Enhanced Differential Evolution Based on Adaptive Mutation and Wrapper Local Search Strategies for Global Optimization Problems

Chun-Liang Lu*[1], Shih-Yuan Chiu[2], Chih-Hsu Hsu[3] and Shi-Jim Yen[4]

[1,3] Department of Applied Information and Multimedia
Ching Kuo Institute of Management and Health
Keelung County
Taiwan, R.O.C.
*leucl@ems.cku.edu.tw
[2,4] Department of Computer Science and Information
Engineering National Dong Hwa University, Hualien County
Taiwan, R.O.C.

ABSTRACT
Differential evolution (DE) is a simple, powerful optimization algorithm, which has been widely used in many areas. However, the choices of the best mutation and search strategies are difficult for the specific issues. To alleviate these drawbacks and enhance the performance of DE, in this paper, the hybrid framework based on the adaptive mutation and Wrapper Local Search (WLS) schemes, is proposed to improve searching ability to efficiently guide the evolution of the population toward the global optimum. Furthermore, the effective particle encoding representation named Particle Segment Operation-Machine Assignment (PSOMA) that we previously published is applied to always produce feasible candidate solutions for solving the Flexible Job-shop Scheduling Problem (FJSP). Experiments were conducted on comprehensive set of complex benchmarks including the unimodal, multimodal and hybrid composition function, to validate performance of the proposed method and to compare with other state-of-the art DE variants such as jDE, JADE, MDE_pBX etc. Meanwhile, the hybrid DE model incorporating PSOMA is used to solve different representative instances based on practical data for multi-objective FJSP verifications. Simulation results indicate that the proposed method performs better for the majority of the single-objective scalable benchmark functions in terms of the solution accuracy and convergence rate. In addition, the wide range of Pareto-optimal solutions and more Gantt chart decision-makings can be provided for the multi-objective FJSP combinatorial optimizations.

Keywords: Differential Evolution, Wrapper Local Search, Particle Segment Operation-Machine Assignment, Flexible Job-shop Scheduling Problem.

1. Introduction

Optimization algorithms have become a useful technique in all-major disciplines and engineering applications [1–2]. Many practical engineering design or decision making problems involve single-objective or multi-objective optimization. In single-objective optimization, the goal is to find the best design solution as to the minimum or maximum value of the objective function. On the contrary, the multi-objective optimization gives rise to Pareto-optimal solutions [3] because of the interaction among different conflicting objectives. Differential Evolution (DE) algorithm is a population-based and stochastic optimizer first developed by Storn and Price [4]. With the advantages of simplicity, less parameter and robustness, the DE algorithm has been given increasing attention and widely used in many fields, such as data mining [5], structural optimization [6], biogeography [7], and so on [8–9]. DE is considered the most recent studies for solving constrained optimization problems, multi-objective global optimizations, and other complex real-world applications. More details on the state-of-the-art investigation within DE can be found in two surveys [10–11] and the references therein.

For the classical DE, the setting of three control parameters: population size Np, the crossover rate Cr and the scale factor F, is very sensitive to the parameter setting and the choice of the best parameters is always problem-dependent [12]. In addition, for a given specific problem, it may be better to adopt different parameter settings during different generation stages of the evolution than use a single mutation strategy with fixed parameter settings [11]. In 2006, Brest et al. [12] presented a self-adaptive method for DE fixes the population size during the evolution process while adapting the control parameters F_i and CR_i associated with each individual. The jDE reproduces new F_i and CR_i values according to uniform distributions on [0.1, 1] and [0, 1], respectively. And experimental results demonstrated that jDE performs remarkably better than the classic DE/rand/1/bin. Later, Zhang et al. presented "DE/current-to-*p*best" with optional archive and controls *F* and *CR* in an adaptive manner named JADE [13], to track the historical record of success status for mutation factors and crossover probabilities with adaptive parameters in generations, and the external archive to store recently inferior solutions and their difference from current population provides promising directions toward the optimum. In 2012, Islam et al. proposed the MDE_*p*BX [14], which adds a variation to the classical "DE/current-to-best/1" mutation scheme by perturbing the current target vector with the best solution in a group of randomly selected individuals. The crossover operation is performed between the current donor vector and any other individual from *p* top-ranked individuals in the present generation. Simulation results demonstrate that MDE_*p*BX enhances the ability of the basic DE for finding solutions in search space and helps alleviating the tendency of premature convergence or stagnation.

In the recent studies [12–15], various mutation and controll parameters setting strategies have been presented for DE algorithm. Although a number of works can advance the search ability of DE, there is still much room for improving the performance of DE. Motivated by these results, the modified mutation scheme based on MDE_*p*BX, in which the mutation operator can be adjusted dynamically on the solution searching status, is presented to bring several individuals appropriately to find new possible solutions. Meanwhile, a wrapper local search (WLS) strategy via trying to increase or decrease current moving vector by the Cauchy distribution is proposed to improve the local search ability and to balance exploration and exploitation in the search space. Moreover, the proposed hybrid DE framework incorporated with PSOMA method that we previously published to produce feasible solutions for the multi-objective Flexible Job-shop Scheduling Problem (FJSP), is designed for finding optimal solutions of multi-objective FJSP. Competitive experimental results are observed with respect to 15 CEC 2005 benchmark functions for single-objective optimizations, and the three benchmark instances based on practical data were employed as multi-objective FJSP verifications.

The remainder of this paper is organized as follows.

Section 2 describes the typical MDE_pBX, FJSP and external repository. The PSOMA scheme, adaptive mutation method, WLS and the hybrid DE model are presented in Section 3. Experiment and comparison results are provided in Section 4. Conclusions remarks are made in Section 5.

2. Related Works

2.1 Classical Differential Evolution Algorithm

DE algorithm is one of the population-based global optimization algorithms, has two stages including initialization and evolution. After randomly initializs, evolution process evolves from one generation to the next through mutation, crossover and selection operations until the termination criteria are reached. The core of DE algorithm is the mutation operation, which uses a weighted, random vector in each step, to replace the target vector with the better trial vector in the next generation. The main steps of the classical DE are summarized as follows.

2.1.1 Initialization

The DE begins by creating an initial population of target vectors consisting of parameter vectors are denoted by $\vec{X}_{i,G} = [x_{i,G}^1, x_{i,G}^2, \cdots, x_{i,G}^D]^T$, $i = (1, 2, \cdots, N_P)$, where *i* is the index for individuals, *G* indicates the current generation, N_p is the population size, *D* is the dimension of the parameters, and $x_{i,G}^j$ denotes the *j*-th component of the *i*-th individual at the *G*-th generation. The initial individuals are randomly determined within a predefined search space considering the lower and upper bounds of each parameter as follows.

$$x_{i,G}^j = x_{\min}^j + rand(0,1) \times (x_{\max}^j - x_{\min}^j),\ j = (1, 2, \cdots, D) \quad (1)$$

where $x_{\min}^j$ and $x_{\max}^j$ denote the lower and upper bounds, respectively, and $rand(\bullet)$ is a uniformly distributed random number between 0 and 1.

2.1.2 Mutation

For the "DE/rand/1/bin" classical mutation strategy, three different vectors consisting of a base vector ($\vec{X}_{r_1^i,G}$) and two difference vectors ($\vec{X}_{r_2^i,G}$ and $\vec{X}_{r_3^i,G}$) are randomly chosen from the populations. The scale factor F_i is a constant and the effective range is usually taken from the range between 0.5 and 1 as pointed out in [4]. Mutation operation is then carried out by perturbing the base vector via a difference vector scaled by a scalar factor F_i. The donor vector $\vec{V}_{i,G} = [v_{i,G}^1, v_{i,G}^2, \cdots, v_{i,G}^D]^T$, $i = (1, 2, \cdots, N_P)$ is expressed as Eq. 2.

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F_i(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \qquad (2)$$

where r_1 , r_2 , r_3 are random and mutually different integers, and they are also different with the vector index i .

2.1.3 Crossover

To increase the diversity of the solutions, the trial vector $\vec{U}_{i,G} = [u_{i,G}^1, u_{i,G}^2, \cdots, u_{i,G}^D]^T$, $i = (1, 2, \cdots, N_P)$ is created by means of crossover operation and is realized between each pair of target vector $\vec{X}_{i,G}$ and its corresponding donor vector $\vec{V}_{i,G}$. In a trial vector, elements are inherited from a donor vector and a target vector. The scheme can be simply formulated for the binomial uniform crossover that is widely used in the literature as shown below.

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & if\,(rand(0,1) \le C_r \;or\; j = j_{rand}, \\ x_{i,G}^j & otherwise, \end{cases} \quad j = 1, 2, ..., D \qquad (3)$$

where $u_{i,G}^j$, $v_{i,G}^j$ and $x_{i,G}^j$ are trial, donor and target vectors from the *i*-th vector, *j*-th dimension at *G*-th generation. C_r is a user-defined probability in the range [0, 1]. j_{rand} is a randomly chosen integer in the range [1, D], which ensures that the trial vector does not duplicate the target vector.

2.1.4 Selection

The selection operation is achieved from the target and trial vectors by comparing their fitness values through the objective function $f(\bullet)$ to select the better individual. In case of minimization problems, the trial vector and the target vector competes in their fitness and the winner has the chance to survive to the next generation.

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & if\,(f(\vec{U}_{i,G}) \le f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & otherwise. \end{cases} \qquad (4)$$

In the current population, target vector is updated when the newly generated trial vector gets better fitness value than its target vector; otherwise, the target vector is retained in the population.

DE algorithm works through a simple cycle of the stages, and the pseudo-code of the classical DE is given below as Algorithm 1.

Algorithm 1. The classical DE algorithm.

```
Basic setting of parameters for DE.
**Initialization** Generate the initial population.
Evaluate the fitness for each individual.
**while** Termination condition is not satisfied **do**
    **Mutation**.
    **Crossover**.
    **Selection**.
    Boundary constraints for each $x_{i,G}^j \in \vec{X}_{i,G}$ .
**end while**
**Output** Global optimum solution $\vec{X}_{Best}$ .
```

2.2 Typical MDE_pBX Approach

Islam et al. developed an adaptive DE algorithm based on novel mutation and crossover strategies called MDE_*p*BX [14] for global optimizations. They presented a *p*-best crossover operation to improve the accuracy in search space by a large extent. Moreover, the Cauchy distribution and Gaussian distribution are also incorporated into the generation for scale factor *F* and crossover rate C_r. The new mutation operator named "DE/current-to-gr_best/1" scheme is as follows.

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F_i(\vec{X}_{gr\_best,G} - \vec{X}_{i,G} + \vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \quad (5)$$

$\vec{X}_{gr\_best,G}$ is the best solution from 15% of individuals selected randomly in current generation. Then, normal binomial crossover is performed between the donor vector and the randomly selected *p*-best vector to generate the trial vector at the same index. Parameter *p* is linearly reduced by generations in the following formula.

$$p = ceil(\frac{N_p}{2} \times (1 - \frac{G-1}{G_{\max}})) \quad (6)$$

G is the current generation number, $G_{\max}$ is the maximum number of generation set up, and $ceil(y)$ is the "ceiling" function returning the lowest integer greater than its argument y. Finally, the parameter adaptation schemes in MDE_*p*BX is independently generated as follows.

$$F_i = Cauchy(F_m, 0.1) \quad (7)$$

Eq. 7 is a random number sampled from a Cauchy distribution with location parameter F_m formulated as Eq. 8, and the scale parameter is 0.1. Location parameter F_m of Cauchy distribution is initialized to be 0.5 and updated at the end of each generation.

$$F_m = w_F \times F_m + (1 - w_F) \times mean_{POW}(S_F) \quad (8)$$

$$w_F = 0.8 + 0.2 \times rand(0,1) \quad (9)$$

$$mean_{POW}(S_F) = \left| \sum_{x \in S_F} (\frac{x^{1.5}}{|S_F|})^{\frac{1}{1.5}} \right| \quad (10)$$

A weight factor w_F is a positive constant number between 0.8 to 1. The $mean_{POW}(\bullet)$ represents a power mean given by Eq. 10. The adaptation of crossover probability Cr_i is similar with the scale factor F_i, and the detailed description as follows.

$$Cr_i = Gaussian(Cr_m, 0.1) \quad (11)$$

$$Cr_m = w_{Cr} \times Cr_m + (1 - w_{Cr}) \times mean_{POW}(S_{Cr}) \quad (12)$$

$$w_{Cr} = 0.9 + 0.1 \times rand(0,1) \quad (13)$$

$$mean_{POW}(S_{Cr}) = \left| \sum_{x \in S_{Cr}} (\frac{x^{1.5}}{|S_{Cr}|})^{\frac{1}{1.5}} \right| \quad (14)$$

where w_{Cr} is set between 0.9 to 1, Cr_m is initialized to be 0.6, and the Gaussian distribution as shown in Eq. 11 substitutes for the Cauchy one in Eq. 7.

2.3 FJSP Problem Definition

FJSP [16] is a generalization of the classical JSP, in which operations are allowed to be processed by any machine from agiven set of available machines. It is quite difficult to achieve an optimal solution due to the high computational complexity and well-known a NP-hard problem. Xia et al. [17] proposed a method which hybridized PSO and simulated annealing (SA). It involved a variable inertia weight w and adopted a weighted concept to transform triple objectives into single objective problems. Ho et al. [18] later proposed a method to estimate bounds of different types of schematic may exist corresponding to the Pareto-optimal fitness value, but Ho's method did not deal with the diversity under the same Pareto-optimal solutions.

The three minimization objectives addressed in [19] including the total workload of all machines, the workload of critical machine and the completion time of critical job are considered simultaneously. The problem is to organize the execution of *n* jobs $J_i (i = 1, 2, ..., n)$ on *m* machines $M_k (k = 1, 2, ..., m)$, where each job J_i needs O_i operations on the order of restraint and each working procedure of job can be processed by multiple processes of *M* machines. The total number of operations in all jobs is O_t, $O_t = \sum_{i=1}^{n} O_i$

M_{ij} means the collection of the useable machines about *j*-th operation of *i*-th job, $\mathrm{M}_{ij} \in \{1, 2, ..., m\}$, $O_{i,j,k}$ means the *j*-th operation of the *i*-th job can use *k*-th machine, $p_{i,j,k}$ means the required time of *j*-th operation of *i*-th job on the *k*-th machine, $1 \le i \le n, 1 \le j \le o_i, 1 \le k \le m$. The task is to find a set of solutions and three criteria of FJSP are considered and described as follows:

(a) The total workload of all machines

$$F_1 = \sum p_{i,j,k} \tag{15}$$

where the element $p_{i,j,k}$ denotes the processing time of *j*-th operation of *i*-th job in *k*-th machine.

(b) The workload of the critical machine

$$F_2 = \max\{W_1, W_2, ..., W_m\} \tag{16}$$

where W_k denotes the workload of the *k*-th machine M_k (the summation of $p_{i,j,k}$ on M_k).

(c) The completion time of the critical job

$$F_3 = \max\{CT_1, CT_2, ..., CT_n\} \tag{17}$$

where CT_i is the completion time of the *i*-th job J_i .

2.4 External Repository

The main difference between Single-Objective Problem (SOP) and Multi-Objective Problems (MOP) is that MOP contains more than one objective that needs to be accomplished simultaneously. In [20], the authors presented an external repository (or archive) to keep the historical record of the non-dominated solutions found along the search process. In Fig.1, the basic external repository is also adopted in our proposed hybrid DE algorithm for solving the multi-objective FJSP. The reservation process of non-dominated solutions is described as follows.

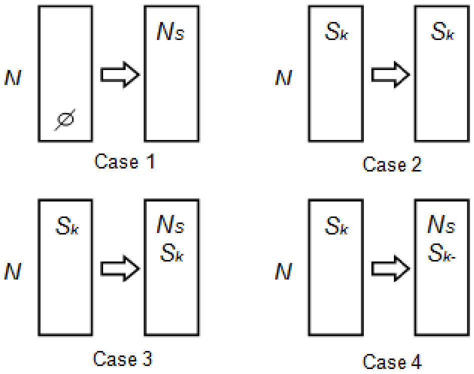


Figure 1. Reservation process of external repository.

Case 1. If an external repository is empty, any non-dominated solution found currently is inserted.

Case 2. If a solution is dominated by any individual in external repository, the solution will be discarded.

Case 3. If a solution is non-dominated by all individuals or equaled by any individual in external repository, the solution will be stored.

Case 4. If a solution dominates at least one solution in external repository, those dominated solutions are removed from the external repository, and this solution will be stored.

All new solutions are compared with solutions in the external repository, and one of four cases is corresponded each time. *N_S* means one of new solutions. *k* in *S_k* means there exists *k* solutions in external repository. *S_k*- means a modified *S_k* after some dominated solutions are removed by *N_S*.

3. The Proposed Methods

As above mentioned to the adaptive DE variants, the parameters updated iteratively according to their successful experience in the generations, such as JADE [13], MDE_pBX [14]. In this work, the proposed DE model based on self-adaptive mutation (SAM) and wrapper local search (WLS) to enhance the performance of DE. In addition, the effective PSOMA [21] that we previously published is successfully merged into DE model for solving the multi-objective FJSP.

3.1 Self-adaptive Mutation (SAM) Approach

The original DE mutation is "DE/rand/1/bin" in [4]. Islam *et al.* proposed the mutation of MDE_*p*BX, although the effect of the method performs better results than many other algorithms, this diversity of solution searching scheme is getting premature convergence at local optima within a smaller number of generations. Taking into consideration of these facts and to overcome the limitations of this feature, the extension of modified mutation searching strategy in [14], which is abbreviated as SAM approach, can be shown below.

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F_i(\vec{X}_{gr\_better,G} - \vec{X}_{i,G} + \vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \tag{18}$$

$$w = w_{\max} - \frac{i_{now}}{i_{\max}}(w_{\max} - w_{\min}) \qquad (19)$$

In Eq. 18, $\vec{X}_{i,G}$ is known as the target vector, $\vec{V}_{i,G}$ is known as the donor vector, the scaling factor F_i is a positive scaling parameter for difference vectors. The $\vec{X}_{r_1^i,G}$ and $\vec{X}_{r_2^i,G}$ are two distinct vectors picked up randomly from the current population, and none of them is equal to $\vec{X}_{gr\_better,G}$ or the target vector. The $\vec{X}_{gr\_better,G}$ is dynamically chosen from top *w*% individuals of each population. The linearly decreasing inertia weight is set as Eq. 19, where i_{now} is the current iteration, $i_{\max}$ is the user-defined maximum iteration, and the pre-defined lower and upper bounds of *w* described as $w_{\min} \le w \le w_{\max}$.

This new proposed self-adaptive mutation operator is a variant of the MDE_pBX. It dynamically uses the best of a group (whose size is w% of the linearly decreasing population size) of randomly selected solutions, from current generation to perturb the parent vector, and unlike in [14] that the $\vec{X}_{better,G}$ always picks the best vector from the entire population fixed-ratio to perturb the target vector. This new simple self-adaptive mutation approach can drive the population to the better direction instead of convergence to the best individual in the iteration and help DE not to fall into local optimum quickly in smaller generation.

3.2 Wrapper Local Search (WLS) Strategy

According to related works, it can be found that classical DE can perform well performance on widely search for exploring unsearched solution space but weakly on searching depth. In 2010 [22] we published the efficient wrapper-based hybrid model for solving the biomedical problem and is capable of producing high prediction accuracy and fewer number of features selection simultaneously. In this study, the wrapper-based feature selection framework is properly adopted to enhance the local search performance for DE, named Wrapper Local Search (WLS) strategy, is involved to adjust the scale of moving vector via trying to increase or decrease current moving vector by the Cauchy distribution. In Figure 2, the proposed WLS method utilized these random wrapper-based selected dimensions (“1” indicates selected, “0” denotes excluded) to identify a suitable balance tradeoff of DE search and local search. It can save much time than one by one the single dimension search [23], and will fine tune the searching direction for finding the global best solution.

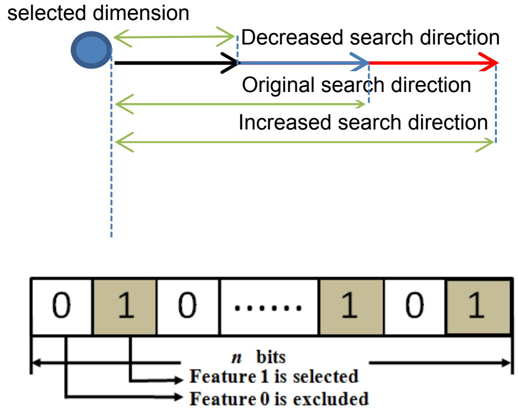


Figure 2. The WLS for one of selected dimensions.

The pseudo-code of the proposed DE algorithm is described as Algorithm 2.

Algorithm 2. The proposed DE model combined with SAM and WLS strategies.

```
Basic setting of parameters for the DE model.
Initialization Generate the initial population.
Evaluate the fitness for each individual.
while Termination condition is not satisfied do
    for $i$=1 to $N_p$ do
        for $j$=1 to $D$ do
            Mutation with SAM approach.
            Crossover.
        end for
    end for
    for $i$=1 to $N_p$ do
        Evaluate the offspring.
        Selection.
    end for
    Apply WLS on the better selected individuals.
end while
Output Global optimum solution.
```

3.3 PSOMA Scheme

The effective particle encoding representation that we previously published and detailed in [21], each dimension contains three components: integer part (machine selection), decimal part (priority order) and real-value number (operation number). In Fig.3 is the structure of PSOMA representation on each dimension. This encoding representation is flexible enough for solving the FJSP to satisfy the precedence constraints and operations in each job by using the structure of the proposed DE model with real-value number.

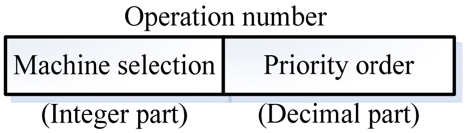


Figure 3. Three components of PSOMA structure.

Jobs / Machines	$O_{1.1}$	$O_{1.2}$	$O_{1.3}$	$O_{2.1}$	$O_{2.2}$	$O_{3.1}$
M_1	3	4	2	4	5	6
M_2	4	5	2	5	6	8
M_3	2	3	3	5	5	6

Table 1. The example of operation schedules for FJSP.

An example of FJSP (3 jobs, 3 machines, and 6 operations) is considered and illustrated in Table 1. Three Jobs J_1, J_2 and J_3 need to be processed by at least one of three machines M_1, M_2 and M_3. Each job contains several operations such as Job 1 is split into $O_{1.1}$, $O_{1.2}$, $O_{1.3}$; Job 2 is split into $O_{2.1}$, $O_{2.2}$ etc. The executing times for each machine to execute each job is given, such as $O_{1.1}$ assigned to M_1 is 3 units; $O_{1.2}$ assigned to M_2 is 4 units, etc.

$O_{1.1}$	$O_{1.2}$	$O_{1.3}$	$O_{2.1}$	$O_{2.2}$	$O_{3.1}$
3.81	1.26	2.53	1.92	2.68	3.37

Figure 4. A possible candidate encoding of the individual.

To explain the PSOMA scheme, one possible candidate encoding representation of the individual shown as Figure 4, to give the detailed description of the decoding procedure for FJSP as follows.

Step 1. $O_{2.1}$ is assigned to $M_{1.}$
Operation set O_1= { $O_{1.1}$, $O_{2.1}$, $O_{3.1}$ }.
Assignment set C_1= { $O_{1.1}$- M_3, $O_{2.1}$- M_1, $O_{3.1}$- M_3 }.
Max($O_{1.1}$, $O_{2.1}$, $O_{3.1}$)=Max(0.81, 0.92, 0.37)=0.92.
Then the { $O_{2.1}$- M_1 } is assigned from $C_{1.}$

Step 2. $O_{1.1}$ is assigned to $M_{3.}$
Operation set O_2= { $O_{1.1}$, $O_{2.2}$, $O_{3.1}$ }.
Assignment set C_2= { $O_{1.1}$- M_3, $O_{2.2}$- M_2, $O_{3.1}$- M_3 }.
Max($O_{1.1}$, $O_{2.2}$, $O_{3.1}$)=Max(0.81, 0.68, 0.37)=0.81.
Then the { $O_{1.1}$- M_3 } is assigned from $C_{2.}$

Step 3. $O_{2.2}$ is assigned to $M_{2.}$
Operation set O_3= { $O_{1.2}$, $O_{2.2}$, $O_{3.1}$ }.
Assignment set C_3= { $O_{1.2}$- M_1, $O_{2.2}$- M_2, $O_{3.1}$- M_3 }.
Max($O_{1.2}$, $O_{2.2}$, $O_{3.1}$)=Max(0.26, 0.68, 0.37)=0.68.
Then the { $O_{2.2}$- M_2 } is assigned from $C_{3.}$

Step 4. $O_{3.1}$ is assigned to $M_{3.}$
Operation set O_4= { $O_{1.2}$, $O_{3.1}$ }.
Assignment set C_4= { $O_{1.2}$- M_1, $O_{3.1}$- M_3 }.
Max($O_{1.2}$, $O_{3.1}$)=Max(0.26, 0.37)=0.37.
Then the { $O_{3.1}$- M_3 } is assigned from $C_{4.}$

Step 5. $O_{1.2}$ is assigned to $M_{1.}$
Operation set O_5= { $O_{1.2}$ }.
Assignment set C_5= { $O_{1.2}$- M_1 }.
Max($O_{1.2}$)=Max(0.26)=0.26.
Then the { $O_{1.2}$- M_1 } is assigned from $C_{5.}$

Step 6. $O_{1.3}$ is assigned to $M_{2.}$
Operation set O_6= { $O_{1.3}$ }.
Assignment set C_6= { $O_{1.3}$- M_2 }.
Max($O_{1.3}$)=Max(0.53)=0.53.
Then the { $O_{1.3}$- M_2 } is assigned from $C_{6.}$

As shown in Figure 5, the operations of all jobs are assigned to each machine done and one of the feasible Gantt chart solution is provided. The precedence constraints are {$O_{1.1}$ before $O_{1.2}$ before $O_{1.3}$} from job 1, {$O_{2.1}$ before $O_{2.2}$} from job 2, and {$O_{3.1}$} from job 3. The operation order should be preserved by the different machine selection. It is obvious that each scheduling candidate solution generated by any encoding representation based on PSOMA should be feasible. The pseudo-code of the proposed hybrid DE model with PSOMA scheme for solving FJSP is shown as Algorithm 3.

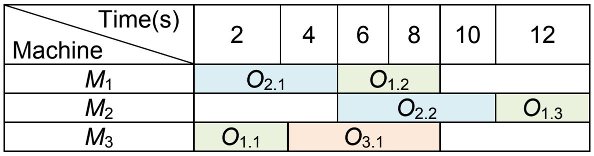


Figure 5. Gantt chart of operation-machine assignments.

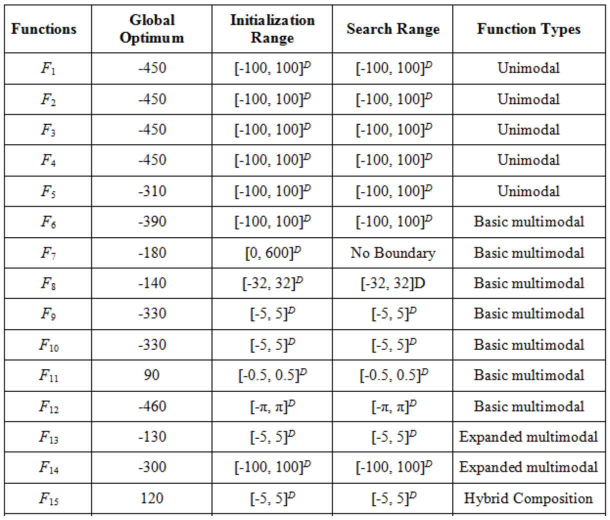
Algorithm 3. The proposed hybrid DE model with PSOMA scheme for FJSP.

```
Basic setting of parameters for the DE model.
Initialization Generate the initial population via.
Evaluate the fitness for each individual.
while Termination condition is not satisfied do
    for $i$=1 to $N_p$ do
        for $j$=1 to $D$ do
            Mutation with SAM approach.
            Crossover.
        end for
    end for
    Decoding by PSOMA for each individual.
    for $i$=1 to $N_p$ do
        Evaluate the offspring.
        Selection.
    end for
    Apply WLS on the better selected individuals.
    Compare Gantt charts with solutions diversity.
    Store multi-objective non-dominated solutions
      and Gantt charts to External Repository.
end while
Output Global optimum solution(s).
```

4. Experimental Results

4.1 Single-objective numerical benchmarks

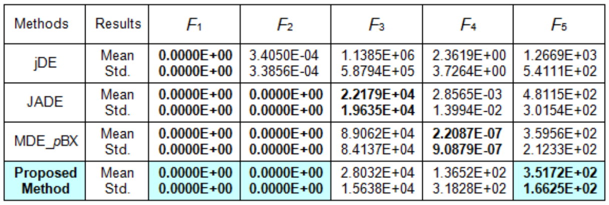
In order to verify the performance of our approach, a selected set of standard test functions from the special session on real-parameter optimization of the IEEE Congress on Evolutionary Computations, CEC 2005 [24] were adopted for testing to related works. The global optimum (equal to function bias), search range, initialization range and function types of each test function are illustrated in Table 2. These functions 1 to 5 are unimodal, functions 6 to 12 are basic multimodal, functions 13 to 14 are expanded multimodal, and the hybrid composition is function 15. In the experiments, 15 numerical test functions with 30 dimensions are conducted for comparing the proposed method with other five related works as jDE, JADE and MDE_*p*BX reported in [14]. The initial population size is set as 100. The fitness evaluation (FEs) is 500,000. All the DE variants algorithms are implemented on Matlab 2013a. The mean values and standard deviation are calculated. The mean and standard deviation of error values are optimized and recorded by 25 independent runs.



Functions	Global Optimum	Initialization Range	Search Range	Function Types
F_1	-450	[-100, 100]^D	[-100, 100]^D	Unimodal
F_2	-450	[-100, 100]^D	[-100, 100]^D	Unimodal
F_3	-450	[-100, 100]^D	[-100, 100]^D	Unimodal
F_4	-450	[-100, 100]^D	[-100, 100]^D	Unimodal
F_5	-310	[-100, 100]^D	[-100, 100]^D	Unimodal
F_6	-390	[-100, 100]^D	[-100, 100]^D	Basic multimodal
F_7	-180	[0, 600]^D	No Boundary	Basic multimodal
F_8	-140	[-32, 32]^D	[-32, 32]D	Basic multimodal
F_9	-330	[-5, 5]^D	[-5, 5]^D	Basic multimodal
F_{10}	-330	[-5, 5]^D	[-5, 5]^D	Basic multimodal
F_{11}	90	[-0.5, 0.5]^D	[-0.5, 0.5]^D	Basic multimodal
F_{12}	-460	[-π, π]^D	[-π, π]^D	Basic multimodal
F_{13}	-130	[-5, 5]^D	[-5, 5]^D	Expanded multimodal
F_{14}	-300	[-100, 100]^D	[-100, 100]^D	Expanded multimodal
F_{15}	120	[-5, 5]^D	[-5, 5]^D	Hybrid Composition

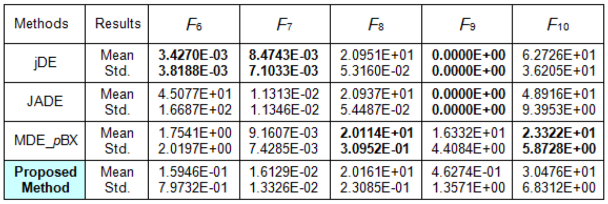
Table 2. The numerical benchmarks on CEC 2005 [23].

The experiment results of 15 test functions with 30 dimensions numerical functions are listed in Table 3 to Table 5. The best results among the four DE approaches are shown in bold. The convergence characteristics of the proposed method and related works are shown in Figure 6-1 to Figure 6-15.



Methods	Results	F_1	F_2	F_3	F_4	F_5
jDE	Mean Std.	**0.0000E+00** **0.0000E+00**	3.4050E-04 3.3856E-04	1.1385E+06 5.8794E+05	2.3619E+00 3.7264E+00	1.2669E+03 5.4111E+02
JADE	Mean Std.	**0.0000E+00** **0.0000E+00**	**0.0000E+00** **0.0000E+00**	**2.2179E+04** **1.9635E+04**	2.8565E-03 1.3994E-02	4.8115E+02 3.0154E+02
MDE_*p*BX	Mean Std.	**0.0000E+00** **0.0000E+00**	**0.0000E+00** **0.0000E+00**	8.9062E+04 8.4137E+04	**2.2087E-07** **9.0879E-07**	3.5956E+02 2.1233E+02
Proposed Method	Mean Std.	**0.0000E+00** **0.0000E+00**	**0.0000E+00** **0.0000E+00**	2.8032E+04 1.5638E+04	1.3652E+02 3.1828E+02	**3.5172E+02** **1.6625E+02**

Table 3. Experimental results of F_1 to F_5 test functions.



Methods	Results	F_6	F_7	F_8	F_9	F_{10}
jDE	Mean Std.	**3.4270E-03** **3.8188E-03**	**8.4743E-03** **7.1033E-03**	2.0951E+01 5.3160E-02	**0.0000E+00** **0.0000E+00**	6.2726E+01 3.6205E+01
JADE	Mean Std.	4.5077E+01 1.6687E+02	1.1313E-02 1.1346E-02	2.0937E+01 5.4487E-02	**0.0000E+00** **0.0000E+00**	4.8916E+01 9.3953E+00
MDE_*p*BX	Mean Std.	1.7541E+00 2.0197E+00	9.1607E-03 7.4285E-03	**2.0114E+01** **3.0952E-01**	1.6332E+01 4.4084E+00	**2.3322E+01** **5.8728E+00**
Proposed Method	Mean Std.	1.5946E-01 7.9732E-01	1.6129E-02 1.3326E-02	2.0161E+01 2.3085E-01	4.6274E-01 1.3571E+00	3.0476E+01 6.8312E+00

Table 4. Experimental results of F_6 to F_{10} test functions.

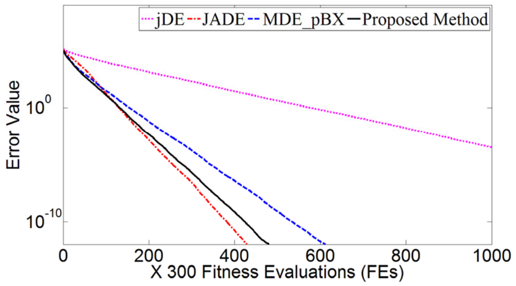


Figure 6-1. Convergence curves of F_1 function.

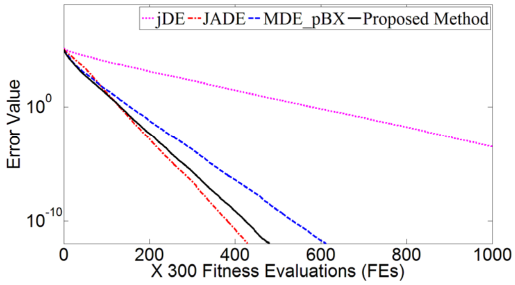


Figure 6-2. Convergence curves of F_2 function.

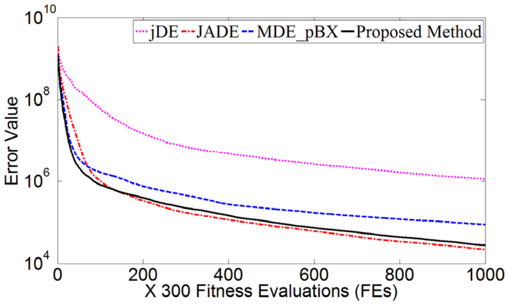


Figure 6-3. Convergence curves of F_3 function.

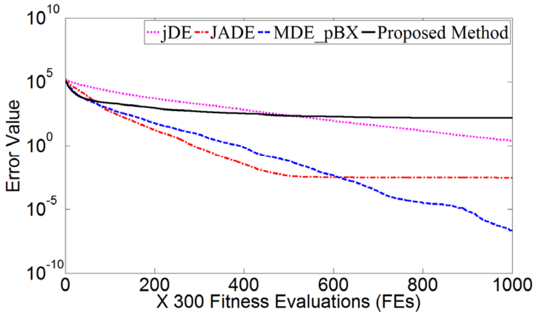


Figure 6-4. Convergence curves of F_4 function.

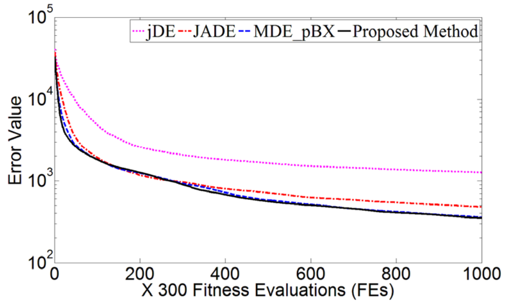


Figure 6-5. Convergence curves of F_5 function.

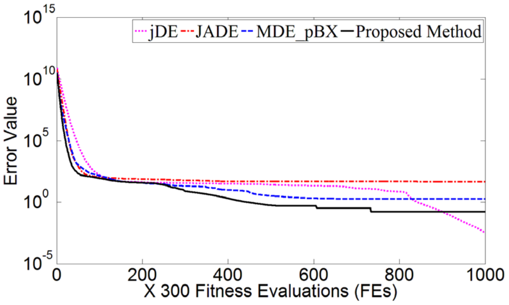


Figure 6-6. Convergence curves of F_6 function.

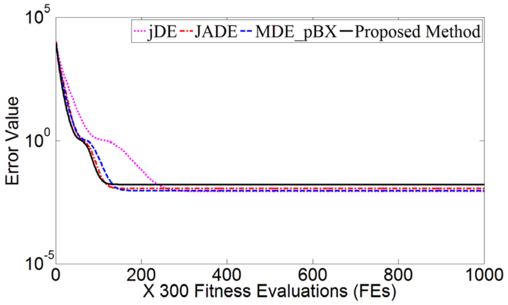


Figure 6-7. Convergence curves of F_7 function.

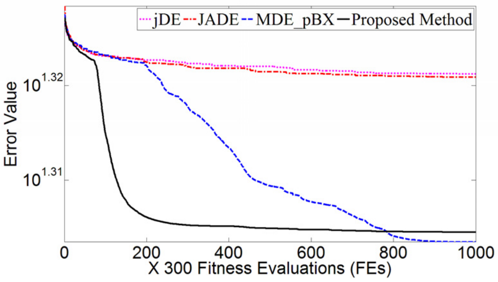


Figure 6-8. Convergence curves of F_8 function.

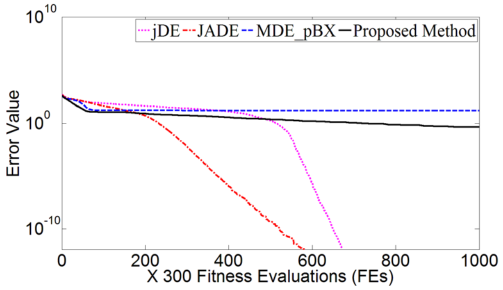


Figure 6-9. Convergence curves of F_9 function.

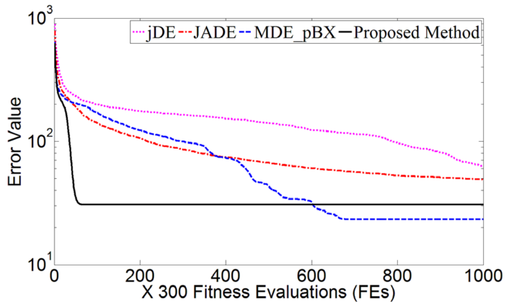


Figure 6-10. Convergence curves of F_{10} function.

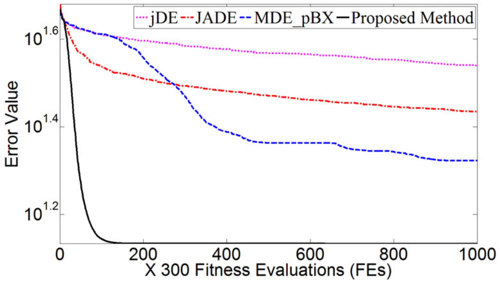


Figure 6-11. Convergence curves of F_{11} function.

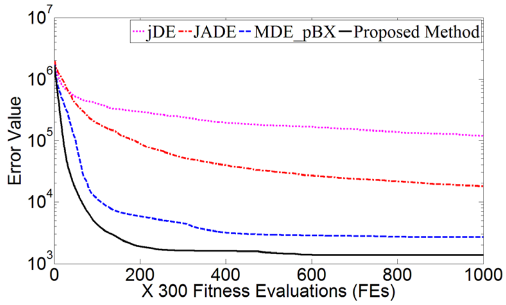


Figure 6-12. Convergence curves of F_{12} function.

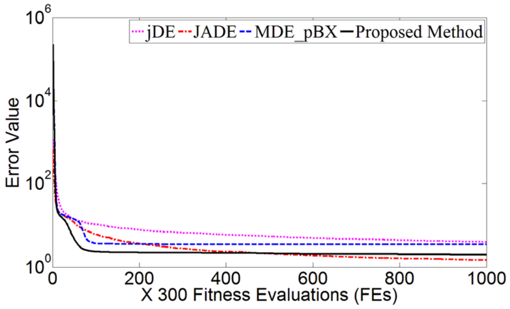


Figure 6-13. Convergence curves of F_{13} function.

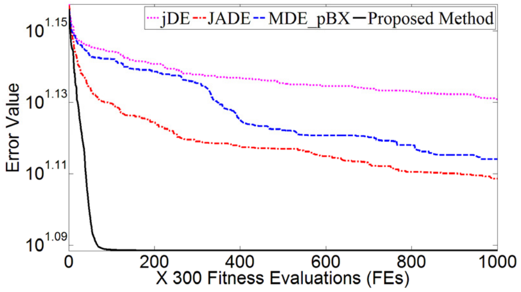


Figure 6-14. Convergence curves of F_{14} function.

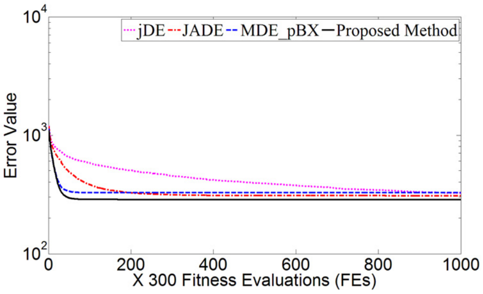
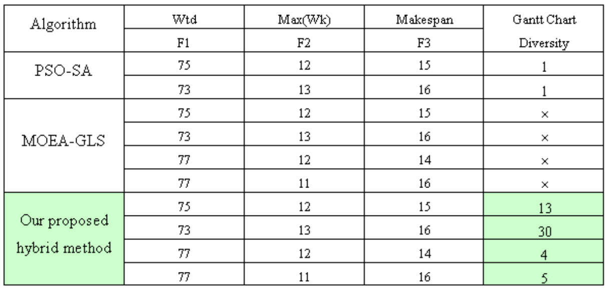


Figure 6-15. Convergence curves of F_{15} function.

From the simulational results, the proposed hybrid DE model gets better results such as F_2, F_5, F_{11}, F_{12}, F_{14} and F_{15} functions and F_1 archive the same results which the real optimum solutions are found. In summary, according to the results shown in Tables 3 to 5, the proposed DE model is highly competitive to the above-mentioned state-of-the-art DEs. The results of the proposed DE are better than, or comparable to, those of the state-of-the-art DEs in terms of the quality of the final solutions.

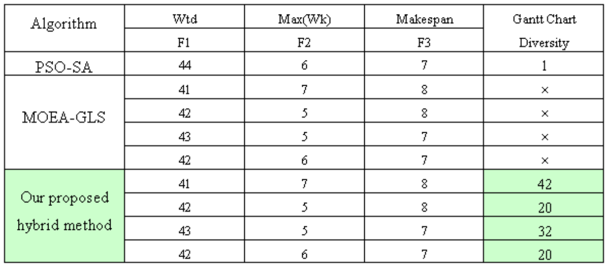
4.2 Multi-objective FJSP benchmarks

To illustrate the effectiveness and performance of the hybrid DE for the multi-objective verification, three FJSP representative benchmarks which are problem 8×8, 10×10, 15×10 have been conducted, and to compare with other related works, such as the PSO-SA [17] and MOEA-GLS [18]. For each problem, the obtained results are reported in table contains three objectives: F_1 (total workload), F_2 (critical machine workload), F_3 (makespan), are mentioned in Section 2. The solutions found from three methods are shown in Table 6 to Table 8.



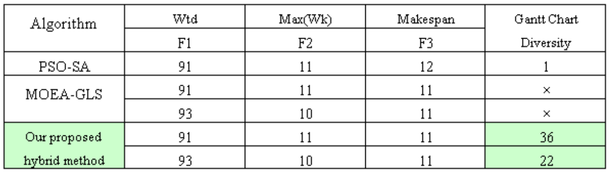
Algorithm	Wtd	Max(Wk)	Makespan	Gantt Chart
	F1	F2	F3	Diversity
PSO-SA	75	12	15	1
	73	13	16	1
MOEA-GLS	75	12	15	×
	73	13	16	×
	77	12	14	×
	77	11	16	×
Our proposed hybrid method	75	12	15	13
	73	13	16	30
	77	12	14	4
	77	11	16	5

Table 6. Experimental results of the FJSP problem 8×8.



Algorithm	Wtd	Max(Wk)	Makespan	Gantt Chart
	F1	F2	F3	Diversity
PSO-SA	44	6	7	1
MOEA-GLS	41	7	8	×
	42	5	8	×
	43	5	7	×
	42	6	7	×
Our proposed hybrid method	41	7	8	42
	42	5	8	20
	43	5	7	32
	42	6	7	20

Table 7. Experimental results of the FJSP problem 10×10.



Algorithm	Wtd	Max(Wk)	Makespan	Gantt Chart
	F1	F2	F3	Diversity
PSO-SA	91	11	12	1
MOEA-GLS	91	11	11	×
	93	10	11	×
Our proposed hybrid method	91	11	11	36
	93	10	11	22

Table 8. Experimental results of the FJSP problem 15×10.

For example, two new non-dominated solutions (77, 12, 14) and (77, 11, 16) can be obtained by the proposed hybrid DE compared with the PSO-SA, and more diversity Gantt chart solutions can be obtained by the hybrid DE compared with the MOEA-GLS. From the simulation results of Gantt chart diversity, the proposed hybrid DE approach performs significantly better than the other two PSO-SA and MOEA-GLS methods in solving all benchmarks. Afterward, three of the Gantt charts from the solution (75, 12, 15) for problem 8×8 are exhibited solutions diversity in Figure 7 to Figure 9 as giving an illustration.

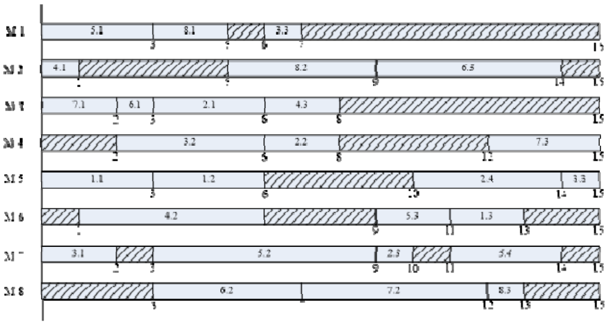


Figure 7. Solution (75, 12, 15) with Gantt Chart I.

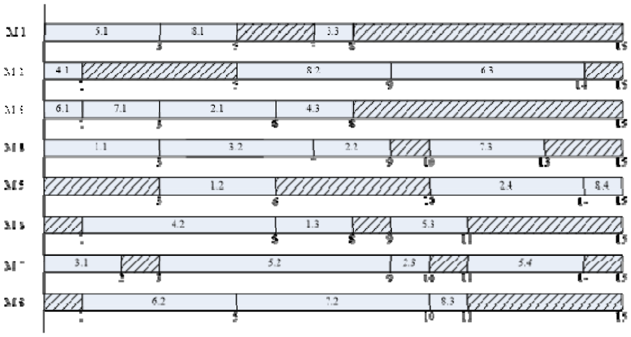


Figure 8. Solution (75, 12, 15) with Gantt Chart II.

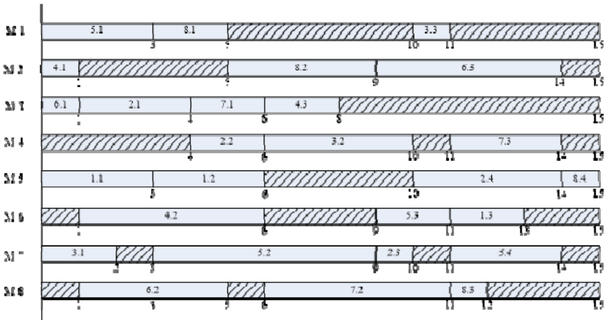


Figure 9. Solution (75, 12, 15) with Gantt Chart III.

5. Conclusion

In this paper, an improved Differential Evolution hybridized adaptive mutation (SAM) and wrapper local search (WLS) is proposed to improve both single-objective and multi-objective optimization performances of DE, and guide the evolution of the population toward the global optimum. Meanwhile, the WLS can disturb individuals to help individuals avoid trap into local minimum in evolution progress. Compare the proposed method with the published algorithms, the experimental results show that the proposed method exhibits better performance for solving most of the test functions from CEC 2005 benchmarks for the single-objective verifications. In addition, the wide range of Pareto-optimal solutions and the more Gantt charts diversity can be obtained for the multi-objective FJSP problems.

Acknowledgments

This work was partially supported by the National Science Council, Taiwan, under Grant NSC 101-2218-E-254-001.

References

[1] Y.C. Lin, “Mixed-integer constrained optimization based on memetic algorithm,” Journal of Applied Research and Technology, Vol. 11, No. 2, pp. 242–250, 2013.

[2] M. Nazir et al., “PSO-GA based optimized feature selection using facial and clothing information for gender classification,” Journal of Applied Research and Technology, Vol. 12, No. 1, pp. 145–152, 2014.

[3] K. Deb, “Multi-Objective Optimization Using Evolutionary Algorithms,” John Wiley, 2001.

[4] R. Storn and K. Price, “Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, Vol. 11, pp. 341–359, 1997.

[5] B. Alatas et al., “MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules,” Applied Soft Computing, Vol. 8, No. 1, pp. 646–656, 2008.

[6] K. Satoshi et al., “Differential evolution as the global optimization technique and its application to structural optimization,” Applied Soft Computing, Vol. 11, pp. 3792–3803, 2011.

[7] Boussaïd et al., “Two-stage update biogeography-based optimization using differential evolution algorithm,” Computers & Operations Research, Vol. 38, pp. 1188–1198, 2011.

[8] Himmat Singh and Laxmi Srivastava, “Modified Differential Evolution algorithm for multi-objective VAR management,” Electrical Power and Energy Systems, Vol. 55, pp. 731–740, 2014.

[9] Wenyin Gong et al., “Engineering optimization by means of an improved constrained differential evolution,” Computer Methods in Applied Mechanics & Engineering, Vol. 268, pp. 884-904, 2014.

[10] F. Neri and V. Tirronen, “Recent advances in differential evolution: A survey and experimental analysis,” Artif. Intell. Rev. 33, pp. 61–106, 2010.

[11] S. Das and P.N. Suganthan, “Differential Evolution: A survey of the state-of-the-art,” IEEE Trans. on Evolutionary Computation, Vol. 15, No. 1 pp. 4–31, 2011.

[12] J. Brest et al., “Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems,” IEEE Trans. on Evolutionary Computation, Vol. 10, No. 6, pp. 646–657, 2006.

[13] J. Zhang et al., “JADE: Adaptive differential evolution with optional external archive,” IEEE Trans. on Evolutionary Computation, Vol. 13, pp. 945–958, 2009.

[14] S.M. Islam et al., “An Adaptive Differential Evolution Algorithm with Novel Mutation and Crossover Strategies for Global Numerical Optimization,” IEEE Trans. on System, Man and Cybernetics, Part B - Cybernetics., Vol. 42, No. 2, pp. 482–500, 2012.

[15] A. Ghosh et al., “Self-adaptive differential evolution for feature selection in hyperspectral image data,” Applied Soft Computing, Vol. 13, pp. 1969–1977, 2013.

[16] D. L. Luo et al., “Ant colony optimization with local search applied to the flexible job shop scheduling problems,” ICCCAS conference in Communications, Circuits and Systems, pp. 1015-1020, 2008.

[17] W. Xia and Z. Wu, “An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems,” Computers and Industrial Engineering. Vol. 48, pp. 409-425, 2005.

[18] N. B. Ho and J. C. Tay, “Solving multiple-objective flexible job shop problems by evolution and local search,” IEEE Trans. on Systems, Man, and Cybernetics, Part C. Vol. 38, No. 5, pp. 674-685, 2008.

[19] Jun-qing Li et al., “An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems,” Computers & Industrial Engineering, Vol. 59, pp. 647–662, 2010.

[20] C. A. Coello and M. S. Lechuga, “MOPSO: A proposal for multiple objective particle swarm optimization,” In Proc. Congress Evolutionary Computation, vol. 1, Honolulu, pp. 1051-1056, 2002.

[21] Chun-Liang Lu et al., “Solving the Flexible Job-shop Scheduling Problem Based on Multi-Objective PSO with Pareto Diversity Search,” International Journal of Intelligent Information Processing, Vol. 4, pp.70-81, 2013.

[22] Min-Hui Lin and Chun-Liang Leu, “A Hybrid PSO-SVM Approach for Haplotype Tagging SNP Selection Problem,” International Journal of Computer Science and Information Security, Vol. 8, No. 6, pp. 60-65, 2010.

[23] Sheng-Ta Hsieh et al., “Real Random Mutation Strategy for Differential Evolution,” The 2012 Conference on Technologies and Applications of Artificial Intelligence, Tainan, Taiwan, 2012.

[24] P.N. Suganthan et al., “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” Technical Report, Nanyang Technological University, Singapore, 2005.