Article

Self-Adaptive Multi-Task Differential Evolution Optimization: With Case Studies in Weapon–Target Assignment Problem

Xiaolong Zheng, Deyun Zhou, Na Li, Tao Wu, Yu Lei and Jiao Shi

School of Electronics and Information, Northwestern Polytechnical University, ADD:127 West Youyi Road, Xi’an 710072, China; xzlheng@mail.nwpu.edu.cn (X.Z.); dyzhou@nwpu.edu.cn (D.Z.); linaflydream@mail.nwpu.edu.cn (N.L.); tao_woe@mail.nwpu.edu.cn (T.W.); jiaoshi@nwpu.edu.cn (J.S.)

* Correspondence: leiy@nwpu.edu.cn

Abstract: Multi-task optimization (MTO) is related to the problem of simultaneous optimization of multiple optimization problems, for the purpose of solving these problems better in terms of optimization accuracy or time cost. To handle MTO problems, there emerges many evolutionary MTO (EMTO) algorithms, which possess distinguished strategies or frameworks in the aspect of handling the knowledge transfer between different optimization problems (tasks). In this paper, we explore the possibility of developing a more efficient EMTO solver based on differential evolution by introducing the strategies of a self-adaptive multi-task particle swarm optimization (SaMTPSO) algorithm, and by developing a new knowledge incorporation strategy. Then, we try to apply the proposed algorithm to solve the weapon–target assignment problem, which has never been explored in the field of EMTO before. Experiments were conducted on a popular MTO test benchmark and a WTA-MTO test set. Experimental results show that knowledge transfer in the proposed algorithm is effective and efficient, and EMTO is promising in solving WTA problems.

Keywords: multi-task optimization; evolutionary multi-task optimization; evolutionary algorithm; differential evolution

1. Introduction

Multi-task optimization (MTO) [1,2] studies the simultaneous optimization of multiple optimization problems, for the purpose of achieving higher optimization performance on each optimization problem, especially when compared to traditional optimization methodologies. MTO assumes that there is some common knowledge or common patterns between related optimization problems (i.e., tasks), and this common knowledge can be exploited during the optimization process of related tasks. Based on these assumptions, studies on MTO explore all possible frameworks or strategies to exploit the relatedness and the common knowledge between tasks, so as to develop more efficient optimization solvers. The results of these studies show that MTO is promising in finding better solutions for a batch of optimization problems [3,4].

Generally, the optimization of multiple optimization problems (tasks) is described as an MTO problem. To solve MTO problems, researchers have explored many optimization techniques over the years, including the works based on Bayesian optimization and the works based on evolutionary algorithms. In 2016, Gupta et al. [1] explores the possibility of evolutionary MTO (EMTO) for the first time. Since then, many efficient EMTO works are published. Some works focus on the adaptation of knowledge transfer to task relatedness [5–8], and others studies focus on the feasibility and effectiveness of EMTO in handling real-world problems [9–12].

Among all published works, we focus our attention to the self-adaptive multi-task particle swarm optimization (SaMTPSO) algorithm developed in [13]. The SaMTPSO is developed using a traditional particle warm optimization algorithm. In the SaMTPSO, three main strategies, i.e., the developed knowledge transfer adaptation strategy, the focus
search strategy and the knowledge incorporation strategy, are devised for the purpose of achieving adaptive knowledge transfer. In these strategies, the knowledge source pool, the success memory and the failure memory of every task help the SaMTPSO to learn about the probability of positively transferring knowledge from a task to another tasks, which is then used to transfer knowledge between the task adaptively. Therefore, this paper tries to introduce these strategies into a differential evolution algorithm, to develop an efficient solver for MTO.

Projectile weapons have become a powerful threat to hostile targets due to their advantage in inflicting damage to the protected asset of participants from great distances. To eliminate this threat, in many countries, air defense emerged and is constantly searching for methods to reduce missile threat, which in return has increased the number and quality of available missiles. Therefore, the weapon–target assignment problem, solving the effective allocation of air defense resources (i.e., weapons and targets), was proposed and has attracted the interest of many researchers. Over the years, many works have been performed, such as research using evolutionary algorithms [14,15] or research using exact algorithms [16,17]; however, the WTA problem, known as an NP-complete problem, is difficult to solve, especially when the number of targets or weapons becomes larger and larger. To this end, this paper explores the possibility of further improving the optimization efficiency of WTA problems via EMTO techniques.

In this paper, the knowledge transfer adaptation strategy and the focus search strategy of the SaMTPSO are developed into the DE algorithms, which then enables the proposed algorithm to dynamically adapt to the complex inter-task relatedness of MTO problems. To better fit into the evolving structure of differential evolution (DE), a novel knowledge incorporation strategy is devised for the mutation of population, which helps the population generate better offspring through the incorporation of more useful information or knowledge. Befittingly, the proposed algorithm is named the self-adaptive multi-task differential evolution (SaMTDE) algorithm. In the experiments, after studying the knowledge transfer performance and the bi-task performance on a popular test benchmark, the SaMTDE was employed to solve WTA problems for the purpose of exploring the possibility of improving the optimization performance on these problems. Experimental results demonstrate the promising future of solving WTA problems via EMTO techniques.

The rest of the paper is organized as follows: In Section II, the mathematics model of MTO, some recent works on MTO and the WTA problem are introduced. In Section III, the details of the proposed algorithm are presented. In Section IV, the first two experiments are discussed to study the performance of the proposed algorithms. In the third experiment, the proposed algorithm was applied to solve the WTA problem; then, the performance of the SaMTDE is compared to that of several typical EMTO algorithms and traditional single-task algorithms. In the final section, some conclusions on this paper are detailed along with some possible future works.

2. Background

Multi-task optimization (MTO) studies how to solve multiple optimization problems simultaneously. In particular, if there is \( K \) optimization problems (each is assumed to be a minimization problem and is treated as a component task) to be optimized, the goal of MTO is to simultaneously optimize these \( K \) optimization tasks and find the optimal solution for each of the \( K \) tasks, i.e., \( \{x^*_1, x^*_2, \ldots, x^*_K\} = \arg\min \{f_1(x_1), f_2(x_2), \ldots, f_k(x_k)\} \), where \( x^*_t, x_t \in X_t, t = 1, 2, \ldots, K \) and \( X_t \) is a \( d_t \)-dimensional search space for the \( t \)-th task \( T_t \) as well as its objective function \( f_t : X_t \rightarrow \mathbb{R} \).

Research on MTO has been performed for many years. In 2013, Swersky et al. [18] employs Bayesian optimization [19,20] to develop their multi-task Bayesian optimization framework for the purpose of handling the problems of optimal hyperparameter settings more efficiently [21]. In 2016, Gupta et al. [1] develops a multi-factorial optimization framework based on a genetic algorithm to solve numerical optimization problems, which is the first work to study the evolutionary MTO (EMTO) techniques. Since then, many
EMTO works have been published [7,22–29]. In [2], Zheng et al. develops the flexible scheme by adapting knowledge transfer to task relatedness in the local region, which is implemented as a self-regulated evolutionary MTO (SREMTO) algorithm. In [30], Feng et al. introduced differential evolution (DE) into the framework of the above MFEA to develop a multi-factorial DE (MFDE) algorithm, where every individual can make use of the transferred knowledge from other tasks with a probability of \( r_{mp} \).

### 2.1. Self-Adaptive Multi-Task Particle Swarm Optimization

In [13], a self-adaptive multi-task particle swarm optimization (SaMTPSO) is proposed via the developed knowledge transfer adaptation strategy, the focus search strategy and the knowledge incorporation strategy.

#### Success Memory

| Index | \( T_1 \) | \( T_2 \) | ... | \( T_K \) |
|-------|-------|-------|-----|-------|
| 1     | \( n_{s_{T1}}^g \) | \( n_{s_{T2}}^g \) | ... | \( n_{s_{Tk}}^g \) |
| 2     | \( n_{s_{T1}}^{g+1} \) | \( n_{s_{T2}}^{g+1} \) | ... | \( n_{s_{Tk}}^{g+1} \) |
| ...   | ...   | ...   | ... | ...   |
| LP    | \( n_{s_{T1}}^{g+LP+1} \) | \( n_{s_{T2}}^{g+LP+1} \) | ... | \( n_{s_{Tk}}^{g+LP+1} \) |

#### Failure Memory

| Index | \( T_1 \) | \( T_2 \) | ... | \( T_K \) |
|-------|-------|-------|-----|-------|
| 1     | \( n_{f_{T1}}^g \) | \( n_{f_{T2}}^g \) | ... | \( n_{f_{Tk}}^g \) |
| 2     | \( n_{f_{T1}}^{g+1} \) | \( n_{f_{T2}}^{g+1} \) | ... | \( n_{f_{Tk}}^{g+1} \) |
| ...   | ...   | ...   | ... | ...   |
| LP    | \( n_{f_{T1}}^{g+LP+1} \) | \( n_{f_{T2}}^{g+LP+1} \) | ... | \( n_{f_{Tk}}^{g+LP+1} \) |

**Figure 1.** Task \( T_i \)'s success memory and failure memory for its \( K \) knowledge sources during the last \( LP \) generations.

In the knowledge transfer adaptation strategy, every component task is optimized by one subpopulation, whose members are part of the SaMTPSO’s population. Then, a knowledge source pool is devised for each task, and each pool consists of the \( K \) component tasks of an MTO problem, which is shown in Figure 1. Each of these tasks is now viewed as a knowledge source. For every individual in the subpopulation of each task, a candidate knowledge source will be chosen from the task’s pool to provide knowledge to help the individual generate better solutions, which is according to a probability learned from the task’s (the subpopulation’s) experience in generating promising solutions via these sources previously. In the strategy, the chosen probability of the \( k \)-th source in task \( T_i \)'s pool is defined as \( p_{t,i,k} \), \( l \in [1, 2, \ldots, K], k \in [1, 2, \ldots, K] \). \( p_{t,i,k} \) is initialized with \( 1/K \) to let all \( K \) knowledge sources of \( T_i \) have equal probability to be chosen at the beginning of optimization. A knowledge source for an individual of subpop \( T_i \) is chosen using the roulette wheel selection method [31]. After evaluating the generated offspring for task \( T_i \) using the transferred knowledge from these chosen sources at generation \( g \), the numbers of offspring generated via the \( k \)-th source that can successfully or unsuccessfully enter the next generation, i.e., \( n_{s_{T1}}^g \) and \( n_{f_{T1}}^g \), are respectively saved into the success memory and the failure memory of \( T_i \) as shown in Figure 1. With the \( LP \) records in the success memory and the failure memory, the corresponding probabilities \( p_{t,i,k} \), as shown in Figure 2, are updated by

\[
p_{t,i,k} = \frac{SR_{t,i,k}}{\sum_{k=1}^{K} SR_{t,i,k}},
\]

(1)
where

\[ SR_{t,k} = \frac{\sum_{j=g-LP+1}^{g} n_{s_{t,k}}^{j} + \sum_{j=g-LP+1}^{g} n_{f_{t,k}}^{j} + \varepsilon}{\sum_{j=g-LP+1}^{g} n_{s_{t,k}}^{j} + \sum_{j=g-LP+1}^{g} n_{f_{t,k}}^{j} + \varepsilon} + b_{p}, \]

where \( b_{p} \) is a based probability parameter to assign a small probability to those sources that have not been chosen by any individual in task \( T_{t} \) within the previous \( LP \) generations and \( \varepsilon = 0.001 \) is a small constant value used to avoid the possibility of

\[ \sum_{j=g-LP+1}^{g} n_{s_{t,k}}^{j} + \sum_{j=g-LP+1}^{g} n_{f_{t,k}}^{j} = 0 \]

in the first item. To ensure the probabilities of choosing sources in each task’s pool are always summed to 1, the SaMTPSO further divided \( SR_{t,k} \) by \( \sum_{k=1}^{KS} SR_{t,k} \) to calculated \( p_{t,k} \).

| Tasks | KSs | T1  | T2  | ... | TK |
|-------|-----|-----|-----|-----|----|
| T1    | p_{1,1} | p_{1,2} | ... | p_{1,K} |
| T2    | p_{2,1} | p_{2,2} | ... | p_{2,K} |
| ...   | ...   | ...   | ... | ...   |
| TK    | p_{K,1} | p_{K,2} | ... | p_{K,K} |

Figure 2. The learned probabilities on \( K \) knowledge sources (denoted as KSs) for every task at generation \( g \).

In the focus search strategy, the SaMTPSO monitors a task’s success memory to observe the task’s failures of knowledge transfer on every knowledge source task. Specifically, if every \( n_{s_{t,k}}^{j} = 0, k \in [1,2,\ldots,K], j \in [1,2,\ldots,LP] \) in the success memory of task \( T_{t} \), then the SaMTPSO activates the focus search strategy for the task \( T_{t} \), which then only allows each individual in \( subpop_{t} \) choose a source task \( \{T_{k}\}_{k=1}^{K} \). As a result, task \( T_{t} \) can only explore and exploit the knowledge from itself, i.e., the knowledge source task is the task itself; therefore, there is only intra-task knowledge transfer remained. For the other tasks, the knowledge transfer adaptation strategy still works if the focus search strategy is not activated for them. For task \( T_{t} \), the focus search strategy will be deactivated if any progress are made for this task.

In the knowledge incorporation strategy, the authors develop two types of knowledge incorporation forms, for the purpose of helping every individual of each subpopulation to better exploit the transferred knowledge from its chosen source.

2.2. Weapon–Target Assignment Problem

The weapon–target assignment (WTA) problem is fundamental for the defense application of military operations. The problem is expected to find a proper assignment of interceptors (weapons) to the incoming missiles (targets) via the minimization of the expected damage of own-force assets. So far, there are two distinct categories of WTA problems, i.e., the static WTA (SWTA) and the dynamic WTA (DWTA) problems. The SWTA describes a simple scenario wherein a known number of targets (with known destructive values) are observed and a finite number of weapons (with known probabilities of successfully destroying the targets, in other words, the probabilities of a kill), engage the targets in a single stage. The DWTA is a multi-stage problem in which some weapons are engaged at the targets at a stage, then the DWTA assesses the outcome of this engagement and proposes new strategy for the next stage. In this paper, we only consider the SWTA problems, the WTA problems in the following paper refers to the SWTA problems.

In the WTA problem, we assume that there are \( W \) weapons and \( M \) targets, and all these weapons must be assigned to the targets. Then the WTA is to find a proper assignment for these weapons and targets through the minimization of the following equation:

\[
\min \sum_{j=1}^{M} \nu_{j} \prod_{i=1}^{W} (1 - \hat{p}_{i,j})^{A_{i,j}}
\] (2)
subject to,
\[ p_{i,j} \in [0.0, 1.0] \]
\[ A_{i,j} = 0 \text{ or } 1 \]
\[ \sum_{j=1}^{M} A_{i,j} = 1, \text{ for } i = 1, 2, \ldots, m, \]

where \( v_j, j = 1, 2, \ldots, M \) is the known destructive value of target \( j \), \( p_{i,j} \) is the probability of weapon \( i \) successfully destroying target \( j \) and \( A_{i,j} \) is a Boolean value indicating whether to assign the \( i \)-th weapon to the \( j \)-th target.

The WTA problem is known as an NP-complete problem; therefore, it is very hard for conventional evolutionary algorithms to find the global optimal solution for a WTA problem. In this paper, we try to study the possibility of more efficiently resolving the WTA problems through the newly emerging EMTO techniques.

3. Self-Adaptive Multi-Task Differential Evolution Optimization

In this section, an efficient self-adaptive multi-task differential evolution (SaMTDE) algorithm is developed by introducing the knowledge transfer adaptation strategy and the focus search strategy of the SaMTPSO. Moreover, a novel knowledge incorporation strategy is developed for the SaMTDE. The basic structure of the SaMTDE is shown in Algorithm 1.

In the SaMTDE, a population with \( N \) individuals is first split into \( K \) subpopulations (\( \{ \text{subpop}_t \}_{t=1}^{K} \)), each optimizing one optimization task. According to the knowledge transfer adaptation strategy in the SaMTPSO, for a task \( T_t \), a success memory and a failure memory as described in Figure 1 are used to record the number of success and failures of individuals in \( \text{subpop}_t \) in using the knowledge from their chosen tasks over the last \( LP \) generations, which are then used to compute the learned probabilities \( \{ p_{t,k} \}_{k=1}^{K} \) for the task \( T_t \) according to Equation (1). Because the learned probabilities \( \{ p_{t,k} \}_{k=1}^{K} \) of \( T_t \) have statistically described the probability of every task in helping task \( T_t \) successfully, every individual in \( \text{subpop}_t \) in the next generation can make full use of these probabilities by the roulette wheel selection method as shown in step 13 in Algorithm 1, so as to choose a most suitable knowledge source task whose knowledge can help it generate better offspring.

For the focus search strategy in this proposed SaMTDE, an activate flag \( isFocus \) is set to 1 if there are no success records in a task \( T_t \)’s success memory in step 24, for the purpose of activating the focus search on \( T_t \), which is achieved by allowing the individuals in \( \text{subpop}_t \) to choose a knowledge source task from the only option as performed in step 11.

3.1. Knowledge Incorporation Strategy and Offspring Generation

In a typical DE algorithm, a mutant vector \( v_i \) is first generated for every individual \( ind_i \) of population using a mutation strategy, such as:
\[ v_i = x_{i_1} + F \times (x_{i_2} - x_{i_3}) \]  

where, \( x_i \) is the genes of \( ind_i \). \( i_1, i_2, i_3 \) are the mutually exclusive indices of individuals in the population. \( F \) is a parameter to scale the difference in each formula.

To devise the knowledge incorporation strategy, a knowledge source has been chosen for every individual in the proposed algorithm in the step 11 and 13 of Algorithm 1, which means the individual can make use of the knowledge transferred from this source. Then the knowledge incorporation strategy generates a mutant vector for an individual by incorporating the knowledge from the individual’s chosen source, in specifically, by revising the above Equation (3) for every individual \( ind_{i,j} \) of subpopulation \( \text{subpop}_t \) as follow:
\[ v_{i,j} = x_{i,j_1} + F \times (x_{i,j_2} - x_{i,j_3}), \]
where the genes $x_{ik}$ of individual $ind_{ik}$ from $subpop_{ik}$ are transferred to help generate better mutant vector for individual $ind_{it}$. $ik$ is the index of the individual’s chosen knowledge source task.

**Algorithm 1 SaMTDE.**

**Input:**

- $N$ (population size)
- $K$ (number of tasks)
- $LP$ (learning period)
- $bp$ (base probability parameter)
- $F, CR$ (DE parameters)

**Output:**

- $\{x_1^*, x_2^*, \ldots, x_K^*\}$ (the found-best solutions of the $K$ tasks)

1: Initialize a population with $N$ individuals, and randomly separate these individuals into $K$ subpopulations $\{subpop_t\}_{t=1}^K$ of size $N_s = N/K$. Individuals of $subpop_t$ are represented as $\{x_{t,i}\}_{i=1}^{N_s}$.
2: Initialize a success memory and failure memory for every task, so do the initial learned probabilities on the $K$ knowledge sources of every task, i.e., $p_{t,k} = \frac{1}{K}$, $t \in [1, 2, \ldots, K], k \in [1, 2, \ldots, K]$.
3: Evaluate the individuals of $subpop_t$ on task $T_t$ to obtain their fitness $\{fitness_{t,i}\}_{i=1}^{N_s}$, the found best solution $x_{t,\text{best}}$ and the corresponding fitness $fitness_{t,\text{best}}$ for $T_t$.
4: Generation $g = 1$.
5: while (stopping conditions are not satisfied) do
   6:     Generation $g \leftarrow g + 1$.
   7:     % subpop$_t$ generates new individuals.
   8:     for task $T_t, t = 1$ to $K$ do
      9:         for every individual of $subpop_t$ do
         10:             if isFocus then
            11:                 Set $ik \leftarrow t$ to perform focus search strategy.
            12:             else
               13:                 Perform roulette wheel selection on $\{p_{t,k}\}_{k=1}^K$ to obtain the index $ik$ of the chosen source task.
            14:             end if
            15:             Set $iks_{t,i} \leftarrow ik$ to save the knowledge source task for every individual.
            16:             Transfer knowledge by generating offspring (see Algorithm (2)).
         17:     end for
   18: end for
19: Evaluate the offspring of $subpop_t$ on task $T_t$, and update the $T_t$’s success memory and failure memory (see Algorithm (3)).
20: % Update probability $p_{t,k}$.
21: if $g > LP$ then
   22:     for task $T_t, t = 1$ to $K$ do
      23:         Update probabilities $\{p_{t,k}\}_{k=1}^K$.
      24:         Set isFocus $\leftarrow 1$ if no success records in $T_t$’s success memory, otherwise set isFocus $\leftarrow 0$.
      25:         Remove $n_{s_{t,k}}^{-L_P+1}$ and $n_{f_{t,k}}^{-L_P+1}, k \in [1, 2, \ldots, K]$, out of the memories.
   26:     end for
27: end if
28: end while

In Equation (4), the genes of a random selected individual $ind_{ik}$ are transferred from a chosen source task $T_{ik}$. In the field of EMTO, task’s best genes found by the whole population are also the best choice to transfer between tasks. Though these genes may be the representative ones on tasks, the frequent transfer of such genes may result in negative impacts from the knowledge source tasks, especially when the source tasks become easily
stuck in local optima. Therefore, the random selection process in the Equation (4) should be better in handling MTO problems.

In Algorithm 2, a mutant vector $v_{t,i}$ is first generated for individual $ind_{t,i}$ according to Equation (4) as shown in step 1. Next, crossover operation is performed on the genes $x_{t,i}$ of $ind_{t,i}$ and its mutant vector $v_{t,i}$ to generate a trail vector $u_{t,i}$ in step 2, which is formulated as follow:

$$u_{t,i} = \begin{cases} v_{t,i} & \text{if } (\text{rand} \leq CR) \text{or } (j = j_{\text{rand}}) \\ x_{t,i} & \text{otherwise} \end{cases}$$  (5)

where crossover rate $CR$ is a parameter of range $[0, 1]$ to control the portion of $u_{t,j}$ that are copied from the mutant vectors $v_{t,j}$. $j_{\text{rand}}$ is an integer that is randomly chosen from range $[1, D]$, which is used to ensure the crossover operation is performed on at least one dimension of the trail vector. $D$ is the dimension of search space.

**Algorithm 2** Knowledge Transfer.

**Input:**
- $ik$ (index for the chosen knowledge source)
- $F, R$ (DE parameters)

**Output:**
- $x_{t,i}$ (individual’s position)

1. Generate a mutant vector $v_{t,i}$ by applying the mutation operation (Equation (4)) to $subpop_t$.
2. Perform crossover (Equation (5)) on the mutant vector to obtain a trail vector $u_{t,i}$.

### 3.2. Evaluations and Selections

After crossover, trail vectors are generated for the population. In Algorithm 3, these trail vectors are evaluated on their corresponding tasks. Specifically, a trail vector $u_{t,i}$ is evaluated on the fitness function $f_{t}^{*}(\cdot)$ of task $T_t$ as shown in step 7. Note that, every trail vector is only evaluated on one task, which greatly reduces the evaluation costs for the proposed SaMTDE.

After evaluation, selection operation is used to update each individual by choosing the better one between the individual and its trail vector. By comparing the fitness of an individual $ind_{t,i}$ and its trail vector $u_{t,i}$, the genes $x_{t,i}$ of individual is updated using the better one as shown in the follow:

$$x_{t,i} \leftarrow \begin{cases} u_{t,i} & \text{if } f(u_{t,i}) \leq f(x_{t,i}) \\ x_{t,i} & \text{otherwise} \end{cases}$$  (6)

which are implemented in the steps 6–15 of Algorithm 3. Meanwhile, the found best solution and the associated fitness for each task will be updated if a best gene is found within these trail vectors in step 10. At the same time, if a trail vector is better than the original individual, then the $ns_{T,ik}^{t}$ is updated as shown in step 8, which has indicated that the knowledge from the chosen source $T_{ik}$ has helped in generating better genes. Contrarily, only the $nf_{T,ik}^{t}$ is updated if a trail vector is not better than the original individual as shown in step 14. After updating, the $ns_{T,ik}^{t}$ and $nf_{T,ik}^{t}$ are stored in the success memory and the failure memory of the corresponding task $T_t$ in step 17.
Algorithm 3 Evaluations.

Input:
- \( K \) (number of tasks)
- \( iks_{t,i} \) (the index of the chosen source by individual \( ind_{t,i} \))

Output:
- \( \{x^*_1, x^*_2, \ldots, x^*_K\} \) (the found best solution on the \( K \) tasks)

1: % Evaluate subpopulation on task \( T_t \).
2: for task \( T_t, t = 1 \) to \( K \) do
3: for every offspring of subpopulation do
4: \( ik \leftarrow iks_{t,i} \).
5: % Evaluation and Selection.
6: if \( f(u_{t,i}) < \text{fitness}_{t,i} \) then
7: \( x_{t,i} \leftarrow u_{t,i}, \text{fitness}_{t,i} \leftarrow f_t(u_{t,i}) \).
8: \( ns^g_{t,ik} \leftarrow ns^g_{t,ik} + 1 \).
9: if \( \text{fitness}_{t,i} < \text{fitness}_{t,i,\text{best}} \) then
10: \( x_{t,\text{best}} \leftarrow x_{t,i}, \text{fitness}_{t,\text{best}} \leftarrow \text{fitness}_{t,i} \).
11: end if
12: else
13: \( x_{t,i} \leftarrow x_{t,i}, \text{fitness}_{t,i} = \text{fitness}_{t,i} \).
14: \( nf^g_{t,ik} \leftarrow nf^g_{t,ik} + 1 \).
15: end if
16: end for
17: Save \( ns^g_{t,ik} \) and \( nf^g_{t,ik} \) into the corresponding memories.
18: end for
19: Update the found best solutions \( \{x^*_1, x^*_2, \ldots, x^*_K\} \).

4. Experiments

A set of experiments were conducted to demonstrate the performance of the developed SaMTDE by comparison with the MFDE, SaMTPSO, SREMTO, MFEA, JADE [32] as well as a single-task DE algorithm. Meanwhile, the SaMTDE was applied to solve multiple weapon–target assignment problems as previously introduced, which demonstrates the promising future of the SaMTDE in handling real-world optimization problems. Before the experiments are discussed, the employed test problems are presented as well as the experiment environments.

4.1. Test Problems

Two test suites were considered in the following experiments. Test suite I, shown in Table 1, contains 9 popular bi-task MTO problems that are used in the CEC’2017 Evolutionary Multi-Task Optimization Competition. These 9 well designed MTO problems differ from each other in the aspects of the overlap degree in their global optima and the inter-task relatedness. In the table, the optima of the two component tasks in problems 1 to 3 are of complete intersection, which means the component tasks have the same optima in the unified search space. For problems 7 to 9, the component tasks have no intersection in their global optima. For problems 4 to 6, the component tasks of partial intersection in their global optima. In each of these problems, the inter-task relatedness is measured via the inter-task similarity \( R_{ij} \), which is computed via the Spearman’s rank correlation coefficient [33]. Details of these problems are given in Table 1.

In test suite II, 9 MTO problems were constructed using WTA problems. As shown in Table 2, each MTO problem consists of two component tasks, each task to be a WTA problem. For a WTA problem, the number of weapons and targets may be different. In problems 1 to 3, the number of weapons and the targets are the same in each task, and are the same as well between the two component tasks of each MTO problem. In problems 4 to 6, the number of weapons and the targets are the same in each task, but are not the same between the two component tasks of each MTO problem. In problems 7 to 9, the number
of weapons and the targets are not the same in each task, but are the same between the
two component tasks of each MTO problem. For the weapons and the targets in each task,
the probability of a weapon destroys a target and the destructive value of every target are
details in Tables A1–A9, which are attached in the Appendix A of this paper.

### Table 1. Details of the 9 test problems in test suite I.

| Problem | Component Tasks | Dimensions $D$ | Global Optima $x^*$ | Optimal Values $f(x^*)$ | Search Ranges | Inter-Task Similarity $R_s$ |
|---------|-----------------|----------------|----------------------|-------------------------|---------------|-----------------------------|
| 1       | $T_1$: Griewank  | 50             | $(0, 0, \ldots, 0)$  | 0                       | $[-100, 100]$ | 1.00                        |
|         | $T_2$: Rastrigin |                | $(0, 0, \ldots, 0)$  | 0                       | $[-50, 50]$   |                             |
| 2       | $T_1$: Ackley    | 50             | $(0, 0, \ldots, 0)$  | 0                       | $[-50, 50]$   | 0.22                        |
|         | $T_2$: Rastrigin |                | $(0, 0, \ldots, 0)$  | 0                       | $[-50, 50]$   |                             |
| 3       | $T_1$: Ackley    | 50             | $[42.096, 42.096, \ldots, 42.096]$ | 0 | $[-50, 50]$ | 0.00                        |
|         | $T_2$: Schwefel  |                | $[420.968, 420.968, \ldots, 420.968]$ | 0 | $[-500, 500]$ | |  |
| 4       | $T_1$: Rastrigin | 50             | $(0, 0, \ldots, 0)$  | 0                       | $[-50, 50]$   | 0.86                        |
|         | $T_2$: Sphere    |                | $(0, 0, \ldots, 20)$ | 0                       | $[-100, 100]$ |                             |
| 5       | $T_1$: Ackley    | 50             | $(0, 0, \ldots, 1)$  | 0                       | $[-50, 50]$   | 0.21                        |
|         | $T_2$: Rosenbrock|                | $(1, 1, \ldots, 1)$  | 0                       | $[-50, 50]$   |                             |
| 6       | $T_1$: Ackley    | 50 (T2: 25)    | $(0, 0, \ldots, 0)$  | 0                       | $[-50, 50]$   | 0.07                        |
|         | $T_2$: Weierstrass|               | $(0, 0, \ldots, 0)$  | 0                       | $[-0.5, 0.5]$ |                             |
| 7       | $T_1$: Rosenbrock| 50             | $(1, 1, \ldots, 1)$  | 0                       | $[-50, 50]$   | 0.94                        |
|         | $T_2$: Rastrigin |                | $(0, 0, \ldots, 0)$  | 0                       | $[-50, 50]$   |                             |
| 8       | $T_1$: Griewank  | 50             | $(10, 10, \ldots, 10)$ | 0 | $[-100, 100]$ | 0.36                        |
|         | $T_2$: Weierstrass|              | $(0, 0, \ldots, 0)$  | 0                       | $[-50, 50]$   |                             |
| 9       | $T_1$: Rastrigin | 50             | $[42.096, 42.096, \ldots, 42.096]$ | 0 | $[-500, 500]$ | 0.00                        |
|         | $T_2$: Schwefel  |                | $[420.968, 420.968, \ldots, 420.968]$ | 0 | $[-500, 500]$ | |  |

### Table 2. Details of the 9 MTO problems regarding the WTA in test suite II.

| Problem | Component Task | # Weapons ($D$) | # Targets |
|---------|----------------|----------------|-----------|
| 1       | $T_1$          | 5              | 5         |
|         | $T_2$          | 5              | 5         |
| 2       | $T_1$          | 10             | 10        |
|         | $T_2$          | 10             | 10        |
| 3       | $T_1$          | 20             | 20        |
|         | $T_2$          | 20             | 20        |
| 4       | $T_1$          | 5              | 5         |
|         | $T_2$          | 10             | 10        |
| 5       | $T_1$          | 5              | 5         |
|         | $T_2$          | 20             | 20        |
| 6       | $T_1$          | 10             | 10        |
|         | $T_2$          | 20             | 20        |
| 7       | $T_1$          | 5              | 10        |
|         | $T_2$          | 5              | 10        |
| 8       | $T_1$          | 5              | 20        |
|         | $T_2$          | 5              | 20        |
| 9       | $T_1$          | 10             | 20        |
|         | $T_2$          | 10             | 20        |

### 4.2. Experimental Setup

In the following experiments, we (1) analyze the effectiveness of knowledge transfer
in the SaMTDE by comparing the performance on test suite I; (2) compare the proposed
SaMTDE with the SaMTPSO, the MFDE, the SREMTO, the MFEA and the DE in terms of
the performance on the 9 popular MTO problems from Table 1; (3) apply the SaMTDE to solve the WTA problems from Table II while comparing the results of the SaMTDE with that of the SaMTPSO, the MFDE, the SREMTO, the MFEA, the DE and the JADE.

The parameter settings for the SaMTDE is according to the SaMTPSO. The crossover rate and the scaling factor in SaMTDE are set experientially according to Zhang et al. [32]. The settings for the SaMTPSO, the MFDE, the SREMTO, the MFEA, the DE and the JADE are as suggested in the original literature.

1. Population size: \( n = 50 \times k \), where \( k \) is the number of component tasks.

2. Parameter settings in SaMTDE:
   - based probability parameter \( bp \): 0.001.
   - learning period \( LP \): 10.
   - crossover rate \( CR \): randomly generated from a normal distribution with mean 0.4 and sigma 0.1.
   - scaling factor \( F \): randomly generated from a cauchy distribution with location parameter 0.3 and scale parameter 0.1.

3. Parameter settings in the DE is same as the settings in the SaMTDE.

4. Parameter settings in SaMTPSO:
   - based probability parameter \( bp \): 0.001.
   - learning period \( LP \): 10.
   - inertia weight parameter \( w \): decreases linearly from 0.9 to 0.4.
   - coefficient: \( c_1 = c_2 = 1.494 \) in the SaMTPSO (settings for the SaMTPSO-S1 in the original literature).

5. Parameter settings in MFDE:
   - population size \( n \): 100.
   - random mating probability \( rmp \): 0.3.
   - scaling factor \( F \): 0.5.
   - crossover rate \( CR \): 0.9.

6. Parameter settings in SREMTO:
   - probability for crossover: \( P_\alpha = 0.7 \).
   - probability for mutation: \( P_\beta = 1.0 \).
   - distribution index of SBX: 1.
   - distribution index of PM: 39.

7. Parameter settings in MFEA:
   - random mating probability \( rmp \): 0.3.
   - distribution index of SBX: 2.
   - distribution index of PM: 5.

8. Parameter settings in JADE:
   - crossover rate \( CR \): randomly generated from a normal distribution with mean 0.5 and sigma 0.1.
   - scaling factor \( F \): randomly generated from a Cauchy distribution with location parameter 0.5 and scale parameter 0.1.

A total of 30 runs were conducted for all considered algorithms on every MTO test problem. The stopping criterion is the maximum number of function evaluations (maxFEs), which is set to 100,000, as is the case in many research papers. After the 30 runs on each MTO problem, the mean and standard deviation of the achieved objective function values (FVs) are presented for every algorithm. To compare the performance difference between multiple algorithms in the scenery of multitasking, a popular performance score is computed for the involved algorithms based on their achieved results on each MTO problem. According to [33], the score of an algorithm \( q, q \in [1, 2, \ldots, Q] \) is computed by
\[
\text{score}_q = \frac{1}{K} \sum_{j=1}^{L} \left( \frac{1}{\sigma_j} \sum_{l=1}^{K} \left( I_{q,j,l} - \mu_j \right) \right)
\]
where \( I_{q,j,l} \) is the algorithm \( q \)'s achieved FVs on the \( j \)-th task.
in $l$-th run, $\mu_j$ and $\sigma_j$ are the mean and the standard deviation of the achieved FVs by all algorithms in all runs. Hence, an algorithm will have a smaller score if it performs better according to this definition.

4.3. Experimental Results and Discussions

4.3.1. The Effectiveness of Knowledge Transfer in the SaMTDE

To study the effectiveness of knowledge transfer in the proposed SaMTDE, this experiment compared the achieved results of the SaMTDE with the DE algorithm on the 9 MTO test problems in Table 1. As previously introduced, the SaMTDE employs the same mutation operation, crossover operation, selection operation and parameter settings as the DE algorithm; by comparing their performance difference, we can directly observe the effectiveness of knowledge transfer in the SaMTDE. Table 3 reports the means and bracketed standard deviations of the achieved results by both the algorithms over 30 runs.

| Problem | Task | SaMTDE                                  | DE                                  |
|---------|------|-----------------------------------------|-------------------------------------|
| 1       | $T_1$ | $T_2$ | Mean (Std) | Score | Mean (Std) | Score |
| 2       | $T_1$ | $T_2$ | $4.30 \times 10^{-3} (8.55 \times 10^{-3})$ | $-4.90 \times 10^{+1}$ | $4.85 \times 10^{-1} (3.11 \times 10^{-1})$ | $2.39 \times 10^{-2} (7.37 \times 10^{-3})$ | $4.90 \times 10^{+1}$ |
| 3       | $T_1$ | $T_2$ | $1.44 \times 10^{-1} (2.87 \times 10^{-1})$ | $-5.06 \times 10^{+1}$ | $2.78 \times 10^{-1} (1.31 \times 10^{+0})$ | $2.42 \times 10^{-2} (8.98 \times 10^{-3})$ | $5.06 \times 10^{+1}$ |
| 4       | $T_1$ | $T_2$ | $6.74 \times 10^{+0} (9.43 \times 10^{+0})$ | $-4.06 \times 10^{+1}$ | $2.12 \times 10^{-1} (3.87 \times 10^{-2})$ | $1.05 \times 10^{+3} (3.14 \times 10^{+2})$ | $4.06 \times 10^{+1}$ |
| 5       | $T_1$ | $T_2$ | $2.59 \times 10^{+7} (7.77 \times 10^{+7})$ | $-3.44 \times 10^{+0}$ | $2.37 \times 10^{-2} (1.01 \times 10^{-2})$ | $2.55 \times 10^{+3} (3.07 \times 10^{+2})$ | $3.44 \times 10^{+0}$ |
| 6       | $T_1$ | $T_2$ | $6.73 \times 10^{-1} (6.89 \times 10^{-1})$ | $-3.18 \times 10^{+1}$ | $2.80 \times 10^{-1} (1.37 \times 10^{+0})$ | $2.94 \times 10^{+5} (5.41 \times 10^{+5})$ | $3.18 \times 10^{+1}$ |
| 7       | $T_1$ | $T_2$ | $1.83 \times 10^{+1} (1.06 \times 10^{+1})$ | $-3.23 \times 10^{+1}$ | $2.87 \times 10^{+0} (1.55 \times 10^{+0})$ | $2.67 \times 10^{-1} (1.52 \times 10^{-1})$ | $3.23 \times 10^{+1}$ |
| 8       | $T_1$ | $T_2$ | $2.82 \times 10^{+1} (2.24 \times 10^{+1})$ | $-3.40 \times 10^{+1}$ | $3.29 \times 10^{+5} (9.75 \times 10^{+5})$ | $2.52 \times 10^{-2} (7.10 \times 10^{-1})$ | $3.40 \times 10^{+1}$ |
| 9       | $T_1$ | $T_2$ | $1.72 \times 10^{-1} (1.74 \times 10^{-1})$ | $-3.23 \times 10^{+1}$ | $3.82 \times 10^{-1} (3.15 \times 10^{-1})$ | $1.13 \times 10^{+1} (3.55 \times 10^{+0})$ | $3.23 \times 10^{+1}$ |
| Mean    |      |      | $-3.07 \times 10^{+1}$ |      |      | $3.07 \times 10^{+1}$ |      |

As shown in Table 3, the proposed SaMTDE achieves better results on all 9 MTO test problems, and ultimately achieved a mean score $-3.07 \times 10^{+1}$. On the problems with complete intersection on their global optima, such as problems 1, 2 and 3, the performance improvement is very significant compared to that of the DE (a traditional single-task algorithm). On the other problems, such as problems 5, 6, 7 and 8, the SaMTDE can still achieve very good performance. Only on problems 4 and 9, the improvement is not very significant. For problem 4, it may due to the fact that the DE has optimized the two component tasks very well. Though the inter-task relatedness are very high across their search spaces ($Rs = 0.86$ as shown in Table 1), the lack of common features in their global optima disables the SaMTDE to further find better results at the final stage of optimization. For problem 9, knowledge transfer in the SaMTDE can not help the SaMTDE find better results due to the lack of inter-task relatedness ($Rs = 0.0$ as shown in Table 1). In summary, knowledge transfer in the proposed SaMTDE algorithm works efficiently.
In Figure 3, the averaged convergence curves of the SaMTDE over 30 runs are presented along with the curves of the DE algorithms. From the figures, we can see that the effectiveness of knowledge transfer in the SaMTDE occurs in around the 200 to the 500 generations. In problems 1 to 3, the significant performance improvement are again presented in a different view. For problem 9, we can see from the corresponding figure, knowledge transfer in the SaMTDE can hardly find better solutions for the two component tasks throughout the optimization process, because the tasks bear nearly no inter-task relatedness.

![Convergence curves for various problems](image)

**Figure 3.** Averaged convergence curves of DEMTO versus the DEMTO_STO on the 9 test problems from CEC’2017 Evolutionary Multi-Task Optimization Competition.

### 4.3.2. Comparison of SaMTDE with the SaMTPSO, the MFDE, SREMTO, the MFEA and the DE

In this experiment, the performance of the SaMTDE on the 9 popular MTO test problems from Table 1 are compared to that of several algorithms, including the SaMTPSO, the MFDE, the MFEA and the DE algorithm. Table 4 presents the means and bracketed standard deviations of the achieved FVs over 30 runs by these algorithms.
Table 4. Achieved FVs (mean, bracketed standard deviation) of the SaMTDE, the SaMTPSO, the MFDE, the SREMTO, the MFEA and the DE. The best results are shown in bold.

| Problem | Task | SaMTDE | SaMTPSO | MFDE | SREMTO | MFEA | DE |
|---------|------|--------|---------|-------|--------|------|----|
|         | Mean (Std) | Score | Mean (Std) | Score | Mean (Std) | Score | Mean (Std) | Score | Mean (Std) | Score | Mean (Std) | Score |
| 1       | T1    | 4.30 × 10⁻³ (8.55 × 10⁻³) | 7.11 × 10⁻² (5.99 × 10⁻²) | 1.01 × 10⁻³ (2.56 × 10⁻³) | 6.86 × 10⁻³ (8.00 × 10⁻³) | 3.62 × 10⁻¹ (5.55 × 10⁻²) | 5.78 × 10⁻¹ (3.11 × 10⁻¹) | 8.78 × 10⁻¹ |
|         | T2    | (2.78 × 10⁻¹) | −4.07 × 10⁻¹ | −2.67 × 10⁻¹ | −4.26 × 10⁻¹ | (2.47 × 10⁻¹) | (2.47 × 10⁻¹) | (3.57 × 10⁻¹) |
| 2       | T1    | 1.44 × 10⁻¹ (2.87 × 10⁻¹) | 3.18 × 10⁻¹ (5.62 × 10⁻¹) | 3.55 × 10⁻¹ (1.62 × 10⁻¹) | 3.48 × 10⁻¹ (7.22 × 10⁻²) | 4.73 × 10⁻¹ (7.09 × 10⁻¹) | 7.07 × 10⁻¹ (2.78 × 10⁻¹) | 4.66 × 10⁻¹ |
|         | T2    | (3.34 × 10⁻¹) | −6.26 × 10⁻¹ | 2.17 × 10⁻⁰ | −6.51 × 10⁻¹ | (2.41 × 10⁻¹) | (2.17 × 10⁻¹) | (4.86 × 10⁻¹) |
| 3       | T1    | 6.74 × 10⁻¹ (9.43 × 10⁻¹) | 5.73 × 10⁻¹ (7.89 × 10⁻¹) | 2.12 × 10⁻¹ (5.17 × 10⁻¹) | 2.11 × 10⁻¹ (2.99 × 10⁻¹) | 2.02 × 10⁻¹ (4.96 × 10⁻¹) | 1.54 × 10⁻¹ (2.12 × 10⁻¹) | 5.32 × 10⁻² |
|         | T2    | (3.61 × 10⁻¹) | −5.20 × 10⁻¹ | −7.46 × 10⁻¹ | 7.76 × 10⁻¹ | 5.89 × 10⁻¹ | 3.35 × 10⁻¹ | (3.81 × 10⁻²) |
| 4       | T1    | 2.59 × 10⁻² (7.77 × 10⁻²) | 3.60 × 10⁻² (1.07 × 10⁻¹) | 8.29 × 10⁻² (2.05 × 10⁻¹) | 2.58 × 10⁻² (7.82 × 10⁻²) | 5.67 × 10⁻² (9.64 × 10⁻²) | 4.98 × 10⁻¹ (2.37 × 10⁻²) | 2.03 × 10⁻¹ |
|         | T2    | (1.31 × 10⁻¹) | 2.82 × 10⁻¹ | −1.79 × 10⁻⁰ | −5.11 × 10⁻¹ | (8.92 × 10⁻¹) | (6.03 × 10⁻⁰) | (1.37 × 10⁻²) |
| 5       | T1    | 6.73 × 10⁻¹ (6.95 × 10⁻¹) | 1.75 × 10⁻¹ (5.10 × 10⁻¹) | 3.11 × 10⁻² (6.10 × 10⁻¹) | 2.33 × 10⁻¹ (2.25 × 10⁻¹) | 3.48 × 10⁻¹ (6.87 × 10⁻¹) | 2.92 × 10⁻¹ (2.80 × 10⁻¹) | 5.06 × 10⁻¹ |
|         | T2    | (1.11 × 10⁻¹) | −3.11 × 10⁻¹ | −8.08 × 10⁻⁰ | −4.51 × 10⁻⁰ | (1.23 × 10⁻¹) | (6.61 × 10⁻¹) | (4.51 × 10⁻¹) |
| 6       | T1    | 1.85 × 10⁻⁰ (1.06 × 10⁻⁰) | 4.36 × 10⁻¹ (1.25 × 10⁻¹) | 1.78 × 10⁻¹ (5.87 × 10⁻¹) | 3.49 × 10⁻⁰ (5.48 × 10⁻¹) | 2.00 × 10⁻¹ (4.86 × 10⁻¹) | 1.30 × 10⁻¹ (2.16 × 10⁻¹) | 2.38 × 10⁻¹ |
|         | T2    | (4.74 × 10⁻¹) | −3.52 × 10⁻¹ | −1.36 × 10⁻¹ | −4.41 × 10⁻¹ | (3.72 × 10⁻¹) | (2.16 × 10⁻¹) | (2.36 × 10⁻¹) |
| 7       | T1    | 1.89 × 10⁻² (6.60 × 10⁻²) | 2.55 × 10⁻² (9.00 × 10⁻²) | 8.10 × 10⁻¹ (4.28 × 10⁻¹) | 1.22 × 10⁻² (7.24 × 10⁻²) | 1.02 × 10⁻³ (7.74 × 10⁻³) | 3.67 × 10⁻¹ (3.29 × 10⁻³) | 5.47 × 10⁻¹ |
|         | T2    | (2.82 × 10⁻¹) | −2.57 × 10⁻¹ | −1.99 × 10⁻¹ | −2.70 × 10⁻¹ | (2.30 × 10⁻¹) | (2.76 × 10⁻²) | (1.01 × 10⁻²) |
| 8       | T1    | 1.72 × 10⁻¹ (1.74 × 10⁻¹) | 5.13 × 10⁻² (2.63 × 10⁻²) | 5.44 × 10⁻¹ (1.42 × 10⁻¹) | 1.46 × 10⁻¹ (4.13 × 10⁻¹) | 8.52 × 10⁻³ (9.84 × 10⁻⁴) | 7.41 × 10⁻¹ (3.82 × 10⁻¹) | 1.82 × 10⁻¹ |
|         | T2    | (5.96 × 10⁻¹) | −2.86 × 10⁻¹ | 1.42 × 10⁻⁰ | −5.96 × 10⁻¹ | (3.62 × 10⁻¹) | (1.88 × 10⁻¹) | (3.53 × 10⁻¹) |
| 9       | T1    | 2.47 × 10⁻² (8.54 × 10⁻³) | 5.36 × 10⁻² (2.17 × 10⁻²) | 9.67 × 10⁻¹ (2.30 × 10⁻¹) | 2.53 × 10⁻² (5.47 × 10⁻²) | 6.01 × 10⁻² (1.16 × 10⁻²) | 6.77 × 10⁻¹ (2.57 × 10⁻²) | 3.09 × 10⁻¹ |
|         | T2    | (9.89 × 10⁻²) | −3.28 × 10⁻¹ | −2.51 × 10⁻¹ | −1.27 × 10⁻¹ | (4.20 × 10⁻²) | (3.81 × 10⁻²) | (2.36 × 10⁻²) |
By comparing the performance difference between the considered algorithms, we can observe that the SaMTDE can achieve okay results compared to that of the other algorithms, and has achieved the best mean score $-3.4 \times 10^1$. Compared to the achieved mean score $-1.85 \times 10^1$ of the SaMTPSO, the SaMTDE has achieved better results. On problems 1 to 8, the SaMTDE can not achieve the best results, but the achieved results are still promising as presented in the table, especially considering the results of its single-task version (i.e., DE algorithm). The DE has the same mutation, crossover and selection operations as well as the same parameter settings; however, the DE can only achieve a mean score $2.51 \times 10^1$ while the SaMTDE has achieved mean score $-3.40 \times 10^1$; therefore, the SaMTDE shows future promise.

4.3.3. Solving the Weapon–Target Optimization problems

In this experiment, the proposed SaMTDE was applied to handle the MTO problems from Table 2, which are constructed by two different weapon–target optimization problems. To decode from the continuous search space of SaMTDE, this example employs a simple strategy to decode directly. For example, a decision variable $x$ in continue space is decoded for a WTA problem by $\text{round}(x)$, where function $\text{round}(\cdot)$ is an rounding function. To demonstrate the performance of the SaMTDE, the achieved results of the SaMTDE are compared with that of the SaMTPSO, the MFDE, the SREMTO, the MFEA, the DE and the JADE. By comparing with the DE and the JADE, we can observe the performance improvement of MTO in handling the WTA problems, which may show us another way to solve WTA problems more efficiently. The means and the bracketed standard deviations of the achieved FVs by every algorithm over 20 runs are presented in Table 5.

From Table 5, we can observe that the SaMTDE obtains the best mean score $-3.2 \times 10^1$ compared to all other algorithms. In 6 of the 9 MTO problems, the SaMTDE achieved the best scores. On the last 3 MTO problems, the SaMTDE has still achieved okay results, such as for problems 2, 6 and 9. Compared to the SaMTPSO, the performance of the SaMTDE is outstanding in all 9 MTO problems.
| Problem | Task | SaMCTDE | SaMTPSO | SFDE | SREMTO | MFEA | DE | JADE |
|---------|------|---------|---------|------|--------|------|----|------|
|         |      | Mean (Std) | Score | Mean (Std) | Score | Mean (Std) | Score | Mean (Std) | Score | Mean (Std) | Score |
| 1       | T1   | 5.29 × 10⁻¹ | -6.23 × 10⁻¹ | 5.29 × 10⁻¹ | -6.53 × 10⁻¹ | 5.29 × 10⁻¹ | -6.31 × 10⁻¹ | 5.29 × 10⁻¹ | -4.92 × 10⁻¹ | 5.31 × 10⁻¹ | 6.73 × 10⁻¹ |
|         | T2   | 3.12 × 10⁻¹ | 2.70 × 10⁻¹ | 3.12 × 10⁻¹ | 3.71 × 10⁻¹ | 3.12 × 10⁻¹ | 4.87 × 10⁻¹ | 3.12 × 10⁻¹ | -3.40 × 10⁻¹ | 3.12 × 10⁻¹ | 3.12 × 10⁻¹ |
| 2       | T1   | 8.98 × 10⁻¹ | 5.11 × 10⁻¹ | 8.98 × 10⁻¹ | 5.11 × 10⁻¹ | 5.11 × 10⁻¹ | 5.11 × 10⁻¹ | 5.11 × 10⁻¹ | 5.11 × 10⁻¹ | 5.11 × 10⁻¹ | 5.11 × 10⁻¹ |
|         | T2   | 8.13 × 10⁻¹ | 9.05 × 10⁻¹ | 8.13 × 10⁻¹ | 9.05 × 10⁻¹ | 9.05 × 10⁻¹ | 9.05 × 10⁻¹ | 9.05 × 10⁻¹ | 9.05 × 10⁻¹ | 9.05 × 10⁻¹ | 9.05 × 10⁻¹ |
| 3       | T1   | 4.62 × 10⁻¹ | -5.28 × 10⁻¹ | 4.62 × 10⁻¹ | -5.28 × 10⁻¹ | 4.62 × 10⁻¹ | -5.28 × 10⁻¹ | 4.62 × 10⁻¹ | -5.28 × 10⁻¹ | 4.62 × 10⁻¹ | -5.28 × 10⁻¹ |
|         | T2   | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ | 9.62 × 10⁻¹ |

Table 5. Achieved FVs (mean, bracketed standard deviation) of the SaMCTDE, the SaMTPSO, the MFDE, the SREMTO, the MFEA, the DE and the JADE. The best results are shown in bold.

Mean: -3.21 × 10⁻¹ | 3.55 × 10⁻¹ | -6.05 × 10⁻¹ | 3.41 × 10⁻¹ | 2.41 × 10⁻¹ | 2.76 × 10⁻¹ | -6.68 × 10⁻¹ | -2.88 × 10⁻¹
5. Conclusions and Future Work

In this paper, an efficient SaMTDE algorithm is proposed by introducing the knowledge transfer adaptation strategy and the focus strategy of the SaMTPSO into a traditional DE algorithm. Meanwhile, a novel knowledge incorporation strategy was devised to fit into the evolution structure of the SaMTDE better. Further, the SaMTDE was employed to handle WTA problems for the first time in the field of EMTO. Experiments were conducted on two test suites, including a popular MTO benchmark and a WTA-MTO benchmark. Experimental results have shown that knowledge transfer in the SaMTDE is effective and efficient, and EMTO techniques have potentials in solving WTA problems.

In the future, more knowledge incorporation strategies may be explored in further improving the performance of the SaMTDE. Further, we may apply the SaMTDE into the application of many other real-world problems, such as the large-scale optimization problems [34] and complicated non-convex optimization tasks [12,35,36].

Author Contributions: Conceptualization, X.Z. and Y.L.; methodology, X.Z. and N.L.; software, X.Z. and T.W.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z. and D.Z.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 62076204), the National Natural Science Foundation of Shaanxi Province (Grant Nos. 2018JQ6003 and 2018JQ6030) and the China Postdoctoral Science Foundation (Grant Nos. 2017M613204 and 2017M623246).

Data Availability Statement: All data generated or analyzed during this study are included in this published article.

Acknowledgments: Thanks are due to Deiyun Zhou for his assistance in writing the draft of this paper, and Jiao Shi for the funding support.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations
The following abbreviations are used in this manuscript:

- **WTA**: Weapon–Target Assignment Problem
- **MTO**: Multi-task Optimization
- **EMTO**: Evolutionary Multi-task Optimization
- **SREMTO**: Self-regulated Evolutionary Multi-task Optimization
- **SaMTPSO**: Self-adaptive Multitask Particle Swarm Optimization
- **MFEA**: Multifactorial Evolutionary Algorithm
- **MFDE**: Multifactorial Differential Evolution
- **DE**: Differential Evolution
- **JaDE**: Adaptive Differential Evolution with Optional External Archive

Appendix A

In the following tables, the details of the nine WTA-MTO test problems in Table 2 are presented, including the destructive value of every target and the probability of every weapon.
Table A1. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 1 in Table 2. Each task is a WTA problem.

| Weapons | Value |
|---------|-------|
| 1       | 0.96  |
| 2       | 0.55  |
| 3       | 0.14  |
| 4       | 0.15  |
| 5       | 0.26  |

| T₁ Targets |
|------------|
| 1          | 0.13  |
| 2          | 0.57  |
| 3          | 0.47  |
| 4          | 0.01  |
| 5          | 0.34  |

| Weapons | Value |
|---------|-------|
| 1       | 0.84  |
| 2       | 0.25  |
| 3       | 0.81  |
| 4       | 0.24  |
| 5       | 0.93  |

| T₂ Targets |
|------------|
| 1          | 0.16  |
| 2          | 0.79  |
| 3          | 0.31  |
| 4          | 0.53  |
| 5          | 0.17  |

Table A2. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 2 in Table 2. Each task is a WTA problem.

| Weapons | Value |
|---------|-------|
| 1       | 0.26  |
| 2       | 0.76  |
| 3       | 0.61  |
| 4       | 0.08  |
| 5       | 0.43  |

| T₁ Targets |
|------------|
| 1          | 0.59  |
| 2          | 0.64  |
| 3          | 0.89  |
| 4          | 0.29  |
| 5          | 0.47  |

| Weapons | Value |
|---------|-------|
| 1       | 0.49  |
| 2       | 0.42  |
| 3       | 0.35  |
| 4       | 0.11  |
| 5       | 0.27  |

| T₂ Targets |
|------------|
| 1          | 0.51  |
| 2          | 0.55  |
| 3          | 0.82  |
| 4          | 0.89  |
| 5          | 0.89  |

Table A3. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 4 in Table 2. Each task is a WTA problem.

| Weapons | Value |
|---------|-------|
| 1       | 0.36  |
| 2       | 0.03  |
| 3       | 0.22  |
| 4       | 0.71  |
| 5       | 0.55  |

| T₁ Targets |
|------------|
| 1          | 0.49  |
| 2          | 0.74  |
| 3          | 0.10  |
| 4          | 0.97  |
| 5          | 0.74  |

| Weapons | Value |
|---------|-------|
| 1       | 0.01  |
| 2       | 0.68  |
| 3       | 0.49  |
| 4       | 0.03  |
| 5       | 0.11  |

| T₂ Targets |
|------------|
| 1          | 0.54  |
| 2          | 0.63  |
| 3          | 0.59  |
| 4          | 0.23  |
| 5          | 0.70  |
Table A4. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 3 in Table 2. Each task is a WTA problem.

| Weapons | Value |
|---------|-------|
| 1   | 0.69  |
| 2   | 0.67  |
| 3   | 0.62  |
| 4   | 0.66  |
| 5   | 0.64  |
| 6   | 0.68  |
| 7   | 0.70  |
| 8   | 0.68  |
| 9   | 0.70  |
| 10  | 0.70  |
| 11  | 0.70  |
| 12  | 0.70  |
| 13  | 0.70  |
| 14  | 0.70  |
| 15  | 0.70  |
| 16  | 0.70  |
| 17  | 0.70  |
| 18  | 0.70  |
| 19  | 0.70  |
| 20  | 0.70  |

| Targets T1 |
|-----------|
| 1   | 0.40  |
| 2   | 0.41  |
| 3   | 0.41  |
| 4   | 0.41  |
| 5   | 0.40  |
| 6   | 0.41  |
| 7   | 0.41  |
| 8   | 0.41  |
| 9   | 0.41  |
| 10  | 0.41  |
| 11  | 0.41  |
| 12  | 0.41  |
| 13  | 0.41  |
| 14  | 0.41  |
| 15  | 0.41  |
| 16  | 0.41  |
| 17  | 0.41  |
| 18  | 0.41  |
| 19  | 0.41  |
| 20  | 0.41  |

| Targets T2 |
|-----------|
| 1   | 0.40  |
| 2   | 0.41  |
| 3   | 0.41  |
| 4   | 0.41  |
| 5   | 0.41  |
| 6   | 0.41  |
| 7   | 0.41  |
| 8   | 0.41  |
| 9   | 0.41  |
| 10  | 0.41  |
| 11  | 0.41  |
| 12  | 0.41  |
| 13  | 0.41  |
| 14  | 0.41  |
| 15  | 0.41  |
| 16  | 0.41  |
| 17  | 0.41  |
| 18  | 0.41  |
| 19  | 0.41  |
| 20  | 0.41  |
Table A5. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 5 in Table 2. Each task is a WTA problem.

| Tasks | Targets | Value |
|-------|---------|-------|
|       |         |       |
| T1    |         |       |
| 1     | 0.65    | 0.81  | 0.13  | 0.22  | 0.91  | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | 0.74  |
| 2     | 0.38    | 0.35   | 0.60   | 0.05  | 0.92  | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | 0.13  |
| 3     | 0.96    | 0.42   | 0.48   | 0.17   | 0.49  | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | 0.14  |
| 4     | 0.18    | 0.60   | 0.18   | 0.27   | 0.35  | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | 0.47  |
| 5     | 0.24    | 0.94   | 0.33   | 0.08   | 0.06  | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | -     | 0.26  |
| T2    |         |       |       |
| 1     | 0.83    | 0.87   | 0.15   | 0.91   | 0.75  | 0.51  | 0.12  | 0.48  | 0.63  | 0.78  | 0.39  | 0.23  | 0.23  | 0.80  | 0.58  | 0.24  | 0.16  | 0.79  |
| 2     | 1.00    | 0.61   | 0.73   | 0.79   | 0.80  | 0.91  | 0.04  | 0.31  | 0.47  | 0.65  | 0.25  | 0.99  | 0.47  | 0.49  | 0.54  | 0.87  | 0.85  | 0.07  | 0.67  |
| 3     | 0.47    | 0.74   | 0.90   | 0.63   | 0.93  | 0.29  | 0.63  | 0.27  | 0.34  | 0.33  | 0.99  | 0.59  | 0.59  | 0.49  | 0.55  | 0.34  | 0.93  | 0.59  |
| 4     | 0.71    | 0.03   | 0.52   | 0.35   | 0.45  | 0.45  | 0.35  | 0.41  | 0.07  | 0.77  | 0.63  | 0.20  | 0.68  | 0.84  | 0.39  | 0.77  | 0.88  | 0.01  | 0.45  |
| 5     | 0.05    | 0.20   | 0.87   | 0.15   | 0.91   | 0.75  | 0.51  | 0.12  | 0.48  | 0.63  | 0.78  | 0.39  | 0.23  | 0.23  | 0.80  | 0.58  | 0.24  | 0.16  | 0.79  |
| 6     | 0.44    | 0.74   | 0.90   | 0.63   | 0.93  | 0.29  | 0.63  | 0.27  | 0.34  | 0.33  | 0.99  | 0.59  | 0.59  | 0.49  | 0.55  | 0.34  | 0.93  | 0.59  |
| 7     | 0.31    | 0.61   | 0.24   | 0.43   | 0.71   | 0.33  | 0.18  | 0.63  | 0.22  | 0.69  | 0.17  | 0.82  | 0.28  | 0.82  | 0.71  | 0.09  | 0.26  | 0.63  | 0.74  |
| 8     | 0.83    | 0.74   | 0.74   | 0.44   | 0.71   | 0.84  | 0.40  | 0.41  | 0.77  | 0.62  | 0.44  | 0.48  | 0.05  | 0.59  | 0.06  | 0.40  | 0.72  | 0.30  | 0.37  | 0.41  |
| 9     | 0.27    | 0.37   | 0.81   | 0.53   | 0.38   | 0.38  | 0.95  | 0.28  | 0.17  | 0.84  | 0.01  | 0.63  | 0.41  | 0.35  | 0.32  | 0.42  | 0.95  | 0.03  | 0.93  | 0.45  |
| 10    | 0.62    | 0.26   | 0.34   | 0.41   | 0.35   | 0.58  | 0.92  | 0.10  | 0.37  | 0.03  | 0.69  | 0.55  | 0.22  | 0.90  | 0.65  | 0.99  | 0.91  | 0.24  | 0.99  | 0.68  |
| 11    | 0.52    | 0.84   | 0.78   | 0.08   | 0.22   | 0.49   | 0.36  | 0.32  | 0.46  | 0.72  | 0.23  | 0.85  | 0.65  | 0.91  | 0.77  | 0.40  | 0.13  | 0.69  | 0.34  | 0.87  |
| 12    | 0.49    | 0.98   | 0.31   | 0.27   | 0.33   | 0.84  | 0.21  | 0.22  | 0.35  | 0.33  | 0.63  | 0.93  | 0.90  | 0.96  | 0.52  | 0.99  | 0.46  | 0.43  | 0.68  | 0.27  |
| 13    | 0.60    | 0.36   | 0.36   | 0.24   | 0.46   | 0.24  | 0.54  | 0.32  | 0.61  | 0.56  | 0.17  | 0.66  | 0.57  | 0.42  | 0.82  | 0.39  | 0.79  | 0.71  | 0.72  | 0.88  |
| 14    | 0.02    | 0.43   | 0.46   | 0.89   | 0.83   | 0.52  | 0.75  | 0.38  | 0.78  | 0.96  | 0.36  | 0.85  | 0.28  | 0.15  | 0.19  | 0.06  | 0.75  | 0.88  | 0.22  | 0.17  |
| Weapons | Value |
|---------|-------|
| 1       | 0.18  |
| 2       | 0.28  |
| 3       | 0.04  |
| 4       | 0.25  |
| 5       | 0.49  |
| 6       | 0.17  |
| 7       | 0.68  |
| 8       | 0.33  |
| 9       | 0.52  |
| 10      | 0.46  |
| 11      | 0.89  |
| 12      | 0.53  |
| 13      | 0.54  |
| 14      | 0.55  |
| 15      | 0.54  |
| 16      | 0.48  |
| 17      | 0.88  |
| 18      | 1.00  |
| 19      | 0.93  |
| 20      | 0.92  |

Table A6. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 6 in Table 2. Each task is a WTA problem.
Table A7. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 7 in Table 2. Each task is a WTA problem.

|    | Weapons | Value |
|----|---------|-------|
|    | 1  | 2  | 3  | 4  | 5  |     |
| **T1** | Targets |     |
| 1  | 0.95 | 0.48 | 0.87 | 1.00 | 0.06 | 0.73 |
| 2  | 0.32 | 0.68 | 0.86 | 0.87 | 0.43 | 0.77 |
| 3  | 0.58 | 0.40 | 0.76 | 0.95 | 0.19 | 0.15 |
| 4  | 0.19 | 0.15 | 0.17 | 0.25 | 0.33 | 0.80 |
| 5  | 0.39 | 0.56 | 0.08 | 0.05 | 0.94 | 0.17 |
| 6  | 1.00 | 0.95 | 0.51 | 0.83 | 0.85 | 0.72 |
| 7  | 0.34 | 0.96 | 0.09 | 0.69 | 0.26 | 0.08 |
| 8  | 0.91 | 0.99 | 0.44 | 0.60 | 0.38 | 0.39 |
| 9  | 0.70 | 0.84 | 0.31 | 0.21 | 0.73 | 0.82 |
| 10 | 0.03 | 0.65 | 0.38 | 0.54 | 0.78 | 0.53 |
| **T2** | Targets |     |
| 1  | 0.70 | 0.79 | 0.07 | 0.06 | 0.64 | 0.01 |
| 2  | 0.69 | 0.41 | 0.52 | 0.86 | 0.77 | 0.22 |
| 3  | 0.85 | 0.13 | 0.20 | 0.07 | 0.68 | 0.07 |
| 4  | 0.29 | 0.16 | 0.30 | 0.88 | 0.49 | 0.71 |
| 5  | 0.65 | 0.82 | 0.98 | 0.39 | 0.81 | 0.88 |
| 6  | 0.30 | 0.03 | 0.17 | 0.44 | 0.08 | 0.28 |
| 7  | 0.94 | 0.54 | 0.78 | 0.35 | 0.03 | 0.69 |
| 8  | 0.78 | 0.82 | 0.34 | 0.03 | 0.43 | 0.14 |
| 9  | 0.83 | 0.95 | 0.47 | 0.73 | 0.14 | 0.37 |
| 10 | 0.74 | 0.46 | 0.10 | 0.35 | 0.81 | 0.84 |
Table A8. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 8 in Table 2. Each task is a WTA problem.

|     | 1   | 2   | 3   | 4   | 5   | Value |
|-----|-----|-----|-----|-----|-----|-------|
|     | 0.98| 0.46| 0.19| 0.28| 0.70| 0.93  |
| 2   | 0.33| 0.63| 0.58| 0.35| 0.39| 0.23  |
| 3   | 1.00| 0.67| 0.55| 0.08| 0.70| 0.80  |
| 4   | 0.42| 0.15| 0.26| 0.76| 0.45| 0.63  |
| 5   | 0.58| 0.58| 0.08| 0.48| 0.35| 0.07  |
| 6   | 0.08| 0.86| 0.92| 0.31| 0.37| 0.94  |
| 7   | 0.29| 0.64| 0.32| 0.73| 0.49| 0.84  |
| 8   | 0.56| 0.94| 0.99| 0.13| 0.22| 0.25  |
| 9   | 0.25| 0.41| 0.11| 0.99| 0.69| 0.66  |
| T1  | 10  | 0.96| 0.10| 0.94| 0.43| 0.40  |
|     | 11  | 0.52| 0.66| 1.00| 0.83| 0.99  |
|     | 12  | 0.12| 0.59| 0.95| 0.76| 0.75  |
|     | 13  | 0.73| 0.22| 0.53| 0.98| 0.42  |
|     | 14  | 0.43| 0.98| 0.29| 0.62| 0.53  |
|     | 15  | 0.78| 0.35| 0.99| 0.19| 0.57  |
|     | 16  | 0.72| 0.80| 0.39| 0.87| 0.79  |
|     | 17  | 0.13| 0.25| 0.79| 0.88| 0.90  |
|     | 18  | 0.81| 0.07| 0.54| 0.50| 0.51  |
|     | 19  | 0.33| 0.25| 0.31| 0.72| 0.17  |
|     | 20  | 0.83| 0.56| 0.00| 0.85| 0.55  |

|     | 1   | 2   | 3   | 4   | 5   | Value |
|-----|-----|-----|-----|-----|-----|-------|
|     | 0.76| 0.89| 0.91| 0.29| 0.65| 0.42  |
| 2   | 0.53| 0.11| 0.36| 0.90| 0.42| 0.90  |
| 3   | 0.84| 0.70| 0.07| 0.84| 0.45| 0.18  |
| 4   | 0.31| 0.42| 0.18| 0.15| 0.93| 0.83  |
| 5   | 0.66| 0.97| 0.42| 0.20| 0.68| 0.01  |
| 6   | 0.37| 0.60| 0.46| 0.35| 0.31| 0.99  |
| 7   | 0.66| 0.54| 0.28| 0.27| 0.98| 0.03  |
| 8   | 0.59| 0.87| 0.83| 0.59| 0.93| 0.60  |
| 9   | 0.28| 0.81| 0.46| 0.58| 0.82| 0.59  |
| T2  | 10  | 0.55| 0.45| 0.65| 0.42| 0.01  |
|     | 11  | 0.13| 0.01| 0.92| 0.54| 0.81  |
|     | 12  | 0.45| 0.13| 0.18| 0.83| 0.86  |
|     | 13  | 0.57| 0.51| 0.57| 0.82| 0.79  |
|     | 14  | 0.13| 0.46| 0.72| 0.35| 0.79  |
|     | 15  | 0.40| 0.62| 0.15| 0.79| 0.11  |
|     | 16  | 0.84| 0.01| 0.87| 0.25| 0.25  |
|     | 17  | 0.88| 0.56| 0.83| 0.06| 0.04  |
|     | 18  | 0.11| 0.67| 0.44| 0.05| 0.85  |
|     | 19  | 0.98| 0.05| 0.80| 0.93| 0.49  |
|     | 20  | 0.10| 0.80| 0.78| 0.22| 0.63  |
Table A9. Details of the destructive value of every target and the probability of kill for every weapon in the two component tasks of problem 9 in Table 2. Each task is a WTA problem.

| Weapons | Value |
|---------|-------|
| T1      | Targets | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1  | 0.81 | 0.01 | 0.11 | 0.08 | 0.01 | 0.36 | 0.93 | 0.53 | 0.12 | 0.43 | 0.36 |
| 2  | 0.49 | 0.29 | 0.72 | 0.73 | 0.31 | 0.69 | 0.63 | 0.27 | 0.86 | 0.97 | 0.19 |
| 3  | 0.26 | 0.51 | 0.61 | 0.12 | 0.77 | 0.79 | 0.23 | 0.24 | 0.19 | 0.11 | 0.29 |
| 4  | 0.94 | 0.60 | 0.92 | 0.22 | 0.98 | 0.64 | 0.39 | 0.35 | 0.86 | 0.90 | 0.05 |
| 5  | 0.43 | 0.45 | 0.23 | 0.61 | 0.78 | 0.48 | 0.61 | 0.17 | 0.35 | 0.76 | 0.77 |
| 6  | 0.80 | 0.53 | 0.06 | 0.44 | 0.49 | 0.82 | 0.05 | 0.93 | 0.92 | 0.64 | 0.55 |
| 7  | 0.04 | 0.79 | 0.14 | 0.60 | 0.55 | 0.22 | 0.28 | 0.58 | 0.36 | 0.13 | 0.10 |
| 8  | 0.75 | 0.53 | 0.56 | 0.93 | 0.12 | 0.45 | 0.77 | 0.80 | 0.17 | 0.11 | 0.04 |
| 9  | 10  | 0.04 | 0.49 | 0.15 | 0.72 | 0.01 | 0.65 | 0.49 | 0.68 | 0.37 | 0.41 |
| T2      | Targets | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1  | 0.74 | 0.69 | 0.61 | 0.85 | 0.52 | 0.46 | 0.13 | 0.31 | 0.70 | 0.51 | 0.91 |
| 2  | 0.15 | 0.87 | 0.47 | 0.53 | 0.61 | 0.60 | 0.16 | 0.23 | 0.36 | 0.76 | 0.38 |
| 3  | 0.63 | 0.56 | 0.16 | 0.81 | 0.34 | 0.10 | 0.08 | 0.57 | 0.12 | 0.45 | 0.92 |
| 4  | 0.85 | 0.72 | 0.03 | 0.38 | 0.64 | 0.03 | 0.63 | 0.31 | 0.59 | 0.72 | 0.72 |
| 5  | 0.48 | 0.00 | 0.67 | 0.70 | 0.42 | 0.45 | 0.59 | 0.74 | 0.68 | 0.37 | 0.72 |
| 6  | 0.70 | 0.87 | 0.01 | 0.50 | 0.03 | 0.72 | 0.45 | 0.28 | 0.89 | 0.35 | 0.75 |
| 7  | 0.71 | 0.65 | 0.18 | 0.33 | 0.91 | 0.91 | 0.51 | 0.12 | 0.33 | 0.06 | 0.38 |
| 8  | 0.82 | 0.59 | 0.03 | 0.62 | 0.36 | 0.35 | 0.43 | 0.36 | 0.94 | 0.26 | 0.89 |
| 9  | 0.67 | 0.09 | 0.62 | 0.57 | 0.47 | 0.58 | 0.46 | 0.09 | 0.85 | 0.94 | 0.30 |
| 10 | 0.68 | 0.01 | 0.25 | 0.27 | 0.30 | 0.69 | 0.51 | 0.10 | 0.79 | 0.65 | 0.32 |
| 11 | 0.34 | 0.26 | 0.81 | 0.75 | 0.46 | 0.80 | 0.69 | 0.89 | 0.65 | 0.56 | 0.97 |
| 12 | 0.04 | 0.19 | 0.09 | 0.29 | 0.89 | 0.24 | 0.65 | 0.14 | 0.45 | 0.58 | 0.18 |
| 13 | 0.94 | 0.18 | 0.28 | 0.12 | 0.32 | 0.66 | 0.24 | 0.19 | 1.00 | 0.53 | 0.30 |
| 14 | 0.65 | 0.61 | 0.48 | 0.75 | 0.01 | 0.62 | 0.27 | 0.43 | 0.03 | 0.87 | 0.15 |
| 15 | 0.83 | 0.62 | 0.27 | 0.34 | 0.63 | 0.72 | 0.93 | 0.57 | 0.87 | 0.45 | 0.59 |
| 16 | 0.71 | 0.62 | 0.60 | 0.47 | 0.94 | 0.62 | 0.99 | 0.72 | 0.43 | 0.96 | 0.65 |
| 17 | 0.41 | 0.59 | 0.35 | 0.69 | 0.14 | 0.28 | 0.10 | 0.34 | 0.16 | 0.14 | 0.20 |
| 18 | 0.79 | 0.22 | 0.21 | 0.04 | 0.73 | 0.01 | 0.02 | 0.95 | 0.06 | 0.33 | 0.71 |
| 19 | 0.81 | 0.91 | 0.09 | 0.87 | 0.84 | 0.53 | 0.54 | 0.49 | 0.25 | 0.80 | 0.13 |
| 20 | 0.17 | 0.69 | 0.83 | 0.18 | 0.12 | 0.69 | 0.51 | 0.82 | 0.93 | 0.93 | 0.07 |

References
1. Gupta, A.; Ong, Y.S.; Feng, L. Multifactorial Evolution: Toward Evolutionary Multitasking. IEEE Trans. Evol. Comput. 2016, 20, 343–357, doi:10.1109/TEVC.2015.2458037.
2. Zheng, X.; Qin, A.K.; Gong, M.; Zhou, D. Self-regulated evolutionary multitask optimization. IEEE Trans. Evol. Comput. 2019, 24, 16–28.
3. Liaw, R.T.; Ting, C.K. Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation, San Sebastián, Spain, 5–8 June 2017; pp. 2266–2273.
4. Li, G.; Zhang, Q.; Gao, W. Multipopulation evolution framework for multifactorial optimization. In Proceedings of the 2018 Genetic and Evolutionary Computation Conference, Kyoto, Japan, 15–19 July 2018; pp. 215–216.
5. Bali, K.K.; Gupta, A.; Feng, L.; Ong, Y.S.; Siew, T.P. Linearized domain adaptation in evolutionary multitasking. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), Sebastián, Spain, 5–8 June 2017; pp. 1295–1302.
6. Chen, Q.; Ma, X.; Sun, Y.; Zhu, Z. Adaptive memetic algorithm based evolutionary multi-tasking single-objective optimization. In Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning, Shenzhen, China, 10–13 November 2017; pp. 462–472.

7. Wen, Y.W.; Ting, C.K. Parting ways and reallocating resources in evolutionary multitasking. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation, San Sebastián, Spain, 5–8 June 2017; pp. 2404–2411.

8. Tang, J.; Chen, Y.; Deng, Z.; Xiang, Y.; Joy, C.P. A Group-based Approach to Improve Multifactorial Evolutionary Algorithm. In Proceedings of the Twenty-Seventh Joint International Conference on Artificial Intelligence 2018, Stockholm, Sweden, 13-19 July 2018; pp. 3870–3876.

9. Sampath, L.; Gupta, A.; Ong, Y.S.; Gooi, H. Evolutionary Multitasking to Support Optimal Power Flow under Rapid Load Variations. South. Power Syst. Technol. China 2018. Available online: https://www.researchgate.net/publication/322518905 (accessed on January 2018).

10. Sagarna, R.; Ong, Y.S. Concurrently searching branches in software tests generation through multitask evolution. In Proceedings of the 2016 IEEE Symposium on Computational Intelligence, Athens, Greece, 6–9 December 2016; pp. 1–8.

11. Bao, L.; Qi, Y.; Shen, M.; Bu, X.; Yu, J.; Li, Q.; Chen, P. An Evolutionary Multitasking Algorithm for Cloud Computing Service Composition. In Proceedings of the 2018 IEEE World Congress on Services, San Francisco, CA, USA, 2–7 July 2018; pp. 130–144.

12. Zhang, B.; Qin, A.K.; Sellis, T. Evolutionary Feature Subspaces Generation for Ensemble Classification. In Proceedings of the 2018 Genetic and Evolutionary Computation Conference, Kyoto, Japan, 15–19 July 2018; pp. 577–584.

13. Zheng, X.; Zhou, D.; Li, N.; Lei, Y.; Wu, T.; Gong, M. Self-adaptive Multi-task Particle Swarm Optimization. arXiv 2021, 11, 42, 276–280.

14. Lu, H.; Zhang, H.; Zhang, X.; Han, R. An improved genetic algorithm for target assignment, optimization of naval fleet air defense. In Proceedings of the 2006 6th World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; Volume 1, pp. 3401–3405.

15. Shang, G. Solving weapon-target assignment problems by a new ant colony algorithm. In Proceedings of the 2008 International Symposium on Computational Intelligence and Design, Hangzhou, China, 12–13 December 2008; Volume 1, pp. 221–224.

16. Karasakal, O. Air defense missile-target allocation models for a naval task group. Comput. Oper. Res. 2008, 35, 1759–1770.

17. Bogdanowicz, Z.R. Advanced input generating algorithm for effect-based weapon–target pairing optimization. IEEE Trans. Man Cybern.-Part A Syst. Hum. 2011, 42, 276–280.

18. Swersky, K.; Snoek, J.; Adams, R.P. Multi-task bayesian optimization. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NA, USA, 5–8 December 2013; pp. 2004–2012.

19. Pelikan, M.; Goldberg, D.E.; Cantú-Paz, E. BOA: The Bayesian optimization algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Orlando, FL, USA, 13–17 July 1999; Volume 1, pp. 525–532.

20. Frazier, P.I. A tutorial on Bayesian optimization. arXiv 2018, arXiv:1807.02811.

21. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. In Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS 2011), Granada, Spain, 12–14 December 2011; Volume 24.

22. Zhou, L.; Feng, L.; Tan, K.C.; Zhong, J.; Zhu, Z.; Liu, K.; Chen, C. Toward adaptive knowledge transfer in multifactorial evolutionary computation. IEEE Trans. Cybern. 2020, 51, 2563–2576.

23. Tang, Z.; Gong, M.; Wu, Y.; Liu, W.; Xie, Y. Regularized Evolutionary Multi-Task Optimization: Learning to Inter-Task Transfer in Aligned Subspace. IEEE Trans. Evol. Comput. 2020, 25, 262–276.

24. Liaw, R.T.; Ting, C.K. Evolutionary manytasking optimization based on symbiosis in bioecoenosis. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.

25. Zhou, L.; Feng, L.; Liu, K.; Chen, C.; Deng, S.; Xiang, T.; Jiang, S. Towards Effective Mutation for Knowledge Transfer in Multifactorial Differential Evolution. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019.

26. Chen, Y.; Zhong, J.; Feng, L.; Zhang, J. An adaptive archive-based evolutionary framework for many-task optimization. IEEE Trans. Emerg. Top. Comput. Intell. 2019, 4, 369–384.

27. Song, H.; Qin, A.K.; Tsai, P.W.; Liang, J.J. Multitasking multi-swarm optimization. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 1937–1944.

28. Cheng, M.Y.; Gupta, A.; Ong, Y.S.; Ni, Z.W. Coevolutionary multitasking for concurrent global optimization: With case studies in complex engineering design. Eng. Appl. Artif. Intell. 2017, 64, 13–24.

29. Da, B.; Gupta, A.; Ong, Y.S.; Feng, L. Evolutionary multitasking across single and multi-objective formulations for improved problem solving. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, Canada, 24–29 July 2016; pp. 1695–1701.

30. Feng, L.; Zhou, W.; Zhou, L.; Jiang, S.; Zhong, J.; Da, B.; Zhu, Z.; Wang, Y. An empirical study of multifactorial PSO and multifactorial DE. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastián, Spain, 5–8 June 2017; pp. 921–928.

31. Feng, L.; Ong, Y.S.; Gupta, A. Genetic algorithm and its advances in embracing memetics. In Evolutionary and Swarm Intelligence Algorithms; Springer: Berlin/Heidelberg, Germany, 2019; pp. 61–84.

32. Zhang, J.; Sanderson, A.C. JADE: Adaptive differential evolution with optional external archive. IEEE Trans. Evol. Comput. 2009, 13, 945–958.
33. Da, B.; Ong, Y.S.; Feng, L.; Qin, A.K.; Gupta, A.; Zhu, Z.; Ting, C.K.; Tang, K.; Yao, X. Evolutionary Multitasking for Single-objective Continuous Optimization: Benchmark Problems, Performance Metric, and Baseline Results. *arXiv* 2017. arXiv:1706.03470.

34. Kazimipour, B.; Li, X.; Qin, A.K. Initialization methods for large scale global optimization. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 2750–2757.

35. Qu, B.Y.; Lang, B.F.; Liang, J.J.; Qin, A.K.; Crisalle, O.D. Two-hidden-layer extreme learning machine for regression and classification. *Neurocomputing* 2016, 175, 826–834.

36. Qin, A.K.; Suganthan, P.N.; Loog, M. Uncorrelated heteroscedastic LDA based on the weighted pairwise Chernoff criterion. *Pattern Recognit.* 2005, 38, 613–616.