Mathematical and Computational Applications

Article

Performance Analysis of Multi-Objective Simulated Annealing Based on Decomposition

Manuel Vargas-Martínez [1,*], Nelson Rangel-Valdez [2], Eduardo Fernández [3], Claudia Gómez-Santillán [1] and María Lucila Morales-Rodríguez [1]

[1] División de Estudios de Posgrado e Investigación Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Ciudad Madero 89440, Mexico
[2] CONACyT Research Fellow at Graduate Program Division, Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Ciudad Madero 89440, Mexico
[3] Facultad de Contaduría y Administración, Universidad Autónoma de Coahuila, Saltillo 26200, Mexico
[*] Correspondence: jmvm100@gmail.com

Abstract: Simulated annealing is a metaheuristic that balances exploration and exploitation to solve global optimization problems. However, to deal with multi- and many-objective optimization problems, this balance needs to be improved due to diverse factors such as the number of objectives. To deal with this issue, this work proposes MOSA/D, a hybrid framework for multi-objective simulated annealing based on decomposition and evolutionary perturbation functions. According to the literature, the decomposition strategy allows diversity in a population while evolutionary perturbations add convergence toward the Pareto front; however, a question should be asked: What is the effect of such components when included as part of a multi-objective simulated annealing design? Hence, this work studies the performance of the MOSA/D framework considering in its implementation two widely used perturbation operators: classical genetic operators and differential evolution. The proposed algorithms are MOSA/D-CGO, based on classical genetic operators, and MOSA/D-DE, based on differential evolution operators. The main contribution of this work is the performance analysis of MOSA/D using both perturbation operators and identifying the one most suitable for the framework. The approaches were tested using DTLZ on two and three objectives and CEC2009 benchmarks on two, three, five, and ten objectives; the performance analysis considered diversity and convergence measured through the hypervolume (HV) and inverted generational distance (IGD) indicators. The results pointed out that there is a promising improvement in performance in favor of MOSA/D-DE.

Keywords: multi-objective; simulated annealing; decomposition; differential evolution

check for updates

Citation: Vargas-Martínez, M.; Rangel-Valdez, N.; Fernández, E.; Gómez-Santillán, C.; Morales-Rodríguez, M.L. Performance Analysis of Multi-Objective Simulated Annealing Based on Decomposition. *Math. Comput. Appl.* **2023**, *28*, 38. https://doi.org/10.3390/mca28020038

Academic Editor: Efrén Mezura-Montes

Received: 30 January 2023
Revised: 28 February 2023
Accepted: 2 March 2023
Published: 8 March 2023

1. Introduction

In the real world, optimization problems have multiple objectives that conflict. In other words, when we attempt to optimize one of these objectives, the rest is affected negatively. These problems are called Multi-Objective Optimization Problems (MOPs). A MOP is defined as follows:

$$\min F(x) = [f_1(x), f_2(x), \dots, f_m(x)], \text{subject to } x \in \Omega, \quad (1)$$

where Ω represents the decision space and $x = x_1, x_2, \dots, x_n$ represents the decision vector (solution). $f_i(x)$ denotes the ith objective obtained from the decision vector x. Evolutionary multi-objective optimization (EMO) is a research area concerned with ways to solve MOPs. The EMO algorithms perform well when the MOPs have two or three objectives. In EMO, the main algorithms proposed are based on Pareto dominance. In particular, these algorithms have been successfully detecting well-converged and well-diversified non-dominated solutions [1]. However, the number of real-world optimization problems is many times more than three. MOPs with four or more objectives are often referred to as many-objective optimization problems (MaOPs) [1,2]. Pareto dominance has been the key to the success of EMO algorithms in MOPs with two and three objectives, but the results are different in the case of MaOPs. The reason is that the selective pressure toward the Pareto front (PF) in MaOPs is insufficient using Pareto dominance. The use of EMOs based on Pareto dominance for MaOPs makes some issues evident, such as (i) deterioration of the searchability due to the increase in the number of objectives [3], (ii) exponential increase in the number of solutions required for approximation of the entire PF, and (iii) difficulty in visualizing the solutions. These issues are open areas of study and have led to new proposals for algorithms to solve MaOPs. Several algorithms that deal with MOPs and MaOPs in the literature use strategies such as the hybridization of heuristics and meta-heuristics strategies. When one algorithm approach is proposed as a strategy that combines a set of meta-heuristics, and these perform together, producing a profitable synergy, it is called Hybrid Meta-heuristic HM [4], HM also known as hybrid algorithm [5]. The hybrid algorithms have the advantage of dealing with complexity, noise, imprecision, uncertainty, and vagueness present in real-world optimization problems [4].

Strategies such as decomposition, simulated annealing and evolutionary genetic operators are approaches that together had produced profitable hybrid algorithms in recent years. Decomposition is an approach that has been applied to MaOPs with success. It is a strategy that divides a MOP into several single-objective sub-problems solved simultaneously. In addition, Bechikh et al. assert that decomposition is the most successful approach to solving MaOPs [6]. One algorithm based on decomposition is MOEA/D [7]. It is easy to apply the single-objective optimizer to the sub-problem associated with each solution while maintaining the solution dispersed [8]. Three advantages are obtained with the application of decomposition: it improves population diversity, allows for parallelism, and speeds up the solution [8]. Regularly, MOEA/D is the base of numerous new algorithms, though hybridized with other strategies. For example, Simulated Annealing (SA) obtained promising results in a new MOEA/D hybrid algorithm [9]. In the hybrid algorithm, SA worked as a Local Search (LS) strategy applied to the sub-problems. Simulated annealing (SA) is a probabilistic LS method for global optimization problems that allows gradual convergence to a near-optimal solution [10]. SA is a metaheuristic that adds exploration and exploitation to solve global optimization problems. SA is a well-known single-objective optimization algorithm that has been applied in hybrid algorithms such as LS strategy [9] and a selection mechanism [11] in other works. Differential evolution (DE) is a state-of-the-art global optimization technique [12]. This is an evolutionary algorithm (EA) with a classical EA framework, but it differs from other EAs in its mutation operator. It mutates the base vectors with scaled population-derived difference vectors and, as generations pass, these differences tend to adapt to the natural scaling of the problem [12]. Mutation in the DE context is interpreted as change or perturbation that involves random elements. Mutation in combination with the crossover (binomial or exponential crossover) is a strategy that improves the diversity and convergence of the population. As a mechanism of recombination, DE has shown promising results in hybrid algorithms [13–15].

A selection of relevant works related with the present work is described in the next lines. The adaptive evolutionary multi-objective approach based on decomposition (EMOSA) was presented by Li and Landa-Silva [9]. It combines SA with the MOEA/D algorithm [7]. EMOSA follows the concept of a population and external population (archive) of non-dominated solutions. Li and Landa-Silva mention that competition between individuals is essential to balance diversity and convergence. For this reason, when the current solution is updated, different criteria are used for population and external population. Dominance, the Tchebycheff aggregation function, and ϵ-dominance are used to update the neighborhood, the population, and the external population, respectively. MOEA/D-DE was proposed by Li and Zhang [13]. The algorithm is a combination of the MOEA/D framework and Differential Evolution operators. MOEA/D-DE and NSGA-II [16] were tested with complicated Pareto sets shapes. MOEA/D-DE showed much better results improving

NSGA-II. Chen et al. [17] extended MOEA/D-DE, replaced the (1) DE operator with a guided mutation, and (2) implemented a new update mechanism. Various combinations of the mechanisms were proved using the instances of the CEC2009 benchmark [18]. In the work of Liu et al. [16], MOEA/D-DE was improved with two mechanisms: (1) using neighborhood information in the update strategy, and (2) adopting a random scale factor to improve the searchability of MOEA/D-DE. This approach was tested with instances from the CEC2009 benchmark and the real-world problem sizing of folded cascode amplifier. Mashwani et al. [14] suggested a strategy based on MOEA/D and using multiple crossover operators called HAEA/D. This strategy uses an adaptive operator of selection to apply different crossover operators (DE, CMX, PCX, TM, SPX, and SBX). HAEA/D was proved using the CEC2009 benchmark. Results showed that HAEAD performs better in most instances of CEC2009 regarding proximity and diversity. MODESA is a hybrid algorithm presented by Chen et al. [19]. DE algorithm is improving by using SA as a local search mechanism. The algorithm was proved using ZDT and DTLZ benchmarks. ESADE algorithm is the approach of Guo et al. [11]. In this approach, SA is used to improve the DE selection method. The primary strategy is replacing the DE greedy rule of selection with a SA rule of selection. CEC2005 benchmark was used to observe the performance of ESADE.

Focusing on some strategies that had good results in the related works, a hybrid framework called MOSA/D is proposed. According to the state-of-the-art research, the decomposition strategy allows diversity in a population [8] while evolutionary perturbations add convergence toward the Pareto front; however, a question should be considered: What is the effect of such components when included as part of a multi-objective simulated annealing design? Hence, this work studies the performance of MOSA/D considering two widely used perturbation operators in its implementation: classical genetic operators and differential evolution. The proposed algorithms are MOSA/D-CGO, based on classical genetic operators, and MOSA/D-DE, based on differential evolution operators. The main contribution of this work is the performance analysis of MOSA/D using both perturbations functions and identifying the one most suitable to the framework.

This paper is organized as follows. In Section 2, the primary strategies within the MOSA-D framework are provided. In addition, in this section, the proposed MOSA/D-CGO and MOSA/D-DE are described. Section 3 describes the experimental setup. Section 4 reports the experimental results. Finally, Section 5 concludes the paper with some future directions.

2. Materials and Methods

This section describes the main strategies used by the algorithms MOSA/D-CGO and MOSA/D-DE. Both algorithms are inspired by the works of Li and Landa-Silva [9] and Engrand [20]. Strategies such as decomposition, competition between individuals, and the logarithmic composite objective function are part of MOSA/D-CGO and MOSA/D-DE. However, the mentioned methods do not focus on the perturbation function as a strategy. For this reason, both algorithms implement perturbation functions to add convergence toward the PF. MOSA/D-CGO is based on classical genetic operators, while MOSA/D-DE is based on differential evolution operators. Maintaining a simple algorithm structure such as the classical SA is also considered (Algorithm 1).

The proposal to transform the classical SA framework into a multi-objective simulated annealing based on the decomposition framework has three main components:

(1) decomposition strategy,
(2) perturbation function, and
(3) probability of acceptance.

2.1. Decomposition Strategy

A strategy based on MOEA/D [7] is proposed to divide a MOP into N scalar subproblems. To divide a MOP, a set of weight vectors $\lambda^1, \ldots, \lambda^N$ (Figure 1) and an aggregation function (Equation (2)) are used.

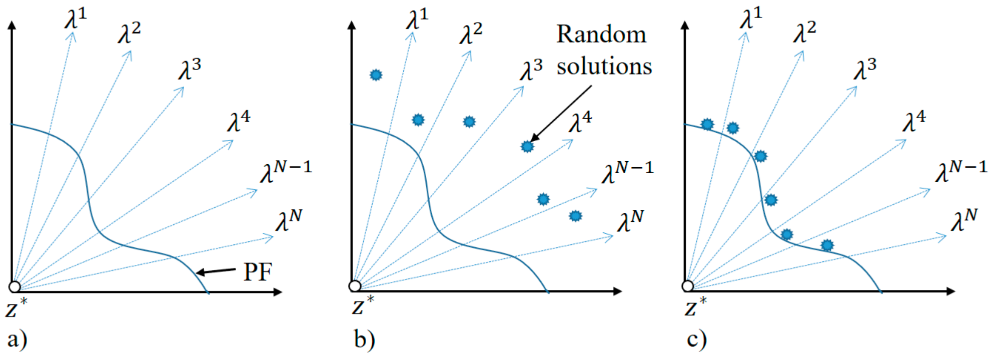


Figure 1. Decomposition strategy. (**a**) A MOP is divided into N sub-problems $\lambda^1, \ldots, \lambda^N$; (**b**) N random solutions are generated; (**c**) over a finite number of generations, solutions are simultaneously improved.

There are a number of aggregation functions; one of the most popular is the Tchebycheff function. Let $\lambda^i = \left(\lambda_1^i, \ldots, \lambda_m^i\right)$ be a weight vector, then the Tchebycheff approach is defined as

$$\text{minimize } g^{te}\left(x \middle| \lambda^i, z^*\right) = \max_{1 \leq j \leq m} \left\{ \lambda_j^i \middle| f_j(x) - z^* \right), \text{ subject to } x \in \Omega, \tag{2}$$

where m is the number of objectives, z^* is the reference point, Ω is the decision variable space, and x is a solution for the ith sub-problem. To improve the solutions for ith sub-problem, a search method such as local search (simulated annealing) must be used and the candidate solutions compared in terms of the Tchebycheff function.

2.2. Perturbation Function

The perturbation function in simulated annealing explores the neighborhood of the current solution to determine a new candidate solution. Classically, the perturbation function uses a random exploration of the neighborhood [21–23]. Two approaches are chosen to create the perturbation function: classical genetic operators and differential evolution.

2.2.1. Classical Genetic Operators

Simulated binary crossover (SBX). SBX creates two-children solutions from two-parent solutions. To create the solutions, it is necessary to follow the procedure described below:

Step 1. Choose a uniform random number $u \in (0, 1)$.

Step 2. Calculate spread factor γ using the equation

$$\gamma = \begin{cases} (2u)^{\frac{1}{(\eta+1)}}, & if\ u \leq 0.5 \\ \frac{1}{[2(1-u)]^{\frac{1}{(\eta+1)}}}, & otherwise \end{cases}, \tag{3}$$

where η is the distribution index, which is a non-negative real number.

Step 3. The children's solutions can be computed using the following equations:

$$c_{1,k} = \frac{1}{2}[(1-\gamma) * p_{1,k} + (1+\gamma) * p_{2,k}], \tag{4}$$

$$c_{2,k} = \frac{1}{2}[(1+\gamma) * p_{1,k} + (1-\gamma) * p_{2,k}]. \tag{5}$$

Polynomial mutation. The idea behind mutation operation is to restore unexpected genetic individuals to avoid local optimal solutions and thus enhance the exploration ability and population diversity. It operates as follows:

$$c_k = p_k + \left(p_k^u - p_k^l\right) * \delta_k, \tag{6}$$

where c_k is the child solution and p_k is the parent solution, p_k^u and p_k^l are the upper and lower bounds of the parent component. Finally, δ_k is a function of u (random number) and η_m (mutation distribution index), as follows:

$$\delta_k = \begin{cases} (2u)^{\frac{1}{(\eta m+1)}}, & if\ u \leq 0.5 \\ 1-[2(1-u)]^{\frac{1}{(\eta m+1)}}, & otherwise \end{cases}. \tag{7}$$

2.2.2. Differential Evolution Operators

This pair of operators is used in a different order from the previous operators. First, the mutation must be calculated; then, the crossover must be executed.

Rand/1 mutation. Rand/1 is a basic mutation of the most popular mutation strategies in differential evolution (DE) [24]. It operates as follows:

$$V_{i,G} = X_{r1,G} + F * (X_{r2,G} - X_{r1,G}) \tag{8}$$

In Equation (8), $X_{r1,G}$, $X_{r2,G}$, $X_{r2,G}$ represent three random solutions of the population at the generation G, and $V_{i,G}$ represents a new mutant solution. F is a parameter called the scale factor.

Binomial crossover. After mutation, a crossover operation is performed between the target vector $X_{i,G}$ and the mutant vector $V_{i,G}$ to produce a trial vector $U_{i,G} = \left\{u_{i,G}^1, u_{i,G}^2, \ldots, u_{i,G}^D\right\}$. The binomial crossover operator is mainly used, and it is defined as

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & if\ rand_j(0,1) \leq Cr\ or\ j = j_{rand} \\ x_{i,G}^j & otherwise \end{cases}. \tag{9}$$

In Equation (9), $Cr \in [0,1]$ represents the crossover rate, and j_{rand} represents the minimum difference between the target vector $X_{i,G}$ and the trial vector $U_{i,G}$.

2.3. Probability of Acceptance

The Boltzmann probability in the classical SA is used to accept or refuse one solution. When the solution is accepted, it has a good probability of guiding new solutions and escaping from local optimal solutions. We propose the probability function in MOSA/D of the work of Engrand [20] that uses a multi-objective version of the Boltzmann probability. It uses a logarithm compositive function. According to Engrand, a new function G is defined as

$$G(X) = \sum_1^m ln f_j(X). \tag{10}$$

The probability of acceptance p is defined by

$$p = Exp\left(-\frac{\Delta G}{T_k}\right). \tag{11}$$

Then, Equation (10) is added into Equation (11):

$$p = Exp\left(-\frac{1}{T_k}\sum_1^m \ln f_j(X_{n+1}) - \sum_1^m \ln f_j(X_n)\right). \tag{12}$$

Using factoring and logarithm identities, the probability of acceptance is expressed in the following way:

$$p = Exp\left(-\frac{1}{T_k}\sum_1^m \ln \frac{f_j(X_{n+1})}{f_j(X_n)}\right), \tag{13}$$

where m is the number of objectives, X_n is the current solution, X_{n+1} is the candidate solution, and T_k is the current temperature of the algorithm.

2.4. Multi-Objective Simulated Annealing Based on Decomposition Framework

The multi-objective simulated annealing based on decomposition (MOSA/D) framework is based on simulated annealing, decomposition strategy from MOEA/D [7], and perturbation functions based on genetic operators (classical and DE operators). The MOSA/D framework (Figure 2) takes a multi-objective optimization problem (MOP) and decomposes it into a set of N sub-problems represented by a set of N weight vectors. Then, ith sub-problem is annealed by L executions at a temperature level of T, obtaining new solutions by a function of perturbation using crossover and mutation operations. The algorithm maintains a population of efficient solutions (P). When a new solution is obtained, it is evaluated to update the efficient solution (P_{i}) associated with the ith sub-problem. Additionally, the framework maintains a solution to explore the feasible surface of ith sub-problem. To update this solution, an evaluation is necessary with the new solution. Both evaluations of the new solution against the efficient and the exploring solution are conducted using the Tchebycheff approach. Tchebycheff approach avoids the use of Pareto dominance in the MOSA/D framework. Furthermore, the framework has a mechanism to escape from local optimums based on a logarithm compositive function. Finally, the perturbation functions are a novel part of the framework. Two algorithms are derived due to the perturbation functions: MOSA/D-CGO and MOSA/D-DE. We intend to observe the performance of both approaches in continuous multiple- and many-objective problems.

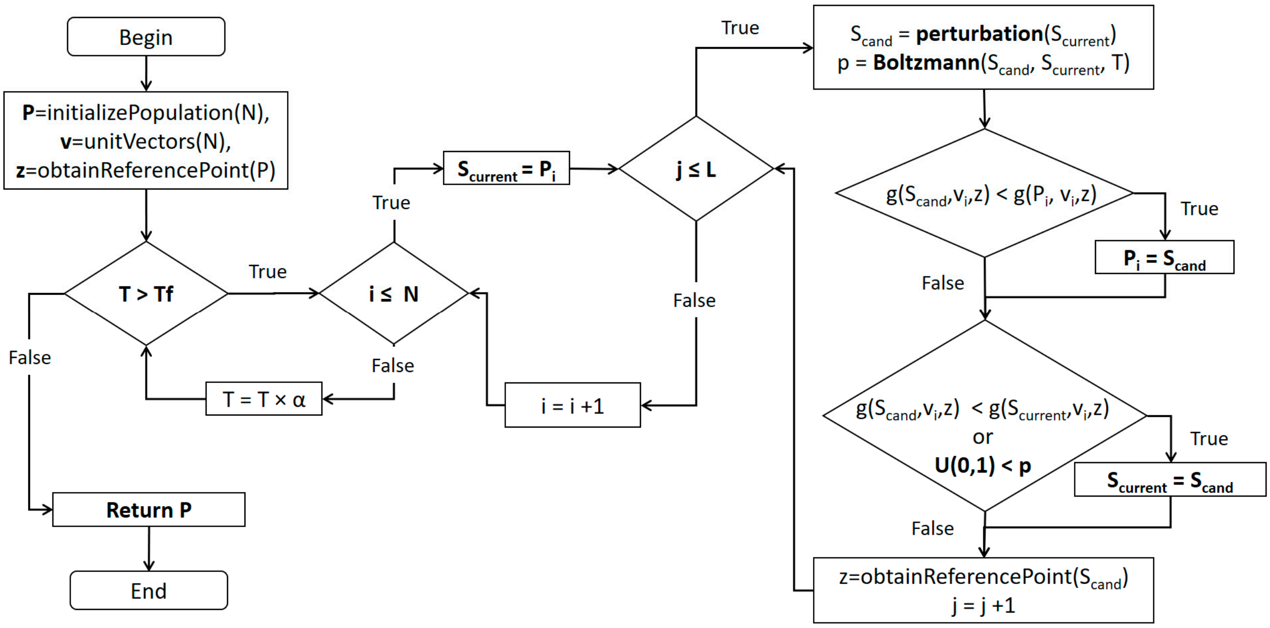

Begin
P=initializePopulation(N),
v=unitVectors(N),
z=obtainReferencePoint(P)
T > Tf
True
False
i ≤ N
True
False
T = T × α
Return P
End
S_current = P_i
j ≤ L
True
False
i = i +1
S_cand = perturbation(S_current)
p = Boltzmann(S_cand, S_current, T)
g(S_cand,v_i,z) < g(P_i, v_i,z)
True
False
P_i = S_cand
g(S_cand,v_i,z) < g(S_current,v_i,z)
or
U(0,1) < p
True
False
S_current = S_cand
z=obtainReferencePoint(S_cand)
j = j +1

Figure 2. MOSA/D framework flowchart.

2.4.1. MOSA/D-CGO Algorithm

The MOSA/D-CGO pseudocode is described by Algorithm 1. The algorithm receives as input the multi-objective problem to be solved, the initial temperature Ti, the final temperature Tf, the time factor α, the size of the population N, and the maximum number of MFE evaluations that it will carry out. The last generation of the population (P) represents the output. The algorithm initializes the random population P of size N, the sub-problems represented by weight vectors v of size N, the temperature $T = Ti$, the $S_{current}$ solution serving as a guide in the search process, and the S_{cand} solution, which is generated from disturbances on $S_{current}$, the calculation of z (reference point). $FE = N$ is updated because it is the size of the initial population (Lines 1 and 2). From Line 3, Algorithm 1 has three loops: the first, second, and third (Lines 3, 4, and 6, respectively).

Algorithm 1 MOSA/D-CGO

```
Input: MOP, Initial temperature Ti, Factor $\alpha$, Markov chain L, Final temperatura Tf, Size of population N, Maximal function evaluations MFE
Output: Last generation of P
1   To initialize: population P(N), weighted vectors v, solution $S_{current}$, solution $S_{cand}$, reference point z, temperature T = Ti
2   FE = N
3   while (T $\geq$ Tf) y (FE $\leq$ MFE)
4       for i = 1 to N
5           $S_{current}$ = $P_i$
6           for j = 1 to L
7               $S_{cand}$ = PerturbationCGO($S_{current}$, P)
8               p = BoltzmannProbability($S_{cand}$, $S_{current}$, T)
9               if g($S_{cand}$, $v_i$, z) < g($P_i$, $v_i$, z)
10                  $P_i$ = $S_{cand}$
11              if g($S_{cand}$, $v_i$, z) < g($S_{current}$, $v_i$, z) or U(0,1) < p
12                  $S_{current}$ = $S_{cand}$
13              z = ObtainReferencePoint ($S_{cand}$)
14              j = j +1
15          i = i +1
16      FE = N $\times$ L
17      T = T $\times$ $\alpha$
18  return P
```

The first loop (main loop) starts (Line 3). Then, while the temperature T is greater than Tf or the maximum number of evaluations is not reached, the main loop will perform the following:

1. Start a second loop that proceeds through all the sub-problems (Line 4),
2. Update $FE = N \times L$ as the multiplication of the number of sub-problems by the annealing cycles (Line 16),
3. The temperature T is updated. It is multiplied by the temperature drop factor α (Line 17).
4. The second loop (from 1 to N sub-problems) has the following tasks (Line 4):
5. The $S_{current}$ solution takes as its value the solution stored in P_i (Line 5),
6. The *i*th sub-problem is annealed for L cycles (Lines 6–14) in the third loop,
7. The sub-problem counter i is updated.
8. The third loop (annealing process) works as follows (Lines 6–14):
9. S_{cand} is obtained by perturbation of $S_{current}$ (Line 7),
10. The Boltzmann probability p of S_{cand} is computed based on Equation (6) (Line 8),
11. The Tchebycheff function g is calculated for S_{cand} and P_i. If g (S_{cand}) is less than g (P_i), then S_{cand} is a better solution than P_i. Then P_i takes the value of S_{cand} (Lines 9–10).
12. The Tchebycheff function g is calculated for S_{cand} and $S_{current}$. If g (S_{cand}) is less than g ($S_{current}$), then $S_{current}$ takes the value of S_{cand} (Lines 11–12).
13. The reference point z is updated (Line 13).
14. The Markov chain counter j is updated (Line 14)
15. At the end of the main loop, the last generation of P is returned, and the algorithm ends (Line 18).

The MOSA/D-CGO implements a perturbation function based on simulated binary crossover (SBX) and polynomial mutation. The pseudocode of *PerturbationCGO*($S_{current}$, P) is shown in Algorithm 2. The algorithm's input is the current solution $S_{current}$ and the population P. At Line 1, all solutions are initialized. Next, parent1 takes the value of $S_{current}$ (Line 2). Next, a random solution is selected from P as parent2 (Line 3). Then the SBX is applied to parent1 and parent2, producing the new solution child (Line 4).

Next, the mutation operator is applied to the solution child (Line 5), producing the candidate solution. Finally, the candidate solution is returned to the main algorithm.

Algorithm 2 PerturbationCGO

```
Input: Current solution $S_{current}$, population P
Output: Candidate Solution $S_{cand}$
1    Initialize solutions: parent1, parent2, child
2    parent1 = $S_{current}$
3    parent2 = RandomSelection(P)
4    child = SBX(parent1, parent2)
5    $S_{cand}$ = PolynomialMutation(child)
6    return $S_{cand}$
```

2.4.2. MOSAD-DE Algorithm

Algorithm 3 is the main function of MOSA/D–DE. It is the same as the MOSA/D-CGO. The difference between MOSA/D-CGO and MOSA/D–DE is the perturbation function.

Algorithm 3 MOSA/D-DE

```
Input: MOP, Initial temperature Ti, Factor $\alpha$, Markov chain L, Final temperatura Tf, Size of
population N, Maximal function evaluations MFE
Output: Last generation of P
1    To initialize: population P(N), weighted vectors v, solution $S_{current}$, solution $S_{cand}$,
     reference point z, temperature T = Ti
2    FE = N
3    while ($T \geq Tf$) y ($FE \leq MFE$)
4    for i = 1 to N
5    $S_{current} = P_i$
6    for j = 1 to L
7    $S_{cand}$ = PerturbationDE($S_{current}$, P)
8    p = BoltzmannProbability($S_{cand}$, $S_{current}$, T)
9    if $g(S_{cand}, v_i, z) < g(P_i, v_i, z)$
10   $P_i = S_{cand}$
11   if $g(S_{cand}, v_i, z) < g(S_{current}, v_i, z)$ or $U(0,1) < p$
12   $S_{current} = S_{cand}$
13   z = ObtainReferencePoint ($S_{cand}$)
14   j = j +1
15   i = i +1
16   $FE = N \times L$
17   $T = T \times \alpha$
18   return P
```

In Algorithm 3, Line 7 references the function *PerturbationDE*($S_{current}$, *P*). Its pseudocode is shown in Algorithm 4. The algorithm begins with the initialization of all solutions used in the algorithm (Line 1). Next, the solution target takes the values of $S_{current}$ (Line 2). The next step is to select three random solutions from the population *P* (Lines 3, 4, and 5). In Line 6, *RandOneMutation()* (based on Equation (12)) produces the mutant solution. However, the mutant solution may exceed the limit values of its decision variables. For this reason, a repair operation is necessary over the mutant solution (Line 7). Then, the binomial crossover is executed to generate the candidate solution (S_{cand}) (Line 8). Finally, the candidate solution (S_{cand}) is returned to Algorithm 3.

Algorithm 4 PerturbationDE

```
Input: Current solution $S_{current}$, population P
Output: Candidate Solution $S_{cand}$
    Initialize solutions: target, mutant, sol1, sol2, sol3
    target = $S_{current}$
    sol1 = RandomSelection(P)
    sol2 = RandomSelection(P)
    sol3 = RandomSelection(P)
    mutant = RandOneMutation(ind1, ind2, ind3)
    mutant = Repair(mutant)
    $S_{cand}$ = BinomialCrossover (target, mutant)
    return $S_{cand}$
```

3. Experimental Setup

This section presents the experimental setup to assess our proposed approaches: MOSA/D-CGO and MOSA/D-DE. Then, both algorithms are described to analyze their performance.

Benchmark problems. The experimental design used the DTLZ [25] and CEC2009 [18]. From DTLZ benchmark, DTLZ1 to DTLZ7 were selected with 3, 5, and 10 objectives for both algorithms MOSA/D-CGO and MOSA/D-DE. The number of variables of decision n was calculated by $n = m + k$, where m is the number of objectives and k is set to 5, 10, and 20 for DTLZ1, DTLZ2-6, and DTLZ7, respectively. The total number of experimental instances is 42 (7 problems $\times$ 3 objective configurations $\times$ 2 algorithms). From CEC2009 benchmark, UF1 to UF7 were selected with 2 objectives and UF8 to UF10 with 3 objectives. For all problems, the number of decision variables was 30. The total number of experimental instances was 20 (7 problems $\times$ 1 objective configurations $\times$ 2 algorithms + 3 problems $\times$ 1 objective configurations $\times$ 2 algorithms).

Parameter settings. In general, for both algorithms, the setting of the parameter is defined in Table 1. The parameter setting for the CGO perturbation function is defined in Table 2. Finally, the parameter setting for the DE perturbation function is defined in Table 3.

Table 1. General parameter setting.

Parameter	Setting
Maximal function evaluations (MFE)	100,000
Size of the population (N)	100
Chain of Markov (L)	20 (DTLZ), 2 (CEC2009)
Initial temperature (Ti)	1
Final temperature (Tf)	0.0000001
Temperature factor (α)	0.98

Table 2. Parameter setting for MOSA/D.

Parameter	Setting
Crossover probability	1
SBX distribution index	15
Mutation probability	1/number of objectives
Mutation distribution index	20

Table 3. Parameter setting for MOSA/D-DE.

Parameter	Setting
Scale Factor (F)	0.5
Crossover rate (Cr)	0.8

Performance indicator. The indicators used to measure the performance of both algorithms were hypervolume (HV) and Inverted Generational Distance (IGD). HV allows measurement of convergence and diversity with a single value. Therefore, to calculate HV as a reference point is necessary. The reference points were obtained from determining both algorithms' maximum objective value for DTLZ1 to DTLZ7 with 3, 5, and 10 objectives (Table 4). The reference points from UF1 to UF7 (2 objectives) and UF8 to UF10 (3 objectives) are shown in Table 5. IGD allows measurement of convergence and diversity with a single value. To calculated IGD is necessary set of points of the real Pareto front.

Table 4. Reference points for DTLZ problems used by HV calculations.

Test Problem	3 Objectives	5 Objectives	10 Objectives
DTLZ1	216.790132039933, 245.572126619941, 265.114937485812	80.8714391297344, 227.104643547038, 235.570177240396, 201.976698417331, 344.81892944792	4.06142015762104, 2.25156894368627, 28.0355439457454, 18.0131641679773, 48.6777918069853, 55.3772497318585, 133.939834807773, 139.713524060566, 222.454058357031, 321.944734395031
DTLZ2	2.38622940286774, 2.47201723363377, 2.6806935648516	2.05896735099775, 2.07725455911758, 2.30808795675219, 2.57449333535658, 2.9380095369211	0.6594305076443, 0.648350703222635, 0.840342066735919, 1.24805308667332, 1.54612890881046, 1.70777551383235, 1.8764060443055, 1.97914511371776, 2.29490194623458, 2.65985264422613
DTLZ3	822.509995384644, 799.103543751443, 1242.39097615921	689.016436482976, 547.613912205914, 500.989149866526, 827.81074606278, 1089.39809110161	49.243814823296, 78.5147608980575, 133.64045805898, 150.770000538526, 217.750852407648, 340.827375270017, 565.932643419409, 495.059846407958, 829.140836249926, 1271.6885408782,
DTLZ4	2.58950883612183, 1.65645180809823, 2.02042410538188	2.66379101299125, 2.13852567972279, 2.04131611330391, 1.99090557186032, 1.98125761314518,	2.64822143939513, 2.20030288801471, 2.2491760776445, 2.083066369613, 2.1065144928458, 2.09386422171755, 1.99635654481651, 2.24556199410806, 2.01705020385728, 2.14178980883573
DTLZ5	2.06970298570428, 2.11909090597952, 2.74503917772623	1.31770044511346, 1.68921445015483, 3.31646087578869, 3.41224588178889, 3.49994217680484	0.291869064968389, 0.49338116154924, 0.525290602300518, 0.611762301869089, 0.751594578115702, 1.10935039999348, 2.68788397225164, 3.29311486089528, 3.41214316344024, 3.49975337563819
DTLZ6	9.71802048756533, 9.50035002572186, 10.5676062054648	9.56330574133902, 9.59158225806973, 10.477256602242, 10.9414071660156, 10.923652672406,	3.6503932212918, 3.29696637156103, 5.62792594695437, 7.16542587886431, 7.97964101213103, 9.77412868397095, 9.89622372940959, 10.6364664206205, 10.9073924572044, 10.9367989501483
DTLZ7	1.0, 0.999999999998507, 24.2793251715135	1.0, 0.999999994782373, 0.999999999986268, 0.999999999997808, 42.0189690066051,	1.0, 0.999999999990235, 0.999999991586879, 1.0, 0.999999998676653, 1.0, 0.99999999981814, 1.0, 0.999999997160675, 21.0

Table 5. Reference points for CEC2009 problems used by HV calculations.

Test Problem	2 Objectives	3 Objectives
UF1	4.056791172 3.587159682	-
UF2	2.700068047 2.02284611	-
UF3	3.643565528 4.753315586	-
UF4	1.205666356 1.221462823	-
UF5	9.178972446 10.07250673	-
UF6	12.40456172 13.75189469	-
UF7	4.169101474 3.206875182	-
UF8	-	9.214592744 10.62998938 11.46084736
UF9	-	9.853718763 12.22418277 8.259884555
UF10	-	9.853718763 12.22418277 8.259884555

4. Results

The experimental results are reported in Tables 6–9, and expressed as the mean and standard deviation of the HV (Tables 6 and 7) and IGD (Tables 8 and 9). The best value of each configuration is highlighted in gray. To observe whether there is a significant difference between the MOSA/D-CGO and MOSA/D-DE results, we employ the Wilcoxon signed-rank test with $\alpha = 0.05$ (two tails). The symbols ↑, ↓, and — represent a significant difference in favor of MOSA/D-DE, a significant difference in favor of MOSA/D-CGO, and no significant difference between MOSA/D-CGO and MOSA/D-DE, respectively.

The results of Table 6 show that MOSA/D-DE outperforms MOSA/D-CGO in six out of seven instances with three objectives. For five objectives, MOSA/D-DE outperforms MOSA/D-CGO in three out of seven instances (two instances without a significant difference). Finally, for 10 objectives, MOSA/D-DE outperforms MOSA/D-CGO in four out of seven instances.

The results of Table 7 show that MOSA/D-DE outperforms MOSA/D-CGO in all instances with two objectives. For three objectives, MOSA/D-DE outperforms MOSA/D-CGO in all instances.

The results of Table 8 show that MOSA/D-DE outperforms MOSA/D-CGO in six out of seven instances with three objectives. For five objectives, MOSA/D-DE outperforms MOSA/D-CGO in four out of seven instances. Finally, for 10 objectives, MOSA/D-DE and MOSA/D-CGO provide similar results in three out of seven (one instance without a significant difference).

The results of Table 9 show that MOSA/D-DE outperforms MOSA/D-CGO in six out seven instances with two objectives. For three objectives, MOSA/D-DE outperforms MOSA/D-CGO in all instances.

Table 6. Means and standard deviation of HV results of 42 instances.

PROBLEM	M	MOSA/D-CGO	MOSA/D-DE
DTLZ1	3	1.411396×10^{7} (7.856170×10^{1})	↑1.411407×10^{7} (2.077398×10^{1})
	5	3.013240×10^{11} (7.546308×10^{4})	↑3.013241×10^{11} (1.109078×10^{4})
	10	1.667917×10^{16} (2.419205×10^{13})	↑1.668377×10^{16} (3.215349×10^{8})
DTLZ2	3	1.508330×10^{1} (2.781792×10^{-2})	↑1.517692×10^{1} (7.637901×10^{-3})
	5	7.361979×10^{1} (7.759884×10^{-1})	↑7.387519×10^{1} (3.847436×10^{-2})
	10	↓2.600539×10^{1} (4.958777×10^{-1})	2.569999×10^{1} (1.980545×10^{-1})
DTLZ3	3	8.163884×10^{8} (6.516296×10^{4})	↑8.165865×10^{8} (4.513141×10^{2})
	5	1.689445×10^{14} (8.081423×10^{12})	↑1.704707×10^{14} (9.718498×10^{7})
	10	1.699675×10^{24} (4.226203×10^{22})	↑1.707961×10^{24} (8.685232×10^{18})
DTLZ4	3	7.958441×10^{0} (1.283030×10^{-2})	↑8.002655×10^{0} (1.583281×10^{-2})
	5	↓4.522733×10^{1} (3.629265×10^{-2})	4.509505×10^{1} (8.641444×10^{-2})
	10	↓2.327196×10^{3} (3.100255×10^{0})	2.298456×10^{3} (1.139568×10^{1})
DTLZ5	3	9.944000×10^{0} (1.078512×10^{-2})	↑9.990017×10^{0} (2.501031×10^{-2})
	5	- - -7.652259×10^{1} (3.121247×10^{-1})	- - -7.657619×10^{1} (1.983684×10^{-1})
	10	3.412050×10^{0} (2.043577×10^{-2})	↑3.428953×10^{0} (8.094812×10^{-3})
DTLZ6	3	↓9.009791×10^{2} (1.712226×10^{1})	7.219506×10^{2} (8.185913×10^{0})
	5	↓1.053329×10^{5} (4.977173×10^{3})	7.647568×10^{4} (1.856713×10^{3})
	10	↓4.179011×10^{8} (6.692261×10^{6})	2.876860×10^{8} (9.403331×10^{6})
DTLZ7	3	1.536594×10^{1} (6.362204×10^{-1})	↑1.699378×10^{1} (2.933636×10^{-1})
	5	- - -2.219751×10^{1} (1.606481×10^{0})	- - -2.263389×10^{1} (9.949435×10^{-1})
	10	0.000000×10^{0} (0.000000×10^{0})	↑5.649215×10^{-6} (1.130646×10^{-5})

The symbols ↑, ↓, and - - - represent a significant difference in favor of MOSA/D-DE, a significant difference in favor of MOSA/D-CGO, and no significant difference between MOSA/D-CGO and MOSA/D-DE, respectively.

Table 7. Means and standard deviation of HV results of 20 instances.

Problem	M	MOSA/D_CGO	MOSA/D-DE
UF1	2	1.207938×10^{1} (3.053312×10^{-1})	↑1.359271×10^{1} (8.924848×10^{-2})
UF2	2	4.420144×10^{0} (1.124631×10^{-1})	↑4.898329×10^{0} (4.114705×10^{-2})
UF3	2	1.272886×10^{1} (2.770899×10^{-1})	↑1.446032×10^{1} (1.560741×10^{-1})
UF4	2	6.199354×10^{-1} (7.645704×10^{-3})	↑6.226053×10^{-1} (4.375474×10^{-3})
UF5	2	5.794553×10^{1} (2.163053×10^{0})	↑7.537953×10^{1} (9.461970×10^{-1})
UF6	2	1.449117×10^{2} (2.789804×10^{0})	↑1.558762×10^{2} (9.956338×10^{-1})
UF7	2	1.023408×10^{1} (3.651312×10^{-1})	↑1.159760×10^{1} (2.062088×10^{-1})
UF8	3	1.021735×10^{3} (2.236911×10^{1})	↑1.112894×10^{3} (4.871298×10^{-1})
UF9	3	8.871173×10^{2} (1.462057×10^{1})	↑9.214801×10^{2} (8.179234×10^{0})
UF10	3	2.676289×10^{2} (5.962540×10^{1})	↑8.332120×10^{2} (1.796126×10^{1})

The symbols ↑ represent a significant difference in favor of MOSA/D-DE, a significant difference in favor of MOSA/D-CGO, and no significant difference between MOSA/D-CGO and MOSA/D-DE, respectively.

Table 8. Means and standard deviation of IGD results of 42 instances.

Problem	M	MOSA/D_CGO	MOSA/D-DE
DTLZ1	3	1.698462×10^{0} (5.231658×10^{-1})	↑2.567411×10^{-2} (3.988398×10^{-4})
	5	1.848446×10^{0} (7.371114×10^{-1})	↑1.362515×10^{-1} (8.662087×10^{-3})
	10	2.605653×10^{0} (8.858904×10^{-1})	↑2.746888×10^{-1} (2.055289×10^{-2})
DTLZ2	3	1.338869×10^{-1} (1.050833×10^{-2})	↑6.539183×10^{-2} (8.887102×10^{-4})
	5	↓3.879495×10^{-1} (2.383225×10^{-2})	4.315983×10^{-1} (1.731880×10^{-2})
	10	↓8.086104×10^{-1} (3.491915×10^{-2})	8.740702×10^{-1} (3.886488×10^{-2})
DTLZ3	3	3.943222×10^{1} (1.066523×10^{1})	↑8.927530×10^{-2} (8.582940×10^{-3})
	5	3.527925×10^{1} (1.256126×10^{1})	↑2.402407×10^{0} (1.052387×10^{0})
	10	3.857988×10^{1} (8.728455×10^{0})	↑1.452693×10^{1} (3.864857×10^{0})
DTLZ4	3	2.193973×10^{-1} (2.708875×10^{-2})	↑7.370770×10^{-2} ($2.095534\text{E} \times 10^{-3}$)
	5	4.264948×10^{-1} (2.618424×10^{-2})	↑3.453491×10^{-1} (1.259533×10^{-2})
	10	- - -7.485737×10^{-1} (2.873640×10^{-2})	- - -7.565568×10^{-1} (2.740685×10^{-2})
DTLZ5	3	3.692489×10^{-2} (4.482185×10^{-3})	↑8.796775×10^{-3} (4.662547×10^{-4})
	5	↓2.173594×10^{-1} (2.151767×10^{-2})	3.198228×10^{-1} (3.323426×10^{-3})
	10	↓8.274774×10^{-1} (3.498018×10^{-2})	9.278413×10^{-1} (8.544034×10^{-3})

Table 8. *Cont.*

Problem	M	MOSA/D_CGO	MOSA/D-DE
DTLZ6	3	$\downarrow 2.360560 \times 10^{0}$ (4.875849×10^{-1})	5.755759×10^{0} (1.295398×10^{-1})
	5	$\downarrow 3.233184 \times 10^{0}$ (5.565340×10^{-1})	7.249152×10^{0} (1.815938×10^{-1})
	10	$\downarrow 4.311238 \times 10^{0}$ (3.579400×10^{-1})	7.249090×10^{0} (1.507697×10^{-1})
DTLZ7	3	2.640245×10^{0} (5.499560×10^{-1})	$\uparrow 6.763103 \times 10^{-1}$ (4.596304×10^{-2})
	5	5.536790×10^{0} (6.953019×10^{-1})	$\uparrow 1.562072 \times 10^{0}$ (2.362084×10^{-1})
	10	5.536790×10^{0} (6.953019×10^{-1})	$\uparrow 1.562072 \times 10^{0}$ (2.362084×10^{-1})

The symbols ↑, ↓, and - - - represent a significant difference in favor of MOSA/D-DE, a significant difference in favor of MOSA/D-CGO, and no significant difference between MOSA/D-CGO and MOSA/D-DE, respectively.

Table 9. Means and standard deviation of IGD results of 20 instances.

Problem	M	MOSA/D_CGO	MOSA/D-DE
UF1	2	3.118639×10^{-1} (3.337199×10^{-2})	$\uparrow 1.245048 \times 10^{-1}$ (6.952854×10^{-3})
UF2	2	1.632088×10^{-1} (1.376776×10^{-2})	$\uparrow 4.207095 \times 10^{-2}$ (4.246406×10^{-3})
UF3	2	5.070647×10^{-1} (3.574368×10^{-2})	$\uparrow 3.410425 \times 10^{-1}$ (1.406431×10^{-2})
UF4	2	$\downarrow 8.882224 \times 10^{-2}$ (3.714504×10^{-3})	9.322984×10^{-2} (2.313839×10^{-3})
UF5	2	2.573650×10^{0} (3.112357×10^{-1})	$\uparrow 1.614455 \times 10^{0}$ (1.012664×10^{-1})
UF6	2	1.323509×10^{0} (1.554659×10^{-1})	$\uparrow 7.905724 \times 10^{-1}$ (4.148593×10^{-2})
UF7	2	4.111049×10^{-1} (4.643987×10^{-2})	$\uparrow 1.801766 \times 10^{-1}$ (2.685634×10^{-2})
UF8	3	8.631345×10^{-1} (1.315110×10^{-1})	$\uparrow 2.057906 \times 10^{-1}$ (1.069731×10^{-2})
UF9	3	8.061542×10^{-1} (1.206518×10^{-1})	$\uparrow 2.987048 \times 10^{-1}$ (1.581723×10^{-2})
UF10	3	6.233193×10^{0} (9.715169×10^{-1})	$\uparrow 1.485037 \times 10^{0}$ (1.454003×10^{-1})

The symbols ↑ and ↓ represent a significant difference in favor of MOSA/D-DE, a significant difference in favor of MOSA/D-CGO, and no significant difference between MOSA/D-CGO and MOSA/D-DE, respectively.

4.1. DTLZ Benchmark Analysis

DTLZ1 and DTLZ3 are test problems with the difficulty of possessing local Pareto-optimal fronts. This allows us to observe the algorithms' ability to escape local Pareto-optimal fronts. In this case, for both problems, MOSA/D-DE has promising HV (Table 6) and IGD (Table 8) results that outperform the MOSA/D-CGO results for three, five, and ten objectives. In Figure 3, it is possible to observe the Pareto front of DTLZ1 and DTLZ3 obtained by MOSA/D-CGO (left side) and MOSA/D-DE (right side) for three objectives. The contribution of DE allows the algorithm to escape from local Pareto-fronts better than the contribution of CGO.

DTLZ2 is a test problem that allows us to observe the ability to converge to concave Pareto fronts. MOSA/D-DE outperforms MOSA/D-CGO in three and five objectives, but with ten objectives, the results favor MOSA/D-CGO in terms of HV results. The performance of differential evolution is promising for three and five objectives. The contribution of classical genetic operators improves the performance with 10 objectives. It is in concordance with IGD results where CGO outperforms DE in five and ten objectives. Figure 4 shows the convergence ability of MOSA/D-DE with three objectives.

DTLZ4 is a test problem derived from DTLZ2 that allows observation of the algorithms capable of maintaining a good distribution of solutions. Based on the HV results of Table 6, MOSA/D-DE has a better distribution for three objectives (Figure 5). Moreover, for five and ten dimensions, the MOSA/D-CGO approach has a better distribution of solutions. In contrast, IGD (Table 8) shows results in favor of MOSA/D-DE with three and five objectives. It may be because MOSA/D-DE has more solutions near of the real Pareto front than MOSA/D-CGO.

DTLZ5 is a test problem derived from DTLZ2 that allows us to observe whether the algorithms can converge to a curve. From the HV results (Table 6), MOSA/D-DE outperforms MOSA/D-CGO with three and ten objectives. For five objectives, the Wilcoxon signed-rank test indicates no significant difference between the MOSA/D-CGO and MOSA/D-DE results. In contrast, IGD results show that MOSA/D-CGO has a more solutions near of the real PF. Figure 6 shows the convergence toward the PF (curve) using MOSA/D-DE (right side).

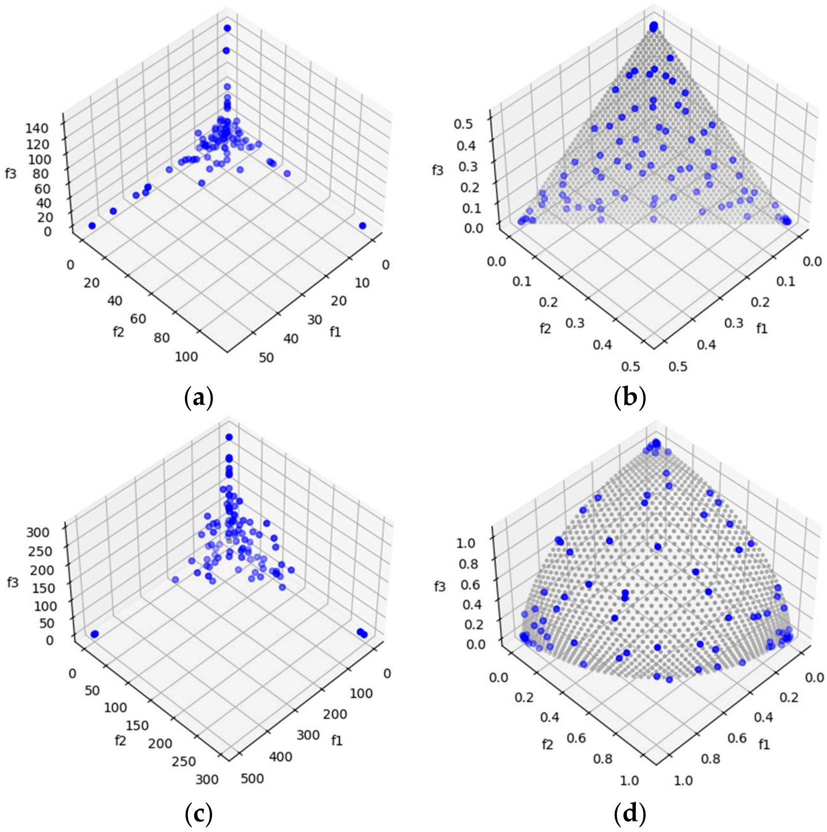


Figure 3. The plots of the approximate PFs with the highest HV value against real PF. (**a**) DTLZ1 approximate PF using MOSA/D-CGO; (**b**) DTLZ1 approximate PF using MOSA/D-DE; (**c**) DTLZ3 approximate PF using MOSA/D-CGO; (**d**) DTLZ3 approximate PF using MOSA/D-DE.

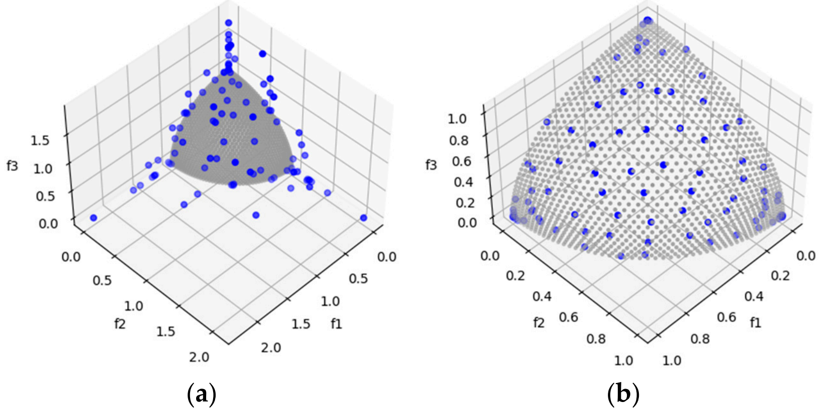
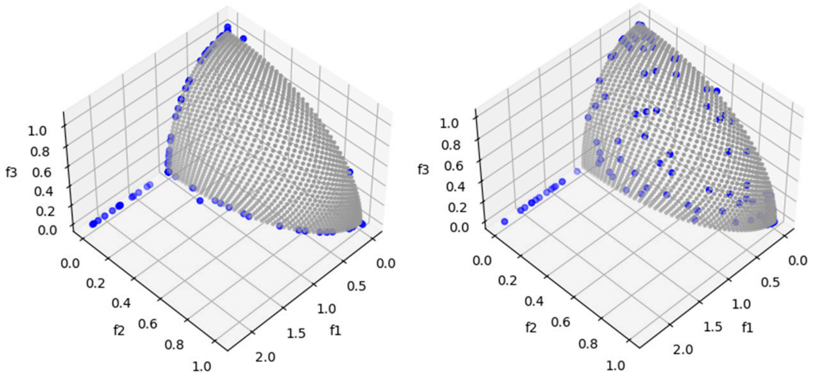


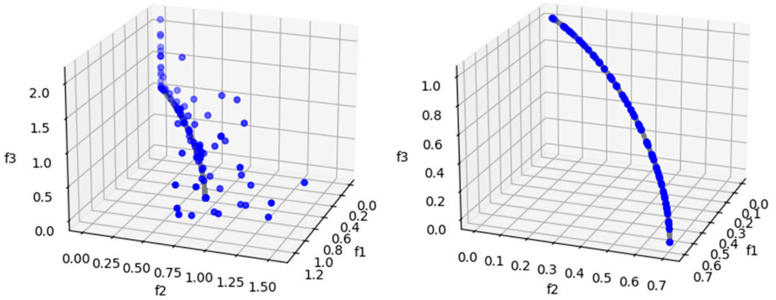
Figure 4. The plots of the approximate PFs with the highest HV value against real PF. (**a**) DTLZ2 approximate PF using MOSA/D-CGO; (**b**) DTLZ2 approximate PF using MOSA/D-DE.

DTLZ6 is a more complex test problem derived from DTLZ5. It allows us to observe whether the algorithms are capable of converging to a curve. From the HV results, MOSA/D-CGO outperforms MOSA/D-DE for three, five and ten objectives. However, in this problem, the convergence of both algorithms deteriorates. For example, in Figure 7, both approaches fail to achieve convergence toward the PF. In results of HV and IGD, better performance of MOSA/D-DE over MOSA/D-CGO is displayed.



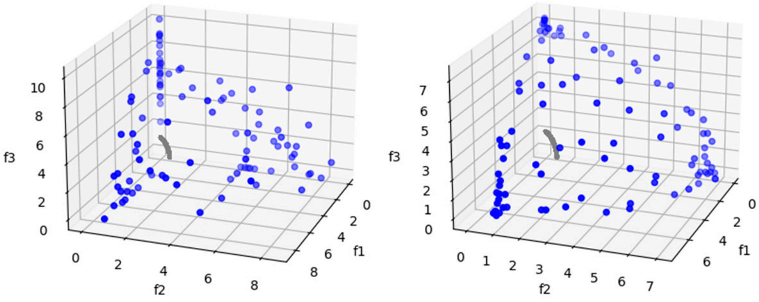
(**a**) (**b**)

Figure 5. The plots of the approximate PFs with the highest HV value against real PF. (**a**) DTLZ4 approximate PF using MOSA/D-CGO; (**b**) DTLZ4 approximate PF using MOSA/D-DE.



(**a**) (**b**)

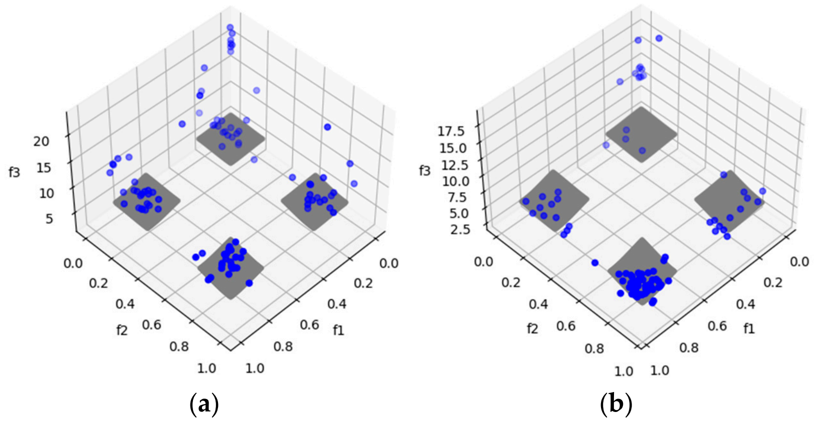
Figure 6. The plots of the approximate PFs with the highest HV value against real PF. (**a**) DTLZ5 approximate PF using MOSA/D-CGO; (**b**) DTLZ5 approximate PF using MOSA/D-DE.



(**a**) (**b**)

Figure 7. The plots of the approximate PFs with the highest HV value against real PF. (**a**) DTLZ6 approximate PF using MOSA/D-CGO; (**b**) DTLZ6 approximate PF using MOSA/D-DE.

DTLZ7 is a test problem with disconnected Pareto-optimal regions in the search space. The purpose of this problem is to observe the algorithms' ability to maintain sub-populations in different Pareto-optimal regions [25]. MOSA/D-DE outperforms MOSA/D-CGO for three and ten objectives based on the HV results (Table 6). For five objectives, the Wilcoxon signed-rank test indicates no significant difference between the MOSA/D-CGO and MOSA/D-DE results for DTLZ7. In addition, IGD (Table 8) results show that MOSA/D-DE outperform MOSA/D-CGO for three, five and ten objectives. However, in this problem, the convergence of the algorithms deteriorates although Figure 8 and shows that the four sub-populations of both approaches are necessary for more convergence toward the PF.

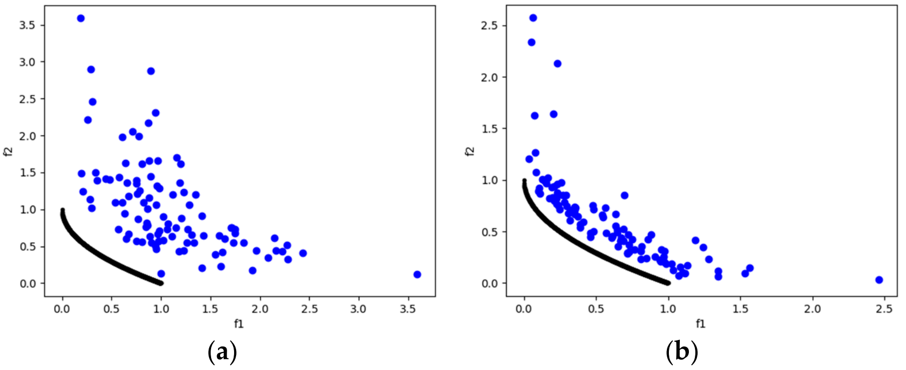


(**a**) (**b**)

Figure 8. The plots of the approximate PFs with the highest HV value against real PF. (**a**) DTLZ7 approximate PF using MOSA/D-CGO; (**b**) DTLZ7 approximate PF using MOSA/D-DE.

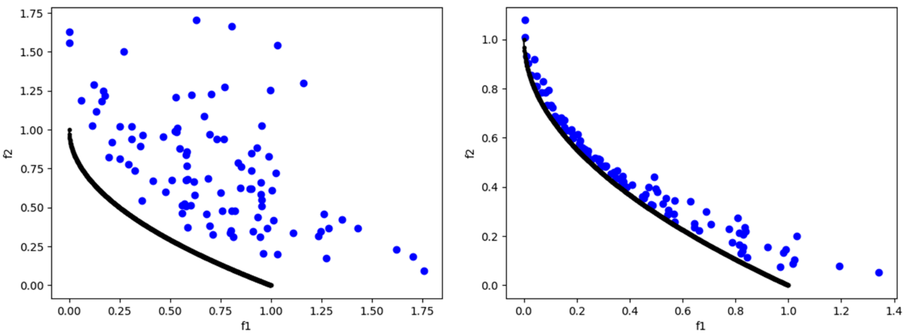
4.2. CEC2009 Benchmark Analysis

CEC2009 benchmark is a set of problems with complicated Pareto set shapes [18] to observe the searchability of multi-objective optimization algorithms. The HV results (Table 7) show that MOSA/D-DE outperforms MOSA/D-CGO in all instances. According to the IGD results (Table 9), MOSA/D-DE outperforms MOSA/D-CGO in experimental instances from UF1 to UF3 and UF5 to UF7 for 2 objectives, while MOSA/D-CGO outperforms MOSA/D-DE in UF4. The plots of approximate PFs for 2 objectives can be seen in Figures 9–15. On the other hand, according to the IGD results (Table 9), MOSA/D-DE outperforms MOSA/D-CGD in all instances of 3 objectives (UF8 to UF10). The plots of approximate PFs for 3 objectives can be seen in Figures 16–18. Therefore, using the DE perturbation function has a better search ability contribution than the CGO perturbation function. MOSA/D-DE has promising results in the UF2 (Figure 10), UF4 (Figure 12), and UF8 (Figure 16) problems. However, the performance of both algorithms needs more search ability in seven out of ten problems of the benchmark. It shows a clear opportunity to improve the mechanism in future works.



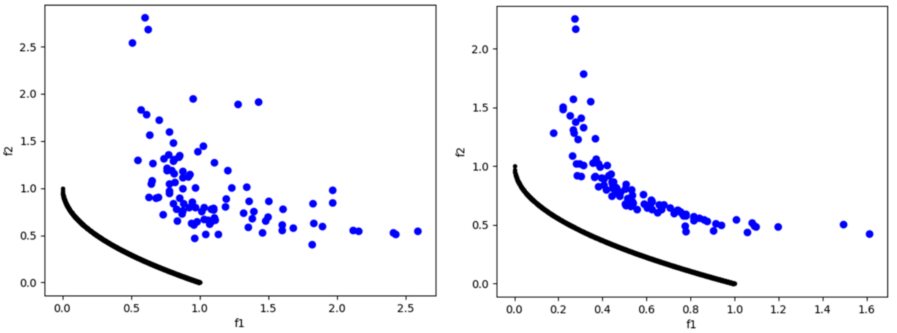
(**a**) (**b**)

Figure 9. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF1 approximate PF using MOSA/D-CGO; (**b**) UF1 approximate PF using MOSA/D-DE.



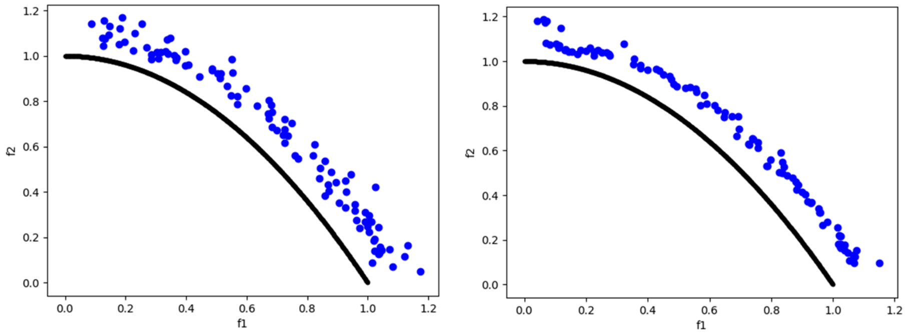
(**a**) (**b**)

Figure 10. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF2 approximate PF using MOSA/D-CGO; (**b**) UF2 approximate PF using MOSA/D-DE.



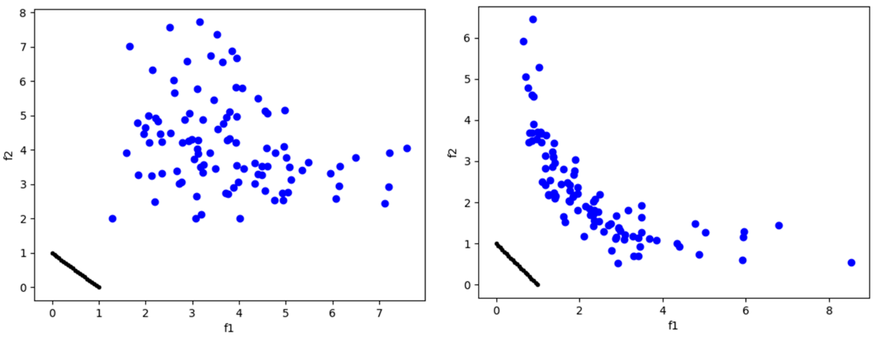
(**a**) (**b**)

Figure 11. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF3 approximate PF using MOSA/D-CGO; (**b**) UF3 approximate PF using MOSA/D-DE.



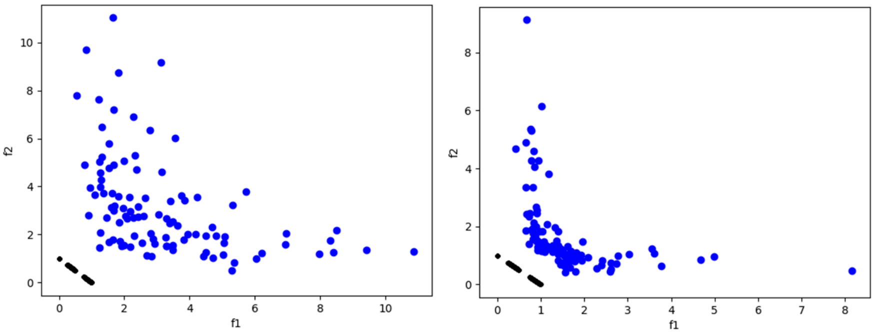
(**a**) (**b**)

Figure 12. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF4 approximate PF using MOSA/D-CGO; (**b**) UF4 approximate PF using MOSA/D-DE.



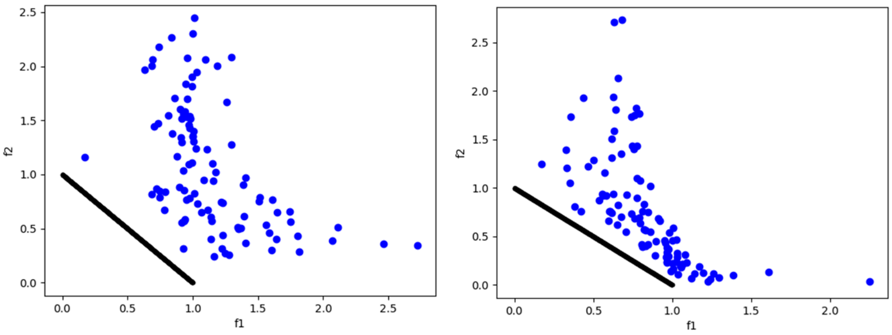
(**a**) (**b**)

Figure 13. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF5 approximate PF using MOSA/D-CGO; (**b**) UF5 approximate PF using MOSA/D-DE.



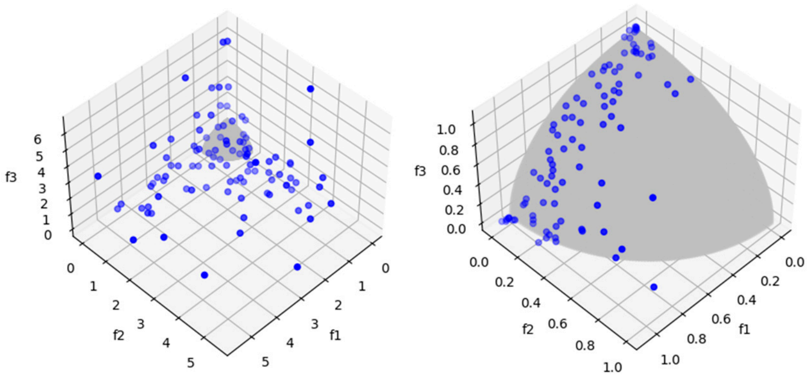
(**a**) (**b**)

Figure 14. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF6 approximate PF using MOSA/D-CGO; (**b**) UF6 approximate PF using MOSA/D-DE.



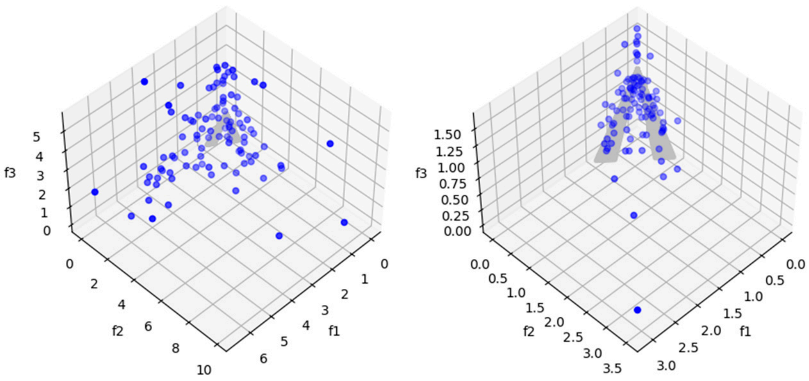
(**a**) (**b**)

Figure 15. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF7 approximate PF using MOSA/D-CGO; (**b**) UF7 approximate PF using MOSA/D-DE.



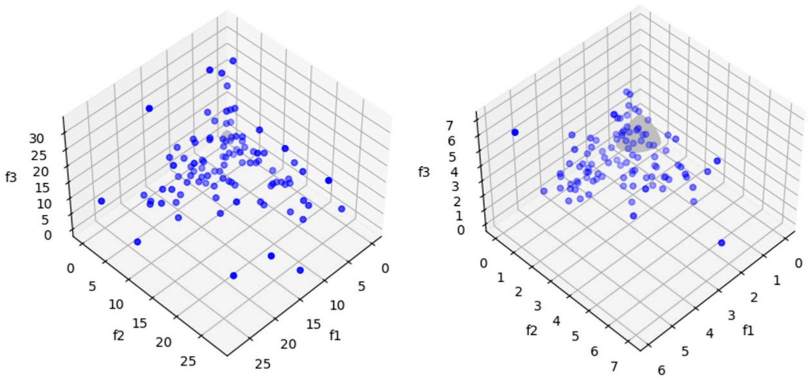
(**a**) (**b**)

Figure 16. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF8 approximate PF using MOSA/D-CGO; (**b**) UF8 approximate PF using MOSA/D-DE.



(**a**) (**b**)

Figure 17. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF9 approximate PF using MOSA/D-CGO; (**b**) UF9 approximate PF using MOSA/D-DE.



(**a**) (**b**)

Figure 18. The plots of the approximate PFs with the highest HV value against real PF. (**a**) UF10 approximate PF using MOSA/D-CGO; (**b**) UF10 approximate PF using MOSA/D-DE.

5. Conclusions

The MOSA/D-CGO and MOSA/D-DE approaches to deal with continuous multiple- and many-objective problems were introduced in this paper. Both algorithms harness the ability of convergence and diversity from simulated annealing and decomposition, respectively. MOSA/D-CGO adopts a perturbation function based on classical genetic operators. MOSA/D-DE adopts a perturbation function based on differential evolution. It is essential to point out that MOSA/D-DE shows some promising abilities: (i) its ability to escape from local Pareto-optimal sets in linear and concave problems, (ii) its ability to converge to concave and curved Pareto fronts. Furthermore, MOSA/D-CGO also shows promising abilities: (i) it obtains reasonably distributed solutions with five and ten objectives. MOSA/D-CGO and MOSA/D-DE show convergence to concave Pareto fronts. Both algorithms show the promising ability to converge to Pareto-optimal regions.

The MOSA/D-CGO and MOSA/D-DE algorithms show deterioration of convergence with complex problems with curved Pareto fronts. Both algorithms also show deterioration of convergence with solutions in separate Pareto regions. In conclusion, based on the experiments, the MOSA/D-DE algorithm has promising results. The results in almost all the experiments conducted for three, five, and ten objectives favor MOSA/D-DE. In addition, MOSAD-DE outperforms MOSA/D-CGO in two abilities: (i) the capacity to escape from local Pareto-optimal fronts and (ii) convergence to curved Pareto fronts. The other abilities as measured by the DTLZ benchmark show that both MOSA/D algorithms have mixed results. In addition, both algorithms were proved using problems with complicated Pareto set shapes. Both algorithms were proved using problems with complicated Pareto set shapes. The HV and IGD results show that MOSA/D-DE outperforms MOSA/D-CGO in almost all experimental instances. Furthermore, MOSA/D-DE has promising results in three out of ten problems. However, the mechanism needs more searchability with the complicated Pareto set shapes. Finally, using perturbation functions in multi-objective simulated annealing significantly impacts the convergence toward the PF. This impact can be improved in future work.

In future work, the aim is to determine another strategy to add convergence into complex problems with curved PF and separate Pareto regions. For example, to converge in a curved PF (DTLZ6), both MOSA fail, but, based on the HV results, MOSA/D-CGO has a slight advantage over MOSA/D-DE. Both algorithms indeed achieve solutions in separate Pareto regions (Figure 8), but there needs to be more convergence. It will be interesting to observe the performance of both algorithms with additional convergence. The search ability with the CEC2009 benchmark show deterioration; for this reason, an update of the mechanism is necessary. MOSA/D algorithms are the hybridized result of simulated annealing, decomposition, and special perturbation functions. This design allows a simple structure (Algorithm 1) that can be easily updated. For example, the algorithms can both use the concept of the neighborhood as in MOEA/D [7] and use a strategy based on preferences to deal with MaOPs. In addition, future work will aim to observe the performance of both MOSA/D with real world problems such as the portfolio selection problem (PSP), knapsack problem, and traveling salesman problem.

Author Contributions: The manuscript was written through the contributions of all authors. Conceptualization, M.V.-M. and N.R.-V.; methodology, M.V.-M., N.R.-V. and E.F.; software, M.V.-M. and N.R.-V.; validation, M.V.-M., N.R.-V. and E.F.; formal analysis, C.G.-S.; investigation, M.L.M.-R.; resources, C.G.-S.; data curation, N.R.-V.; writing—original draft preparation, M.V.-M.; writing—review and editing, M.V.-M. and N.R.-V.; visualization, C.G.-S.; supervision, E.F.; project administration, M.L.M.-R.; funding acquisition, M.L.M.-R., C.G.-S. and N.R.-V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The files of the results produced by this work are available at https://www.dropbox.com/sh/fk8bt8v8gy6ghbz/AABaDrhbC2DKwkB_YH2L64xZa?dl=0, accessed on 29 January 2023.

Acknowledgments: José Manuel Vargas Martínez acknowledges CONACyT Ph.D. scholarship (Grant 788668). Also, the authors acknowledge the support obtained from: IxM CONACyT project 3058-“Optimización de Problemas Complejos”, Laboratorio Nacional de Tecnologías de Información (LaNTI) at TecNM/Instituto Tecnológico de Ciudad Madero, and TecNM research networks “Tecnologías Computacionales Aplicadas” and “Electromovilidad”.

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

References

1. Chand, S.; Wagner, M. Evolutionary Many-Objective Optimization: A Quick-Start Guide. *Surv. Oper. Res. Manag. Sci.* **2015**, *20*, 35–42. [CrossRef]
2. Ishibuchi, H.; Tsukamoto, N.; Nojima, Y. Evolutionary Many-Objective Optimization: A Short Review. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation, Hong Kong, China, 1–6 June 2008; pp. 2424–2431.
3. Falcón-Cardona, J.G.; Coello Coello, C.A. A new indicator-based many-objective ant colony optimizer for continuous search spaces. *Swarm Intell.* **2017**, *11*, 71–100. [CrossRef]
4. Rodriguez, F.J.; García-Martinez, C.; Lozano, M. Hybrid Metaheuristics Based on Evolutionary Algorithms and Simulated Annealing: Taxonomy, Comparison, and Synergy Test. *IEEE Trans. Evol. Comput.* **2012**, *16*, 787–800. [CrossRef]
5. Mashwani, W.K. Comprehensive Survey of the Hybrid Evolutionary Algorithms. *Int. J. Appl. Evol. Comput.* **2013**, *4*, 1–19. [CrossRef]
6. Bechikh, S.; Elarbi, M.; Ben Said, L. Many-objective optimization using evolutionary algorithms: A survey. *Adapt. Learn. Optim.* **2017**, *20*, 105–137. [CrossRef]
7. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
8. Xu, Q.; Xu, Z.; Ma, T. A short survey and challenges for multiobjective evolutionary algorithms based on decomposition. In Proceedings of the CITS 2019—International Conference on Computer, Information and Telecommunication Systems, Beijing, China, 28–31 August 2019; pp. 1–5. [CrossRef]
9. Li, H.; Landa-Silva, D. An adaptive evolutionary multi-objective approach based on simulated annealing. *Evol. Comput.* **2011**, *19*, 561–595. [CrossRef] [PubMed]
10. Amine, K. Multiobjective Simulated Annealing: Principles and Algorithm Variants. *Adv. Oper. Res.* **2019**, *2019*, 8134674. [CrossRef]
11. Guo, H.; Li, Y.; Li, J.; Sun, H.; Wang, D.; Chen, X. Differential Evolution Improved with Self-Adaptive Control Parameters Based on Simulated Annealing. *Swarm Evol. Comput.* **2014**, *19*, 52–67. [CrossRef]
12. Opara, K.R.; Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **2019**, *44*, 546–558. [CrossRef]
13. Li, H.; Zhang, Q. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **2009**, *13*, 284–302. [CrossRef]
14. Mashwani, W.K.; Salhi, A.; Yeniay, O.; Jan, M.A.; Khanum, R.A. Hybrid Adaptive Evolutionary Algorithm Based on Decomposition. *Appl. Soft Comput. J.* **2017**, *57*, 363–378. [CrossRef]
15. Mashwani, W.; Salhi, A.; Jan, M.; Khanum, R.; Sulaiman, M. Impact Analysis of Crossovers in a Multi-Objective Evolutionary Algorithm. *Sci. Int.* **2015**, *27*, 4943–4956.
16. Liu, B.; Fernández, F.V.; Zhang, Q.; Pak, M.; Sipahi, S.; Gielen, G. An Enhanced MOEA/D-DE and Its Application to Multiobjective Analog Cell Sizing. In Proceedings of the 2010 IEEE Congress on Evolutionary Computation CEC, Barcelona, Spain, 18–23 July 2010. [CrossRef]
17. Chen, C.M.; Chen, Y.P.; Zhang, Q. Enhancing MOEA/D with Guided Mutation and Priority Update for Multi-Objective Optimization. In Proceedings of the 2009 IEEE Congress on Evolutionary Computation CEC, Trondheim, Norway, 18–21 May 2009; pp. 209–216. [CrossRef]
18. Zhang, Q.; Zhou, A.; Zhao, S.; Suganthan, P.N.; Liu, W. Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition. In Proceedings of the 2009 IEEE Congress on Evolutionary Computation CEC, Trondheim, Norway, 18–21 May 2009; pp. 1–30.
19. Chen, B.; Zeng, W.; Lin, Y.; Zhong, Q. An Enhanced Differential Evolution Based Algorithm with Simulated Annealing for Solving Multiobjective Optimization Problems. *J. Appl. Math.* **2014**, *2014*, 931630. [CrossRef]
20. Engrand, P. Multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management. In Proceedings of the 1997 ICONE International Conference on Nuclear Engineering, Nice, France, 25–29 May 1997; p. 477.
21. Ulungu, E.L.; Teghem, J.; Fortemps, P.H.; Tuyttens, D. Heuristic for multi- objective combinatorial optimization problems by simulated annealing. In *MCDM: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 1995; pp. 229–238.
22. Ulungu, E.L.; Teghem, J.; Fortemps, P.H.; Tuyttens, D. MOSA method: A tool for solving multiobjective combinatorial optimization problems. *J. Multi-Criteria Decis. Anal.* **1999**, *8*, 221–236. [CrossRef]
23. Czyżak, P.; Jaszkiewicz, A. Pareto simulated annealing-a metaheuristic technique for multiple-objective combinatorial optimization. *J. Multi-Criteria Decis. Anal.* **1998**, *7*, 34–47. [CrossRef]
24. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [CrossRef]
25. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. Scalable Test Problems for Evolutionary Multiobjective Optimization. In *Evolutionary Multiobjective Optimization*; Springer: London, UK, 2005; pp. 105–145. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.