ALGORITHMS FOR SOLVING LINEAR PROGRAMS: SIMPLEX VS. ELLIPSOID

by: Carolina B. Baguio[1]

1. Introduction

A linear programming problem of the form

$$\begin{array}{lll} \text{Min} & C^T x & \\ \text{s.t.} & Ax \geq, b & (1A) \\ & x \geq, o & \end{array}$$

can be solved either by George B. Dantzig's simplex method, which has efficiently solved large linear programs for over three decades and by the latest ellipsoid algorithm developed by Yudin, Nemirovskii and Shor.

You may ask which of these two is efficient and good. But what is a good algorithm, by the way? How is the speed of the algorithm to be measured? Perhaps, one may compare the performance of true algorithms for the same problem on a few instances of special interest, but how relevant is that to their performance on other instances? On the other hand, one may base the comparison on a large number of instances chosen in a regular or random fashion, but this can be very expensive and yet might not provide a reliable indication of performances on instances much higher than those tested. Each instance is associated with particular numerical data and a problem is the class of instances of a specified form. It is expected that large instances will usually be solved more slowly than small instances. A good algorithm is one that is polynomially bounded, is of complexity $O(n^P a^q)$ for some p and q. The exponents p and q should be as small as possible (see Klee [3]).

2. Objectives of the paper:

1. To give an update on the computing efficiency of the simplex method and ellipsoid algorithms.
2. To outline the different steps of the dimensional reduction variant of the ellipsoid algorithm.

---

[1] Assistant Professor, College of Sciences and Mathematics, Math Department.

3. To illustrate the steps of the algorithm through a simplex example and compare the results with both the graphical and simplex methods.

3. Khachiyan's Stunning Theoretical Breakthrough

The simplex method has appeared in practice to run in polynomial time, i.e., the number of operations to solve a problem grows at worst polynomially with the number of digits needed to specify it. But it seems that neither a proof of this fact nor a satisfactory theoretical explanation for the success of the simplex method has ever appeared in any literature. In fact, with the several standard pivot rules, the simplex method is known to solve artificially-constructed examples in exponential time. It has been found out by many researches along this field that a major and long outstanding problem is whether there exists a pivot rule for the simplex method that permits it to run in (low order) polynomial time. This problem has remained unsolved for quite some time. However, the broader questions of whether there is any algorithm that will solve linear programs in polynomal time has been settled lately.

A young Russian mathematician, Yeonid G. Khachiyan, has made a somewhat incredible theoretical breakthrough by showing that linear programs can be solved in polynomial time by a variant of an iterative ellipsoidal algorithm developed by N.Z. Shor. The algorithm is simple, and has the feature not shared by the simplex method: that successive iterates are neither normally basic solutions nor even rational when the data are rational [1].

Apparently, the ellipsoid algorithm does not appear to be competitive with the simplex for some practical problems. However, Khachiyan's finding does renew interest in seeking for a version of the simplex method that always runs in polynomial time. Moreover, it motivates researchers to possibly refine the algorithm and arrive at other approaches in solving large-scale programs as efficiently as the simplex.

4. Dimensional Reduction Variant of the Ellipsoid Algorithm

In order to solve (1A), duality theory [ə] is applied to generate a system

$$MZ \leq d^1 \qquad (1B)$$

Where $Z = \begin{bmatrix} x \\ y \end{bmatrix}$

$$M = \begin{bmatrix} C \\ -A \\ O \\ -I \\ O \end{bmatrix} \begin{bmatrix} -B \\ O \\ A^T \\ O \\ -I \end{bmatrix} \qquad \text{and } d^1 = \begin{bmatrix} O \\ -b \\ c \\ O \\ O \end{bmatrix}$$

If $Z^* = \begin{bmatrix} x^* \\ y^* \end{bmatrix}$ solve (1B) then x^* solves the primal (1A)

To solve (1B) using the ellipsoid method, the system must be perturbed: $Mz \cdot d^1 + O$ where O is a vector of appropriately chosen numbers. The solution obtained from such a perturbation necessarily is an approximate solution of (1B).

Geometrically, the method generates a sequence of ellipsoids with decreasing volumes and whose centers approach the solution set. The solution sets of linear programming problems are tiny simplexes or triangular cylinders and the ellipsoids may become very elongated. For this reason, there is the inherent instability if Khachiyan's algorithm is implemented. But Judin and Nemirovski [a] redefined the ellipsoid to be a positive definite linear transformation. Using this idea to define a change of variables, the problem can be transformed to one of finding the solution in a spheroid, thus making the algorithm stable.

Based on these geometric ideas, we shall discuss the acceleration techniques of the variant of ellipsoid algorithms.

A. Consider the system (1B). We shall start with the unit sphere by setting

$$Z_o = 0$$
$$d_o = \frac{d^1}{r_o}$$
$$S_o = I$$

where r_o is the appropriately chosen radius. The algorithm generates a sequence

Z_k, d_k, S_k where

Z_k = approximate solution in the original space

d_k = the updated right-hand side in the current space

S_k = the change of variables mapping the current space back to the original one.

Hence, the system of linear inequalities is transformed at each iteration, and the test for solution is to check if $d_k \geq O$. If so, $>_oK$ is a solution to the original problem. Otherwise, some constraint must be violated, say constraint i.

For a simpler notation, let us denote the current iterate without a subscript, and the next iterate with a subscript "plus," +. If Z, d, and S are the current iterates, then the next ones are defined by:

$$\bar{m}_{\lambda}/= S^T m_i \,, \qquad (4A)$$
$$g = m_i/||\,. \qquad (4B)$$
$$q = \frac{1}{n+1} g, \qquad (4C)$$
$$Z_+ = Z +. Sq, \qquad (4D)$$
$$d_+ = d - MSq \qquad (4E)$$
$$S_+ = S A^{1/2} \qquad (4F)$$

Where $A^{1/2}$ is given by the unique Choleskey decomposition of

$$A = \frac{n^2}{n^2-1} \left[1 - \frac{2}{n+1} \; g \; g^T \right] \qquad (4G)$$

A is a symmetric positive definite. Hence, $A^{1/2}$ is a lower triangular matrix with positive diagonal elements.

B. Basic Method vs. Acceleration Techniques

This constructs an ellipsoid based on the subspace parallel to the violated constraint. (Fig. 1). This ellipsoid contains the half-spheroid containing the solution set. Instead, we may construct the ellipsoid based on the constraint itself. (Figure 2). The solution set is proportionally larger within the latter ellipsoid, and hence, the growth factor of the solution set within the unit spheroid will be larger and the algorithm accelerated.

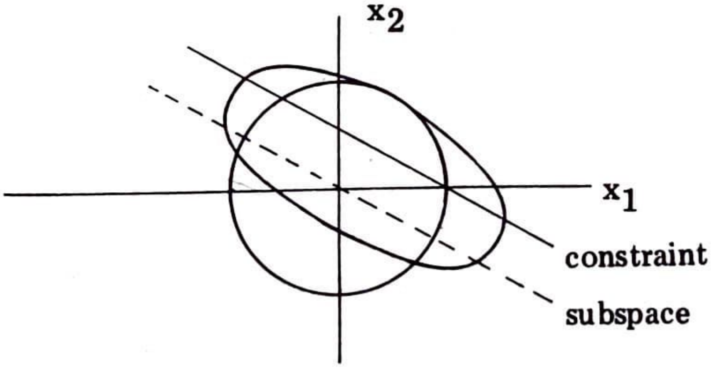


Figure 1: Basic Ellipsoid Method

(a) Deep-Cut Method

The first accelerated variant is defined by replacing (4B) and(4G) with

$$q = - \frac{(1 + n\delta)}{(n + 1)} g, \qquad (4H)$$

$$A = \frac{n^2 (1 - \delta^2)}{n^2 - 1} \left[1 - \frac{2 (1 + n\delta)}{(n+1)(1+\delta)} gg^T \right], \qquad (4I)$$

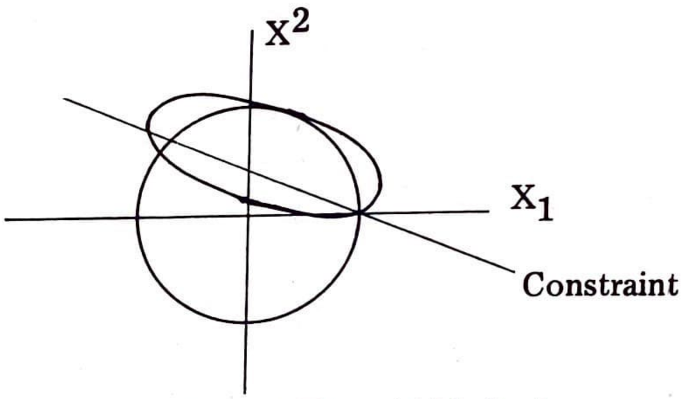


Figure 2. Accelerated Ellipsoid Method

Where S is the distance between the constraint and the origin, given by

$$= | d_i | / 11 \overline{m}_i 11 \qquad (4J)$$

and d_i denotes the ith component of d ,

(b) The second acceleration technique is based on the duality theory. Through a process of dimensional reduction, this notion provides a more fundmental acceleration than does the deeper cut technique.

The complementary slackness conditions of (A) and its dual are:

$$Y^T [A x - b] = O \qquad (4K)$$

$$[Y^T A - C^T] x = O \qquad (4L)$$

The above conditions must be satisifed for optimality, and to avoid degeneracy exactly one of each pair $(Y_i, A_{xi} - b_i)$ defined by (4K) shall be positive and one equal to zero. Similar remarks apply to the pairs defined by (4L).

(c) Reducing dimension

While performing the accelerated ellipsoid steps, we may find constraints lying outside the current spheroids. If the constraint, is not currently satisfied we may conclude that the system (1B) is infeasible. If, on the other hand, the constraint is satisfied, we may conclude that it is nonbinding and hence its complement must be binding at any solution of the system (1B). The nonbinding constraint is eliminated from further consideration and the dimension of the system can be reduced by restricting the search to the hyperplane defined by the binding constraint.

This procedure has several advantages. First, the growth factor is controlled by the ratio (n+1)/n where n denotes the number of variables. By reducing the dimension, this ratio is increased, thus accelerating the algorithm. Secondly, assuming nondegeneracy, we must eventually eliminate one of every pair defined by (4K) and (4L). Hence, we will be performing n dimensional reductions and get a point satisfying all binding constraints. This yields an exact solution to the complementary slackness conditions and hence to the linear program.

D. Test for constraints lying outside the sphere at each iteration in the accelerated variant algorithm:

If the inequality $|d_{\hat{\lambda}}| / \|\bar{m}_{\hat{\lambda}}\| > 1$ (4M) is satisfied then the constraint could be nonbinding and may be eliminated (if $d_i > 0$); the system is infeasible ($d_{\hat{\lambda}} < 0$). The elimination procedure on a binding constraint j = j (i) is carried out as follows: The solution must lie in the intersection of the hyperplane defined by the binding constraint and the sphere (Fig. 3). A special step is taken to the binding constraint (Fig. 3) given by:

$$
\begin{aligned}
q &= \delta g \\
Z_+ &= Z + Sq \\
d_+ &= d - M\,Sq, \\
S_+ &= S
\end{aligned}
$$

where now = $|d_j| \,/\!/\!/ m_j /\!/$, defines both distance and direction.

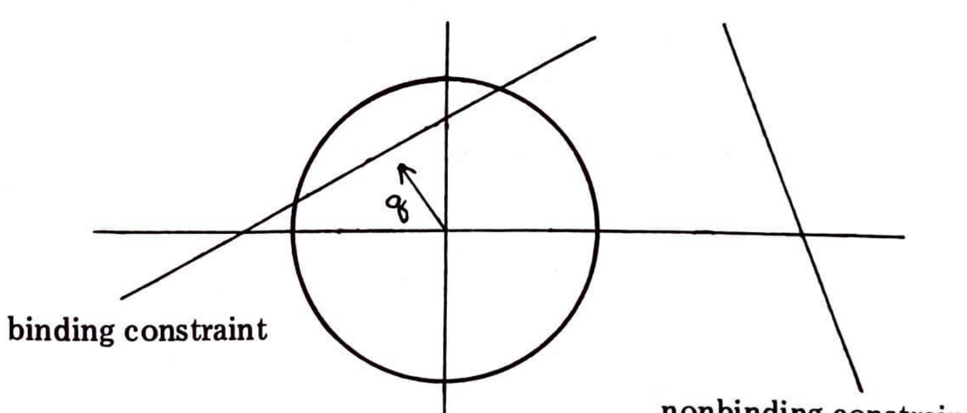


Figure 3: Special Step

Finally, using a Householder transformation, H, which defines a reflection, the gradient of the binding constraint is transformed onto the first coordinate axis in the current space [4]. The restriction to this subsphere yields a smaller dimensional problem. Since H is orthogonal, the system

$$M\ H^T\ H\ Sw \leq d \qquad (4N)$$

is equivalent to the original problem (1B) scaled to the unit sphere. We reduce the dimension by deleting rows i and j from M and zeroing the first column of MH^T. The radius of the new subsphere is $\hat{r} = \sqrt{1-S2}$ and the reduced system is denoted by

$$\hat{M}\ \hat{S}\ \hat{w} \leq d \qquad (4\hat{N})$$

The right-hand side is rescaled in the unit sphere, $\hat{d} = \hat{d}/\hat{r}$. For a step, q, made in the subsphere, the corresponding step in the previous higher — dimensional space is given by $\hat{r}\ \hat{H}^T\ \hat{S}\ \hat{q}$.

The accelerated variant is then applied to the reduced system until another constraint is identified which lies outside the current spheroid. We proceed in this manner until either the system is determined to be infeasible or n reductions have been performed yielding a solution.

5. An illustrative Example

(a) We will choose a simple linear program to illustrate the various steps of the algorithm:

$$\begin{aligned} &\text{Min } x \\ &\text{s. t.} \quad x \geq 1 \\ &\qquad\quad x \geq 0 \end{aligned} \qquad (5A)$$

the dual of which is

$$\begin{aligned} &\text{Max } y \\ &\text{s.t.} \quad Y \leq 1 \\ &\qquad\quad Y \geq 0 \end{aligned} \qquad (5B)$$

In our notation in (1B), we have

$$M = \begin{bmatrix} -1 & -1 \\ -1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and } d^1 = \begin{bmatrix} 0 \\ -1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

(a) First Iteration —

Choosing a circle of radius $r_o = 2$, we have

$$Z_o = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad d_o^T = (0 -½, ½, 0, 0), \text{ and } S_o = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Check: $Mz \leq d$

$$\begin{bmatrix} 1 & -1 \\ -1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leqslant \begin{bmatrix} 0 \\ -½ \\ ½ \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leqslant \begin{bmatrix} 0 \\ -½ \\ ½ \\ 0 \\ 0 \end{bmatrix}$$

(b) Second Iteration

Apparently, constraint 2 is violated and an accelerated ellipsoid step (Fig. 4) is calculated:

$$\bar{m}_2 = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad g = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad \delta = ½$$

$$q = \begin{bmatrix} 2/3 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} 1/9 & 0 \\ 0 & 1 \end{bmatrix}, \quad A^{½} = \begin{bmatrix} 1/3 & 0 \\ 0 & 1 \end{bmatrix}$$

and $Z_1 = \begin{bmatrix} 2/3 \\ 0 \end{bmatrix}$, $d_1T = (2/3_1 \; 1/6_1 \; 1/2_1 \; 2/3_1 \; 0)$

and $S_1 = \begin{bmatrix} 1/3 & 0 \\ 0 & 1 \end{bmatrix}$

(c) Third Iteration

Next, constraint 4 is outside the new sphere (Fig. 5) and must be inactive since

$$\frac{2/3}{\|\bar{m}_4\|} = 2 > 1, \text{ where } \bar{m}_4 = \begin{bmatrix} -1/3 \\ 0 \end{bmatrix}$$

Its complement is constraint 3 which must be binding since

$$1/2/\|\bar{m}_3\| = 1/2 < 1 \quad \text{where } \bar{m}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

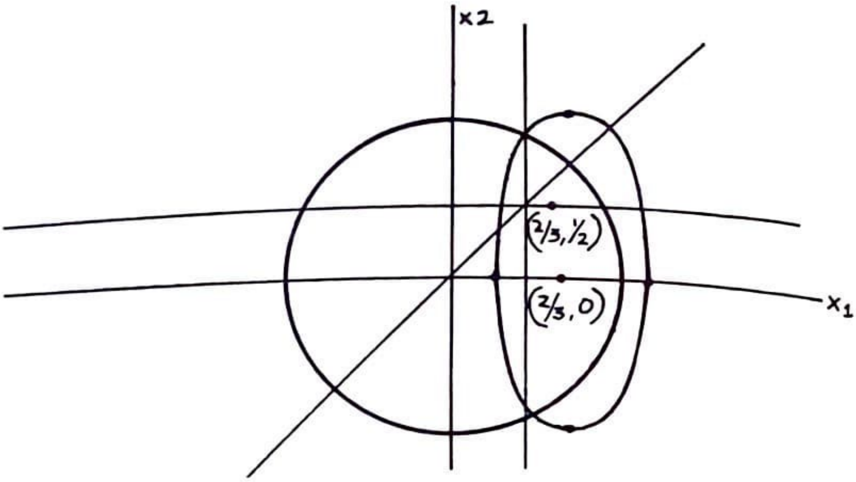


Figure 4: Accelerated ellipsoid step

The special step (Fig. 5) is calculated:

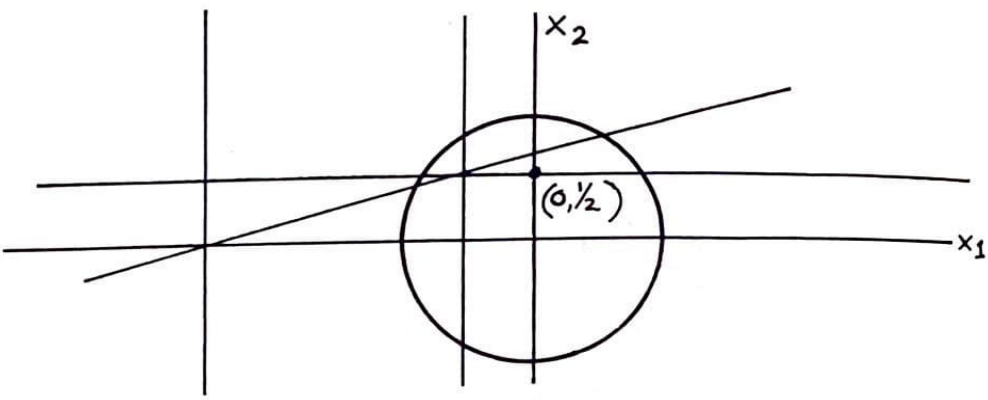


Figure 5: Step to binding constraint

$$\bar{m}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad g = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \delta = ½$$

$$q = \begin{bmatrix} 0 \\ ½ \end{bmatrix}, \quad Z_2 = \begin{bmatrix} 2/3 \\ 1/2 \end{bmatrix}, \quad S_2 = S_1 = \begin{bmatrix} 1/3 & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$d_2^T = (-1/6, 1/6, 0, 2/3, 1/2)$$

Eliminating on the third constraint gives

$$H = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}, \quad MH^T = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ -1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad HS = \begin{bmatrix} 0 & -1 \\ 1/3 & 0 \end{bmatrix}$$

and the reduced dimensional system is

$$\hat{M}\hat{S} = \begin{bmatrix} 0 & -1 \\ 0 & 1 \\ * & * \\ * & * \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1/3 & 0 \end{bmatrix} \leq \begin{bmatrix} -1/3 & 3 \\ 1/3 & 3 \\ * & \\ * & \\ 1/ & 3 \end{bmatrix}$$

while the new starting radius is $r = \sqrt{3/2}$

Since the problem is now in one dimension (Fig. 6), we take a final step to any violated constraint, in this case the first constraint,
to obtain:
violated constraint, in this case the first constraint,

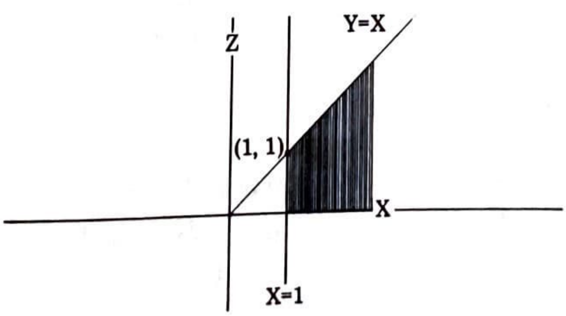
to obtain: $\bar{m}_1 = \begin{bmatrix} 1/3 \\ 0 \end{bmatrix}, \quad g = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \delta = -1/3\sqrt{3}$

$$\hat{q} = \begin{bmatrix} -1/\sqrt{3} \\ 0 \end{bmatrix}, \quad \hat{r}\ H^T \hat{S}\hat{q} = \begin{bmatrix} -1/6 \\ 0 \end{bmatrix},$$

$$Z_3 = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

Since all remaining constraints are satisfied, Z_3 is a solution in the unit sphere and $Z_* = r_0 Z_3 = 2Z_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is a solution to the original problem.

(b) Using the graphical Method, we have



From the shaded region, we can see that the minimum value of the function f(x) = x is at the point (1, 1), which is consistent with the result of (a).

(c) Using the simplex method, it is convenient to use the dual problem.

$$\begin{array}{ll} \text{Max} & Y \\ \text{s.t.} & y \leq 1 \\ & y \geq 0 \end{array}$$

Add only one slack variable S_1. So, the problem becomes

$$\begin{array}{ll} \text{Max} & Y \\ \text{s.t.} & Y + S_1 = 1 \\ & Y, S_1 \geq 0 \end{array}$$

Initial Tableau:

Basis	Z	Y	S_1	Current values
S_1	0	1	1	1
Max	–1	1	0	0

S_1 leaves the basis while Y enters it.

Second Iteration yields the maximum values

Basis	Z	Y	S_1	Current values
Y	0	1	1	1
Max	–1	0	–1	–1

Hence, Max Z = 1 and Y = 1 (dual value)

Scanning the last row, the primal value can be picked up as X = 1 which is in the S_1 column. And Min Z = 1, by the Weak-Duality Theory.

6. Concluding Remarks

This paper so far has exposed the theoretical breakthrough in the search for algorithm which solves linear programs in a polynomial time which is an inherent limitation of the simplex method of George B. Dantzig. This recent development seems uncompetitive with the simplex method since its practical importance appears premature. Nevertheless, we can predict that Khachiyan's result will inspire mathematicians to discover more efficient techniques for solving very large programs which resist current methods.

In the discussion of the ellipsoid algorithm, it is apparent that the choice of starting guess Z_0 and initial radius r_0 will surely have a bearing on the practical importance of the new algorithm. In terms of Khachiyan's polynomial time result, $r_0 = 2_L$, where L is a logarithmic function of the elements of M and d and the dimension of the problem. In our example in the last section we could have used a very large radius — $r_0 = 2048$. This will make the constraint lie relatively close to the origin and the accelerated step is practically the same as the basis ellipsoid step.

What this paper really points to is a development of simple heuristics to generate an initial Z_0 and a radius r_0 such that $\| Z_0 - Z \| < r_0$. It is highly evident that much development has yet to be done. It could be that such heuristics combined with the dimensional reduction algorithm will provide a procedure which might be competitive with the simplex method.

7. References

/1/ Editorial, (1980). Mathematics of Operations Research, 5:1 F[1]

/2/ Jones, P.C. and Marwil, E.S. (1982). Dimensional Reduction Variant of the Ellipsoid Algorithm for Linear Programming Problems, *Mathematics of Operations Research* 7:1 245-252, M[1]

/3/ Klee, Victor, (1980). Combinatorical Optimization: What is the State of the Art, *Mathematics of Operations Research* 5:1 1-21 F[1]

/4/ Stewart, Q. Q. (1980). Introduction to Matrix Computations. Academic Press, New York.

/5/ Todd, Michael J. (1982). On Minimum Volume Ellipsoids Containing Part of a Given Ellipsoid , *Mathematics of Operations Research,* 5:1 F[1]