KHACHIYAN'S ALGORITHM FOR LINEAR INEQUALITIES: OPTIMIZATION AND IMPLEMENTATION*

By

FREDERIC BISSHOPP

Brown University

Abstract. An optimized version of Khachiyan's algorithm is developed here, and an APL implementation of it is provided. The operation count for M linear constraints on N variables is brought from the $O(N^4M^2)$ of the original algorithm to $O(N^4M \ln N)$ with a smaller coefficient. There is no significant change of the storage requirement, which remains at $O(NM)$ locations.

Introduction. The problem at hand is the solvability of a set of M linear constraints on N variables, i.e.

$$Ax < b \tag{1}$$

where x is a column of N real numbers, A is an M by N matrix of integers, and b is a column of M integers. The inequalities can be regarded as the set of bounds in an optimization problem where an object function $\Phi(x_1 \cdots x_N)$ is to be maximized subject to linear constraints. If $\Phi = c^{\mathsf{T}}x$ where c^{T} is a row of N integers, the full problem can be posed in the form of (1) by introduction of a dual problem [3]. The implementation to be given here is of the set (1) alone.

The solvability of (1) is based on the observation that every solution is an interior point of a region bounded by portions of a number of hyperplanes,

$$A_i^{\mathsf{T}}x = b_i, \tag{2}$$

where A_i^{T} is the ith row of A. The transpose of A_i^{T}, to be denoted by A_i, is the ith column of A^{T}. The role of the strict inequalities is pointed out in [3], and the related problem where $<$ is replaced by $\leq$ is discussed there.

A nondegenerate vertex of (1) is defined as a solution of

$$\bar{A}v = \bar{b} \tag{3}$$

where A consists of N linearly independent rows of $\bar{A}$ and $\bar{b}$, of the corresponding entries of b. A degenerate vertex is defined by $\bar{M} < N$ linearly independent rows of A and the corresponding entries of b. If the boundary of a solution set contains a degenerate vertex, then that set is unbounded, i.e. solutions where $\|x\|_2 \to \infty$ exist. Conversely, if a solution set is bounded, its boundary contains at least $N + 1$ nondegenerate vertices. Also, a bounded solution set contains (or is) a simplex which has exactly $N + 1$ nondegenerate

* Received January 15, 1980.

vertices on its boundary. In Secs. 2 and 3 of this paper boundedness of the solution set will be assumed in order to provide a clean description of the algorithm. In Sec. 4 one part of the algorithm will be slightly modified to take account of possible unbounded solution sets.

In a direct approach to the solvability of (1), the search for a vertex on the boundary of the solution set requires the isolation of as many as $M!/N!(M-N)!$ vertices, each taking $O(N^3)$ operations (add, multiply and assign). In practice, M is often significantly larger than N, and sometimes N is quite large as well, so a direct approach can involve prohibitively many operations.

The advantages of Khachiyan's algorithm [1, 2] are:

(1) It is constructive: if (1) is solvable the algorithm produces a point in the solution set.

(2) If no solution is found within a number of $O(N^3M)$ of iterations, (1) is not solvable.

(3) The number of operations per iteration is of $O(NM)$.

(4) The storage requirement is $O(NM)$ locations.

Regardless of the number of operations required to find a vertex, the number of vertices is of $O(M^N)$ while the total operation count of Khachiyan's algorithm is $O(N^4M^2)$. The improvement of Khachiyan's original estimate is given in [3].

The algorithm presented here is very similar to Khachiyan's algorithm, but the emphasis is on the isolation of least upper bounds and greatest lower bounds on the quantities that figure in the number of iterations. An initial determination of improved bounds takes $O(NM \log M)$ operations, but we are adequately compensated by an immediate reduction of the maximum number of iterations to a number of $O(N^3 \ln N)$. The leading term in the bound on the number of iterations is $2N(N+1)^2 \ln N$ as $N \to \infty$.

The iteration within this algorithm is an optimized version of Khachiyan's iteration in which the volume of each member of the sequence of ellipsoids is minimized over all the free parameters. Also, a failure criterion is included, so that the algorithm does not necessarily run the full course of iterations to find that (1) is not solvable. The extra effort involved is not significant, and each iteration still takes $O(NM)$ operations. In Sec. 5 it is argued that the optimized iteration decides solvability of (1) within a number of iterations that approaches $(1/4)N(N+1)^2 \ln N$ as $N \to \infty$.

An APL implementation of the algorithm is provided in Sec. 6; randomly chosen examples are included.

1. The most distant vertex. If the inequalities (1) have a nonempty solution set its boundary has at least one vertex, be it nondegenerate or degenerate, defined by

$$\bar{A}v = \bar{b} \tag{4}$$

where $\bar{A}$ is $\bar{M} \le N$ linearly independent rows of A. If $\bar{M} < N$, select $\bar{M}$ linearly independent columns of $\bar{A}$ and set $(N - \bar{M})$ elements of v equal to zero. The bound on the radius of that point on a degenerate vertex is necessarily less than the corresponding bound on the radius of a nondegenerate vertex. Thus, to find a sphere that contains all nondegenerate vertices and portions all degenerate vertices, we may assume $\bar{M} = N$ and $\bar{A}$ is not singular.

An essential feature of the formulation of Khachiyan's algorithm is that the entries in A and b are integers. This means that the elements of v are the rational numbers,

$$v_i = p_i/q, \qquad i = 1 \cdots N, \tag{5}$$

where q is the determinant of $\bar{A}$ and p_i is the determinant of the matrix formed by replacement of the ith column of $\bar{A}$ by $\bar{b}$ (Cramer's rule). Since $\bar{A}$ is a nonsingular matrix of integers, the lower bound on the magnitude of q is 1. Since $|\bar{A}|$ is the volume of the rectangular parallelepiped defined by the rows of $\bar{A}$, an upper bound on the magnitude of q is the product of the Euclidean norms of the rows. The bound is attained if the rows of $\bar{A}$ are orthogonal, so

$$1 \leq |q| \leq \prod_{i=1}^{N} \|\bar{A}_i^{\mathrm{T}}\|_2 \tag{6}$$

where $\bar{A}_i^{\mathrm{T}}$ is the ith row of $\bar{A}$. Given no further information about A, the best bound on q at all vertices is

$$1 \leq |q| \leq \prod_{i=1}^{N} (\nabla\|A_i^{\mathrm{T}}\|_2) \equiv Q \tag{7}$$

where the symbol ∇ indicates sorting the Euclidean norms of the rows of A in descending order.

By the same argument for upper bounds, the bounds for the numerators of the elements of v at all vertices are

$$0 \leq |p_j| \leq \prod_{1}^{N} (\nabla\|C_{ji}^{\mathrm{T}}\|_2) \equiv P_j, \qquad j = 1, \ldots, N \tag{8}$$

where C_j is obtained by replacement of the jth column of A by b and C_{ji}^{T} is the ith row of C_j. From the lower bound on $|q|$ and the upper bounds on $|p_j|$, the upper bound on radii of nondegenerate vertices and portions of degenerate vertices is

$$R_0 = \|P\|_2 = \left(\sum_{1}^{N} P_j^2\right)^{1/2}. \tag{9}$$

Since $N + 1$ sorts of M numbers are needed here, it is appropriate to compare these bounds with others that have been given in [1, 2, 3]. There the bounds have been defined in terms of

$$2^L = (1 + MN) \prod_{1}^{M} (1 + |b_i|) \prod_{1}^{N} (1 + |A_{ij}|). \tag{10}$$

Without sorting, the corresponding bound on Q is

$$\begin{aligned}\prod_{1}^{M} \prod_{1}^{N} (1 + |A_{ij}|) &> \prod_{1}^{M} (1 + \|A_i^{\mathrm{T}}\|_1) > \prod_{1}^{M} (1 + \|A_i^{\mathrm{T}}\|_2) \\ &> \prod_{1}^{N} (\nabla\|A_i^{\mathrm{T}}\|_2).\end{aligned} \tag{11}$$

If the matrix A has just one nonzero entry, ± 1, in each row, for example, the bounds are

$$2^M \quad \text{and} \quad Q = 1. \tag{12}$$

Another comparison between present and previous bounds is based on the following heuristic device: let the As and bs be chosen randomly from the integers in an interval

$[\alpha_0, \alpha_1]$. A rough measure of the expectation of the bounds is obtained by substitution of the mean value of the magnitudes of the elements of A and b for the random values. Let $\alpha > 0$ be the mean value of the magnitudes: then the heuristic bounds are

$$(1+\alpha)^{MN} \quad \text{and} \quad Q \approx (\alpha\sqrt{N})^N,$$

$$(1+N)(1+\alpha)^{M(N+1)} \quad \text{and} \quad R_0 \approx \frac{1}{\alpha}(\alpha\sqrt{N})^{N+1}. \tag{13}$$

This kind of estimation will be used later in the discussion of the number of iterations in the algorithm. It should be noted that if α is chosen to be the maximum of the magnitudes rather than the mean, then the expressions for the heuristic bounds provide upper bounds on Q and R_0.

2. The smallest simplex. Here and in Sec. 3 it will be assumed that the solution set of (1) is bounded; in Sec. 4 that restriction will be removed.

Suppose first that the boundary of the solution set has a degenerate vertex defined by $\bar{M} < N$ rows of A and b. Since $N - \bar{M}$ of the coordinates of such a vertex are arbitrary, the solution set has boundary points where $\|v\|_2$ is arbitrarily large: it cannot be bounded.

Near a nondegenerate vertex, v_0 (say), the boundary of the solution set has N distinct rays emanating from v_0. The kth ray is defined by deletion of the kth row of the nonsingular equation,

$$\bar{A}v = \bar{b}, \tag{14}$$

and by choice of the half-line through v_0 that is consistent with (1). Suppose now that any of the rays through v_0 has no other vertex on it. Then the solution set has arbitrarily distant boundary points, and again it cannot be bounded.

Given v_0 and the nearest vertices on the N rays through v_0, the solution set contains (or is) the simplex defined by

$$v_j = (p_{j1} \cdots p_{jN})^T/q_j \qquad j = 0, \ldots, N \tag{15}$$

where the p's and q's are integers. Now

$$\int_0^x dx_k \int_0^{x_k} dx_{k-1} \cdots \int_0^{x_2} dx_1 = \frac{x^k}{(k+1)!} \tag{16}$$

(by induction), and it follows easily that the volume of the simplex defined by $N+1$ vertices is (apart from sign)

$$\begin{aligned} V &= \frac{1}{N!}\left|v_1 - v_0, \ldots, v_N - v_0\right| = \frac{1}{N!}\begin{vmatrix} 1 & 1 & \cdots & 1 \\ v_0 & v_1 & \cdots & v_N \end{vmatrix} \\ &= \frac{1}{N!}\left(\prod_0^N \frac{1}{q_j}\right)\begin{vmatrix} q_0 & \cdots & q_N \\ p_0 & \cdots & p_N \end{vmatrix}. \end{aligned} \tag{17}$$

The rays are distinct, the displacements, $v_i - v_0$, are linearly independent, the bound on the magnitude of the determinant of integers is ± 1, and the lower bound on V is

$$V \geq \frac{1}{N!}\prod_0^N \frac{1}{q_j} \geq \frac{1}{N!}Q^{-(N+1)}. \tag{18}$$

The heuristic comparison of this with the bound previously cited is:

$$2^{-NL} \approx ((1 + MN)(1 + \alpha)^{MN})^{-N} \tag{19}$$

while

$$\frac{1}{N!} Q^{-(N+1)} \approx \frac{1}{N!} (\alpha\sqrt{N})^{-N(N+1)} > (N(\alpha\sqrt{N})^{N+1})^{-N} \tag{20}$$

where α is the mean value of the magnitudes of the entries of A and b.

3. The iterative search for a solution. Here again we assume the solution set of (1) is bounded, for then the sphere of radius R_0 (Eq. (9), Sec. 2) contains the entire solution set and, in particular, the smallest simplex. The aim of the iterative search is to use the current "worst violation of (1)" to define increasingly smaller ellipsoids that still contain the solution set. If the center of the current ellipsoid is a solution of (1), the iteration terminates with it in hand; otherwise the iteration is terminated when any of the violations of (1) indicates there is no solution. One or the other of these things must happen before the volume of the current ellipsoid becomes less than the volume of the smallest simplex.

The counting of operations is of considerable importance in the optimization of Khachiyan's iteration. If one merely adds the calculations needed to perform the optimization, the result takes $O(NM^2)$ operations. Nevertheless, the optimized iteration *will* be developed in that fashion: then it will be reorganized to bring the operation count back to $O(NM)$ at a cost of including $N(M - N)$ more storage locations.

At the outset the bounding sphere/ellipsoid is defined by

$$(x - x_K)M_K^{-1}(x - x_K) = 1 \tag{21}$$

where $K = 0$, $x_K = 0$ and $M_K = R_0^2 I$. Let it be supposed now that we have a Kth positive definite, symmetric matrix M_K and a Kth center x_K that defines a Kth bounding ellipsoid as in Eq. (21). The purpose of the iteration is to construct a smaller ellipsoid that still contains the solution set.

By a rotation of coordinates (never to be computed),

$$M_K = R^{\mathrm{T}}\Lambda^2 R, \tag{22}$$

where

$$R^{\mathrm{T}}R = I \quad \text{and} \quad \Lambda^2 \text{ is diagonal.} \tag{23}$$

Now let

$$\xi = \Lambda^{-1}R(x - x_K) \quad \text{and} \quad \mathscr{A} = AR^{\mathrm{T}}\Lambda. \tag{24}$$

Then the current ellipsoid is

$$\xi^{\mathrm{T}}\xi = 1, \tag{25}$$

the current transformation of (1) is

$$\mathscr{A}\xi = A(x - x_K) < b - Ax_K, \tag{26}$$

and, after the $O(NM)$ operations of the right-hand side of Eq. (26): if the elements of $Ax_K - b$ are all negative then x_K is a solution of (1).

Otherwise, we continue with Eq. (26) expressed in terms of components of ξ in the directions of the rows of $\mathscr{A}$. Let $\mathscr{A}_i^{\mathrm{T}}$ be the ith row of $\mathscr{A}$: its transpose, $\mathscr{A}_i$ is the ith column of $\mathscr{A}^{\mathrm{T}}$. The ith unit vector in ξ-space is

$$\hat{\mathscr{A}}_i \equiv \mathscr{A}_i / \|\mathscr{A}_i\|_2, \tag{27}$$

and the ith component of Eq. (26) is

$$\hat{\mathscr{A}}_i^{\mathrm{T}} \xi < (b - Ax_K)_i / \|\mathscr{A}_i\|_2 \equiv -\gamma_i \tag{28}$$

where (from Eq. (24))

$$\gamma_i = (A_i^{\mathrm{T}} x_K - b_i) / \sqrt{A_i^{\mathrm{T}} M_K A_i}. \tag{29}$$

This is the long computation that is not included in Khachiyan's iteration; it takes $O(NM^2)$ operations to compute the M components of γ from M_K and A.

At this point, if any element of γ is *one* or more there is no solution within the current ellipsoid ($\xi^{\mathrm{T}}\xi = 1$), and (1) is not solvable. Otherwise, we continue with the "worst violation" by choosing γ_K to be the largest element of γ, with $\mathscr{A}_K^{\mathrm{T}}$ to denote the corresponding row of $\mathscr{A}$. According to prior decisions, $0 < \gamma_K < 1$ and the solution set is now in the region,

$$\xi^{\mathrm{T}}\xi < 1 \quad \text{and} \quad \hat{\mathscr{A}}_K^{\mathrm{T}} \xi < -\gamma_K. \tag{30}$$

In any plane containing $\hat{\mathscr{A}}_K$ the projection of the bounding region is

$$x^2 + y^2 < 1 \quad \text{and} \quad x < -\gamma_K \tag{31}$$

where (temporarily) x is $\hat{\mathscr{A}}_K^{\mathrm{T}} \xi$ and y is the component of ξ in any direction perpendicular to $\mathscr{A}_K$. The projection of the next bounding ellipsoid is an ellipse,

$$(x - c)^2/a^2 + y^2/b^2 = 1, \tag{32}$$

and the ellipse for which ab^{N-1} is minimized, subject to the constraint that it shall contain the points $(x, y) = (-1, 0)$ and $(x, y) = (-\gamma_{K'} \pm (1 - \gamma_K^2)^{1/2}$, is defined by

$$a = \frac{N(1 - \gamma_K)}{N + 1}, \; b^2 = \frac{N^2(1 - \gamma_K^2)}{N^2 - 1}, \; c = -\frac{1 + \gamma_K N}{1 + N}. \tag{33}$$

It follows now that

$$x_{K+1} = x_K + cR\Lambda\hat{\mathscr{A}}_K = x_K - \left(\frac{1 + \gamma_K N}{1 + N}\right) \frac{M_K A_K}{\sqrt{A_K^{\mathrm{T}} M_K A_K}} \tag{34}$$

and

$$\begin{aligned} M_{K+1} &= R^{\mathrm{T}}\Lambda(a^2 \hat{\mathscr{A}}_K \hat{\mathscr{A}}_K^{\mathrm{T}} + b^2(I - \hat{\mathscr{A}}_K \hat{\mathscr{A}}_K^{\mathrm{T}}))\Lambda R \\ &= b^2 M_K + (a^2 - b^2) \frac{(M_K A_K)(A_K^{\mathrm{T}} M_K)}{A_K^{\mathrm{T}} M_K A_K}. \end{aligned} \tag{35}$$

Note that $M_K A_K$ is a column of $M_K A^{\mathrm{T}}$ and $A_K^{\mathrm{T}} M_K$, the corresponding row of AM_K, is its transpose since M_K is symmetric. The second term of Eq. (35) is an outer product of $M_K A_K$ with its transpose, and therefore M_{K+1} is symmetric. This formally completes the iteration, but it has taken $O(NM^2)$ operations.

To reorganize the iteration, we observe that, except for Eq. (35), the matrix M_K

always appears either as a row of AM_K or as a column of $M_K A^{\mathrm{T}}$, and again M_K is symmetric. Thus if we define (and store)

$$\mathscr{M}_K = M_K A^{\mathrm{T}}, \tag{36}$$

with columns $\mathscr{M}_{Kj}$, then the computational steps of the iteration are:

$$\gamma_j = (A_j^{\mathrm{T}} x_K - b_j)/(A_j^{\mathrm{T}} \mathscr{M}_{Kj})^{1/2}, \tag{37}$$

$$x_{K+1} = x_K + c\frac{\mathscr{M}_{KK}}{(A_K^{\mathrm{T}}\mathscr{M}_{KK})^{1/2}} \tag{38}$$

$$\mathscr{M}_{K+1} = b^2\mathscr{M}_K + (a^2 - b^2)\frac{\mathscr{M}_{KK}(A_K^{\mathrm{T}}\mathscr{M}_K)}{A_K^{\mathrm{T}}\mathscr{M}_{KK}}, \tag{39}$$

where $\mathscr{M}_{KK}$ is the column of $\mathscr{M}_K$ for which the corresponding element of γ is largest. When grouped as indicated above, the calculations take $O(NM)$ operations.

4. Augmented problems. Here we address the problem of assigning a larger value to the radius of the initial sphere, so that the smallest simplex in a solution set of (1) is necessarily included. Even though the iteration has success and failure criteria, it is still necessary that the ellipsoids contain a finite part of a potential solution set if a decision is to be made in a finite number of iterations.

To obtain a relatively slightly increased initial radius, consider any nondegenerate vertex defined by $\bar{A}$ and augment the set of inequalities to include

$$-\bar{A}_i^{\mathrm{T}} x < -(\bar{b}_i + 1) \quad \text{if} \quad \bar{b}_i \geq 0, \qquad -\bar{A}_i^{\mathrm{T}} x < -(\bar{b}_i - 1) \quad \text{if} \quad \bar{b}_i \leq 0. \tag{40}$$

The augmented solution set is bounded and has the same bound on the smallest simplex. The best that can be done to bound the most distant vertex without a specification of $\bar{A}$ is to replace b by $1 + |b|$ in the matrices C_j that give bounds on the numerators p_j (Eq. (8). Then

$$R_0 \approx \sqrt{N^{N+1}}\left(\alpha^2 + \frac{2\alpha + 1}{N}\right)^{N/2} < \frac{1}{\alpha}(\alpha\sqrt{N})^{N+1}\exp\left(\frac{2\alpha + 1}{2\alpha^2}\right), \tag{41}$$

where α is the mean value of the magnitudes of As and bs that are chosen randomly in a common interval. Note that as the original radius becomes larger (for fixed N) the augmented radius becomes a smaller multiple of it.

At this point it should also be noted that a degenerate vertex can be bounded by the addition of $2(N - \bar{M})$ inequalities, $x_i < 1$ and $-x_i < -1$. These inequalities have no effect on the computation of R_0.

5. The number of iterations. Now we are able to find bounds and estimates for the number of steps that will be taken before a decision is made. We note that the scale factors Λ^{-1} used in the transformation from $x - x_K$ to ξ (Eq. (24)) reappear as Λ in the expression for M_{K+1} (Eq. (35)). Thus the ratio of volumes within the $(K + 1)$st and Kth ellipsoids is the same in x-space as it is in ξ-space, i.e.

$$V_{K+1}/V_K = ab^{N-1} = \frac{N(1 - \gamma_K)}{N + 1}\left(\frac{N^2(1 - \gamma_K^2)}{N^2 - 1}\right)^{(N-1)/2}. \tag{42}$$

An absolute upper bound on the number of iterations,

$$K < K_M = 2(N+1)\ln(N!\,(\pi R_0^2)^{N/2} Q^{N+1}/(N/2)!), \tag{43}$$

follows from

$$ab^{N-1} < \left(1 - \frac{1}{N+1}\right)\left(1 + \frac{1}{N^2-1}\right)^{(N-1)/2} < e^{-1/2(N+1)}. \tag{44}$$

The leading term of the heuristic estimate of K_M (from Eqs. (41) and (13)) is

$$K_M \sim 2N(N+1)^2 \ln N \quad \text{as} \quad N \to \infty, \tag{45}$$

and the operation count is of $O(N^4 M \ln N)$. Note that α does not appear in Eq. (45): the result is also an upper bound on the leading contribution to K_M in the limit where $N \to \infty$.

The effect on the algorithm of the optimization of the iteration is rather difficult to assess. By experiment it was found that the longest runs of the iteration were characterized by values of γ_K that were distributed, with little scatter, about $1/N$ for almost the entire run. At the very end of such runs, γ_K grows rapidly to $O(1)$ and a decision follows. A rough estimate of the expected number of iterations has been made as follows: first, let M_K be replaced by $R_K^2 I$, where R_K is a measure of the average radius of the Kth ellipsoid. Then Eqs. (29) and (34) are

$$\gamma_i \approx \frac{\hat{A}_i^{\mathrm{T}} x_K}{R_K} - \frac{b_i}{R_K \|A_i\|_2}, \tag{46}$$

$$x_{K+1} \approx x_K - \left(\frac{1 + \gamma_K N}{1 + N}\right) R_K \hat{A}_K, \tag{47}$$

where $\hat{A}_i$ is a unit-vector. Early in the iteration $\|x_K\|_2$ is large; Eq. (47) indicates $O(R_K/N)$. The first term of Eq. (46) is dominant and, for the row of A most nearly in the direction of x_K, $\gamma_K \approx 1/N$ and

$$\begin{aligned} ab^{N-1} &\approx \left(1 - \frac{1}{N}\right)\left(1 - \frac{1}{N^2}\right)^{(N-1)/2} \left(1 - \frac{1}{N+1}\right)\left(1 + \frac{1}{N^2-1}\right)^{(N-1)/2} \\ &\approx e^{-2/N+1} \end{aligned} \tag{48}$$

The optimized iteration runs approximately four times faster than the original until the second term of Eq. (46) becomes comparable with the first, and then both success and failure criteria become significant. With elements of A and b chosen randomly in the same interval, $\|A_i\|_2 \approx \sqrt{N}\,|b_i|$, and it is expected that decisions occur when

$$R_K \approx \sqrt{N}. \tag{49}$$

With these rough estimates, the expected number of iterations is

$$\begin{aligned} K_E &= \frac{N(N+1)}{4} \ln(R_0^2/N) \\ &\sim \frac{N(N+1)^2}{4} \ln N \quad \text{as} \quad N \to \infty. \end{aligned} \tag{50}$$

In examples included in Sec. 6, Eq. (50) has been found to provide slightly high estimates of the numbers of iterations in the longest runs.

6. Implementation and examples. The APL function $\nabla X \leftarrow A$ KCHN $B\nabla$ (Program 1) takes as arguments an M-by-N matrix of integers A and an M-element row of integers B. An N-element row, X, is returned, and X is either a solution of $(A + . \times X) < B$ or it is the last attempt before (1) is found unsolvable. Before starting the iteration the function prints the expected number of iterations (Eq. (50), first line), the maximum number of iterations (Eq. (43)), the asymptotic estimate (Eq. (45)), and the heuristic estimate where the entries of A and $1 + |b|$ are all replaced by mean values of the magnitudes. The user may then enter STOP, GO or 0 to suspend execution at the line labeled STOP, start the iteration at the line labeled GO, or terminate execution.

```
      ⎕CR'KCHN'
X←A KCHN B;M;N;D;B0;J;K;LV;C;RR;K1;A0;AM;G;GK;AK;BB;MK
⍝  (M,N)=ρA, M=ρB, (A+.×X)<B, SOLUTION OR LAST TRY
→((ρB)≠M←(ρA)[1])/ERR
→(M≤N←(ρA)[2])/ERR
X←Nρ0
→(∧/B>J←K←0)/OK
⍝  SMALLEST SIMPLEX, LN(1÷VOL)
LV←(⍟!N)+(N+1)×(+/⍟C[Nρ⍋C←+/A×A])÷2
⍝  INITIAL RADIUS SQUARED
RR←×/C[Nρ⍋C←+/C×C← 0 1 ↓A,B0←1+|D←-B]
BR1:→(N>ρRR←RR,×/C[Nρ⍋C←+/C×C← 0 1 ↓((MρJ←J+1)⌽A),B0])/BR1
RR←+/RR
'EXPECTED VALUE OF K:',⍕⌊(N×N+1)×(⍟RR÷N)÷4
'COMPUTED BOUND:',⍕⌊2×(N+1)×LV+(N×(⍟○RR)÷2)-⍟!N÷2
'ASYMPTOTIC BOUND:',⍕⌊K1←2×N×(N+1)×(N+1)×⍟N
B0←(+/B0)÷M×A0←(+/|,A)÷M×N
K1←K1+(2×N+1)×(⍟!N)+(N×(⍟2)+N×(⍟1-(1-B0×B0)÷N)÷2)+(1+2×N×N)×⍟A0
'HEURISTIC BOUND:',⍕⌊K1
→⍎⍞,0ρ⍞←'ENTER STOP, GO, OR 0: '
⍝  ITERATIVE SEARCH, AM IS TRANSPOSE CURLY M
GO:AM←RR×A
BR2:→(∨/1≤G←D÷(+/AM×A)*÷2)/NO
AK←(1-GK←G[J←G⍳⌈/G])÷1+÷N
BB←(1-GK×GK)÷1-÷N×N
X←X-AM[J;]×(1+N×GK)÷(1+N)×(MK←AM[J;]+.×A[J;])*÷2
→(∧/0>D←(A+.×X)-B)/OK
K←K+1
AM←(BB×AM)-(AM+.×A[J;]×(BB-AK×AK)÷MK)∘.×AM[J;]
→BR2
ERR:'INCORRECT DIMENSIONS'
STOP:→S∆KCHN←STOP
OK:→0,0ρ⎕←'SOLUTION AT K=',⍕K
NO:'NO SOLUTION AT K=',⍕K
```

PROGRAM 1

```
      SETUP 10 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:204
COMPUTED BOUND:1917
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1661
ENTER STOP, GO, OR 0: GO
NO SOLUTION AT K=202
5.43
      SETUP 10 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:200
COMPUTED BOUND:1909
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1686
ENTER STOP, GO, OR 0: GO
SOLUTION AT K=7
0.319
      SETUP 10 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:205
COMPUTED BOUND:1919
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1694
ENTER STOP, GO, OR 0: GO
SOLUTION AT K=18
0.648
      SETUP 10 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:205
COMPUTED BOUND:1909
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1696
ENTER STOP, GO, OR 0: GO
SOLUTION AT K=15
0.53
      SETUP 10 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:208
COMPUTED BOUND:1963
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1724
ENTER STOP, GO, OR 0: GO
SOLUTION AT K=27
0.879
```

```
      SETUP 15 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:209
COMPUTED BOUND:1978
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1694
ENTER STOP, GO, OR 0: GO
NO SOLUTION AT K=170
5.59
      SETUP 15 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:213
COMPUTED BOUND:1988
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1735
ENTER STOP, GO, OR 0: GO
NO SOLUTION AT K=171
5.7
      SETUP 15 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:206
COMPUTED BOUND:1948
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1662
ENTER STOP, GO, OR 0: GO
NO SOLUTION AT K=149
4.97
      SETUP 15 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:219
COMPUTED BOUND:2048
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1805
ENTER STOP, GO, OR 0: GO
NO SOLUTION AT K=170
5.59
      SETUP 15 5
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:211
COMPUTED BOUND:1952
ASYMPTOTIC BOUND:579
HEURISTIC BOUND:1690
ENTER STOP, GO, OR 0: GO
NO SOLUTION AT K=185
6.26
```

EXAMPLES 1

```
      SETUP 20 10
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:1689
COMPUTED BOUND:14704
ASYMPTOTIC BOUND:5572
HEURISTIC BOUND:13415
ENTER STOP, GO, OR 0: GO
SOLUTION AT K=124
6.17
      SETUP 20 10
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:1685
COMPUTED BOUND:14700
ASYMPTOTIC BOUND: 5572
HEURISTIC BOUND:13307
ENTER STOP, GO, OR 0: GO
SOLUTION AT K=104
5.4
      SETUP 20 10
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:1693
COMPUTED BOUND:14803
ASYMPTOTIC BOUND:5572
HEURISTIC BOUND:13507
ENTER STOP, GO, OR 0: GO
NO SOLUTION AT K=1527
72.6
      SETUP 20 10
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:1684
COMPUTED BOUND:14770
ASYMPTOTIC BOUND:5572
HEURISTIC BOUND:13433
ENTER STOP, GO, OR 0: GO
SOLUTION AT K=1602
78
      SETUP 20 10
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:1707
COMPUTED BOUND:14888
ASYMPTOTIC BOUND:5572
HEURISTIC BOUND:11657
ENTER, GO, OR 0: GO
SOLUTION AT K=43
2.29
```

```
      SETUP 30 15
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:5802
COMPUTED BOUND:49459
ASYMPTOTIC BOUND:20797
HEURISTIC BOUND:45859
ENTER STOP. GO, OR 0: 0
0.444

      SETUP 40 20
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:14118
COMPUTED BOUND:118194
ASYMPTOTIC BOUND:52844
HEURISTIC BOUND:110548
ENTER STOP, GO, OR 0: 0
0.799


      SETUP 40 20
      TIME'X←A KCHN B'
EXPECTED VALUE OF K:14119
COMPUTED BOUND:118690
ASYMPTOTIC BOUND:52844
HEURISTIC BOUND:111246
ENTER STOP, GO, OR 0: STOP
KCHN[30]
      RR
6.09E59
      →0
0.792
```

EXAMPLES 2

The auxiliary function ∇ SETUP $MN\nabla$ (Examples 1, 2) assigns global variables A and B with $M = MN[1]$ and $N = MN[2]$ and with entries randomly chosen in the interval $[-10, 10]$. The auxiliary function $\nabla T \leftarrow$ TIME $E\nabla$ (Examples 1, 2) executes the character argument E and returns elapsed CPU time in seconds. The machine is the IBM 370/158 at Brown University.

References

[1] L. G. Khachiyan, *A polynomial algorithm for linear programming*, Doklady Akad. Nauk U.S.S.R. **244**, 1093-1096 (1979)

[2] Translation of [1], Soviet Math. Doklady **20**, 191-194 (1979)

[3] P. Gacs and L. Lovasz, *Khachian's algorithm for linear programming*, Computer Science Dept., Stanford University, 1979