International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 13 Issue 01 January 2025, Page no. – 4727-4730

Index Copernicus ICV: 57.55, Impact Factor: 8.316

DOI: 10.47191/ijmcr/v13i1.04

Neighborhood Approach for Solving One Class of Mixed Integer Non-Linear Programming

Hardi Tambunan[1], Herman Mawengkang[2]

[1]Department of Mathematics Education, Faculty of Mathematics and Science, Universitas HKBP Nommensen, Medan

[2]Department of Mathematics, University of Sumatera Utara

ARTICLE INFO	ABSTRACT
Published Online: 10 January 2025 Corresponding Author: **Hardi Tambunan**	Integer programming is not new subject in optimization. However, given its practical applicability, we face computational difficulties in solving the large-scale problems. In this paper we solve a class of mixed-integer nonlinear programming problem by adopting a strategy of releasing non-basic variables from their bounds found in the optimal continuous solution in such a way to force the appropriate non-integer basic variables to move to their neighborhood integer points.

KEYWORDS: Active constraint, neighborhood, non-linear programming

I. INTRODUCTION

Discussion and solution of the problem of mixed integer non-linear programming (MINLP) is still important to be discussed until now. That is because MINLP problems can be used as a tool for solving problems in various sectors. Some innovative applications can be seen as in the area of the synthesis process [1,2,3,4]. In the field of chemical processes [5], industrial area [6,7], network transmission [8, 9,10], engineering sector [11,12,13,14]. Application in research operations in the education sector [15,16,17].

MINLP refers to mathematical programming with continuous and discrete variables and nonlinearities in the objective function and constraints. This problem is defined by the following model.

$$Min. Z = f(x, y) \quad (1)$$

Subject to

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

$$g(x) + by \le 0 \quad (3)$$

$$x \in X \subset \mathbb{R}^n_+, y \in R \subset \mathbb{R}^n_+ \quad (4)$$

where $f: \mathbb{R}^n \to \mathbb{R}$ and $h: \mathbb{R}^n \to \mathbb{R}^p$, $g: \mathbb{R}^n \to \mathbb{R}^q$ are continuous and generally well-behaved functions defined on the n-dimensional compact polyhedral convex set $X = \{x: x \in \mathbb{R}^n, A_1 x \le a_1\}$; $U = \{y: y \in Y, integer\ A_2 y \le a_2\}$ is a discrete set, where for most applications Y is the unit hypercube $Y \varepsilon \{0,1\}^m . B, A_1, A_2$, and c, a_1, a_2 are respectively matrices and vectors of comfortable dimensions; the vectors are column vectors unless specified otherwise.

From the worst-case complexity point of view, finding a feasible MINLP solution is as hard as finding a feasible Nonlinear Programming solution, which is NP-hard [18]. In this paper we address a strategy of releasing non basic variables from their bounds, combined with the "active constrained" method and the notion of super basic for efficiently tackling a particular class of MINLP problems.

II. LITERATURE REVIEW

MINLP problem solving includes innovative approaches and related to techniques developed in mixed integer programming (MIP), and implementation of mixed integer linear programming (MILP) and nonlinear programming (NLP) algorithms. Several methods have been carried out by linearization objective functions and nonlinear constraints to find MILP masters who can estimate and represent the original MINLP problem solving, namely Outer Approximation (OA) [19,20,21]. Generalized Bender's Decomposition (GBD) [22]. Cutting Plane [23]. Branch-And-Bound (B&B) [24,25].

Heuristic approaches to solving MINLPs include Variable Neighbourhood Search [26]. Generally in an integer program, a decrease in vector gradient is used normally to find a condition that cannot be obtained optimally, even convex problems. So it is necessary to specify a certain condition for the local testing test procedure so that a suboptimal solution is obtained that is worth the best integer. Scarf [27] proposed a quantity test to determine the optimal value in integer program problems. The test is done by searching through neighbours worthy points that are proposed to see whether a nearby point is also feasible, and the results provide an improvement in the objective function.

Suppose $[\boldsymbol{\beta}]_k$that an integer point is included in a limited set of neighbourhoods $\boldsymbol{N}([\boldsymbol{\beta}]_{k)}$.. Defined a neighbourhood system that is related to$[\boldsymbol{\beta}]_k$. The integer point fulfils two requirements, viz

$$1.If\,[\beta]_j \in N([\beta]_k) than [\beta]_k \in [\beta]_i,\ j \neq k$$
$$2.N([\beta]_k) = [\beta]_k + N(O)$$

Based on the two neighbourhood requirements above, the strategy of the integration process can be carried out as follows;

(1) Suppose a non-integer component $(\boldsymbol{x}_k)$of the optimal vector $(\boldsymbol{x}_B)$,and the points that are adjacent to$(\boldsymbol{x}_k)$ is $[\boldsymbol{x}_k]$ ***dan*** $[\boldsymbol{x}_k]+\mathbf{1}$. If one of the points satisfies the constraint and gives an optimal minimum decrease in the value of the objective function, it will move to the other components, if no feasible integer solution is obtained.

(2) Suppose $[\boldsymbol{x}_k]$ the integer point of a feasible solution that meets the above requirements. if $[\boldsymbol{x}_k]+\mathbf{1} \in \boldsymbol{N}([\boldsymbol{x}_k])$, than $[\boldsymbol{x}_k]+\mathbf{1}$ is one point that is not feasible for the objective function obtained to satisfy$[\boldsymbol{x}_k]$. In this case, $[\boldsymbol{x}_k]$ is called the optimal integer feasible solution.

III. METHODOLOGY

The MINLP problem solving method is carried out using a neighborhood approach to find integer solutions and combined with active constraints

A. Derivation of The Method

The 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_{j^*} for $i \neq k$, are positive, then the corresponding element of x_B will decrease, and eventually may pass through 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 non-basic $(x_N)_{j^*}$ 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

$$\theta_1 = \min_{i\neq k|\alpha_{j^*}>0}\left\{\frac{\beta_i}{\alpha_{j^*}}\right\} \tag{5}$$

$$\theta_2 = \Delta_{j^*} \tag{6}$$

How far one can release the non-basic $(x_N)_{j^*}$ from its bound of zero, such that vector x remains feasible, will depend on the ratio test θ^* given below

$$\theta^* = \min(\theta_1, \theta_2) \tag{7}$$

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 non-basic variable, $(x_N)_{j^*}$ corresponding to any positive element of vector $\alpha_{j'}$, is given by

$$\Delta_{f'} = \frac{f_k}{\alpha_{kj}} \tag{8}$$

In order to maintain the feasibility, the ratio test θ^* is still needed. Consider the movement of a particular non basic variable, Δ, as expressed in Eq (15-8).

The only factor that one needs to calculate is the corresponding element of vector α. A vector α_j can be expressed as

$$\alpha_j = B^{-1}a_j, j = 1 \ldots, n-m \tag{9}$$

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

$$v_k^T = e_k^T B^{-1} \tag{10}$$

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

$$\alpha_{kj^*} = v_k^T a_{j^*} \tag{11}$$

The vector of reduced costs d_j is used to measure the deterioration of the objective function value caused by releasing a non-basic variable from its bound. Consequently, in deciding which non-basic should be released in the integer 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 particular non-basic variable as given in Eqns. (8) or (9), 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 non-basic variable $(x_N)_{j^*}$ so as to integer a basic variable $(x_B)_k$ may be measured by the ration

$$\left|\frac{d_k}{\alpha_{kj^*}}\right| \tag{12}$$

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

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

$$\min_j\left\{\left|\frac{d_k}{\alpha_{kj^*}}\right|\right\},\ j = 1, \ldots, n-m \tag{13}$$

From the active constraint strategy [29] and the partitioning of the constraints corresponding to basic (B), super-basic (S) and non-basic (N) variables we can write

$$\begin{bmatrix} B & S & N \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x_B \\ x_S \\ x_N \end{bmatrix} = \begin{bmatrix} b \\ b_N \end{bmatrix} \quad (14)$$

Or

$$Bx_B + Sx_S + Nx_N = b \quad (15)$$

$$x_N = b_N \quad (16)$$

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

$$x_B = B^{-1}b - (B^{-1}S)x_S - (B^{-1}N)x_N \quad (17)$$

Expression (16) indicates that the non-basic variables are being held equal to their bound. It is evident through the "nearly" basic expression of Eqn. (17), the integer search strategy discussed in the previous section, designed for MILP problem can be implemented. Particularly, we would be able to release a non-basic variable from its bound, Eq (16) and exchange it with a corresponding basic variable in the process of searching for integers, although the solution would be degenerate.

B. Integer Solution

MINLP problem solving including innovative approaches and implementations for mixed integer linear programming (MILP) algorithms. So, before we proceed to the case of MINLP problems, it is worthwhile to discuss the basic strategy of process for linear case [28].

Consider a MILP problem with the following form

Minimize z = $c^T x$ (18)

Subject to $Ax \leq b$ (19)

$x \geq 0$ (20)

x_j integer for some $j \in J$

A component of the optimal basic feasible vector $(x_B)_k$, to MILP solved as continuous can be written as $(x_B)_k = \beta_k - \alpha_{k1}(x_N)_1 - \cdots - \alpha_{kj}(x_N)_j - \cdots - \alpha_{kn} - m(x_N)_N n - m$ (21)

If $(x_B)_k$ is an integer variable and assume that β_k is not an integer, the partitioning of β_k into the integer and fractional components is that given

$$\beta_k = [\beta_k] + f_k, 0 \leq f_k \leq 1 \quad (22)$$

suppose to increase $(x_B)_k$ to its nearest integer, $([\beta] + 1)$. Based on the idea of suboptimal solutions may elevate a particular non-basic variable, say $(x_N)_{j^*}$, above its bound of zero, provided α_{kj^*}, as one of element of the vector α_{j^*}, is negative. Let Δ_{j^*} be amount of movement of the non-variable $(x_N)_{j^*}$, such that the numerical value of scalar $(x_B)_k$ is integer. Referring to Eqn. (21), Δ_{j^*} can then be expressed as

$$\Delta_{f^*} = \frac{1-f_k}{-\alpha_{kj^*}} \quad (23)$$

while the remaining non basic stay at zero. It can be seen that after substituting (22) into (23) for $(x_N)_{j^*}$ and taking into account the partitioning of β_k given in (22), we obtain

$$(x_B)_k = [\beta] + 1 \quad (24)$$

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

C. The Algorithm

After solving the relaxed problem, the procedure for searching a suboptimal but integer-feasible solution from an optimal continuous solution can be described as follows.

Let $x = [x] + f, \quad 0 \leq f \leq 1$ be the continuous solution of the relaxed problem, $[x]$ is the integer component of non-integer variable x and f is the fractional component.

Stage 1.

Step 1. Get row i^* the smallest integer infeasibility, such that $\delta_{i^*} = \min\{f_i, 1 - f_i\}$,

Step 2. Calculate $v_{i^*}^T = e_{i^*}^T B^{-1}$,

Step 3. Calculate $\sigma_{ij} = v_{i^*}^T a_j$,

Step 4. Calculate $\alpha_{j^*} = B^{-1}\alpha_{j^*}$, i.e. solve $B\alpha_{j^*} = \alpha_{j^*}$ for α_{j^*},

Step 5. Do Ratio test,

Step 6. Exchanging basis,

Step 7. Stop if there are no other non-integer basic variable. Otherwise repeat from step 1.

Stage 2. Do integer line search to improve the integer feasible solution.

IV. CONCLUSIONS

In this paper we solve a class of mixed-integer nonlinear programming problem by adopting a strategy of releasing non-basic variables from their bounds found in the optimal continuous solution in such a way to force the appropriate non-integer basic variables to move to their neighborhood integer points. A study of the search strategy for integer solutions has been made. The number of steps to look for an integer solution will be limited if the number of integer variables contained in the problem is also limited. Thus, the computational time for the process of finding an integer solution does not always depend on the number of integer variables, maybe many integer variables have integer values in the continuous optimal solution

REFERENCES

1. Duran, M.A., and Grossmann, I. E. 1986a. An Outer-Approximation Algorithm for A Class of Mixed-Integer Nonlinear Programs. Mathematical Programming 36, 307-339.
2. Kocis, G. R; and Grossmann, I.E. 1988. Global Optimizzation of Nonconvex MINLP Prolems in Process Synthesis. Ind. Eng. Chem 27, 1407-1422.
3. Floudas, C.A. 1989. Global Optimum Search for Nonconvex NLP and MINLP Problems. Computers Chem. Eng. 13, 1117-1132.
4. Ryoo, H. S., and Sahinidis, N.V. 1995. Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design. Computers Chem. Vol.19. No. 5, 551-566.

5. Grossman, I. E., and Floudas, C. A. 1987. Active Constraint Strategy for Flexibility Analysis in Chemical Proses. Comput.Chem. engng, Vol. 11, No. 6, 675 - 693.
6. Harjunkoski, I; Westerlund, T; and Porn, R. 1999. Numerical and Environmental Considerations on a Complex Industrial Mixed Integer Non-Linear Programming (MINLP) Problem. Comput. & Chem. Eng.,23, 1545-1561.
7. Kallrath, J. 2005. Solving Planning and Design Problems in the Process Industry Using Mixed Integer and Global Optimization. Annals of Operations Research 140, 339-373.
8. Cai. X., et al. 2001. Solving Large Non-Convex Water Resources Management Models Using Genenalized Benders Decomposition. INFORMS, Vol. 49, No. 2, 235–245
9. Bragalli, C; et al. 2006. An MINLP Solution Method for a Water Network Problem. Springer: Lecture Notes in Computer Science vol. 4168. 696-707.
10. Martin, A; Moller, M; and Moritz, S. 2006. Mixed Integer Models for the Stationary Case of Gas Network Optimization. Math. Program., 105,563-582.
11. Grossman, I.E., and Sahinidis, N.V (eds). 2002. Special Issu on Mixed Integer Programming and Its Application to Engineering, Part I, Optim. Eng., 3 (4)
12. Kravanja, S., and Zula, T. 2010. The MINLP Approach to Structural Synthesis. Challenges, Opportunities and Solutions in Structural Engineering and Construction. Taylor & Francis Group, London,
13. Fugenschuh, A, et al. 2010. Mixed-Integer Nonlinear Problems in Transportation Applications. 2nd International Conference on Engineering Optimization. September 6-9, 2010, Lisbon, Portugal
14. Guerra, A., Newman, A.M., and Leyffer, S. 2011. Concrete Structure Design Using Mixed-Integer Nonlinear Programming with Complementarity Constraints. SIAM J. Optim, Vol 21, No.3, 833-863.
15. Junoh, A. K, *et al.* 2012. Classification of Examination Marks According to Bloom’s Taxonomy by Using Binary Linear Programming, *IACSIT Press*, Singapore, *IPCSIT*, vol. 36, 20-25
16. Tambunan, H. 2016. Mathematical Model for Mapping Students Cognitive Capability, International Journal of Evaluation and Research in Education 5(3), 221-226.
17. Tambunan, H & Mawengkang, H. 2018. Integer Linear Programming Approach for Detection Learning Outcomes Achievement. Far East Journal of Mathematical and Sciences, 5(1), 95-109.
18. Mawengkang, H., and Murtagh, B. A. 1986. Solving Nonlinear Integer Programs with Large-Scale Optimization Software. *Annals of Operations Research* 5425-437
19. Duran, M. A. and Grossmann, I.E. 1986b. A Mixed-Integer Nonlinear Programming Algorithm for Process Systems Synthesis. AIChE Journal Vol 32, No.4, 592-606
20. Fletcher, R., and Leyffer, S. 1994. Solving Mixed Integer Nonlinear Programs by Outer Approximation. Mat. Program, 66, 327-349.
21. Kesavan, P; et al. 2004. Outer Approximation Algorithms for Separable Nonconvex. Math. Program, Ser. A 100, 517–535
22. Geoffrion, A. M. 1972. A Generalized Benders Decomposition, J. Op-tim. Theory Appl, 10 (4), 237–260.
23. Westerlund, T; and Petersson, F. 1995. A Cutting Plane Method for Solving Convex MINLP Problems. Computers Eng., 19, 131-136.
24. Gupta, O.K; and Ravindran, V. 1985. Branch and Bound Experiments in Convex Nonlinear Integer Programming. Mangement Science, 31, 1533-1546.
25. Belotti, P.,Lee, J.,Liberti,L., Margot,F., and Wachter,A. 2009. Branching and Bounds Tightening Techniques for non-Convex MINLP, *Optimization Methods and Software,* 24(**4**):597-634,
26. Liberti, L., Nannicini, G., and Mladenovic,N, 2009. A Good Recipe for Solving MINLPs. In: V. Maniezzo, T. Stutzle, S.Voss (eds.) Metaheuristics: *Hybridizing metaheuristics and Mathematical Programming. Annals of Information Systems.* Springer, vol. 10,. 231-245.
27. Scraf, H.E. (1986). Testing for Optimality In the Absence of Convexity. @ Cambridge University Press. 117-133.
28. Tambunan, H., and Mawengkang, H. 2016. Solving Mixed Integer Non-Linear Programming Using Active Constraint. Global Journal of Pure and Applied Mathematics. Vol. 12, Number 6, 5267–5281