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.

\(^2,4\) 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-making 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-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_pBX [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_pBX 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 control 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_pBX, 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 initializes, 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 $\tilde{X}_{i,0} = [\tilde{x}_{i,0}^1, \tilde{x}_{i,0}^2, \cdots, \tilde{x}_{i,0}^D]^T$, $i = (1, 2, \cdots, N_p)$, 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 $\tilde{x}_{i,0}^j$ 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.

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

where $x_{\min}^j$ and $x_{\max}^j$ denote the lower and upper bounds, respectively, and $\text{rand}(\cdot)$ 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 \( \tilde{X}_{i,G} \) and two difference vectors \( \tilde{X}_{j,G} \) and \( \tilde{X}_{k,G} \) 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 \( \tilde{V}_{i,G} = [v_{i,G}^1, v_{i,G}^2, \ldots, v_{i,G}^D] \), \( i = 1, 2, \ldots, N_p \) is expressed as Eq. 2.

\[
\tilde{V}_{i,G} = \tilde{X}_{i,G} + F_i(\tilde{X}_{j,G} - \tilde{X}_{k,G})
\]  

(2)

where \( r_1, r_2, \) and \( r_3 \) 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 \( \tilde{U}_{i,G} = [u_{i,G}^1, u_{i,G}^2, \ldots, u_{i,G}^D] \), \( i = 1, 2, \ldots, N_p \) is created by means of crossover operation and is realized between each pair of target vector \( \tilde{X}_{i,G} \) and its corresponding donor vector \( \tilde{V}_{i,G} \). 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.

\[
u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if } \text{rand}(0,1) \leq C_r \text{ or } j = j_{\text{rand}}, \ j=1,2,\ldots,D \\ x_{i,G}^j & \text{otherwise}, \end{cases}
\]

(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 \([0, 1]\). \( j_{\text{rand}} \) 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 \( f(\cdot) \) 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.

\[
\tilde{X}_{i,G+1} = \begin{cases} \tilde{U}_{i,G} & \text{if } f(\tilde{U}_{i,G}) \leq f(\tilde{X}_{i,G}) \\ \tilde{X}_{i,G} & \text{otherwise}. \end{cases}
\]

(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.**

1: Basic setting of parameters for DE.
2: Initialization Generate the initial population.
3: Evaluate the fitness for each individual.
4: while Termination condition is not satisfied do
5: Mutation.
6: Crossover.
7: Selection.
8: Boundary constraints for each \( x_{i,G}^j \in \tilde{X}_{i,G} \).
9: end while
10: Output Global optimum solution \( \tilde{X}_{\text{best}} \).

2.2 Typical MDE_pBX Approach

Islam et al. developed an adaptive DE algorithm based on novel mutation and crossover strategies called MDE_pBX [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.
\[ \tilde{v}_{i,G} = \tilde{X}_{i,G} + F_t(\tilde{X}_{p,best,G} - \tilde{X}_{i,G} - \tilde{X}_{i',G} - \tilde{X}_{i'',G}) \]  

(5)

\[ \tilde{X}_{p,best,G} \] 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.

\[ p = \text{ceil} \left( \frac{N_p}{2} \times (1 - \frac{G - 1}{G_{\text{max}}}) \right) \]  

(6)

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

\[ F_i = \text{Cauchy}(F_m, 0.1) \]  

(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.

\[ F_m = w_p \times F_m + (1 - w_p) \times \text{mean}_{\text{pow}}(S_p) \]  

(8)

\[ w_p = 0.8 + 0.2 \times \text{rand}(0, 1) \]  

(9)

\[ \text{mean}_{\text{pow}}(S_p) = \frac{\sum_{s \in S_p} \left( \frac{1}{S_p} \right)^{1.5}}{1.5} \]  

(10)

A weight factor \( w_p \) is a positive constant number between 0.8 to 1. The \( \text{mean}_{\text{pow}}(\bullet) \) represents a power mean given by Eq. 10. The adaptation of crossover probability \( C_{r_i} \) is similar with the scale factor \( F_i \), and the detailed description as follows.

\[ C_{r_i} = \text{Gaussian}(C_{r_m}, 0.1) \]  

(11)

\[ C_{r_m} = w_{C_r} \times C_{r_m} + (1 - w_{C_r}) \times \text{mean}_{\text{pow}}(S_{C_r}) \]  

(12)

\[ w_{C_r} = 0.9 + 0.1 \times \text{rand}(0, 1) \]  

(13)

\[ \text{mean}_{\text{pow}}(S_{C_r}) = \frac{\sum_{s \in S_{C_r}} \left( \frac{1}{S_{C_r}} \right)^{1.5}}{1.5} \]  

(14)

where \( w_{C_r} \) is set between 0.9 to 1, \( C_{r_m} \) 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 a given 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 \( J_i (i = 1, 2, \ldots, n) \) on \( m \) machines \( M_j (k = 1, 2, \ldots, m) \), where each job \( J_i \) needs \( O_{ij} \) 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 \( O_i = \sum_{j=1}^{m} O_{ij} \). Each machine has a set of possible machines that can process the job, \( M_{ij} \), and the required time of \( j \)-th operation of \( i \)-th job on the \( k \)-th machine, \( P_{ijk} \), means the time required for the \( j \)-th operation of \( i \)-th job to use \( k \)-th machine. 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

\[ F_1 = \sum p_{i,j,k} \]  

(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

\[ F_2 = \max \{ W_1, W_2, \ldots, W_m \} \]  

(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

\[ F_3 = \max \{ CT_1, CT_2, \ldots, CT_n \} \]  

(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.

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_pBX, 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.

\[ \tilde{v}_{i,G} = \tilde{x}_{i,G} + F \left( \tilde{x}_{y_{B_{best}} - G} - \tilde{x}_{i,G} + \tilde{x}_{y_{i,G}} - \tilde{x}_{i,G} \right) \]  

(18)
Enhanced Differential Evolution Based on Adaptive Mutation and Wrapper Local Search Strategies for Global Optimization Problems, Chun-Liang Lu et al. / 1131-1143

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 \( \vec{x}_{i,G}^{\text{r}} \) and \( \vec{x}_{i,G}^{\text{g}} \) are two distinct vectors picked up randomly from the current population, and none of them is equal to \( \vec{x}_{i,G}^{\text{r}} \) or the target vector. The \( \vec{x}_{i,G}^{\text{g}} \) is dynamically chosen from top \( w \% \) individuals of each population. The linearly decreasing inertia weight is set as Eq. 19, where \( i_{\text{now}} \) is the current iteration, \( i_{\text{max}} \) is the user-defined maximum iteration, and the pre-defined lower and upper bounds of \( w \) described as \( w_{\text{min}} \leq w \leq w_{\text{max}} \).

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 \( \vec{x}_{i,G}^{\text{g}} \) 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.

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.

1: Basic setting of parameters for the DE model.
2: **Initialization** Generate the initial population.
3: Evaluate the fitness for each individual.
4: while Termination condition is not satisfied do
5:     for \( i=1 \) to \( N_p \) do
6:         for \( j=1 \) to \( D \) do
7:             **Mutation** with SAM approach.
8:         end for
9:     end for
10:     for \( i=1 \) to \( N_p \) do
11:         **Evaluation** the offspring.
12:     end for
13: end while
14: **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.

### Table 1. The example of operation schedules for FJSP.

<table>
<thead>
<tr>
<th>Jobs</th>
<th>Machines</th>
<th>( O_{1.1} )</th>
<th>( O_{1.2} )</th>
<th>( O_{1.3} )</th>
<th>( O_{2.1} )</th>
<th>( O_{2.2} )</th>
<th>( O_{3.1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_1 )</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>( M_2 )</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

#### 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 \( C_1 = \{ O_{1.1}, M_3, O_{2.1}, M_1, O_{3.1}, M_2 \} \).
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 \( C_2 = \{ O_{1.1}, O_{3.1}, O_{2.2}, M_2, O_{3.1}, M_2 \} \).
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 \( O_3 = \{ O_{1.2}, O_{2.2}, O_{1.1} \} \).
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.
Algorithm 3: The proposed hybrid DE model with PSOMA scheme for FJSP.

1: Basic setting of parameters for the DE model.
2: Initialization Generate the initial population via.
3: Evaluate the fitness for each individual.
4: while Termination condition is not satisfied do
5:     for $i=1$ to $N_p$ do
6:         for $j=1$ to $D$ do
7:             Mutation with SAM approach.
8:         end for
9:     end for
10: Decoding by PSOMA for each individual.
11: for $i=1$ to $N_p$ do
12:     Evaluate the offspring.
13: end for
14: Selection.
15: end for
16: Apply WLS on the better selected individuals.
17: Compare Gantt charts with solutions diversity.
18: Store multi-objective non-dominated solutions and Gantt charts to External Repository.
19: end while
20: 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_pBX 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.

<table>
<thead>
<tr>
<th>Functions</th>
<th>Global Optimum</th>
<th>Initialization Range</th>
<th>Search Range</th>
<th>Function Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td>-450</td>
<td>[-100, 100]^3</td>
<td>[-100, 100]^3</td>
<td>Unimodal</td>
</tr>
<tr>
<td>$F_2$</td>
<td>-450</td>
<td>[-100, 100]^3</td>
<td>[-100, 100]^3</td>
<td>Unimodal</td>
</tr>
<tr>
<td>$F_3$</td>
<td>-450</td>
<td>[-100, 100]^3</td>
<td>[-100, 100]^3</td>
<td>Unimodal</td>
</tr>
<tr>
<td>$F_4$</td>
<td>-450</td>
<td>[-100, 100]^3</td>
<td>[-100, 100]^3</td>
<td>Unimodal</td>
</tr>
<tr>
<td>$F_5$</td>
<td>-310</td>
<td>[-100, 100]^3</td>
<td>[-100, 100]^3</td>
<td>Unimodal</td>
</tr>
<tr>
<td>$F_6$</td>
<td>-390</td>
<td>[-100, 100]^3</td>
<td>[-100, 100]^3</td>
<td>Basic multimodal</td>
</tr>
<tr>
<td>$F_7$</td>
<td>-180</td>
<td>[0, 600]^3</td>
<td>[0, 600]^3</td>
<td>Basic multimodal</td>
</tr>
<tr>
<td>$F_8$</td>
<td>-140</td>
<td>[-32, 32]^3</td>
<td>[-32, 32]^3</td>
<td>Basic multimodal</td>
</tr>
<tr>
<td>$F_9$</td>
<td>-330</td>
<td>[-5, 5]^3</td>
<td>[-5, 5]^3</td>
<td>Basic multimodal</td>
</tr>
<tr>
<td>$F_{10}$</td>
<td>-330</td>
<td>[-5, 5]^3</td>
<td>[-5, 5]^3</td>
<td>Basic multimodal</td>
</tr>
<tr>
<td>$F_{11}$</td>
<td>90</td>
<td>[-0.5, 0.5]^3</td>
<td>[-0.5, 0.5]^3</td>
<td>Basic multimodal</td>
</tr>
<tr>
<td>$F_{12}$</td>
<td>-160</td>
<td>[-16, 16]^3</td>
<td>[-16, 16]^3</td>
<td>Expanded multimodal</td>
</tr>
<tr>
<td>$F_{13}$</td>
<td>-130</td>
<td>[-5, 5]^3</td>
<td>[-5, 5]^3</td>
<td>Expanded multimodal</td>
</tr>
<tr>
<td>$F_{14}$</td>
<td>-300</td>
<td>[-100, 100]^3</td>
<td>[-100, 100]^3</td>
<td>Expanded multimodal</td>
</tr>
<tr>
<td>$F_{15}$</td>
<td>120</td>
<td>[-5, 5]^3</td>
<td>[-5, 5]^3</td>
<td>Hybrid Composition</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Results</th>
<th>$F_1$</th>
<th>$F_2$</th>
<th>$F_3$</th>
<th>$F_4$</th>
<th>$F_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>jDE</td>
<td>Mean Std.</td>
<td>0.0008E+06</td>
<td>3.4595E-04</td>
<td>1.1038E-04</td>
<td>2.3619E-06</td>
<td>1.8258E-03</td>
</tr>
<tr>
<td>JADE</td>
<td>Mean Std.</td>
<td>0.0000E+06</td>
<td>3.8958E-04</td>
<td>9.8764E-04</td>
<td>3.7256E-04</td>
<td>5.4191E-02</td>
</tr>
<tr>
<td>MDE_pBX</td>
<td>Mean Std.</td>
<td>0.0000E+06</td>
<td>9.9567E-04</td>
<td>4.9567E-04</td>
<td>7.9567E-04</td>
<td>3.9567E-02</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>Mean Std.</td>
<td>0.0000E+06</td>
<td>9.9567E-04</td>
<td>4.9567E-04</td>
<td>7.9567E-04</td>
<td>3.9567E-02</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>Results</th>
<th>$F_6$</th>
<th>$F_7$</th>
<th>$F_8$</th>
<th>$F_9$</th>
<th>$F_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>jDE</td>
<td>Mean Std.</td>
<td>3.4279E-03</td>
<td>8.4430E-03</td>
<td>2.3917E+01</td>
<td>6.2777E+01</td>
<td>5.1205E+00</td>
</tr>
<tr>
<td>JADE</td>
<td>Mean Std.</td>
<td>4.0723E-03</td>
<td>1.1334E+02</td>
<td>2.0378E+01</td>
<td>6.2777E+01</td>
<td>5.1205E+00</td>
</tr>
<tr>
<td>MDE_pBX</td>
<td>Mean Std.</td>
<td>1.0694E+02</td>
<td>7.3657E+00</td>
<td>6.3958E+00</td>
<td>7.9567E+00</td>
<td>3.9567E+02</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>Mean Std.</td>
<td>1.5346E+02</td>
<td>1.6123E+01</td>
<td>2.6714E+01</td>
<td>1.6123E+01</td>
<td>2.6714E+01</td>
</tr>
</tbody>
</table>

Table 4. Experimental results of $F_6$ to $F_{10}$ test functions.
Enhanced Differential Evolution Based on Adaptive Mutation and Wrapper Local Search Strategies for Global Optimization Problems, Chun-Liang Lu et al. / 1131-1143

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.
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.

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.

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Whd</th>
<th>Max(W%)</th>
<th>Makespan</th>
<th>Gantt Chart Diversity</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO-SA</td>
<td>75</td>
<td>12</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>13</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>MOEA-GLS</td>
<td>70</td>
<td>12</td>
<td>15</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>13</td>
<td>16</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>77</td>
<td>12</td>
<td>14</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>77</td>
<td>11</td>
<td>16</td>
<td>x</td>
</tr>
<tr>
<td>Our proposed hybrid method</td>
<td>75</td>
<td>12</td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>13</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>77</td>
<td>12</td>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>77</td>
<td>11</td>
<td>16</td>
<td>5</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Whd</th>
<th>Max(W%)</th>
<th>Makespan</th>
<th>Gantt Chart Diversity</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO-SA</td>
<td>44</td>
<td>6</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>7</td>
<td>3</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>5</td>
<td>3</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>5</td>
<td>7</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>6</td>
<td>7</td>
<td>x</td>
</tr>
<tr>
<td>MOEA-GLS</td>
<td>41</td>
<td>7</td>
<td>3</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>5</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>5</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>6</td>
<td>7</td>
<td>20</td>
</tr>
<tr>
<td>Our proposed hybrid method</td>
<td>91</td>
<td>11</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>91</td>
<td>11</td>
<td>11</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>93</td>
<td>10</td>
<td>11</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>91</td>
<td>11</td>
<td>11</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>92</td>
<td>10</td>
<td>11</td>
<td>32</td>
</tr>
</tbody>
</table>

#### 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


