Int j simul model 14 (2015) 3, 525-538
ISSN 1726-4529 Original scientific paper

A VARIABLES CLUSTERING BASED DIFFERENTIAL EVOLUTION ALGORITHM TO SOLVE PRODUCTION PLANNING PROBLEM

Yang, K. W.; Zhang, P. L.; Ge, B. F. & Dou, Y. J.
College of Information System and Management, National University of Defense Technology, Changsha, Hunan, 410073, P. R. China
E-Mail: kayyang27@nudt.edu.cn, kayyang27@gmail.com, paulplayergg@gmail.com

Abstract
The scenario tree based multistage stochastic programming model is popular to describe production planning problem. However, as its high-dimensional variables and large-scale solution space, the addressed model is hardly to be solved in an acceptable time. To deal with this challenge, we propose a variables clustering based differential evolution algorithm which combines two novel strategies i.e. the children cluster based parallel evolution operations and the entirely randomized parameters for each child-individual. A case of weapons production planning is studied to validate the proposed algorithm. The results show that this algorithm has the fastest convergence and the best global searching capability in 6 test instances with different scales of the variables and the solution space, compared with classical differential evolution algorithm, genetic algorithm and particle swarm optimization algorithm.
(Received, processed and accepted by the Chinese Representative Office.)

Key Words: Weapons Production Planning, Multistage Stochastic Programming, Differential Evolution, Variable Clustering

1. INTRODUCTION

The scenario tree based stochastic programing as a kind of mathematical programming technique is more and more popular to model the multistage uncertain optimization problem [1-3]. The scenario tree can model the stochastic evolution processes of the uncertain parameters as a group of discrete scenarios without the need to know the detailed probability distribution. Moreover, there is no limitation for the number of the uncertain parameters. Hence, in the context of production planning problem, the uncertain parameters of the capability requirements can be easily interpreted as a scenario tree. Furthermore, the stochastic programming technique can adjust the production time and the production budget to hedge against various scenarios.

However, the challenge of this application is that the variable dimensions and the solution space exponentially increase with the planning stage and the scenarios. Moreover, they are usually larger than 1000 dimensions and 101000 solutions respectively in the context of weapons production planning. In recent literatures [4-6], the scenario tree based multistage stochastic programming model is solved by some business software such as CPLEX. However, the model of production planning problem belongs to the nonlinear integer programming with real exponent and it is hard to be converted into the standard form which could be solved by CPLEX. Some vigorous mathematical approaches are also studied to solve the multistage stochastic programming model, but these approaches require the model to satisfy some strict restraints such as convex programming [7], linear programming [8] and/or 0-1 mixed integer programming [9]. Unfortunately, the multistage stochastic programing model in the context weapons production planning does not satisfy these constraints.

In recent years, more and more researchers developed the heuristic and/or meta-heuristic algorithms to attack the large-scale production planning problem in many fields. Wu and

Chuang developed a new heuristic search algorithm to solve the dedicated and flexible capacity planning problems [10]. Jiao et al. proposed a generic algorithm (GA) with specific encoding scheme to synchronize product portfolio generation and selection in production planning [11]. Chen and Lin introduced a binary particle swarm optimization algorithm (PSO) with dynamic inertia weight and mutation mechanism for the production planning problem in the thin film transistor Array process [12]. Ramezanian et al. combined GA with tabu search to solve the aggregate production planning problem [13]. Interested readers may refer to [14-16].

However, neither did these studies consider the scenario tree based variables structure in multistage stochastic programming model nor systematically develop an evolutionary algorithm to specifically handle the high-dimensional variables in production planning. At the same time, differential evolution (DE) which is considered as an important branch of the evolutionary algorithms is widely used to solve optimization problems in many fields [17-20]. The simple mutation operation and one-on-one competition strategy of DE reduce the complexity of traditional genetic operations, which brings to a strong global convergence and robustness.

Based on these surveys, we develop a DE algorithm to solve the multistage stochastic programming model in the context of production planning problem. In order to handle the high-dimensional variables and the large-scale solution space, we propose a variable clustering approach to join in the developed algorithm, it contains the children cluster based parallel evolution operations and the entirely randomized parameters for each child-individual. To the best of our knowledge, this improved DE algorithm is first proposed here and could be applied to other engineering fields.

The structure of this paper is organized as follows. Section 2 gives the multistage stochastic programming model for weapon production planning problem. Section 3 elaborates how the variables clustering approach improves the classical DE algorithm. Section 4 provides a case to illustrate the advantages of the improved algorithm. Conclusions and future work are drawn in Section 5.

2. MULTISTAGE STOCHASTIC PROGRAMMING MODEL

2.1 List of symbols

Weapons production planning is defined as the selection of the optimal weapon model from each category and then the optimization of the production time and the production budget of each selected weapon in multi-periods. The objective is to make the selected weapons continuously meet the military capability requirements over the whole planning horizon. However, the capability requirements are evolving and uncertain with ever more demanding military tasks. In this paper we take weapons production planning as a kind of multistage uncertain optimization problem.

Variables

k_m:	Sequence number the kth weapon model in the mth category;
$t_{k_m,p}$:	Production time of the k_mth weapon model in the pth period;
$q_{k_m,p}$:	Production budget of the k_mth weapon model in the pth period;

Parameters

m:	Sequence number of the weapon categories, $m = 1, 2, \dots, M$;
$Rt_{p,c}$:	Time requirement of the cth capability requirement;
$Rq_{p,c}$:	Quantity requirement of the cth capability requirement;
p:	Sequence number of the period over the whole planning horizon, $p = 1, 2, \dots, P$;
c:	Sequence number of the capability requirements, $c = 1, 2, \dots, C$;

$a_{k_m,p,c,l}$:	The lth coefficient of the L-order polynomial of the k_mth weapon model for the cth capability in the pth period;
$\alpha_{k_m,p,c,l}$:	The lth degree of the L-order polynomial of the k_mth weapon model for the cth capability in the pth period;
$b_{k_m,p,c,h}$:	The hth coefficient of the H-order polynomial of the k_mth weapon for the cth capability in the pth period;
$\beta_{k_m,p,c,h}$:	The hth degree of the H-order polynomial of the k_mth weapon for the cth capability in the pth period;
$t'_{k_m,p}$:	Effective production time of the k_mth weapon model in the pth period;
$q'_{k_m,p}$:	Effective production budget of the k_mth weapon model in the pth period;
T:	Constant time span of each period in the planning horizon;
$WC_{k_m,p,c}$:	The cth real capability value of the k_mth weapon model in the pth period;
$WC'_{k_m,p,c}$:	The cth effective capability value of the k_mth weapon model in the pth period;
$WC_{p,c}$:	The cth capability value of the weapons portfolio in the pth period;
$w_{m,p,c}$:	Weight of the mth weapon category for the cth capability requirement in the pth period;
$ql_{k_m,p}$:	Lower bound of the production budget of the mth weapon category in the pth period;
$qu_{k_m,p}$:	Upper bound of the production budget of the mth weapon category in the pth period;
Q_p:	Overall budget constraint for the weapons portfolio in the pth period;
z_p:	Sequence number of the possibilities of the capability requirements, $z_p = 1, 2, \ldots, Z_p$;
s:	Sequence number of the scenarios, $s = 1, 2, \ldots, S$;
n_p:	Sequence number of the nodes in the pth period of the scenario tree, $n_p = 1, 2, \ldots, N_p$.

2.2 Overview of weapons production planning

Suppose that there are C capability requirements and M weapon categories, in which each capability requirement is supported by more than one weapon category and vice versa. Each weapon category contains a number of weapon models. For each weapon model, the capability values depend on the corresponding production time and production budget. In each weapon category, the weapon models have different capability values when they are assigned with equivalent production time and production budget. To understand the concept of weapons production planning, we need to know the weapons multi-period production process which is introduced in Fig. 1.

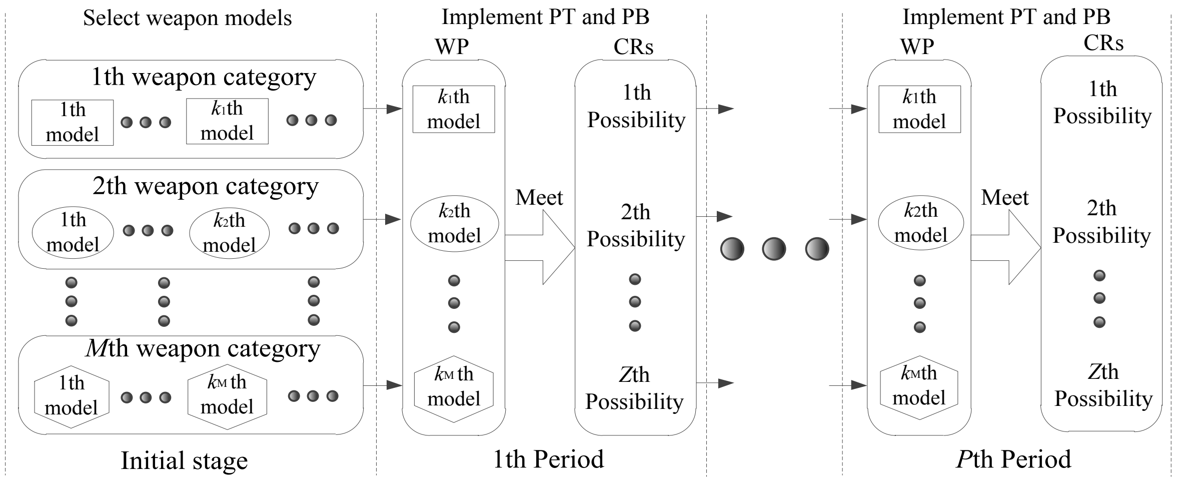

Figure 1: Weapons multi-period production process.

In Fig. 1, the weapons production process consists of two kinds of variables, one is to select the optimal weapon model from each category which will be implemented at initial stage, and the other one is to optimize the production time (*PT*) and the production budget (*PB*) of each selected weapon which will be implemented in multi-period. All of the selected weapons constitute a weapons portfolio (WP). The objective is to minimize the capability gaps between the weapons portfolio and the capability requirements (CRs) over the planning horizon. In this context, both of the capability deficiencies and surpluses between capabilities of the weapons portfolio and the capability requirements are recognized as the capability gaps.

2.3 Formulations of capabilities for the weapons portfolio

To calculate the capability gaps between the weapons portfolio and the capability requirements, we need to know the capabilities of the weapons portfolio. Each capability of the weapons portfolio comes from the corresponding one of each weapon in this portfolio. Thus, we firstly calculate the capability value of each weapon. Given the production time and the production budget in each period, if this weapon does not have the c^{th} capability, $WC_{k_m,p,c} = 0$; otherwise, $WC_{k_m,p,c}$ is calculated as follows:

$$WC_{k_m,p,c} = \begin{cases} (\sum_{l=1}^{L} a_{k_m,p,c,l} \times t_{k_m,p}^{\alpha_{k_m,p,c,l}}) \times (\sum_{h=1}^{H} b_{k_m,p,c,h} \times q_{k_m,p}^{\beta_{k_m,p,c,h}}); & p == 1 \\ (\sum_{l=1}^{L} a_{k_m,p,c,l} \times t'^{\alpha_{k_m,p,c,l}}_{k_m,p}) \times (\sum_{h=1}^{H} b_{k_m,p,c,h} \times q'^{\beta_{k_m,p,c,h}}_{k_m,p}); & 1 < p < P \end{cases} \tag{1}$$

$$\text{s.t.} \quad t'_{k_m,p} = t_{k_m,p} \times \left(1 + \sum_{p'=1}^{p-1} (t_{k_m,p'} \Big/ (T \times (p - p'))) \right); \tag{2}$$

$$q'_{k_m,p} = q_{k_m,p} \times \left(1 + \sum_{p'=1}^{p-1} (q_{k_m,p'} \Big/ (qu_{k_m,p'} \times (p - p'))) \right); \tag{3}$$

$$a_{k_m,p,c,l}, b_{k_m,p,c,l}, \alpha_{k_m,p,c,l}, \beta_{k_m,p,c,l} \in \mathbb{R}^+ \tag{4}$$

$$WC'_{k_m,p,c} = \begin{cases} WC_{k_m,p,c} \times (t_{k_m,p} / Rt_{p,c}) & t_{k_m,p} < Rt_{p,c} \\ WC_{k_m,p,c} & t_{k_m,p} == Rt_{p,c} \\ WC_{k_m,p,c} \times (Rt_{p,c} / t_{k_m,p}) & t_{k_m,p} > Rt_{p,c} \end{cases} \tag{5}$$

where eq. (5) represents that $WC'_{k_m,p,c}$ is smaller while the disparity between $t_{k_m,p}$ and $Rt_{p,c}$ is larger. This is because the weapon has not been completely developed when $t_{k_m,p} > Rt_{p,c}$, on the contrary, the weapon lacks enough funding to maintain the achieved capability when $t_{k_m,p} < Rt_{p,c}$.

After acquiring the c^{th} capability value of each weapon model, we can calculate the corresponding capability value of the weapons portfolio through weighted sum, which is calculated as follows:

$$WC_{p,c} = \sum_{m=1}^{M} w_{m,p,c} \times WC'_{k_m,p,c} \tag{6}$$

where $w_{m,p,c} = 0$, if the m^{th} weapon category does not have the c^{th} capability.

2.4 Formulations of the uncertain capability requirements

In a practical defence engineering environment, the capability requirements are incrementally evolving over the whole planning horizon while they have uncertain values in each period. Considering this property, the uncertain capability requirements are formulated as follows:

$$\{R_{p,z,c} \,|\, R_{p,z,c} = (Rt_{p,z,c}, Rq_{p,z,c})\} \tag{7}$$

$$\text{s.t.} \quad Rt_{p',z,c} \le Rt_{p,z,c}, Rq_{p',z,c} \le Rq_{p,z,c}, \quad p' < p, \tag{8}$$

$$p, p' = 1,2,...,P, \;\; c = 1,2,...,C, \;\; z_p = 1,2,...,Z_p \tag{9}$$

$$Rq_{p,z,c} \in \mathbb{R}^+, Rt_{p,z,c}, P, C, Z_p \in \mathbb{N}^+ \tag{10}$$

where eq. (7) represents that the c^{th} capability requires the value $Rq_{p,z,c}$ at the time of $Rt_{p,z,c}$ in the p^{th} period. $R_{p,z,c}$ has Z_p possibilities in each period. Constraint (8) represents $Rq_{p,z,c}$ and $Rt_{p,z,c}$ increase with the period. An example of 3-period possible evolution processes of the capability requirements is given in Fig. 2.

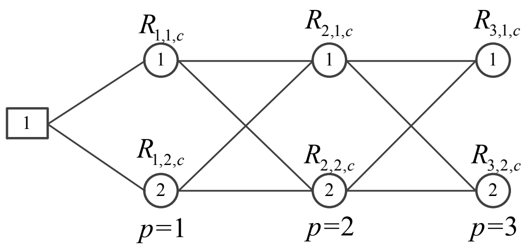

Figure 2: 3-period evolution of the capability requirements with two possibilities.

In Fig. 2, each circular node represents one possible state of the capability requirements. The rectangular node is a root node which represents the initial stage. Since the capability requirement in each period is independent with each other, and the capability requirements have two possibilities in each period, i.e. $Z_p = 2$. Thus, there are total 8 possible evolution processes of the capability requirements.

2.5 Multistage stochastic programming model

The multistage stochastic programming model provides different production times and production budgets to hedge against each possible group of capability requirements. Since the variables in p^{th} period correlate with all of the ones in the previous $(p–1)^{th}$ period, the scenario tree could be used to represent the variables structure corresponding to the possible evolution processes of the capability requirements. The possibilities of the capability requirements equal to the branches of the scenario tree in each period. A scenario tree with 2-2-2 branches is shown in Fig. 3.

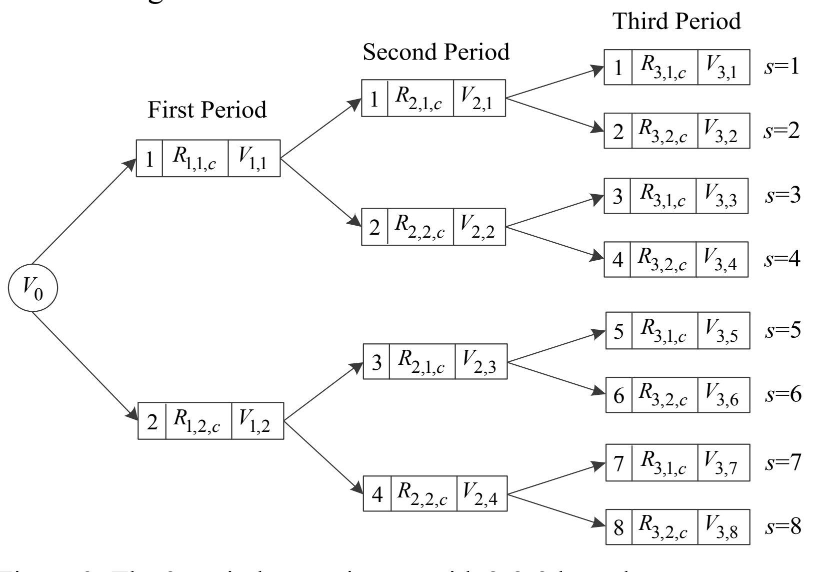
Third Period
Second Period
First Period
V_0
1 $R_{1,1,c}$ $V_{1,1}$
2 $R_{1,2,c}$ $V_{1,2}$
1 $R_{2,1,c}$ $V_{2,1}$
2 $R_{2,2,c}$ $V_{2,2}$
3 $R_{2,1,c}$ $V_{2,3}$
4 $R_{2,2,c}$ $V_{2,4}$
1 $R_{3,1,c}$ $V_{3,1}$ s=1
2 $R_{3,2,c}$ $V_{3,2}$ s=2
3 $R_{3,1,c}$ $V_{3,3}$ s=3
4 $R_{3,2,c}$ $V_{3,4}$ s=4
5 $R_{3,1,c}$ $V_{3,5}$ s=5
6 $R_{3,2,c}$ $V_{3,6}$ s=6
7 $R_{3,1,c}$ $V_{3,7}$ s=7
8 $R_{3,2,c}$ $V_{3,8}$ s=8

Figure 3: The 3-period scenario tree with 2-2-2 branches.

In Fig. 3, the path from the root node (circular node) to a leaf node (rectangular node) is called a scenario s, i.e. a possible evolution process of the capability requirements over the entire planning horizon. The variable set V_0 in root node corresponds to the selected weapon models to encounter all of the scenarios. Then, the variable set $V_{p,n}$ corresponds to the production time and the production budget in each leaf node. For example, if there are five weapon categories, V_0 is:

$$V_0 = \{k_1, k_2, k_3, k_4, k_5\} \tag{11}$$

The variable set in any one leaf node such as $V_{2,3}$ is:

$$V_{2,3} = \{t_{k_1,2,3}, t_{k_2,2,3}, t_{k_3,2,3}, t_{k_4,2,3}, t_{k_5,2,3}, q_{k_1,2,3}, q_{k_2,2,3}, q_{k_3,2,3}, q_{k_4,2,3}, q_{k_5,2,3}\} \tag{12}$$

The objective of weapons production planning is to minimize the weighted sum of the capability gaps between the weapons portfolio and the capability requirements in each leaf node. Thus, the multistage stochastic programming model for weapons production planning is formulated as:

$$\min \sum_{p=1}^{P}\sum_{n_p=1}^{N_p} \pi(n_p) \times \sum_{c=1}^{C} \left| WC_{p,n,c} - Rq_{p,z,c} \right| \tag{13}$$

$$\text{s.t.} \quad z_p = \begin{cases} Z_p; & \operatorname{mod}(n_p/Z_p) = 0 \\ \operatorname{mod}(n_p/Z_p); & \operatorname{mod}(n_p/Z_p) \neq 0 \end{cases} \tag{14}$$

$$N_p = \prod_{p'=1}^{p} Z_{p'} \qquad P, N_p, Z_p \in \mathbb{N}^{+} \tag{15}$$

$$t'_{k_m,p,n} = t_{k_m,p,n} \times \left(1 + \sum_{p'=1}^{p-1} \left(f_{pre}^{p'}(t_{k_m,p,n}) \middle/ (T \times (p - p')) \right) \right); \tag{16}$$

$$q'_{k_m,p,n} = q_{k_m,p,n} \times \left(1 + \sum_{p'=1}^{p-1} \left(f_{pre}^{p'}(q_{k_m,p,n}) / (qu_{k_m,p'} \times (p - p')) \right) \right) \tag{17}$$

$$0 < t_{k_m,p,n} \le T; \quad t_{k_m,p,n},\ T \in \mathbb{N}^{+} \tag{18}$$

$$ql_{k_m,p} \le q_{k_m,p,n} \le qu_{k_m,p}; \quad q_{k_m,p,n}, ql_{k_m,p}, qu_{k_m,p} \in \mathbb{N}^{+} \tag{19}$$

$$\sum_{m=1}^{M} q_{k_m,p,n} \le Q_p \tag{20}$$

where $\pi(n_p)$ is considered as the unconditional probability of the n_pth possibility of the capability requirements in each node. Constraints (14, 15) describe the relation between n_p and Z_p. mod(•) represents the remainder operation. $f_{pre}^{p'}(\bullet)$ in constraints (16, 17) locates the variables of the ancestor node in the p'th period. For example, as shown in Fig. 3, if $p = 3$, $n_p = 8$, then $f_{pre}^{1}(t_{k_m,3,8}) = t_{k_m,1,2}$, $f_{pre}^{2}(t_{k_m,3,8}) = t_{k_m,2,4}$. It means that the variables corresponding to different scenarios at the same period must be equal if those scenarios share the same ancestor node at this period. Furthermore, the variables in all leaf nodes (circular nodes) must share the same variables of the weapon models in the root node (rectangular node). Constraint (18) and (19) represent the available range of the production time and the production budget, respectively. Constraint (20) represents the overall available budget for the weapons portfolio in each period.

3. ALGORITHM DEVELOPMENTS

3.1 Classical DE algorithm for the multistage stochastic programming model

We first develop the classical differential evolution (CDE) algorithm to solve the addressed multistage stochastic programming model. It is introduced as follows:

Step 1 (Population Initialization). Generate the value of each variable randomly in the available range according to constraints (18, 19). The individual encoding is shown as follows:

$$V_0\ V_{1,1},...,V_{1,N},...,V_{p,n},...,V_{P,N}\begin{cases}\overbrace{k_1,...,k_m,...,k_M}^{V_0}\\ \overbrace{t_{k_1,p,n},...,t_{k_m,p,n},...,t_{k_M,p,n}\ q_{k_1,p,n},...q_{k_m,p,n},...,q_{k_M,p,n}}^{V_{p,n}}\end{cases}\tag{21}$$

Step 2 (Evaluation). For each current individual, calculate the objective value by eq. (13), and then calculate the violation of the overall budget constraint by eq. (20). Record the current global optimization individual. If there is no individual who satisfies the overall budget constraint, take the individual with minimum violated degree of the budget constraint as the global optimization one.

Step 3 (Mutation). Execute mutation operation which is formulated as:

$$U_i = X_{i_1} + F\times(X_{i_2} - X_{i_3})\tag{22}$$

where X_{i_1}, X_{i_2} and X_{i_2} are the randomly chosen individuals from the current population *X*. The mutation scaling factor *F* is a real constant number which is often set to 0.5 [21].

Step 4 (Crossover). Select each variable of the current individual X_i with certain probability to replace the corresponding bit of the temporary individual U_i, which is formulated as:

$$U_i(j)=\begin{cases}U_i(j), & R_j(0,1)<cr\\ X_i(j), & otherwise\end{cases}\tag{23}$$

where $R_j(0, 1)$ is a uniform random number between 0 and 1, and *cr* is the crossover rate which is often set to 0.9 [21].

Step 5 (Handling constraints of variables). Judge each bit in each individual whether or not violate the constraints (18, 19) in the temporary population. The bit which violates the corresponding constraint randomly obtains a number as its value within the available range.

Step 6 (Selection). Evaluate the objective value and the violation of budget constraint (20) for each temporary individual U_i, and then select the better individual into next iteration.

Step 7 (Loop). Record the current global optimization individual. Repeat steps 3 to 7 until the maximum number of iterations or the desired solution is obtained.

3.2 Variables clustering approach

According to the scenario tree in Fig. 3, the objective function eq. (13) is the weighted sum of all sub-objective value corresponding to each leaf node. We define the node objective function as follows:

$$f_{p,n}=\pi(n_p)\times\sum_{c=1}^{C}\left|WC_{p,n,c}-Rq_{p,z,c}\right|\tag{24}$$

where $f_{p,n}$ is correlated with the variables of the node itself, and the ancestor nodes in the same scenario and the root node.

For example, in Fig. 3, the node objective functions in the scenario 1 and scenario 8 are respectively presented as follows:

$$f_{3,1}=f(V_0,V_{1,1},V_{2,1},V_{3,1});\ f_{2,1}=f(V_0,V_{1,1},V_{2,1});\ \ f_{1,1}=f(V_0,V_{1,1})\tag{25}$$

$$f_{3,8}=f(V_0,V_{1,2},V_{2,4},V_{3,8});\ f_{2,4}=f(V_0,V_{1,2},V_{2,4});\ f_{1,2}=f(V_0,V_{1,2})\tag{26}$$

Obviously, the node objective functions in different scenarios are correlated with each other via V_0. Thus, if V_0 is invariant, all of the leaf variable sets can be clustered into two independent parts.

In Fig. 4, the two children clusters are independent with each other. $V_{1,1}$ and $V_{1,2}$ are respectively recognized as the root node variable set in corresponding children cluster. If $V_{1,1}$ and $V_{1,2}$ are invariant, the else variable sets in each children cluster can be partitioned into two grandchildren clusters, respectively. Hence, for any scale of scenario tree, the variables can be clustered if the corresponding conditions are satisfied. Additionally, the individual and the population generated by the i^{th} children (resp. grandchildren) cluster are named as the i^{th} child-individual (resp. grandchild-individual) and the i^{th} children-population (resp. grandchildren-population), respectively.

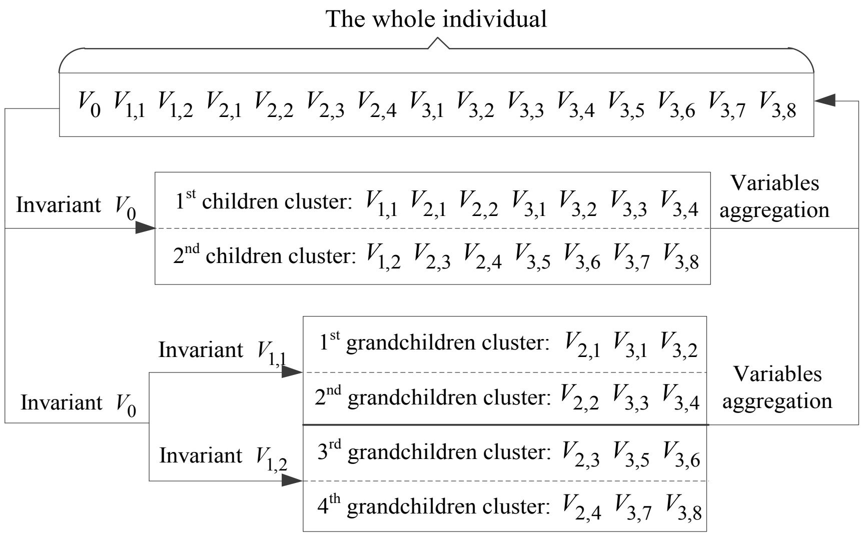
The whole individual
V_0 $V_{1,1}$ $V_{1,2}$ $V_{2,1}$ $V_{2,2}$ $V_{2,3}$ $V_{2,4}$ $V_{3,1}$ $V_{3,2}$ $V_{3,3}$ $V_{3,4}$ $V_{3,5}$ $V_{3,6}$ $V_{3,7}$ $V_{3,8}$
Invariant V_0
1st children cluster: $V_{1,1}$ $V_{2,1}$ $V_{2,2}$ $V_{3,1}$ $V_{3,2}$ $V_{3,3}$ $V_{3,4}$
2nd children cluster: $V_{1,2}$ $V_{2,3}$ $V_{2,4}$ $V_{3,5}$ $V_{3,6}$ $V_{3,7}$ $V_{3,8}$
Variables aggregation
Invariant $V_{1,1}$
Invariant V_0
Invariant $V_{1,2}$
1st grandchildren cluster: $V_{2,1}$ $V_{3,1}$ $V_{3,2}$
2nd grandchildren cluster: $V_{2,2}$ $V_{3,3}$ $V_{3,4}$
3rd grandchildren cluster: $V_{2,3}$ $V_{3,5}$ $V_{3,6}$
4th grandchildren cluster: $V_{2,4}$ $V_{3,7}$ $V_{3,8}$
Variables aggregation

Figure 4: The variables clustering for the scenario tree with 2-2-2 branches.

After the variables clustering, each children cluster could implement mutation and crossover independently. We consider the randomized parameters to enlarge the searching space of the mutation and crossover for each children cluster. Some researchers utilized different probability distribution models as the mutation scaling factor *F*, such as uniform distribution [22], Gaussian distribution [23] and Cauchy distribution [24]. Yang introduced the uniform random number as the crossover rate *cr* [25]. Based on these surveys, we first take *F* and *cr* with random numbers simultaneously in this context. Thus, *F* and *cr* are set as follows:

$$\begin{cases} F = N_{z,i}(0,1) \\ cr = R_{z,i}(0,1) \end{cases} \tag{27}$$

where $N_{z,i}(0,1)$ denotes a Gaussian random number with mean 0 and standard deviation 1, and $R_{z,i}(0,1)$ is an uniform random number between 0 and 1. Compared with the constant setting for *F* and *cr*, eq. (27) gives the stochastic values for them respectively to operate the i^{th} child individual in the z^{th} children cluster population. Since *cr* has to be between in 0 and 1, we adopt the uniform random number as *cr*. *F* does not need to be between in 0 and 1, thus we adopt the Gaussian random number, in order to give it a larger variant range.

The selection operation in the CDE algorithm either reserves the entire bits of the i^{th} current individual or updates them completely with the entire bits of the i^{th} temporary individual. In the context of variables clustering, we can select each temporary child-individual and/or temporary grandchild-individual independently to partially update the current individual, so that the selection operation could be implemented more accurately and efficiently. The procedure of the child-individual based selection is proposed as follows:

Step 1: Set $i = 0$; $j = 0$. Obtain the number of children clusters Z_1 and the number of grandchildren clusters Z_2. It is assumed that the number of grandchildren clusters in each children cluster is equivalent with each other.

Step 2: Take $i = i + 1$, then implement the selection operation as shown in Fig. 4 between the ith current child-individual and the ith temporary one.

Step 3: Judge whether the ith current child-individual is updated. If no, go to Step 4; else judge whether $i == Z_1$. If yes, go to Step 7, else go to Step 2.

Step 4: Judge whether the value of each root node variable in the ith current child-individual is equivalent to the corresponding one in the ith temporary one or not. If yes, go to Step 5; else judge whether $i == Z_1$. If yes, go to Step 7, else go to Step 2.

Step 5: Take $j = j + 1$. Implement the selection operation as shown in Fig. 4 between the jth current grandchild-individual and the jth temporary one.

Step 6: Judge whether $j == Z_2$. If no, go to Step 5, else take $j = 0$ and judge whether $i == Z_1$. If yes, go to step 7, else go to step 2.

Step 7: Judge whether the temporary individual and the updated current individual is better than the current global optimal one or not, respectively. If yes, update the current global optimal individual. Then go to next iteration.

4. COMPUTATIONAL RESULTS

4.1 Problem description

A 3-period intelligence, surveillance and reconnaissance (ISR) weapons production planning problem is studied to validate the proposed algorithm. ISR weapons are indispensable components of military operations, they must work together to provide commanders and soldiers with a comprehensive understanding of battlefield situation. Thus, multiple ISR weapons need to be planned to satisfy the capability requirements. There are five weapon categories need to be considered, and the corresponding variables and their ranges are shown in Table I.

Table I: The variables and their ranges of the weapon categories.

Weapon	k_m	$t_{k_m,p}$	$q_{k_m,1}$	$q_{k_m,2}$	$q_{k_m,3}$
RS	[1, 5]	(0, 20)	(0, 300)	(0, 400)	(0, 300)
AWCS	[1, 4]	(0, 20)	(0, 200)	(0, 300)	(0, 250)
MHAUAS	[1, 6]	(0, 20)	(0, 100)	(0, 150)	(0, 100)
BLSDL	[1, 3]	(0, 20)	(0, 50)	(0, 100)	(0, 80)
C2S	[1, 4]	(0, 20)	(0, 120)	(0, 160)	(0, 140)

All of the notations in Table I are listed in the following: (1) RS is the reconnaissance satellite; (2) AWCS is the airborne warning and control system; (3) MHAUAS is the medium to high altitude unmanned aerial system; (4) BLSDL is the beyond line of sight data link; (5) C2S is the command and control system. Each category weapon has three kinds of variables, i.e. model selection variable k_m, production time variable $t_{k_m,p}$ and production budget variable $q_{k_m,p}$. For example, the data on RS in Table I represents that it has five optional weapon models. The production time must be planned within a given interval i.e. 0 to 20 in each period and the unit of production time is month. The production budget must be planned within 0 to 300, 0 to 400 and 0 to 300 for each period, respectively, and the unit of production budget is million dollars.

To explore the performance of the proposed algorithm, we construct the scenario trees from 3-3-3 to 8-8-8 branches corresponding to the different scale of the possible evolving capability requirements. The related data are shown as in Table II.

Table II: The data on six instances under different scale of the scenario tree.

Branches	**3-3-3**	**4-4-4**	**5-5-5**	**6-6-6**	**7-7-7**	**8-8-8**
Children clusters	3	4	5	6	7	8
Grandchildren clusters	9	16	25	36	49	64
Leaf nodes	39	84	155	258	399	584
Variables	395	845	1555	2585	3995	5845
Constraints	434	925	1710	2843	4394	6429
Scale	$>10^{780}$	$>10^{1680}$	$>10^{3100}$	$>10^{5160}$	$>10^{7980}$	$>10^{11680}$

The scenario trees with different branches have the same structure which was shown in Fig. 3. Since there are five weapon categories to be planned, each leaf node has corresponding five production time variables and five production budget variables.

4.2 Performance comparison between CDE, GA and PSO

We first validate the advantages that we adopt DE as the evolutionary algorithm framework in this context. Thus, we compare the performances of CDE, GA and PSO. For the variables take decimal integer number encoding, the heuristic crossover based GA [26] is appropriate to solve this case. Meanwhile, one kind of classical PSO algorithm [27] is also employed to solve this case. Therefore, we use CDE, GA and PSO, respectively, to solve the addressed case with 3-3-3 branches scenario tree. The evolution curves of them which correspond to the average results of 30 independent runs, respectively, are shown in Fig. 5.

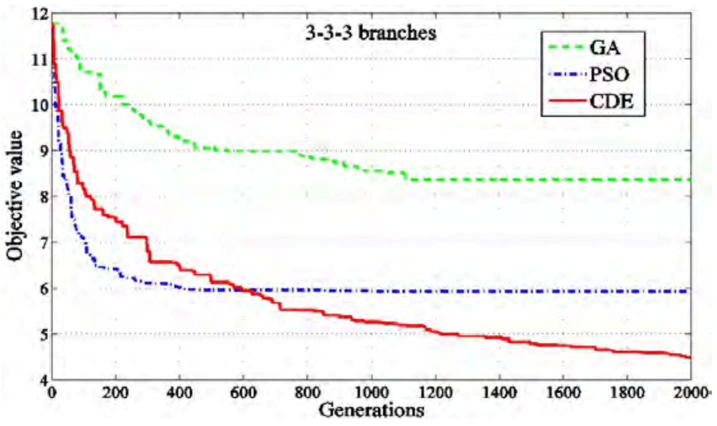

Figure 5: Evolution curves comparison under 3-3-3 branches scenario tree.

In Fig. 5, the evolution result of CDE is superior to GA and PSO. The final global optimization values of GA, PSO and CDE are 7.4461, 5.9348 and 4.4969 respectively. Furthermore, GA and PSO are trapped into local optimum in the later periods of iterations. Meanwhile, the population of CDE still keeps diversity and could obtain the better global optimum. Nevertheless, CDE does not converge in the final stage of iterations according to its evolutionary curve. Moreover, the convergence speed deteriorates with the increase of the scale of the addressed case. We use the scenario tree with 5-5-5 branches to show this trend, which is shown in Fig. 6.

In Fig. 6, both of GA and PSO converge to a local optimal value during the 1000~1200 generations. In contrast, CDE still searches the better solution but does not converge till the finish of the iterations. The final global optimization values of GA, PSO and CDE are 8.3699, 7.8139 and 7.7908 respectively. Therefore, we can conclude that CDE has the best global searching capability among these three algorithms while it needs a faster speed of the convergence.

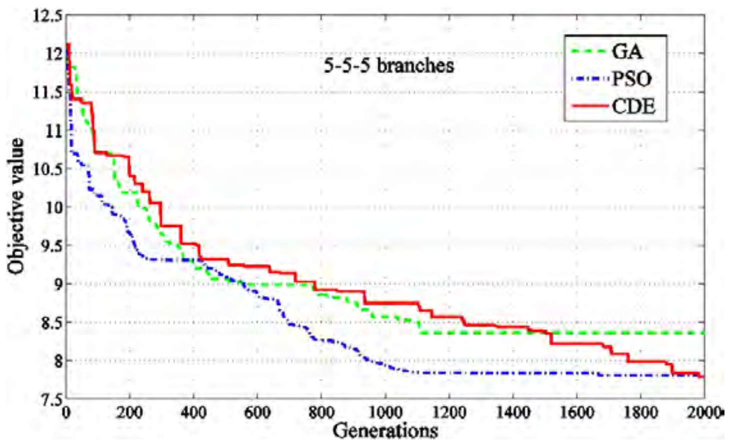

Figure 6: Evolution curves comparison under 5-5-5 branches scenario tree.

4.3 Validation of the variables clustering based DE algorithm

The variables clustering approach primarily brings two progresses for CDE: the first one is the child-individual based parallel evolution operations, and the second one is the randomized parameters setting for the mutation and crossover of each children-population. We will analyse the effect of each progress itself and the combinatorial effect of the two progresses. Thus, the CDE with first progress is named as VDE; the CDE with second progress is named as CDE-R; the CDE combing the two progresses simultaneously is named as VDE-R.

Therefore, we use VDE, CDE-R and VDE-R, respectively, to solve the addressed case with 3-3-3 branches scenario tree. The evolution curves of them which correspond to the average results of 30 independent runs, respectively, are shown in Fig. 7.

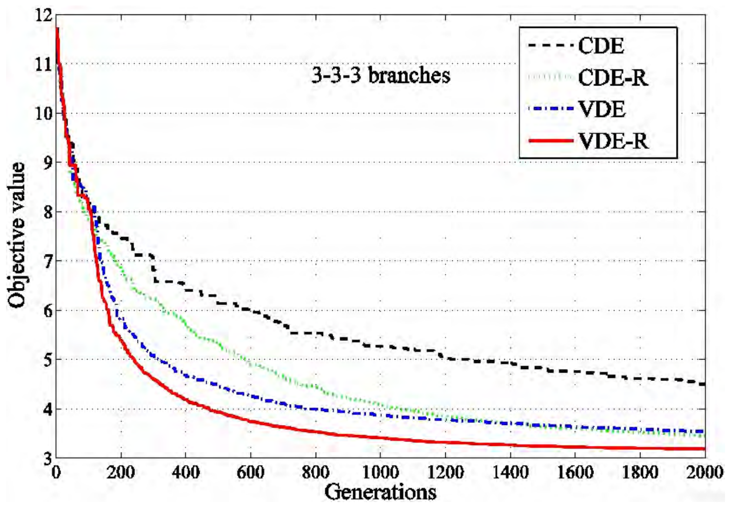

Figure 7: Evolution curves comparison under 3-3-3 branches.

In Fig. 7, all of the three improved algorithms are superior to the CDE. VDE-R has the best optimization result, and it is hard to distinguish the performances of CDE-R and VDE. The final global optimization values of VDE, CDE-R and VDE-R are 3.5444, 3.4570 and 3.1876, respectively. Furthermore, VDE, CDE-R and VDE-R achieve the value 4.4969, which the CDE finally obtained at approximate 490, 760 and 320 generations respectively. Thus, we conclude that each progress and the combinatorial progress are all effective for the CDE. Moreover, VDE-R not only has the best global optimization capability but also has the fastest speed of convergence. Additionally, VDE has the faster convergence speed than CDE-R.

To further confirm the conclusion, we use VDE, CDE-R and VDE-R, respectively, to solve the addressed case with 5-5-5 branches scenario tree. The evolution curves of them which correspond to the average results of 30 independent runs, respectively, are shown in Fig. 8.

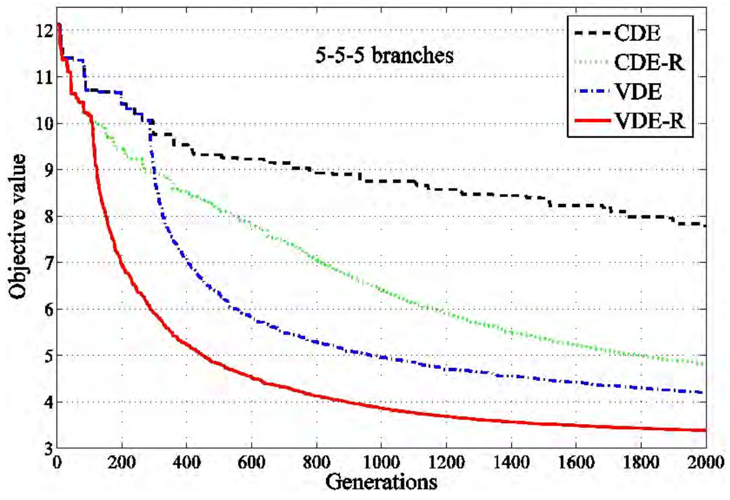

Figure 8: Evolution curves comparison under 5-5-5 branches.

According to Fig. 8, the results on 5-5-5 branches confirm the conclusions from the 3-3-3 branches. The three improved algorithms have the outstanding performance compared with CDE. Thus, they are also superior to the GA and PSO according to combing the results in Figs. 7 and 9. Furthermore, the final global optimization values of VDE, CDE-R and VDE-R are 4.1989, 4.8162 and 3.3835, respectively. It is implied that although both of the scale of variables and the solution space enlarge from 3-3-3 branches to 5-5-5 branches, the three improved algorithms perform far less deteriorations than CDE, especially the VDE-R.

4.4 Statistical comparison under increasing scale of the scenario tree

Next, we analyse the performances of CDE and the three improved algorithms in each instance which is shown in Table II, and the mean values of 30 independent runs corresponds to each algorithm are shown in Fig. 9.

VDE-R performs the best, followed by VDE and CDE-R, and the worst is CDE. When the scale of the case increases to 7-7-7 branches, the performances of the three improved algorithms deteriorate sharply. Nevertheless, VDE-R performs the best effectiveness between 3-3-3 to 6-6-6 branches, in other words, when the variables scale does not exceed more than approximate 2500 dimensions.

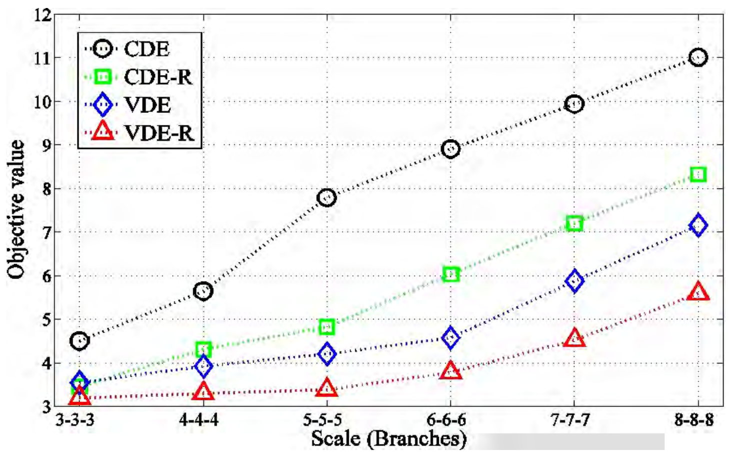

Figure 9: Comparison under different scales.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we developed a differential evolution algorithm to solve the multistage stochastic programming model in weapons production planning, in which a variables clustering based approach is proposed to handle the high-dimensional variables and the large-scale solution space. That is, the variables are partitioned into several children clusters, and which are implemented the DE evolution operations independently. Meanwhile, the mutation scaling factor *F* and the crossover rate *cr* are set as Gaussian random number and uniform random number, respectively, for each child-individual.

It should be noted that the variables clustering based DE algorithm still shows deterioration when the variables scale is larger than approximate 2500 dimensions. Therefore, we will consider the cooperative coevolution [28] to enrich the variables clustering approach in future research. Furthermore, more practical cases will be studied to further validate and improve the variables clustering based DE algorithm.

ACKNOWLEDGEMENT

This research is supported in part by the National Science Foundation of China under Contract no. 71031007, 71001104 and 71201168. We are grateful for these financial supports. We are also grateful to the anonymous reviewers for their valuable comments and constructive criticism.

REFERENCES

[1] Ahmed, S.; Sahinidis, N. V. (2003). An approximation scheme for stochastic integer programs arising in capacity expansion, *Operations Research*, Vol. 51, No. 3, 461-471, doi:10.1287/opre.51.3.461.14960

[2] Geng, N.; Jiang, Z.; Chen, F. (2009). Stochastic programming based capacity planning for semiconductor wafer fab with uncertain demand and capacity, *European Journal of Operational Research*, Vol. 198, No. 3, 899-908. doi:10.1016/j.ejor.2008.09.029

[3] Feng, Y.; Ryan, S. M. (2013). Scenario construction and reduction applied to stochastic power generation expansion planning, *Computers & Operations Research*, Vol. 40, No. 1, 9-23, doi:10.1016/j.cor.2012.05.005

[4] Chen, Z.-L.; Li, S.; Tirupati, D. (2002). A scenario-based stochastic programming approach for technology and capacity planning, *Computers & Operations Research*, Vol. 29, No. 7, 781-806, doi:10.1016/S0305-0548(00)00076-9

[5] Kara, S. S.; Onut, S. (2010). A two-stage stochastic and robust programming approach to strategic planning of a reverse supply network: the case of paper recycling, *Expert Systems with Applications*, Vol. 37, No. 9, 6129-6137, doi:10.1016/j.eswa.2010.02.116

[6] Zanjani, M. K.; Nourelfath, M.; Ait-Kadi, D. (2010). A multi-stage stochastic programming approach for production planning with uncertainty in the quality of raw materials and demand, *International Journal of Production Research*, Vol. 48, No. 16, 4701-4723, doi:10.1080/00207540903055727

[7] Vigerske, S.; Nowak, I. (2007). Adaptive discretization of convex multistage stochastic programs, *Mathematical Methods of Operations Research*, Vol. 65, No. 2, 361-383, doi:10.1007/s00186-006-0124-y

[8] Casey, M. S.; Sen, S. (2005). The scenario generation algorithm for multistage stochastic linear programming, *Mathematics of Operations Research*, Vol. 30, No. 3, 615-631, doi:10.1287/moor.1050.0146

[9] Escudero, L. F.; Garín, M. A.; Merino, M.; Pérez, G. (2010). On BFC-MSMIP strategies for scenario cluster partitioning, and twin node family branching selection and bounding for multistage stochastic mixed integer programming, *Computers & Operations Research*, Vol. 37, No. 4, 738-753, doi:10.1016/j.cor.2009.06.023

[10] Wu, C.-H.; Chuang, Y.-T. (2012). An efficient algorithm for stochastic capacity portfolio planning problems, *Journal of Intelligent Manufacturing*, Vol. 23, No. 6, 2161-2170, doi:10.1007/s10845-011-0562-0

[11] Jiao, J. R.; Zhang, Y.; Wang, Y. (2007). A heuristic genetic algorithm for product portfolio planning, *Computers & Operations Research*, Vol. 34, No. 6, 1777-1799, doi:10.1016/j.cor.2005.05.033

[12] Chen, Y.-Y.; Lin, J. T. (2009). A modified particle swarm optimization for production planning problems in the TFT Array process, *Expert Systems with Applications*, Vol. 36, No. 10, 12264-12271, doi:10.1016/j.eswa.2009.04.072

[13] Ramezanian, R.; Rahmani, D.; Barzinpour, F. (2012). An aggregate production planning model for two phase production systems: solving with genetic algorithm and tabu search, *Expert Systems with Applications*, Vol. 39, No. 1, 1256-1263. doi:10.1016/j.eswa.2011.07.134

[14] Ferreira, K. D.; Wu, D. D. (2011). An integrated product planning model for pricing and bundle selection using Markov decision processes and data envelope analysis, *International Journal of Production Economics*, Vol. 134, No. 1, 95-107, doi:10.1016/j.ijpe.2011.01.003

[15] Chen, T.-L.; Lu, H.-C. (2012). Stochastic multi-site capacity planning of TFT-LCD manufacturing using expected shadow-price based decomposition, *Applied Mathematical Modelling*, Vol. 36, No. 12, 5901-5919, doi:10.1016/j.apm.2012.01.037

[16] Kébé, S.; Sbihi, N.; Penz, B. (2012). A Lagrangean heuristic for a two-echelon storage capacitated lot-sizing problem, *Journal of Intelligent Manufacturing*, Vol. 23, No. 6, 2477-2483, doi:10.1007/s10845-011-0514-8

[17] Bao, J.; Chen, Y; Yu, J.-s. (2010). A regeneratable dynamic differential evolution algorithm for neural networks with integer weights, *Journal of Zhejiang University – SCIENCE C (Computers & Electronics)*, Vol. 11, No. 12, 939-947, doi:10.1631/jzus.C1000137

[18] Cui, C.-g.; Li, Y.-j.; Wu, T.-j. (2010). A relative feasibility degree based approach for constrained optimization problems, *Journal of Zhejiang University – SCIENCE C (Computers & Electronics)*, Vol. 11, No. 4, 249-260, doi:10.1631/jzus.C0910072

[19] Qu, B.-y.; Suganthan, P.-N. (2010). Multi-objective differential evolution with diversity enhancement, *Journal of Zhejiang University – SCIENCE C (Computers & Electronics)*, Vol. 11, No. 7, 538-543,doi:10.1631/jzus.C0910481

[20] Yang, Z.; Li, X.; Bowers, C. P.; Schnier, T.; Tang, K.; Yao, X. (2012). An efficient evolutionary approach to parameter identification in a building thermal model, *IEEE Transactions On Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 42, No. 6, 957-969, doi:10.1109/TSMCC.2011.2174983

[21] Price, K. V.; Storn, R. M.; Lampinen, J. A. (2005). *Differential Evolution: A Practical Approach to Global Optimization*, Springer-Verlag, Berlin

[22] Das, S.; Konar, A.; Chakraborty, U. K. (2005). Two improved differential evolution schemes for faster global search, *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO 2005)*, Washington, 991-998

[23] Abbass, H. A.; Sarker, R.; Newton, C. (2001). PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems, *Proceedings of the 2001 Congress on Evolutionary Computation*, Vol. 2, 971-978, doi:10.1109/CEC.2001.934295

[24] Yang, Z.; Yao, X.; He, J. (2008). Making a difference to differential evolution, Siarry, P.; Michalewicz, Z. (Eds.), *Advances in Metaheuristics for Hard Optimization*, Spring-Verlag, Berlin, 397-414, doi:10.1007/978-3-540-72960-0_19

[25] Yang, Z.; Tang, K.; Yao, X. (2008) Self-adaptive differential evolution with neighborhood search, *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, Hong Kong, 1110-1116

[26] Haupt, R. L.; Haupt, S. E. (2004). *Practical genetic algorithms*, John Wiley & Sons, Hoboken

[27] Eberhart, R.; Kennedy, J. (1995). A new optimizer using particle swarm theory, *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, 39-43

[28] Yang, Z.; Tang, K.; Yao, X. (2008). Large scale evolutionary optimization using cooperative coevolution, *Information Sciences*, Vol. 178, No. 15, 2985-2999, doi:10.1016/j.ins.2008.02.017