# Hybrid Coding Collaborative DE-ACO Algorithm for Solving Mixed-Integer Programming Problems

Yuelin Gao
Institute of Information and System Science, Beifang University for Nationalities, Yinchuan, China
Email: gaoyuelin@263.net

Xia Chang and Junmei Liu
Institute of Information and System Science, Beifang University for Nationalities, Yinchuan, China
Email: {changxia0104, liujunmei800911}@163.com

***Abstract*—This paper presents a hybrid coding collaborative ant colony-differential evolution algorithm for solving bound constrained mixed integer programming problems. In this algorithm, a real number and integer hybrid coding strategy is used, and the population evolution is realized by colony optimization and differential evolution. It is shown by numerical experiments that the proposed algorithm is effective. The proposed algorithm is combined with penalty function method to solved the general mixed integer programming problems. Numerical experiments show that the proposed method achieves satisfactory results.**

***Index Terms*—co-evolutionary, hybrid coding, mixed-integer programming (MIP), differential evolution (DE), ant colony optimization (ACO)**

## I. Introduction

We consider the general mixed-integer programming problem (MIP) as follows:

$$\begin{cases} \min \ f(x,y), \\ \text{s.t.} \ \ g_i(x,y) \le 0 \quad i=1,2,...,m, \\ \qquad h_j(x,y)=0 \quad j=1,2,...,n, \\ \qquad x^L \le x=(x_1,x_2,...,x_{n_c}) \le x^U, \\ \qquad y^L \le y=(y_1,y_2,...,y_{n_I}) \le y^U. \end{cases} \quad (1)$$

where the function $f(x,y)$, $g_i(x,y)$, and $h_j(x,y)$ are all real continuous functions, $x$ is $n_c$ -dimensional real variable, $y$ is $n_I$ -dimensional integer variable, $x^L$ and $y^L$ are the upper bounds of $x$, $y$ respectively , $x^U$ and $y^U$ are the lower bound of $x$, $y$ respectively.

MIP exists widely in many areas, such as machinery, chemical industry, resource management, production scheduling, military affairs and so on. Many combinatorial optimization problems also belong to MIP, e.g. knapsack problem TSP, site selection, distribution problem etc. MIP is generally recognized as a NP hard problem. Some methods to solve the MIP are proposed by domestic and foreign scholars, which can be roughly divided into two categories. One is a deterministic method, including branch and bound method [1, 15, 16], outer-approximation algorithm (OA) [2], Dantzig-Wolf decomposition(GBD) [5], cutting plane method [17] and so on. These deterministic methods are effective to middle-scale and small-scale MIP. The other is a random method which has drawn wide attention recently, including Genetic Algorithms (GA) [4], differential evolution (DE) [5-8], and so on [18,19]. Although these random methods have acquired favorable results, their convergence has not been proved. In this paper, we pay our attention to the bound constrained mixed-integer programming problem (BCMIP) as follows:

$$\begin{cases} \min \ f(x,y), \\ \text{s.t.} \ \ x^L \le x=(x_1,x_2,...,x_{n_C}) \le x^U, \\ \qquad y^L \le y=(y_1,y_2,...,y_{n_I}) \le y^U. \end{cases} \quad (2)$$

A hybrid coding DE and ACO co-evolutionary algorithm HC-DE-ACO is proposed to solve the BCMIP. During the iteration process, the hybrid coding is applied to each individual coding, which includes the real coding part and the integer coding part. The real coding part is evolved by DE and the integer coding part is evolved by ACO. The entire population is co-evolved by ACO and DE. MIP is changed to BCMIP with semi-penalty function which can be solved by the HC-PSO-ACO too.

This article is organized as follows. Section II briefly reviews DE and ACO algorithms. Section III presents the Co-Evolutionary HC-DE-ACO Algorithm. Numerical test results and application are provided in Section IV. Conclusions are drawn in Section V.

Manuscript received December 7, 2012; revised March 6, 2013; accepted April 9, 2013.

The work is supported by the Foundations of National Natural Science China (No.60962006, No.61102008) and by the Foundations of research projects of State Ethnic Affairs Commission (No.12BFZ005).

Corresponding author: Yuelin Gao.

© 2014 ACADEMY PUBLISHER
doi:10.4304/jcp.9.1.101-106

## II. The Basic Principles of DE and ACO

### A. The Basic DE

Differential evolution (DE) [9,10] is a simple heuristic search method, introduced by (Storn and Price 1995), which is known for its remarkable performance in continuous numerical problems.

For an optimization problem $\min F(x)$, DE starts from an initial population which contains of $N$ candidate solutions $x_i^t$. $i$ is the index of individual, $t$ is generation number. Any random quantity $v_i^t$ is generated according to equation (6) and (7). In the mutation operation, $r1, r2, r3 \in \{1, 2, ..., N\}$ are randomly chosen and different from each other and also different from the current index $i$, $F \in [0, 2]$ is scaling factor. We have

$$v_i^t = x_{best}^t + F(x_{r2}^t - x_{r3}^t) \tag{3}$$

In crossover operation, the new vector $y_i^t = [y_{i1}^t, y_{i2}^t, ..., y_{iD}^t]$ is co-produced by random vector $v_i = [v_{i1}, v_{i2}, ..., v_{iD}]$ and target vector $x_i = [x_{i1}, x_{i2}, ..., x_{iD}]$.

$$y_{ij}^t = \begin{cases} v_{ij}, \text{ if rand } b(j) \le \text{CR or} \\ \quad j = \text{rand } r(i) \\ x_{ij}, \text{ if rand } b(j) > \text{CR and} \\ \quad j \ne \text{rand } r(i) \end{cases} \tag{4}$$

where $j \in [1, D]$, rand $b(j)$ is the $i^{\text{th}}$ independent random number uniformly distributed in the range of [0,1]. rand $r(i)$ is a randomly chosen index from the set [1, 2, ..., D], which ensures that there is at least one parameter from $x_i^t$ to $v_i^t$. CR $\in$ [0, 1] is called crossover probability that controls the diversity of the populations.

The selection operation decides whether the trial vector $y_i^{t+1}$ would be a member of the population of the next generation $t+1$:

$$x_i^{t+1} = \begin{cases} y_i^{t+1} & \text{if } \phi(y_i^{t+1}) < \phi(x_i^t) \\ x_i^t & \text{otherwise} \end{cases} \tag{5}$$

where $\phi(x)$ denotes fitness function.

### B. The Basic ACO

Ant Colony Optimization is first proposed by Italian scholar M.Dorigo *et al*. in 1990s [11]. After extensive research, people found that the ants' individuals exchange and pass information by the substance called pheromones (pheromone). So that ants can collaborate to complete complex tasks. Following is a brief introduction of the ant colony algorithm [12, 13].

We suppose the number of ants is $m$, every simple ant has some factors as follows: it chooses the next city by the distance between the cites and the probability function with the variable describing the strength of the pheromone on each edge ($\tau_{ij}(t)$ represents the strength of the pheromone on edge $e(i, j)$ at time $t$); an ant musts move on the legal routes: it is not permitted to move to the visited cities unless the whole route has been completed, and a taboo table is used to control this condition ($tabu_k$ denotes the $k$-*th* ant's taboo table and the $tabu_k(s)$ denotes the $s$-*th* element of the taboo table). When an ant finished a circle, the pheromone will be left in the visited edges. The pheromone in each edge is equal initially. Set $\tau_{ij}(0) = C$ ($C$ is a constant). Ant $k$ shift its direction by the strength of each path's pheromone in the course of its moving, $p_{ij}^k$ represents the probability that ant $k$ transfers from position $i$ to position $j$ at time $t$.

In formula (6), $\text{allowed}_k = \{0,1,...,n\text{-}1\} - \text{tabu}_k$ represents the ant $k$' allowed-choose cities in the next step. It is different from the actual ant colony, artificial ant colony

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t).\eta_{ij}^{\beta}(t)}{\sum\limits_{s \in allowed_k} \tau_{is}^{\alpha}(t).\eta_{is}^{\beta}(t)}, & \text{if } j \in \text{allowed}_k, \; tabu_k \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

system has memory function, $(k = 1, 2, ..., m)$ is used to record the ant $k$'s visited cities, and $tabu_k$ is dynamically adjusted in the process of evolution. $\eta_{ij}$ represents the visibility of edge $(i, j)$, and we get $\eta_{ij} = 1/d_{ij}$ by a heuristic algorithm, where, $d_{ij}$ represents the distance between city $i$ and city $j$. $\alpha$ represents the relative importance of the trajectory and $\beta$ represents the relative importance of the visibility. $\rho$ represents the persistence of the trajectory, while $1-\rho$ can be seen as the attenuation of the trajectory, which simulates the phenomena that the previous pheromone gradually disappear as the time goes on. After the ants finishing a whole circle, the amount of the pheromone is adjusted as follows:

$$\tau_{ij}(t+n) = \rho\tau_{ij}(t) + \Delta\tau_{ij}. \tag{7}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{n} \Delta\tau_{ij}^k. \tag{8}$$

where $\Delta\tau_{ij}^k$ represents the amount of pheromone on path $ij$ which ant $k$ remains in this circle, $\Delta\tau_{ij}$ represents the incremental amount of the pheromone on path $ij$. For the update formula of $\Delta\tau_{ij}$, three different models have been given by M. Dorigo [13], which are respectively called ant-cycle system, ant-quantity system and ant-density system.

In the ant-cycle system, we have :

$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{L_K}, & \text{if Ant k passes } ij \text{ in the cycle,} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

where $L_K$ represents the whole route's length that ant $k$ has been visited in one circle. $Q$ represents the strength of pheromone, which is taken as a constant.

In the ant-quantity system, we have:

$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{d_{ij}}, & \text{if Ant } k \text{ passes } ij \\ & \text{at the time } t \text{ and } t+1, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

© 2014 ACADEMY PUBLISHER

where $d_{ij}$ represents the distance between $i$ and $j$ . $Q$ represents the strength of the pheromone, which is taken as a constant.

In the ant-density system, we have

$$\Delta\tau_{ij}^{k}=\begin{cases}\text{Q}, & \text{if Ant } k \text{ passes } ij \\ & \text{at the time } t \text{ and } t+1, \\ 0, & otherwise.\end{cases} \quad (11)$$

The model (10) and the model (11) have strong local search ability due to the utilization of the local information, while model (9) has strong global search ability due to the utilization of the global information. Therefore, TSP is usually solved by the model (9).

## III. HC-DE-ACO Algorithm

The feasible region of BCMIP is a super-rectangle denoted as $\Omega=\{(x,y):x^L\le x\le x^U,\ y^L\le y\le y^U\}$, which can be divided into two super-rectangles represented as $\Omega_x=\{x:x^L\le x\le x^U\}$ and $\Omega_y=\{y:y^L\le y\le y^U\}$ respectively. we have $\Omega=\Omega_x\times\Omega_y$, where the former super-rectangle's dimension is $n_c$ , the later one is $n_I$ . Since $f(x,y)$ is a consequent function, the BCMIP can be equally written as follows:

$$\min_{x\in\Omega_x}[\min_{y\in\Omega_y} f(x,y)] \quad (12)$$

We combine DE and ACO to solve formula (12). Each single feasible point $(x,y)=(x_1,x_2,\cdots,x_n,y_1,y_2,\cdots,y_m)^T$ in its feasible region $\Omega$ of BCMIP is taken as an individual, each individual is expressed by hybrid coding, that is $x$ is expressed by real coding, while $y$ is expressed by integer coding. The entire population is co-evolved by PSO and ACO, the real coding is evolved by DE algorithm, while the integer coding is evolved by ACO algorithm. The hybrid coding HC-DE-ACO co-evolutionary algorithm will be specifically described in following.

$l_i=y_i^U-y_i^L+1$ represents the number of possible values of $y_i$ . $y_i$ has $l_i$ nodes, each value of the variable composes a solution space [13]. For example, when $y_i$ takes the $m_i$-$th$ $(1\le m_i\le l_i)$ node, we will have the corresponding solution space as follows:

$$(y_1,y_2,...,y_m)=(y_1^L+m_1-1,y_2^L+m_2-1,\cdots,y_m^L+m_n-1) \quad (13)$$

where $m$ variables are selected to make the previous problem become a $m$ decision-making problem. We suppose that there are $l_i$ nodes in level $i$ , initially all ants are in the first level, the probability that a ant selects the $i$-$th$ node in level $j$ is as follow:

$$\text{p}_{ij}=\frac{\tau_{ij}}{\sum_{i=1}^{l_i}\tau_{ij}} \quad (14)$$

$\tau_{ij}$ can be seen as the attractive strength of the $i$-$th$ node in level $j$ . We have the update equation as follows:

$$\tau_{ij}^{\text{new}}=\rho\tau_{ij}^{\text{old}}+\frac{\text{Q}}{f} \quad (15)$$

where we set the mount of ants is 30 viz. $m=30$ , the attenuation coefficient of attract strength is 0.1 viz. $\rho=0.1$ ,and $\text{Q}=10$ . $f$ represents the objective function.

The HC-DE-ACO algorithm will be described in detail below. The algorithm has two evolutionary cycles. In the outside circle, the evolution of the real part uses differential evolution while in the inner loop the evolution of the integer part is the ant colony optimization by collaborative evolution.

The basic framework of HC-DE-ACO algorithm is as follows:

Step1. Initialize the population size $N$ and the maximum number of iteration $T\max$ .

Step2. Randomly produce original population.

Step3. Calculate the fitness value of each individual in the initial population, and record the current optimal solution and the optimal value.

Step4. Fix the best $N_C$ real individual generated by the ant colony optimization and produce the corresponding real individual by differential evolution.

Step5. Fix the best $N_I$ individual generated by differential evolution and produce the corresponding integer individual by the ant colony optimization. The each real individual and the corresponding integer individual are combined to produces new evolutionary population individuals.

Step6. Compute each individual fitness value of the new evolutionary population. Update the current best solution and the optimal value.

Step7. If $t>T\max$ , reserve the obtained value of the integer variable, evolve real values by DE, update the current best optimal solution by combining the reserved value of integer values, and output the global optimal solution and global optimal value. Otherwise, go to Step4.

TABLE I-1
Comparison of Calculated Results

| $f_k(x)$ | Meaning | |
|---|---|---|
| $k$ | $f_k(x^*)$ | $Nfe$ |
| 1 | 6.9413908E-11 | 33304 |
| 2 | 1.0000722E+00 | 37992 |
| 3 | 9.9377519E-07 | 50144 |
| 4 | 6.9756512E-07 | 26241 |
| 5 | 6.9756512E-07 | 71906 |
| 6 | 9.6768219E-07 | 12411 |
| 7 | 3.0143892E-07 | 26782 |
| 8 | 7.7798657E-07 | 18112 |
| 9 | 6.6047082E-07 | 20073 |
| 10 | 8.7374176E-07 | 65252 |

© 2014 ACADEMY PUBLISHER

## IV. Numerical Experiments

In order to compare conveniently, this paper takes ten BCMIPs as test functions which have a variety of properties in multi-extreme and non-convexity. These functions are described in literature [8].

To verify the validity of the HC-DE-ACO algorithm for BCMIP, we carried out our numerical experiments under Matlab 7 environment. The experimental results of HC-DE-ACO are compared with that of MIHDE [8] and MIDE [14]. The dimension of each BCMIP used as test function is 40.

Let $N_c = 20$, $N_I = 20$, $F$=0.5, $CR = 0.6$ in MIHDE [8] and MIDE [14], $F$=0.5 in HC-DE-ACO, Population scale $Np = 10$ in MIDE [14], Population scale $Np = 3$ in MIHDE [8] and HC-DE-ACO. VTR is up to the excellent value. If we seek that the optimal value less than or equal to the VTR, we think that the problem is solved. $f_k(x)$ is the $k$ function, $f_k(x^*)$ is the global optimal value of the $k$ function and $Nfe$ is the number of the function's calculations.

In Table I, we can see that for Test functions 1, 2, 5, 6, 7, 8, and 9，HC-DE-ACO is better in the convergent accuracy than the other two algorithms. For the test functions 3, 4, and 10, the HC-DE-ACO gets higher value in the convergent precision as well. And HC-DE-ACO is higher in computational time than MIHDE, that can be accepted. Thus HC-DE-ACO is feasible for BCMIP.

TABLE I-2
Comparison of Calculated Results

| MIDE （$NP=10$） | | HC-DE-ACO （$NP=3$） | |
|---|---|---|---|
| $f_k(x^*)$ | $Nfe$ | $f_k(x^*)$ | $Nfe$ |
| 5.7760000E+03 0 | >200000 | 4.7349564E-040 | 64829 |
| 1.0000000E+03 | 1191 | 1.0000000E+000 | 49772 |
| 3.4281849e+04 | >2000000 | 5.3421475E-008 | 55984 |
| 1.1640583E-01 | >2000000 | 8.4114134E-008 | 39655 |
| 4.0984231E-01 | >2000000 | 0.0000000E+000 | 20833 |
| 9.8174770E-03 | >2000000 | 1.1779085E-032 | 37907 |
| 1.4137167E+00 | >2000000 | 1.1779085E-032 | 38001 |
| 6.0000000E-01 | >2000000 | 1.3497838E-032 | 55324 |
| 2.5000000E+00 | >2000000 | 1.3497838E-032 | 38453 |
| 2.7075000E-01 | >2000000 | 2.7846542E-008 | 87206 |

Finally, we transformed the general mixed-integer programming problem (MIP) into the following half penalty function bound constrained mixed integer programming problem:

$$\begin{cases} \min F(x,y) = f(x,y) + \alpha\sum_{i=1}^{m}(\max\{g_i(x,y),0\})^2 \\ \qquad\qquad + \beta\sum_{j=1}^{n}h_j^2(x,y), \\ \text{s.t.}\quad x^L \le x = (x_1,x_2,...,x_{nC}) \le x^U, \\ \qquad y^L \le y = (y_1,y_2,...,y_{nI}) \le y^U. \end{cases} \tag{16}$$

where $\alpha$ and $\beta$ are large enough normal number (taken to be $10^8$). We solve the problem (16) by HC-DE-ACO, and the results are compared with the literature [8]. The values of other parameters are the same as that of the parameters in the previous test.

From Table II, we can see that, for Problems 1, 3, and 5, HC-DE-ACO is the same as MIHDE in reaching the global minimum, moreover, for Problems 2 and 4, HC-DE-ACO did not reach a global minimum and its convergence precision is below to MIHDE.

TABLE II
Comparison of Calculated Results

| Problem | MIHDE | HC-DE-ACO | Global minimum |
|---|---|---|---|
| *1* | 2 | 2 | 2 |
| *2* | 2.124 | 2.2131 | 2.124 |
| *3* | 1.07654 | 1.07654 | 1.07654 |
| *4* | 99.24520 | 101.1079 | 99.24520 |
| *5* | 3.557463 | 3.557463 | 3.557463 |

## V. Conclusion

A kind of hybrid coding differential evolution-ant colony optimization collaborative algorithm HC-DE-ACO is proposed. In this algorithm, each individual in the population is represented as a hybrid coding combined by the real coding and the integer coding. By DE and ACO, the real part and the integer part collaboratively evolve respectively so as to reach the evolution of the entire population. Numerical experiments show that the HC-DE-ACO algorithm is feasible and effective.

By the combination of HC-DE-ACO algorithm and penalty method, the mixed integer programming problem (MIP) is solved and satisfactory results are achieved.

## Appendix

Problem 1[8]

$$\begin{cases} \min\ f(x,y) = 2x + y, \\ \text{s.t.}\ \ 1.25 - x^2 - y \le 0, \\ \qquad x + y - 1.6 \le 0, \\ \qquad 0 \le x \le 1.6, \\ \qquad y \in \{0,1\}. \end{cases} \tag{17}$$

Its global minimum is 2 and global optimal solution is (0.5,1).

Problem 2[8]

© 2014 ACADEMY PUBLISHER

$$\begin{cases} \min \quad f(x,y) = 2x_1 + x_2 - y, \\ \text{s.t.} \quad x_1 - 2\exp(-x_2) = 0, \\ \quad -x_1 + x_2 + y \le 0, \\ \quad 0.5 \le x_1 \le 1.4, \\ \quad y \in \{0,1\}. \end{cases} \tag{18}$$

Its global minimum is 2.124 and global optimal solution is（1.375, 0.375, 1）.

Problem 3[8]

$$\begin{cases} \min \quad f(x,y) = -0.7y + 5(x_1 - 0.5)^2 + 0.8, \\ \text{s.t.} \quad -x_2 - \exp(x_1 - 0.2) \le 0, \\ \quad x_2 + 1.1y + 1 \le 0, \\ \quad x_1 - 1.2y - 0.2 \le 0, \\ \quad 0.2 \le x_1 \le 1, \\ \quad -2.22554 \le x_2 \le -1, \\ \quad y \in \{0,1\}. \end{cases} \tag{19}$$

Its global minimum is 1.07654and global optimal solution is（0.94194,-2.1,1）.

Problem 4[8]

$$\begin{cases} \min \quad f(x,y) = 7.5y_1 + 5.5y_2 + 5x_3 + 7x_4 + 6x_5, \\ \text{s.t.} \quad y_1 + y_2 - 1 = 0, \\ \quad x_6 - 0.9x_1[1 - \exp(-0.5x_4)] = 0, \\ \quad x_7 - 0.8x_2[1 - \exp(-0.4x_5)] = 0, \\ \quad x_6 + x_7 - 10 = 0, \\ \quad x_1 + x_2 - x_3 = 0, \\ \quad x_6 y_1 + x_7 y_2 - 10 = 0, \\ \quad x_4 - 10y_1 \le 0, \\ \quad x_5 - 10y_2 \le 0, \\ \quad x_1 - 20y_1 \le 0, \\ \quad x_2 - 20y_2 \le 0, \\ \quad x \ge 0, y \in \{0,1\}. \end{cases} \tag{20}$$

Its global minimum is 99. 245209 and global optimal solution is（13.362272, 3, 514237, 0, 1, 0）.

Problem 5[8]

$$\begin{cases} \min \quad f(x,y) = (y_1 - 1)^2 + (y_2 - 1)^2 + (y_3 - 1)^2, \\ \text{s.t.} \quad -\ln(1 + y_4) + (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2, \\ \quad y_1 + y_2 + y_3 + x_1 + x_2 + x_3 - 5 \le 0, \\ \quad x_1^2 + x_2^2 + x_3^2 + y_3^2 - 5.5 \le 0, \\ \quad y_1 + x_1 - 1.2 \le 0, \\ \quad y_2 + x_2 - 1.8 \le 0, \\ \quad y_3 + x_3 - 2.5 \le 0, \\ \quad y_4 + x_1 - 1.2 \le 0, \\ \quad y_2^2 + x_2^2 - 1.64 \le 0, \\ \quad y_3^2 + x_3^2 - 4.25 \le 0, \\ \quad y_2^2 + x_3^2 - 4.64 \le 0, \\ \quad x \ge 0, \\ \quad y \in \{0,1\}. \end{cases} \tag{21}$$

Its global minimum is 3.557463 and global optimal solution is（0.2, 1.280624, 1.954483, 1, 0, 0, 1）.

## ACKNOWLEDGMENT

The work is supported by the Foundations of National Natural Science China under Grant No.60962006 and No.61102008 and by the Foundations of research projects of State Ethnic Affairs Commission under Grant No.12BFZ005.

## REFERENCES

[1] I. Nowak and S. Vigerske, “LaGO: A(heuristic) branch and cut algorithm for nonconvex MINL-Ps,” *Cent. Eur. J. Oper. Res*., vol. 16, pp. 127–138, 2008.

[2] M. L. Bergamini, I. Grossmann, N. Scenna, and P. Aguirre, “An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms,” *Comput. Chem Eng*., vol. 32, no. 3, pp. 477–493, March 2008.

[3] F. V. Martin and W. P. Savelsbergh, “A generic view of Dantzig–Wolfe decomposition in mixed integer programming,” *Oper. Res. Lett*., vol.34, no. 3, pp. 296 – 306, May 2006.

[4] T. Yokota, M. Gen, and Y. X. Li, “Genetic algorithm for non-linear mixed integer programming problems and its applications,” *Compu. Ind. Eng.*, vol. 30, no. 4, pp. 905–917, September 1996.

[5] J. P. Chiou and F. S.Wang, “A hybrid method of differential evolution with application to optimal control problems of a bioprocess system,” in Proc. *IEEE World Congress on Computational Intelligence*, pp. 627–631, 1998.

[6] L. Lakshminarasimman and S. Subramanian, “ Hydrothermal coordination using modified mixed integer hybrid differential evolution,” *International Journal of Energy Technology and Policy*, vol. 5, no. 4, pp. 422–439, August 2007.

[7] Y. C. Lin, K. S. Wang, and F. S. Wang, “Plant scheduling and planning using mixed-integer hybrid differential evolution with multiplier updating,” in *Proc. IEEE Congress on Evolutionary Computation*, pp. 593–600, 2000.

[8] Y. C. Lin, K. S. Wang, and F. S. Wang, “A mixed-coding scheme of evolutionary algorithms to solve mixed-integer nonlinear programming problems,” *Computers Math. Applic*., vol. 47, no. 8-9, pp. 1295–1307, April-May 2004.

[9] R. Storn and K. Price, “Differential Evolution-a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces,” *Technical Report*, International Computer Science Institute, vol. 8, pp. 22–25, 1995.

[10] R. Storn and K. Price, “Differential Evolution–a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *J. Global Optim.*, vol. no. 4, pp. 341–359, December 1997.

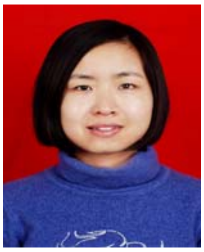
[11] A. Colorni, M. Dorigo, and V. Maniezzo, “An investigation of some properties of an ant algorithm,” in *Proce. Parallel Problem Solving from Nature Conference*, pp. 509–520, 1992.

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

© 2014 ACADEMY PUBLISHER

[13] S. Gao and J. Yang, "Swarm algorithm and its application," Beijing, China Water Conservancy and Hydropower Publishing, 2006.
[14] Y. C. Lin, F. S. Wang, and K. S. Huang, "A hybrid method of evolutionary algorithms for mixed integer nonlinear optimization problem," in *IEEE Proc. Evolutionary Computation*, pp. 2159–2166, 1999.
[15] Y. Gao, Y. Wang, and X. Du, "A two-level relaxed bound method for a nonlinear class of 0-1 knapsack problems," *Intelligent Information Management Systems and Technologies*, vol.3, no. 1, pp.461–470, 2005.
[16] Y. Gao, C. Xu, and J. Liu, "Linear programming relax PSO hybrid bound algorithm for a class of nonlinear programming problems," in *Proc. 2006 Int. Conf. Computational Intelligence and Security,* vol. 1, pp. 380–383, 2006.
[17] Y. Gao, C. Xu, and Y. Yang, "An outcome-space finite algorithm for solving linear multiplicative programming," *Appl. Math. Comput.*, vol. 179, no. 2, pp. 494–505, August 2006.
[18] Y. C. Lin, Y. C. Lin, W. C. Chang, and K. L. Su, "Evolutionary neural networks with mixed-integer hybrid differential evovolution," *Journal of Computers*, vol. 6, no. 8, pp. 1591–1596, Aug 2011.
[19] Y. Zhou and S. Pei, "A hybrid co-evolutionary partical swarm optimization algorithm for solving constrained engineering design problems," *Journal of Computers*, vol. 5, no. 6, pp. 965–972, Jun 2010.


**Yuelin Gao**, born in 1963, Ph. D., professor, Master supervisor. His main research interests currently include theory, methods and applications of optimization, intelligent information processing, financial computation and financial engineering.
Email:gaoyuelin@263.net



**Xia Chang**, born in 1982, Ph. D., lecturer. Her research interests include image sparse presentation, intelligence computation, and image understanding.
Email:changxia0104@163.com

**Junmei Liu**, born in 1980, Master. Her main research interests currently include theory, methods and applications of optimization and intelligent computation.
Email: liujunmei800911@163.com

© 2014 ACADEMY PUBLISHER