Cs. FABIAN
SIGRID MOGA
Department of Economic Cybernetics
Academy of Economic Studies

Two Simulation Procedures for Solving Convex Integer Programming Problems

The authors of this paper describe two simulation procedures ("controlled simulation" and "semicontrolled simulation") with a view to solving convex integer programming problems.

The numerical experiments carried out with classical examples of Lawler and Bell test [4] are meant to illustrate the efficiency of the procedures described in the paper.

INTRODUCTION

Nonlinear integer programming is an attractive and rather little dealt with field of mathematical programming owing to the difficulties which arise when solving both integer programming problems (see, as an example, [1] and nonlinear programming problems (see, as an example, [2]).

Since operative production programming sometimes leads to highly complex problems to be solved by means of nonlinear integer programming, the research activity in this field and solving such problems have become still more necessary.

As far as precise methods are concerned, the former method is of a cutting plane type and is due to Kelley [3] for convex-function problems.

For monotonous functions we know Lawler and Bell's lexicographic method [4], whereas Korte, Krelle and Oberhofer [5] have given a lexicographic method combined with a branch-and-bound method.

We shall also mention the method used by Hammer and Rudeanu [6] for pseudoboolean programming, as well as Kelley's modified method and Korte-Krelle-Oberhofer's method successfully amended by Fabian in [7]. The reffered to methods are suitable for special problems and usually yield good results in small-size problems [1]. The simulation method offers another

possibility of approaching nonlinear integer problems. It is particularly advisable for:

— simulation methods and models classification [8];

— linear programming problems [9];

— a very small class of nonlinear problems (separable functions and with one constraint on the sum of variables) [10].

DEFINING THE PROBLEM

Consider the following problem:

$$\begin{aligned} z &= \min f_0(x) \\ f_i(x) &\leqslant 0, \quad i = 1, \ldots, m \\ 0 &\leqslant x \leqslant d, \quad x - \text{integer}, \end{aligned} \tag{1}$$

where $f_i(x)$, $i = 0, \ldots, m$ are convex functions, differentiable by partia derivatives bounded in the hiperparallelepiped as defined by

$$0 \leqslant x \leqslant d \quad \text{with } x = (x_1, \ldots, x_n) \text{ and } d = (d_1, \ldots, d_n).$$

We write

$$G = \{x \mid f_i(x) \leqslant 0, \ i = 1, \ldots, m, \ 0 \leqslant x \leqslant d\} \tag{2}$$

for the domain of feasible points of the problem (1), which is obviously bounded.

CONTROLLED AND SEMICONTROLLED SIMULATION PROCEDURES

Consider first, this problem:

$$\begin{aligned} &min\, f(x) \\ &0 \leqslant x \leqslant d. \end{aligned} \tag{3}$$

We know [2] that if function $f(x)$ is differentiable in x and y is a direction so that $\nabla f(x)\, y < 0$, then there is a λ so that for any μ, $0 < \mu \leqslant \lambda$ we shall have

$$f(x + \mu y) < f(x).$$

Had we the certainty of the existence of a $\lambda \geqslant 1$ having the aforesaid property, the previous result could be extended for integer x, y, μ.

It is obvious that direction y must have opposed sign components as against the components corresponding to the gradient in x in order to achieve inequality $\nabla f(x)y < 0$.

There arises the problem of determining a direction y not only as regards the sign but also as regards the size of vector y components. For this purpose we apply a currently used procedure [11] for generating the observed values

which a random variable with arbitrary distribution function will assume by means of random variables uniformly distributed in the interval [0, 1].

We write

$$\nabla f(\tilde{x}) = (g_1(\tilde{x}), \ldots, g_n(\tilde{x}))$$

and

$$h_j(\tilde{x}) = \frac{|g_i(\tilde{x})|}{\sum_{i=1}^{n} |g_j(\tilde{x})|}, \quad j = 1, \ldots, n. \tag{4}$$

Let $\xi_1, \ldots, \xi_N$, N be random numbers uniformly distributed in [0, 1] and $y_0 = (0, \ldots, 0)$. If

$$\sum_{j=1}^{k-1} h_j(\tilde{x}) \leqslant \xi_i < \sum_{j=1}^{k} h_j(\tilde{x}) \tag{5}$$

then we write

$$y_k = y_k + \delta_k,$$

where

$$\delta_k = \begin{cases} 1, & g_k(\tilde{x}) < 0 \\ -1, & g_k(\tilde{x}) \geqslant 0 \end{cases}$$

for $i = 1, \ldots, N$.

This is the way of passing (by simulation) from integer $\tilde{x}$ to $\tilde{x} + \mu y$, also integer on a direction y, in integer components. Following several generations, the mean of the achieved directions will be found on the direction of the gradient.

Because there are also directions more efficient than the gradient direction (Fletcher-Reeves, Davidon-Fletcher-Powell, etc. [2]) for finding the optimal nonlinear programming, we can use these directions as well in a simulation procedure.

In order to solve problem (1) we consider $\tilde{x} \in G$ to be an original point or achieved while applying the procedure by making use of $\nabla f_0(\tilde{x})$. By means of formulas (4) and (5) we find a direction vector y to satisfy the constraints $0 \leqslant \tilde{x} + \mu y \leqslant d$, for instance $\mu = 1$, and we continue the procedure.

The original point $\tilde{x}$ may be obtained also by solving a linear programming problem of the form:

$$min \left\{ x_{n+1} \left| \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \in P^0, \; 0 \leqslant x \leqslant d, \text{ integer } x \right\}\right.$$

where

$$P^\circ = \left\{ \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \middle| \; x \nabla f_j(x^\circ) - \alpha_j x_{n+1} \leqslant x^\circ \nabla f_j(x^\circ) - f_j(x^\circ), \; j = 0, \ldots, m \right\}$$

having

$$\alpha_j = \begin{cases} 1, & j = 0 \\ 0, & j = 1, \ldots, m \end{cases}$$

and x° is an absolutely arbitrary point of $0 \leqslant x \leqslant d$.

When obtaining each admissible point $\tilde{x}$ we shall consider

$$G := G \cap \{x | f_0(x) - f_0(\tilde{x}) \leqslant 0\}. \tag{6}$$

If the original point or a current point $\tilde{x} \notin G$, then we determine index j (corresponding to the least satisfied constraint)

$$f_j(\tilde{x}) = \max_{i = 0, \ldots, m} f_i(\tilde{x}) \tag{7}$$

Making use of $\wedge f_j(\tilde{x})$ and relationships (4) and (5) we mean to enter the domain as quickly as possible by simulating directions according to the gradient of the least satisfied constraint; this is a reason to follow the "controlled" procedure. In order to be able to successfully use this "controlled" procedure, $f_i(x)$ $i = 1, \ldots, m$ convex functions have been assumed.

If, in order to go back to domain G a pure simulation is used, having $h_j = \frac{1}{n}$, $j = 1, \ldots, n$ and δ_k assuming values $+1$ and -1 with a 0.5 probability, we define the procedure as "semicontrolled". In such a case, a number of y components is modified N times by ± 1 thus obtaining point $\bar{x}$.

Further, point $\tilde{x}$ or $\bar{x}$ is kept, as

$$\min_{i = 0, \ldots, m} (\max f_i(\bar{x}), \max_{i = 0, \ldots, m} f_i(\bar{x})) \tag{8}$$

is reached in $\tilde{x}$ or $\bar{x}$, respectively.

As a criterion for stopping the simulation procedure was used condition

$$(LP = NP) \vee (LS = NS)$$

where

LP is the number of times for which an admissible solution remains unaltered;
LS is the number of times for which an inadmissible solution remains inadmissible;

NP and NS are imposed values for LP and LS.

If LP becomes large, we can assume that the optimal solution has been reached, and if LS becomes large the problem has no solution or the field has become very small in agreement with relationship (6). In this latter case, the last admissible point is probably the optimal one. The procedures can also be formulated through the language of the Markov chains. Other "controlled" simulation procedures are used to set up cutting networks [12] or to build up synthesis functions and carry out several efficiency functions programming [13].

The described procedures can be extended also to nonconvex constrained problems. Experiments in this respect are underway and the achieved results will be dwelt on in another paper.

EXPERIMENTAL RESULTS

For carrying out some numerical experiments by means of the described procedures two programs have been worked out: SINEL-SD (Semicontrolled Nonlinear Simulation) and SINEL-D (Controlled Nonlinear Simulation), both of them using the FORTRAN IV language. The runnings were carried out on the IBM-370/135 computer at the Department of Economic Cybernetics Laboratories.

The experiments were carried out with Lawler and Bell [4] test examples from the literature, having values $N = 1$, $N = \frac{n}{2}$ and $N = n$, $NP = 15$, $NS = 100$.

a) COMPARING VARIANTS SINEL-SD AND SINEL-D

An admissible point with components close to $d_j/2$, $j = 1, ..., m$ was chosen as the original point. The tested problems were those suggested by Lawler and Bell [4], after eliminating the two nonconvex constraints. The results of this experiment are given in Table 1. The optimal solution was achieved in most cases with variant SINEL-D in a shorter time.

b) COMPARING THE ACHIEVED RESULTS WITH DIFFERENT ORIGINAL POINTS

These experiments were carried out with SINEL-D program with Lawler and Bell test examples [4] in the original point (0, ...,0) which is not admissible and with an admissible original point around the point of coordinates $d_j/2$, $j = 1, ...,m$. The results are given in Table II. We have drawn the conclusion that the original point has no importance in achieving the optimum, since it only concerns the time necessary to obtain these results.

The number of generations during the process can eventually be modified.

In the beginning it should be advisable to have N larger values, and smaller values towards the end, as experiments prove so far.

c) COMPARING SINEL-D WITH OTHER METHODS

Lawler and Bell test problems [4] were solved by means of original Kelley programs (KEOR), Kelley modified and combined with Korte, Kelle, Oberhofer modified (KEKO) [7] and compared with SINEL-D. The results are given in Table III.

It is a general conclusion that SINEL-D attains the optimum in a shorter time than the other methods considered.

Table I

Comparison of variants SINEL-SD and SINEL-D

Number	Problem	SINEL – SD						SINEL – D					
		NG = 7		NG = 3		NG = 1		NG = 7		NG = 3		NG = 1	
		Z	T	Z	T	Z	T	Z	T	Z	T	Z	T
1	L–B–1	11.	3.55	8.	16.3	11.	6.33	14.	7.74	15.	3.22	8.	3.26
2	L–B–2	0.	0.30	0.	0.52	0.	1.4	0.	0.30	0.	0.52	0.	0.62
3	L–B–3	10.	0.06	9.	0.29	8.	5.9	8.	1.32	8.	4.07	8.	2.52
4	L–B–4	24.	18.32	40.	0.98	24.	3.92	36.	8.06	24.	2.26	24.	5.0
5	L–B–5	19.	6.5	16.	7.97	18.	1.67	18.	14.85	15.	6.8	18.	2.00

Table II

Results achieved with SINEL-D. Original different points

Number	Problem	$X_j^0 = d_j/2$						$X_j^0 = 0$					
		NG = 7		NG = 3		NG = 1		NG = 7		NG = 3		NG = 1	
		Z	T	Z	T	Z	T	Z	T	Z	Z	T	T
1	L–B–1	14	14.6	15	3.22	8	3.26	14	7.73	11	3.3	8	3.4
2	L–B–2	0	0.30	0	0.52	0	0.81	0	1,25	0	0.56	0	4,78
3	L–B–3	8	1.3	8	4.07	8	3.82	8	7.67	8	1.32	8	1.07
4	L–B–4	36	8.06	24	2.26	24	7.67	24	18.08	42	14.32	46	7.9
5	L–B–5	18	14.85	15	6.8	17	4.40	19	3.78	16	1.3	18	1.07
6	L–B–7	12	26.55	12	11.45	24	0.74	18	3.73	12	20.81	20	8.22
7	L–B–8	52	0.00	52	0.00	50	8.42	78	5.70	52	16.87	50	10.37
8	L–B–9	14		14	0,14	14	4,03	14	22.24	14	16.08	14	4.49
9	L–B–10	0		0	0,13	0	0,14	0	2.04	0	7.49	0	4.96

Table III

Comparison of SINEL- D with other methods

Number	Problem	KO		KEKO		SINEL-D	
		Z	T	Z	T	Z	T
1	L—B—1	7.44	5.07	8	2.64	8	3.26
2	L—B—2	0.00	2.57	0	0.79	0	0.50
3	L—B—3	7.33	2.22	8	1.64	8	1.32
4	L—B—4	16.00	2.62	24	1.38	24	2.26
5	L—B—5	13.66	67.42	15	21.88	15	6.8
7	L—B—7	9.47	3.53	12	66.57	12	11.45
8	L—B—8	39.70	10.18	50	523.23	50	8.42
9	L—B—9	12.65	3.03	14	258.89	14	4.94
10	L—B—10	0.0	3.82	0	181.36	0	8.04

REFERENCES

1. GARFINKEL S. R., NEMHAUSER L. G.: *Integer Programming.* John Wiley & Sons Inc., New York — London — Sydney — Toronto, 1972.
2. DUMITRU V.: *Programare neliniară, algoritmi, programe, rezultate numerice.* Editura Academiei Republicii Socialiste Romania, Bucureşti, 1975.
3. KELLEY I. E.: *The Cutting Plane Method for Solving Convex Programs.* J. Soc. Indust. Appl. Math. Vol. 8, 4/1960, pp. 703—713.
4. LAWLER E. L., BELL M. D.: *A Method for Solving Discrete Optimization Problems.* Op.Res. Vol. 14, 1967, pp. 1098—1112.
5. KORTE B., KRELLE W., OBERHOFER W.: *Ein lexikographischer Suchalgorithmus zur Lösung allgemeiner ganzzahliger Programierungsaufgaben.* Unternehmensforschung Vol. 13, Teil I, Teil II, 1969, Nachtrag Vol. 14, 1970.
6. HAMMER P. L., RUDEANU S.: *Boolean Methods in Operations Research.* Springer-Verlag, Berlin — Heidelberg — New York, 1967.
7. FABIAN CS.: *Some Algorithms for Nonlinear Integer Programming.* Economic Computation and Economic Cybernetics Studies and Research, No. 3, 1976.
8. COSTACHE N., STOICA M.: *Clasificarea modelelor şi metodelor de simulare.* Studii şi cercetări de calcul economic şi cibernetică economică, No.3, 1973.
9. HOFSTEDT K., PIEHLER J.: *Ein Simulationsverfahren zur Lösung ganzzahliger linearer Optimalprobleme.* Math. Operationsforschung und Statistik, Vol. 4, No2, 1973, pp. 99—105.
10. ICHIMURA M., WAKIMOTO K.: *Adaptive Monte Carlo Method for Solving Constrained Minimization Problems in Integer Nonlinear Programming. Control Theory, Numerical Methods and Computer Systems Modelling.* Lecture Notes in Economics and Mathematical Systems, Vol. 107, Springer-Verlag, Berlin—Heidelberg—New York, 1975.
11. LARSON HAROLD I.: *Introduction to the Theory of Statistics.* John Wiley & Sons Inc., New York—London—Toronto, 1973.
12. FABIAN Cs. and others: *Consideraţii asupra problemelor de croire* (II). Studii şi cercetări de calcul economic şi cibernetică economică, No1, 1972.
13. CIOBANU GH.: *Aplicarea simulării digitale la rezolvarea problemelor de programare liniară cu mai multe funcţii obiectiv.* Studii şi cercetări de calcul economic şi cibernetică economică, No.1, 1976.
14. KEMENY I. G., SNELL I.L.: *Finite Markov Chains.* D. von Nostrand Company Inc., Princeton—New Jersey—Toronto—London—New York, 1960.

DEUX PROCEDURES DE SIMULATION POUR LA RESOLUTION DES PROBLEMES DE PROGRAMMATION ENTIERE CONVEXE

On propose deux procedures de simulation, dénommées „dirigée“ et „semi-dirigée“ pour la résolution des problèmes de programmation entière convexe.

Les expériences numériques, effectuées à l'aide d'exemples classiques de test Lawler et Bell [4], tendent à démontrer l'efficacité des procédures avancées.

ДВЕ ПРОЦЕДУРЫ СИМУЛИРОВАНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ ЦЕЛОЧИСЛЕННОГО ВЫПУКЛОГО ПРОГРАММИРОВАНИЯ

В настоящей работе предложены две процедуры симулирования, названные «управляемой» и «полууправляемой» для решения задач целочисленного выпуклого программирования.

Численные расчеты, сделанные с классическими примерами критериев проверки Лаулэра и Белла [4] призваны проиллюстрировать эффективность предложенных в работе процедур.

9. V. 1977