International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 3, Issue 9 (September 2014), PP.11-20

An Improved Approach For Solving Mixed-Integer Nonlinear Programming Problems

Abil Mansyur[1], Herman Mawengkang[2]
[1](*Department of Mathematics, State University of Medan, Indonesia*)
[2](*Department of Mathematics, University of Sumatera Utara, Indonesia*)

Abstract:- This special class of a 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 separable 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 the problem. After solving the problem by ignoring the integrality requirements, this strategy is used to force the appropriate non-integer basic variables to move to their neighborhood 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 results show that the proposed integerizing strategy is promising in tacking certain classes of mixed integer nonlinear programming problems.

Keywords:- Nonlinear programming, active constraints, direct search, integer programming, neighbourhood search

I. INTRODUCTION

Many decision problems in scientific, engineering and economic application involve both discrete decisions and nonlinear system dynamics that affect the quality of final design or plan. These type of problem can be modeled as mixed integer nonlinear programming problems (MINLP)
Mathematically, this problem is defined by the following model.

$$\min z = c^T y + f(x) \quad (1)$$

$$\text{s.t.} \quad h(x) \leq 0 \quad (2)$$

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

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

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 function defined on the *n*-dimensional compact polyhedral convex set $X = \{x: x \in \mathbb{R}^n, A_1 x \leq a_1\}$; $U = \{y: y \in Y, \text{integer } A_2 y \leq a_2\}$ is a discrete set, say the nonnegative integer points of some convex polytope, where for most applications Y is the unit hypercube $Y \varepsilon \{0,1\}^m$. $B, A_1, A_2,$ and c, a_1, a_2 are respectively matrices and vectors of comfortable

There are various applications for the MINLP model, including the process industry and the financial engineering, management science and operations research sectors. It includes problems in process flow sheets, portfolio selection, batch processing in chemical engineering (consisting of mixing, reaction, and centrifuge separation), and optimal design of gas or water transmission networks. Other areas of interest include the automobile, aircraft, and VLSI manufacturing areas. An impressive collection of MINLP applications can be found in [9] and [10]. The needs in such diverse areas have motivated research and development in MINLP solver technology, particularly in algorithms for handling large-scale, highly combinatorial and highly nonlinear problems.

Methods for solving MINLPs include innovative approaches and related techniques taken and extended from MIP, such as, Outer Approximation (OA) methods [5,7,10], Branch-and-Bound (B&B) [1,11,16], Extended Cutting Plane methods [19], and Generalized Bender's Decomposition (GBD) [8] for solving MINLPs have been discussed in the literature since the early 1980's. These approaches generally rely on the successive solutions of closely related NLP problems. For example, B&B starts out forming a pure continuous NLP problem by dropping the integrality requirements of the discrete variables (often called the relaxed MINLP or RMINLP). Moreover, each node of the emerging B&B tree represents a solution of the RMINLP with adjusted bounds on the discrete variables.

Heuristic approaches to solving MINLPs include Variable Neighbourhood Search [13], automatically tuned variable fixing strategies [2], Local Branching [14], feasible neighbourhood search [14], Feasibility Pump [3,4,6], heuristics based on Iterative Rounding [15]. Recently [12] propose a MINLP heuristic called the Relaxed-Exact-Continuous-Integer Problem Exploration (RECIPE) algorithm. The algorithm puts together a global search phase based on Variable Neighbourhood Search [13] and a local search phase based on a MINLP heuristic. In heuristic approaches, however, one of the main algorithmic difficulties connected to MINLPs is to find a feasible solution. 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 [14].

Due to the fact that the functions in MINLPs are not smooth, therefore in this paper we use a direct search method, known as unconstrained optimization techniques that do not explicitly use derivatives. More information regarding to direct search method in optimization can be found in [19].

In this paper we address a strategy of releasing nonbasic variables from their bounds, combined with the "active constrained" method and the notion of superbasic for efficiently tackling a particular class of MINLP problems.

The rest of this paper is organized as follows. In Section 2 we give a brief notion of solving nonlinear programming problem.. The basic approach of the proposed method is presented in Section 3. How to derive the proposed method is given in Section 4. The algorithm is presented in Section 5. Section 6 addresses a computational experience. The conclusions can be found in Section 7.

II. SOLVING NONLINEAR PROGRAMMING

In many real-world problems it turns out that most of the variables are linear and only a small percentage of the variables are involved nonlinearly in the objective function and/or in the constraints. Therefore, the standard problem may be expressed in the form

$$\text{Minimize :} \quad \mathrm{F}(x) + d^T y \tag{5}$$

$$\text{subject to } \; f(x) + A_1 y = b_1 m_1 \;\; rows \tag{6}$$

$$A_2 x + A_3 y = b_2 m_2 \quad rows \tag{7}$$

$$l \leq \begin{bmatrix} x \\ y \end{bmatrix} \leq u \quad m = m_1 + m_2 \tag{8}$$

where $f(x) = [f^1(x), \dots, f^{m_1}]^T$, and it is assumed that the functions $\mathrm{F}(x)$ and $f(x)$ are twice continuously differentiable. There are two types of variables involved in the problem, viz.,

1. The n_1"nonlinear" variable *x* which occur nonlinearly in either the objective function $\mathrm{F}(x)$or the first $m_1$constraints.
2. The n_2 "linear" variables $y, n = n_1 + n_2$, which, generally, will include a full set of *m* slack variables so that the equality and inequality constraints can be accomodated in (6) and (7) by appropriate bounds in (8).

The algorithm proceeds by conducting a sequence of "major iterations". At the start of each major iteration, the nonlinear constraints are linearized at some base point x_k and the nonlinearities are adjoined to the objective function with Langrange multipliers. Define

$$\tilde{f}(x, x_k) = f(x_k) + J(x_k)(x - x_k) \tag{9}$$

where $J_k = [J(x_k)]_{ij} = \partial f^i / \partial x_j$ is the Jacobian matrix of first partial derivatives of the constraint functions. We solve the following linearly constrained sub-problem a the kth major iteration.

$$\text{Minimize}_{x,y} \; \mathrm{L}(x, y, x_k, \lambda_k, \rho) = F(x) + d^T y - \lambda_k^T (f - \tilde{f}) + \tfrac{1}{2} \rho (f - \tilde{f})^T)(f - \tilde{f}) \tag{10}$$

$$\text{Subject to } \; J_k x + A_1 y = b_1 + J_k x_k - f(x_k) \tag{11}$$

$$A_2 x + A_3 y = b_2 \tag{12}$$

$$l \leq \begin{bmatrix} x \\ y \end{bmatrix} \leq u \tag{13}$$

The objective function (10) is a modified augmented Lagrangian, where the penalty parameter ρenhances the convergence properties from initial estimates far removed the optimum. The Lagrange multiplier estimates λ_k are taken as the optimal values at the solution of the previous subproblem. As the sequence of major iterations approaches the optimum (as measured by the relative change in successive estimates of λ_k and the degree to which the nonlinear constraints are satisfied at x_k) the penalty parameter ρ is reduced to zero and a quadratic rate of convergence of the subproblem is achieved.

The linearly constrained subproblem contains matrix equations in the form of $Ax = b$ in which we may partition the variables by introducing the notion of *superbasic* variables as follows:

$$\mathrm{A}x = [B\ S\ N]\begin{bmatrix} x_B \\ x_S \\ x_N \end{bmatrix} = b \tag{14}$$

The $m \times m$ "basis" matrix *B* is square and nonsingular and its columns correspond to the basic variables (x_B). The *r* columns of *N* correspond to the nonbasic variables (x_N) (provided there are *r* variables fixed on their bounds). The matrix S, with *n-m-r* columns, correspond to the remaining variables which are called superbasic variables (x_S). Following Eqn (14), the active contraints may be presented as

$$\hat{A}_x = \begin{bmatrix} B & S & N \\ & & I \end{bmatrix}\begin{bmatrix} x_B \\ x_S \\ x_N \end{bmatrix} = \begin{bmatrix} b \\ b_N \end{bmatrix} \tag{15}$$

thereby

$$B_{x_B} + S_{x_S} + N_{x_N} = b \tag{16}$$

$$x_N = b_N \tag{17}$$

where b_N is some combination of upper and lower bounds.

Expression (17) indicates that the nonbasic variables are being held equal to one or other of their bounds and stay there for the next step $\Delta_{x.}$ The superbasic variables x_S are free to move in any direction and provide the driving force to minimize the function, while the basic variables x_B must follow to satisfy the equation

$$B\Delta_{x_B} + S\Delta_{x_S} = 0 \tag{18}$$

Thus Δ_x can be written in terms of the change in superbasic variables as follows

$$\Delta_x = Z\Delta_{x_S} \tag{19}$$

where

$$Z = \begin{bmatrix} -B^{-1}S \\ I \\ 0 \end{bmatrix}$$

The matrix *Z* serves as a "reduction" matrix and pre-multiplies the gradient vector to form a reduced gradient $\boldsymbol{h} = \boldsymbol{Z}^T\boldsymbol{g}$, where $\boldsymbol{g} = \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{x}}$, and also pre- and post-multiplies the Hessian matrix of second partial derivatives (*G*) to yield a Newton-like step in the reduced space of superbasic variables, i.e.,

$$Z^T G Z_{p_S} = -Z^T g,\ \ p = Z_{p_S} \tag{20}$$

where $p$is a feasible descent direction.

Let the gradient of the objective function be partitioned as $[g_B, g_{S,}g_N]$. If π satisfies

$$B^T\pi = g_B \tag{21}$$

The reduced gradient can then be expressed as

$$Z^T g = g_S - S^T\pi \tag{22}$$

for the feasible direction p, we have

$$p = \begin{bmatrix} p_B \\ p_S \\ p_N \end{bmatrix} = Z_{p_S} = \begin{bmatrix} -B^{-1}S_{p_S} \\ p_S \\ 0 \end{bmatrix} \tag{23}$$

where *p* has been partitioned into $[p_B, p_S, p_N]$. The tird part of this equation, $p_N = 0$, indicates that no change will be made to the current nonbasic variables. Whenever the reduced gradient $Z^T g$ is nonzero, only the variables in $[B \quad S]$ are optimized. If the bound of a variable is encountered, that variable is made nonbasic and removed from the superbasic (or basic) set.

Let the pricing of nonbasic columns be expressed as

$$\sigma = g_N - N^T\pi \tag{24}$$

we could then write

$$\begin{bmatrix} B^T & \\ S^T & \\ N^T & I \end{bmatrix}\begin{bmatrix} \pi \\ \sigma \end{bmatrix} = \begin{bmatrix} g_B \\ g_S \\ g_N \end{bmatrix} \tag{25}$$

where π and σ are exact Lagrange multipliers for the current active constraints. If the elements of vector σ are nonzero and of appropriate sign, one or more nonbasic variables are made superbasic and optimization continues for the new set $[B \quad S]$. If not, an optimum has been achieved for the original problem.

III. 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., Mixed Integer Linear Programming (MILP) problems.
Consider a MILP problem with the following form

Minimize $\quad P = c^T x \qquad (26)$

Subject to $\quad Ax \leq b \qquad (27)$

$x \geq 0 \qquad (28)$

x_j integer for some $j \in J \qquad (29)$

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

$$(x_B)_k = \beta_k - \alpha_{k1}(x_N)_1 - \cdots - \alpha_{kj*}(x_N)_j - \cdots - \alpha_{k,n-m}\ (x_N)_{n-m} \tag{30}$$

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 β_k is not an integer, the partitioning of β_k into the integer and fractional components is that given

$$\beta_k = [\beta_k] + f_k, 0 \leq f_k \leq 1 \tag{31}$$

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 non 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

$$\Delta_{f^*} = \frac{1 - f_k}{-\alpha_{kj^*}} \tag{32}$$

while the remaining nonbasic stay at zero. It can be seen that after substituting (31) into (32) for $(x_N)_{j^*}$ and taking into account the partitioning of β_k given in (31), we obtain

$$(x_B)_k = [\beta] + 1 \tag{33}$$

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

It is now clear that a 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* (26)-(29) *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 variables x_B consists of all the slack variables then all integer variables would be in the nonbasic vector x_N and therefore integer valued.

IV. 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_{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 nonbasic $(x_N)_{j^*}$ such that all components of *x* remain feasible. This ratio test would include two cases.

1. A basic variable $(x_B)_{j\neq k}$ 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

$$\theta_1 = \min_{i\neq k|\alpha_{j^*}>0}\left\{\frac{\beta_i}{\alpha_{j^*}}\right\} \tag{34}$$

$$\theta_2 = \Delta_{j^*} \tag{35}$$

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

$$\theta^* = \min(\theta_1, \theta_2) \tag{36}$$

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

$$\Delta_{j'} = \frac{f_k}{\alpha_{kj'}} \tag{37}$$

In order to maintain the feasibility, the ratio test θ^* is still needed.

V. THE ALGORITHM

The first four sets of figure 1 partition the full index set , {1, 2, ..., n}, i.e. $J_B \cup J_S \cup J_L \cup J_U$ = {1, 2, ..., *n*} and $J_\alpha \cap J_\beta = \emptyset$, $\alpha \neq \beta$. The set J_I of indices corresponding to integer variables is assumed to be of small cardinality, and $m + n_S + n_L + n_U = n$.

The approach assumes that the continuous problem is solved, and seeks an integer-feasible solution in the close neighbourhood of the continuous solution. The general philosophy is to leave non-basic integer variables at their respective bounds (and therefore integer valued) and conduct a search in the restricted space of basics, superbasics, and nonbasic continuous variables, $j \notin J_I$.

The algorithm may be broadly summarized as follows:

1. Obtain solution of the continuous relaxation as a nonlinear programming problem.
2. CYCLE1: remove integer variables from the basis by moving a suitable nonbasic away from its bound. The hope is to drive an infeasible integer basic variable to an integer value, and then to pivot it into the superbasic set; the previous nonbasic replacing it in the basis.

Table 1. Some notation is first needed. We define the required index sets.

Name	Meaning	Cardinality
J_B	set of indices for basic variables	$\lvert J_B \rvert = m$
J_S	set of indices for superbasic variables	$\lvert J_S \rvert = n_S$
J_L	set of indices for nonbasic variables at their lower bounds	$\lvert J_L \rvert = n_L$
J_U	set of indices for nonbasic variables at their upper bounds	$\lvert J_U \rvert = n_U$
J_I	**set of indices for integer variables**	$\lvert J_I \rvert = n_I$

Tabel 1. Index sets for extended simplex partition

3. CYCLE2, pass1: adjust integer-infeasible superbasics by fractional steps to reach complete integer-feasibility.
4. CYCLE2, pass2: adjust integer feasible superbasics. This phase aims to conduct a highly-localized neighbourhood search see Scarf [83] to verify local optimality.

In Cycle1, there are several steps.

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

Step 2. Do a pricing operation

$$v_{i^*}^T = e_{i^*}^T B^{-1}$$

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

With j corresponds to

$$\min_j \left\{ \left| \frac{d_j}{\sigma_{ij}} \right| \right\}$$

Calculate the maximum movement of nonbasic j at lower bound and upper bound

Otherwise go to next non-integer nonbasic or superbasic j (if available). Eventually the column j^* is to be increased form LB or decreased from UB. If none go to next i^*.

Step 4.

Solve $B\alpha_{j^*} = \alpha_{j^*}$ for α_{j^*}

Step 5. Do ratio test for the basic variables in order to stay feasible due to the releasing of nonbasic j^* from its bounds.

Step 6. Exchange basis

Step 7. If row $i^* = \{\emptyset\}$ go to Stage 2, otherwise

Repeat from step 1.

VI. Computational Experience : A Planning Problem For Positioning A New Product A Multiattribute Space

This is a marketing problem faced by a firm which wishes to position a new brand product in an existing product class. It is natural that an individual choice for his/her most preferred products are influenced essentially by the perceptions and values of the products (e. g. the design of the product). Individuals usually differ in their choice of an object out of an existing set, and they would also differ if asked to specify an ideal object. Due to these differences, the aim of the problem considered here is to optimally design a new product in order to attract the largest number of consumers

a. Mathematical Statement of the Problem

The mathematical programming formulation of the problem is due to Duran and Grossmann [5]. Let N be the number of consumers who are a representative sample of the common population for a certain price range of a product class. Alse, let M be the number of an existing product (e. g. different brands of cars) in a market which are evaluated by consumers and are located in a multiattribute space of dimension K. We then define

z_{ik} - ideal point on attribute k for the ith consumer, $i = 1, \ldots, N; k = 1, \ldots, K$

w_{ik} - weight given to attribute k by the ith consumer, $i = 1, \ldots, N; k = 1, \ldots, K$

δ_{jk} - ideal point on attribute k for the ith consumer, $i = 1, \ldots, N; k = 1, \ldots, K$

Furthermore, a region (hyper ellipsoid) defining the distance of each consumers to the ideal point can be determined in terms of the existing product, in a way to produce a formulation such that each consumer will select a product which is closest to his/her ideal point. It was mentioned above that the objective of the problem is to optimally design a new product $(x_k, k = 1, \ldots, K)$ so as to attract the largest number of consumers.

Duran and Grossmann [13] have extended the scope of the positioning problem by introducing the revenue of the firm from the new product sales to consumer i (c_i) as well as a function f for representing the cost of reaching locations of the new product within an attribute space. Now, the objective of the problem would be to maximize the profits the firm. The binary variable (y_i) is introduced for ecah consumer to denote whether he/she is attracted by the new product or not.

Consider a positioning problem in which there are 10 existing products (M), 25 consumers (N) and attributes (K). The algebraic representation of such a problem can be written as follows.

$$\text{Maximize } F = \sum_{i=1}^{25} c_i y_i - 0.6x_1^2 + 0.9x_2 + 0.5x_3 - 0.1x_4^2 - x_5$$

Subject to

$$\sum_{k=1}^{5} w_{ik}(x_k - z_{ik})^2 - (1 - y_i)H \le R_i^2, \qquad i = 1, \ldots, 25$$

$$x_1 - x_2 + x_3 + x_4 + x_5 \leq 10$$
$$0.6x_1 - 0.9x_2 - 0.5x_3 + 0.1x_4 + x_5 \leq 0.64$$
$$x_1 - x_2 + x_3 - x_4 + x_5 \geq 0.69$$
$$0.157x_1 + 0.05x_2 \leq 1.5$$
$$0.25x_2 + 1.05x_4 - 0.3x_5 \geq 4.5$$
$$2.0 \leq x_1 \leq 4.5$$
$$0.0 \leq x_2 \leq 8.0$$
$$3.0 \leq x_3 \leq 9.0$$
$$0.0 \leq x_4 \leq 5.0$$
$$4.0 \leq x_5 \leq 10.0$$
$$0 \leq y_i \leq 1 \text{ and integer } \forall_i$$

Where

$$R_i^2 = \min_{j=1,\ldots,10} \{\sum_{k=1}^{5} w_{ik} (\delta_{jk} - z_{ik})^2\}, \qquad i = 1, \ldots, 25$$

$$C^T = [1, 0.2, 1, 0.2, 0.9, 0.9, 0.1, 0.8, 1.0, 0.4, 1, 0.3, 0.1, 0.3, 0.5, 0.9, 0.8, 0.1, 0.9, 1, 1, 1, 0.2, 0.7, 0.7]$$

and $H = 1000$

The data for the coordinates of existing product (δ_{jk}), ideal points (z_{ik}) and attribute weights (w_{ik}) can be obtained in Duran and Grossmann (1986b).

It can be seen that the above formulation is a MINLP model and it contains 25 binary variables, 5 continuous bounded variables, 30 inequality constraints (25 of them acting nonlinearly) and a nonlinear objective function.

b. Discussion of the Result

We solved the problem on PC with processor Intel(R) Core (TM) i5-2300 CPU @ 280 GHZ and RAM 4.00GB. We used our Nonlinear Programming software in order to get the optimal continuous solution. The results are presented in Table 1. It can be observed that five binary variables have had integer value (all of them are in upper bound). The binary variable y_i happens to be a superbasic in the continuous result with non-integer value. We moved this variable to its closest integer by using a truncation strategy and kept the integer result as superbasic. The corresponding basic variables would be affected due to this movement. Therefore it is necessary to check the feasibility of the results. The proposed integerizing algorithm was then implemented on the remaining non-integer binary variables. The integer results can also be found in Table 1.

It is interesting to note that our result $(F = 8.14313)$ is slightly better that Duran and Grossmann's [5] result $(F = 7.78913)$. The binary variable y_i has a value of 1.0 in our result instead of 0.0 as in Duran and Grossmann's result. The total computational time to get the integer result by using our proposed algorithm is 10.98 seconds.

Table 1. The Results of the Positioning Problem.

Variable	Activity in Cont.Soln.	Activity after integ. Process
x_1	2.0	2.0
x_2	8.0	7.81528
x_3	7.32849	6.29911
x_4	3.52381	3.56779
x_5	4.0	4.0
y_1	0.93153	1.0
y_2	0.70970	0.0
y_3	0.67548	0.0
y_4	0.50181	0.0
y_5	0.77537	0.0
y_6	1.0	1.0
y_7	0.78191	0.0
y_8	1.0	1.0
y_9	0.82922	0.0
y_{10}	0.11168	0.0
y_{11}	0.81785	0.0
y_{12}	0.74375	0.0
y_{13}	0.93852	0.0
y_{14}	0.61360	0.0
y_{15}	1.0	1.0
y_{16}	0.69117	0.0
y_{17}	1.0	1.0
y_{18}	0.91958	0.0
y_{19}	0.83079	0.0
y_{20}	0.97451	1.0
y_{21}	0.93383	0.0
y_{22}	0.57154	0.0
y_{23}	0.49858	0.0
y_{24}	0.91093	0.0
y_{25}	1.0	1.0
Obj.value(F)	16.41964	8.14313

VII. CONCLUSIONS

This paper has presented a direct search method for achieving integer-feasibility for a class of mixed-integer nonlinear programming problems in a relatively short time. The direct search approach used the strategy of releasing nonbasic variable from their bounds, combined with the “active constraint” method and the notion of superbasic. After solving a problem by ignoring the integrality requirements, this strategy is used to force the appropriate non-integer basic variables to move to their neighborhoods integer points.

A study of the criteria for choosing a nonbasic variable to work with in the integerizing strategy has also been made. The number of integerizing steps would be finite if the number of integer variables contained in the problem is finite. However, it should be noted that the computational time for the integerizing process does not necessarily depend on the number of integer variables, since many of the integer variables may have an integer value at the continuous optimal solution.

The new direct search method has been shown to be successful on a range of problems, while not always able to achieve global optimality. In a number of cases to obtain the suboptimal point is acceptable, since the exponential complexity of the combinatorial problems in general precludes branch-and-bound, except on small to medium problems.

Computational testing of the procedure presented this paper has demonstrated that it is a viable approach for large problems.

REFERENCES

[1]. I.E. Grossmann and N.V. Sahinidis (eds). Special Issue on Mixed-integer Programming and its Application to Engineering, Part I, Optim. Eng., 3 (4), Kluwer Academic Publishers, Netherlands, 2002.

[2]. I.E. Grossmann and N.V. Sahinidis (eds). Special Issue on Mixed-integer Programming and its Application to Engineering, Part II, Optim. Eng., 4 (1), Kluwer Academic Publishers, Netherlands, 2002.

[3]. M. A. Duran and I. E Grossmann, 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. Belotti, J. Lee, L. Liberti, F. Margot, A. Wachter, Branching and Bounds Tightening Techniques for non-convex MINLP, Optimization Methods and Software, **24(4)**, 2009, 597-634.

[6]. O.K. Gupta and A. Ravindran. Branch and Bound Experiments in Convex Nonlinear Integer Programming, Manage Sci., 31 (12), 1985, 1533–1546.

[7]. G. Nannicini, P. Belotti. Rounding-based heuristics for non-convex MINLPs. In: P. Bonami, L. Liberti, A. Miller, A. Sartenear (eds). Proceedings of the European Workshop on MINLP. CIRM, Marseille, France, 2009 .

[8]. Tamara G. Kolda, Robert M. Lewis, Virginia Torezon. Optimization by Direct Search: New Perspective on Some Classical and Modern Methods. SIAM Review, 45 (1), 2003, pp. 385-482.

[9]. A.M. Geoffrion, A Generalized Benders Decomposition, J. Op-tim. Theory Appl., 10 (4), 1972, 237–260

[10]. L. Liberti, G. Nannicini, N. Mladenovic. A good recipe for solving MINLPs. In: V. Maniezzo, T. Stutzle, S.Voss(eds.) Metaheuristics: Hybridizing metaheuristics and Mathematical Programming. Annals of Information Systems, vol. 10, 2009, pp. 231-245. Springer.

[11]. T. Berthold and A. Gleixner. Undercover-Primal MINLP Heuristic. In P. Bonami, L. Liberti, A. Miller and A. Sartenaer (eds.) Proceedings of the European Workshop on MINLP, Pp 103-113, Marseille, 2010.

[12]. H. Mawengkang and B. A Murtagh, Solving Nonlinear Integer Programs with Large-Scale Optimization Software. Annals of Operations Research 5425-437, 1986.

[13]. C. D'Ambrosio, A. Frangioni, L. Liberti, A.Lodi. Experiments with a Feasibility Pump approachfor non-convex MINLPs. In: P. Festa(ed) Proceedings of the 9th Synposiumon Experimental Algorithms (SEA 2010), Lecture Notes in Computer Science, vol. 6049. Springer, Berlin, 2010.

[14]. C. D'Ambrosio, A. Frangioni, L. Liberti, A. Lodi. A storm of Feasibility Pump for non-convex MINLP. Tech. Rep. OR-10-13, DEIS, Universita di Bologna, 2010.

[15]. M. Fischetti, F. Glover, A. Lodi. The Feasibility Pump. Mathematical Programming A 104(1), 2005, 91-104.

[16]. G. Nannicini, P. Belotti. Rounding-based heuristics for non-convex MINLPs. In: P. Bonami, L. Liberti, A. Miller, A. Sartenear (eds). Proceedings of the European Workshop on MINLP. CIRM, Marseille, France, 2009.

[17]. L. Liberti, N. Miladenovie, G. Nannicini. A Recipe for Finding Good Solutions to MINLPs. Mathematical Programming Computation, 2011.

[18]. I. Quesada and I.E. Grossmann. An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems, Computers Chem. Eng., 16 (10/11), 1992, 937–947.

[19]. H. E. Scarf, Testing for Optimality in the Absence of Convexity. In: W. P. Heller, R. M Starr and D. A. Starett (Eds) Cambridge University Press, 1986, pp 117-134.

[20]. S. A. Vavasis, Nonlinear Optimization: Complexity Issues. Oxford University Press, Oxford, 1991.

[21]. T. Westerlund and F. Petersson . A Cutting Plane Method for Solving Convex MINLP Problems, Computers Chem. Eng., 19, 1995, 131–136.

[22]. A.M. Geoffrion, A Generalized Benders Decomposition, *J. Op-tim. Theory Appl., 10 (4)*, 1972, 237–260

[23]. C. D'Ambrosio, A. Frangioni, L. Liberti, A.Lodi. Experiments with a Feasibility Pump approachfor non-convex MINLPs. In: P. Festa(ed) Proceedings of the 9th Synposiumon Experimental Algorithms (SEA 2010), *Lecture Notes in Computer Science, vol. 6049. Springer, Berlin, 2010.*

[24]. C. D'Ambrosio, A. Frangioni, L. Liberti, A. Lodi. *A storm of Feasibility Pump for non-convex MINLP. Tech. Rep. OR-10-13*, DEIS, Universita di Bologna, 2010.

[25]. G. Nannicini and P. Belotti, Local Branching for MINPs. *Technical Report workingpaper*, CMU, 2009.

[26]. G. Nannicini, P. Belotti. Rounding-based heuristics for non-convex MINLPs. In: P. Bonami, L. Liberti, A. Miller, A. Sartenear (eds). *Proceedings of the European Workshop on MINLP. CIRM*, Marseille, France, 2009 .

[27]. H. E. Scarf, *Testing for Optimality in the Absence of Convexity*. In: W. P. Heller, R. M Starr and D. A. Starett (Eds) Cambridge University Press, 1986, pp 117-134.

[28]. H. Mawengkang and B. A Murtagh, Solving Nonlinear Integer Programs with Large-Scale Optimization Software. *Annals of Operations Research 5425-437, 1986.*

[29]. I.E. Grossmann and N.V. Sahinidis (eds). Special Issue on Mixed-integer Programming and its Application to Engineering, Part I, *Optim. Eng., 3 (4)*, Kluwer Academic Publishers, Netherlands, 2002.

[30]. I.E. Grossmann and N.V. Sahinidis (eds). Special Issue on Mixed-integer Programming and its Application to Engineering, Part II, *Optim. Eng., 4 (1)*, Kluwer Academic Publishers, Netherlands, 2002.

[31]. I. Quesada and I.E. Grossmann. An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems, *Computers Chem. Eng*., 16 (10/11), 1992, 937–947.

[32]. L. Liberti, N. Miladenovie, G. Nannicini. A Recipe for Finding Good Solutions to MINLPs. *Mathematical Programming Computation,* 2011.

[33]. L. Liberti, G. Nannicini, N. Mladenovic. A good recipe for solving MINLPs. In: V. Maniezzo, T. Stutzle, S.Voss(eds.) Metaheuristics: *Hybridizing metaheuristics and Mathematical Programming. Annals of Information Systems,* vol. 10, 2009, pp. 231-245. Springer.

[34]. M. A. Duran and I. E Grossmann, An Outer-Approximation Algorithm For a class of Mixed-Integer Nonlinear Programs, *Mathematical Programming* 36, 1986, 307.

[35]. M. Fischetti, F. Glover, A. Lodi. The Feasibility Pump. *Mathematical Programming A* 104(1), 2005, 91-104.

[36]. O.K. Gupta and A. Ravindran. Branch and Bound Experiments in Convex Nonlinear Integer Programming, *Manage Sci*., 31 (12), 1985, 1533–1546.
[37]. P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wachter, Branching and Bounds Tightening Techniques for non-convex MINLP, *Optimization Methods and Software,* **24(4)**, 2009, 597-634.
[38]. Roger Fletcher and Sven Leyffer, Solving Mixed Integer Nonlinear Programming by Outer Approximation, *Mathematical Programming*, Vol. 66, 1994, 327.
[39]. S. A. Vavasis, *Nonlinear Optimization: Complexity Issues.* Oxford University Press, Oxford, 1991.
[40]. Tamara G. Kolda, Robert M. Lewis, Virginia Torezon. Optimization by Direct Search: New Perspective on Some Classical and Modern Methods. *SIAM Review*, 45 (1), 2003, pp. 385-482.
[41]. T. Berthold and A. Gleixner. Undercover-Primal MINLP Heuristic. In P. Bonami, L. Liberti, A. Miller and A. Sartenaer (eds.) *Proceedings of the European Workshop on MINLP,* Pp 103-113, Marseille, 2010.
[42]. T. Westerlund and F. Petersson . A Cutting Plane Method for Solving Convex MINLP Problems, *Computers Chem. Eng*., 19, 1995, 131–136.