AN IMPROVED SEARCH ALGORITHM FOR SOLVING MIXED-INTEGER NON LINEAR PROGRAMMING PROBLEM

Pengarapen Bangun, Almira Amir, Erlinawaty Simanjuntak, Herman Mawengkang

Department of Mathematics, University of Sumatera Utara, Medan - Indonesia

Abstract

The special nonlinear mathematical programming problem which is addressed in this paper has a structure characterized by a subset of variables restricted to assume discrete values, which are linear and seperable from the continuous variables. The strategy of releasing nonbasic variables from their bounds, combined with the "active constraint" method and the notion of superbasics, has been developed for efficiently tackling such a problem by ignoring the integrality requirements, this strategy is used to force the appropriate non-integer basic variables to move to their neighbourhood integer points. A study of criteria for choosing a nonbasic variable to work with in the integerizing strategy has also been made. Successful implementation of these algorithms was achieved on various test problems. The result show that the proposed integerizing strategy is promosing in tackling certain classes of mixed integer programming problems.

1. Introduction

The special Mixed-Integer nonlinear programming problem which is addressed to assume discrete values, which are linear and separable from the continuous variables. This problem is defined by the following mixed-integer nonlinear programming (MINLP) program.

$$\text{Min } z = c^T y + f(x) \quad (1)$$

$$h(x) \leq 0 \quad (2)$$

$$g(x) + B y \leq 0 \quad (3)$$

$$x \in X \subset R_+^n, \ y \in Y \subset R_+^m \quad (4)$$

where $f: R^n \to R$ and $h: R^n \to R^p$, $g: R^n \to R^q$ are continuous and generally well-behaved function defined on the n-dimensional compact polyhedral convex set $X = \{x : x \in R^n, A_1 x \leq a_1\}$; $U = \{y : y \in Y, \text{ integer}, A_2\, y \leq a_2\}$ is a discrete set, say the nonnegative integer points of some convex polytope, where for most applications Y is the unit hypercube $Y = \{0,1\}^m$. B, A_1, A_2, and c, a_1, a_2 are respectively matrices and vectors of comfortable dimensions; the vectors are column vectors unless specified otherwise. There has been little reported evidence of previous attempts to solve large nonlinear integer programs. Survey papers by Hansen [5] and Cooper [2] both point out that the paucity computational testing on algorithms that have been proposed. One of the more promosing approaches to nonlinear (0-1) programs is their reductionto an multilinear (0-1) program, followed by linearization to an equivalent set covering problem. Balas and Mazzola [1] present a linearization technique without having to generate additional variables and present computational experience on (0-1) programs of size up to fifty variables and twenty constraints. The applicability of this approach to larger problems needs further investigation. Vassilev and Enova [9] propose an approximate algorithm as a generalization of the algorithm of internal feasible integer directions. A direct search approach for solving such a problem is proposed by Murtagh and Sugden [8]. Fletcher and Leyffer [4] use outer approximation use for solving MINLP problems in which nonlinearities appear in the integer variables.

2. The Basic Approach

Before we proceed to the case of MINLP problems, it is worthwhile to discuss the basic strategy of process for linear case, i.e, Mined Integer Linear Programming (MILP) problems.

Consider a MILP problem with the following form

$$\text{Minimize } P = c^T x \tag{5}$$
$$\text{subject to } Ax \leq b \tag{6}$$
$$x \geq 0 \tag{7}$$
$$x_j \text{ integer for some } j \in J \tag{8}$$

A component of the optimal basic feasible vector $(x_B)_k$, to MILP solved as a continuous can be written as

$$(x_B)_k = \beta_k - \alpha_{k1}(x_N)_1 - \ldots - \alpha_{kj}(x_N)_j - \ldots - \alpha_{kn} - m(x_N)n - m \tag{9}$$

Note that, this expression can be found in the final tableau of Simplex procedure. If $(x_B)_k$ is an integer variable and we assume that c_k is not an integer, the partitioning of β_k into the integer and fractional components is that given

$$\beta_k = [\beta_k] + f_k,\ 0 \leq f_k \leq 1 \tag{10}$$

Suppose we wish to increase $(x_B)_k$ to its nearest integer, ($[\beta] + 1$). Based on the idea of suboptimal solutions we may elevate a particular nonbasic variable, say $(x_N)_{j^*}$, above its bound of zero, provided α_{kj^*}, as one of the element of the vector α_{j^*}, is negative. Let Δ_{j^*}

be amount of movement of the nonbasic variable $(x_N)_{j^*}$, such that the numerical value of scalar $(x_B)_k$ is integer. Referring to Eqn. (9), Δ_{j^*} can then be expressed as

$$\Delta_{j^*} = \frac{1-f_k}{-\alpha_{kj^*}} \tag{11}$$

while the remaining nonbasic stay at zero. It can be seen that after substituting (11) into () for $(x_N)_{j^*}$ and taking into account the partitioning of β_k given in (10), we obtain

$$(x_B)_k = [\beta] + 1$$

Thus, $(x_B)_k$ is now an integer.

It is now clear that aa nonbasic variable plays an important role to integerize the corresponding basic variable. Therefore, the following result is necessary in order to confirm that must be a non-integer variable to work with in integerizing process.

Theorem 1. *Suppose the MILP problem* (5)-(8) *has an optimal solution, then some of the nonbasic variables, $(x_N)_j$, $j = 1, \ldots, n$, must be non-integer variables.*

Proof.
Solving problem as a continuous of slack variables (which are non-integer, except in the case of equality constraint). If we assume that the vector of basic variable x_B consists of all the slack variables then all integer variables would be in the nonbasic vector x_N and therefore integer-valued.

3. Derivation Of The Method

It is clear that the other components, $(x_B)_{i\neq k}$, of vector x_B will also be affected as the numerical value of the scalar $(x_N)_{j^*}$ increases to Δ_{j^*}. Consequently, if some element of vector α_{j^*}, i.e., α_{ij^*} for $i \neq k$, are positive, then the corresponding element of x_B will decrease, and eventually may pass throught zero. However, any component of vector x must not go below zero due to the non-negativity restriction. Therefore, a formula, called the minimum ratio test is needed in order to see what is the maximum movement of the nonbasic $(x_N)_{j^*}$ such that all components of x remain feasible. This ratio test would include two cases.

1. A basic variable, $(x_B)_{i\neq k}$, decreases to zero (lower bound) first.
2. the basic variable, $(x_B)_k$, increases to an integer.

Specifically, corresponding to each of these two cases above, one would compute

$$\theta_1 = \min_{i\neq k \mid \alpha_{ij^*}>0} \left\{ \frac{\beta_i}{\alpha_{ij^*}} \right\} \tag{12}$$

$$\theta_2 = \Delta_{j^*} \tag{13}$$

How far one can release the nonbasic c froms its bound of zero, such that vector x remains feasible, will depend on the ratio test θ^* given below

$$\theta^* = \min(\theta_1, \theta_2) \tag{14}$$

Obviously, if $\theta^* = \theta_1$, one of the basic variable $(x_B)_{i \neq k}$ will hit the lower bound before $(x_B)_k$ becomes integer. If $\theta^* = \theta_2$, the numerical value of the basic variable $(x_B)_k$ will be integer and feasibility is still maintained. Analogously, we would be able to reduce the numerical value of the basic variable $(x_B)_k$ to its closest integer $[\beta_k]$. In this case the amount of movement of a particular nonbasic variable, $(x_N)_{j^*}$, corresponding to any positive element of vector $\alpha_{j'}$, is given by

$$\Delta_{j'} = \frac{f_k}{\alpha_{kj'}} \tag{15}$$

In order to maintain the feasibility, the ratio test θ^* is still needed. Consider the movement of a particular nonbasic variable, Δ, as expressed in Eqns. (11) and (15). The only factor that one needs to calculate is the corresponding element of vector α . A vector α_j can be expressed as

$$\alpha_j = B^{-1}\, \alpha_j\, ,\ j = 1, \dots, n\text{-}m \tag{16}$$

Therefore, in order to get a particular element of vector α_j we should be able to distinguish the corresponding column of matrix $[B]^{-1}$. Suppose we need the value of element α_{kj^*}, letting v_k^T be the k-th column vector of $[B]^{-1}$, we then have

$$v_k^T = e_k^T B^{-1} \tag{17}$$

Subsequently, the numerical value of α_{kj^*} can be obtained from

$$\alpha_{kj^*} = v_k^T \alpha_{j^*} \tag{18}$$

In Linear Programming (LP) terminology the operation conducted in Eqns. (17) and (18) is called the pricing operation. The vector of reduced costs d_j can be used to measure the deterioration of the objective function value caused by releasing a nonbasic variable from its bound. Consequently, in deciding which nonbasic should be released in the integerizing process, the vector d_j must be taken into account, such that deterioration is minimized. Recall that the minimum continuous solution provides a lower bound to any integer-feasible solution. Nevertheless the amount of movement of a particular nonbasic variable as given in Eqns. (11) or (15), depends in some way on the corresponding element of vector α_j. Therefore it can be observed that the deterioration of the objective function value due to releasing a nonbasic variable $(x_N)_{j*}$ so as to integerize a basic variable $(x_B)_k$ may be measured by the ratio

$$\left|\frac{d_k}{\alpha_{kj*}}\right| \tag{19}$$

where $|a|$ means the absolute value of scalar a.

In order to minimize the deteoration of the optimal continuous solution we then use the following strategy for deciding which nonbasic variable may be increased from its bound of zero, that is,

$$\min_J \left\{\left|\frac{d_j}{\alpha_{ij}}\right|\right\}, \quad j = 1, \ldots, n\text{-}m \tag{20}$$

The notion of suboptimal solutions to the LP can be extended analogously to the case of nonlinear programming (NLP), although the optimal solution to the NLP problem may be global or local depending on the convexity of the problem functions. It should be emphasized that local solution to a NLP problem cannot be considered as a suboptimal solution in terms of the global solution. In other words, suboptimality has nothing to do with the global or local nature of the solution. The framework of the approach to handle the MINLP problem is provided by MINOS code. Therefore the optimal continuous solution to the nonlinear problem, as well as the linear problem is obtained by using the MINOS software. From the "active constraint" strategy in minos and the partitioning of the linearized constraints corresponding to basic (B), superbasic (S) and nonbasic (N) variables we can write

$$\begin{bmatrix} B & S & N \\ & & I \end{bmatrix} \begin{bmatrix} x_B \\ x_N \\ x_S \end{bmatrix} = \begin{bmatrix} b \\ b_N \end{bmatrix} \tag{21}$$

or

$$Bx_B + Sx_S + Nx_N = b \tag{22}$$

$$x_N = b_N \tag{23}$$

The basic matrix B is assumed to be square and nonsingular, we get

$$x_B = \beta - Wx_S - \alpha x_N \tag{24}$$

where

$$\beta = B^{-1} b \tag{25}$$

$$W = B^{-1} S \tag{26}$$

$$\alpha = B^{-1} N \tag{27}$$

Expression (23) indicates that the nonbasic variables are being held equal to their bound. It is evident through the "nearly" basic expression of Eqn. (24), the integerizing strategy discussed in the previous section, designed initially for MILP problem can be implemented for the nonlinear case. Particularly, we would be able to release a nonbasic variable from its bound, Eqn. (23) and exchange it with a corresponding basic variable in the integerizing process, although the solution would be degenerate. Furthermore, the Theorem (1) above can also be extended for MINLP problem.

Theorem 2. *Suppose the MINLP problem has a bounded optimal continuous solution, then we can always get a non-integer y_j in the optimum basic variable vector.*

Proof.

1. If these variables are nonbasic, then they will be at their bound. Therefore they have integer value
2. If a y_j is superbasic, it is possible to make y_j basic and bring in a nonbasic at its bound to replace it in the superbasic.

However, the ratio test expressed in (14) cannot be used as a tool to guarantee that the integer solution optimal found still remains in the feasible region. Instead, we use the feasibility test from Minos in order to check whether the integer solution is feasible or infeasible.

4. Pivoting

Currently, we are in apposition where a particular basic variable, $(x_B)_k$, is being integerized , there by a corresponding nonbasic variable, $(x_N)_{j^*}$, is being released from its bound of zero. Suppose the maximum movement of $(x_N)_{j^*}$ satisfies

$$\theta^* = \Delta_{j^*}$$

such that $(x_B)_k$ is integer valued. To exploit the manner of changing the basis found in MINOS, we would be able to move $(x_N)_{j^*}$ into B to replace $(x_B)_k$ and integer-valued $(x_B)_k$ into S in order to maintain the integer solution. we now have a degenerate solution since a basic variable is at its bound. The integerizing process continues with a new set of [B, S]. In this case, eventually we may end up with all of the integer variables being superbasic.

Theorem 3. *A suboptimal solution exists to the MILP and MINLP problem in which all of the integer variables are superbasic.*

Proof.

1. If all of the integer variables are in N, then they will be at bound.
2. If an integer variable is basic it is possible to either
   - Interchange it with a superbasic continuous variable, or
   - Make this integer variable superbasic and bring in a nonbasic at its bound to replace it in the basis which gives a degenerate solution.

The other case which can happen is that a different basic variables $(x_B)_{i \neq k}$ may hit its bound before $(x_B)_k$ becomes integer. Or in other words, we are in a situations where

$$\theta^* = \Delta_1$$

In this case we move the basic variable $(x_B)_I$ into N and its position in the basic variable vector would be replaced by nonbasic $(x_N)_{j*}$. Note that $(x_B)_k$ is still a non-integer basic variable with a new value.

5. Conclusion

The strategy of releasing nonbasic variables from their bounds, combined with the "active constraint" method and the nation of superbasics, has been developed for efficiently ackling mixed integer nonlinear programming problems. After solving a problem by ignoring thee integrality requirements, this strategy is used to force the appropriate non-integer basic variables to move to their neighbourhood integer points. Computational testing of the procedure presented this paper has demonstrated that it is a viable approach for large problems.

References

[1] E. Balas and J.B. Mazzola, Nonlinear 0-1 programming : Linearization Techniques, and II Dominance Relation and algorithms, *Math. Prog.* 30(1984)1.
[2] M.W. Cooper, A Survey of methods for pure nonlinear integer programming, *Management Science* 27(1981)353.
[3] M. A. Duran and I.E. Grossman, An Outer Approximation Algorithm For a class of Mixed – Integer Nonlinear Programs, *Mathematical Programming* 36(1986)307.
[4] Roger Fletcher and Sven Leyffer, Solving Mixed Integer Nonlinear Programming by Outer Approximation, *Mathematical Programming, Vol.* 66(1994)327.
[5] P. Hansen, Methods of nonlinear 0-1 programming, *Ann. Discreate Math.* 5(1979)53.
[6] H. Mawengkang and B. A. Murtagh, Solving Nonlinear Integer Programs with large-scale optimization Software, *Annals of Op. Res.* (1986)42.
[7] B. A. Murtagh and M. A. Saunders, *MINOS / AUGMENTED User's Manuaal, Report SOL* 80-14, System Optimization Laboratory, Stanford University (1980).
[8] B. A. Murtagh and S. J. Saunders. A Direct Search Approach to Non-Linear integer Programming, *Optimization Methods and Software, Vol.* 4(1994)171.
[9] V. Vassilev aand Krassimira Genova, An Approximate Algorithm for Nonlinear Integer programming. *EJOR.* 74(1994)170.