Transaction on Evolutionary Algorithm and Hybrid Optimization
ISSN: 2229-8711 Online Publication, June 2012
www.pcoglobal.com/gjto.htm
NP-D17/GJTO

Global Journal of Technology & Optimization
Volume 3, 2012

# AN IMPROVED DIRECT SEARCH APPROACH FOR SOLVING MIXED-INTEGER NONLINEAR PROGRAMMING PROBLEMS

**Herman Mawengkang[1], Mangku M. Guno[2], Dedy Hartama[2], Arie S. Siregar[2], Hikmah A. Adam[2], Ommi Alfina[2]**

[1]Graduate School of Mathematics, University of Sumatera Utara
[2]Graduate School of Computer Science, University of Sumatera Utara
Email: hmawengkang@yahoo.com

Received January 2012, Revised March 2012, Accepted April 2012

**Abstract**
The 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 non-basic variables from their bounds, combined with the "active constraint" method and the notion of super-basics, 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 non-basic variable to work with in the integer zing strategy has also been made. Successful implementation of these algorithms was achieved on various test problems. The results show that the proposed integer zing strategy is promising in tackling certain classes of mixed integer nonlinear programming problems.

**Keywords:** Mixed integer programming, nonlinear programming, direct search, neighbourhood search, large scale optimization.

## 1. Introduction

Mixed Integer Nonlinear Programming (MINLP) refers to mathematical programming with continuous and discrete variables and nonlinearities in the objective function and constraints. The special class of Mixed-Integer nonlinear programming problem which is addressed in this paper is to assume discrete values, which are linear and separable from the continuous variables. This problem is defined by the following model.

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

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

$$g(x) + by \leq 0 \tag{3}$$

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

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 $X = \{x: x \in \mathbb{R}^n, A_1 x \leq a_1\}$; $U = \{y: y \in Y, \text{integer } A_2 y \leq a_2\}$ is a discrete set, say the nonnegative integer points of some convex polytope, where for most applications Y is the unit hypercube $Y \in \{0,1\}^m$. $B, A_1, A_2,$ and $c, a_1, a_2$ are respectively matrices and vectors of comfortable dimensions; the vectors are column vectors unless specified otherwise.

There 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 superbasics 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 neighbourhood search.Tthe 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.

## 2. Neighbourhood Search

It should be noted that, generally, in integer programming the reduced gradient vector, which is normally used to detect an optimality condition, is not available, even though the problems are convex. Thus we need to impose a certain condition for the local testing search procedure in order to assure that we have obtained the "best" suboptimal integer feasible solution.

Scarf [18] has proposed a quantity test to replace the pricing test for optimality in the integer programming problem. The test is conducted by a search through the neighbours of a proposed feasible point to see whether a nearby point is also feasible and yields an improvement to the objective function.

Let $[\beta]_k$ be an integer point belongs to a finite set of neighbourhood $N([\beta]_k)$. We define a neighbourhood system associated with $[\beta]_k$, that is, if such an integer point satisfies the following two requirements

1. If $[\beta]_j \in N([\beta]_k)$ then $[\beta]_k \in [\beta]_j), j \neq k$.
2. $N([\beta]_k) = [\beta]_k + N(0)$

With respect to the neighbourhood system mentioned above, the proposed integerizing strategy can be described as follows.

Given a non-integer component, $x_k$, of an optimal vector , $x_B$, The adjacent points of $x_k$, being considered are $[x_k]$ and $[x_k]+1$. If one of these points satisfies the constraints and yields a minimum deterioration of the optimal objective value we move to another component, if not we have integer-feasible solution.

Let $[x_k]$ be the integer feasible point which satisfies the above conditions. We could then say if $[x_k]+1 \in N([x_k])$ implies that the point $[x_k]+1$ is either infeasible or yields an inferior value to the objective function obtained with respect to $[x_k]$. In this case $[x_k]$ is said to be an "optimal" integer feasible solution to the integer programming problem. Obviously, in our case, a neigbourhood search is conducted through a proposed feasible points such that the integer feasible solution would be at the least distance from the optimal continuous solution.

## 3. 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

$$\text{Minimize} \quad P = c^T x \tag{5}$$

$$\text{Subject to} \quad Ax \leq b \tag{6}$$

$$x \geq 0 \tag{7}$$

$$x_j \text{ integer for some } j \in J \tag{8}$$

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

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

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 $\beta_k$ is not an integer, the partitioning of $\beta_k$ 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 non-basic variable, say $(x_N)_{j^*}$, above its bound of zero, provided $\alpha_{kj^*}$, as one of the element of the vector $\alpha_{j^*}$, is negative. Let $\Delta_{j^*}$ 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), $\Delta_{j^*}$ 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 (10) into (11) for $(x_N)_{j^*}$ and taking into account the partitioning of $\beta_k$ given in (10), we obtain

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

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

It is now clear that a non-basic variable plays an important role to integrate 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 integer zing process.

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

**Proof:**

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

## 4. 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 $\Delta_{j^*}$. Consequently, if some element of vector $\alpha_{j^*}$, i.e., $\alpha_{ij^*}$ 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 is $(x_N)_{j^*}$ 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)
2. The basic variable, $(x_B)_k$ increases to an integer.

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

$$\theta_1 = \min_{i \neq k | a_{j^*} > 0} \left\{ \frac{\beta_i}{\alpha_{j^*}} \right\} \quad (12)$$

$$\theta_2 = \Delta_{j^*} \quad (13)$$

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 $\theta^*$ given below

$$\theta^* = \min(\theta_1, \theta_2) \quad (14)$$

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

$$\Delta_{j^*} = \frac{f_k}{\alpha_{kj}} \quad (15)$$

In order to maintain the feasibility, the ratio test $\theta^*$ is still needed. Consider the movement of a particular non-basic variable, $\Delta$, as expressed in Eqns.(11) and (15).

The only factor that one needs to calculate is the corresponding element of vector $\alpha$. A vector $\alpha_j$ can be expressed as

$$\alpha_j = B^{-1} a_j, j = 1 \ldots, n - m \quad (16)$$

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

$$v_k^T = e_k^T B^{-1} \quad (17)$$

Subsequently, the numerical value of $\alpha_{kj^*}$ can be obtained from

$$\alpha_{kj^*} = v_k^T a_{j^*} \quad (18)$$

in Linear Programming (LP) terminology the operation conducted in Eqns. (17) and (18) is called the pricing operation. The vector of reduced costs $d_j$ is 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 integer zing 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. (11) or (15), depends in some way on the corresponding element of vector $\alpha_j$. 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 rize a basic variable $(x_B)_k$ may be measured by the ration

$$\left| \frac{d_k}{\alpha_{kj^*}} \right| \quad (19)$$

where $|\alpha|$ 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,

$$\min_j \left\{ \left| \frac{d_k}{\alpha_{kj^*}} \right| \right\}, \quad j = 1, \ldots, n - m \quad (20)$$

From the "active constraint" strategy and the partitioning of the constraints corresponding to basic $(B)$, superbasic $(S)$ and nonbasic $(N)$ variables we can write

$$\begin{bmatrix} B & S & N \\ & & I \end{bmatrix} \begin{bmatrix} x_b \\ x_N \\ x_S \end{bmatrix} = \begin{bmatrix} b \\ b_N \end{bmatrix} \quad (21)$$

or

$$Bx_b + Sx_S + Nx_N = b \quad (22)$$

$$x_N = b_N \quad (23)$$

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

$$x_B = \beta - Wx_S - \alpha x_N \quad (24)$$

Where

$$\beta = B^{-1} b \quad (25)$$

$$W = B^{-1} S \quad (26)$$

$$\alpha = B^{-1} N \quad (27)$$

Expression (23) indicates that the non-basic variables are being held equal to their bound. It is evident through the "nearly" basic expression of Eqn. (24), the integer zing 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, Eqn.(23) and exchange it with a corresponding basic variable in the integer zing 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 non-basic, they will be at their bound. Therefore they have integer value.
2. If $y_j$ is super-basic, it is possible to make $y_j$ basic and bring in a non-basic at its bound to replace it in the super-basic.

However, the ratio test expressed in (14) cannot be used as a tool to guarantee that the integer solution optimal found gill 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.

### 4.1. Pivoting

Currently, we are in a position where particular basic variable, $(x_B)_k$ is being integer zed, thereby a corresponding non-basic variable, $(c_N)_{j^*}$, 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 in linear programming, 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 integer zing process continues with a new set $[B, S]$. In this case, eventually we may end up with all of the integer variables being super-basic.

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

**Proof:**

1. If all of the integer variables are in $N$, then they will be a bound.

2. If an integer variable is basic it is possible to either
   - Interchange it with a super-basic continuous variable, or
   - Make this integer variable super-basic and bring in a non-basic at its bound to replace it in the basis which gives a degenerate solution.

The other case which can happen is that 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 situation where

$$\theta^* = \Delta_1$$

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

**5. 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 $x = [x] + f, \quad 0 \le f \le 1$

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

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

this is a pricing operation

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

With $j$ corresponds to

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

I. *For non-basic j at lower bound*

If $\sigma_{ij} < 0$ and $\delta_{i^*} = f_i$ calculate $\Delta = \dfrac{(1-\delta_{i^*})}{-\sigma_{ij}}$

If $\sigma_{ij} > 0$ and $\delta_{i^*} = 1 - f_i$ calculate $\Delta = \dfrac{(1-\delta_{i^*})}{\sigma_{ij}}$

If $\sigma_{ij} < 0$ and $\delta_{i^*} = 1 - f_i$ calculate $\Delta = \dfrac{\delta_{i^*}}{-\sigma_{ij}}$

If $\sigma_{ij} > 0$ and $\delta_{i^*} = f_i$ calculate $\Delta = \dfrac{\delta_{i^*}}{\sigma_{ij}}$

II. For *non-basic j at upper bound*

If $\sigma_{ij} < 0$ and $\delta_{i^*} = 1 - f_i$ calculate $\Delta = \dfrac{(1-\delta_{i^*})}{-\sigma_{ij}}$

If $\sigma_{ij} > 0$ and $\delta_{i^*} = f_i$ calculate $\Delta = \dfrac{(1-\delta_{i^*})}{\sigma_{ij}}$

If $\sigma_{ij} > 0$ and $\delta_{i^*} = 1 - f_i$ calculate $\Delta = \dfrac{\delta_{i^*}}{\sigma_{ij}}$

If $\sigma_{ij} < 0$ and $\delta_{i^*} = f_i$ calculate $\Delta = \dfrac{\delta_{i^*}}{-\sigma_{ij}}$

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

**Step 4.** Calculate

$\alpha_{j^*} = B^{-1} a_{j^*}$

i.e. solve $B\alpha_{j^*} = a_{j^*}$ for $\alpha_{j^*}$

**Step 5.** Ratio test; there would be three possibilities for the basic variables in order to stay feasible due to the releasing of non-basic $j^*$ from its bounds.

*If $j^*$ lower bound*

Let

$$A = \min_{i' \neq i^* | \alpha_{ij^*} > 0} \left\{ \frac{x_{B_{i'}} - l_{i'}}{\alpha_{ij^*}} \right\}$$

$$B = \min_{i' \neq i^* | \alpha_{ij^*} < 0} \left\{ \frac{u_{i'} - x_{B_{i'}}}{-\alpha_{ij^*}} \right\}$$

$$C = \Delta$$

the maximum movement of $j^*$ depends on:

$$\theta^* = \min(A, B, C)$$

*If $j^*$ upper bound*

Let

$$A' = \min_{i' \neq i^* | \alpha_{ij^*} < 0} \left\{ \frac{x_{B_{i'}} - l_{i'}}{\alpha_{ij^*}} \right\}$$

$$B' = \min_{i' \neq i^* | \alpha_{ij^*} > 0} \left\{ \frac{u_{i'} - x_{B_{i'}}}{-\alpha_{ij^*}} \right\}$$

$$C' = \Delta$$

the maximum movement of $j^*$ depends on:

$$\theta^* = \min(A', B', C')$$

**Step 6.** Exchanging basis for the three possibilities

1. If $A$ or $A'$ $x_{B_{i'}}$ becomes non-basic at lower bound $l_{i'}$ $x_{j^*}$ becomes basic (replaces $x_{B_{i^*}}$)
   - $x_{i^*}$ stays basic (non-integer)
2. If $B$ or $B'$ $x_{B_{i'}}$ becomes non-basic at upper bound $u_{i'}$ $x_{j^*}$ becomes basic (replaces $x_{B_{i^*}}$)
   - $x_{i^*}$ stays basic (non-integer)
3. If $C$ or $C'$
   - $x_{j^*}$ becomes basic (replaces $x_{i^*}$)
   - $x_{i^*}$ becomes super-basic at integer-valued

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

Repeat from step 1.

Stage 2. Adjust integer feasible super-basics using line search approach.

The super-basics can be varied at some points subject to preserving of the basic variables. Thus a search through the neighborhood system, as defined in Section 2, will verify the (local) optimality of the integer-feasible solution obtained.

Define $J_1$ as the index of the super-basic variable with the smallest fractional component, and $J_B$ is the index set of the basic variable. The integer line search would proceed as follows.

Step 1. This is basically a one dimensional steepest descent.

Choose $j' \in J_1$. The criterion for selecting $j'$ will be that of maximum reduced cost $\lambda_{j'}$.

Step 2. Calculate $\alpha_{ij'}$. Also determine direction of move – check sign of $\lambda_{j'}$. and adjust the unit tests in Step 3 in light of this.

Step 3. Check that a unit move is possible:

$$\frac{u_i - x_i}{+\alpha_{ij'}} \geq 1, \quad \forall i \mid i \in J_B | \alpha_{ij'} < 0$$

$$\frac{x_i - l_i}{-\alpha_{ij'}} \geq 1, \quad \forall i \mid i \in J_B | \alpha_{ij'} < 0$$

Step 4. Move by 1 unit; check that objective improves, i.e. search in the neighbourhood system.

## 6. Computational Experience 1: A Process System Synthesis Problem

**Mathematical Statement of the problem**

This synthesis problem is the one of simultaneously determining the optimal structural and operating parameters for a process so as to satisfy a given design specification. The decision variables are defined as follows.

$y$ is a binary variable which is associated with each process unit (piece of equipment) to denote its potential existence in the final optimal configuration, and

$x$ are the continuous which represent process parameters such as flow rates of materials.

Generally, the objective is to minimize the annual costs, including both investment and operation costs.

Minimize

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

Subject to

$$-1.5\ln(x_{19} + 1) - \ln(x_{21} + 1) - x_{14} \leq 0$$
$$-\ln(x_{10} + x_{17} + 1) \leq 0$$
$$-x_3 - x_5 + x_{10} + 2x_{17} + 0.8x_{19} + 0.8x_{21} - 0.5x_9 - x_{14} - 2x_{25} \leq 0$$
$$-x_3 - x_5 + 2x_{17} + 0.8x_{19} + 0.8x_{21} - 2x_9 - x_{14} - 2x_{25} \leq 0$$
$$-2x_{17} - 0.8x_{19} - 0.8x_{21} + 2x_9 + x_{14} + 2x_{25} \leq 0$$
$$-0.8x_{19} - 0.8x_{21} + x_{14} \leq 0$$
$$-x_{17} + x_9 + x_{25} \leq 0$$
$$-0.4x_{19} - 0.4x_{21} + 1.5x_{14} \leq 0$$
$$0.16x_{19} + 0.16x_{21} - 1.2x_{14} \leq 0$$
$$x_{10} - 0.8x_{17} \leq 0$$
$$-x_{10} + 0.4x_{17} \leq 0$$
$$\exp(x_3) - 10y_1 \leq 1$$
$$\exp(x_3) - 10y_2 \leq 1$$
$$x_9 - 10y_3 \leq 0$$
$$0.8x_{19} + 0.8x_{21} - 10y_4 \leq 0$$
$$2x_{17} - 2x_9 - 2x_{25} - 10y_5 \leq 0$$
$$x_{19} - 10y_6 \leq 0$$
$$x_{21} - 10y_7 \leq 0$$
$$x_{10} + x_{17} - 10y_8 \leq 0$$
$$y_1 + y_2 = 1$$
$$y_4 + y_5 \leq 1$$
$$-y_4 + y_6 + y_7 = 0$$
$$y_3 - y_8 \leq 0$$
$$0 \leq y_j \leq 1 \text{ and integer for } j = 1, \cdots, 8$$
$$l \leq x \leq u$$
$$x = x_j; (j = 3,5,10,17,19,21,9,14,25) \in R^9$$
$$l^T = (0,0,0,0,0,0,0,0,0), \quad u^T = (2,2,1,2,2,2,2,1,3)$$

The above formulation contains 8 binary variables, 9 bounded continuous variables, 23 inequality constraints. Nonlinearities appear in the objective function and in four inequalities.

**Discussion of the results**

We solved this problem using PC with processor Intel(R) Core (TM) i5-2300 CPU @ 280 GHZ and RAM 4.00GB. The continuous optimal solution was obtained by using NLP software. Only one binary variable is integer-valued (at its lower bound) in the continuous solution. A binary variable $y_7$ is in super-basic set with non integer value. We then moved this variable to its closest integer by using truncation strategy and kept it super-basic. We must check the feasibility of the corresponding basic variables due to this movement. We integer zed the remaining non-integer binary variables by using our proposed integer zing strategy. Both the continuous and the integer results of the synthesis problem can be seen in the Table 1.

Table 1. The Results of the Ssynthesis Problem.

| Variable | Activity in Cont.Soln. | 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 |

Our objective result is in agreement with the result obtained by [2] .

## 7. Computational Experience 2: 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 individual choices for his/her most prefered 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

**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. Also, 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 multi attribute space of dimension $K$. We then define

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

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

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

Further more, 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 $(x_k, k = 1, \dots, K)$ so as to attract the largest number of consumers.

Duran and Grossmann [5] 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.

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

Subject to

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

$$x_1 - x_2 + x_3 + x_4 + x_5 \le 10$$

$$0.6x_1 - 0.9x_2 - 0.5x_3 + 0.1x_4 + x_5 \le 0.64$$

$$x_1 - x_2 + x_3 - x_4 + x_5 \ge 0.69$$

$$0.157x_1 + 0.05x_2 \le 1.5$$

$$0.25x_2 + 1.05x_4 - 0.3x_5 \ge 4.5$$

$$2.0 \le x_1 \le 4.5$$

$$0.0 \le x_2 \le 8.0$$

$$3.0 \le x_3 \le 9.0$$

$$0.0 \le x_4 \le 5.0$$

$$4.0 \le x_5 \le 10.0$$

$$0 \le y_i \le 1 \text{ and integer } \forall_i$$

where

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

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

and $H = 1000$

The data for the coordinates of existing product $(\delta_{jk})$, 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 containts 25 binary variables, 5 continuous bounded variables, 30 inequality constraints (25 of them acting nonlinearly) and a nonlinear objective function.

**7.1 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 Tabel 2. 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 2.

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.

**8. 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 are 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 benn 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.

Table 2. The Results of the Positioning Problem.

| Variable | Activity in Cont. Soln. | Activity after integer. 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 |

**References**


[1] P. Belotti, J. Lee, L. Liberti, F. Margot, A. Wachter, Branching and Bounds Tightening Techniques for non-convex MINLP, Optimization Methods and Software, 2009, vol. 24(4): pp. 597-634.

[2] 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, Marseille, 2010, pp 103-113.

[3] 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, Springer, Berlin, 2010, vol. 6049.

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

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

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

[7] Roger Fletcher and Sven Leyffer, Solving Mixed Integer Nonlinear Programming by Outer Approximation, Mathematical Programming, 1994, Vol. 66: 327.

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

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

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

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

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

[13] 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, Springer, 2009, vol. 10, pp. 231-245.

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

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

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

[18] 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.

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

[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., 1995, vol. 19, pp. 131–136.