On relaxation methods for systems of linear inequalities

Jan TELGEN *
Center for Operations Research and Econometrics (CORE), 1348 Louvain-la-Neuve, Belgium

Received June 1980
Revised May 1981

In their classical papers Agmon and Motzkin and Schoenberg introduced a relaxation method to find a feasible solution for a system of linear inequalities. So far the method was believed to require infinitely many iterations on some problem instances since it could (depending on the dimension of the set of feasible solutions) converge asymptotically to a feasible solution, if one exists. Hence it could not be used to determine infeasibility.

Using two lemma's basic to Khachian's polynomially bounded algorithm we can show that the relaxation method is finite in all cases and thus can handle infeasible systems as well. In spite of more refined stopping criteria the worst case behaviour of the relaxation method is not polynomially bounded as examplified by a class of problems constructed here.

1. Introduction

In their classical papers Agmon [1] and Motzkin and Schoenberg [17] introduced a relaxation method to determine the feasibility of a system of linear inequalities. The method is called 'relaxation' method because the constraints are considered one at a time. The same principle is used in the ellipsoidal method developed by Khachian [13].

Khachian's method excited an unprecedented interest among mathematicians and computer scientists. Even in the worst case it requires only a polynomially bounded number of steps, whereas all other methods known thus far require an exponential or superexponential number of steps [11,14,22]. Within the theory of computational complexity (see e.g. [3,9]), Khachian's algorithm provided the missing link to establish membership in the class $\mathcal{P}$ (of polynomially solvable, i.e. 'easy' problems) for linear programming and a whole list of LP-equivalent problems [20,21].

Another important side effect of Khachian's algorithm is the explicit use of number theoretic arguments in this area of mathematical programming. This paper illustrates that effect: using two lemmas of a number theoretic nature (also used by Khachian [13]) we can prove finiteness for the relaxation method. So far the theoretical results on the performance of the relaxation method were rather disappointing: if the system is feasible, the method might terminate in a feasible solution or converge asymptotically towards a feasible solution (depending on the dimension of the set of feasible solutions). If the system is infeasible the relaxation method could not detect that fact.

After describing the relaxation method and some variants in Section 2, we prove the finiteness of the method in Section 3. As a consequence of its finiteness, the relaxation method can also determine infeasibility of the system. However, as shown in Section 5, none of the variants of the relaxation method is polynomially bounded: a class of problem instances is constructed on which the relaxation method requires an exponential number of iterations. We conclude with a brief discussion of the consequences of our result, and with a possible extension of the relaxation method.

2. The relaxation method

Consider the system of linear inequalities

$$Ax \leqslant b \tag{2.1}$$

where A is $m \times n$, x is $n \times 1$ and b is $m \times 1$. Denote the rows of A as A_i^{T}. We assume that $A_i^{\mathrm{T}} A_i = 1$ $(i = 1, \dots, m)$. Agmon [1] and Motzkin and Schoenberg [17] proposed to find a feasible solution to the system (2.1) by the *relaxation method*:

* Present address: Department of Applied Mathematics, Rabobank Nederland, 3705 AR Zeist, Netherlands.

The research reported here was done while the author was visiting the Management Science Program, University of Tennessee as a NATO Science Fellow for the Netherlands Organization for the Advancement of Pure Research (ZWO). Comments by the referee were most helpful in preparing the revised edition.

North-Holland Publishing Company
European Journal of Operational Research 9 (1982) 184–189

Initialize: Set $x^0 = 0 \in \mathbb{R}^n$ and $k = 0$; select $0 < \lambda < 2$.

Step 1: If x^k is feasible, stop; the system is feasible.

Step 2: Select a violated constraint $A_l^T x^k > b_l$.

Step 3: Determine

$$x^{k+1} = x^k - \lambda(A_l^T x^k - b_l)A_l, \qquad (2.2)$$

Set $k = k + 1$ and continue with Step 1.

For this relaxation method Agmon [1] and Motzkin and Schoenberg [17] proved:

Theorem 2.1. *If the system* (2.1) *is feasible, the relaxation method yields either*

(i) *termination in a feasible solution*; *or*

(ii) *asymptotical convergence of* x^k *to a feasible solution.*

It is useful to note that if the system (2.1) is infeasible, this is not detected by the relaxation method, since generally it is impossible to distinguish between case (ii) above and infeasibility.

Various modifications have been proposed on this basic scheme. The choice of a particular violated constraint in Step 2 is often replaced by the selection of the most violated constraint, i.e., such that $A_i^T x^k - b_i$ is maximal. Especially in Russian literature these possibilities have been discussed extensively (e.g. [4,6,7,9]).

Eckhardt [5] showed that a variant of the relaxation method terminates in a finite number of steps if there is a solution to $Ax \leqslant b - \epsilon$, with $\epsilon > 0$, $\epsilon \in \mathbb{R}^m$.

Jeroslow [12] considered the existence of a solution that satisfies all constraints up to some constant ϵ. He showed that the relaxation method with $\lambda = 1$ finds such a solution in at most D^2/ϵ^2 steps, where D is the distance from the starting point to a feasible solution, if one exists.

Maurras [16] proposed to replace (2.2) by

$$x^{x+1} = x^k - [A_l^T x^k - b_l + e]A_l. \qquad (2.3)$$

He showed that this variant of the relaxation method could determine in $2D^2/\epsilon^2$ steps the nonexistence of a point in the feasible region such that it satisfies all constraints with a slack of at least ϵ.

In the next section we show that the relaxation method is capable of more. Using two number theoretic lemmas basic to Khachian's ellipsoidal algorithm it turns out that the original relaxatio method is finite for all feasible systems. Moreove the maximum number of iterations can be de termined in advance. Hence the relaxation metho can be used to determine the (in)feasibility of th system (2.1).

3. Finiteness

Define the maximum violation in the point x a

$$\theta(x) = \max_i \{A_i^T x - b_i\} = A_l^T x - b_l \qquad (3.1$$

and the length of the binary encoding of all problem data as

$$L = \sum_i \sum_j \log(|a_{ij}| + 1) + \sum_i \log(|b_i| + 1) + \log nm + 2 \qquad (3.2$$

In the latter definition a_{ij} and b_i are required to b integers (see e.g. [2,8]); this is no real restrictior even if we want $A_i^T A_i = 1$ for all i. We simpl determine L first and then rescale the problem.

Two lemmas of number theoretic nature basi to Khachian's ellipsoidal algorithm [13] are:

Lemma 3.1. *If the system* (2.1) *is feasible, then ther is a feasible solution* $\hat{x}$ *with* $|\hat{x}_j| \leqslant 2^L/2n$, $j = 1,\ldots,n$.

Lemma 3.2. *If the system* (2.1) *is infeasible, then fo all* $x \in \mathbb{R}^n$: $\theta(x) \geqslant 2 \cdot 2^{-L}$.

For proofs of these lemmas, see e.g. [2,8,18].

Khachian [13] constructs a series of *ellipsoids* with decreasing volume, which contain a feasible solution, if one exists. In a similar way the relaxation method may be interpreted as implicitly constructing a series of *hyperspheres* with decreasing volume. As an example, consider Fig. 1.

Denote a hypersphere with center x^k and radius r_k as C_k. Consider C_{k+1} with

$$x^{k+1} = x^k - \lambda \cdot \theta(x^k) \cdot A_l \qquad (3.3)$$

as usual, and

$$r_{k+1}^2 = r_k^2 - \{\theta(x^k)\}^2 \lambda(2 - \lambda). \qquad (3.4)$$

According to Lemma 3.1 the radius r_0 of the initial

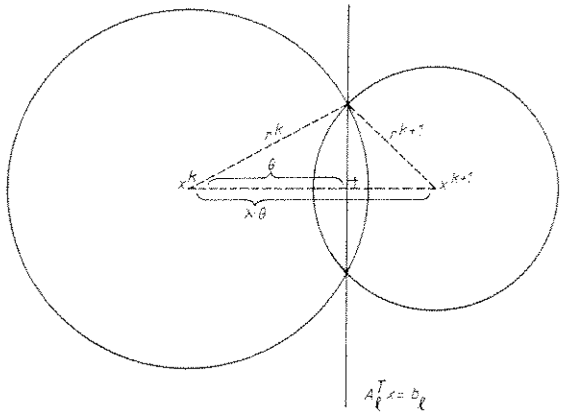

Fig. 1.

hypersphere C_0 need not be larger than $2^{L-1} \cdot n^{-1/2}$.

Now we prove that every possible feasible solution in C_k is also contained in C_{L+1}. Denote

$$\overline{C}_k = \{x \in \mathbb{R}^n \mid A_l^T x \leq b_l\} \cap C_k \tag{3.5}$$

then we have to prove:

Lemma 3.3. $\overline{C}_k \subset C_{k+1}$.

Proof. We need to prove that $x \in \overline{C}_k$ implies

$$(x^{k+1} - x)^T (x^{k+1} - x) - r_{k+1}^2 \leq 0.$$

Substituting (3.2) and (3.3) yields for the left-hand side

$$(x^k - x)^T (x^k - x) - 2\lambda\theta(x^k) A_l^T (x^k - x)$$
$$+ \lambda^2 (A_l^T x^k - b_l)^2 - r_k^2 + \{\theta(x^k)\}^2 \lambda(2 - \lambda).$$

For $x \in C_k$ we have $(x^k - x)^T (x^k - x) \leq r_k^2$ and $A_l^T x \leq b_1$. Therefore the expression above is smaller than

$$-2\lambda\theta(x^k)(A_l^T x^k - b_l) + \lambda^2 (A_l^T x^k - b_l)^2$$
$$+ 2\lambda\{\theta(x^k)\}^2 - \lambda^2 \{\theta(x^k)\}^2 = 0.$$

At this point we can prove the finiteness of the relaxation method:

Theorem 3.4. *Either the relaxation method detects feasibility of the system (2.1) within* $k = \lceil 2^{4L}/n\lambda(2 - \lambda) \rceil$ *iterations or the system is infeasible*

Proof. If the system (2.1) is feasible, there is a feasible solution within C_0 (Lemma 3.1). If there is a feasible solution within C_k, then there is a feasible solution within C_{k+1} (Lemma 3.3). In each iteration the square of the radius of C_k is reduced by at least $\{\theta(x^k)\}^2 \lambda(2 - \lambda)$. But $\forall k$, $\theta(x^k) > 2 \cdot 2^{-L}$, otherwise the system is feasible (Lemma 3.2), thus $r_{k+1}^2 < r_k^2 - 4 \cdot 2^{-2L} \lambda(2 - \lambda)$. Using Lemma 3.2 the feasibility of the system can be decided upon if $r_k < 2 \cdot 2^{-L}$, which implies that the maximal value of k can be determined from:

$$2^{2L-2}/n - k \cdot 4 \cdot 2^{-2L} \lambda(2 - \lambda) = 4 \cdot 2^{-2L}$$

thus

$$k = \frac{2^{2L-2}/n - 4 \cdot 2^{-2L}}{4 \cdot 2^{-2L} \lambda(2 - \lambda)} \leq \frac{2^{4L-4}}{n\lambda(2 - \lambda)}.$$

4. Improving the relaxation method

Apart from proving finiteness of the relaxation methods, the lemmas may be used to construct stopping criteria as well. Indeed we have the following criteria:

(a) if $\theta(x^k) < 2 \cdot 2^{-L}$, then the system is feasible (Lemma 3.2). This criterion of course subsumes the case that x^k is feasible.

(b) if $r_{k+1}^2 \leq 4 \cdot 2^{-2L}$, and (a) does not hold, then the system is infeasible (Lemmas 3.2 and 3.3).

(c) if $k \geq \lceil 2^{4L}/n\lambda(2 - \lambda) \rceil$ and (a) does not hold, then the system is infeasible (Theorem 3.4).

The application of these three stopping rules requires determining $\theta(x^k)$ and updating r_k^2. Depending on the specific variant of the relaxation method used, the cost of these operations ranges from 1 to $O(mn)$ additional calculations per iteration.

Since the determination of $\theta(x^k)$ is the most expensive computation it might be wise to avoid that. It is quite possible to replace $\theta(x^k)$ by any $A_i^T x - b_i > 0$, and still have a converging algorithm. Stopping criterion (b) can still be used, whereas (c) can only be used if we restrict ourselves to constraints with $A_i^T x - b_i > 2 \cdot 2^{-L}$. Thus this variant is finite only in the latter case.

It is easily seen that the convergence of the algorithm is fastest if the reduction in the radius r_k^2 is maximized, i.e., if $\lambda = 1$ and $\theta(x^k)$ is determined as in (3.1). If the latter is considered too expensive a viable alternative might be to compute $\theta(x^k)$ at certain points only (say every 10 iterations).

Note that criterion (b) will generally apply long before criterion (c) applies, i.e., the maximum number of iterations will not be required even on infeasible problems. As an example consider the system

$$x_1 \leqslant 0, \qquad x_2 \geqslant 4$$

with $x^0 = 0 \in \mathbb{R}^2$ and $n = 2$. Since $L = 4$ and $r_0^2 = 2^5$ in the worst case 2^{15} iterations are required, whereas criterion (b) terminates application of the method after 2 iterations.

5. Worst-case behaviour

The upper bound on the number of iterations in the relaxation method (Theorem 3.4) is not sufficient to show that the method is polynomially bounded. In fact we can prove the contrary by constructing a class of problems on which the relaxation method requires an exponential number of iterations. We do this for the case $\lambda = 1$.

Consider the system

$$\begin{aligned} &-x_1 + 2^\alpha \cdot x_2 \leqslant -2^\alpha, \\ &-x_2 \leqslant 0 \end{aligned} \tag{5.1}$$

sketched in Fig. 2.

From (3.2) we have $L = 2\alpha + 4 = \Omega(\alpha)$. Let $x^0 = (0,0)^T$ then it is easily calculated that

$$x^2 = \left(\frac{2^\alpha}{2^{2\alpha}+1}, 0\right)^T$$

and generally for k even:

$$x^{k+2} = \left(x^k + \frac{2^\alpha - x_1^k}{2^{2\alpha}+1}, 0\right)^T.$$

Obviously the relaxation method applied to this problem will stop on criterion (a). From Lemma 3.2 it is easily checked that a necessary condition for criterion (a) to hold is: $x_1^k > 2^\alpha - 1$. But since the x_1-coordinates of the iteration points increase by at most $2^\alpha/2^{2\alpha}+1$ per 2 iterations, the required number of iterations is at least N, with

$$2^\alpha - 1 = \frac{1}{2} N \cdot \frac{2^\alpha}{2^{2\alpha}+1}$$

thus

$$N = 2\frac{2^\alpha - 1}{2^\alpha}(2^{2\alpha}+1) \geqslant 2^{2\alpha} = \Omega(2^L).$$

Thus by increasing α (or L) we can construct problem instances on which the relaxation method needs at least an exponential number of iterations.

It is relatively simple to extend this analysis to the general case; basically the same instance requires an exponential number of iterations for all $0 < \lambda \leqslant 2$.

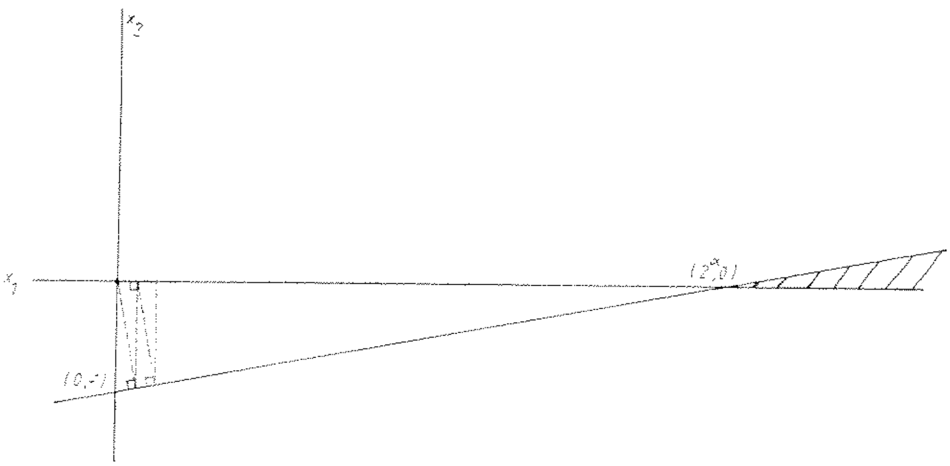

Fig. 2.

6. Conclusion

The relaxation method has some nice properties. First of all the calculations are very simple and the method is easily implemented in any of its variants. There are no data transformations and the original problem sparsity is maintained. Moreover, the method can make extensive use of the sparsity inherent to practical problems. There is no problem with numerical accuracy as no 'old' information is used in the iterations. Furthermore, only two pieces of information ($\theta(x^k)$ and r_k^2) need to be stored. Also a known approximate solution can be used in a simple way.

Now we have added to this list the finiteness of the relaxation method, let us reconsider its relative advantages. For the problem of identifying a feasible solution to a system of linear inequalities three main tools are available: the simplex method, the ellipsoidal method and the relaxation method. Theoretically the ellipsoidal method is the only 'good' (polynomial) one; in practice, however, the average case performance is much more important than the worst case behaviour and as such the simplex method is superior. So the relaxation method would be useful for problems where its strong points are essential, i.e., in very large problems. If a problem is too big for the simplex method, it can still be solved by the relaxation method. An example of a practical problem (oil refinery scheduling) for which this was the case is given by Hendrikse-Baardman [10] (see also [15]).

Other possible applications of the relaxation method include hybrids with the ellipsoidal method, to decrease the number of calculations in the latter method. In this hybrid method especially in the first few iterations the relaxation method might be used to reduce the size of the original hypersphere with simple operations, using the sparsity of the problem and existing a priori information. Then at a certain point in time Khachian's method could take over exploiting its polynomial boundedness. If the switching point is a function of k or $\theta(x^k)$ (e.g. $k=100$ or as soon as $\theta(x^k)$ is smaller than some fixed amount) the hybrid method is still polynomial. If the switching point depends on r_k the hybrid method is not necessarily polynomially bounded.

Finally we should mention an intermediate method between the ellipsoidal and the relaxation methods. Considering hyperspheres as a restricted class of ellipsoids, we may note that a less restricted class is given by non-titled ellipsoids, i.e., ellipsoids of which the axes are parallel to the coordinate axes. These non-titled ellipsoids are described mathematically as

$$T=\left\{x\in\mathbb{R}^n \mid (x-x^0)^{\mathrm{T}}D(x-x^0)\leqslant 1\right\}$$

where x^0 is the center and D is a diagonal matrix. Then a method similar to the ellipsoidal (and relaxation) method could be given using these non-titled ellipsoids. One interesting property of such a method is the fact that for A_i is a unit vector the method degenerates into the (polynomially bounded) ellipsoidal method, whereas for $A_i=(1,1,\ldots,1)^{\mathrm{T}}$ the method is equivalent to the (not polynomially bounded) relaxation method. In other words, the performance of this method depends on the sparsity of the problem.

References

[1] S. Agmon, The relaxation method for linear inequalities, Canad. J. Math. 6 (1954) 382–392.

[2] B. Aspvall and R.E. Stone, Khachian's linear programming algorithm, Report STAN-CS-79-76, Department of Computer Science, Stanford University (1979).

[3] A.V. Aho, J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Computer Algorithms (Addison-Wesley, Reading, MA, 1974).

[4] L.M. Bregmann, The relaxation method of finding the common point of convex sets and its application to the solution of problems, U.S.S.R. Comput. Math. Math. Phys. 4 (1964) 151–158.

[5] U. Eckhardt, Iterative Lösung linearer Ungleichungssysteme, Dissertation, RWTH Aachen (1972).

[6] I.I. Eremin, The relaxation method of solving systems of inequalities with convex functions on the left sides, Dokl. Akad. Nauk S.S.S.R. 160 (1965) 219–222.

[7] D.K. Fadeev and W.N. Fadeeva, Numerische Methoden der linearen Algebra (München, 1964).

[8] P. Gacs and L. Lovasz, Khachiyan's algorithm for linear programming, Report STAN-CS-79-750, Department of Computer Science, Stanford University (1979).

[9] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness (Freeman, San Francisco, 1979).

[10] M. Hendrikse-Baardman, Alternatives for the simplex method (dutch), Master thesis, Econometric Institute, Erasmus University, Rotterdam (1980).

[11] R.G. Jeroslow, The simplex algorithm with the pivot rule of maximising criterion improvement, Discrete Math. 4 (1973), 367–377.

[12] R.G. Jeroslow, Some relaxation methods for linear inequalities, Report No. 366R, G.S.I.A., Carnegie-Mellon University (1975).

[13] L.G. Khachian, A polynomial algorithm in linear programming, Dokl. Akad. Nauk S.S.S.R. 244 (1979) 191–194.

[14] V. Klee and G.J. Minty, How good is the simplex method? in: O. Shisha Ed., Inequalities III (Academic Press, New York, 1972).

[15] G.S. Kumar, An approach to linear programming using relaxation methods, Report, Department of Management, Northern Illinois University (1980).

[16] J.F. Maurras, Good algorithms: old ideas, unpublished research paper, Electricité de France (1978).

[17] T.S. Motzkin and I.J. Schoenberg, The relaxation method for linear inequalities, Canad. J. Math. 6 (1954) 393–404.

[18] M.W. Padberg and M.R. Rao, The Russian method for linear inequalities, Report 79-124, College of Business Administration, New York University (1979).

[19] B.T. Polyak, Gradient methods for solving equations a inequalities, U.S.S.R. Comput. Math. Math. Phys. 4 (19 17–32.

[20] A.H.G. Rinnooy Kan and J. Telgen, The complexity linear programming, Statist. Neerlandica 35 (1981) 91–1

[21] J. Telgen, Redundancy and Linear Programs (Mathem cal Centre, Amsterdam, 1979); revised edition: (1981).

[22] N. Zadeh, A bad network problem for the simplex meth and other minimum cost flow algorithms, Math. Progra ming 5 (1973) 255–266.