CENTER FOR PARALLEL OPTIMIZATION

AN INTERIOR DUAL PROXIMAL POINT ALGORITHM FOR LINEAR PROGRAMS

by

R. Setiono

Computer Sciences Technical Report #879

September 1989

An Interior Dual Proximal Point Algorithm for Linear Programs

Rudy Setiono*

September 1989

Abstract

An interior point algorithm for obtaining a proximal point solution of a linear program is presented. Results from our implementation of this algorithm have been very encouraging. For 36 test problems including 32 NETLIB problems, we obtain a total time speedup of 5.6 over the MINOS 5.0 simplex package. We also describe an implementation of our algorithm for linear programs with upper-bounded variables, such as the multicommodity Patient Distribution System models of the Military Airlift Command. We have been able to solve some of these multicommodity problems with 8-figure accuracy and speedup of as much as 24 over the MINOS 5.0. Furthermore our run times on the Astronautics ZS-1 are comparable with those of AT&T's KORBX times for some of the problems.

1 Introduction

In this paper we present very encouraging computational results for an interior point algorithm applied to the dual of a proximal point formulation of a linear program. One fundamental reason for this approach is that the

*Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706. Research supported by National Science Foundation Grants DCR-8521228 and CCR-8723091 and Air Force Office of Scientific Research Grants AFOSR-86-0172 and AFOSR-89-0410

interiority condition for the dual problem is trivially satisfied by maintaining positivity of some of the dual variables. The key ingredients of the method are :

1. The nonnegatively-constrained convex quadratic objective function of the dual of the proximal point linear program.

2. The logarithmic penalty formulation of the dual problem.

3. Execution of one step of the Newton method using an efficient sparse linear algebra package.

The method was tested on 36 problems, including 32 NETLIB problems and on the multicommodity Patient Distribution System (PDS) model of the Military Airlift Command. On the 36 test problems, our method outperformed the MINOS 5.0 simplex package by a ratio of 5.6 to 1 in total time. On the PDS models we obtained a speedup of up to 24 over MINOS. We were able to solve the 10-day PDS model (16558 rows, 49932 columns) on the Astronautics ZS-1 pipeline vector machine (with megaflop rate of 5.2-8.2) in 4.7 hours. In contrast the same problem was solved in 3.3 -4.5 hours on AT&T's KORBX machine [2] (with estimated Mflop rate of 8.5-69.3) using various versions of Karmarkar's method [7].

2 Problem Formulation

We consider linear program in the standard form

$$\min_x cx \ \text{ s.t. } Ax = b,\ x \geq 0 \tag{1}$$

where $c \in \mathbb{R}^n, \mathrm{b} \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$ and its dual

$$\max_{u,v} bu \ \text{ s.t. } A^t u + v = c,\ v \geq 0 \tag{2}$$

The proximal point algorithm [1,14] applied to the linear program (1) generates a sequence $\{x^i\}$ as follows:

$$x^{i+1} := \arg\min\ cx + \frac{\epsilon^i}{2}\left\|x - x^i\right\|^2 \ \text{ s.t. } Ax = b,\ x \geq 0 \tag{3}$$

where $\{\epsilon^i\}$ is a bounded sequence of positive numbers and $x^0 \in \mathbb{R}^n$. Instead of applying an interior method to the subproblem (3), we choose to do that to its dual which is considerably simpler. The dual [9] of the quadratic program (3) is

$$\min_{v \geq 0, u} \frac{1}{2} \left\| A^t u + v - c + \epsilon^i x^i \right\|^2 - \epsilon^i b u \tag{4}$$

If we let $(u(\epsilon^i), v(\epsilon^i))$ be a solution of the dual program (4), then the primal solution $x(\epsilon^i)$ is given by

$$x(\epsilon^i) = x^i + \frac{1}{\epsilon^i}(A^t u(\epsilon^i) + v(\epsilon^i) - c)$$

This in effect is an augmented Lagrangian [1,13] update for the dual linear program (2). Our proposed algorithm consists of taking one Newton step for the log penalty barrier function associated with subproblem (4) with appropriately decreasing values of ϵ^i and the penalty parameter γ^i. We describe the algorithm in detail in the next section.

3 Quadratic Interior Dual Proximal Point Algorithm (QIDPP)

Consider the barrier penalty minimization problem associated with the dual problem (4) with penalty parameter $\gamma^i > 0$

$$\min_{u,v} F(u,v) := \frac{1}{2} \left\| A^t u + v - c + \epsilon^i x^i \right\|^2 - \epsilon^i b u - \gamma^i \sum_{j=1}^{n} \log v_j \tag{5}$$

The gradient and the Hessian of $F(u,v)$ are

$$\nabla F(u,v) = \begin{pmatrix} \nabla_u F(u,v) \\ \nabla_v F(u,v) \end{pmatrix} = \begin{pmatrix} A\left(A^t u + v - c + \epsilon^i x^i\right) - \epsilon^i b \\ A^t u + v - c + \epsilon^i x^i - \gamma^i V^{-1} \mathrm{e} \end{pmatrix}$$

$$\nabla^2 F(u,v) = \begin{pmatrix} AA^t & A \\ A^t & I + \gamma^i V^{-2} \end{pmatrix}$$

where $V := diag(v_1, v_2, \ldots, v_n)$. The Newton direction can then be obtained by solving the linear system

$$\nabla^2 F(u^i, v^i) \begin{pmatrix} u - u^i \\ v - v^i \end{pmatrix} + \nabla F(u^i, v^i) = 0$$

We are now ready to state the complete algorithm.

<u>Algorithm QIDPP</u>:

- Initialization

  1. Choose any $u^0 \in \mathbb{R}^m$, $v^0 \in \mathbb{R}^n_+$. Set $i = 0$
     Choose $\gamma^0 > \gamma_{min} > 0$ and $\epsilon^0 > \epsilon_{min} > 0$ and $0 < \rho < 1$.
     (ρ is an attenuation factor for ϵ and γ)
  2. Compute
  $$x^0 := \frac{1}{\epsilon^0}\left(A^t u^0 + v^0 - c\right)$$

- Iteration

  1. Solve the linear system
  $$\nabla^2 F(u^i, v^i)\begin{pmatrix} u - u^i \\ v - v^i \end{pmatrix} + \nabla F(u^i, v^i) = 0 \tag{6}$$
     Let $(\overline{u}^i, \overline{v}^i)$ be the solution of the above linear system.
  2. Update
  $$\begin{aligned} x^{i+1} &:= x^i + \frac{1}{\epsilon^i}\left(A^t \overline{u}^i + \overline{v}^i - c\right) \\ u^{i+1} &:= \overline{u}^i \end{aligned} \tag{7}$$
  3. Compute stepsize λ
  $$\begin{aligned} \lambda &:= \min\left\{\min_{j\in J}\left(\frac{v^i_j}{v^i_j - \overline{v}^i_j}\right), 1\right\} \\ where\ J &:= \{j | v^i_j - \overline{v}^i_j > 0\} \end{aligned} \tag{8}$$
  4. Update
  $$v^{i+1} := v^i + 0.98\lambda\left(\overline{v}^i - v^i\right)$$

- Termination
  If $(x^{i+1}, u^{i+1}, v^{i+1})$ is feasible to the primal-dual programs (1) and (2) and $|cx^{i+1} - bu^{i+1}|$ is sufficiently small, then stop
  Else

1. Set $i := i + 1$
2. if $\gamma^i > \gamma_{min}$ then $\gamma^{i+1} = \rho\gamma^i$
   if $\epsilon^i > \epsilon_{min}$ then $\epsilon^{i+1} = \rho\epsilon^i$
3. Go to Iteration

Remark 1 *Choosing an interior point to start this algorithm is trivial, since the dual problem (4) has only nonnegativity constraints. This is the main advantage of this algorithm over the primal algorithm implemented by Gill et al [6], the dual affine algorithm implemented by Monma and Morton [11] or the primal-dual affine algorithm implemented by McShane et al [10] and Lustig [8] where a phase I is needed to start the algorithms.*

Remark 2 *Solution of the $m+n$ linear system (6) in the $m+n$ variables (u,v) can be achieved by first solving the m linear equations in m unknowns*

$$A\left[I - \left(I + \gamma^i(V^i)^{-2}\right)^{-1}\right] A^t\left(u - u^i\right) =$$

$$A\left(I + \gamma^i(V^i)^{-2}\right)^{-1} \nabla_v F(u^i, v^i) - \nabla_u F(u^i, v^i) \tag{9}$$

for u and then computing

$$v - v^i = -\left(I + \gamma^i(V^i)^{-2}\right)^{-1}\left(\nabla_v F(u^i, v^i) + A^t\left(u - u^i\right)\right)$$

In our implementation, the Yale Sparse Matrix Package [3,4] was used to solve the system of linear equations (9).

Remark 3 *By using $(\overline{u}^i, \overline{v}^i)$ as opposed to (u^{i+1}, v^{i+1}) in computing x^{i+1}, we are guaranteed to have a sequence $\{x^i\}$ such that $Ax^i = b$, except for possibly x^0.*

4 Computational Results

The algorithm QIDPP was implemented on a Microvax 3200. The code was written in FORTRAN and the FORTRAN f77 compiler was used with the "-O" option. All the test problems that we solved are publicly available through *Netlib* [5], except 4 problems : Rabo, Truss1, Truss2 and Truss3.

Rabo comes from the mortgage division of Rabo Bank of the Netherlands. The truss problems are structural design problems which were made available by Prof. Michael C. Ferris of the University of Wisconsin. The dimensions of all our linear programming test problems are given in Table 1.

For comparison purposes, we solved these problems using MINOS 5.0 [12], one of the most widely used linear programming packages. Results obtained by MINOS 5.0, using the default parameter settings, are given in Table 2 for the 36 test problems.

Table 3 shows the results from the QIDPP algorithm. We have used *one* set of parameter values that we found work best for *all* 36 test problems. The objective values for QIDPP and MINOS agree to 7 or 8 digits for most problems. Additional information about QIDPP results are listed in Table 4. The primal infeasibility, dual infeasibility and complementarity are computed as follows:

$$\begin{aligned} Primal\ Infeasibility &= \frac{\|Ax-b\|}{\|b\|} \\ Dual\ Infeasibility &= \frac{\|(A^tu-c)_+\|}{\|(-c)_+\|} \\ Complementarity &= \frac{\|X(c-A^tu)\|}{\|x\|\ \|u\|} \end{aligned}$$

where $X := diag(x_1, x_2, \ldots, x_n)$.

The relative primal infeasibility achieved by this algorithm is on the average 1 order of magnitude better than either an interior primal proximal point algorithm or a pure primal interior point algorithm [15]. Finally, in Table 5 we compare the run times of MINOS and QIDPP. On these 36 test problems our total time speedup over MINOS was 5.6.

5 Linear Programs with upper bounds

Since many important linear programs, such as multicommodity problems with capacitated arcs, have upper bounds on some or all of the variables, we discuss in this section methods for handling such bounds without increasing the size of the constraint matrix.

Consider the following linear programs where all of the variables are bounded

$$\min_x cx \text{ s.t. } Ax = b,\ 0 \le x \le d \tag{10}$$

This problem is equivalent to

$$\min_{x,y} cx \tag{11}$$

subject to

$$\begin{aligned}\begin{pmatrix} A & 0 \\ I & I \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} b \\ d \end{pmatrix} \\ x, y &\ge 0\end{aligned}$$

This is a problem with $m+n$ constraints in $2n$ variables. The dual of the linear program (10) is

$$\min_{u,v} bu + dv \tag{12}$$

subject to

$$\begin{aligned} A^t u + v &\le c \\ v &\le 0 \end{aligned}$$

The primal and dual proximal point minimization problems corresponding to the above primal-dual programs are respectively

$$\min_{x,y} cx + \frac{\epsilon^i}{2}\left\|x - x^i\right\|^2 + \frac{\epsilon^i}{2}\left\|y - y^i\right\|^2 \tag{13}$$

subject to

$$\begin{aligned}\begin{pmatrix} A & 0 \\ I & I \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} b \\ d \end{pmatrix} \\ x, y &\ge 0\end{aligned}$$

and

$$\min_{(w,z)\ge 0, u, v} \frac{1}{2}\left\|A^t u + v + w - c + \epsilon^i x^i\right\|^2 + \frac{1}{2}\left\|v + z + \epsilon^i y^i\right\|^2 - \epsilon^i bu - \epsilon^i dv \tag{14}$$

The barrier penalty minimization problem associated with the dual problem (14) is

$$\begin{aligned}\min_{u,v} G(u,v,w,z) := \frac{1}{2}\left\|A^t u + v + w - c + \epsilon^i x^i\right\|^2 &+ \frac{1}{2}\left\|v + z + \epsilon^i y^i\right\|^2 \\ -\epsilon^i bu - \epsilon^i dv - \gamma^i \sum_{j=1}^n \log w_j &- \eta^i \sum_{j=1}^n \log z_j\end{aligned}$$

where $\gamma^i > 0$ and $\eta^i > 0$. The total number of variables in this problem is $m + 3n$. This dimensionality is considerably bigger than the n-dimensional size of the original linear program (10). However, we will give a transformation which will enable us to solve a system of equations in m unknowns at each iteration of the proximal point algorithm.

The optimality condition for the last problem above is

$$\begin{aligned} \nabla G(u,v,w,z) &:= \begin{pmatrix} \nabla_u G(u,v,w,z) \\ \nabla_v G(u,v,w,z) \\ \nabla_w G(u,v,w,z) \\ \nabla_z G(u,v,w,z) \end{pmatrix} \\ &:= \begin{pmatrix} A\left(A^t u + v + w - c + \epsilon^i x^i\right) - \epsilon^i b \\ A^t u + 2v + w - c + \epsilon^i x^i + z + \epsilon^i y^i - \epsilon^i d \\ A^t u + v + w - c + \epsilon^i x^i - \gamma^i W^{-1}\mathbf{e} \\ v + z + \epsilon^i y^i - \eta^i Z^{-1}\mathbf{e} \end{pmatrix} = 0 \end{aligned}$$

where $W = diag(w_1, w_2, \ldots, w_n)$ and $Z = diag(z_1, z_2, \ldots, z_n)$. The linearization of the gradient function $\nabla G(u,v,w,z)$ around the point (u^i, v^i, w^i, z^i) gives us the following system of linear equations

$$\begin{pmatrix} AA^t & A & A & 0 \\ A^t & 2I & I & I \\ A^t & I & I + \gamma^i (W^i)^{-2} & 0 \\ 0 & I & 0 & I + \eta^i (Z^i)^{-2} \end{pmatrix} \begin{pmatrix} u \\ v \\ w \\ z \end{pmatrix} = \begin{pmatrix} Ac + \epsilon^i b - \epsilon^i A x^i \\ c + \epsilon^i d - \epsilon^i x^i - \epsilon^i y^i \\ c + 2\gamma^i (W^i)^{-1}\mathbf{e} - \epsilon^i x^i \\ 2\eta^i (Z^i)^{-1}\mathbf{e} - \epsilon^i y^i \end{pmatrix} \tag{15}$$

For ease of notation, define the diagonal matrices

$$\begin{aligned} D_1 &:= \left(I + \gamma^i (W^i)^{-2}\right)^{-1} \\ D_2 &:= \left(I + \eta^i (Z^i)^{-2}\right)^{-1} \\ and\ E &:= (2I - D_1 - D_2)^{-1} \end{aligned}$$

and vectors

$$\begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} := \begin{pmatrix} Ac + \epsilon^i b - \epsilon^i A x^i \\ c + \epsilon^i d - \epsilon^i x^i - \epsilon^i y^i \\ c + 2\gamma^i (W^i)^{-1}\mathbf{e} - \epsilon^i x^i \\ 2\eta^i (Z^i)^{-1}\mathbf{e} - \epsilon^i y^i \end{pmatrix}$$

The solution of the $(m + 3n)$ dimensional system (15) can then be obtained by first solving the m dimensional linear system

$$A(I - D_1)(I - E(I - D_1))A^t u =$$
$$h_1 - AD_1h_3 - A(I - D_1)E(h_2 - D_1h_3 - D_2h_4) \qquad (16)$$

for u. The other 3 unknowns can then be computed as follows:

$$\begin{aligned} v &= E\left(h_2 - D_1h_3 - D_2h_4 - (I - D_1)A^t u\right) \\ w &= -D_1\left(A^t u + v - h_3\right) \\ z &= -D_2(v - h_4) \end{aligned}$$

We have thus reduced the dimensionality of the problem from $m + 3n$ to m which is of the same size for an LP without upper bounds.

6 Performance of QIDPP for LP with upper bounded variables

As test problems with upper bounds for QIDPP, we have solved some of the Patient Distribution System (PDS) problems that were developed by the staff of the Military Airlift Command at the Scott Air Force Base. These problems are multicommodity network problems. We listed the sizes of these problems in Table 6.

MINOS and QIDPP were both run on an Astronautics ZS-1 [16]. The algorithm QIDPP was implemented in standard FORTRAN. The optimal value obtained by MINOS and the time MINOS took to get these values are listed in Table 7. MINOS was run with default parameter values, except for *log frequency 200, summary frequency 200* and *solution no*. Table 8 shows the objective values, number of iterations and CPU time for QIDPP. The time to transform the MPS file (the format required by MINOS) into sparse format is not included here. The QIDPP program was run with one set of parameter values for *all* problems. The Yale Sparse Matrix Package was used to solve the linear system (16). Variables with no upper bound are given dummy upper bounds. This dummy bound is set at 100000.

Table 9 shows the errors of the solution obtained by QIDPP. They are computed as follows:

1. Relative error in objective value

$$\frac{|cx^* - cx|}{|cx^*|}$$

cx^* is the optimal value obtained by MINOS

2. Absolute error in primal feasibility

$$\|Ax_+ - b, (-x)_+, (x_+ - d)_+\|_\infty$$

3. Absolute error in dual feasibility

$$\left\|(A^t u + v - c)_+, v_+\right\|_\infty$$

In Table 10 we give a comparison between MINOS times and QIDPP times. Finally, we listed the time from the KORBX machine as reported by W. J. Carolan et al [2] and QIDPP time in Table 11. The optimal objective function value for PDS-10 problem that we obtain is not the same as the one from the KORBX because the data that we have is similar but not identical.

For larger problem, a preconditioned conjugate gradient approach is currently being developed [15] as replacement for the YSMP package for solving the linear systems (16).

7 Conclusions

We have presented an algorithm for solving a linear program based on an interior point method applied to the dual of a proximal point formulation of the linear program. In the dual problem that we are solving, the only constraints present are nonnegativity constraints. These simple constraints allow us to start the algorithm without a phase 1 or feasibility phase which is necessary for primal interior algorithms. When an efficient routine such as the Yale Sparse Matrix Package is utilized to solve the linear system of equations that arises at each iteration of the algorithm, the method has been demonstrated to outperform a standard simplex linear programming package. We have also shown how a linear program with bounded variables can be solved without increasing the dimensionality of the problem. Patient

Distribution System models, which are large multicommodity network flow problems with some capacitated arcs, were used to test our algorithm. Our actual computing times were comparable to those of other variants of the interior method implemented on a much more powerful machine than ours.

No.	Problem Name	Rows	Columns	Non-Zeros
1	Afiro	28	32	88
2	Adlittle	57	97	465
3	ScSd1	78	760	3148
4	Share2b	97	79	730
5	Share1b	118	225	1182
6	Scagr7	130	140	553
7	ScSd6	148	1350	5666
8	Beaconfd	174	262	3476
9	Israel	175	142	2358
10	Truss1	200	1602	4984
11	Sc205	206	203	552
12	BrandY	221	249	2150
13	E226	224	282	2767
14	ScTap1	301	480	2052
15	BandM	306	472	2659
16	Rabo	317	560	5201
17	Scfxm1	331	457	2612
18	Scorpion	389	358	1708
19	ScSd8	398	2750	11334
20	Ship04s	403	1458	5810
21	Ship04l	403	2118	8450
22	Scagr25	472	500	2029
23	Scrs8	491	1169	4029
24	Truss2	500	4312	13584
25	Scfxm2	661	914	5229
26	Pilot.we	723	2789	9218
27	Ship08s	779	2387	9501
28	Ship08l	779	4283	17085
29	25fv47	822	1571	11127
30	CzProb	930	3523	14173
31	Scfxm3	991	1371	7846
32	Truss3	1000	8806	27836
33	ScTap2	1091	1880	8124
34	Ship12s	1152	2763	10941
35	Ship12l	1152	5427	21597
36	ScTap3	1481	2480	10734

Table 1: **Linear Programming Test Problems**

No.	Problem Name	Objective Value	Iterations	CPU Time
1	Afiro	-0.46475314 e+03	6	1.20
2	Adlittle	0.22549496 e+06	116	5.05
3	ScSd1	0.86666667 e+01	217	22.93
4	Share2b	-0.41573224 e+03	120	6.87
5	Share1b	-0.76589319 e+05	300	18.68
6	Scagr7	-0.23313898 e+07	66	6.23
7	ScSd6	0.50500000 e+02	483	62.45
8	Beaconfd	0.33592486 e+05	39	11.55
9	Israel	-0.89664482 e+06	265	24.27
10	Truss1	0.11436413 e+05	1301	189.78
11	Sc205	-0.52202061 e+02	114	11.57
12	BrandY	0.15185099 e+04	331	31.78
13	E226	-0.18751929 e+02	584	52.37
14	ScTap1	0.14122500 e+04	304	33.17
15	BandM	-0.15862802 e+03	422	51.93
16	Rabo	0.66510241 e+05	857	134.71
17	Scfxm1	0.18416759 e+05	428	45.82
18	Scorpion	0.18781248 e+04	169	24.18
19	ScSd8	0.90500000 e+03	1321	294.25
20	Ship04s	0.17987147 e+07	390	66.95
21	Ship04l	0.17933245 e+07	579	115.55
22	Scagr25	-0.14753433 e+08	532	71.77
23	Scrs8	0.90429695 e+03	577	96.48
24	Truss2	0.72752363 e+05	7673	2966.73
25	Scfxm2	0.36660262 e+05	1014	182.93
26	Pilot.we	-0.27200991 e+07	5118	1743.85
27	Ship08s	0.19200982 e+07	657	184.55
28	Ship08l	0.19090552 e+07	960	336.57
29	25fv47	0.55018459 e+04	7027	2416.82
30	CzProb	0.21851967 e+07	2547	966.55
31	Scfxm3	0.54901255 e+05	1467	385.35
32	Truss3	0.55933897 e+07	13391	8777.53
33	ScTap2	0.17248071 e+04	1569	434.77
34	Ship12s	0.14892361 e+07	759	353.00
35	Ship12l	0.14701879 e+07	1415	769.37
36	ScTap3	0.14240000 e+04	1571	585.07

Table 2: **MINOS 5.0 Results**

Pr. No.	Problem Name	Total Iterations	CPU Time	Primal Objective Value	Dual Objective Value
1	Afiro	30	1.06	-0.46475314 e+03	-0.46475314 e+03
2	Adlittle	26	2.50	0.22549496 e+06	0.22549495 e+06
3	ScSd1	28	11.60	0.86666671 e+01	0.86666666 e+01
4	Share2b	27	5.18	-0.41573230 e+03	-0.41573229 e+03
5	Share1b	63	14.22	-0.76589315 e+05	-0.76589319 e+05
6	Scagr7	35	4.32	-0.23313898 e+07	-0.23313898 e+07
7	ScSd6	29	21.17	0.50500001 e+02	0.50499999 e+02
8	Beaconfd	29	21.98	0.33592486 e+05	0.33592486 e+05
9	Israel	35	174.57	-0.89664482 e+06	-0.89664485 e+06
10	Truss1	26	47.93	0.11436413 e+05	0.11436413 e+05
11	Sc205	43	7.48	-0.52202061 e+02	-0.55202061 e+02
12	BrandY	33	22.70	0.15185099 e+04	0.15185099 e+04
13	E226	38	29.45	-0.18751928 e+02	0.18751929 e+02
14	ScTap1	29	14.06	0.14122500 e+04	0.14122499 e+04
15	BandM	35	30.70	-0.15862802 e+03	-0.15862802 e+03
16	Rabo	32	355.33	0.66510242 e+05	0.66510238 e+05
17	Scfxm1	37	32.71	0.18416759 e+05	0.18416759 e+05
18	Scorpion	28	12.03	0.18781248 e+04	0.18781248 e+04
19	ScSd8	27	43.31	0.90500001 e+03	0.90499998 e+03
20	Ship04s	26	21.68	0.17987148 e+07	0.17987147 e+07
21	Ship04l	26	33.53	0.17933246 e+07	0.17933245 e+07
22	Scagr25	42	19.49	-0.14753433 e+08	-0.14753433 e+08
23	Scrs8	46	44.14	0.90429695 e+03	0.90429694 e+03
24	Truss2	27	277.41	0.72752368 e+05	0.72752367 e+05
25	Scfxm2	38	71.52	0.36660261 e+05	0.36660261 e+05
26	Pilot.we	86	440.35	-0.27201075 e+07	-0.27201075 e+07
27	Ship08s	26	34.86	0.19200982 e+07	0.19200981 e+07
28	Ship08l	26	72.78	0.19090553 e+07	0.19090551 e+07
29	25fv47	39	509.50	0.55018469 e+04	0.55018469 e+05
30	CzProb	58	92.73	0.21851967 e+07	0.21851967 e+07
31	Scfxm3	39	114.78	0.54901254 e+05	0.54901254 e+05
32	Truss3	32	830.11	0.55933898 e+07	0.55933895 e+07
33	ScTap2	31	124.60	0.17248072 e+04	0.17248071 e+04
34	Ship12s	28	39.30	0.14892362 e+07	0.14892361 e+07
35	Ship12l	27	93.03	0.14701879 e+07	0.14701879 e+07
36	ScTap3	36	174.33	0.14240000 e+04	0.14240000 e+04

Table 3: **QIDPP:Quadratic Interior Dual Proximal Point Results**

Pr. No.	Problem Name	Primal Infeasibility	Dual Infeasibility	Complementarity
1	Afiro	0.2 e-11	0.1 e-06	0.1 e-07
2	Adlittle	0.3 e-09	0.2 e-09	0.2 e-09
3	ScSd1	0.3 e-08	0.2 e-08	0.4 e-09
4	Share2b	0.5 e-06	0.9 e-09	0.9 e-10
5	Share1b	0.2 e-04	0.0 e-00	0.1 e-12
6	Scagr7	0.4 e-11	0.3 e-06	0.5 e-09
7	ScSd6	0.1 e-07	0.0 e-00	0.2 e-09
8	Beaconfd	0.2 e-09	0.2 e-04	0.4 e-13
9	Israel	0.8 e-09	0.2 e-10	0.3 e-10
10	Truss1	0.2 e-08	0.6 e-06	0.1 e-09
11	Sc205	0.4 e-10	0.0 e-00	0.1 e-18
12	BrandY	0.1 e-08	0.1 e-02	0.5 e-10
13	E226	0.2 e-06	0.2 e-10	0.2 e-15
14	ScTap1	0.2 e-08	0.7 e-06	0.4 e-09
15	BandM	0.1 e-08	0.3 e-04	0.5 e-10
16	Rabo	0.2 e-10	0.2 e-05	0.3 e-09
17	Scfxm1	0.6 e-09	0.2 e-02	0.4 e-09
18	Scorpion	0.7 e-05	0.0 e-00	0.1 e-13
19	ScSd8	0.3 e-08	0.1 e-06	0.6 e-10
20	Ship04s	0.4 e-08	0.1 e-04	0.2 e-09
21	Ship04l	0.1 e-06	0.0 e-00	0.6 e-10
22	Scagr25	0.2 e-12	0.2 e-08	0.7 e-10
23	Scrs8	0.8 e-06	0.9 e-09	0.2 e-16
24	Truss2	0.3 e-07	0.2 e-04	0.4 e-09
25	Scfxm2	0.4 e-09	0.6 e-02	0.5 e-10
26	Pilot.we	0.1 e-10	0.6 e-03	0.1 e-14
27	Ship08s	0.4 e-09	0.0 e-00	0.3 e-09
28	Ship8l	0.2 e-06	0.0 e-00	0.3 e-10
29	25f47	0.2 e-08	0.3 e-04	0.1 e-07
30	CzProb	0.8 e-07	0.1 e-02	0.4 e-07
31	Scfxm3	0.7 e-10	0.3 e-04	0.1 e-11
32	Truss3	0.1 e-06	0.5 e-06	0.6 e-11
33	ScTap2	0.4 e-09	0.6 e-05	0.2 e-07
34	Ship12s	0.7 e-09	0.2 e-04	0.1 e-09
35	Ship12l	0.3 e-08	0.5 e-06	0.2 e-10
36	ScTap3	0.8 e-08	0.2 e-03	0.2 e-08

Table 4: **QIDPP:Quadratic Interior Dual Proximal Point Results**
(Continued)

Pr. No.	Problem Name	MINOS (seconds)	QIDPP (seconds)	Minos/QIDPP Time Ratio
1	Afiro	1.20	1.06	1.13
2	Adlittle	5.05	2.50	2.02
3	ScSd1	22.93	11.60	1.98
4	Share2b	6.87	5.18	1.33
5	Share1b	18.68	14.22	1.31
6	Scagr7	6.23	4.32	1.44
7	ScSd6	62.45	21.17	2.95
8	Beaconfd	11.55	21.98	0.53
9	Israel	24.27	174.57	0.14
10	Truss1	189.78	47.93	3.96
11	Sc205	11.57	7.48	1.55
12	BrandY	31.78	22.70	1.40
13	E226	52.37	29.45	1.78
14	ScTap1	33.17	14.06	2.36
15	BandM	51.93	30.70	1.69
16	Rabo	134.71	355.53	0.38
17	Scfxm1	45.82	32.71	1.40
18	Scorpion	24.18	12.03	2.01
19	ScSd8	294.25	43.31	6.79
20	Ship04s	66.95	21.68	3.09
21	Ship04l	115.55	33.53	3.45
22	Scagr25	71.77	19.49	3.68
23	Scrs8	96.48	44.14	2.19
24	Truss2	2966.73	277.41	10.69
25	Scfxm2	182.93	71.52	2.56
26	Pilot.we	1743.85	440.35	3.96
27	Ship08s	184.55	34.86	5.29
28	Ship08l	336.57	72.78	4.62
29	25fv47	2416.82	509.50	4.74
30	CzProb	966.55	92.73	10.42
31	Scfxm3	385.35	114.78	3.36
32	Truss3	8777.53	830.11	10.57
33	ScTap2	434.77	124.60	3.49
34	Ship12s	353.00	39.30	8.98
35	Ship12l	769.37	93.03	8.27
36	ScTap3	585.07	174.33	3.36
-	**TOTAL**	21482.63	3846.44	5.59

Table 5: **Comparison between Minos and QIDPP (Microvax 3200)**

Days	Rows	Columns	Up. Bound	Non-Zeros
1	1473	3816	605	8139
2	2953	7716	2134	16571
3	4593	12590	3839	27099
4	6372	18615	5555	39944
5	8099	24192	7370	51978
6	9881	29351	9240	63220
10	16558	49932	16148	107605

Table 6: **Patient Distribution System (PDS) Problem Data**

Days	Objective Value	Ph. I It.	Total It.	CPU Hr.
1	2.9083930523 e+10	420	965	0.033
2	2.8857862010 e+10	1715	3003	0.227
3	2.8597374145 e+10	7551	10591	1.818
4	2.8341928581 e+10	10552	15750	5.076
5	2.8054052607 e+10	14026	22515	10.282
6	2.7761037639 e+10	18984	29952	18.521
10†	na	na	na	na

Table 7: **PDS : MINOS Results**

† Not attempted

Days	Primal Objective	Dual Objective	Iter.	CPU Hr.
1	2.9083930524 e+10	2.9083930521 e+10	47	0.008
2	2.8857862009 e+10	2.8857862007 e+10	49	0.029
3	2.8597374136 e+10	2.8597374143 e+10	47	0.078
4	2.8341928579 e+10	2.8341928580 e+10	57	0.210
5	2.8054052595 e+10	2.8054052606 e+10	56	0.590
6	2.7761037575 e+10	2.7761037602 e+10	62	0.865
10	2.6727095005 e+10	2.6727094976 e+10	82	4.682

Table 8: **PDS : QIDPP Results**

No.	Objective Value		Absolute Infeasiblity	
Days	PRIMAL	DUAL	PRIMAL	DUAL
1	3 e-11	7 e-11	2 e-04	3 e-04
2	4 e-11	1 e-10	7 e-04	2 e-04
3	3 e-10	7 e-11	4 e-04	3 e-04
4	7 e-11	4 e-11	3 e-04	5 e-04
5	4 e-10	4 e-11	8 e-04	4 e-04
6	2 e-09	1 e-09	1 e-03	4 e-04
10	na	na	8 e-04	2 e-04

Table 9: **PDS : QIDPP Errors**

No. Days	MINOS		QIDPP		Speed-Up
	Iter.	**CPU Hr.**	**Iter.**	**CPU Hr.**	
1	965	0.033	47	0.008	4.1
2	1715	0.227	49	0.029	7.8
3	10591	1.818	47	0.078	23.3
4	15750	5.076	57	0.210	24.2
5	22515	10.282	56	0.590	17.4
6	29952	18.521	51	0.865	21.4
10	na	na	82	4.682	na

Table 10: **PDS : Comparison between MINOS and QIDPP (Astronautics ZS-1)**

No. of Days	**2**	**6**	**10**
KORBX SYSTEM			
Primal algorithm			
Iterations	33	57	66
Time	105.1 sec	38.6 min	3.3 hrs
Obj. Value	2.8858 e+10	2.7761 e+10	2.5959 e+10
Dual algorithm			
Iterations	46	63	66
Time	124.6 sec	42.0 min	3.4 hrs
Obj. Value	2.8858 e+10	2.7761 e+10	2.5959 e+10
Primal-Dual algorithm			
Iterations	38	57	85
Time	135.4 sec	43.7 min	4.5 hrs
Obj. Value	2.8858 e+10	2.7761 e+10	2.5959 e+10
Power Series algorithm			
Iterations	21	24	51
Time	148.4 sec	28.2 min	3.3 hrs
Obj. Value	2.8858 e+10	2.7761 e+10	2.5959 e+10
ASTRONAUTICS			
QIDPP algorithm			
Iterations	49	62	82
Time	103.4 sec	51.9 min	4.7 hrs
Obj. Value	2.8858 e+10	2.7761 e+10	2.6727 e+10

Table 11: **PDS : Comparison of the Interior Algorithms on the KORBX and Astronautics**

8 References

[1] D.P. Bertsekas. (1982). Constrained optimization and Lagrange multiplier methods, *Academic Press*, New York.

[2] W.J. Carolan, J.E. Hill, J.L. Kennington, S. Niemi and S.J. Wichman. (1989). An Empirical Evaluation of the KORBX algorithms for the military airlift applications, Technical Report 89-OR-06, Dept. of Computer Science and Engineering, Southern Methodist University, TX.

[3] S.C. Eisenstat, M.C. Gursky, M.H. Schultz and A.H. Sherman. (1982). Yale Sparse Matrix Package I: The symmetric codes. *International Journal for Numerical Methods in Engineering.* Vol. 18, pp. 1145-1151.

[4] S.C. Eisenstat, M.C. Gursky, M.H. Schultz and A.H. Sherman. (1977). Yale Sparse Matrix Package I: The symmetric codes. Research Report # 112, Yale University, CT.

[5] D.M. Gay. (1985). Electronic mail distribution of linear programming test problems, *Mathematical Programming Society COAL Newsletter*, December.

[6] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin and M.H. Wright. (1986). On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method, *Mathematical Programming 36*, pp. 183-209.

[7] N. Karmarkar. (1984). A new polynomial-time algorithm for linear programming, *Combinatorica 4*, pp. 373-395.

[8] I.J. Lustig. (1988). A generic primal-dual interior point algorithm, Technical Report SOR 88-3, Department of Civil Engineering and Operations Research, Princeton University, NJ.

[9] O.L. Mangasarian. (1969). Nonlinear Programming, *Mc Graw-Hill*, New York.

[10] K.A. McShane, C.L. Monma and D. Shanno. (1988). An implementation of a primal-dual interior point method for linear programming, Technical

Report, RUTCOR, Rutgers University, NJ.

[11] C.L. Monma and A.J. Morton. (1987). Computational experience with a dual affine variant of Karmarkar's method for linear programming, *Operations Research Letters 6*, pp. 261-267

[12] B.A. Murtagh and M.A. Saunders. (1983). MINOS 5.0 user's guide, Technical Report SOL 83-20, Stanford Optimization Laboratory, Stanford, CA, 1983.

[13] R.T. Rockafellar. (1973). A dual approach to solving nonlinear programming problems by unconstrained optimization, *Mathematical Programming 5*, pp. 354-373.

[14] R.T. Rockafellar. (1976). Monotone operators and the proximal point algorithm, *SIAM Journal on Control and Optimization 14*, pp. 877-898.

[15] R. Setiono. (1990). Interior algorithms for mathematical programming. PhD Thesis (in preparation), Computer Sciences Department, University of Wisconsin-Madison, WI.

[16] J.E. Smith. (1989). Dynamic Instruction Scheduling and the Astronautics ZS-1, *Computer* ,July, pp. 21-35.