9 (2) July 2023 112-121 ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
Register: *Jurnal Ilmiah Teknologi Sistem Informasi*

Contents lists available at www.journal.unipdu.ac.id

Register

Journal Page is available to www.journal.unipdu.ac.id/index.php/register

Research article

Developing an Enhanced Algorithms to Solve Mixed Integer Non-Linear Programming Problems Based on a Feasible Neighborhood Search Strategy

Mochamad Wahyudi [a,*], *Firmansyah Firmansyah* [b], *Hengki Tamando Sihotang* [c], *Lise Pujiastuti* [d], *Herman Mawengkang* [e]

[a,b] *Departement of Computer Science, Universitas Bina Sarana Informatika, Jakarta, Indonesia*
[c] *Department of Operations Research, Institute of Computer Science, Medan, Indonesia*
[d] *Departement of Information System, STMIK Antar Bangsa, Tangerang, Indonesia*
[e] *Faculty of Mathematics and Natural Science, Universitas Sumatera Utara, Medan, Indonesia*
email: [a,*] wahyudi@bsi.ac.id, [b] firmansyah.fmy@bsi.ac.id, [c] hengki_tamando@yahoo.com, [d] lise.pujiastuti@gmail.com, [d] mawengkang@usu.ac.id
* Correspondence

ARTICLE INFO

Article history:
Received 24 June 2023
Revised 20 July 2023
Accepted 1 August 2023
Available online 19 August 2023

Keywords:
Algorithm;
Integer Programming;
Large-Scale Problems;
Neighbourhood Search;
Optimization;

Please cite this article in IEEE style as:
M. Wahyudi, F. Firmansyah, H.T. Sihotang, L. Pujiastuti, and H. Mawengkang, "Developing an Enhanced Algorithms to Solve Mixed Integer Non-Linear Programming Problems Based on a Feasible Neighborhood Search Strategy," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 9, no. 2, pp. 112-121, 2023.

ABSTRACT

Engineering optimization problems often involve nonlinear objective functions, which can capture complex relationships and dependencies between variables. This study focuses on a unique nonlinear mathematics programming problem characterized by a subset of variables that can only take discrete values and are linearly separable from the continuous variables. The combination of integer variables and non-linearities makes this problem much more complex than traditional nonlinear programming problems with only continuous variables. Furthermore, the presence of integer variables can result in a combinatorial explosion of potential solutions, significantly enlarging the search space and making it challenging to explore effectively. This issue becomes especially challenging for larger problems, leading to long computation times or even infeasibility. To address these challenges, we propose a method that employs the "active constraint" approach in conjunction with the release of nonbasic variables from their boundaries. This technique compels suitable non-integer fundamental variables to migrate to their neighboring integer positions. Additionally, we have researched selection criteria for choosing a nonbasic variable to use in the integerizing technique. Through implementation and testing on various problems, these techniques have proven to be successful.

1. Introduction

An optimization problem is based on a collection of parameters and independent variables, which often include conditions on the acceptable values of the variables. These limitations are referred to as problem constraints. Another essential aspect of the optimization problem is the "good" measure, also known as the objective function, which depends on the issue variables. The objective function represents what needs to be optimized. To achieve the best value for the objective function, a collection of variable values must satisfy the optimization problem. A standard expression commonly used to represent and facilitate problem-solving for optimization issues involves the use of constraint function 'g' and objective function 'f,' both of which are real-valued scalar functions and frequently employed in this context.

The algorithm developed, based on the feasible neighborhood search strategy, presents an innovative approach to solving mixed-integer nonlinear programming problems. By carefully considering the interaction between integer variables and nonlinear functions, this algorithm aims to efficiently explore the solution space and discover optimal or near-optimal solutions. The strategy

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

involves iteratively examining the neighborhood of feasible solutions and updating the current solution using local search procedures. This approach not only considers the objective function but also ensures adherence to the problem's constraints. Many optimization problems involve the inclusion of integer or discrete variables. These variables can be used in decision models with binary options, such as determining investments in portfolio problems, or they can represent whole units, such as the number of employees required. Additionally, continuous variables like labor time and manufacturing volume might also be present. Nonlinearity may arise in optimization models when dealing with physical qualities, such as fluid concentration balancing, or when addressing problems related to economies of scale. The Mixed Integer Nonlinear Programming (MINLP) model serves as the optimization model for nonlinear problems that incorporate both discrete and continuous variables.

In general, MINLP problems refer to mathematical formulations that combine discrete and continuous variables with nonlinearity in both constraints and the objective function. This paper focuses on a specific subclass of MINLP issues, where discrete variables are distinguished from continuous variables based on their linearity and dependencies. When expressed in algebraic form, a MINLP issue takes on the following most basic form:

$$\begin{aligned} &\text{Min } Z = f\ (x, y) \\ &s.t.\, g_j(x, y) \leq 0 \quad j \in J \\ &x \in X, y \in Y \end{aligned} \tag{1}$$

In the equation system above, we have nonconvex functions denoted as f(·) and g(·), with inequalities represented by the index J. The variable x is continuous and y represents discrete variables. For simplicity, we assume that the set X is a convex compact set, possibly taking a form such as $X = \{x \mid x \in \boldsymbol{R}^n,\ Dx \leq d,\ x^L \leq x \leq x^U\}$; The discrete set Y belongs to a polyhedral integer set, $Y = \{y \mid y \in \boldsymbol{Z}^m,\ Ay \leq a\}$, and is limited to 0-1 values for a large number of applications, $y \in \{0,1\}^m$. It is important to note that *f(·), g(·)* the objective function f(·) and constraint function g(·) are predominantly linear in *y*. For instance, they may involve logic constraints and fixed cost charges.

The MINLP algebraic form reveals that models may consist of convex or nonconvex functions. For Convex MINLP models, various methods can be employed to address the issues [1], including nonlinear-programming-based branch-and-bound [2]–[4], multi-tree methods (outer approximation [5][6], Generalized Benders Decomposition [7], extended cutting-plane method [8]), single-tree methods (LP/NLP-based B&B) [9], and presolve techniques (coefficient tightening for MINLP) [10]. Cutting-plane methods, such as mixed-integer rounding cuts, perspective cuts, disjunctive cuts, Gomory cuts, and lift-and-project cuts, are popular for solving convex MINLP models [1]. A comprehensive review of techniques applied to convex MINLP models can be found in [11]–[14].

Nonconvex MINLP models present challenges due to the presence of nonconvexities in both objective functions and constraints. Consequently, even when integrality restrictions are relaxed on integer decision variables, obtaining a convex relaxation efficiently in a B&B framework for the resulting nonconvex feasible region requires additional effort [1]. Despite the added complexity of approximating an efficient convex relaxation for the feasible region of an MINLP, many techniques specialized for MINLPs yield numerous local optima without guaranteeing global optimality [1]. Popular techniques used to solve nonconvex MINLP models include piecewise linear modeling, generic relaxation strategies, spatial B&B, and relaxation of structured nonconvex sets [11]–[15]. Heuristic techniques are also widely employed to solve nonconvex MINLP models [1], such as mixed-integer-based rounding, feasibility pump, undercover, relaxation enforced neighborhood search, and diving.

Engineers have developed the mathematical approach of optimization to create products and buildings inexpensively and effectively [16]. Numerous optimization techniques, such as Genetic Algorithms (GA)[17], Generalized Reduced Gradient (GRG), and Sequential Quadratic Programming (SQP), have been established to handle a range of issues [18]. To effectively utilize these methodologies, several tools like Matrix Laboratory (MATLAB) and Microsoft Excel Solver have been employed[19]. GRG and SQP are recognized as two of the top deterministic optimization techniques, as revealed in comparative research [20]. Conversely, GA, which is based on evolutionary principles inspired by nature, is considered the best stochastic approach [21]–[23].

The challenge of optimizing a cost function while solving an equation system arises in optimization problems with linear equality constraints. The reduced gradient approach, for example

 http://doi.org/10.26594/register.v9i2.3706

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

[24], solves a series of subproblems with implicitly lowered variables. This is one of several algorithms established for tackling linearly-constrained optimization problems. These simplified issues are created by expressing a subset of variables, referred to as the "basic" variables, in terms of additional variables under the influence of linear constraints. The reduced gradient approach, developed by Wolfe [25], can be seen as an expansion of the conventional simplex method. The convex simplex approach is known to fail by converging to a non-optimal point. Reduced gradient algorithms are applied in engineering difficulties, such as water distribution [26] power flow [27], machine learning [28], and optimal control [29]. In this study, the reduced gradient approach has been suggested as one of the key methods for tackling limited optimization issues.

The MINLP model finds numerous applications in the process industry, management science, finance, engineering fields, and more. It covers concerns in process flow diagrams, portfolio choice, batch processing in chemical engineering (which entails mixing, centrifuge separation, and reaction), and the best way to construct water or gas transmission networks. It is also relevant in the fabrication of VLSI chips, automobiles, and airplanes [30] [31], showcasing a wide array of MINLP applications. Research and development in MINLP solver technology have been driven by demands in a wide range of fields, especially for tackling large-scale, highly nonlinear, and combinatorial problems. Various approaches have been explored in the literature for solving MINLPs since the early 1980s, including Outer Approximation (OA) methods [24][32], Extended Cutting Plane methods [8][33], Branch-and-Bound (B&B) [34][35], and Generalised Bender's Decomposition (GBD) [36][37]. These methods often rely on the sequential resolution of closely connected NLP issues. By removing the integrality restrictions for discrete variables, B&B, for instance, creates a pure continuous NLP issue (also known as the relaxed MINLP or RMINLP). Additionally, each node of the growing B&B tree represents an RMINLP solution with modified limits on the discrete variables. In response to a challenge stated by Murtagh and Sugden [1], Vassilev and Enova [38] present an approximation approach.

2. Materials and Methods

This work has focused on a group of algorithms in which the search direction of the active constraint surface is defined to lie within the domain of the Z matrix, which is orthogonal to the conventional constraint matrix. Hence, Z is represented as an n x s matrix, satisfying the condition $\hat{\boldsymbol{A}}\boldsymbol{Z} = \boldsymbol{0}$, when $\hat{\boldsymbol{A}}\boldsymbol{x} = \hat{\boldsymbol{b}}$ for the current set of n-s active constraints.

The application of the Generalized Reduced Gradient method to solve the model starts by employing the Langrange function, and then follows the steps outlined in the algorithm.

The algorithm proceeds as follows:

Step 1. (Convergence test in the current subspace). If $||h|| >$ TOLRG, proceed to step 3.

Step 2. ("PRICE" - estimate Lagrange multiplier, add one superbase).

a) Calculate $\lambda = g_N - N^T\pi$

b) Choose $\lambda_{q_1} < -$TOLDJ ($\lambda_{q_2} > +$TOLDJ), where λ is the greatest element corresponding to variables in its upper (lower) limit. If not, STOP; Kuhn-Tucker's conditions are obeyed, and an optimal solution has been obtained.

c) If not,

  (i) Choose q = q_1 or q_2 accordingly with $|\lambda_{q1}| = \max(|\lambda_{q1}|, |\lambda_{q2}|)$;

  (ii) Add a_q as a new column S;

  (iii) Add λ_q as a new element h;

  (iv) Add the appropriate new column to R.

d) Add s by 1.

Step 3. (Calculate search direction, $p = Zp_s$)

a) Complete $R^TRp_s = -h$.

b) Complete LU$p_B = -Sp_s$.

c) Set $p = \begin{bmatrix} p_B \\ p_S \\ 0 \end{bmatrix}$

Step 4. (Ratio Test, "CHUZR")

a) Test $\alpha_{max} \geq 0$, where α is the largest value of $x + \alpha p$ that remains feasible.

b) If $\alpha_{max} = 0$, go to step 7.

http://doi.org/10.26594/register.v9i2.3706

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

Step 5. (Line search)

a) Find α, the optimal α^*, where

$$F(x + \alpha^* p) = \min_{0<\theta\leq\alpha_{max}} f(x + \theta p)$$

b) Update x to $x + \alpha p$ and set f and g and g to their values in the new x.

Step 6. (Calculate reduced gradient, $\bar{h} = Z^T g$).

a) Complete $U^T L^T \pi = g_B$.

b) Calculate, $\bar{h} = g_s - S^T \pi$. the new reduced gradient.

c) Update R using $R^T R$ variable-metric recursion with α, p_s and the reduced gradient change, $\bar{h} - h$.

d) Set $h = \bar{h}$.

e) If $\alpha < \alpha_{max}$, return to step 1. No additional constraints were found so they continue in this subspace.

Step 7. (Adjust the base if necessary; remove one superbase). If for some p ($0 < p \leq m + s$) variables corresponding to column p from [B S] has reached their limits with $\alpha < \alpha_{max}$:

a) If the base variable reaches its limit ($0 < p \leq m$),

(i) Swap the p-th column for the q-th of $\begin{bmatrix} B \\ X_B^T \end{bmatrix}$ and $\begin{bmatrix} S \\ X_S^T \end{bmatrix}$, respectively, such that B remains nonsingular, and q is chosen respectively (this will require vector πp that satisfies $U^T L^T \pi_p = e_p$);

(ii) Adjust L, U, R and π accordingly to replicate this change in B;

(iii) Calculate the new gradient $h = g_s - S^T \pi$;

(iv) Go to (c).

b) Else, the superbase variable reaches its limit ($m < p \leq m + s$). Determine $q = p - m$.

c) Make the q-th variable in S nonbasic at the appropriate bounds, thus:

(i) Remove the q-th column from $\begin{bmatrix} S \\ X_S^T \end{bmatrix}$ and $\begin{bmatrix} R \\ h^T \end{bmatrix}$.

(ii) Return R to the triangular matrix.

d) Subtract s by 1 and go to step 1.

3. Results and Discussion

3.1. Computational Experience 1

Portfolio optimization is a process that involves strategically allocating assets within an investment portfolio to achieve specific objectives. It is a mathematical approach used to determine the ideal combination of investments that can provide the best balance between risk and return. The pioneering study by Markowitz [39] was established on the principle that investors seek higher anticipated return and lower volatility. In this context, let S represent the set of investments, and each security is denoted by $|S|$. The level of expected return is represen by and ρ. The model can be formulated as a quadratic programming problem, as shown in the equations below:

$$\min \sum_{i\in S} \sum_{j\in S} \sigma_{ij} x_i x_j, \tag{2}$$

$$\sum_{i\in S} r_i x_i = \rho, \tag{3}$$

$$\sum_{i\in S} x_i = 1, \tag{4}$$

$$x_i \geq 0, \qquad i \in S, \tag{5}$$

In the equation above, security i represents the amount of x_i money invested, $r_i = E(R_i)$ with R_i denotes the expected return of the random variable representing the return of security i, and σ_{ij} and j respectively represent the covariance among security returns and security. The common assumption adopted is that the rates of return follow a multivariate normal distribution. The quadratic mixed integer programs of the portfolio optimization problem is presented as a subroutine called CALCFG.

```
SUBROUTINE CALCFG(MODE,N,X,F,G,NSTATE,NPROB)
IMPLICIT REAL*8(A-H,O-Z)
```

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

```
        DIMENSION X(N),G(N)
OBJECTIVE FUNCTION
F = X(1)**2+36.0*X(2)**2+X(3)**2+X(5)**2+ 6.0*X(1)*X(2)+0.2*X(1)*X(3)+X(1)*X(5)+X(2)*X(3)+X(3)*X(5)
PARTIAL DERIVATIVES
        G(1) = 2.0*X(1)+6.0*X(2)+0.2*X(3)+X(5)
        G(2) = 72.0*X(2)+6.0*X(1)+X(3)
        G(3) = 2.0*X(3)+0.2*X(1)+X(2)+X(5)
        G(4) = 0.0
        G(5) = 2.0*X(5)+X(1)+X(3)
        RETURN
        END
```

3.2. Computational Experience 2 – System Reliability Optimization

3.2.1. Introduction

A system must be successfully developed during the design phase while also being created to fulfill its functional requirements. Regarding the latter aspect, various system constraints must be considered to ensure dependability in the system. According to Fair et al [40], historically, there have been two approaches to increasing the dependability of a multi-stage system.

a) Redundancy: This involves using more parts or assemblies than what is necessary for the current system function.
b) Overdesign: The components of the system are significantly larger in most of the design dimensions.

Both of these approaches only slightly increase reliability at the cost of consuming crucial resources.

According to Misra and Ljubojevic [41], the ideal system reliability problem involves identifying the ideal component dependability and the ideal number of redundancies needed to achieve the greatest possible system reliability performance [41]. In the domains of reliability theory and application, redundancy is an intriguing topic that can significantly improve a system's reliability. The use of various operations research methodologies to address the issue of optimum redundancy allocation has also received considerable attention [42]. We focus on problems where the reliability component is fixed, and the goal is to determine the best amount of redundancies to be added at each stage.

3.2.2. Problem Formulation

The n-stage problem and its associated system reliability can often be mathematically stated as shown below. The goal is to maximize the nonlinear function of the variables.

$$R = \prod_{j=1}^{n} [1 - (1 - r_j)^{x_j}] \tag{6}$$

as per constraints

$$\sum_{j=1}^{n} g_j\,(x_j) \le b_i, i = 1, \dots, m \tag{7}$$

$$l \le x \le u \tag{9}$$

$$x_j \text{ is a positive integer, } j = 1, \dots, n \tag{10}$$

Where reliability factors are represented as r_j, $j = 1, \dots, n$. The g_j constraint functions are required to be linear.

3.2.3. Numerical Example

Consider Rein Luus [43] formulation of the 15-stage problem with 2 linear constraints.

$$\text{Maximize } R = \sum_{j=1}^{15} \{1 - (1 - r_j)^{x_j}\} \tag{11}$$

Subject to

 http://doi.org/10.26594/register.v9i2.3706

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

$$\sum_{j=1}^{5} c_j x_j \leq 400 \tag{12}$$

$$\sum_{j=1}^{15} w_j x_j \leq 414 \tag{13}$$

$$x_j \text{ is a positive integer } \forall j \tag{14}$$

Here, the *cost factors* c_j ($j = 1, \ldots, 15$) are given as 5, 4, 9, 7, 7, 5, 6, 9, 4, 5, 6, 7, 9, 8, and 6. The *weight factors* w_j ($j = 1, \ldots, 15$) are provided as 8, 9, 6, 7, 8, 8, 9, 6, 7, 8, 9, 7, 6, 5, and 7. The *reliability factors,* r_j ($j = 1, \ldots, 15$) are listed as 0.9, 0.75, 0.65, 0.80, 0.85, 0.93, 0.78, 0.66, 0.78, 0.91, 0.79, 0.77, 0.67, 0.79, and 0.67 and. For each x_j, $j = 1, \ldots, 15$, the initial value is set to 2.0.

3.2.4. Solution Procedure

As observed, the problem is a straightforward nonlinear integer programming issue. Such a class of problems cannot be handled by the integerizing technique, which was specifically designed for mixed-integer programs. We cannot expect any variable to be zero-bounded, particularly due to the nature of the objective function (4) and constraint (7). However, since there are fewer rows (m) than variables (n), there will be an increased number of super basic variables compared to basic variables at the best continuous solution. This scenario is evident from the limitations (5) and (6). As a result, the integerizing procedure will not be able to utilize enough non-integer slack variables.

As a result, we approach this category of problems differently. We adopt a methodology that involves studying a reduced issue where the majority of variables are held constant as integers, and only a small fraction is allowed to fluctuate in discrete increments. The procedure can be summarized as follows:

Step 1. Ignore the integrality constraints as you solve the issue.

Step 2. Utilize heuristic rounding to obtain a (sub-optimal) integer-feasible solution from the continuous solution..

Step 3. Divide the integer variables set I to I_1, based on their boundaries for the nonbasic variables in the continuous solution. Hence, they become I_2, $I = I_1 + I_2$.

Step 4. Conduct an objective function search, maintaining I_1 nonbasic variable and allow discrete value changes in I_2.

Step 5. Examine the minimized cost of variables in from the solution obtained in step 4. If any variables are relieved from their boundaries, they are added to set I_2 and step 4 is repeated. Otherwise, the procedure is terminated.

The aforementioned process provides a framework for creating specialized approaches for specific problem classes. Step 2 of the above technique can be applied to the dependability problem, Eqns, (4)–(7), by simply rounding down the continuous optimum solution. Due to the nature of limitations (5) and (6), the resulting solution is feasible. In this case, we can consider the set I_1 to be empty since none of the variables are allowed to have values equal to zero. As a result, in this instance, we can combine steps 4 and 5. By examining the reduced costs of the variables, we can make discrete adjustments to the values obtained from step 2.

Table 1 shows the outcome of this approach for solving the aforementioned system dependability problem. As can be observed, our results are in agreement with those of Rein Luus [43]. Our approach significantly reduces the overall computing time to 7.36 seconds, which is faster than Luus' 7.8 seconds. On the other hand, Ong (1984) achieved an optimal finding of 0.9456 for the problem, but unfortunately, he did not report the computational time.

http://doi.org/10.26594/register.v9i2.3706

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

Table 1. Reliability Problem Results

Stage j	Allocation (x_j)	
	Step 1	Step 4
1	2.64483	3.0
2	3.94357	4.0
3	5.32838	5.0
4	3.64478	3.0
5	4.10837	3.0
6	2.34418	2.0
7	3.66858	4.0
8	5.21032	5.0
9	3.83400	4.0
10	2.54777	3.0
11	3.57855	3.0
12	3.92977	4.0
13	5.09453	5.0
14	3.95416	5.0
15	4.95609	5.0
Obj. Value (R)	0.95401	0.94475

3.3. Computational Experience 2 – Synthesis Problem of a Process System

3.3.1. Mathematical problem description

The design problem is often mathematically defined to work with discrete and continuous variables methodologies, concurrently determining the optimal structural and operational constraints for a system to meet specific design requirements. These variables are the defined decision variables.

The variable y is binary and is associated with each process unit to indicate whether it is included in the final ideal construction or not. The continuous variables x describe process characteristics such as material flow rates. The overall objective is to minimize yearly costs, which include both investment and operating costs.

Minimize:

$$
\begin{aligned}
F = {} & 5y_1 + 8y_2 + 6y_3 + 10y_4 + 6y_5 + 7y_6 + 4y_7 + 5y_8 - 10x_3 - 15x_5 + 15x_{10} + 80x_{17} \\
& + 25x_{19} + 35x_{21} - 40x_9 + 15x_{14} - 35x_{25} + \exp(x_3) + \exp\left(\frac{x_5}{1.2}\right) \\
& - 65\ln(x_{10} + x_{17} + 1) - 90\ln(x_{19} + 1) - 80\ln(x_{21} + 1) + 120
\end{aligned}
$$

Subject to:

$$
\begin{aligned}
& -1.5\ln(x_{19} + 1) - \ln(x_{21} + 1) - x_{14} \le 0 \\
& -\ln(x_{10} + x_{17} + 1) \le 0 \\
& -x_3 - x_5 + x_{10} + 2x_{17} + 0.8x_{19} + 0.8x_{21} - 0.5x_9 - x_{14} - 2x_{25} \le 0 \\
& -x_3 - x_5 + 2x_{17} + 0.8x_{19} + 0.8x_{21} - 2x_9 - x_{14} - 2x_{25} \le 0 \\
& -2x_{17} - 0.8x_{19} - 0.8x_{21} + 2x_9 + x_{14} + 2x_{25} \le 0 \\
& -0.8x_{19} - 0.8x_{21} + x_{14} \le 0 \\
& -x_{17} + x_9 + x_{25} \le 0 \\
& -0.4x_{19} - 0.4x_{21} + 1.5x_{14} \le 0 \\
& 0.16x_{19} + 0.16x_{21} - 1.2x_{14} \le 0 \\
& x_{10} - 0.8x_{17} \le 0 \\
& -x_{10} + 0.4x_{17} \le 0 \\
& \exp(x_3) - 10y_1 \le 1 \\
& \exp(x_3) - 10y_2 \le 1 \\
& x_9 - 10y_3 \le 0 \\
& 0.8x_{19} + 0.8x_{21} - 10y_4 \le 0 \\
& 2x_{17} - 2x_9 - 2x_{25} - 10y_5 \le 0 \\
& x_{19} - 10y_6 \le 0 \\
& x_{21} - 10y_7 \le 0 \\
& x_{10} + x_{17} - 10y_8 \le 0 \\
& y_1 + y_2 = 1 \\
& y_4 + y_5 \le 1 \\
& -y_4 + y_6 + y_7 = 0 \\
& y_3 - y_8 \le 0
\end{aligned}
$$

 http://doi.org/10.26594/register.v9i2.3706

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

$0 \leq y_j \leq 1$ and integer for $j = 1, \cdots, 8$

$l \leq x \leq u$

$x = x_j$: $(j = 3,5,10,17,19,21,9,14,25) \in R^9$

$l^T = (0,0,0,0,0,0,0,0,0)$, $u^T = (2,2,1,2,2,2,2,1,3)$

The aforementioned formulation consists of eight binary variables, nine bounded continuous variables, and 23 inequality constraints. Four of the inequalities, as well as the objective function, exhibit nonlinearities. To handle the nonlinear constraints, it is necessary to insert a subroutine called CALCON, which is defined as follows:

SUBROUTINE CALCON (Mode, M, N, Njac, X, F, G, NSTATE, NPROB)

3.3.2. Discussion of the Results

NLP software was used to find the continuous best solution. In the continuous solution, there is only one binary variable with integer values. However, in the superbasic set, there is a binary variable with a non-integer value. To address this, we employed a truncation approach and relocated this variable to the nearest integer while keeping its superbasis. After this shift, we needed to determine whether the associated basic variables remained feasible. Using our suggested integerizing method, we then converted the remaining binary non-integer variables to integers. Table 1 above displays both the integer and continuous solutions to the synthesis issue.

Table 2. Synthesis Problem Results

Variables	Activity in Cont. Solns.	Activity after integ. Process
x_3	1.90293	0.0
x_5	2.0	2.0
x_{10}	0.52752	0.46784
x_{17}	0.65940	0.58480
x_{19}	2.0	2.0
x_{21}	1.08333	0.0
x_9	0.65940	0.0
x_4	0.41111	0.26667
x_{25}	0.0	0.58480
y_1	0.57055	0.0
y_2	0.42945	1.0
y_3	0.06594	0.0
y_4	0.30833	1.0
y_5	0.0	0.0
y_6	0.2	1.0
y_7	0.10833	0.0
y_8	0.11869	1.0
Obj.value(F)	15.08219	68.00974

The objective conclusion agrees with the one reached by [32].

4. Conclusion

The "active constraint" technique, the concept of super basic variables, and the scheme of releasing nonbasic variables from their constraints have been developed to effectively address mixed-integer nonlinear programming issues. This technique is used to make suitable non-integer fundamental variables migrate to their nearby integer positions once a problem is solved by ignoring the integrality criteria. The technique described in this study has been computationally tested, and the results show that it is a feasible solution for large-scale problems.

Author Contributions

M. Wahyudi: Conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, software, supervision, validation, visualization, writing – original draft, and writing - review & editing. F. Firmansyah: Investigation, software, and writing – review & editing. H. T. Sihotang: Data curation, investigation, methodology, supervision, and writing - review & editing. L. Pujiastuti: Investigation, software, and writing - review & editing. H. Mawengkang: Formal analysis, investigation, methodology, supervision, and writing - review & editing.

http://doi.org/10.26594/register.v9i2.3706

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

References

[1] B. A. Murtagh and S. J. Sugden, “A direct search approach to nonlinear integer programming,” *Optim. Methods Softw.*, vol. 4, no. 3, pp. 171–189, 1994.

[2] Y. Utami, D. Vinsensia, A. Nissa, and S. Sulastri, “Forecasting the Sum of New College Students with Linear Regression Approach,” *J. Tek. Inform. CIT Medicom*, vol. 14, no. 1, pp. 10–15, 2022.

[3] P. Bonami *et al.*, “An algorithmic framework for convex mixed integer nonlinear programs,” *Discret. Optim.*, vol. 5, no. 2, pp. 186–204, 2008.

[4] J. Li *et al.*, “The application of nonlinear programming on ration formulation for dairy cattle,” *J. Dairy Sci.*, vol. 105, no. 3, pp. 2180–2189, 2022.

[5] P. Muts and I. Nowak, “Towards multi-tree methods for large-scale global optimization,” in *Optimization of Complex Systems: Theory, Models, Algorithms and Applications*, 2020, pp. 498–506.

[6] P. Muts, I. Nowak, and E. M. T. Hendrix, “On decomposition and multiobjective-based column and disjunctive cut generation for MINLP,” *Optim. Eng.*, vol. 22, pp. 1389–1418, 2021.

[7] A. Msabawy and F. Mohammad, “Continuous sizing optimization of cold-formed steel portal frames with semi-rigid joints using generalized reduced gradient algorithm,” *Mater. Today Proc.*, vol. 42, pp. 2290–2300, 2021.

[8] T. Westerlund and F. Pettersson, “An extended cutting plane method for solving convex MINLP problems,” *Comput. Chem. Eng.*, vol. 19, pp. 131–136, 1995.

[9] C. Coey, M. Lubin, and J. P. Vielma, “Outer approximation with conic certificates for mixed-integer convex problems,” *Math. Program. Comput.*, vol. 12, no. 2, pp. 249–293, 2020.

[10] R. Quirynen and S. Di Cairano, “Tailored Presolve Techniques in Branch-and-Bound Method for Fast Mixed-Integer Optimal Control Applications,” *arXiv Prepr. arXiv2211.12700*, 2022.

[11] M. Bussieck and S. Vigerske, “MINLP solver software,” Berlin, Germany, 691, 2010. doi: 0296-matheon-6848.

[12] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, “Branching and bounds tighteningtechniques for non-convex MINLP,” *Optim. Methods Softw.*, vol. 24, no. 4–5, pp. 597–634, 2009.

[13] M. A. Lasemi, A. Arabkoohsar, A. Hajizadeh, and B. Mohammadi-Ivatloo, “A comprehensive review on optimization challenges of smart energy hubs under uncertainty factors,” *Renew. Sustain. Energy Rev.*, vol. 160, p. 112320, 2022.

[14] Z. Zhang, J. Wang, and J. Li, “Lossless convexification of nonconvex MINLP on the UAV path-planning problem,” *Optim. Control Appl. Methods*, vol. 39, no. 2, pp. 845–859, 2018, doi: https://doi.org/10.1002/oca.2380.

[15] J. Lee and S. Leyffer, *Mixed integer nonlinear programming*, vol. 154. USA: Springer Science & Business Media, 2011.

[16] E. Salajegheh, “Optimum design of plate structures using three-point approximation,” *Struct. Optim.*, vol. 13, pp. 142–147, 1997.

[17] P. M. F. Marsoit, “Quantum-inspired fuzzy genetic programming for enhanced rule generation in complex data analysis,” *Int. J. Enterp. Model.*, vol. 15, no. 3, pp. 176–186, 2021.

[18] M. Pyrz and J. Zawidzka, “Optimal discrete truss design using improved sequential and genetic algorithm,” *Eng. Comput.*, vol. 18, no. 8, pp. 1078–1090, 2001.

[19] S. Adadurov, Y. Fomenko, A. Khomonenko, and A. Krasnovidov, “Integration of the matlab system and the object-oriented programming system c# based on the microsoft com interface for solving computational and graphic tasks,” in *Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020, Volume 1 9*, 2020, pp. 581–589.

[20] J. A. Dauda, S. A. Rahmon, I. A. Tijani, F. Mohammad, and W. O. Okegbenro, “Design optimisation of reinforced concrete pile foundation using generalised reduced gradient algorithm,” *Front. Eng. Built Environ.*, vol. 2, no. 3, pp. 133–153, 2022.

[21] O. Yeniay, “A comparative study on optimization methods for the constrained nonlinear programming problems,” *Math. Probl. Eng.*, vol. 2005, pp. 165–173, 2005.

[22] K. S. Han and K. G. Ortizan, “Optimizing robust routing and production planning in stochastic supply chains: Addressing uncertainty of timing and demand for enhanced resilience and efficiency,” *Int. J. Enterp. Model.*, vol. 16, no. 2, pp. 106–114, 2022.

 http://doi.org/10.26594/register.v9i2.3706

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
9 (2) July 2023 112-121

[23] H. T. Sihotang, P. M. F. Marsoit, and P. T. Marsoit, “Robust learning and optimization in distributionally robust stochastic variational inequalities under uncertainty,” *Int. J. Enterp. Model.*, vol. 17, no. 1, pp. 24–34, 2023.

[24] R. Fletcher and S. Leyffer, “Solving mixed integer nonlinear programs by outer approximation,” *Math. Program.*, vol. 66, pp. 327–349, 1994.

[25] J. B. Rosen, “The gradient projection method for nonlinear programming. Part I. Linear constraints,” *J. Soc. Ind. Appl. Math.*, vol. 8, no. 1, pp. 181–217, 1960.

[26] A. M. Geoffrion, “Generalized Benders decomposition,” *J. Optim. Theory Appl.*, vol. 10, no. 4, pp. 237–260, 1972, doi: 10.1007/BF00934810.

[27] A. O. Erick and K. A. Folly, “Power flow management in electric vehicles charging station using reinforcement learning,” in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[28] T. Ahmad, R. Madonski, D. Zhang, C. Huang, and A. Mujeeb, “Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: Key developments, challenges, and future research opportunities in the context of smart grid paradigm,” *Renew. Sustain. Energy Rev.*, vol. 160, p. 112128, 2022.

[29] I. E. Grossmann and N. V Sahinidis, “Special issue on mixed integer programming and its application to engineering, part I,” *Optim. Eng*, vol. 3, no. 4, pp. 52–76, 2002.

[30] J. Kronqvist and R. Misener, “A disjunctive cut strengthening technique for convex MINLP,” *Optim. Eng.*, vol. 22, no. 3, pp. 1315–1345, 2021, doi: 10.1007/s11081-020-09551-6.

[31] N. V Sahinidis, “Mixed-integer nonlinear programming 2018,” *Optimization and Engineering*, vol. 20. Springer, pp. 301–306, 2019.

[32] M. A. Duran and I. E. Grossmann, “An outer-approximation algorithm for a class of mixed-integer nonlinear programs,” *Math. Program.*, vol. 36, pp. 307–339, 1986.

[33] T. Westerlund, H. Skrifvars, I. Harjunkoski, and R. Pörn, “An extended cutting plane method for a class of non-convex MINLP problems,” *Comput. Chem. Eng.*, vol. 22, no. 3, pp. 357–365, 1998.

[34] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, “Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning,” *Discret. Optim.*, vol. 19, pp. 79–102, 2016.

[35] D. S. Nau, V. Kumar, and L. Kanal, “General branch and bound, and its relation to A∗ and AO∗,” *Artif. Intell.*, vol. 23, no. 1, pp. 29–58, 1984.

[36] M. J. Bagajewicz and V. Manousiouthakis, “On the generalized Benders decomposition,” *Comput. Chem. Eng.*, vol. 15, no. 10, pp. 691–700, 1991.

[37] X. Cai, D. C. McKinney, L. S. Lasdon, and D. W. Watkins Jr, “Solving large nonconvex water resources management models using generalized benders decomposition,” *Oper. Res.*, vol. 49, no. 2, pp. 235–245, 2001.

[38] V. Vassilev and K. Genova, “An approximate algorithm for nonlinear integer programming,” *Eur. J. Oper. Res.*, vol. 74, no. 1, pp. 170–178, 1994.

[39] M. E. Mangram, “A simplified perspective of the Markowitz portfolio theory,” *Glob. J. Bus. Res.*, vol. 7, no. 1, pp. 59–70, 2013.

[40] C. Sansone and M. Vento, “Signature verification: increasing performance by a multi-stage system,” *Pattern Anal. Appl.*, vol. 3, no. 2, pp. 169–181, 2000.

[41] K. B. Misra and M. D. Ljubojevic, “Optimal Reliability Design of a System: A New Look,” *IEEE Trans. Reliab.*, vol. R-22, no. 5, pp. 255–258, 1973, doi: 10.1109/TR.1973.5215673.

[42] F. A. Tillman, C.-L. Hwang, and W. Kuo, “Optimization Techniques for System Reliability with Redundancy˧A Review,” *IEEE Trans. Reliab.*, vol. R-26, no. 3, pp. 148–155, 1977, doi: 10.1109/TR.1977.5220100.

[43] R. Luus, “Optimization of system reliability by a new nonlinear integer programming procedure,” *IEEE Trans. Reliab.*, vol. 24, no. 1, pp. 14–16, 1975.