatrix} -2^L e_n \\ -2^L e_n \\ -2^L\sqrt{n} \end{bmatrix}$; $x = [0 \cdots 0]^T$; $s = Ax - b$; $\kappa(i) = 2^L$, $i = 1 \ldots 2n$; $\kappa(2n+1) = 2^L\sqrt{n}$; $x_{prev} = x$; $m_{prev} = m$; $\mu_{prev} = \mu$; $y = \mu S^{-1}e - \mu S^{-2}A(A^T S^{-2} A)^{-1}[A^T S^{-1}e - \frac{c}{\mu}]$. Iterate, if necessary, to get approximate μ-center. Set constants $m_{max} := 4093nL$, and $s_{min} := 2^{-(3L+3)}/(24576n^{1.5}L)$.

Step k : The Iterative step

If $m \geq m_{max}$ or $\min_i\{a_i^T x - b_i\} < s_{min}$ then STOP: the best feasible point found so far is optimal.

We have a point $x = x^k$, which is an approximate μ-center. H denotes the Hessian $\nabla^2 f(x, \mu)$. Calculate $\gamma_i(x) = \frac{a_i^T x - b_i}{\kappa(i)}$ for i=1 to m, except for the lowerbound constraints that get added in Case 2.2 . Define $\gamma_i(x) := 1$ if i indexes the lower bound constraint. For a hyperplane indexed by j, if $\gamma_j(x) > 2$, calculate $\sigma_j(x)$ as in equation (5). Define $\Gamma := \max_i(\gamma_i(x))$.

Case 1 : $\Gamma > 2$

If for some j we have $\sigma_j(x) < 0.04$

then Subcase 1.1

Drop the hyperplane a_j, resulting in polytope Q_d. Take one Newton step to find approximate μ-center x_d. Get dual solution y.

else Subcase 1.2

Let j be an index such that $\gamma_j(x) > 2$. Reset $\kappa(j) = a_j^T x - b_j$.

Case 2 : $\Gamma \leq 2$

Subcase 2.1 : x is not feasible in $(\hat{P})$

Then $\exists$ j such that $\hat{a}_j^T x < \hat{b}_j$. Choose β such that $\frac{\hat{a}_j^T H^{-1} \hat{a}_j}{(\hat{a}_j^T x - \beta)^2} = \frac{1}{16}$ and $\beta < \hat{a}_j^T x$. Add $\hat{a}_j^T x \geq \beta$ to the polytope, getting new polytope Q_a. Set $m = m + 1$. Set $\kappa(m) = \hat{a}_j^T x - \beta$. Find new approximate μ-center x_a in one Newton step.

Subcase 2.2 : x is feasible in $(\hat{P})$

Test x for optimality (is $1.25m\mu < 2^{-L}$?). If optimal, get dual solution y and STOP. Else, set $l = c^T x - \frac{5}{4}m\mu$. Let l_{prev} denote previous lower bound. If $l_{prev} \geq l$, the lower bound cannot be improved upon. If $l_{prev} \leq l$, remove existing constraint $c^T x \geq l_{prev}$, add $c^T x \geq l$. Set $\mu = \mu\rho$ where $\rho \in (0.5, 1)$ is a constant independent of the problem. Take Newton steps with linesearch to find new approximate μ-center.

of hyperplanes at any iteration exceeds a certain level depending on the problem instance, the volume of the current polytope would be too small to contain a small enough ball (and therefore the convex set). Also, if the smallest slack is smaller than a certain number (that depends on the problem size), then the polytope would be too narrow to contain a small enough ball. By construction, our optimal set is non-empty — we assume the existence of a feasible point, and the polytope is bounded. We can be sure that Q^* is small enough if these criteria are satisfied. We make use of these criteria in showing the polynomial complexity of our algorithm, and therefore summarize them here:

1. There exists a constant ν, independent of the problem, such that throughout the algorithm we must have $\bar{m} < \nu nL$. (See Theorem 1.)

2. Given 1, if $\min_i(s_i) \leq \frac{2^{-(3L+3)}}{6(\nu+3)n^{1.5}L}$, then the polytope Q would have become too narrow. (See Lemma 10 and Theorem 2.)

As in [1], we show polynomial complexity by showing that condition 2 is inevitable after a polynomial number of iterations.

The term 'approximate center' has been used thus far without definition. This will be discussed in the next section. We also observe that from the dual solution y, a solution $\hat{y}$ to $(\hat{P})$ can be constructed easily as follows: first let $I(j) = \{k : k^{th}$ row of A is a copy of the j^{th} row of $\hat{A}\}$. (Note that polytope is defined at the current iteration by $Ax \geq b$.) Then,

$$\hat{y}_j = \sum_{i \in I(j)} y_i \quad \text{for } j = 1 \text{ to } \hat{m}.$$

In much of the analysis of this algorithm, we have only assumed that the current iterate is an approximate μ-center. However, for some of the proofs, we assume that we have an exact μ-center. This helps in focusing attention on the key issue without having to keep track of details that would arise from having only an approximate μ-center.

The algorithm can be used without first finding an initial feasible point. In this case, we would be unable to initialize x_{prev}. If the problem is feasible, then the algorithm would eventually find a feasible point, allowing x_{prev} to be initialized. If the problem is infeasible then the algorithm would eventually terminate because the number of constraints became too large or one of the slacks became too small. In order to simplify the exposition slightly, we don't consider this more general case.

4 Local Convergence Analysis

We first want to establish that each step in the algorithm doesn't take too much work. We do this by first defining a measure of the proximity to a μ-center, and then bounding this measure at each iteration.

The first lemma provides us with this measure of how close a point is to the μ-center corresponding to the current μ.

Lemma 1 *The quantity*

$$\Phi(x) = \nabla f(x,\mu)^T [\nabla^2 f(x,\mu)]^{-1} \nabla f(x,\mu),$$

analogous to

$$\Psi(x) = \nabla F(x)^T [\nabla^2 F(x)]^{-1} \nabla F(x)$$

used by Atkinson and Vaidya, is the square of the Roos-Vial measure of centrality (see [25]) $\delta(x,\mu)$, which is given by

$$\begin{aligned} \delta(x,\mu) &= \min_y \|\frac{Sy}{\mu} - e\| \\ & \quad s.t \;\; A^T y = c. \end{aligned}$$

Proof: Consider the quadratic program with optimal value $\delta(x,\mu)^2$:

$$\begin{aligned} \min \;\; & \left(\frac{Sy}{\mu} - e\right)^T \left(\frac{Sy}{\mu} - e\right) \\ & s.t. \; A^T y \;=\; c. \end{aligned}$$

The KKT conditions for this problem are

$$\begin{aligned} \frac{S^2 y}{\mu^2} \;-\; \frac{Se}{\mu} \;+\; A\lambda &= 0 \\ A^T y &= c. \end{aligned}$$

Solving for λ, we obtain

$$\lambda = \frac{1}{\mu^2} \left(A^T S^{-2} A\right)^{-1} \left[\mu A^T S^{-1} e \;-\; c\right].$$

It follows that

$$\begin{aligned} y &= \mu^2 S^{-2} \left[\frac{Se}{\mu} \;-\; \frac{1}{\mu^2} A \left(A^T S^{-2} A\right)^{-1} \left[\mu A^T S^{-1} e \;-\; c\right]\right] \\ &= \mu S^{-1} e \;-\; \mu S^{-2} A \left(A^T S^{-2} A\right)^{-1} \left[A^T S^{-1} e \;-\; \frac{c}{\mu}\right], \end{aligned}$$

so,

$$\frac{Sy}{\mu} \;=\; e \;-\; S^{-1}A\left(A^TS^{-2}A\right)^{-1}\left[A^TS^{-1}e \;-\; \frac{c}{\mu}\right]$$

and

$$\begin{aligned}\|\frac{Sy}{\mu} \;-\; e\|^2 \;&=\; \left[A^TS^{-1}e \;-\; \frac{c}{\mu}\right]^T\left(A^TS^{-2}A\right)^{-1}\left[A^TS^{-1}e \;-\; \frac{c}{\mu}\right]\\ &=\; \nabla f(x,\mu)^T[\nabla^2 f(x,\mu)]^{-1}\nabla f(x,\mu)\\ &=\; \Phi(x).\end{aligned}$$

□

This lemma essentially relates the "Newton Decrement" as studied by Nesterov and Nemirovskii [21] with the Roos-Vial measure of centrality. (Indeed, much more general results regarding the application of Newton's method to such functions may be found in [21].) Having defined the Roos-Vial measure, we now need to define a term used in the algorithm — a point x is referred to as an *approximate μ-center* if $\delta(x,\mu) < \frac{1}{4}$. We now state some properties of the Roos-Vial measure.

Lemma 2 *If we have a point x such that $\delta(x,\mu) < 1$, then*

$$y^\star := arg\{\min_y \|\frac{Sy}{\mu} - e\| \;:\; A^Ty = c\}$$

is feasible in $(\hat{D})$: ie., $y^\star \geq 0$.

Proof: This lemma is clear from the form of $y^\star$ given in the proof of Lemma 1 above, and from the fact that $(\frac{Sy}{\mu})_i$ is between 0 and 2. (See also [10, 25].) □

Lemma 3 *If we have a point x such that $\delta(x,\mu) < 1$, then taking Newton steps results in quadratic convergence to $x(\mu)$.*

Proof: See [25, 10]. □

The significance of these lemmas is clear — in order to show local convergence, we now merely need to bound $\delta(x,\mu)$ above at each step of the iteration. In Lemmas 4 and 5, we establish that the Roos-Vial measure does not change appreciably in Cases 1.1 and 2.1 of the algorithm. (Results very similar to the following two lemmas may be found in Lemmas 9 and 12 of [13].)

Lemma 4 *If we have a point x which is an approximate μ-center of the current polytope Q, and we drop a hyperplane a_0 in Subcase 1.1, then $\delta_d(x,\mu) < \frac{1}{2}$, where the subscript d denotes quantities after dropping the hyperplane a_0.*

Proof: Let there be $m+1$ hyperplanes $a_0 \cdots a_m$ defining the current polytope. We have a point x, with $\delta(x,\mu) < \frac{1}{4}$. Call the gradient $g := \nabla f(x,\mu) = \frac{c}{\mu} - \sum_{i=0}^{m} \frac{a_i}{s_i}$. Then

$$\nabla f_d(x,\mu) = g + \frac{a_0}{s_0} \quad \text{and} \quad H_d = H - \frac{a_0 a_0^T}{s_0^2}.$$

Therefore, by the Sherman-Morrison-Woodbury formula,

$$H_d^{-1} = H^{-1} + \frac{H^{-1} a_0 a_0^T H^{-1}}{s_0^2(1 - \frac{a_0^T H^{-1} a_0}{s_0^2})}.$$

Therefore,

$$\begin{aligned}
[\delta_d(x,\mu)]^2 &\leq (g + \frac{a_0}{s_0})^T H^{-1} (g + \frac{a_0}{s_0}) + \frac{[(g + \frac{a_0}{s_0})^T H^{-1} a_0]^2}{s_0^2(0.96)} \\
&= g^T H^{-1} g + 2\frac{g^T H^{-1} a_0}{s_0} + \frac{a_0^T H^{-1} a_0}{s_0^2} + \frac{[\frac{g^T H^{-1} a_0}{s_0} + \frac{a_0^T H^{-1} a_0}{s_0^2}]^2}{0.96}. \qquad (7)
\end{aligned}$$

Also, we know that

$$|\frac{g^T H^{-1} a_0}{s_0}| = |[(H)^{-\frac{1}{2}} g]^T [(H)^{-\frac{1}{2}} \frac{a_0}{s_0}]| \leq \|(H)^{-\frac{1}{2}} g\| \|(H)^{-\frac{1}{2}} \frac{a_0}{s_0}\| = \frac{1}{4}.\frac{1}{5} = 0.05. \qquad (8)$$

Using (8) in (7), we clearly have

$$\delta_d(x,\mu)^2 \leq \frac{1}{16} + 0.1 + 0.04 + \frac{(0.04 + 0.05)^2}{0.96} < \left(\frac{1}{2}\right)^2.$$

□

Lemma 5 *If we have a point x which is an approximate μ-center of the current polytope Q, and we add a hyperplane a_0 in Subcase 2.1, then $\delta_a(x,\mu) < \frac{1}{2}$, where the subscript a denotes quantities after adding the hyperplane a_0.*

Proof: We have x, with $\delta(x,\mu) < \frac{1}{4}$. Call the gradient $g := \nabla f(x,\mu) = \frac{c}{\mu} - \sum_{i=1}^{m} \frac{a_i}{s_i}$. Then

$$\nabla f_a(x,\mu) = g - \frac{a_0}{s_0} \quad \text{and} \quad H_a = H + \frac{a_0 a_0^T}{s_0^2}.$$

Therefore, by the Sherman-Morrison-Woodbury formula,

$$H_a^{-1} = H^{-1} - \frac{H^{-1} a_0 a_0^T H^{-1}}{s_0^2(1 + \frac{a_0^T H^{-1} a_0}{s_0^2})}.$$

Therefore,

$$\begin{aligned}[\delta_a(x,\mu)]^2 &\leq (g-\frac{a_0}{s_0})^T H^{-1}(g-\frac{a_0}{s_0}) \\ &= g^T H^{-1} g - 2\frac{g^T H^{-1} a_0}{s_0} + \frac{a_0^T H^{-1} a_0}{s_0^2},\end{aligned}$$

and, using (8) to bound the cross-product term,

$$\leq \frac{1}{16} + 2(\frac{1}{20}) + 0.04 < (\frac{1}{2})^2.$$

□

Note that if we instead add p cuts directly through the current iterate, the results of Goffin and Vial [6] can be used to show that a new approximate μ-center can be recovered in $O(p\log(p))$ Newton steps.

We now return to the case that only one cut is added at a time. From Lemmas 4 and 5, we can see that local convergence is not a problem in all but Case 2.2, which we have not established yet. It follows from the lemmas that in Case 1.1 as well as 2.1, all we need is one Newton step to find an approximate center [25]. As is well known in the case of long step methods, it is generally not true that the next μ-center can be found in $O(1)$ iterations after reducing the barrier parameter by a constant factor. In general, after the barrier parameter is reduced, Newton steps with a linesearch are taken to find the next approximate μ-center. So we cannot establish a result for Case 2.2 similar to the other cases. Instead, we show in the next section that the number of inner steps necessary after a barrier parameter reduction is bounded above by $O(nL)$. We then use this to establish the complexity of the algorithm. But first, however, we need to justify the lower bound that we use in Case 2.2, in the following simple and classical lemma.

Lemma 6 *Let x be an approximate μ-center of the current polytope Q. Then*

$$l = c^T x - \frac{5}{4}m\mu$$

is a valid lower bound on the optimal value of problem $\hat{P}$.

Proof: Since x is an approximate μ-center, we have

$$\|\frac{Sy}{\mu} - e\| \leq \frac{1}{4}.$$

Therefore

$$s^T y \leq (1+\frac{1}{4})m\mu = \frac{5}{4}m\mu.$$

Also, it is easily seen that $s^T y = c^T x - b^T y$, so

$$b^T y \geq c^T x - \frac{5}{4} m\mu.$$

Therefore $c^T x - \frac{5}{4} m\mu$ is a valid lower bound on the optimal value of $c^T x$ over the polytope Q. Since $\mathcal{X} \subseteq Q$, the result holds. □

It is interesting to note that the point x does not have to be feasible in $(\hat{P})$. This simple lemma is quite far-reaching, and we make significant use of it in the forthcoming analysis. The proof of the lemma immediately implies the following corollary:

Corollary 1 *If x is feasible in $(\hat{P})$ and if $1.25m\mu < 2^{-L}$ then the current duality gap is smaller than 2^{-L}, so the algorithm may be terminated with optimality. Further, we must always have $1.25m_{prev}\mu_{prev} \geq 2^{-L}$, otherwise the algorithm would have stopped in Case 2.2.*

5 Global Convergence

We prove global convergence and derive the complexity of our algorithm along the lines of [1] — first, we bound the maximum number of hyperplanes that can be present at any iteration by showing that if this upper bound is exceeded, the optimal set is empty (which is not the case since we assume the existence of a feasible point, and the feasible set is bounded). Then, using this limit on the number of hyperplanes, we show that in $O(nL^2)$ iterations, the polytope would become too narrow to contain a small enough ball around the optimal point(s), which again is not the case. Therefore, our algorithm must terminate in polynomial time. Also in this section are results pertaining to the number of innersteps necessary after a reduction of the barrier parameter. These lemmas are included in this section rather than the previous one (where they rightfully belong) because of the functions that need to be introduced.

We first state some propositions that we require in the analysis. These are to be found in [1], and are stated here without proof. Of course, the proofs in [1] are not exactly applicable, but the changes required are very slight. Let M be a positive definite matrix. Let $E(M, z, r)$ be the ellipsoid defined by

$$E(M, z, r) = \{x \in \Re^n : (x - z)^T M(x - z) \leq r^2\}. \tag{9}$$

Proposition 1 *Let x be a point such that $\delta(x, \mu) \leq 1$. Let H denote the Hessian of the barrier function at x, and let ω denote $x(\mu)$. Then,*

$$\omega \in E(H, x, \alpha)$$

where

$$\alpha \leq \frac{\delta(x,\mu)}{1-\delta(x,\mu)}.$$

(See Lemma 7, [1].)

Proposition 2 *If $y \in E(H(z),z,\alpha) \cap intQ$, and $\alpha < 1$, then*

$$(1+\alpha)^2 d^T\{H(z)\}^{-1}d \geq d^T\{H(y)\}^{-1}d \geq (1-\alpha)^2 d^T\{H(z)\}^{-1}d$$

for all $d \in \Re^n$, where $H(z)$ and $H(y)$ denote the Hessian of $f(x,\mu)$ at z and y respectively. (See Corollary 4, [1].)

The following proposition is useful in bounding linear terms over ellipsoids.

Proposition 3 *Let w be any fixed vector in $\Re^n$. Then we have*

$$\max \quad \{w^T(x-z) \quad : \quad x \in E(M,z,r)\} = r\sqrt{w^T M^{-1} w}.$$

(See eqn. (2), [1]).

We also include another proposition which deals with the error term in the Taylor expansion of the barrier function. For a proof, see Lemma 5.1, [29].

Proposition 4 *Let $y \in E(\nabla^2 f(z,\mu),z,\alpha)$, where $\alpha < 1$. We can expand $f(y,\mu)$ around the point z as*

$$f(y,\mu) = f(z,\mu) + \nabla f(z,\mu)^T(y-z) + \frac{1}{2}(y-z)^T\nabla^2 f(z,\mu)(y-z) + \ error.$$

Then the error in using this quadratic approximation satisfies

$$|error| \leq \frac{\alpha^3}{3(1-\alpha)}.$$

We can now prove a lemma showing that the width of the polytope Q^* can be bounded, which will enable us to show that Q^* is contained in an ellipsoid, and then to bound the number of constraints in any relaxation.

Lemma 7 *Let ω be a μ-center. Let $h^T x \geq b_h$ be any hyperplane used in defining the current polytope Q. Define width(h) := maximum variation of $h^T x$ over Q^*. Let ρ be the factor by which μ is reduced. Then,*

$$width(h) \leq (m + 2.5\frac{m_{prev}}{\rho})(h^T\omega - b_h)$$

Proof: Since ω is a μ-center, we know that at ω

$$\frac{c^T}{\mu} = \sum_{i=1}^{m} \frac{a_i^T}{s_i}$$

where $s_i \equiv s_i(\omega) = a_i^T\omega - b_i$. Therefore, for all x in Q, and in particular for all x in Q^*,

$$\frac{c^T(x-\omega)}{\mu} = \sum_{i=1}^{m} \frac{a_i^T(x-\omega)}{s_i}.$$

Adding m to both sides gives

$$m + \frac{c^T(x-\omega)}{\mu} = \sum_{i=1}^{m} \frac{a_i^T x - b_i}{s_i}.$$

Since $\omega \in Q$, we have from Lemma 6 that

$$c^T\omega \geq c^T x_{prev} - \frac{5}{4} m_{prev}\mu_{prev}.$$

For all $x \in Q^*$, we also have

$$c^T x \leq \max\{c^T\omega, c^T x_{prev}\} + \frac{5}{4} m_{prev}\mu_{prev}.$$

Combining these two gives $c^T x - c^T\omega \leq \frac{5}{2} m_{prev}\mu_{prev}$. Also, $\mu_{prev} = \frac{\mu}{\rho}$ because μ_{prev} is simply the value of the barrier parameter before it was last reduced by a factor of ρ. Therefore,

$$\sum_{i=1}^{m} \frac{a_i^T x - b_i}{s_i} \leq (m + 2.5\frac{m_{prev}}{\rho})$$

for all $x \in Q^*$. Now let $x^{\max}$ and $x^{\min}$ be the points that maximize and minimize $h^T x$ over Q^*. Then

$$\text{width}(h) = h^T x^{\max} - h^T x^{\min} \leq h^T x^{\max} - b_h.$$

Therefore,

$$\begin{aligned} \frac{\text{width}(h)}{h^T\omega - b_h} &\leq \frac{h^T x^{\max} - b_h}{h^T\omega - b_h} \leq \sum_{i=1}^{m} \frac{a_i^T x^{\max} - b_i}{s_i} \\ &\leq (m + 2.5\frac{m_{prev}}{\rho}). \end{aligned}$$

This completes the proof. □

Now we can prove a lemma that allows us to contain the polytope Q^* in an ellipsoid.

Lemma 8 *Let ω be a μ-center. Then, we can say the following about the polytope Q^*:*

$$Q^* \subseteq E(\nabla^2 f(\omega, \mu), \omega, \sqrt{m}(m + 2.5\frac{m_{prev}}{\rho})).$$

Proof: Again, denoting $\nabla^2 f(\omega, \mu)$ by H,

$$\begin{aligned} \forall x \in Q,\ (x-\omega)^T H(x-\omega) &= (x-\omega)^T \sum_i \frac{a_i a_i^T}{s_i^2}(x-\omega) \\ &= \sum_i [\frac{a_i^T(x-\omega)}{s_i}]^2. \end{aligned}$$

Now, $|a_i^T(x-\omega)| \leq \text{width}(a_i)$. Therefore, for all $x \in Q^*$,

$$\frac{|a_i^T(x-\omega)|}{s_i} \leq \frac{\text{width}(a_i)}{s_i} \leq (m + 2.5\frac{m_{prev}}{\rho}).$$

From this, it follows that

$$\forall x \in Q^*,\ (x-\omega)^T H(x-\omega) \leq m(m + 2.5\frac{m_{prev}}{\rho})^2.$$

Therefore,

$$Q^* \subseteq E(\nabla^2 f(\omega, \mu), \omega, \sqrt{m}(m + 2.5\frac{m_{prev}}{\rho})).$$

□

We now use Atkinson and Vaidya's lemma about the determinant of the Hessian. The proof of this lemma requires that unimportant constraints be dropped, ensuring that all remaining constraints have a significant impact on the Hessian.

Lemma 9 *Suppose that Case 2 is about to occur in the algorithm. At this point, the current polytope Q is determined by $\bar{m}$ hyperplanes in addition to those of the box and the lower-bound constraint of the form $c^T x \geq l$. Then,*

$$det(\nabla^2 f(\omega, \mu)) > 2^{-n(2L+1)}(1.01)^{\bar{m}}.$$

Proof: See [1], Theorem 13. The proof here very naturally extends to our algorithm. We note that the lower-bound constraint contributes another positive-definite term to the Hessian, and therefore can only increase the determinant. Therefore we can afford to ignore that constraint here. □

We are now in a position to demonstrate that the number of hyperplanes defining the polytope at any iteration can never get arbitrarily large.

Theorem 1 *There exists a constant ν independent of m, n and L, such that $\bar{m} < \nu nL$ throughout the course of the algorithm.*

Proof: First we recollect that $\bar{m}$ is simply the number of hyperplanes in addition to the box hyperplanes and the lower-bound hyperplane. We prove this theorem by showing that if $\bar{m} > \nu nL$, then the volume of the polytope Q^* must be too small to contain a ball of radius 2^{-L}.

We know from Lemma 8 that, for the current polytope Q^*,

$$vol(Q^*) \leq \frac{2^n[\sqrt{m}(m + 2.5\frac{m_{prev}}{\rho})]^n}{\sqrt{det(H)}}.$$

We may stop if

$$vol(Q^*) \leq (\frac{2^{-L}}{n})^n.$$

Thus, using the result of Lemma 9, and the fact that $(1.01)^{\bar{m}} > 2^{0.014\bar{m}}$, we just need

$$\frac{2\sqrt{m}(m + 2.5\frac{m_{prev}}{\rho})}{2^{-(L+0.5)}2^{0.007\bar{m}/n}} < (\frac{2^{-L}}{n}).$$

Consider the first iteration with $\bar{m} \geq \nu nL$. We thus have $\bar{m} = \nu nL$. We also have

$$m = 2n + 1 + \bar{m} \leq (\nu + 3)nL =: \pi nL$$

and also,

$$2.5\frac{m_{prev}}{\rho} \leq 2.5\frac{\pi nL}{\rho} \leq 5\pi nL$$

since ρ is always greater than $\frac{1}{2}$. We now want

$$\frac{2\sqrt{2}\sqrt{\pi nL}6\pi n^2 L}{2^{-2L}} < 2^{0.007\nu L}.$$

Separating terms and taking logarithms (to the base 2), and dividing by L, we want

$$\frac{1.5}{L} + 2.5\frac{\log n}{L} + \frac{\log 6}{L} + 1.5\frac{\log L}{L} + 2 < 0.007\nu - 1.5\frac{\log(\nu + 3)}{L}.$$

By definition, $L > \log_2 n > 1$. Therefore, the term on the left is bounded above by 10. Thus, $\nu = 4093$ is sufficient. □

Now all that remains to be shown is that if the number of hyperplanes always satisfies $m \leq \pi nL$, then the polytope Q will become too narrow to contain a ball of radius 2^{-L} in $O(nL^2)$ iterations. This would contradict our assumption concerning the width of the polytope, so the algorithm should have terminated earlier, with $1.25m_{prev}\mu_{prev}$ reaching a value smaller than 2^{-L}.

First, we examine the width of Q^*. If $\min_i(s_i(x)) \leq \frac{2^{-(3L+3)}}{6(\nu+3)n^{1.5}L}$, and x is an exact μ-center, then, from Lemma 7, we can conclude that for some i,

$$\text{width}(a_i) \leq 2^{-(3L+3)}/\sqrt{n}.$$

Since the hyperplanes are assumed to have norm 1, the width of the polytope Q^* has become too small to contain a ball of radius $2^{-3L-2}/\sqrt{n}$. Notice that Q^* is very narrow along a line that includes the current point. The following lemma relates the width of Q to the width of Q^*.

Lemma 10 *If Q^* has become too narrow to contain a ball of radius $2^{-3L-2}/\sqrt{n}$ then Q has become too narrow to contain a ball of radius 2^{-L}.*

Proof: We construct a cone K that contains $Q \setminus Q^*$ as follows. Consider any point on the optimal face of Q, and take the origin of the cone to be this point. Call this point x^*. Note that we do not need to know this point, we just need to know that it exists. Construct the cone so that the extreme rays of the cone pass through the extreme points of the intersection of Q^* with its supporting hyperplane given by $c^T x = \zeta$, where ζ is defined in equation (6). Thus, K is the cone generated by the optimal point and the face of Q^* given by the hyperplane $c^T x = \zeta$. Notice that any point $\check{x} \in Q$ with $c^T\check{x} > \zeta$ must be in K, since there is a point on the line segment joining x^* and $\check{x}$ which has value ζ, and by convexity of the feasible region, this point must be in Q^*, so the line segment is on a ray of K.

Because c has norm one, the largest possible value for $c^T x$ over $\{x : -2^L e \leq x \leq 2^L e\}$ is at most $\sqrt{n}2^{L+1}$ more than the optimal value $c^T x^*$. Let $K_b := K \cap \{x : c^T x \leq c^T x^* + \sqrt{n}2^{L+1}\}$. Notice that we have $Q \subseteq Q^* \cup K_b$ and that K_b is a scaled version of $K \cap Q^*$. The scale factor is $\sqrt{n}2^{L+1}/(\zeta - c^T x^*)$.

In order to bound the width of K_b, we need to compare ζ and $c^T x$. We have

$$\begin{array}{rcll} \zeta - c^T x^* & \geq & \zeta - c^T x_{prev} & \text{because } x_{prev} \text{ is feasible} \\ & \geq & 1.25 m_{prev}\mu_{prev} & \text{from the definition of } \zeta \\ & \geq & 2^{-L} & \text{from Corollary 1.} \end{array}$$

Assume s_i is the small slack in Q^*, so the width of Q^* along the direction a_i is less than $2^{-3L-1}/\sqrt{n}$. We can now bound the width of K_b in the direction a_i. This width must be no larger than

$$(\sqrt{n}2^{L+1}/2^{-L})(2^{-3L-1}/\sqrt{n}) = 2^{-L}.$$

Since $Q \subseteq Q^* \cup K_b$, the width of Q in any direction is no larger than the sum of the widths of Q^* and K_b. The result follows. □

Therefore, we may stop if any of the slacks become too small. We will show that this must happen after a polynomial number of iterations if μ has not already been reduced sufficiently to prove optimality.

We need to consider a potential function. In Lemma 11, we show that the potential function increases by a constant amount in Case 1 or Subcase 2.1 of the algorithm. In Lemma 12, we obtain an upper bound on the possible decrease of the potential function at each Subcase 2.2 iteration. Lemmas 13, 14, and 15 then enable us to bound the total number of inner steps in Subcase 2.2. We then combine these lemmas in Theorem 2 to show that the potential function must increase and that if the potential function reaches a certain value then the polytope Q is too narrow. Convergence follows.

Lemma 11 *The function*

$$\Delta(x, \mu, l) := \frac{c^T x - l}{\mu} - \sum_i \ln(\frac{s_i}{\kappa(i)}) \tag{10}$$

grows by at least a constant (Λ) at each iteration of the algorithm that results in Case 1 or Subcase 2.1 .

Proof: Recall that l is the best lower bound. We observe that in the iterations that we are concerned with in this lemma, this number l is not changed. Therefore, we can instead guarantee an increase in the function

$$N(x, \mu) := \frac{c^T x}{\mu} - \sum_i \ln(\frac{s_i}{\kappa(i)}). \tag{11}$$

We can prove this by considering each individual case. In Case 1.2, where a kappa value is merely reset, it is trivial to see that $N(x, \mu)$ increases by $\ln 2$, since x remains unchanged. Now consider the case where a hyperplane is dropped. Here, since μ is not modified, we suppress the dependence on μ, and simply write $N(x)$. Assume we have an exact μ-center ω, then, if we drop a hyperplane with index i,

$$N_d(\omega) = N(\omega) + \ln \frac{s_i}{\kappa(i)},$$

where the subscript d denotes quantities after dropping the hyperplane. Since the criteria for dropping the hyperplane are satisfied, we know that

$$N_d(\omega) \geq N(\omega) + \ln 2.$$

Now we need to bound $N_d(\omega) - N_d(\omega_d)$. Since the kappas are unchanged within this iteration, we may instead bound $f_d(\omega) - f_d(\omega_d)$. Firstly, we know from (7) that

$$\delta_d(\omega, \mu)^2 \leq 0.04 + \frac{(0.04)^2}{0.96} \leq 0.205^2$$

since $g = 0$. Therefore, we have $\delta_d(\omega, \mu) \leq 0.205$. Hence,

$$\omega_d \in E(\nabla^2 f_d(\omega), \omega, \alpha)$$

where, by Proposition 1,

$$\alpha = \frac{0.205}{1 - 0.205} < 0.258.$$

Now, we have

$$\begin{aligned} f_d(\omega_d) - f_d(\omega) &= \nabla f_d(\omega)^T(\omega_d - \omega) + \frac{1}{2}(\omega_d - \omega)^T \nabla^2 f_d(\omega)(\omega_d - \omega) + \text{ error} \\ &\geq \nabla f_d(\omega)^T(\omega_d - \omega) + \text{ error}. \end{aligned}$$

Using Proposition 3 to bound the linear term, and Proposition 4 to bound the error term, we have

$$f_d(\omega_d) - f_d(\omega) \geq -\alpha\sqrt{\nabla f_d(\omega)^T(\nabla^2 f_d(\omega))^{-1}\nabla f_d(\omega)} - \frac{\alpha^3}{3(1-\alpha)}.$$

We also have

$$\sqrt{\nabla f_d(\omega)^T[\nabla^2 f_d(\omega)]^{-1}\nabla f_d(\omega)} = \delta_d(\omega, \mu) \leq 0.205.$$

Therefore,

$$f_d(\omega_d) - f_d(\omega) \geq -(0.258)(0.205) - \frac{0.258^3}{3(1 - 0.258)} \geq -0.060605.$$

Therefore, we can guarantee that

$$N_d(\omega_d) - N(\omega) \geq \ln\ 2 - 0.060605 = 0.6325.$$

Now, assume that hyperplane $a_0^T x \geq \beta$ is being added. The analysis of this part very closely follows the analysis of Atkinson and Vaidya [1], and need not be repeated here. Thus, we can say that in all cases of the algorithm except Case 2.2, the function $\Delta(x, \mu)$ increases by at least a constant. □

Note that if p cuts are added directly through the current iterate then the increase in the potential function is at least as great as that obtained when just one shallow cut is added. Thus the approach of Goffin and Vial [6] can be used to regain an approximate μ-center in this case, without jeopardising the global convergence proof. It may be that many of the new constraints have λ_j greater than 2 at the new analytic center, so it is conceivable that we will need to perform $O(p)$ Newton steps to drop these additional constraints. The only change in the other lemmas in this paper is

that we may have to increase the upper bound on the maximum number of constraints in the problem by the maximum number of constraints added at any one time.

We return to the case of adding just one cut at a time. We now need to take care of Subcase 2.2 . First, we note that there can be no more than $O(L)$ executions of this subcase, since after that the barrier parameter would have become sufficiently small, leading to termination in Subcase 2.2. We then show that the number of inner steps required after an occurrence of Case 2.2 is at most $O(nL)$. It follows that there are at most $O(nL^2)$ Newton steps arising from Case 2.2.

Within each iteration of this type, we need to analyze what happens to $\Delta(x)$. Since the kappas remain unchanged in this iteration, we can instead work with the function

$$\Theta(x, \mu, l) := \frac{c^T x - l}{\mu} - \sum_{i=1}^{m} \ln(s_i) = \frac{c^T x - l}{\mu} + F(x). \tag{12}$$

We have a lemma about this function.

Lemma 12 *Assume we have a point x_1, which is the exact μ-center corresponding to a barrier parameter value μ_1. Case 2.2 of the algorithm occurs, and we reduce μ to μ_2, and find the appropriate center x_2. Let l_1 and l_2 denote the lower bounds that are active at the beginning and at the end of iteration step 2.2 .*

We then have

$$\Theta(x_2, \mu_2, l_2) \geq \Theta(x_1, \mu_1, l_1) - 6\nu nL. \tag{13}$$

Proof: Let there be m constraints. Let the index of the lower bound constraint be m. Let l be the potentially better lower bound that arises in this iteration : ie., $l = c^T x_1 - \frac{5}{4} m\mu_1$. It is important to note that since the lower bound constraint is used to define the polytope at any iteration, the function $F(x)$ in fact depends indirectly on the lower bound that is currently active. We make this explicit in this proof by modifying notation and using $F(x, l)$ instead of $F(x)$ — i.e., if there are m hyperplanes and the index of the lower bound constraint is m,

$$F(x, l) = -\sum_{i=1}^{m-1} \ln\ s_i - \ln(c^T x - l).$$

Let $x_0(\mu_0)$ denote the μ-center at which Case 2.2 occurred previously, and let m_0 be the number of hyperplanes at that stage. We know that $l_1 \geq c^T x_0 - 1.25 m_0 \mu_0$, since the lower bound l_1 was either adopted at this previous iteration, or was better than the potential lower bound at this iteration. Therefore

$$c^T x_1 - l_1 \leq c^T x_1 - c^T x_0 + 1.25 m_0 \mu_0.$$

But since x_0 was feasible in $(\hat{P})$, we know that

$$c^T x_0 \geq l = c^T x_1 - 1.25m\mu_1.$$

From these two and because $\frac{1}{\rho} < 2$, we can say that

$$\frac{c^T x_1 - l_1}{\mu_1} \leq 1.25(m + \frac{m_0}{\rho}) < 3.75\pi nL \leq 4\nu nL. \tag{14}$$

Now,

$$\begin{aligned}\Theta(x_2, \mu_2, l_2) - \Theta(x_1, \mu_1, l_1) &= \frac{c^T x_2 - l_2}{\mu_2} - \sum_{i=1}^{m-1} \ln s_i(x_2) - \ln(c^T x_2 - l_2) \\ &\quad - \frac{c^T x_1 - l_1}{\mu_1} + \sum_{i=1}^{m-1} \ln s_i(x_1) + \ln(c^T x_1 - l_1).\end{aligned}$$

Adding and subtracting $\ln(c^T x_2 - l_1)$, we have

$$\Theta(x_2, \mu_2, l_2) - \Theta(x_1, \mu_1, l_1) \geq -3.75\pi nL - \ln(\frac{c^T x_2 - l_2}{c^T x_2 - l_1}) + F(x_2, l_1) - F(x_1, l_1).$$

The logarithmic term can be neglected since it works in our favour. But we need to bound $F(x_2, l_1) - F(x_1, l_1)$. Since x_1 minimizes $\Delta(x, \mu_1, l_1)$, we have

$$\frac{c^T x_1}{\mu_1} + F(x_1, l_1) \leq \frac{c^T x_2}{\mu_1} + F(x_2, l_1).$$

Therefore we have

$$F(x_2, l_1) - F(x_1, l_1) \geq \frac{c^T x_1 - c^T x_2}{\mu_1}.$$

We can no longer say that $c^T x_2 \leq c^T x_1$, since by changing the lower bound, we have changed the polytope. However, since we know x_1 is feasible in $(\hat{P})$, we have from Lemma 6

$$c^T x_2 - 1.25m\mu_2 \leq c^T x_1.$$

Therefore, we have

$$\frac{c^T x_1 - c^T x_2}{\mu_1} \geq -1.25m\rho > -1.25m > -1.25\pi nL.$$

Thus, we have

$$\Theta(x_2, \mu_2, l_2) - \Theta(x_1, \mu_1, l_1) \geq -5\pi nL \geq -6\nu nL.$$

□

Thus, we see that any Case 2.2 occurrence can set back the value of $\Delta(x, \mu)$ by at most $O(nL)$. It is important to note that in equation (14) in this proof we merely use the fact that $\frac{1}{\rho} < 2$ — i.e., μ is reduced by no more than a factor of half. This is the reason we required $\rho \in (0.5, 1)$ rather than the usual $\rho \in (0, 1)$. Clearly though, any value between zero and one may be used, and only the numerical details of some proofs need to be changed.

Now, we need to establish that the next approximate μ-center can be found in a reasonable number of iterations. To do this, we bound $\Delta(x_1, \mu_2, l_2) - \Delta(x_2, \mu_2, l_2)$, and then show that a constant decrease can be obtained at each inner step, thus establishing what we need.

Lemma 13 *Assume we have a point x_1 that is the exact center corresponding to μ_1, and Case 2.2 occurs at this stage. Therefore, the lower bound may possibly have been reset from l_1 to l_2, and $\mu_2 = \rho\mu_1$. We can then say*

$$\Theta(x_1, \mu_2, l_2) \leq \Theta(x_1, \mu_1, l_1) + 5\nu nL.$$

Proof: Now

$$\Theta(x_1, \mu_2, l_2) - \Theta(x_1, \mu_1, l_1) = \frac{c^T x_1 - l_2}{\mu_2} - \frac{c^T x_1 - l_1}{\mu_1} + \ln(\frac{c^T x_1 - l_1}{c^T x_1 - l_2}).$$

From the definition of l_2, we have

$$\begin{aligned} \frac{c^T x_1 - l_2}{\mu_2} &\leq \frac{1.25m\mu_1}{\mu_2} \\ &< 2.5m \leq 2.5\pi nL. \end{aligned}$$

We can ignore the second term since

$$\frac{c^T x_1 - l_1}{\mu_1} \geq 0.$$

We need to bound the logarithmic term above. If the lower bound has not been reset and $l_2 = l_1$ then the logarithmic term is zero. Otherwise, first, we have from equation (14),

$$c^T x_1 - l_1 \leq 3.75\pi nL\mu_1,$$

and then, since $m \geq 2n$ always,

$$c^T x_1 - l_2 = 1.25m\mu_1 \geq 2.5n\mu_1.$$

Thus we can say

$$\begin{aligned}\ln(\frac{c^T x_1 - l_1}{c^T x_1 - l_2}) &\leq \ln(\frac{3.75\pi nL\mu_1}{2.5n\mu_1}) \\ &= \ln(1.5\pi L) \\ &\leq 1.5\pi L.\end{aligned}$$

Thus we have

$$\Theta(x_1, \mu_2, l_2) - \Theta(x_1, \mu_1, l_1) \leq 2.5\pi nL + 1.5\pi L \leq 4\pi nL \leq 5\nu nL.$$

□

Lemma 14 *Let each inner step at a Case 2.2 occurrence be taken based on a line-search. We can then say that the function $\Theta(x, \mu, l)$ is reduced at each inner step by at least a constant amount.*

Proof: Since neither μ nor l is modified at any of these inner steps, it suffices to prove the above lemma for $f(x, \mu)$. For a proof of this, see [26], Lemma 3.3. □

Lemma 15 *The total number of inner steps needed at each occurrence of Case 2.2 is no more than $O(nL)$.*

Proof: Follows from Lemmas 12,13 and 14. □

We can now complete the analysis.

Theorem 2 *The algorithm terminates in at most $O(nL^2)$ iterations.*

Proof: If we have the point x after p non-Case 2.2 iterations and q occurrences of Case 2.2 of the algorithm, then

$$\Delta(x, \mu) \geq \Lambda p - 6q\nu nL.$$

Because the norm of each constraint is equal to one, each $\kappa(i)$ may be bounded above by $3\sqrt{n}2^{L+1}$ (see [1]), so we have

$$F(x) = -\sum_i \ln(s_i) \geq \Lambda p - 6q\nu nL - m\ln(3\sqrt{n}2^{L+1}) - \frac{c^T x - l}{\mu}.$$

From (14), we have

$$\frac{c^T x - l}{\mu} \leq 5\nu nL.$$

Therefore, we have

$$F(x) \geq \Lambda p - 6q\nu nL - m\ln(3\sqrt{n}2^{L+1}) - 5\nu nL.$$

Also, it can be seen that if

$$F(x) \geq (\nu+3)nL[3L + 3 + \ln(6(\nu+3)n^{1.5}L)],$$

then stopping condition 2 will be met, since $-F(x)$ is the sum of the logs of the slacks, so at least one of the slacks must be smaller than

$$\exp(-(3L + 3 + \ln(6(\nu+3)n^{1.5}L))) < \frac{2^{-3L-3}}{6(\nu+3)n^{1.5}L},$$

which is sufficiently small. Since q can be no more than $O(L)$, it can be seen that $F(x)$ will definitely satisfy this in $O(nL^2)$ non-Case 2.2 iterations. Also, we note that the $O(L)$ Case 2.2 occurrences, each of which requires no more than $O(nL)$ inner steps, lead to a total of $O(nL^2)$ iterations. Consequently, the algorithm terminates in $O(nL^2)$ iterations. □

It is worth noting that in an actual implementation of this algorithm, we could add the cut right through the current point itself, as shown in, for example, [6, 22], and the new center can still be found easily enough. The basic idea is to first take an affine step to increase the slack associated with this new hyperplane from its current zero value, and then take centering steps to find the next approximate center. Details may be found in [6, 22].

6 Conclusions

We have extended the work of Atkinson and Vaidya to yield a column generation algorithm for linear programming that directly uses the logarithmic barrier function, and allows addition and deletion of columns. This improves upon their algorithm when applied to linear programs in that it is a long step method — potentially, much greater progress in objective function value will be achieved with a long step method than with a direct application of Atkinson and Vaidya's algorithm. Also, making progress in objective function value does not require the addition of constraints.

7 Appendix

In this section, we show how Atkinson and Vaidya's algorithm [1] can be applied directly to linear programming problems.

The Algorithm

Step 0 : Initialization

$$k \;=\; 0 \quad A = \begin{bmatrix} I_n \\ -I_n \end{bmatrix} \quad b = \begin{bmatrix} -2^L e_n \\ -2^L e_n \end{bmatrix}$$

$$x = [0 \cdots 0]^T \;; \quad s = Ax - b \;; \quad \kappa(i) = 2^L, \; i = 1 \; to \; 2n$$

Step k : The Iterative step

Have point x, which is an approximate analytic center.
H denotes the Hessian $\nabla^2 F(x)$. Calculate $\gamma_i(x) = \frac{a_i^T x - b_i}{\kappa(i)}$ for i=1 to m . For a hyperplane indexed by j, if $\gamma_j(x) > 2$, calculate $\sigma_j(x)$ as in equation (5).
Define $\Gamma := \max_i(\gamma_i(x))$.

Case 1 : $\Gamma > 2$

If for some j we have $\sigma_j(x) < 0.04$
then Subcase 1.1
Drop the hyperplane a_j, resulting in polytope Q_d.
Take primal-dual steps to find approximate analytic center x_d.
else Subcase 1.2
Let j be an index such that $\gamma_j(x) > 2$.
Reset $\kappa(j) = a_j^T x - b_j$.

Case 2 : $\Gamma \leq 2$

Subcase 2.1 : x is not feasible in $(\hat{P})$
Then $\exists$ j such that $\hat{a}_j^T x < \hat{b}_j$.
Choose β such that $\frac{\hat{a}_j^T H^{-1} \hat{a}_j}{(\hat{a}_j^T x - \beta)^2} = \frac{1}{16}$.
Add $\hat{a}_j^T x \geq \beta$ to the polytope, getting new polytope Q_a.
Find new approximate analytic center x_a.

Subcase 2.2 : x is feasible in $(\hat{P})$
Test x for optimality. If optimal, get dual solution y and STOP.
Else, Choose β such that $\frac{c^T H^{-1} c}{(-c^T x - \beta)^2} = \frac{1}{16}$.
Add $-c^T x \geq \beta$ to the polytope, getting new polytope Q_a.

Find new approximate analytic center x_a.

Stopping Criteria

1. If $m > \nu nL$, STOP.
2. If $\min_i s_i \leq \frac{2^{-(L+1)}}{\nu nL}$, STOP.

References

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

[2] O. Bahn, O. Du Merle, J. L. Goffin, and J. P. Vial. A cutting plane method from analytic centers for stochastic programming. *Mathematical Programming*, 69:45–73, 1995.

[3] J. Elzinga and T. J. Moore. A central cutting plane algorithm for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975.

[4] J.-L. Goffin, J. Gondzio, R. Sarkissian, and J.-P. Vial. Solving nonlinear multicommodity network flow problems by the analytic center cutting plane method. *Mathematical Programming*, 76:131–154, 1997.

[5] 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.

[6] 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. To appear in *Mathematical Programming*.

[7] J. Gondzio. Warm start of the primal-dual method applied in the cutting plane scheme. *Mathematical Programming*, 83:125–143, 1998.

[8] J. Gondzio, O. du Merle, R. Sarkissian, and J.-P. Vial. ACCPM — A library for convex optimization based on an analytic center cutting plane method. *European Journal of Operational Research*, 94:206–211, 1996.

[9] M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, Germany, 1988.

[10] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming, Algorithms and Complexity.* PhD thesis, Faculty of Mathematics and Informatics, TU Delft, NL–2628 BL Delft, The Netherlands, September 1992. Subsequently published by Kluwer Publishers, Dordrecht, The Netherlands, 1994.

[11] 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.

[12] D. den Hertog, C. Roos, and T. Terlaky. A build-up variant of the path-following method for LP. *Operations Research Letters*, 12:181–186, 1992.

[13] D. den Hertog, C. Roos, and T. Terlaky. Adding and deleting constraints in the logarithmic barrier method for LP. In D.-Z. Du and J. Sun, editors, *Advances in Optimization and Approximation*, pages 166–185. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[14] J. Kaliski, D. Haglin, C. Roos, and T. Terlaky. Logarithmic barrier decomposition methods for semi-infinite programming. *International Transactions in Operations Research*, 4:285–303, 1997.

[15] Z.-Q. Luo. Analysis of a cutting plane method that uses weighted analytic center and multiple cuts. *SIAM Journal on Optimization*, 7:697–716, 1997.

[16] Z.-Q. Luo, C. Roos, and T. Terlaky. Complexity analysis of a logarithmic barrier decomposition method for semi-infinite linear programming. *Applied Numerical Mathematics*, 29(3):379–395, 1999. Special issue dedicated to the HPOPT-I workshop held in Delft, September 19–20, 1996.

[17] N. Megiddo. On finding primal– and dual–optimal bases. *ORSA Journal on Computing*, 3:63–65, 1991.

[18] J. E. Mitchell. Computational experience with an interior point cutting plane algorithm. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, February 1997. Revised: March 1999, December 1999. Accepted for publication in *SIAM Journal on Optimization*.

[19] J. E. Mitchell. An interior point cutting plane algorithm for Ising spin glass problems. In P. Kischka and H.-W. Lorenz, editors, *Operations Research Proceedings, SOR 1997, Jena, Germany*, pages 114–119. Springer-Verlag, 1998.

[20] J. E. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In H. L. Frenk *et al.*, editor, *High Performance Optimization*, chapter 14, pages 349–366. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[21] Y. E. Nesterov and A. S. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming : Theory and Algorithms*. SIAM Publications. SIAM, Philadelphia, USA, 1993.

[22] 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.

[23] S. Ramaswamy and J. E. Mitchell. A long step cutting plane algorithm that uses the volumetric barrier. Technical report, DSES, Rensselaer Polytechnic Institute, Troy, NY 12180, June 1995.

[24] J. Renegar. A polynomial-time algorithm based on Newton's method for linear programming. *Mathematical Programming*, 40:59–93, 1988.

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

[26] C. Roos and J. P. Vial. Long steps with the logarithmic penalty barrier function in linear programming. In J. Gabszevwicz, J. F. Richard, and L. Wolsey, editors, *Economic Decision–Making : Games, Economics and Optimization, dedicated to J. H. Dreze*, pages 433–441. Elsevier Science Publisher B.V., Amsterdam, The Netherlands, 1989.

[27] G. Sonnevend. An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In A. Prekopa, J. Szelezsan, and B. Strazicky, editors, *Proceedings of the 12th IFIP Conference on Systems Modeling and Optimization, Budapest, 1985*, volume 84 of *Lecture Notes in Control and Information Sciences*, pages 866–876, Berlin, West–Germany, 1986. Springer Verlag.

[28] S. P. Tarasov, L. G. Khachiyan, and I. I. Erlikh. The method of inscribed ellipsoids. *Soviet Mathematics Doklady*, 37:226–230, 1988.

[29] P. M. Vaidya. A locally well-behaved potential function and a simple Newton-type method for finding the center of a polytope. In N. Megiddo, editor, *Progress in Mathematical Programming*, pages 79–90. Springer-Verlag, New York, 1989.

[30] P. M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73:291–341, 1996.

[31] Y. Ye. A potential reduction algorithm allowing column generation. *SIAM Journal on Optimization*, 2:7–20, 1992.

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

[33] Y. Ye, O. Güler, R. A. Tapia, and Y. Zhang. A quadratically convergent $O(\sqrt{n}L)$–iteration algorithm for linear programming. *Mathematical Programming*, 59:151–162, 1993.

[34] Y. Zhang, R. A. Tapia, and J. E. Dennis. On the superlinear and quadratic convergence of primal–dual interior point linear programming algorithms. *SIAM Journal on Optimization*, 2(2):304–324, 1992.