A Homogeneous Self-Dual Interior Point Cutting Plane Method

John E. Mitchell

Mathematical Sciences,
Rensselaer Polytechnic Institute,
Troy NY 12180 USA.

email: mitchj@rpi.edu
http://www.math.rpi.edu/~mitchj

Erling D. Andersen, Kees Roos, Tamas Terlaky

ITS/TWI/SSOR,
Delft University of Technology,
Delft, 2628 CD , The Netherlands.

Seattle INFORMS Conference
October 25–28, 1998
Session SC27

Abstract

We investigate adding cutting planes within the context of the homogeneous self-dual linear programming algorithm. Restart methods are described and complexity analysis is presented.

1 Overview

- Column generation applications
- Homogeneous self-dual methods for linear programming
- Convergence proof for one variant
- An implemented variant
- Computational results
- Conclusions

2 Introduction

- Interested in linear programming problems with a **large number of constraints** or a **large number of variables**.
- Solve problems using **constraint generation** or **column generation** methods.
- Solve linear programming subproblems using **interior point methods**
- These techniques are used to solve:
  - Integer programming problems
  - Nondifferentiable optimization problems
  - Multicommodity network flow problems
  - Stochastic programming problems
  - Semi-infinite programming problems
  - Airline crew scheduling problems
  - Bounded error parameter estimation
  - Adaptive filtering

3 Homogeneous self-dual linear programs

Standard primal-dual pair of linear programs:

$$\begin{array}{llcl} \min & c^T x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \end{array} \qquad (\mathbf{P}) \qquad \begin{array}{llcl} \max & b^T y & & \\ \text{subject to} & A^T y & \leq & c \end{array} \qquad (\mathbf{D})$$

where x and c are n-vectors, b and y are m-vectors, and A is an $m \times n$ matrix.

Let $\hat{b} = b - Ae$ and $\hat{c} = c - e$, where e denotes the vector where every component is equal to one. Let $\hat{d} = c^T e + 1$. Construct the following linear programming problem:

$$\begin{array}{lrcrcrcrcl} \min & & & & & & & (n+1)w & & \\ \text{subject to} & & & Ax & - & bt & + & \hat{b}w & = & 0 \\ (\mathbf{HLP}) & -A^T y & & & + & ct & - & \hat{c}w & \geq & 0 \\ & b^T y & - & c^T x & & & + & \hat{d}w & \geq & 0 \\ & -\hat{b}^T y & + & \hat{c}^T x & - & \hat{d}t & & & = & -(n+1) \\ & & & x, t & \geq & 0, & & y, w & & \text{free}, \end{array}$$

where t and w are scalars.

Properties of (HLP)

1. $x = e$, $y = 0$, $t = 1$, $w = 1$ is **strictly feasible** in (**HLP**).
2. The **dual** problem to (**HLP**) is equivalent to (**HLP**).
3. (**HLP**) has optimal value 0.
4. Let (x^*, y^*, t^*, w^*) be an optimal solution to (**HLP**). Assume $t^* > 0$. Then x^*/t^*, y^*/t^* are **optimal** to (P) and (D).
5. If $t^* = 0$ in a **strictly complementary** optimal solution (x^*, y^*, t^*, w^*) then either $x^* \geq 0$ with $Ax^* = 0$ and $c^T x^* < 0$ or $A^T y^* \leq 0$ and $b^T y^* > 0$, so either (P) or (D) is **infeasible**.

Solving (HLP)

- An **interior point method** can be used to find a *strictly complementary* optimal solution to (**HLP**).
- The linear algebra required to perform an iteration is comparable to that required with other formulations.
- The numerical stability is comparable, and is able to identify infeasibility efficiently.

4 Convergence for the feasibility problem

- Solve the problem:

  Does there exist $y \in C \subseteq \Re^m$?

  where C is a convex set defined by an oracle.

- Assume that if C is nonempty, it contains a ball of radius ϵ. Assume $C \subseteq \{y : -e \leq y \leq e\}$.

- **Initialize** with dual feasible region

$$\{y : -e \leq y \leq e\},$$

  where e denotes a vector of ones of appropriate dimension.

- Assume know primal and dual strictly feasible iterates for the current relaxation.
  - Initially, $y = 0$, $x = e$.
  - Add constraints through current iterate. Use a direction to regain strictly feasible iterates.

- Current homogeneous problem has the form:

$$
\begin{array}{lccccl}
\min & & & (n+1)w & & \\
\text{subject to} & Ax & & & = & 0 \\
(\mathbf{HSDF}) & -A^T y & +c\tau & & \geq & 0 \\
& -c^T x & & +(n+1)w & \geq & 0 \\
& & -(n+1)\tau & & = & -(n+1) \\
& x, & \tau, & w & \geq & 0.
\end{array}
$$

- Define $s := c\tau - A^T y$, $\kappa := (n+1)w - c^T x$.

- Primal-dual logarithmic **barrier**:

$$
\begin{aligned}
&(x, s, \tau, \kappa) := \\
&\quad x^T s + \tau\kappa - (n+1) - \Sigma_{i=1}^n \ln(x_i s_i) - \ln(\tau\kappa).
\end{aligned}
$$

- Our **proximity measure** is

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

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

$$u_i := \begin{cases} \sqrt{\tau\kappa} & \text{if } i = 0 \\ \sqrt{x_i s_i} & \text{otherwise} \end{cases} \tag{2}$$

$$(u^{-1})_i := 1/u_i \tag{3}$$

- (x, s, τ, κ) is **approximately centered** if it satisfies $\delta(x, s, \tau, \kappa) < \theta$ for some θ, with $0 < \theta < 0.5$.

The algorithm

1. **Initialize** with $w = 1$.
2. If the problem has too many dual constraints, **STOP with infeasibility**.
3. Find an **approximate center** for the current value of w.
4. Call the **separation oracle** at the point y/τ.
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**.

Restarting after adding a cut

- The **new primal** problem is

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

- The **new dual** problem is

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

- 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:

$$d_x = -D^2 A^T (AD^2 A^T)^{-1} a_{n+1} \qquad (4)$$

$$d_y = -(AD^2 A^T)^{-1} a_{n+1}, \qquad (5)$$

where $D := X^{0.5} S^{-0.5}$.

- So we **update** for a positive scalar α:

$$x^+ := x + \alpha d_x \qquad (6)$$

$$x_{n+1} := \alpha \qquad (7)$$

$$y^+ := y + \alpha d_y \qquad (8)$$

Finding a new approximate center

- Define

$$\Delta_{PD} := \sqrt{a_{n+1}^T (AD_{PD}^2 A^T)^{-1} a_{n+1}}. \qquad (9)$$

- Take step length

$$\alpha = \beta / \Delta_{PD}. \qquad (10)$$

- The new iterates are strictly primal and dual **feasible** for appropriate β.

Theorem 1 *Assume the old iterate was approximately centered. Let* δ^+ *denote the measure of centrality after taking the defined step. If* $\beta = \sqrt{0.5(1-2\theta)}$ *then* $(\delta^+)^2 \leq \theta^2 + 0.5(-1 + 3/(1-2\theta))$*. If* $\theta = 0.25$ *then* $\delta^+ \leq 1.601$*.*

Theorem 2 *If* $\theta = 0.25$ *then the potential function value at the new iterate is* $^+ \leq 8.672$*. A new approximate center can be regained in* $O(1)$ *Newton steps.*

(Proof uses Lemmas II.67 and II.70 and Theorem II.49 in *Theory and Algorithms for Linear Optimization*, by Roos, Terlay, and Vial.)

Global convergence

Theorem 3 *The upper bound on the number of constraints that need to be added before the algorithm can be terminated with infeasibility is* $O(m^2/\epsilon^2)$*, and the number of Newton steps required has the same complexity.*

This follows from a very similar analysis to that of Goffin, Luo, Ye.

5 An implemented variant

Recall (**HLP**):

$$
\begin{array}{llcccccccl}
\min & & & & & & (n+1)w & & \\
\text{subject to} & & Ax & - & bt & + & \hat{b}w & = & 0 \\
(\mathbf{HLP}) & -A^T y & & + & ct & - & \hat{c}w & \geq & 0 \\
& b^T y \; - & c^T x & & & + & \hat{d}w & \geq & 0 \\
& -\hat{b}^T y \; + & \hat{c}^T x & - & \hat{d}t & & & = & -(n+1) \\
& & x, t & \geq & 0, & & y, w & & \text{free,}
\end{array}
$$

In practice, we want to:

- Solve the **optimization** problem.
- Add **deep cuts**, so current iterate is infeasible.
- Add **many cuts** simultaneously.

- Notation: We add columns to A. We denote the index set of the current columns by I. We let a_j denote the jth column of A.

Our algorithm

1. **Initialize:** Choose initial I^0, choose β, ϵ.
2. Set $w^0 = 1$, $(x^0, t^0, y^0, s^0, \kappa^0) = (e, 1, 0, e, 1)$.
3. Iteration counter $k := 0$.
4. **Outer loop:** while $w^k > \beta\epsilon$:
   (a) **Inner loop:**
   For fixed w^k, find approximate analytic center for feasible region for (**HLP**).
   (b) **Find cuts:** Let $J := \{j : a_j^T y^k > c_j t^k\}$.
   (c) If $J \neq \emptyset$:
   (d) **Add a subset of violated constraints:**
   (e) Let $\hat{J} \subseteq J$ and $\hat{J} \neq \emptyset$. Update $I^{k+1} := I^k \cup \hat{J}$.
   (f) $w^{k+1} := w^k$.
   (g) else:
   (h) Reduce parameter: $w^{k+1} := \beta w^k$.
   (i) Let $I^{k+1} := I^k$.
   (j) end if
   (k) Update iteration count: $k := k + 1$.
5. end while

Recentering after adding cuts

- For a given index set I and variable w, let $Q(I, w)$ be the set of points (x, y, s, t) that are **approximately centered** in the feasible region of the corresponding problem (**HLP**).
- When we add cuts J, get $I' := I \cup J$.
- We have $(\tilde{x}, \tilde{y}, \tilde{s}, \tilde{t}) \in Q(I, w)$, and we need to find $(x', y', s', t') \in Q(I', w)$.

- Thus, need to approximately solve:

$$
\begin{array}{lllll}
\text{min} & \ln(t\kappa) + \Sigma_{j\in I'} \ln(x_j s_j) & & \\
\text{subject to} & A_{I'}x_{I'} - bt & = & w(A_{I'}e - b) \\
(\mathbf{AC'}) & A_{I'}^T y + s_{I'} - c_{I'}t & = & w(e - c_{I'}) \\
 & -c_{I'}^T x_{I'} + b_{I'}^T y - \kappa & = & w(-c_{I'}^T x - 1) \\
 & x_{I'}, s_{I'}, t, \kappa & \geq & 0.
\end{array}
$$

- A first iterate $(\hat{x}, \hat{y}, \hat{s}, \hat{t}, \hat{\kappa})$ for $(\mathbf{AC'})$:

$$
\begin{array}{ll}
\hat{y} := \tilde{y}, \quad \hat{t} := \tilde{t}, \quad \hat{\kappa} := \tilde{\kappa} & \\
\hat{x}_i := \tilde{x}_i, \;\; \hat{s}_i := \tilde{s}_i & \text{for } i \in I \\
\hat{x}_i := w, \;\; \hat{s}_i := 1 & \text{for } i \in J.
\end{array}
$$

- This iterate is infeasible in $(\mathbf{AC'})$.

- Use the **infeasible-primal-dual** method to recover a new approximate analytic center, for $(\mathbf{AC'})$.

Results: block angular LPs

Name	nrow	ncol	mnrow	nblks	gncol
cocof	63	96	8	2	21
cocot	63	96	38	24	75
decomp1	7	5	2	2	4
decomp2	7	7	2	2	4
decomp3	7	7	2	2	5
hpw3at	66	126	35	6	43
hpw3p	108	126	30	14	53
hpw3t	108	126	35	48	92
hpw7m	72	65	15	4	27
hpw7t	72	65	16	5	32
psc07-01	1820	1839	181	93	556
psc07-02	1738	1757	181	106	455
ptcd07-01	1820	1839	270	34	304
ptcd07-02	1738	1757	270	42	338
pds-1	1474	3729	87	11	237
pds-2	2954	7535	181	11	372
pds-3	4594	12287	303	11	607

Table 1: **Problem descriptions**

nrow: Number of rows
ncol: Number of columns
mnrow: Number of rows in master problem
nblks: Number of blocks
gncol: Number of columns added to the master

Name	Outer	Inner	Rgap
cocof	25	56	1.1e-008
cocot	22	48	6.1e-008
decomp1	11	19	5.4e-008
decomp2	11	18	5.4e-008
decomp3	11	16	3.1e-008
hpw3at	22	40	1.2e-010
hpw3p	21	48	1.4e-008
hpw3t	26	49	8.0e-009
hpw7m	19	41	5.9e-009
hpw7t	20	43	7.6e-009
psc07-01	52	159	2.2e-009
psc07-02	41	128	3.8e-009
ptcd07-01	60	145	2.4e-009
ptcd07-02	64	156	2.4e-009
pds-1	66	218	2.7e-008
pds-2	79	250	1.9e-007
pds-3	131	402	7.5e-008

Table 2: **Computational results**

Outer: Number of outer iterations = Number of oracle calls

Inner: Number of Newton steps used to reoptimize the master.

Rgap: Relative duality gap

Comments:

1. The **stopping criteria** require that $x_i, s_i \geq -10^{-6}$ should be satisfied, and that the relative duality gap is no larger than 10^{-6}.
2. Relatively **few columns** are generated.
3. Number of **outer iterations** are reasonable, but not extremely good.
4. The reoptimization seems to work well. On average **3–4 iterations** per iteration. However, the warm-start can take a large number of iterations if really deep cuts are added.
5. The algorithm is generally slow timewise, because the master problem gets dense.
6. This is really a straightforward implementation.
7. Each subproblem is solved from scratch using an interior code without warmstart, which of course makes things slower.
8. Due to rounding errors and finite precision of the computations, it is occasionally difficult to to determine whether a cut is genuine.

So far, the times are not that good. However there are several things to try:

1. Use the simplex algorithm or interior-point algorithm with **warm-start** to solve the **subproblems**.
2. Use the simplex method to generate **alternate optimal extreme points**, so generate more cuts per subproblem per major iterations. (For the pds-* problems, the same three subproblems keeps on generating cuts.)
3. Try to improve the **warmstart** procedure for the master problem.

6 Conclusions

- We have shown that a interior point cutting plane method based on the Goldman-Tucker homogeneous model can be used to solve the convex feasibility problem in a **fully polynomial** number of calls to an oracle, with a similar number of Newton steps.
- The proof of this result used a **different measure of centrality** than other interior point cutting plane proofs.
- Initial **computational experience** with an interior point homogeneous cutting plane algorithm is promising, but needs more work.

References

[1] K. H. Afkhamie, Z.-Q. Luo, and K. M. Wong. Interior point column generation algorithms for adaptive filtering. Technical report, Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, L8S 4L7, Canada, June 1997.

[2] E. D. Andersen and K. D. Andersen. The APOS linear programming solver: an implementation of the homogeneous algorithm. Technical Report 9337, CORE, Universite Catholique de Louvain, Belgium, April 1997.

[3] 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.

[4] E.-W. Bai, Y. Ye, and R. Tempo. Bounded error parameter estimation: A sequential analytic center approach. Technical report, Department of Management Science, University of Iowa, Iowa City, Iowa 52242, February 1997.

[5] 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.

[6] J.-L. Goffin, A. Haurie, and J.-P. Vial. Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science*, 38:284–302, 1992.

[7] 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.

[8] J. Kaliski, D. Haglin, C. Roos, and T. Terlaky. Logarithmic barrier decomposition methods for semi-infinite programming. Technical Report 96–51, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, The Netherlands, 1996.

[9] J. E. Mitchell. Interior point methods for combinatorial optimization. In Tamás Terlaky, editor, *Interior Point Methods in Mathematical Programming*, chapter 11, pages 417–466. Kluwer Academic Publishers, 1996.

[10] 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.

[11] J. E. Mitchell and B. Borchers. Solving real-world linear ordering problems using a primal-dual interior point cutting plane method. *Annals of Operations Research*, 62:253–276, 1996.

[12] J. E. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180–3590, September 1997. Accepted for publication in *Proceedings of HPOPT97*, Rotterdam, The Netherlands.

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

[14] 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.

[15] Y. Ye, M. J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19:53–67, 1994.