Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Letter to the Editor—Reduction of Integer Polynomial Programming Problems to Zero-One Linear Programming Problems

Lawrence J. Watters,

TABLE II

Considering M^+							
s	J_s	z_s	y_i^s				
0	ϕ	0	10	6	−4	−12	0
.	.	.	.	.	.	.	.
1	4	16	6	3	1	−2	2
!2	4, 3	20	2	1	1	0	0
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

REFERENCES

1. E. Balas, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Opns. Res.* **13**, 517–546 (1965).
2. F. Glover and S. Zionts, "A Note on the Additive Algorithm of Balas," *Opns. Res.* **13**, 546–549 (1965).
3. R. J. Freeman, "Computational Experience with a 'Balasian' Integer Programming Algorithm," *Opns. Res.* **14**, 935–941 (1966).
4. W. D. James, "Some Remarks on a Numerial Example of Balas and on the Note by Glover and Zionts," *Opns. Res.* **14**, 941–942 (1966).
5. B. Fleischmann, "Computational Experience with the Algorithm of Balas," *Opns. Res.* **15**, 153–155 (1967).

REDUCTION OF INTEGER POLYNOMIAL PROGRAMMING PROBLEMS TO ZERO-ONE LINEAR PROGRAMMING PROBLEMS

Lawrence J. Watters

The Rand Corporation, Santa Monica, California

(Received January 13, 1967)

THIS letter considers the nonlinear programming problem:

Optimize $f(x_1, x_2, \cdots, x_n)$, (1a)

subject to $g_i(x_1, x_2, \cdots, x_n) \geqq b_i, \quad (i=1, 2, \cdots, m)$ (1b)

where the variables $x_1, x_2, \cdots, x_n$ assume finite integer values and $f(x_1, x_2, \cdots, x_n)$ and $g_i(x_1, x_2, \cdots, x_n)$ are polynomials. Conversion of an integer formulation to a zero-one formulation (i.e., one in which the variables may assume only the values 0 or 1) is easily accomplished.* Hence, the ensuing discussion will be directed at reducing zero-one polynomial formulations to zero-one linear formulations.

Techniques are available for attacking problem (1); however, the majority of these techniques are either suboptimal or susceptible to the vagaries and complexities of continuous-variable nonlinear algorithms. In addition to the available techniques surveyed by BEALE,[3] BALINSKI,[2] and LAWLER AND WOOD,[8] approximation methods have been developed by REITER AND RICE[10] and HILLIER.[7] Hillier also describes an exact method that involves three stages:

1. Replace the original nonlinear constraints with a set of uniformly *tighter* linear constraints and find the corresponding (suboptimal) solution.
2. Replace the original nonlinear constraints with a set of uniformly *looser* linear constraints.
3. Apply a modified version of the 'bound-and-scan' algorithm[6] to the problem as reformulated in step 2, using the suboptimal solution found in step 1 as the required initial feasible solution.

TRANSFORMATION OF POLYNOMIAL ZERO-ONE FORMULATIONS

THE BINARY nature of the variables permits appropriate transformations to be made that effectively reduce polynomial zero-one formulations to equivalent linear zero-one formulations.

Consider a simple quadratic function, $f(x_1, x_2, \cdots, x_n)$ in which the variables must be either 0 or 1. Obviously, x_j^2 can be replaced by x_j without affecting the value of the function. Also, a new variable, x_{jk}, is needed to replace a product x_jx_k such that its values correspond to the values of x_j and x_k as follows:

x_j	x_k	x_{jk}
0	0	0
0	1	0
1	0	0
1	1	1

The desired correspondence is obtained by simply requiring that

$$x_j+x_k-x_{jk}\leqq 1,$$

$$-x_j-x_k+2x_{jk}\leqq 0,$$

$$x_{jk}=0 \text{ or } 1.$$

Higher degree functions can be reduced in a similar manner. In general, given a set, Q, composed of q zero-one variables, the product $\prod_{j\epsilon Q} x_j^p$ (for any positive values of p) can be replaced by a single new variable x_Q and imposing the additional

* Specifically, if the integer variable x has a finite upper bound u, the variable x can be expressed in terms of binary variables y_j as follows: $x_j=\sum_{j=0}^{j=k} 2^j y_j$, where k is the smallest integer such that $u\leqq 2^{k+1}-1$.

constraints

$$\sum_{j\epsilon Q} x_j - x_Q \leqq q-1, \tag{2}$$

$$-\sum_{j\epsilon Q} x_j + q x_Q \leqq 0, \tag{3}$$

$$x_Q = 0 \text{ or } 1. \tag{4}$$

Note that if any $x_j=0$, then constraint (2) is nonrestrictive, constraint (3) becomes $x_Q<1$, and therefore $x_Q=0$. If all $x_j=1$, then constraint (2) becomes $1\leqq x_Q$, constraint (3) becomes $x_Q\leqq 1$, the equality holds, and the desired relation is obtained.

AN APPLICATION

THE TRANSFORMATION has been applied to a particular type of investment problem in which the high-risk investment alternatives have nondeterministic interrelated cash flows subject to multiperiod probabilistic budget constraints.[11] The problem involved optimizing a nonlinear function $f(x_1, \cdots, x_N)$, subject to

$$\text{Prob}\,\{\textstyle\sum_{j=1}^{j=N} \tilde{a}_{ij} x_j \leqq b_i\} \geqq \alpha_i, \qquad (i=1, 2, \cdots, M)$$

$$x_j = 0 \text{ or } 1,$$

where the $\tilde{a}_{ij}$ are random variables, b_i are constants, and $0<\alpha_i\leqq 1$.

Specifically, the function $f(x_1, \cdots, x_N)=\sum_{j=1}^{j=N}\sum_{k=1}^{k=N} c_{jk}x_jx_k$ was maximized, and the joint distribution of the random variables $\tilde{a}_{ij}(j=1, 2, \cdots, N)$ was multivariate normal, where

$$E\{\textstyle\sum_{j=1}^{j=N} \tilde{a}_{ij}x_j\} = \sum_{j=1}^{j=N} \mu_{ij}x_j, \qquad \text{and}$$

$$\text{Var}\,\{\textstyle\sum_{j=1}^{j=N} \tilde{a}_{ij}x_j\} = \sum_{j=1}^{j=N} \sigma_{ij}^2 x_j^2 + 2\sum_{j=1}^{N-1}\sum_{k=j+1}^{k=N} \sigma_{ijk}x_jx_k.$$

The probabilistic constraints were expressed in terms of equivalent zero-one polynomial constraints, thereby converting the chance-constrained problem to the form of problem (1). A computer program has been used to make the necessary transformations and to solve resulting problems (containing up to 45 variables and 80 constraints) with a modified version of the Balas algorithm.[1]

In problems with numerous cross-products, the number of additional variables and constraints required to effect the transformation may become quite large and generally intractable with available zero-one solution techniques. However, additional improvements in zero-one algorithms, such as those suggested by GEOFFRION,[4] GLOVER,[5] AND PETERSEN,[9] will further the usefulness of this method for solving integer polynomial programming problems.

REFERENCES

1. E. BALAS, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Opns. Res.* **13,** 517–546 (1965).
2. M. L. BALINSKI, "Integer Programming: Methods, Uses, Computation," *Management Sci.* **12,** 253–313 (1965).
3. E. M. BEALE, "Survey of Integer Programming," *Opnal. Res. Quart.* **16,** 219–228 (1965).

4. A. M. GEOFFRION, *Integer Programming by Implicit Enumeration and Balas' Method*, The Rand Corporation, RM-4783-PR, February, 1966.
5. F. GLOVER, "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem," *Opns. Res.* **13**, 879–919 (1965).
6. F. S. HILLIER, "An Optimal Bound-and-Scan Algorithm for Integer Linear Programming," *Technical Report No. 3*, Contract NONR-225 (89), Stanford University, Stanford, Calif., August 19, 1966.
7. ———, "Chance-Constrained Programming with Zero-One or Bounded Decision Variables," *Technical Report No. 4*, Contract NONR-225 (89), Stanford University, Stanford, Calif., August 26, 1966.
8. E. L. LAWLER AND D. E. WOOD, "Branch-and-Bound Methods: A Survey," *Opns. Res.* **14**, 699–719 (1966).
9. C. C. PETERSEN, "Computational Experience with Variants of the Balas Algorithm, Applied to the Selection of R&D Projects," *Management Sci.* **13**, 736–750 (1967).
10. S. REITER AND D. B. RICE, "Discrete Optimizing Solution Procedures for Linear and Nonlinear Integer Programming Problems," *Management Sci.* **12**, 829–850 (1966).
11. L. J. WATTERS, "R&D Project Selection: Interdependence and Multiperiod Probabilistic Budget Constraints," Ph.D. Dissertation in Industrial Engineering, Arizona State University, Tempe, 1966.

STRONGER CUTS IN INTEGER PROGRAMMING†

Fred Glover

University of Texas, Austin, Texas

(Received March 1967)

In a recent note ROBERT WILSON [*Opns. Res.* **15**, 155–156 (1967)] shows how to calculate stronger cuts than those proposed by Gomory for his all-integer algorithm. We supplement Wilson's observations by providing a basis for calculating other cuts, drawing on results of GLOVER [*Management Sci.* **13**, 254–268 (1966)].

IN AN interesting note,[5] ROBERT WILSON points out that one can calculate stronger cuts than are given by GOMORY in the construction of his all-integer integer programming algorithm.[3] Wilson goes on to remark that stronger cuts than those of Gomory's fractional algorithm[4] can be similarly obtained.

Alternatively, cuts that are stronger than the fractional Gomory cuts are derived and exemplified in reference 1.‡ We would like to point out here

1. Wilson's observations can also be inferred from results contained in reference 1.

† This research was supported by the Miller Research Institute with the University of California.

‡ The work of reference 5 and the work of reference 1 were done independently.