Fuzzy Programming for Mixed-Integer Optimization Problems

Yung-Chin Lin[a], Yung-Chien Lin[b], Kuo-Lan Su[a], Wei-Cheng Lin[c], Tsing-Hua Chen[b]

[a]*Department of Electrical Engineering, National Yunlin University of Science & Technology, Yunlin, Taiwan*
[b]*Department of Electrical Engineering, WuFeng University, Chiayi, Taiwan*
[c]*Department of Electrical Engineering, I-Shou University, Kaohsiung, Taiwan*
e-mail: {yclin@wfu.edu.tw; chien-lin@ wfu.edu.tw; sukl@yuntech.edu.tw; byron@isu.edu.tw; thchen@wfu.edu.tw}

Abstract: Mixed-integer optimization problems belong to NP-hard combinatorial problems. Therefore, they are difficult to search for the global optimal solutions. The mixed-integer optimization problems are always described by precise mathematical programming models. However, many practical mixed-integer optimization problems inherit more or less imprecise nature. Under this circumstance, if we take into account the flexibility of constraints and the fuzziness of objectives, the original mixed-integer optimization problems can be formulated as fuzzy mixed-integer optimization problems. Mixed-integer differential evolution (MIHDE) is an evolutionary search algorithm, and has been successfully applied to many complex mixed-integer optimization problems. In this paper, a fuzzy mixed-integer mathematical programming model is developed to formulate the fuzzy mixed-integer optimization problem. And then the MIHDE is introduced to solve this fuzzy mixed-integer programming problem. Finally, the illustrative example shows that satisfactory results can be obtained by the proposed method. This demonstrates that the MIHDE can effectively handle the fuzzy mixed-integer optimization problems.

Keywords: fuzzy programming, mixed-integer optimization, evolutionary algorithm**.**

I. Introduction

Many real-world optimization problems involve integer or discrete design variables in addition to continuous design variables. This kind of problems is called mixed-integer optimization problems. Mixed-integer optimization problems belong to NP-hard combinatorial problems, therefore they are difficult to search for the global optimal solutions. On the other hand, the mixed-integer optimization problems are always described by precise mathematical programming models. However, many practical mixed-integer optimization problems inherit more or less imprecise nature. Under this circumstance, if we take into account the flexibility of constraints and the fuzziness of objectives, the original mixed-integer optimization problems can be formulated as fuzzy mixed-integer optimization problems.

In the fuzzy mixed-integer optimization problems, the constraints and objectives are defined by fuzzy sets and denoted as "fuzzy constraints" and "fuzzy goals" [1]. Combined with fuzzy constraints, fuzzy goals and fuzzy decision, a fuzzy mixed-integer optimization problem can be transformed into a mixed-integer optimization problem. Therefore, one can use a mixed-integer optimization approach to solve such a mixed-integer optimization problem.

Evolutionary algorithms (EAs) [2, 3] are powerful search algorithms based on the mechanism of natural selection. Unlike conventional search approaches, they simultaneously consider many points in the search space so as to increase the chance of global convergence. Recently, EAs have exhibited promising results for solving complex problems such as highly nonlinear, non-differentiable and multi-modal optimization problems [4]. Mixed-integer differential evolution (MIHDE) [5] is an evolutionary algorithm. A mixed coding is introduced in MIHDE to implement the evolutionary process of continuous and integer variables. The MIHDE has been successfully applied to many complex mixed-integer optimization problems [5-8].

In this paper, a fuzzy mixed-integer programming model is developed to formulate the fuzzy mixed-integer optimization problem. The MIHDE is introduced to solve this fuzzy mixed-integer programming problem. Finally, the illustrative example show that satisfactory results can be obtained by the proposed method. This demonstrates that the MIHDE can effectively handle the fuzzy mixed-integer optimization problems.

II. Fuzzy Mixed-Integer Mathematical Programming

If we soften the rigid requirements of a mixed-integer optimization problem, the mixed-integer optimization problem can be stated by a fuzzy mixed-integer programming model as follows:

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) \tag{1}$$

subject to $$g_j(\mathbf{x},\mathbf{y}) \precsim 0,\ j = 1,\ldots,m_i \tag{2}$$

where $\mathbf{x}$ represents an n_C-dimensional vector of continuous variables, $\mathbf{y}$ is a n_I-dimensional vector of integer variables, and the symbol "$\precsim$" respectively denote the softened or fuzzy versions of constraints $g_j(\mathbf{x},\mathbf{y})$. It means that the minimized objective function can be further improved with properly softened constraints.

The fuzzy goal and fuzzy constraints can be quantified by the membership functions. Here a linear membership function such as triangular function is employed. The membership functions are represented by $\mu_f(\mathbf{x},\mathbf{y})$ and $\mu_{g_j}(\mathbf{x},\mathbf{y})$ as defined by equations (3) and (4).

$$\mu_f(\mathbf{x},\mathbf{y}) = \begin{cases} 0 & \text{if } f(\mathbf{x},\mathbf{y}) \le f^0 \\ \dfrac{f^1 - f(\mathbf{x},\mathbf{y})}{f^1 - f^0} & \text{if } f^0 < f(\mathbf{x},\mathbf{y}) < f^1 \\ 1 & \text{if } f(\mathbf{x},\mathbf{y}) \ge f^1 \end{cases} \quad (3)$$

$$\mu_{g_j}(\mathbf{x},\mathbf{y}) = \begin{cases} 0 & \text{if } g_j(\mathbf{x},\mathbf{y}) \ge g_j^0 \\ \dfrac{g_j^0 - g_j(\mathbf{x},\mathbf{y})}{g_j^0 - g_j^1} & \text{if } g_j^1 < g_j(\mathbf{x},\mathbf{y}) < g_j^0 \\ 1 & \text{if } g_j(\mathbf{x},\mathbf{y}) \le g_j^1 \end{cases} \quad (4)$$

where f^0 and g_j^0 are respectively denote the values of f and g_j such that the grades of the membership functions $\mu_f(\mathbf{x},\mathbf{y})$ and $\mu_{g_j}(\mathbf{x},\mathbf{y})$ are 0, and f^1 and g_j^1 represent the values of f and g_j such that the grades of the membership function $\mu_f(\mathbf{x},\mathbf{y})$ and $\mu_{g_j}(\mathbf{x},\mathbf{y})$ are 1.

The fuzzy decision $\mu_D(\mathbf{x},\mathbf{y})$ is expressed as

$$\mu_D(\mathbf{x},\mathbf{y}) = \{\mu_f(\mathbf{x},\mathbf{y})\} \cap \left\{\bigcap_1^{mi} \mu_{g_j}(\mathbf{x},\mathbf{y})\right\} \quad (5)$$

If we follow the fuzzy decision of Bellman and Zadeh [9], the optimal solution $(\mathbf{x}^*,\mathbf{y}^*)$ of fuzzy decision $\mu_D(\mathbf{x},\mathbf{y})$ can be selected by maximizing the smallest membership function such that

$$\mu_D(\mathbf{x}^*,\mathbf{y}^*) = \max\ \min\{\mu_f(\mathbf{x},\mathbf{y}), \mu_{g_1}(\mathbf{x},\mathbf{y}), \ldots, \mu_{g_{mi}}(\mathbf{x},\mathbf{y})\} \quad (6)$$

By introducing the auxiliary variable λ, the max-min problem can be transformed into the following mixed-integer nonlinear programming (MINLP) problem:

$$\text{maximize } \lambda \quad (7)$$

subject to

$$\lambda \le \mu_f(\mathbf{x},\mathbf{y}) \quad (8)$$

$$\lambda \le \mu_{g_j}(\mathbf{x},\mathbf{y}), \quad j = 1,\ldots,m_i \quad (9)$$

In order to solve this MINLP problem effectively, the MIHDE algorithm is applied to solve this problem and find the maximizing decision. The details of MIHDE are described in the following section.

III. Mixed-Integer Hybrid Differential Evolution

Let us consider a general MINLP problem as follows:

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) \quad (10)$$

$$\mathbf{x}^L \le \mathbf{x} \le \mathbf{x}^U$$
$$\mathbf{y}^L \le \mathbf{y} \le \mathbf{y}^U$$

where x represents an n_C-dimensional vector of real-valued variables, y is an n_I-dimensional vector of integer-valued variables, and $(\mathbf{x}^L,\mathbf{y}^L)$ and $(\mathbf{x}^U,\mathbf{y}^U)$ are the lower and upper bounds of the corresponding decision vectors.

The procedure of MIHDE includes the following 5 steps.

1) Representation and Initialization

MIHDE uses N_p decision vectors $\{\mathbf{z}_i^G\} = \{(\mathbf{x}^G,\mathbf{y}^G)_i\}$, $i = 1,\ldots,N_p$ to denote a population of N_p individuals in the G-th generation. The decision vector (chromosome), $(\mathbf{x},\mathbf{y})_i$, is represented as $(x_{1i},\ldots,x_{ji},\ldots,x_{n_C i}, y_{1i},\ldots,y_{ji},\ldots,y_{n_I i})$. The decision variables (genes), x_{ji} and y_{ji}, are directly coded as real-valued and integer-valued numbers. The initialization process generates N_p decision vectors $(\mathbf{x},\mathbf{y})_i$ randomly, and should try to cover the entire search space uniformly as in the form:

$$(\mathbf{x}^0,\mathbf{y}^0)_i = (\mathbf{x}^L,\mathbf{y}^L) + \lfloor \rho_i\{(\mathbf{x}^U,\mathbf{y}^U) - (\mathbf{x}^L,\mathbf{y}^L)\}\rfloor, i = 1,\ldots,N_p \quad (11)$$

where $\rho_i = \mathrm{Diag}(\rho_{i,1}, \rho_{i,2}, \ldots, \rho_{i,n_C+n_I})$ is a diagonal matrix, the diagonal elements $(\rho_{i,1}, \rho_{i,2}, \ldots, \rho_{i,n_C+n_I})$ are random numbers in the range $[0,1]$, the other elements are zero, and the rounding operator $\lfloor \mathbf{a} = \rho_i(\mathbf{x}^U - \mathbf{x}^L), \mathbf{b} = \rho_i(\mathbf{y}^U - \mathbf{y}^L)\}\rfloor$ in (11) is defined as $(\mathbf{a}, \mathrm{INT}[\mathbf{b}])$ in which the operator $\mathrm{INT}[\mathbf{b}]$ is expressed as the nearest integer-valued vector to the real-valued vector b.

2) Mutation

The i-th mutant individual $(\mathbf{u}^G,\mathbf{v}^G)_i$ is obtained by the difference for two random individuals as expressed in the form

$$(\mathbf{u}^G,\mathbf{v}^G)_i = (\mathbf{x}^G,\mathbf{y}^G)_p + \lfloor \rho_m\{(\mathbf{x}^G,\mathbf{y}^G)_k - (\mathbf{x}^G,\mathbf{y}^G)_l\}\rfloor$$
$$= (\mathbf{x}^G,\mathbf{y}^G)_p + (\rho_m(\mathbf{x}_k^G - \mathbf{x}_l^G), \mathrm{INT}[\rho_m(\mathbf{y}_k^G - \mathbf{y}_l^G)]) \quad (12)$$

where random indices $k,l \in 1,\ldots,N_p$ are mutually different. The operator $\mathrm{INT}[\mathbf{b} = \rho_m(\mathbf{y}_k^G - \mathbf{y}_l^G)]$ in (12) is to find the nearest integer vector to the real vector b. The mutation factor ρ_m is a real random number between zero and one. This factor is used to control the search step among the direction of the differential variation $(\mathbf{x}^G,\mathbf{y}^G)_k - (\mathbf{x}^G,\mathbf{y}^G)_l$.

3) Crossover

In crossover operation, each gene of the *i*-th individual is reproduced from the mutant vector $(\mathbf{u}^G,\mathbf{v}^G)_i$ = $(u_{1i}^G, u_{2i}^G,\ldots,u_{n_C i}^G, v_{1i}^G, v_{2i}^G,\ldots,v_{n_I i}^G)$ and the current individual $(\mathbf{x}^G,\mathbf{y}^G)_i = (x_{1i}^G, x_{2i}^G, \ldots, x_{n_C i}^G, y_{1i}^G, y_{2i}^G, \ldots, y_{n_I i}^G)$ as follows:

$$u_{li}^{G+1} = \begin{cases} x_{li}^G, & \text{if a random number} > \rho_c \\ u_{li}^G, & \text{otherwise}; \ l = 1,\ldots,n_C,\ i = 1,\ldots,N_p \end{cases} \tag{13}$$

$$v_{li}^{G+1} = \begin{cases} y_{li}^G, & \text{if a random number} > \rho_c \\ v_{li}^G, & \text{otherwise}; \ l = 1,\ldots,n_I,\ i = 1,\ldots,N_p \end{cases} \tag{14}$$

where the crossover factor $\rho_c \in [0,1]$ is a constant and the value can be specified by the user.

4) Evaluation and Selection

The operation includes two evaluation phases. The first phase is performed to produce the new population in the next generation as (15). The second phase is used to obtain the best individual as (16).

$$(\mathbf{x}^{G+1}, y^{G+1})_i = \arg\min\{ f((\mathbf{x}^G, \mathbf{y}^G)_i), f((\mathbf{u}^{G+1}, \mathbf{v}^{G+1})_i)\} \quad i = 1,\ldots,N_p \tag{15}$$

$$(\mathbf{x}^{G+1}, \mathbf{y}^{G+1})_b = \arg\min\{ f((\mathbf{x}^{G+1}, \mathbf{y}^{G+1})_i),\ i = 1,\ldots,N_p\} \tag{16}$$

where $(\mathbf{x}^{G+1}, \mathbf{y}^{G+1})_b$ is the best individual with the smallest objective function value.

5) Migration

In order to increase the exploration of the search space, a migration operation is introduced to generate a diversified population. Based on the best individual $(\mathbf{x}^{G+1}, \mathbf{y}^{G+1})_b = (x_{1b}^{G+1}, x_{2b}^{G+1}, \ldots, x_{n_C b}^{G+1}, y_{1b}^{G+1}, y_{2b}^{G+1}, \ldots, y_{n_I b}^{G+1})$, the j-th gene of the i-th individual can be diversified by the following equations:

$$x_{ji}^{G+1} = \begin{cases} x_{jb}^{G+1} + \rho_1 (x_j^L - x_{jb}^{G+1}), & \text{if a random number} < \dfrac{x_{jb}^{G+1} - x_j^L}{x_j^U - x_j^L} \\ x_{jb}^{G+1} + \rho_1 (x_j^U - x_{jb}^{G+1}), & \text{otherwise}; \ j = 1,\ldots,n_C,\ i = 1,\ldots,N_p \end{cases} \tag{17}$$

$$y_{ji}^{G+1} = \begin{cases} y_{jb}^{G+1} + \mathrm{INT}[\rho_2 (y_j^L - y_{jb}^{G+1})], & \text{if a random number} < \dfrac{y_{jb}^{G+1} - y_j^L}{y_j^U - y_j^L} \\ y_{jb}^{G+1} + \mathrm{INT}[\rho_2 (y_j^U - y_{jb}^{G+1})], & \text{otherwise}; \ j = 1,\ldots,n_I,\ i = 1,\ldots,N_p \end{cases} \tag{18}$$

where ρ_1 and ρ_2 are the random numbers in the range [0,1].

The migration operation in MIHDE is performed only if a measure for the population diversity is not satisfied, that is when most of individuals have clustered together, the migration has to be actuated to make some improvements. In this study, we propose a measure called the population diversity degree η to check whether the migration operation should be performed. In order to define the measure, we first introduce the following gene diversity index for each real-valued gene x_{ji}^{G+1} and for each integer-valued gene y_{ki}^{G+1},

$$dx_{ji} = \begin{cases} 0, & \text{if} \left| \dfrac{x_{ji}^{G+1} - x_{jb}^{G+1}}{x_{jb}^{G+1}} \right| < \varepsilon_2; \ j = 1,\ldots,n_C,\ i = 1,\ldots,N_p,\ i \neq b \\ 1, & \text{otherwise} \end{cases} \tag{19}$$

$$dy_{ji} = \begin{cases} 0, & \text{if } y_{ji}^{G+1} = y_{jb}^{G+1}; \ j = 1,\ldots,n_I,\ i = 1,\ldots,N_p,\ i \neq b \\ 1, & \text{otherwise} \end{cases} \tag{20}$$

where dx_{ji} and dy_{ji} are the gene diversity indices and $\varepsilon_2 \in [0,1]$ is a tolerance for real-valued gene provided by the user. According to (19) and (20), we assign the j-th gene diversity index for the i-th individual to zero if this gene clusters to the best gene. We now define the population diversity degree η as a ratio of total diversified genes in the population. From (19) and (20) we have the population diversity degree as

$$\eta = \left\{ \sum_{\substack{i=1 \\ i \neq b}}^{N_p} \left[\sum_{j=1}^{n_C} dx_{ji} + \sum_{j=1}^{n_I} d_{ji} \right] \right\} / \{ (n_C + n_I)(N_p - 1) \} \tag{21}$$

From equation (19), (20) and (21), the value of η is in the range [0,1]. Consequently, we can set a tolerance for population diversity, $\varepsilon_1 \in [0,1]$, to assess whether the migration should be actuated. If η is smaller than ε_1, then MIHDE performs the migration to generate a new population to escape a local solution. Contrary, if η is not less than ε_1, then MIHDE suspends the migration operation to keep a constant search direction to a target solution.

IV. Computational Example

Consider a design problem of pressure vessel, as presented by Sandgren [10], is shown in Figure 1.

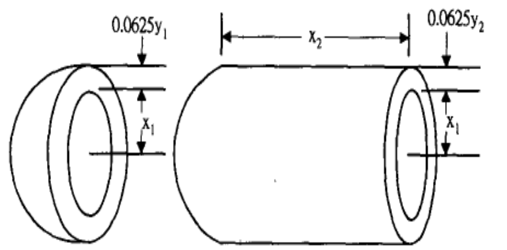

Figure 1. Pressure Vessel design.

The design variables are the dimensions required for the specifications of the vessel, i.e.

$$(\mathbf{x},\mathbf{y}) = (x_1, x_2, y_1, y_2).$$

The objective function is the combined costs of material, forming and welding of the pressure vessel. The constraints are set in accordance with the respective ASME codes. The mixed-integer optimization problem is expressed as:

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x},\mathbf{y}) = 0.6224(0.0625y_1)x_1x_2 + 1.7781(0.0625y_2)x_1^2 + 3.1611(0.0625y_1)^2 x_2 + 19.84(0.0625y_1)^2 x_1 \quad (22)$$

subject to

$$g_1(\mathbf{x},\mathbf{y}) = 0.0193x_1 - 0.0625y_1 \le 0$$
$$g_2(\mathbf{x},\mathbf{y}) = 0.00954x_1 - 0.0625y_2 \le 0$$
$$g_3(\mathbf{x},\mathbf{y}) = 750 \times 1728 - \pi x_1^2 x_2 - 4/3\, \pi x_1^3 \le 0$$
$$g_4(\mathbf{x},\mathbf{y}) = x_2 - 240 \le 0$$
$$10.0 \le x_1 \le 100.0$$
$$10.0 \le x_2 \le 240.0$$
$$10 \le y_1 \le 32$$
$$10 \le y_2 \le 32$$

Using the MIHDE algorithm to solve this mixed-integer optimization problem, the obtained optimal solution is $(\mathbf{x}^*, \mathbf{y}^*) = (x_1^*, x_2^*, y_1^*, y^*_2) = (38.8754, 221.4069, 12, 10)$, $f(\mathbf{x}^*, \mathbf{y}^*) = 6521.3198$

Instead of giving the crisp values for this mixed-integer optimization problem, the fuzzy goal and the fuzzy constraints are described in Table 1.

Table 1. Fuzzy goal and fuzzy constraints.

	$\mu_f = 0$ or $\mu_{g_j} = 0$	$\mu_f = 1$ or $\mu_{g_j} = 1$
$f(\mathbf{x},\mathbf{y})$	$f^0 = 6200.0$	$f^1 = 6500.0$
$g_1(\mathbf{x},\mathbf{y})$	$g_1^0 = 5.0$	$g_1^1 = -5.0$
$g_2(\mathbf{x},\mathbf{y})$	$g_2^0 = 5.0$	$g_2^1 = -5.0$
$g_3(\mathbf{x},\mathbf{y})$	$g_3^0 = 5.0$	$g_3^1 = -5.0$
$g_4(\mathbf{x},\mathbf{y})$	$g_4^0 = 5.0$	$g_4^1 = -5.0$

For this fuzzy mixed-integer optimization problem, the obtained optimal solution by MIHDE is
$(\mathbf{x}^*, \mathbf{y}^*) = (x_1^*, x_2^*, y_1^*, y^*_2) = (40.6027, 196.1014, 12, 10)$,
$f(\mathbf{x}^*, \mathbf{y}^*) = 6350.6731$,
$\lambda = 4.9664$

From computational result, the cost of pressure vessel can be decreased through the fuzzy programming for the original mixed-integer optimization problem. This implied that the cost function can be further improved if the constraints are softened to a more favorable degree.

V. Conclusions

In this paper, a fuzzy mixed-integer mathematical programming model is developed to formulate the fuzzy mixed-integer optimization problem. And then the MIHDE is introduced to solve this fuzzy mixed-integer programming problem. Finally, the illustrative example shows that satisfactory results can be obtained by the proposed method. This demonstrates that the MIHDE can effectively handle the fuzzy mixed-integer optimization problems.

References

[1] H. J. Zimmermann, "Application of fuzzy sets theory to mathematical programming," *Information Science*, vol. 36, pp. 29-58, 1985.

[2] Z. Michalewicz, "Genetic Algorithm + Data Structure = Evolution Programs," Springer-Verlag, 1994.

[3] T. Back, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. New York: Oxford Univ. Press, 1997.

[4] Z. Michalewicz, Z. and M. Schoenauer, "Evolutionary Algorithms for Constrained Parameter Optimization Problems," *Evolutionary Computation*, vol. 4, no. 1, pp. 1-32, 1996.

[5] Y. C. Lin, K. S. Hwang, and F. S. Wang, "A mixed-coding scheme of evolutionary algorithms to solve mixed-integer nonlinear programming problems," *Computers & Mathematics with applications*, vol. 47, pp. 1295-1307, 2004.

[6] Y. C. Lin, K. S. Hwang, and F. S. Wang, "Co-evolutionary hybrid differential evolution for mixed-integer optimization problems," *Engineering Optimization*, vol. 33, no. 6, pp. 663-682, 2001.

[7] Y. C. Lin, K. S. Hwang, and F. S. Wang, "An evolutionary Lagrange method for mixed-integer constrained optimization problems," *Engineering Optimization*, vol. 35, no. 3, pp. 267-284, 2003.

[8] Y. C. Lin, Y. C. Lin, and K. L. Su, "Production planning based on evolutionary mixed-integer nonlinear programming," *ICIC Express Letters*, vol. 4, no. 5(B), pp. 1881-1886, 2010.

[9] R. E. Bellman and L. A. Zadeh, "Decision making in a fuzzy environment," *Management Science*, vol. 17, pp. 141-164, 1970.

[10] E. Sandgren, "Nonlinear integer and discrete programming in mechanical design," *ASME J. Mechanical Design*, vol. 112, pp. 223-229, 1990.