TR/03/93 May, 1993

Experimental Investigations in Combining Primal Dual Interior Point Method and Simplex Based LP Solvers

Ron Levkovitz*, Gautam Mitra*,
and Mehrdad Tamiz**

* Brunel University, Uxbridge,Middx,UK
** University of Portsmouth, UK

CONTENTS

Experimental Investigations in Combining Primal Dual Interior Point Method and Simplex Based LP Solvers

Ron Levkovitz

Gautam Mitra

Brunei, the university of west London, UK

Mehrdad Tamiz

University of Portsmouth UK

Abstract: The use of a primal dual interior point method (PD) based optimizer as a robust linear programming (LP) solver is now well established. Instead of replacing the sparse simplex algorithm (SSX), the PD is increasingly seen as complementing it. The progress of PD iterations is not hindered by the degeneracy or the stalling problem of the SSX, indeed it reaches the 'near optimum' solution very quickly. The SSX algorithm, in contrast, is not affected by the boundary conditions which slow down the convergence of the PD. If the solution to the LP problem is non unique, the PD algorithm converges to an interior point of the solution set while the SSX algorithm finds an extreme point solution. To take advantage of the attractive properties of both the PD and the SSX, we have designed a hybrid framework whereby cross over from PD to SSX can take place at any stage of the PD optimization run. The cross over to SSX involves the partition of the PD solution set to active and dormant variables. In this paper we examine the practical difficulties in partitioning the solution set, we discuss the reliability of predicting the solution set partition before optimality is reached and report the results of combining exact and inexact prediction with SSX basis recovery.

Keywords: primal dual interior point methods, simplex algorithm, basis recovery

1. Introduction

The use of the primal dual interior point method (PD) for the solution of linear programs provides a number of benefits which are summarized below. For large or highly degenerate LPs PD is in many cases faster than the SSX solver. Whereas SSX based algorithms require considerable adaptation and control parameter tuning from one model class to another, default settings of PD (predictor corrector and barrier) are sufficient to process a wide class of LPs. PD is not only robust in this way, its progress is not hindered by the degeneracy or the stalling problem of the SSX; indeed it reaches the "near optimal" solution very quickly. SSX algorithms, in contrast, are not affected by the boundary conditions which slow down the convergence of PD.

There are three well known and well exercised extensions of traditional LP namely, successive linear programming, integer programming, and post-optimal analysis. In all these cases optimum solutions for a family of problems need to be computed which in turn involves reoptimization from the last computed primal and dual optimum basis (extreme point solution). The extreme point solution is the corner stone of SSX algorithms and the use of the corresponding basis as starting point is naturally applied in this context to solve efficiently a family of similar problems. PD algorithms, on the other hand, usually converge to a point in the interior of the optimal face. This property and the behaviour of the PD algorithm near the boundary (Megiddo 89) make it difficult to apply PD in the same way for a family of similar LP's. Although some extension to LP and in particular sensitivity and post optimality analysis were developed for PD (Adler 89) (Guler 92), an extreme point solution is still preferable for many applications which require warm start. For these applications, the fast initial convergence of PD to a near optimal solution can be followed up by the superior near optimal to optimal convergence of SSX algorithms. This strategy combines the fast initial convergence of the PD with the basis representation of the SSX. Many researchers (see for example (Megiddo 88,91), (Bixby 92), (Mitra 88)) consider this latter approach to be a promising computational scenario. This hybrid approach, however, requires a substantial performance superiority of PD and an efficient PD-SSX integration to make it worthwhile.

The PD optimal solution can be used either for PD post-optimality analysis or to cross over to SSX. In both cases, a clear partition between the variables that are dormant at the PD optimal solution (variables that are either at their upper or lower bound) and those which are

active at the PD optimal solution is needed. In general, if the active and dormant sets are known an optimal solution can be easily retrieved.

The research issues reported in this paper are mainly concerned with efficiently finding such a partition, either from a near optimal point or from an intermediate solution, and with using this partition to warm start SSX via basis recovery. In section 2 of this paper we give some theoretical background and establish the termination criteria for which a partition can be determined, in section 3 we describe and review methods for predicting the partition from an intermediate solution, develop a heuristic for finding such a partition and asses the quality of the results. In section 4 we present a basis recovery technique that was designed to take advantage of exact and near exact partitions. Our conclusions are summarized in section 5.

2. Termination criteria and complementarity in the PD solution

Consider the following primal and dual LP problems:

(*Primal*) *Min $c^T x$ S.T. $Ax=b$, $x \geq 0$*

(*Dual*)   *Max* $b^T y$ *S.T.* $A^T y+z=c,$ $z \geq 0$

        (2.1)

            $A \in R^{m \times n}, x,z,c \in R^n, y,b \in R^m$

(Upper bounds are omitted for simplicity)

A PD algorithm applied to these problems generates a sequence of strictly interior primal (dual) intermediate solution points. This sequence theoretically converges to an optimal primal (dual) solution which is a boundary point of the primal (dual) feasible polyhedron. The actual termination of the algorithm, however, is not on the boundary of the polyhedron but in the interior close to an optimal solution.

At the optimal solution, complementarity is enforced, mat is :

$c^T x-b^T y=x^T z=0$ and $XZe=0$

                                                                                                                        (2

.2)

Where X, Z are diagonal matrices whose diagonals are the vectors x and z respectively and $e=(1_1,....1_n)^T$.

We define the *indicator* sets $\sigma(v)$, $\overline{\sigma}(v)$ of indices of *active* (positive) and *dormant* (non positive) variables of a non negative vector v as:

$$v \in R^n, v \geq 0, \sigma(v) = \{j, v_j > 0\}, \overline{\sigma}(v) = \{1,...n\} - \sigma(v)$$

(2.3)

Of all the primal optimal solutions and the dual optimal solutions of (2.1) there exists at least one solution pair (x^*), (y^*z^*) where the strict complementarity of (2.2) applies, or in terms of the indicator set σ defined in (2.3):

$$\sigma(x) \cap \sigma(z^*) = \phi \ and \ \sigma(x^*) \cup \sigma(z^*) = \{1,...n\}$$

(2.4)

We call this solution a strictly complementary solution.

Guler and Ye (91) show that a class of interior point methods which also contains the predictor corrector and barrier PD algorithms (Mehrotra 90) generates a sequence of feasible pairs (x^k, z^k) such that:

$$\frac{\min(X^k Z^k e)}{(x^k)^k z^k} > \Omega(\frac{1}{n})$$

(2.5)

(where $\Omega(\frac{1}{n})$ is the order of at least $\frac{1}{n}$).

For this class of algorithms, they also proved the following theorem:

Theorem 2.1: At iteration k, let $\sigma^k = \{j : x_j^k \geq z_j^k\}$ and assume the LP data is rational. If L is the input length of the problem then for all algorithms that satisfy (2.5) if $(x^k)^T z^k \leq 2^{-3L}$,

$\sigma^k = \sigma(x)$.

Theorem 2.1 shows that when an the intermediate solution point is close enough to the optimal solution, the set $\sigma(x^*)$ of active primal variable indices can be identified. The dormant variables can be set to zero and a smaller problem can be solved to retrieve the 'exact' optimal solution on a boundary of the polyhedron.

Although theorem 2.1 gives a theoretical stopping criterion for the PD , in practice, a more realistic criterion is needed. In most PD algorithms, the termination criteria are based on the feasibility and the duality gap. The algorithm terminates successfully if the following criteria are satisfied:

$$\frac{c^T x - b^T y}{1 + \left\| b^T y \right\|_1} < e^1$$

$$\left\| b - Ax \right\|_1 < e^2$$

(2

.6)

$$\left\| c - A^T y - z \right\|_1 < e^3$$

For convenience and simplicity, $\epsilon^1, \epsilon^2, \epsilon^3$, are set to the same value, that is: $\epsilon^1 = \epsilon^2 = \epsilon^3 = \epsilon$. Different implementations of PD algorithms use different values for this ϵ. This makes the comparison between algorithms performance difficult and can result in non-complementary solution and wrong partition of the solution set. The following table summarize the value of some reported e values.

<< Table 2.1 should come here >>

It is important to find a proper termination value since a very small ϵ maybe difficult to reach due to numerical software and hardware difficulties and a too large ϵ value, can result in a wrong partition of the active and dormant variable sets. In order to find the proper termination e in PD we introduce the following three definitions:

Let $\overline{\sigma}^*(x), \overline{\sigma}^*(z)$ define the set of all the dormant primal and dual variables in the PD optimal solution then

Definition 2.1:

We say that the optimal solution set as found by the PD can be *determined* if by applying the following criteria we are able to retrieve $\overline{\sigma}^*(x), \overline{\sigma}^*(z)$:

$$\overline{\sigma}^*(x) = \left\{ j \mid x_j < \widetilde{\epsilon}, j = 1,\,..,n \right\}$$
$$\overline{\sigma}^*(z) = \left\{ j \mid z_j < \widetilde{\epsilon}, j = 1,\,..,n \right\}$$

That is, for every variable j, either z_j, x_j or both are clearly converging to 0 as determined by a small tolerance parameter $\overline{\epsilon} > 0$.

5

Definition 2.2:

Following theorem 2.1 we say that the optimal solution set can be *recognized* if we can retrieve $\overline{\sigma}^*(x), \overline{\sigma}^*(z)$ by using the following criteria:

$$\overline{\sigma}^*(x) = \left\{ j \mid x_j < \overline{\in} \ or \ (x_1 > \overline{\in} \ and \ z_j > x_j) \right\}$$

$$\overline{\sigma}^*(z) = \left\{ j \mid z_j < \overline{\in} \ or \ (z_1 > \overline{\in} \ and \ x_j > z_j) \right\}$$

Definition 2.3:

If $\overline{\sigma}^*(x), \overline{\sigma}^*(z)$ can be found by applying some approximation functions we say that the solution set can be *predicted*. We call these functions *predicting function.*

It is clear from Theorem 2.1 that $\overline{\sigma}^*(x), \overline{\sigma}^*(z)$ can be determined or recognized if we are in a close enough neighbourhood of the optimal solution. Below, we try to establish the termination criteria $\in$ for which definition 2.1 and 2.2 apply.

Practical termination Criteria

We extend our PD predictor corrector solver (Levkovitz 92) to establish a practical termination criterion value that gives a proper solution partition for a range of real life LP problem (see also Mehrotra (90) for a general description of high order primal dual interior point methods). The solver produces an intermediate solution at every PD iteration starting from the iteration for which the relative duality gap is reduced below 0.1. The solution set is partitioned using the recognizable set criteria (definition 2.2) and compared to the partition achieved for $\in = 10^{-16}$. In addition, we try to find the relative duality gap for which the solution set can be fully determined (definition 2.1). For both cases we set $\widetilde{\varepsilon} = 10^{-10}$. In the following table, every primal variable that is found to be dormant by the recognizing criteria is considered as hit. If a variable is found to be dormant by the recognizing criteria but is active in the optimal solution set, it is considered as a miss. This procedure is applied to some NETLIB problems with and without scaling. Typical results are given in tables 2.2 and 2.3.

The first column of the tables gives the value of the relative duality gap, the second column

the number of true dormant primal variables and the third column the number of wrongly recognized primal dormant variables.

<< Table 2.2 should come here >>

<< Table 2.3 should come here >>

From the above tables it is easy to see that while a solution set can be determined in both problems for $\mathcal{E} < 10^{-11}$, in the problem ORNA3 (as in many other problems) it can not be recognized earlier. The reason for such a behaviour is demonstrated in two typical patterns of behaviour as presented in Figures 2.1 and 2.2. In figure 2.1, only when the relative duality gap is reduced below 10^{-11} it is clear that the primal variable is active and the dual variable is dormant. In figure 2.2, both primal and dual variables seem to converge to zero but while the dual variable actually converges to zero, the primal variable stabilizes on low but positive value. These figures demonstrate that although the duality gap is almost monotonically reduced, the value of a single variable can vary considerably from one iteration to another. The behaviour observed in Figure 2.1 can be attributed to the large step size taken in the PD algorithm (0.9995 in our case, see Levkovitz (92) or Lustig et al. (90) for further details). By limiting the step size, the convergence of the variables to their final value can become more orderly. Any gains, however, are unlikely since the number of iterations also grows.

<< Figure 2.1 should come here >>

7

<< Figure 2.2 should come here >>

3. Predicting the partition of the optimal primal and dual solution sets

From results of the previous section we conclude that if a proper partition of the solution set is to be made, the relative duality gap should be at least $10^{-12} - 10^{-14}$ before $\overline{\sigma}^*(x), \overline{\sigma}^*(z)$ can be confidently determined. Also, it is clear that applying the recognition criterion before this duality gap is reached can be misleading. However, it is still desirable to terminate PD before reaching a relative duality gap of $\mathcal{C}<10^{-12}$ or alternatively to identify at least some of the active and dormant variables, some of the reasons for this are listed below:

(i) at an early stage of the PD search procedure, namely 50% of the total number of iterations, PD finds a near optimum solution which is often within 80-90% of the final solution.

(ii) the numerical stability of the algorithm deteriorates when the algorithm gets close to the boundary, the computation of the trajectory in the kth iteration of PD includes the computation of the diagonal matrix $D^k = X^k(Z^k)^{-1}$ and the solution for λ of the symmetric system of equations $AD^KA^T\lambda=\rho$. The diagonal matrix can have some very high value entries for active variables and some near zero entries for dormant variables. Although equation 2.5 limits the possible variation, in practice the matrix AD^KA^T can still become ill conditioned and cause numerical errors.

(iii) As dormant or active variables are identified, the size of the problem can be reduced by removing them. In particular, the removal of dormant variables maintains primal feasibility and reduces the computation work of every iteration. Also, the removal of primal variables that converge to zero can partially resolve the numerical difficulties.

As a result, there is considerable interest in establishing if reliably predicted solution sets can be obtained earlier. Karmarkar, Gay (89), Mehrotra (91), Zhang et al. (91), El-Barki et al. (91), Levkovitz (91) and others have put forward heuristics based on indicator functions which are calculated at every iteration. These heuristics attempt to predict the partition to active and dormant variable sets in the optimal solution. Broadly, we can

classify these indicator functions to three groups: indicators that are based on either the primal or the dual variables, indicators that are based on both primal and dual variables and indicators that also take into account the changes in the values of variables from one iteration of the PD algorithm to another.

In their comprehensive survey of indicator functions El-Barki et al. (91) define the properties of the ideal indicator functions:

(i) Sharp separation: the indicator functions should clearly determine which of the primal variables is dormant and which is active.

(ii) Uniform separation: the indicators of all the dormant variables should converge to the same value. Similarly, the indicators of all active variables should converge to the same value.

(iii) Inexpensive to compute: the indicators are typically calculated after every PD iteration, therefore, the calculation of the indicator value must be of lower complexity than the complexity of the PD iteration.

(iv) Quick convergence: the indicator function value converges to its limit faster then the LP intermediate solution converges to the optimal one.

(v) Reliable

(vi) Scale independent

It is easy to see that if the above requirements (and in particular requirement (iii) and (iv) together) are satisfied then there is a lower complexity algorithm hidden inside the PD, however , no researcher to-date has presented such an algorithm.

From the results published by various researchers (see for example El-Barki et al. (91) and Gay (89)) it seems that although the complexity of calculating most indicator functions is lower then the PD complexity by orders of magnitudes (typically $O(n)$ or $O(mn)$ compared to $0(n^{2.5})$ of the PD iteration), the number of iterations that is saved by using predictions is marginal. This behaviour can be attributed to the difficulty of finding a general parameter setting that suits large number of problems. Further, if the indicator functions give a wrong prediction (as often happens), the resulted active set may not include an optimal solution set and lead to extra computational work. El-Barki, Tapia and Zhang (91) find mat the most successful indicators are those based on the primal and dual variables. In particular, they use   a combination of the following indicator functions:
(i) primal dual indicators for the j'th variable in the k+1 iterate

$$\frac{x_j^{k+1}}{z_j^{k+1}}$$

The indicator goes to infinity if the variable is active in the solution, otherwise, it goes to 0.

(ii) The sum of the primal and dual Tapia indicators for the j'th variable in the k+1 iterate

$$\frac{x_j^{k+1}}{x_j^k} + (1 - \frac{z_j^{k+1}}{z_j^k})$$

The indicator goes to 2 if the variable is active in the solution, otherwise it goes to 0. For our investigation we use the following primal and dual indicator functions:

(i) Primal dual indicators for the j'th variable in the k+1 iterate

$$Q_j^{k+1} = \frac{x_j^{k+1}}{z_j^{k+1}}$$

(ii) Iteration dependent primal dual indicator for the j'th variable in the k+1 iterate:

$$\delta_j^{k+1} = \frac{x_j^{k+1} \mid x_j^k}{z_j^{k+1} \mid z_j^k}$$

The first set of indicators gives the ratio of the primal and dual variables in the current iterate, it is clear that when the solution is close to optimal then:

$$\begin{cases} Q_j^k \underset{k\to\infty}{\to} \infty \; j \in \sigma^*(x) \\ Q_j^k \underset{k\to\infty}{\to} 0 \; j \in \sigma^*(z) \end{cases}$$

The second set of indicators, in contrast, measures the changes in the primal and dual values from iteration to iteration. The following figures demonstrate the different behaviour of the three indicators on four variables of the NETLIB problem Stair:

< < Figure 3.1 should come here > >

10

<< Figure 3.2 should come here >>

<< Figure 3.3 should come here >>

As can be seen from the above figures, the behaviour of the indicator functions assumes a meaningful pattern only after the relative duality gap and infeasibility are below a certain tolerance. Therefore, in our experiments, we calculate the values of the indicators only when the relative duality gap is reduced below 0.1. Our prediction is based on the observation that in the majority of the variables, if the reduction in the δ indicator for a particular variable j is strong enough and happens in no less than two consecutive iterations then the variable is going to be dormant in the optimal solution. In addition, the current value of the primal dual indicator is also taken into consideration.

These are utilized by incorporating the following procedure into our PD code:

Let S(1..n) be an array of integer variables, for all) set S(j)=0,

Let PI and P2 be the criteria parameters:

If ϵ<0.1 then

In the k+I iteration do if (Q_j^{k+1} < P2 and δ_j^{k+1} < $\dfrac{1}{P1}$) than $S(j) = -1 + S(j)$

Endif

It is easy to see that the computational and space complexity of calculating the indicator function in any iteration is in O(n), this incurs a relatively small overhead on the computation. If we decide to terminate and predict the solution set we use the following procedure:

Let P3 be the dormant threshold parameter

for j=1,n do

if S_j^k < -P3 then $\overline{\sigma^*}(x) = \overline{\sigma^*}(x) \cup \{j\}$

enddo

In the subsequent experiments, the value of P3 is set to 2, the values of PI and P2 are calibrated on certain NETLIB problems.

For a particular problem, by setting PI and P2 to appropriate values, a reliable prediction

can be guaranteed. Values that are suitable for one problem, however, can result in wrong prediction in another model. A very conservative setting, on the other hand, can guarantee that no active variables will be chosen as 0 but is likely to reduce the advantage of prediction to marginal at best.

Tables 3.1-3.3 demonstrate this difficulty, in table 3.1 the parameter setting is sufficient to find most dormant variables. Until the last iteration, however, a small number of active variables are also identified as dormant. In tables 3.2 and 3.3, two sets of parameters are used, the first set was calibrated to the problem GANGES and the second set to the problem MAROS. It is clear that parameters that give a reliable prediction for the first problem give an unreliable prediction for the second one and vice versa. Since this happens in many other problems we could not find a parameter settings that give consistently reliable predictions. Our results indicate that the use of indicators to terminate PD or reduce the size of the model is risky. On many models, a meaningful reduction can be done only when the duality gap is fairly small and the corresponding computational savings are minimal.

<< Table 3.1 should come here >>

<< Table 3.2 should come here >>

<< Table 3.3 should come here >>

4. Cross over to SSX and basis recovery

In non degenerate LP problems, the optimal solution point set is restricted to a single extreme point of both primal and dual feasible polyhedrons. In most practical cases,

however, the LP problem is primal or dual degenerate (or both) and the optimal solution point set describes a face of the feasible polyhedron. It is well known that the optimal solution generated by the PD algorithm converges to an interior point of this face (Megiddo 89). In many cases, however, an optimal extreme point solution is required. This optimal extreme point solution and the corresponding primal and dual optimal basis provide a powerful representation which arises in LP and its duality theory. Consider the LP problem of (2.1) and assume that the constraint matrix A is of full rank. For our purposes it is sufficient to state that the optimal basis of this LP problem is a submatrix B, $B \in R^{m \times m}$ such that $(x^*)^T = [(B^{-1}b)^T, 0], x^* \in R^n$ is an optimal solution of the primal problem and $y^* = (B^T)^{-1}c_B$ is the optimal solution for the dual problem. A primal (non degenerate) basic solution requires exactly m primal variables to be active and their corresponding dual slack variables to be 0. Compared to optimal SSX solution for a given LP, the PD solution has more active variables if the problem is dual degenerate or less if it is primal degenerate. In the integration of PD and SSX a basis recovery procedure has to be constructed which applies to both these cases. Megiddo (91), Bixby et al. (91), and Bixby and Saltzman (92) among others put forward methods for retrieving the basis from an interior point solution. In particular, Megiddo's methods is based on the existence of both primal and dual optimal and complementarity solutions. The algorithm uses these solutions to retrieve a basic solution (in a strongly polynomial time) while maintaining the complementarity and optimality of the intermediate solutions. Other basis recovery algorithms use either the primal or the dual solutions to retrieve the basis solution.

Usually, the LP problem in its general form has a combination of equality and inequality constraints. To suit the requirements of the PD algorithm, the original problem is converted to a standard LP problem with only equality constraints.

We now describe a method that we have developed for the crossover procedure. This procedure was especially developed to utilize the prediction of the optimal solution partition.

Given the original LP problem P we create a related problem $\overline{P}$ in the following way:

We define a subproblem that corresponds to the columns and rows of the original matrix which are active in the PD optimal solution. The column index sets C_S, C_N correspond to the variables of the original matrix which are active and dormant in the solution. R_E R_F row index sets such that corresponding logical variables take zero or positive values

13

respectively.

We construct the related problem $\overline{P}$ as shown below by fixing variables in the set C_N to their respective (lower) bounds and making the rows in the set R_F free rows. We note that if the PD prediction is correct then the appropriate subset of the given optimum solution to P $\hat{x}$ is a feasible solution to $\overline{P}$.

$<<$ Figure 4.1 should come here $>>$

Let $Min \ \hat{c}^T\hat{x}, S.T. \hat{A}\hat{x} \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} = b, \hat{x} \geq 0, \hat{A} \in R^{m \times \hat{n}}, \hat{x}, \hat{c} \in R^{\hat{n}}, b \in R^m$ be the original LP problem P.

This problem is translated to the standard form primal LP problem of (2.1) by the addition of up to m new variables that correspond to the inequality constraints.

Let $l = |I| : I = \left\{ i | \hat{A}_{j*}^T x \leq b_i \ or \ \hat{A}_{j*}^T x \geq b_i \right\}$ (where $\hat{A}_{j*}$ is the i'th row of the matrix $\hat{A}$) then l denotes the number of logicals (slacks and surplus variables) introduced whereby $n = l + \hat{n}$. Let $1 \leq i_q \leq l$ be the index of the logical variable for the row index $i \in I$ (that is, the row $\hat{A}_i$ is the i_q th non equality row in the matrix $\hat{A}$), then the basis recovery procedure is defined as follows:

1. Store the PD predicted solution for the standard form LP problems. If the solution is optimal the value of all the dormant variables is set to zero. Similarly, if the solution is intermediate, the value all the variables that are predicted to be dormant is set to zero.

2. Build restricted problem in the following way

for all structural variables $x_p j = 1,.., \hat{n}$

  if(x_j>0 or (x_j=0 and z_j=0))then

   set x_j as a candidate for the basis

  else

   do not include x_j it in the restricted problem)

   endif

endfor

14

for all the constraints r_p, i=1,...,m
  if ($i \in I$ and ,$x_{i+\hat{n}} > 0$) then
    the constraint is non-binding, make row r_i a free row
    in the restricted problem
  else
      make row r_i an equality row.
  endif
endfor

3. build the initial basis: take an initial unit basis and introduce variables from the candidate list according to their distance from the (lower) bound. A starting basis is obtained when

(i) the candidate list is exhausted
or
(ii) no more variables can be introduced.

4. Solve the restricted problem using the primal simplex algorithm and save the basis.

5. Restore the original problem and use the saved basis from step 4
    if the solution is optimal STOP.

6. if the solution is primal feasible then
    use the primal simplex algorithm to obtain a solution and STOP

7. if the solution is not primal feasible then
    use the dual simplex algorithm to obtain primal feasibility goto step 6.

If the prediction of the optimal basis is slightly wrong (this can happen if PD is terminated before an optimal solution is reached) then step 2 is terminated with a 'no feasible solution' status. The resulting basis, however, is usually near optimal and thus requires a low number of iterations in step 3 to reach optimality (see tables 4.1-4.5). The basis recovery algorithm is applied either from an optimal or an intermediate PD solution. If the solution is optimal, the solution set is correct and thus the restricted problem has a feasible and an optimal solution. Otherwise, the restricted problem can be infeasible, unbounded or may not include the proper optimal solution.

In table 4.1 we present the results of using the optimal PD solution in the basis recovery procedure. The basis recovery procedure is implemented within our general LP solver FORTLP. In the first four columns of this table we present the problem name and its statistics, in the fifth and sixth columns we give the number of PD and basis recovery iterations respectively. For comparison, the last column gives the iteration count of CPLEX

for these problems (Bixby 90) (this comparison can be slightly misleading since CPLEX uses pricing strategies that are not implemented in FORTLP).

In tables 4.2 and 4.3 we use the prediction criteria together with basis recovery in two representative problems. The headers of the tables give the problem name and parameter settings for the prediction heuristic. The first column of the tables give the relative duality gap, the second and the third columns the number of prediction hits and misses respectively and the last column the total number of basis recovery iterations. From the tables it can be seen that after a sufficient number of dormant variables is identified, the number of basis recovery iterations is greatly reduced, from that point to optimality, the reduction is the number of iteration is gradual and in general does not justify the extra number of PD iterations. It is interesting to notice that the existence of a small number of wrongly predicted dormant variables has little influence on the number of crossover iterations.

< < Table 4.1 should come here> >

< < Table 4.2 should come here > >

< < Table 4.3 should come here > >

Finally, we investigate utilizing the prediction for a family of problems. Tables 4.4-4.5 contain the corresponding results. The prediction parameters calibrated for the NETLIB staircase problem SCFXM1 are then applied to other problems in this family. In Table 4.4 the problem names and statistics are given, the prediction hits and misses and the number of basis recovery iterations from intermediate solutions are summarized in table 4.5. It is clear that for this family of problems, the parameter settings is sufficient to produce a near exact partition for all problems when the duality gap is reduced below 10^{-5}. Thereafter, the number of basis recovery iterations is not greatly reduced.

< < table 4.4 should come here > >

< < table 4.5 should come here > >

5. Conclusions

In the report we investigate the PD to SSX crossover as well as the scope of utilizing an exact and near-exact partition of the optimal solution set. We have highlighted the fact that if the problem is degenerate, the PD and SSX optimal solution partitions are different. It is known that both partitions have their own merits and give alternative information. The SSX partition describes the strictly complementary exterior optimal solution point while
the PD partition describes the optimal face. We have showed that if the optimal PD partition is required, a relatively high precision solution is needed. We have demonstrated that if the   SSX optimal solution or partition are needed, the optimal PD solution together with the optimal PD partition can be effectively utilized in a crossover procedure. Our results show   that although this may not always be the best approach, in many cases, this approach is superior to running SSX from cold start. The hybrid PD-SSX framework can be enhanced by using a prediction of the optimal partition. Although we could not find a fully reliable prediction method we show that if a family of related problems is solved, a near exact prediction of the optimal partition combined with crossover scheme can outperform the standard approach.

6. References

(Adler 89) Ilan Adler, Renato D.C. Monteiro, A geometric view of parametric linear programming, Industrial Engineering and Operations Research Department, University of California at Berkeley, Berkeley CA 94720 USA, February 1989.

(Altman 92)A. Altman, J Gondzio, An efficient implementation of a higher order primal dual interior point method for large scale linear programming, Systems Research Institute, Polishacademy of Sciences, Newleska 6, 01-447, Warsaw Poland, 1992.

(Bixby 90) R.E. Bixby, Implementing the simplex method: the initial basis, TR-90-32, Department of Mathematical Sciences, Rice University, Houston, TX 77251, USA, December 1990.

(Bixby 91) R.E. Bixby, J.W. Gregory, I.J. Lustig, R.E. Marsten, D.F. Shanno, A very large scale linear programming: a case study in combining interior point and simplex methods.

TR-91-11, Department of Mathematical Sciences, Rice University, Houston, TX 77251, USA, May 1991.

(Bixby 92) R.E. Bixby, M.J. Saltzman, Recovering an optimal basis from an interior point solution, Technical report #607, Department of Mathematical Sciences, Clemenson University, Clemenson, South Carolina, USA March 1992.

(El-Barky 91) A.S. El-Barky, R.A. Tapia, Y. Zhang, A study of indicators for identifying zero variables in interior point methods, presented to ICIAM 91, Washington D.C. July, 1991.

(Forrest 90) J. Forrest, J. A. Tomlin, OSL Optimization subroutine library version 2.0, User Guide and Reference Manual, IBM, 1990.

(Gay 89) David M. Gay, Stopping tests that compute optimal solutions for interior point linear programming algorithms, Numerical Analysis Manuscript 89-11, AT&T Bell Laboratories, Murray Hill N.J., USA 07974, December 1989.

(Guler 91) O. Guler, Y. Ye, Convergence behaviour of some interior point algorithms, Working paper 91-4, The College of Business Administration, The University of Iowa, Iowa City, IA, 1991.

(Guler 92) O. Guler, C. Roos, T. Terlaky, J. Ph. Vial, Interior point approach to the theory of linear programming, Research Report 1992.3, Faculte des Sciences Economiques et Sociales, Universite de Geneve, February 1992.

(Levkovitz 91) R. Levkovitz, G Mitra, M. Tamiz, Integration of the interior point method within simplex: experiments in feasible basis recovery, Presented in APMOD91 Symposium, Brunel University, U.K., January 1991.

(Levkovitz 92) R. Levkovitz, Interior point methods for large scale linear programs, theory and computational algorithms, Ph.D. Thesis, Brunel University, Uxbridge UK, October 1992.

(Lustig 90) Irving J. Lustig, Roy E. Marsten, David F. Shanno, On implementing Mehrotra's predictor corrector interior point method for linear programming, Rutcor Research Report # 26-90, Rutcor, Rutgers University, New Brunswick ,NJ 08903 USA, May 1990.

(Megiddo 88) N. Megiddo, Switching from a primal dual newton algorithm to a primal dual (interior) simplex algorithm, RJ 6327 (61996), Computer Science/Mathematics, IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099, USA, 7 November 1988.

(Megiddo 89) N. Megiddo, M. Shub, Boundary behaviour of interior point algorithms in linear programming, Mathematics of Operations Research Vol 14. No. 1., February 1989.

(Megiddo 91) N. Megiddo, On finding primal and dual optimal bases, ORSA Journal on Computing VoL 3 No. 1, Winter 1991.

(Mehrotra 90) S. Mehrotra, High order methods and their performance, Technical report 90-16R1, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 60208-3119, U.S.A, July 1990.

(Mehrotra 91) S. Mehrotra, Y. Ye, On finding the optimal facet of linear programming, Technical report 91-10, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 60208-3119, U.S.A, June 1991.

(Mitra 88) G. Mitra, M. Tamiz, J. Yadegar, Experimental investigation of an interior search method within a simplex framework, Communications of ACM Vol. 21 No. 12, December 1988.

(Zhang 91) Yin Zhang, Richard A. Tapia, On the convergence of interior point methods to the centre of the solution set in linear programming, Department of Mathematics and Statistics, University of Maryland, Baltimore County Campus, Baltimore, Maryland 21228 USA, September 1991.

Research Report	Termination Criterion $\in$
Mehrotra (90)	10^{-14}
Levkovitz (92)	10^{-10}
Aaltman (92)	10^{-6}-10^{-8}
Lustig et al. (90)	10^{-8}

Table 2,1 Termination Criterion Values

Relative Duality Gap	Hits	Misses
8.3E-3	1488	11
6.4E-3	1157	10
2.7E-3	1198	9
9.0E-4	1248	5
3.8E-4	1267	4
1.2E-4	1313	3
9.4E-6	1375	2
5.0E-7	1429	0
2.5E-10	1453	0
6.2E-13 Determined Solution set	1454	0

Table 2.2 Applying the recognizable and determined set criteria to the problem MAROS:

Relative Duality Gap	Hits	Misses
2.7E-7	864	562
1.3E-10	864	5
6.0E-12 Determined solution set	864	0

Table 2.3 Applying the recognizable and determined set criteria to the problem ORNA3
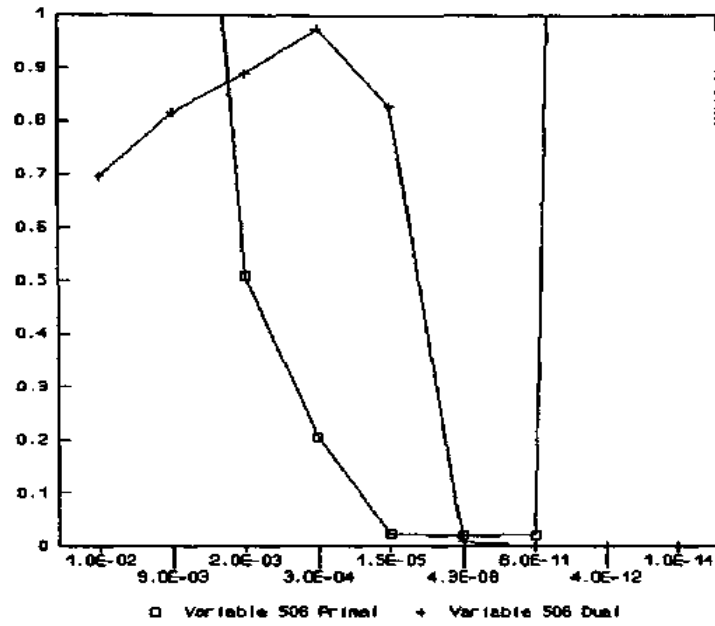
Figure 2.1: The NETLIB problem GROW22, behaviour of variable 506
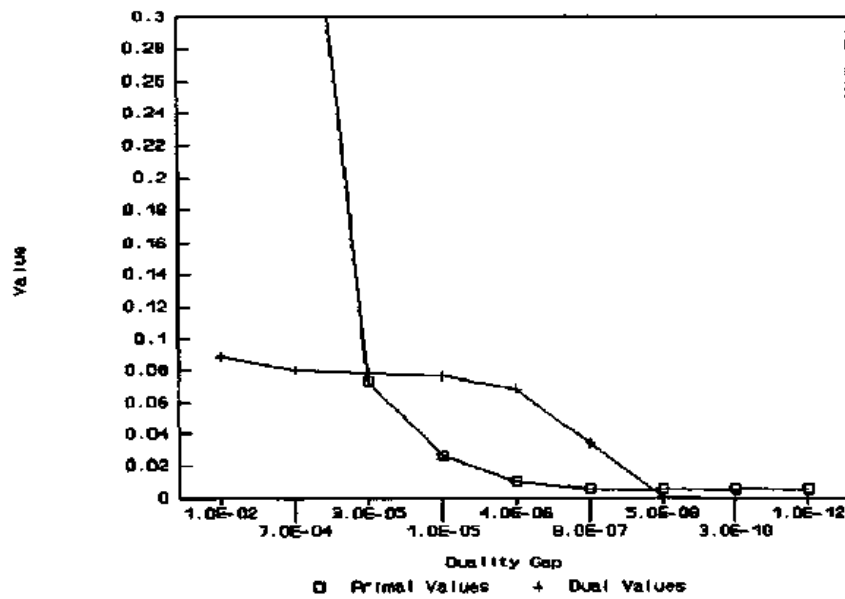

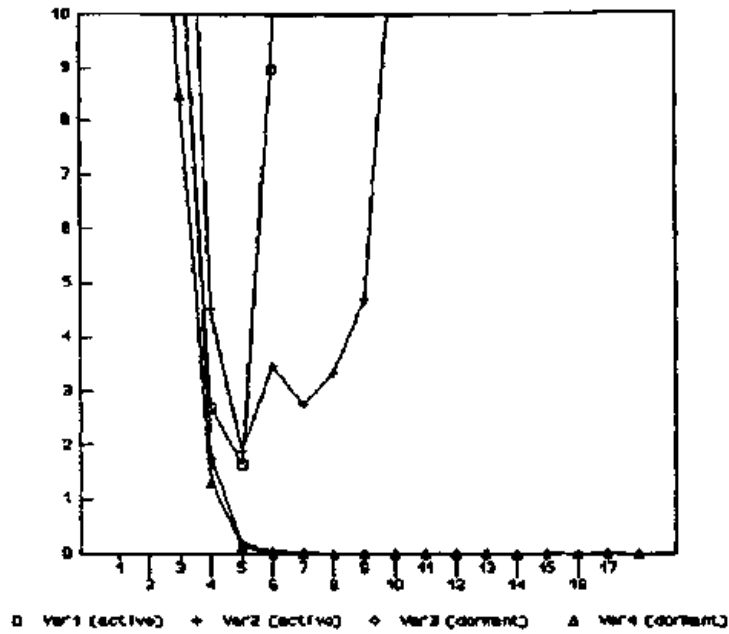Figure 2.2: The NFTLIB problems ganges, behaviour of variable 627

21

Figure 3.1 the primal dual indicators on four variables of the problem Stair
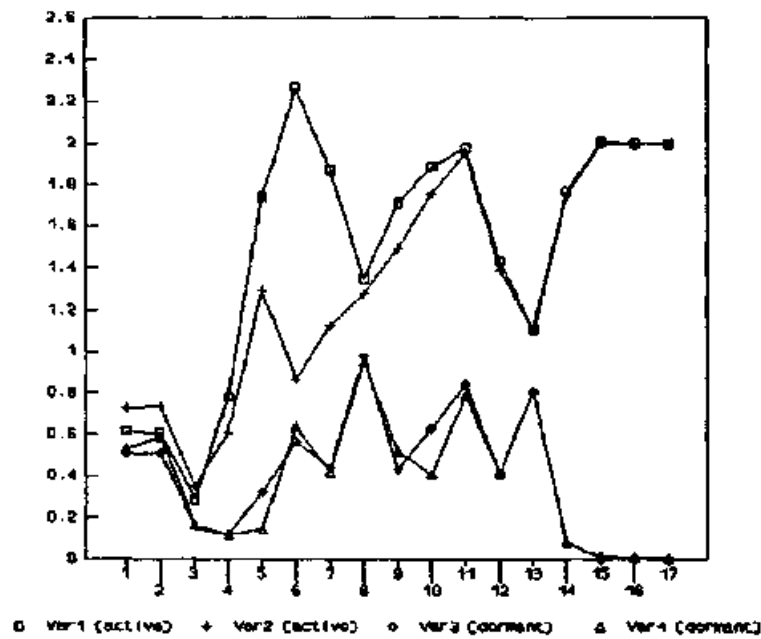

Figure 3.2 the sum of Tapia's indicators on four variables of the problem Stair

22
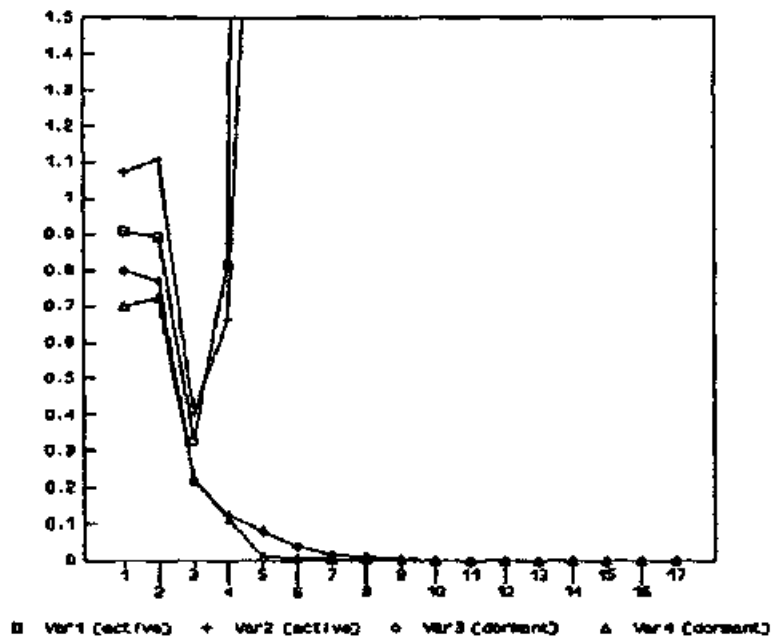
Figure 3.3 the 5 indicators on four variables of the problem Stair

Problem Ship 121 Rows 1152 Columns 5427 nonzeros 21597: P1=2, P2=100		
Duality Gap	Hits	Misses
1.2 E-2	4539	14
4.2 E-3	4736	16
6.2 E-4	4751	18
1.2 E-4	4792	4
4.6 E-5	4793	5
2.6 E-6	4805	5
2.5 E-9	4805	1
1.8 E-12	4805	0

Table 3.1 Prediction of the partition in the problem SHIP12L

Problem Maros Rows 847 Columns 1443 nonzeros 10006				
	Pl = 10, P2 = 10000		Pl=5, P2=100	
Duality Gap	Hits	Misses	Hits	Miss
1.3 E-3	1148	11	462	0
6.4 E-3	1157	10	464	0
2.7 E-3	1198	9	467	0
9.0 E-3	1248	5	473	0
3.8 E-4	1267	4	475	0
1.2 E-4	1313	3	479	0
9.4 E-6	1375	2	484	0
5.0 E-7	1429	0	1429	0
2.5 E-10	1453	0	1453	0
6.2 E-13	1454	0	1454	0

Table 3.2 Prediction of the partition in the problem MAROS

Problem Ganges Rows 1125 Columns 1522 nonzeros 6569				
	Pl = 10, P2 = 10000		Pl=5, P2=100	
Duality Gap	Hits	Misses	Hits	Miss
1.3 E-3	1	0	56	74
1.8 E-4	10	0	86	174
2.8 E-5	11	0	87	175
1.3 E-5	11	0	87	175
3.5 E-6	47	0	109	173
7.5 E-7	187	0	249	170
8.5 E-7	201	0	259	158
5.1 E-9	259	0	259	25
6.0 E-12	259	0	259	0

Table 3.3 Prediction of the partition in the problem GANGES

C_S C_N

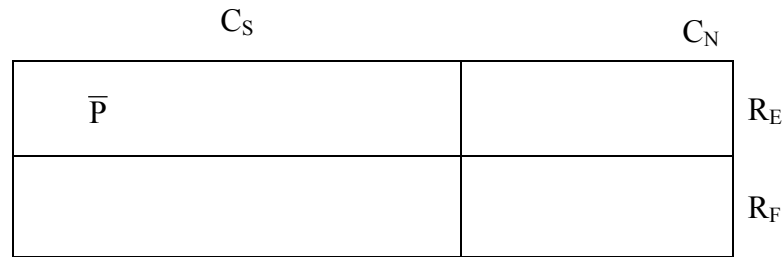| $\overline{P}$		R_E
		R_F

Figure 4.1 construction of the restricted problem

Model	Rows	Cols.	Non Zeros	PD Iterations (Duality gap 10^{-12})	Recovery Iterations	CPLEX simplex
Stocfor1	118	111	474	17	89	31
Israel	175	142	2358	18	52	168
e226	224	282	2767	18	88	336
car2	400	1200	33000	15	100	4000[+]
degen2	445	534	4449	12	1300	800
agg3	517	302	4531	18	100	129
fffff80	525	854	6235	23	203	735
bn11	644	1175	6129	28	800	3197
25fv47	822	1571	11127	24	252	2576
orna3	883	764	4869	15	100	1200[+]
truss	1001	8806	36642	16	400	9487
Ship121	1152	5427	21597	16	90	897
ganges	1310	1681	7021	17	240	598
stocfor2	2158	2031	9492	24	300	1072
bn12	2325	3489	16124	29	1500	5418
ken7	2427	3602	8404	14	434	-
stocfor3	16676	15695	74004	34	1500	10740

Table 4.1 Basis recovery from optimal solution
[+] Solved with OSL version 2.0 (Forrest and Tomlin 90)

Problem Ship121 P1=2, p2=100			
Duality Gap	Hits	Misses	Recovery Iterations
1.2 E-2	4539	14	213
4.2 E-3	4736	16	115
6.2 E-4	4751	18	131
1.2 E-4	4792	4	95
4.6 E-5	4793	5	106
2.6 E-6	4805	5	114
2.5 E-9	4805	1	91
1.8 E12	4805	0	90

Table 4.2 Using prediction and basis recovery for the problem SHIP12L

Problem Car2.mps P1=2, p2=100			
Duality Gap	Hits	Misses	Recovery Iterations
1.4 E-3	798	0	1123
1.8 E-5	791	0	260
7.9 E-7	798	0	150
2.6 E-9	800	0	100
1.4 E-12	800	0	100

Table 4.3 Using prediction and basis recovery for the problem CAR2

Model	Rows	Columns	Nonzeros	Simplex iterations
SCFXM1	331	457	612	530
SCFXM2	661	914	5229	1120
SCTAP3	1481	2480	10734	1930
SCTAP2	1091	1880	8124	1400
SCFXM3	991	1371	7846	1650

Table 4.4 Problem statistics

Duality Gap	SCFXM1			SCFXM2			SCFXM1			SCTAP2			SCTAP3		
	Hit	Miss	Rec	Hit	Miss	Rec	Hit	Miss	Rec	Hit	Miss	Rec	Hit	miss	Rec
1.0 E-3	264	2	163	76	0	1006	357	30	615	1679	0	148	2314	0	205
1.0 E-4	276	2	155	458	4	550	812	8	462						
1.0 E-5	282	0	141	556	1	298	836	5	395	1679	0	148	2314	0	205
1.0 E-6	288	0	127	568	0	253	845	1	400						
1.0 E-7	292	0	119				845	1	400						
1.0 E-8				568	0	252				1679	0	148			
1.0 E-9													2314	0	205
1.0 E-10	272	0	119	568	0	252	846	0	416	1679	0	148	2314	0	205
1.0 E-11	272	0	119	568	0	252	846	0	416	1679	0	148	2314	0	205

Table 4.5 Basis recovery results using prediction for a family of problems