to Integer Programming, D. Sc. Dissertation, Washington University, St. Louis, Missouri.

[34] Sabbagh, M.S. (1983). A General Lexicographic Partial Enumeration Algorithm for the Solution of Integer nonlinear Programming Problems, Doctor of Science Dissertation, Department of Operations Research, The George Washington University, Washington D.C.

[35] Salkin, H. M. (1975). Integer Programming, Addison Wesley, Reading, Massachusetts.

[36] Sinha, P. and Zoltners, A. (1979) The Multiple Choice knapsack problem, Operations Res., Vol. 27, PP. 503-515.

[37] Soland, R. M. (1973). Optimal Defensive Missile Allocation: A Discrete Min-Max Problem. Operations res., Vol. 21, PP. 590-596.

[38] Taha, H. A. (1975). Integer Programming, Applications, and Computatios, Academic press, New York.

[39] Weinstein, I. J. and Yu, S. O. (1972). Comment on an Integer maximization Problem, Operatios res., Vol. 21, Pp. 648-650.

[40] Yormark, J. (1979). Efficient Frotiers for Nonliear Multidimensional Knapsack Problems. Working Paper, Graduate School of Business Administration, University of Southern California, Los Angeles, California.

[12] De Maio, A. and Roveda, C. (1969). Letter to the Editor. Management Sci., Vol. 15, p. b-481.
[13] Denardo, E. V. and Fox, B. L. (1979a). Shortest Route Methods: 1. Reaching, Pruning, and Buckets, Operations res., Vol. 27, PP. 161-168.
[14] Denardo, E. V. and Fox, B. L. (1979). Shortest Route methods: 2. Group Knapsacks, Expanded Networks, and Branch and Bound, Operations res., Vol. 27, PP. 548-566.
[15] Dharadhikari, V. K. (1975). Solving Discrete-Variable Multiple-Constraint Non-Linear programs: The decision-State method, ph.D. Dissertation, department of Computer Science and Operations Research, Southern methodist University, dallas, Texas.
[16] Dragan, I. (1968). Un Algorithme Lexicographique pour la Resolution des Programmes Polynomiaux en Variables Entieres, R.L.R.O., Vol. 2, PP. 81-90.
[17] Gross, D., miller, D. R. and Soland, R.M. (1981). A Closed Queuing Network Model for Multi-Echelon Repariable Item Provisioning. Technical Paper Serial T-446, Program in Logistics, The George Washington University.
[18] Gupta, O.K. (1980). Branch and Bound Experiments in Nonlinear Integer Programming, ph.D. Thesis, School of Industrial Engineering, Purdue University, West Lafayette, Indiana.
[19] Gupta, O. K. and Ravindran, A. (1982). Branch and Bound Experiments in Nonlinear Mixed Integer Programming, Committee on Algorithms Newsletter, Mathmatical Programming Society, No. 7.
[20] Gupta, O.K., and A. Ravindran. (1985). Branch and Bound Experiments in convex Nonlinear Integer Programming. Mgmt. Sci. 31. 1533-1546.
[21] Hansen, P.(1979). Methods of Nonlinear 0-1 Programming, Ann. Discrete Math., Vol. 5, PP. 53-70.
[22] Harrison, R. A. (1982). Aircraft Production and development Schedules. Technical Paper Serial T-463, Program in Logistics, The George Washington University.
[23] Hochbaum, D. S. (1990a). On the Impossibility of Strongly Polynomial Algorithms for the Allocation Problem and Its Extensions. In Proceedings of Integer Programming and Combinatorial optimization, waterloo, 261-274.
[24] Hu, T. C. (1969). Integer Programming and Network Flows, Addison wesley, Reading, Massachusetts.
[25] Kunzi, H. P. and Oettli, W. (1973). Integer Quadratic Programming, Recent Advances in mathematical programming, Graves and Wolfe (ed.), McGraw Hill, New York.
[26] Lawler, E. L. and Bell, M. D. (1966). A Method for Solving Discrete Optimization Problems. Operations res., Vol. 14, PP. 1098-1112.
[27] Mao, J. C. T. and Wallingford, B. A. (1968). An Extension of Lawler and Bell's Method of discrete Optimization with Examples from Capital Budgeting, Management Sci., Vol. 15, PP. b51-b60.
[28] McCallum, C. J. Undated. An Algorithm for Certain Quadratic Integer Programs. Bell Laboratories Technical Report, Holmdel, New Jersey.
[29] Miller, B. L. (1971). On Minimizing Non-Separable Functions Defined on the Integers with an Inventory Application, SIAM J. Appl. Math., Vol. 2, 166-185.
[30] Morin, T. L. (1978). Computational Advances in Dynamic Programming, Dynamic Programming and its Applications, M. L. Puterman, (ed), Academic Press, New York.
[31] Morin, B. L. and Marsten, R. E. (1976a). An Algorithm for Nonlinear Knapsack Problems. Management Sci., Vol. 22, PP. 1147-1158.
[32] Morin, B. L. and Marsten, R. E. (1976b). Branch and Bound Strategies for Dynamic Programming, Operations Res., Vol. 24, PP. 611-627.
[33] Tao, V. B. (1967). Some Approaches

and using this new algorithm we apply rule 1, 19 times; rule 2, 14 times; rule 3, 8 times; and rule 4, 2 times and obtain $X^* = (2,1, 1, 1, 1)$, $f(x^*)=8$.

Example 2

Minimize $f(X)=x_1x_7+3x_2x_6+x_3x_5+7x_4$,

Subject to:

$x_1+x_2+x_3 \geq 6$,

$x_4+x_5+6x_6 \geq 8$,

$x_1x_6+x_2+3x_5 \geq 7$,

$4x_2x_7+ 3x_4x_5 \geq 25$,

$3x_1+2x_3+x_5 \leq 7$,

$3x_1x_3+6x_4+4x_5 \leq 20$,

$4x_1+ 2x_3 + x_6x_7 \leq 15$,

$0 \leq X \leq U$,

$U=(2,7,3, 3, 5, 7, 7)$.

In this problem there are N=147456 points and using this new algorithm we apply rule 1, 960 times; rule 2, 589 times; rule 3, 275 times; and rule 4, 2 times and obtain $X^{\#}=(0, 4, 2, 0, 2, 1, 2)$, $f(X^{\#})=16$.

Conclusiuns

In this paper, we have developed a general lexicographic partial enumeration algorithm for the solution of integer nonlinear programming problems with discrete isotone nondecreasing objective function and constraints.

The positive points of this algorithm are:

(a) The algorithm finds the global constrained optimial solution by function evaluations only, and so it does not need any other condition than being able to obtain the function values.

(b) The presence of linear constraints in an integer nonlincar programming problem can speed up the finding of a solution. This will be the topic of an upcomig paper.

(c) The algorithm can be adjusted to solve a broader class of optimization problems. This will be the subject of another paper.

The only limiting factor about the use of this algorithm is the problem size which is a common factor in all systematic search algorithms which try to find grobal optimal solutions.

References

[1] Abadie, J., Dayan, H. and Akoka, J. (1976). Quelques. Experiences Numeriques sur la Programmation Non Lineaire en Nombres Entiers, R.A. I.R.O. Recherche Operationalle, Vol. 10, PP. 65-70.

[2] Apostol, T.M. (1975). Mathematical Analysis, Second Edition, Addison-Wesley Publishing Company, Reading Massachusetts, PP. 127-133.

[3] Aust, R.J. (1976). A Dynamic Programming Branch and Bound Algorithm for Pure Integer Programming, Computers and Operations Res., Vol. 5, PP.27-38.

[4] Balas, E. (1968). Duality in Discrete Programming III: Nonlinear Objective Function and Constraints, Management Science Research Report NO. 125, Carnegie Mellon University.

[5] Balas, E. (1969). Duality in Discrete Programming: The Quadratic Case, Mangement Sci., Vol. 16, PP. 14-32.

[6] Bretthauer, K. M., B. Shetty and S. Syam (1995). A Branch and Bound Algorithm for Integer Quadratic Knapsack Problems. ORSAJ. Comput. 7, 109-116.

[7] Brucker, P. (1984). An O(n) Algorithm for Quadratic Knapsack Problems. O.R. Letts. 3, 163-166.

[8] Buzen, J.P. (1973). Computational Algorithms for Closed Queuing Networks with Exponential Servers. Comm. of the ACM, Vol.16, PP. 527-531.

[9] Cooper, M.W. (1980). The Use of Dynamic Programming Methodology for the Solution of a Class of Nonlinear Programming Problems, Naval Res. Logistics Quarterly, Vol. 27, PP. 89-95.

[10] Cooper, M. W. (1981). A Survey of methods for Pure Nonlinear Integer Programming. Management Sci., Vol. 27, PP. 353-361.

[11] Dantzig, G. (1962). Linear programming and extensions, Princeton University Press, Princeton, N.J.

$x_k > 0$. Now taking values of $i > 0$ in decreasing order, beginning with $i=k-1$, find the first value of i such that $x_i<u_i$. Then set $x^\#_r=x_r$ for $r=1, ..., i-1$ and $x^\#_i=x_i+1$, and $x^\#_j=0$, for $j=i+1, ..., n$.

Examples

$X=0$ -----> $X^@=U$ ----> $X^\#$ does not exist.

$X=(0, ..., 0, 1)$ ----> $X^@=(0, ..., 0, u_n)$ ---> $X^\#=(0, ..., 0, 1, 0)$.

$X=(0,..., 0, x_n)$ ----> $x^@=(0, ..., 0, u_n)$ ----> $X^\#=(0,..., 0, 1, 0)$.

$X=(x_1, ..., x_j, 0, x_{j+2}, 0, ..., 0)$ --->. $X^@=(x_1, ..., x_j, 0, u_{j+2}, ..., u_n)$ ---> $X^\#=(x_1, ..., x_j, 1, 0, ..., 0)$.

$X=(x_1, 0, ..., 0)$-----> $X^@=U$ ---> $X^\#$ does not exist.

$X=U$ -----> $X^@=U$ ----> $X^\#$ does not exist.

The algorithm

Let us consider the optimization problem that was introduced earlier:

Minimize $f(X)$,

Subject to $g_i(X) \geq b_i$, $i=1, ..., k$,

$g_i(X) \leq b_i$, $i=k+1, ..., m$,

$0 \leq X \leq U$,

$X=(x_1, ..., x_n)$, $U=(u_1, ..., u_n)$.

Where u_j is the integer upperbound for the integer variable x_j.

Here, $f(.)$ and $g_i(.)$ $(i=1, ..., m)$ are discrete isotone nondecreasing functions.

We might solve this problem by examining each of the $X \in S$ in lexicographic order, starting with $X = 0$ and ending with $X=U$. However this process can be considerably shortened by invoking certain rules, which we explain below. In general, the more points we skip over the more efficient the algorithm becomes. As we proceed through the list of vectors, we keep a record of X^*, the incumbent optimal solution. and $f^*=f(X^*)$, the incumbent optimal value. The following rules indicate conditions under which certain vectors in the lexicographic ordering can be skipped. The vector $X \in S$ is the one currently being examined and keep in mind that the functions $f(.)$ and $g_i(.)$, S are discrete isotone nondecreasing functions.

Rule 1:

If for any $X \in S$, $f(X) \geq f^*$ skip to $X^\#$.

Justification: clearly $f(Z) \geq f(X) \geq f^*$ for any $Z \in S$ such that $X \leq Z \leq X^@$.

Therefore, no better solution will be found between X and $X^@$ and we can safely skip to $X^\#$.

Rule 2:

(a) If $g_i(X^@)<b_i$ for any $i=1, ..., k$, skip to $X^\#$.

Justification: clearly $g_i(X) \leq g_i(Z) \leq g_i(X^@) < b_i$ for any $Z \in S$ such that $X \leq Z \leq X^@$.

Therefore, no new vector between X and $X^@$ will be found such that this ith constraint will be satisfied, and we can safely skip to $X^\#$.

(b) If $g_i(X)>b_i$ for any $i=k+1, ..., m$, skip to $X^\#$.

Justification: clearly $g_i(X^@) \geq g_i(Z) \geq g_i(X) > b_i$ for any $Z \in S$ such that $X \leq Z \leq X^@$.

Therefore, no new vector between X and $X^@$ will be foud such that this ith constraint will be satisfied, and we can safely skip to $X^\#$.

Rule 3:

If $g_i(X^@) \geq b_i$ for $i=1, ..., k$ and $g_i(X) \leq b_i$ for $i=k+1, ..., m$, but $g_i(X) < b_i$ for any $i=1, ..., k$, continue the enumeration with X' because it might satisfy the constraints.

Rule 4:

If $f(X) < f^*$ and X is feasible, let X^* <--- X, f^* < --- $f(X)$. and continue the enumeration with $X^\#$ because this point might reduce the objective function further.

Example 1:

Minimize $f(X)=X^2_1+X^2_2+X^2_3+X^2_4+X^2_5$,

subject to:

$X_2+X_3+2x_4 \geq 4$,

$X_4+2x_5 \geq 3$,

$2X_1+X_2 \geq 5$,

$x_2+x_4+2x_5 \leq 6$,

$2x_2+x_5 \leq 4$,

$4x_1+x_2 \leq 13$,

$0 \leq X \leq U$,

$U=(3, 2, 3, 3, 3)$.

In this problem there are N= 768 points

established, are often helpful in determining if a given function is isotone nondecreasing. Suppose g(.) and h(.) are isotone nondecreasing functions on T. Also, assume a and b are nonnegative constants. Then in each case the function f(.) defined below is also an isotone nondecreasing function on T.

(a) f(.)=g(.) + a or f(.)=g(.) - a.

(b) f(.)=af(.)+bh(.).

(c) f(.)=w(g(.)), where w is monotone nondecreasing function on R.

(d) f(.)=minimum {g(.), h(.)}.

(e) f(.)=maximum {g(.), h(.)}.

If, in addition, g(.) and h(.) are nonnegative, then the following functions are also isotone nondecreasing on T:

(f) f(.)=g(.) h(.)

(g) $f(.)=g(.)^a$

Let $S = \{(s_1, ..., s_n)\}$ where $s_j=0,1,..., u_j$ and $j=1, ..., n$. If for every $X \in S$ and every $Y \in S$ in which $X \geq Y$ implies $g(X) \geq g(Y)$, then the function g(.) is called discrete isotone nondecreasing function on S.

Let Z be an n-vector of nonnegative integers. It is easy to prove that g(.) is a discrete isotone nondecreasing function on S iff $h(X, Z)=g(X+Z)-g(X) \geq 0$, for all Z and X such that $X \in S$ and $(X+Z) \in S$.

Nontice that a function which is not an isotone nondecreasing function may be a discrete isotone nondecreasing function.

Example

$f(X)=X^2_1-X_1+X^2_2$ is not an isotone nondecreasing function on $T=\{(X_1,X_2)|X_1 \geq 0, X_2 \geq 0\}$ because we have f(0)=0 and f (1/2, 0)=1/4. However, this is a discrete isotone nondecreasing function on the set S.

From now on we require that the objective function and functional constraints to be discrete isotone nondecreasing function on S.

Lexicographic (complete) ordering: Let $Y=(y_1, ..., y_n) \in R^n$. A vector Y is said to be lexicographically positive, written $Y >^L 0$, if $y_1=y_2=... y_{j-1}=0$, and $y_j>0$, for some $j=1, ...,n$. For $X\&Y \in S$ we write $X <^L Y$ to mean $(Y-X) >^L 0$. We write $X \leq^L Y$ to mean either (a) $(Y-X) >^L 0$, or (b) X=Y. We say "X precedes Y" (in the lexicographic ordering) to mean $X <^L Y$.

The set S, defined earlier, has $N=(u_1+1)(u_2+1) ... (u_n+1)$ elements. The lexicographic ordering allows us to uniquely order the N elements of S as $S_1, ..., S_N$ such that $S_1 <^L S_2 <^L S_3 ... <^L S_{N-1} <^L S_N$. We have, in this ordering, $S_1=(0, ..., 0)$, $S_2=(0, ..., 1)$,..., $S_{N-1}=(u_1,..., u_{n-1}, u_n-1)$, and $s_N=(u_1,..., u_n)$. In the following for every X we define the vectors X′, $X^@$, and $X^\#$. The next immediate vector of X is called X′. If $X=S_k$ and $k=1, ..., N-1$ then $X'=S_{k+1}$ and if $X=S_N=U$ we say X′ does not exist.

$X^@$ may be used to obain useful function bounds for the constraints of the form $g_i(X) \geq b_i$ and is defined in the following way: if $X= S_k$, $k=1, ..., N$ then one can find the largest m such that $m \geq k$ and $S_k \leq^L S_i \leq^L S_m$ implies $S_k \leq S_i \leq S_m$ for all $i=k, ..., m$. That is, the largest m such that for all the vectors between X and s_n lexicographic ordering implies partial ordering.

One can obtain $X^@$ of any $X \in S$ in the following way. Taking values of $j \geq 1$ in decreasing order, beginning with j=n, find the first value of j, say j=k, such that $x_k>0$. Then set $x^@_i=x_i$ for $i=l, ..., k-1$ and $x^@_i=u_i$ for $i=k, ..., n$. If there is no value of j with these properties let $X^@=U$.

Examples

$X=0$ ----------> $X^@=U$.

$X=(0, ..., 0, 1)$ --------> $X^@=(0, ..., 0, u_n)$.

$X=(0, ..., 0, x_n)$ --------> $X^@=(0, ...,0, u_n)$.

$X=(x_1, ..., xj, 1, 0, ..., 0)$ --> $X^@=(x_1, ..., xj, u_{j+1}, ..., u_n)$.

$X=(x_1, 0, ..., 0)$ ------>$X_@=U$.

$X=U$ --------> $X_@=U$.

If the function value at $X^@$ dictates to continue our enumeration with the vector immediatly following $X^@$ which is called $X^\#$ and is defined in the following way: if $X^@=U$ we say $X^\#$ does not exist otherwise $X^\#=(X^@)'$.

one can directly obtain $X^\#$ of any $X \in S$ in the following way. Taking values of $j \geq 1$ in decreasing order, beginning with j=n, , find the first value of j, say j=k, such that

A General Lexicographic Partial Enumeration Algorithm for the Solution of Integer Nonlinear Programming Problems with Discrete Isotone Nondecreasing Objective Function and Constraints

Mohammad S. Sabbagh

Assistant Professor
Department of Industrial Engineering and Systems
Planning Center
Isfahan University of Technology

Abstract

This paper presents a general lexicographic partial enumeration algorithm for the solution of integer nonlinear programming problems. The algorithm requires that the objective and the functional constraints to be discrete isotone nondecreasing functions.

Introduction

This paper presents a general lexicographic partial enumeration algorithm for solving integer nonlinear programming problems of the form:

Minimize f (X),

Subject to $g_i(X) \geq b_i$, i = 1, ..., k,

$g_i(X) \leq b_i$, i = k+1, ..., m,

$0 \leq X \leq U$.

$X=(x_1, ..., x_n)$, $U=(u_1, ..., u_n)$.

Where u_i is the integer upperbound for the iteger variable x_i.

Here, f(.) and g_i (.), i=1, ..., m are discrete isotone nondecreasing functions.

Some of the features of this algorithm are as follows:

1. It locates the global constrained integer optimal solution by function evaluations only, and so does not require that the functions be continuous or even defined for noninteger values of the variables. It is not even necessary to have explicit algebraic expressions for the functions.
2. It is easy to program and requires a small amout of computer memory.
3. It is not necessary to transform the variables to weighted sums of binary variables.

Definitions

In the following definitions let T be a subset of R^n.

Vector partial ordering: Let $X \in T$ and $Y \in T$. We write $X \leq Y$ to mean $x_j \leq y_j$ for j=1, ..., n. Similarly, X=Y means $x_j=y_j$ for j=1, ..., n.

Isotone nondecreasing function: We say g(.) is an isotone nondecreasing function on T iff for every $X \in T$ and every $Y \in T$ and $X \geq Y$ implies $g(X) \geq g(Y)$.

It can be proved that if f(.) is differentiable on T and $\nabla f(X) \geq 0$ for every $X \in T$ then f(.) is an isotone nondecreasing function on T. Moreover, if f(.) is continuously differentiable on T, then f(.) is an isotone nondecreasing function on T iff $\nabla f(X) \geq 0$, for every $X \in T$.

The following results, which are easily