Received February 28, 2020, accepted March 15, 2020, date of publication March 20, 2020, date of current version March 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982251

A Competitive Co-Evolutionary Approach for the Multi-Objective Evolutionary Algorithms

VAN TRUONG VU[1], **LAM THU BUI**[1], **AND TRUNG THANH NGUYEN**[2]
[1]Le Quy Don Technical University, Hanoi 100000, Vietnam
[2]Liverpool John Moores University, Liverpool L3 5UG, U.K.

Corresponding author: Trung Thanh Nguyen (t.t.nguyen@ljmu.ac.uk)

This work was supported in part by the Newton Fund and managed by the Royal Academy of Engineering under Grant NRCP1516-1-134, and in part by the Vietnam Ministry of Science and Technology under Grant HNQT/SPDP/14.19.

ABSTRACT In multi-objective evolutionary algorithms (MOEAs), convergence and diversity are two basic issues and keeping a balance between them plays a vital role. There are several studies that have attempted to address this problem, but this is still an open challenge. It is thus the purpose of this research to develop a dual-population competitive co-evolutionary approach to improving the balance between convergence and diversity. We utilize two populations to solve separate tasks. The first population uses Pareto-based ranking scheme to achieve better convergence, and the second one tries to guarantee population diversity via the use of a decomposition-based method. Next, by operating a competitive mechanism to combine the two populations, we create a new one with a view to having both characteristics (i.e. convergence and diversity). The proposed method's performance is measured by the renowned benchmarks of multi-objective optimization problems (MOPs) using the hypervolume (HV) and the inverted generational distance (IGD) metrics. Experimental results show that the proposed method outperforms cutting-edge co-evolutionary algorithms with a robust performance.

INDEX TERMS Dual-population, convergence, diversity, co-evolution, competitive.

I. INTRODUCTION

There exist many practical problems in which often-conflicted objectives need to be optimized simultaneously; especially prolems in machine learning where we are seeking a model with the best performance in both accuracy and generalization measures. These problems are called multi-objective optimization problems (MOPs). Unlike single-objective optimization which can be easy to find the best single solution, in multi-objective optimization (MOO), a set of optimal solutions (called Pareto-optimal solutions) will be usually selected. Obviously, finding the largest number of Pareto-optimal solutions possible from the MOO is a vital but time-consuming task. Therefore, the MOO tries to find a set of solutions that satisfy both criteria: as close as possible to the Pareto-optimal front and as diverse as possible [1].

Unlike single-solution-based algorithms, population-based algorithms like evolutionary algorithms (EAs) can find a number of solutions simultaneously and hence it has become a major approach for dealing with MOPs [2]. Recently, Multi-objective Evolutionary Algorithms (MOEAs) have become one of the present trends in developing EAs. Various MOEAs like Pareto-based algorithms ([3], [4]), indicator-based algorithms [5], decomposition-based algorithms [6], or direction-based algorithm [7] have been proposed. These MOEAs differ both in convergence as well as in diversity preservation. In general, these algorithms can be divided into three groups. The first one (i.e Pareto-based algorithms) allocates priority on handling the convergence and the second one (i.e decomposition algorithm) focuses on the diversity. Meanwhile, the last group (i.e indicator-based algorithms) considers both convergence and diversity by using an indicator like Hypervolume (HV). Typical indicator-based algorithms are IBEA (Indicator-based evolutionary algorithm [5]); dynamic neighborhood MOEA based on HV indicator (DNMOEA/HI) [8]; a HV estimation algorithm (HypE) [9]; and S-metric selection evolutionary multiobjective optimisation algorithms (SMS-EMOA) [10]. These algorithms have an advantage that they do not require any additional diversity preservation mechanisms. However, when the number of objectives increases, the computational complexity of these algorithms also increases very quickly. This is their biggest

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Liu.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

weakness. This drawback has limited its application in solving multi-and many-objective problems.

In general, using only a single algorithm to solve the problem of balancing between convergence and diversity in MOPs is not easy. Therefore, the current trend is to combine multiple algorithms. This approach can be divided into two main groups: Multi-algorithm approach [15] (i.e. using multiple algorithms on the same population) and multi-population approaches [16] (i.e. using multiple populations, each corresponds to one objective). In [15], to balance between convergence and diversity, the researchers introduced a multi-algorithm based on Non-dominated Sorting Genetic Algorithm II (NSGA-II) and IBEA (named MABNI). To be specific, NSGA-II and IBEA ran on the same population. After a series of test trials, especially the ZDT and the DTLZ ones, the MABNI produced good results. In [16], instead of using the same population, the authors used multiple populations to cope with multiple objectives, in which each objective was optimized by one population. The authors adopted particle swarm optimization (PSO) for each swarm and designed a co-evolutionary multi-swarm PSO algorithm (named CMPSO). The experiment results showed that CMPSO is suitable for solving MOPs with two and three objectives.

The multi-population approach can be regarded as a co-evolutionary algorithm (CoEA). The general idea of CoEA is to break down a problem into a set of sub-problems and uses multiple populations to optimize different sub-problems. The CoEA can be categorized into two groups [17] which are competitive and cooperative ones. In the competitive approach, the fitness of each individual in one population is measured by the competition with some individuals in other populations. With regard to the latter group, a collaborative mechanism is used to determine the fitness of each individual. The first version of cooperative co-evolution was proposed in [18]. In [19] a framework of the CoEA was used for the flexible pickup and delivery problem with time windows. In this study, there are two separate populations; one is employed for diversification purpose while the other is used for evolutionary intensification. In [20] based on a cooperative CoEA with dual populations, a new hybrid learning algorithm was introduced to design a radial basis function neural network (RBFNN) models with feature selection. While the purpose of the first population is to find out the most significant input characteristics of RBFNN, the second one aims at discovering the optimal RBFNN structure. Sharing the same idea, the authors in [21] employed a 2-population cooperative CoEA (named differential evolution-based coevolutionary multi-objective optimization algorithm (DECMO)). In DECMO, SPEA2 (Strength Pareto Evolutionary Algorithm 2) and DEMO (DE for Multiobjective Optimization)/GDE3 (Generalized Differential Evolution) models with a similar fitness mechanism were used in the first and second population respectively. In general, the cooperative mechanism between multiple populations is favorably utilized by CoEAs to deal with MOPs. To interested readers, papers ([16] and [22]) are good references for further understanding on this area of research. Unlike the cooperative CoEAs, there exists a lack of research addressing the competitive CoEAs ([22]–[24]). In [23], the authors proposed a competitive and cooperative co-evolutionary model (named CCPSO) for designing multi-objective particle swarm optimization algorithm. In [24], a combination of competitive and cooperative mechanisms was proposed to solve MOPs in a dynamic environment.

Recently, there have been many studies addressing the problem of balancing convergence and diversity in solving more complex problems such as constrained multi-objective optimization problems (CMOPs) ([25]–[27]), dynamic multi-objective optimization [28], many objectives ([29], [30]), or ensemble learning problems (with the objectives of maximizing accuracy and diversity of the ensemble). The main idea of these studies is mainly based on a combination of two Pareto-based and decomposition-based methods. In [25], the authors used a co-evolutionary algorithm using the two-archive strategy (called C-TAEA) for solving the CMOPs. In particular, C-TAEA utilized two populations, one named convergence-oriented archive (CA) and the other named diversity-oriented archive (DA). CA's mission is to maintain convergence and feasibility. The DA, meanwhile, is responsible for preserving the convergence and diversity of the evolution process. The empirical results on benchmark and real-world problems showed the competitiveness of the proposed method in comparison with other state-of-the-art algorithms.

In [31], Ke Li *et al.* dealt with convergence and diversity simultaneously by employing a dual-cooperative co-evolution paradigm (DPP). With the first population, a Pareto-based mechanism was operated in order to maintain a solution set with satisfactory. The solutions of this population are randomly spread. Regarding the second population, diversity was preserved by the application of a decomposition-based mechanism. In order to guarantee this trait, solutions in this population are uniformly spread. Finally, a restricted mating selection mechanism (RMS) was employed to harmonize interactions between two co-evolving populations. In the RMS, two mating parents are chosen from both populations. Each of them is restrictively selected from its neighboring sub-regions with a large probability. Because of this selection, there is a possibility that the individual in the first population may not be found. If this happens, an alternative individual can be taken from the corresponding sub-region in the second population. In such a case, both mating parents are selected from the same population, rendering the co-evolutionary mechanism meaningless. To address these shortcoming, Vu *et al.* [32] improved this model by proposing a new restricted selection mechanism as well as some small improvements in the DPP model to shorten the running time as well as achieving better results. Li *et al.* [33] proposed a dual-population approach for balancing convergence and diversity. The authors utilized a grid dominance relationship to maintain the convergence and a decomposition based

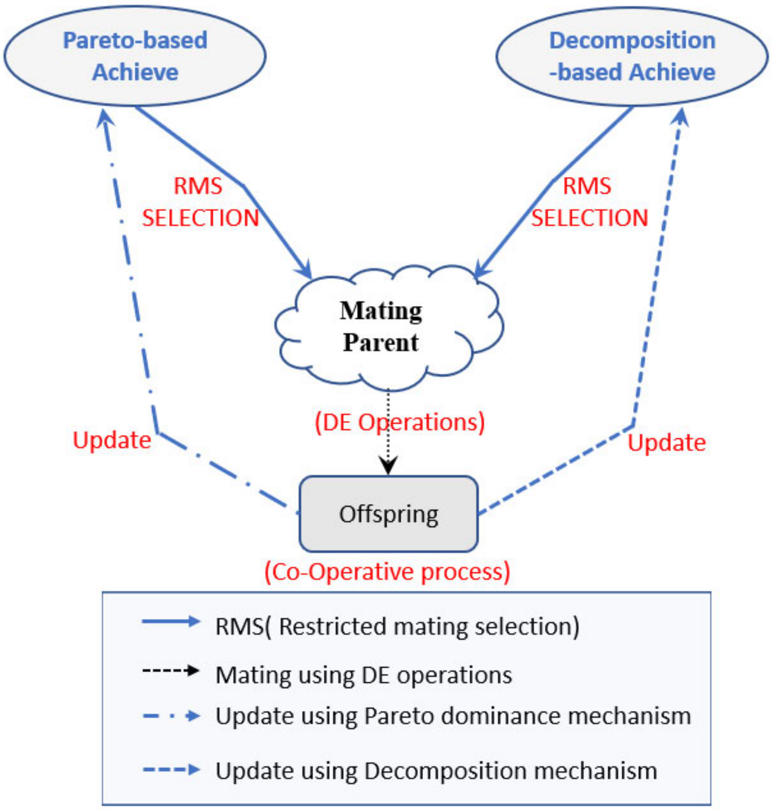


FIGURE 1. The pseudo-code of the DPP algorithm.

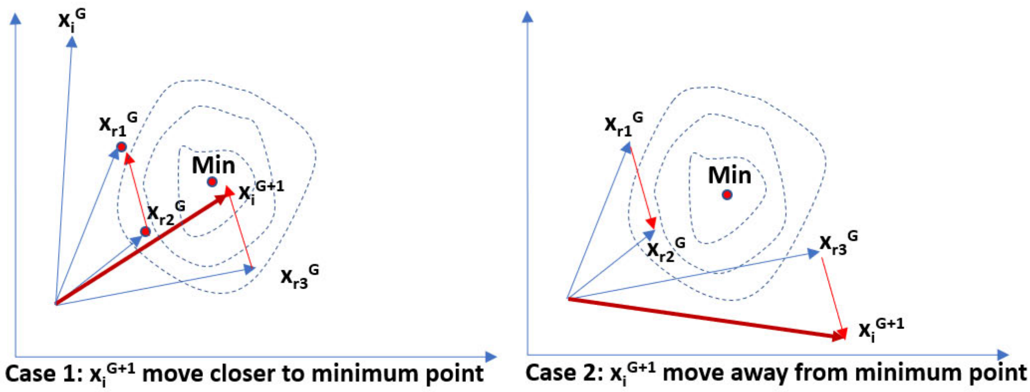


FIGURE 2. The way to generate offsprings from mating parents using DE operators.

selection principle to preserve well distributed solutions like the second population in the DPP.

Inspired by the co-evolution paradigm with encouraging results [31], we continue to explore in this direction. Specifically, in our research, a competitive co-evolutionary approach is developed to solve multiobjective optimization problems. The difference between this paper and existing studies is detailed as follows. First, we utilize an other mating selection mechanism instead of the RMS mechanism to select two mating parents. Second, to generate two offsprings from the selected parents, we use a competitive model instead of the co-operative one.

In summary, our main contributions are summarized as follows:

(a) We present a new dual-population competitive co-evolutionary approach (DPPCP) that uses a competitive co-evolutionary mechanism instead of a co-operative one for interaction between two populations.

(b) We propose a new neighbor-based selection mechanism (NBSM) to select mating solutions instead of using restricted mating selection (RMS) mechanism like previous studies.

(c) We perform extensive experiments on the proposed algorithm to compare and analyze results with existing and related algorithms.

The rest of this paper is organized as follows. In Section II, background algorithms are presented with two well-known algorithms (NSGA-II and MOEA/D) and the dual-population paradigm (DPP). Afterward, the detail of the proposed method is shown in Section III. Then, experimental results and discussions are given in Section IV. Finally, the paper is concluded in Section V.

II. BACKGROUND

A. NON-DOMINATED SORTING GENETIC ALGORITHM II (NSGA-II)

NSGA-II [3] is one of the most common algorithms among Pareto-based EMO algorithms. This is an improved version of NSGA [34]. In NSGA-II, based on the objective function values, each solution knows how many solutions it dominates and how many solutions that dominate it. Thereafter, a non-dominated sorting mechanism will be used to rank solutions and assign them to Pareto-fronts $(F_0, \ldots, F_l, \ldots, F_p)$. All solutions on the same front will not dominate each other or be dominated by one another. Solutions on a front will dominate solutions on other fronts with higher ranks. Both populations of parents and offspring are joined in a hybrid population. Half of them are selected for the new population. To construct a new population, the selection will start from F_0 (i.e. the lowest rank front) to a front denoted as F_l. Because all solutions on F_l have the same convergence, they need a diversity mechanism to compare. NSGA-II used the crowding-distance as a secondary selection strategy. This way, NSGA-II always tries to keep the convergence as much as possible.

In Pareto-based algorithms, convergence and diversity are considered in turn. In NSGA-II, for example, at each generation, solutions are ranked using a non-dominated sorting method. As a result, a population is divided into multiple fronts. Individuals with lower ranks (i.e. corresponds to better convergence) are preselected. Then, solutions on the last front are selected up to the full size of a population by using a diversity selection approach (i.e. crowding distance). Therefore, in NSGA-II, the preservation of diversity is secondary. It only guarantees diversity for a limited number of solutions in the population; the rest is selected mainly based on the convergence regardless of their diversity. This causes a limitation in solving problems with many-objective (i.e. more than three objectives) or difficult problems with the complicated Pareto-optimal set.

B. MULTIOBJECTIVE EVOLUTIONARY ALGORITHM BASED ON DECOMPOSITION (MOEA/D)

To balance convergence and diversity, a decomposition-based approach is also applied. In this approach, a complex MOP is decomposed into several sub-problems and these sub-problems are solved in a collaborative manner [11]. A MOP may be divided into a group of single-objective problems (e.g. MOEA/D [6] and MOEA/D-DE (MOEA/D based on differential evolution) [12]) or a group of sub-MOPs without using any aggregation function (e.g. NSGA-III [13] and MOEA/D-M2M (a version of multiobjective optimization evolutionary algorithm-based decomposition) [11]). Because different solutions in the population are associated with different sub-problems, diversity is naturally maintained [6]. Whereas, by optimizing sub-problems, the convergence criterion will be satisfied. However, the limitation of this approach is that algorithms may struggle to preserve diversity in high dimensional objective space. As discussed in [14] the reason comes from the contour lines of aggregation functions used in decomposition-based MOEAs.

MOEA/D [6] is a decomposition-based method. It decomposes MOPs into a set of single-objective optimization sub-problems through an aggregation method (such as the weighted sum, Tchebycheff and boundary intersection approaches [35]). In order to address these sub-problems, a population-based algorithm is applied. In MOEA/D, each solution is associated with a sub-problem and the population consists of the best solution for each sub-problem. Therefore, the diversity among these sub-problems will result in the diversity in the population. In addition, a set of evenly spread weight vectors is used by MOEA/D to identify the search directions. Therefore, MOEAD can produce a uniform distribution of Pareto solutions.

C. THE DUAL-POPULATION PARADIGM (DPP)

Given in Fig.1 is the general architecture of DPP model [31], which employed two co-evolving populations. The Pareto-based mechanism is used in the first population (named A_p) and the decomposition-based mechanism is used in the second population (named A_d). These populations engage in a parallel evolution. At each generation, a restricted mating selection mechanism (RMS) allows them to interact with each other. In the RMS, the mating parent include three solutions, of which two are selected from A_d and the remaining one is selected from A_p. Thanks to this, the parents could give all the positive characteristics (i.e. the convergence and diversity) to the offspring. To update both A_p and A_d, the offspring utilizes the corresponding archiving mechanism.

In the RMS process, there exist two cases. In the first case, if no solution is included in the selected sub-region in A_p, an alternative solution will be chosen by the RMS in the corresponding one in A_d. In the second case, if more than one solution is found in the sub-region, only one solution will be selected.

This algorithm gives some promising results. However, there are two areas for possible improvements, as discussed below:

1) RESTRICTED MATING SELECTION METHOD

The authors restrict the mating parents from neighboring sub-regions with a high probability (and there is only a low probability of these mating parents to be selected from the whole population). However, they only randomly select a neighboring sub-region from A_p regardless of whether this sub-region contains any solutions in the A_p or not. This leads to a high possibility that the selected sub-region does not

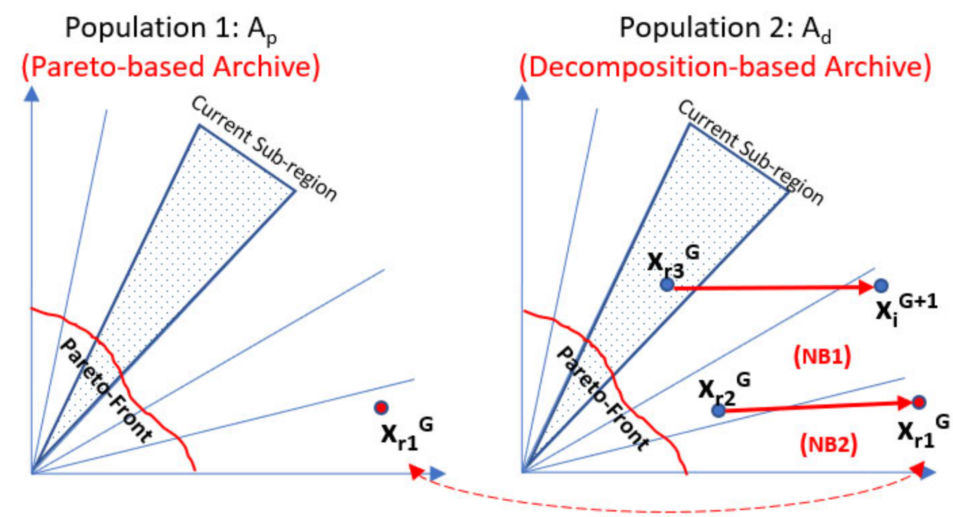


FIGURE 3. A simple illustration of generating spring from mating parents.

contain any solutions (so an alternative solution has to be borrowed from the corresponding sub-region in A_d). This may lead to an imbalance between the two populations.

2) THE INTERACTION BETWEEN TWO CO-EVOLVING POPULATIONS

In DPP, the authors define the interaction as the way to generate offspring from mating parents. To be specific, they use differential evolution (DE) for offspring generation. This means they need three solutions (such as, $x_{r1}^G, x_{r2}^G, x_{r3}^G$, where x_{r3}^G is the current solution, x_{r1}^G is a solution selected from A_p and x_{r2}^G is a solution selected from A_d) to create new offspring (x_{r3}^{G+1}).

$$x_i^{G+1} = x_{r3}^G + F * (x_{r1}^G - x_{r2}^G) \quad (1)$$

It is worth noting that in Eq.1, $F * (x_{r1}^G - x_{r2}^G)$ is a direction vector. This vector is vital because it may help to direct the current vector to a new location that is closer to the global extremes or maybe even make it move further away from this position. Take Fig.[2] as an example. In case 1, using Eq.1, from the three parents $x_{r1}^G, x_{r2}^G, x_{r3}^G$, we can obtain an offspring solution x_i^{G+1} whose position is closer to the global extreme position (denoted by *Min*) than its parents. On the contrary, in case 2, also using Eq.1, although the parents $x_{r1}^G, x_{r2}^G, x_{r3}^G$ are close to *Min*, the offspring solution x_i^{G+1} is actually further away from *Min* than its parents. In DPP, the authors select x_{r3}^G and x_{r1}^G from A_d and x_{r2}^G from A_p with the hope that x_{r1}^G has good convergence properties and x_{r2}^G has a promising diversity. In this way, we have a large chance to generate offspring having both of advantages. However, there still exist two major drawbacks:

(+) Choosing two out of three solutions from the A_d and only one from A_p may cause an imbalance in the co-evolutionary process.

(+) Since the direction vector is made up of two solutions in two different populations, it could lead to unpromising outcomes, especially when the two populations are imbalanced (i.e. the convergence of a population is much better than the other). Let us consider a simple example in Fig.3. x_{r2}^G is quite close to the Pareto front. Meanwhile, x_{r1}^G is far from the Pareto front. Suppose that we are running with the A_d population, by iterating over each sub-region, for each sub-region (assuming the current sub-region contains x_{r3}^G), we make a random selection of two neighbour sub-regions (e.g. NB1, NB2). In these 2 sub-regions, NB1 contains a solution (e.g. x_{r2}^G), while NB2 does not contain any solution. In this case, NB2 will borrow a solution in the corresponding sub-region on the A_p population(e.g. x_{r1}^G). After mating, based on Eq.1,we might obtain the offspring x_i^{G+1}. It can be seen that x_i^{G+1} has shifted to a position that is far from the Pareto-front. This leads to poorer results.

This paper attempts to address the aforementioned drawbacks. To do so, we propose a new dual-population co-evolutionary approach named DPPCP (*The dual-population competitive co-evolutionary approach*). This approach differs from the DPP model in two ways. First, it uses a competitive co-evolution rather than co-evolution to interact between two co-evolving populations. Second, it uses a neighbor-based selection mechanism (NBSM) instead of the RMS to select three different solutions on each distinct population.

These two models are explained in more detail in the next sections.

III. THE DUAL-POPULATION COMPETITIVE CO-EVOLUTIONARY APPROACH (DPPCP)

The general diagram of the DPPCP is given in Fig.4 and the pseudo-code of the proposed algorithm DPPCP is shown in Algorithm 1. There are two co-evolving populations: the first one (named A_p) is evolved by using the

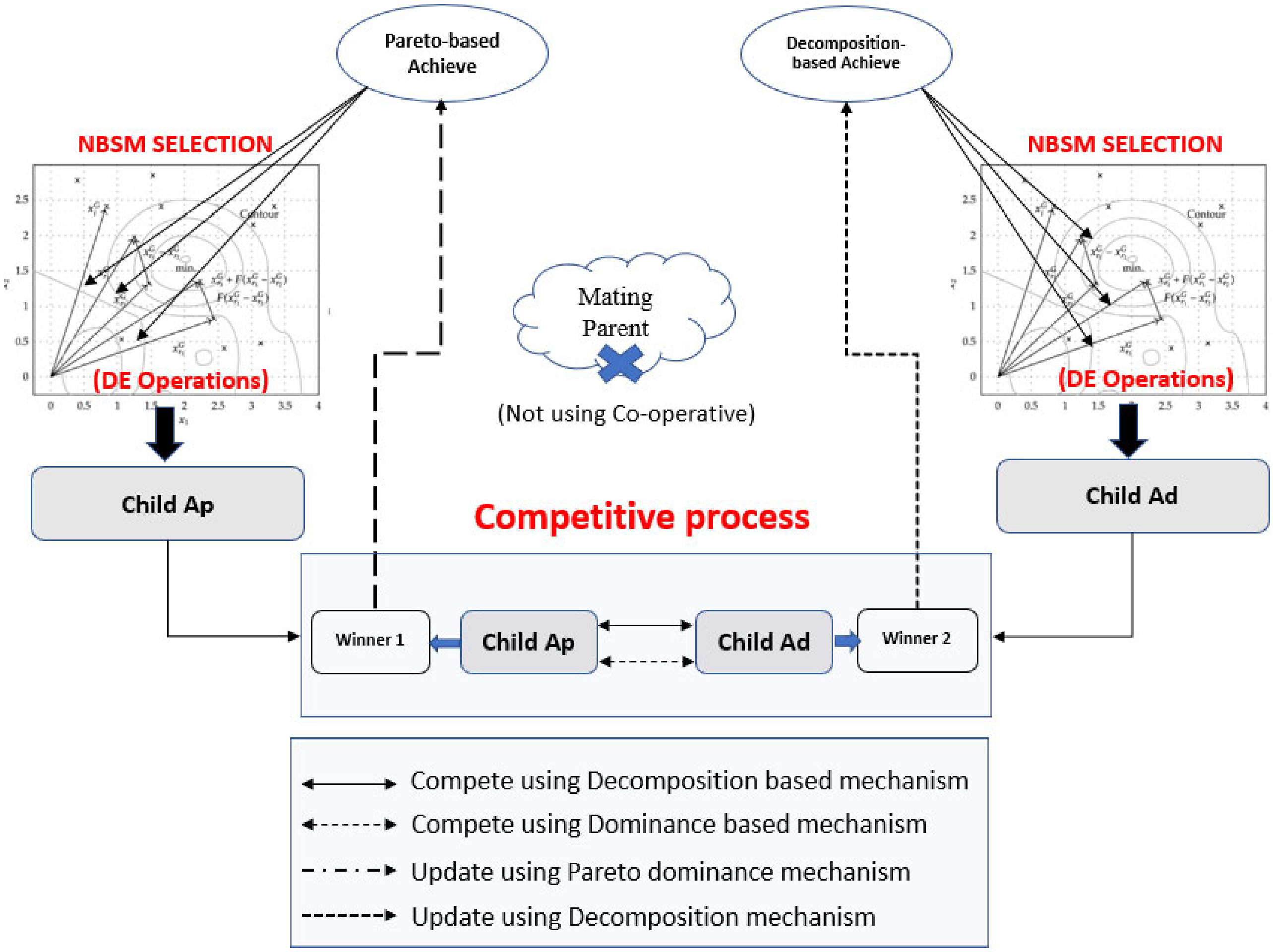


FIGURE 4. System architecture of the dual-population competitive co-evolutionary approach.

Pareto-based mechanism; the other one (named A_d) utilizes the decomposition-based mechanism to evolve. At each generation, we use a neighbor-based selection mechanism (NBSM) to select three candidate solutions from each of the populations. After that, we use differential evolution (DE) to create two offspring named $Child_{Ap}$ (i.e. the offspring in population A_p) and $Child_{Ad}$ (the offspring in population A_d). Next, we let $Child_{Ad}$ compete with $Child_{Ap}$ using Pareto dominance-based metrics and choose the winner to update A_p. Similarly, we let $Child_{Ap}$ compete with $Child_{Ad}$ using decomposition-based metrics and use the winner to update A_d. At the end of the co-evolution process, the final population is a combination of both A_p and A_d populations. The reason for this decision is that each of them uses a different optimal mechanism. While A_p uses true Pareto front, A_d utilizes idea point (a solution with the best objective values known since running the algorithm) as the best goal to achieve. The roles of the two populations are the same. Therefore, in order to preserve the good properties of both populations (i.e. diversity and convergence), we decided to keep both populations in the final selected population.

As mentioned above, there are two differences between the DPPCP model and the DPP model: First, in the DPPCP, we do not use a co-operative co-evolutionary mechanism. In other words, we have eliminated the mating parents step to generate the offspring. Instead, we use a competitive mechanism to make two offspring interact with each other. Second, we use the NBSM mechanism to select three solutions in each population and use them to create two separate offspring.

In general, the model is divided into four main steps: *Initialization, NBSM selection, Competitive process, and Update population*.

A. INITIALIZATION

At the first step, A_p and A_d (with the same size N) are randomly generated. However, the distribution of individuals in the two populations is different. In A_d, N solutions are assigned to different N sub-regions. To make sure that there is only one solution for each sub-region, the algorithm divides the original region into N sub-regions (denoted as S_i) by using N uniformly distributed unit vectors denoted as λ_i (See Fig.5). Each λ_i will be identified corresponding to $solution_i$ (or each

Algorithm 1 DPPCP Algorithm

input : M: The number of generations.
T: The neighboring numbers
N: The population size
output: Final Population A_p and A_d

$[A_p, A_d] = initializePopulation()$
$W = InitializeUniformWeight()$
$B = InitializeNeighborhood()$
$Z^* = InitializeIdealPoint()$
$Z^{nad} = InitializeNadirPoint()$
$m \leftarrow 0$
while $m < M$ **do**
    $offspringAp \leftarrow \varnothing$
    for $i \leftarrow 1$ **to** N **do**
        $Child_{Ap}, Child_{Ad} = NBSMSelection(A_p, A_d, i, Bi)$
        $Winner1 == CompeteDominate(Child_{Ap}, Child_{Ad})$
        $Winner2 == CompeteDecompostion(Child_{Ap}, Child_{Ad})$
        UpdateAp(Winner1, A_p);
        UpdateAd(Winner2, A_d);
        Update Z^* and Z^{nad}
        m++;
    end
end
Return $P \leftarrow A_p \cup A_d$

solution is assigned to only one sub-region). The algorithm utilizes the λ vectors to calculate the Euclidean distance between these vectors. Based on these distances, the algorithm can determine which sub-regions are the neighbours of a solution. In the next step (i.e. evolutionary step), when a new solution is created, it is necessary to determine which sub-region it belongs to. This is done based on the calculation of the distance between the new solution and the λ vectors. A sub-region will be selected if it contains the λ vector which is closest to the new solution. However, it should be noted that, instead of including this new solution in this sub-region directly, a competition between the new solution and the existing solution in this sub-region will take place. The better solution (based on the fitness functions) will be selected to assign to this sub-region. By this way, there is exactly one solution in each sub-region and A_d is distributed evenly (i.e. diversity) in the objective space. Unlike A_d, A_p does not rely on the even spread of N unit vectors. Therefore, N solutions in A_p are randomly assigned to N sub-regions (Fig.6 gives an intuitive example of the distribution of solutions in each population. A_p does not contain any solution in sub-regions 0, 1, 3, 5, while sub-regions 2 and 4 contain more than one solutions). This leads to a situation that a sub-region may either not have any solution or contain more than one solutions. Next, we find the T closest neighborhood sub-regions for each solution(by using the Euclidean distance). These neighborhoods play a vital role in the next steps.

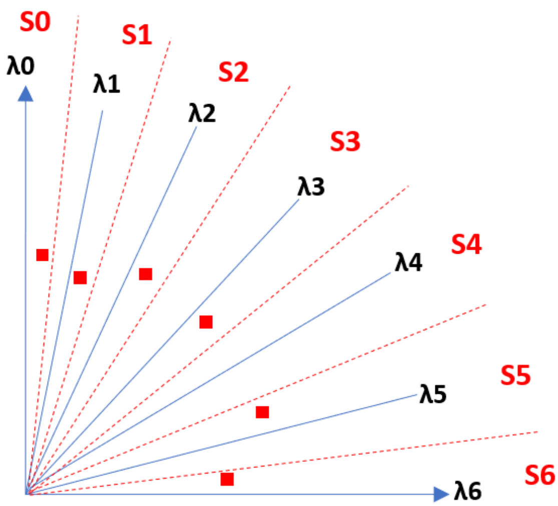


FIGURE 5. A simple illustration of initializing the population for A_d.

Algorithm 2 NBSMSelection(A_p,A_d,i,B_i)

input : A_p: the pareto-based population
A_d: the decomposition-based population
i: the current sub-region index
B_i: a set contains neighborhood indexes of the current sub-region.
T: the neighborhood size;
N: the population size
output: Q: Two mating parent

$P_1 = MatSelectionAp()$
$P_2 = MatSelectionAd()$
$Solution1 = A_p[r_{1p}]$
$Solution2 = A_p[r_{2p}]$
if $SubRegion(r_{1p})$ *does not contain any solutions* **then**
    $Solution1 = A_d[r_{1p}]$
end
if $SubRegion(r_{2p})$ *does not contain any solutions* **then**
    $Solution2 = A_d[r_{2p}]$
end
$Child_{A_p} = DE(Solution1, Solution2, A_p[i])$
$Child_{Ad} = DE(A_d[r_{1d}], A_d[r_{2d}], A_d[i])$
Return Q = $(Child_{A_p}, Child_{A_d})$

B. THE NEIGHBOR-BASED SELECTION MECHANISM (NBSM)

In [6] the authors showed that: when solving continuous MOPs, in some mild conditions, neighborhood solutions should have similar structures. This means the neighborhood information is very important and it should be better if we use this important information in the orientation process for new solutions. For that reason, we prefer to choose mating parents

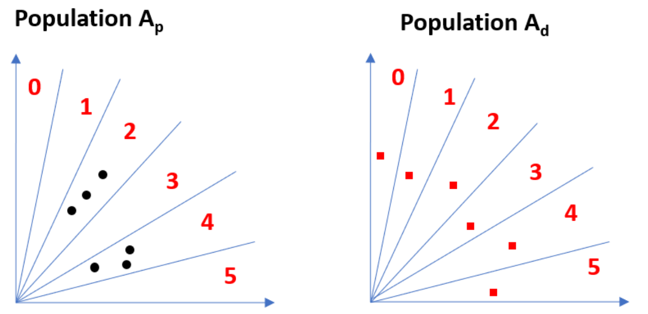


FIGURE 6. A simple illustration of distribution of solutions in sub-regions.

from several neighboring sub-regions. As for the traditional DE operator [39], x_{r1}^G and x_{r2}^G (two components of the direction vectors in Eq.2) are randomly selected from the whole population. This random mating selection mechanism can explore well. However, since there is no guidance information towards the Pareto set, it may lead to a degeneration problem. The RMS mechanism in [31] improved this weakness by using more information from neighboring sub-regions than from the whole population. However, as mentioned above, a drawback of the RMS mechanism is that the probability of selecting a sub-region in A_p that contains at least a solution is relatively low. At that time, the RMS borrows an alternate solution in the A_d, which can lead to an imbalance between two populations of the co-evolutionary process. This is the reason why we propose another selection mechanism (i.e. NBSM).

Algorithm 3 MatSelectionAd(A_d, i, B_i)

input : A_d: the decomposition-based population
i: the current sub-region index
B_i: a set contains neighborhood indexes of the current sub-region.
T: the neighborhood size;
N: the population size
θ: the neighborhood selection probability
output: [I,J]: Two sub-region indexes.

if *rand* $< \theta$ **then**
  Randomly select two indices I, J, from B_i
end
else
  Randomly select two indices I, J from $\{1, 2, \ldots, N\}$
end

The pseudo-code of the NBSM mechanism is presented in Algorithm 2.

There are two underlying principles of the NBSM. Firstly, we want fairness in choosing the number of solutions to hybridize in the coevolutionary step. Secondly, the three chosen solutions used in the DE operator must be on the same population (in order to avoid the phenomenon as shown in Figure 3).

Algorithm 4 MatSelectionAp(A_p,i,B_i)

input : A_p: the pareto-based population
i: the current sub-region index
B_i: a set contains neighborhood indexes of current sub-region.
T: the neighborhood size;
N: the population size
θ: the neighborhood selection probability
output: [I,J]: Two sub-region indexes.

listNeighborAp $\leftarrow \varnothing$
if *rand* $< \theta$ **then**
  // Select two sub-region indexes in A_p
  for $i \leftarrow 0$ **to** T **do**
    for $j \leftarrow 0$ **to** N **do**
      if $A_p[j] \in B_i[i]$ **then**
        Add j to *listNeighborAp*;
      end
    end
  end
  while *size of listNeighborAp* < 2 **do**
    Randomly select an index r from $1, 2 \ldots, N$
    Add r to *listNeighborAp*
  end
  Randomly select two indices J and K from *listNeighborAp*.
end
else
  Randomly select an index from $\{1, 2, \ldots, N\}$
end

To generate new offspring (i.e. $Child_{Ap}$ or $Child_{Ad}$), we imitate the idea from MOEA/D-DE [12]. Specifically, in MOEA/D-DE, a solution $\bar{y}$ is generated from x^{r1} (i.e. the current solution), x^{r2} and x^{r3} according to Eq.2, and a new solution is generated by a mutation operator on $\bar{y}$ with a small probability, according to Eq.3

$$\bar{y}_k = \begin{cases} x_k^{r1} + F * (x_k^{r2} - x_k^{r3}), & \text{with probability} < \text{CR} \\ x_k^{r1}, & \text{with probability 1-CR} \end{cases} \tag{2}$$

where CR and F are two control parameters

$$y_k = \begin{cases} \bar{y}_k + \sigma_k * (u_k - l_k), & \text{with probability } p_m \\ \bar{y}_k, & \text{with probability 1-}p_m \end{cases} \tag{3}$$

$$\sigma_k = \begin{cases} (2 * rand)^{\frac{1}{\eta}+1} - 1, & \text{if } rand < 0.5 \\ 1 - (2 - 2 * rand)^{\frac{1}{\eta}+1}, & \text{otherwise} \end{cases} \tag{4}$$

where *rand* is a uniform random number in [0,1]; p_m is the mutation rate; u_k and l_k are the upper and lower bound of the k^{th} decision variable, respectively.

Another major difference between the two RMS and NBSM mechanisms is the solution selection procedure in A_p. For each small partition, we conduct a search across the entire T neighborhood sub-regions instead of just choosing

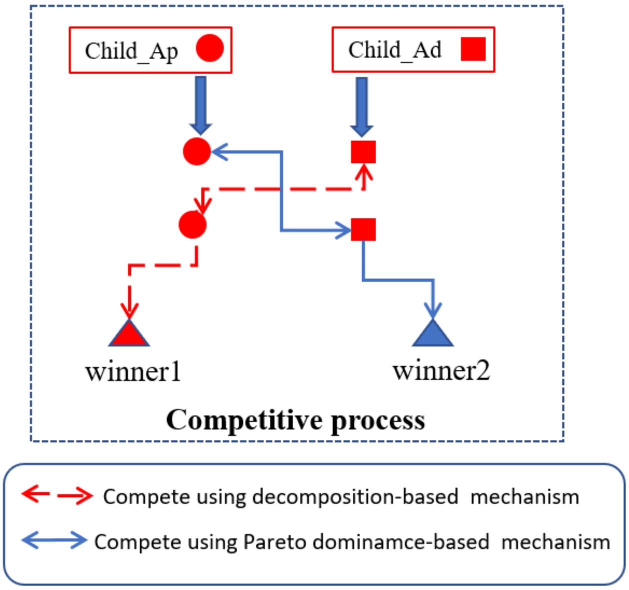


FIGURE 7. **The competitive mechanism.**

a random sub-region as in the RMS mechanism. This way, the probability of finding three solutions is much higher. In the case of any individuals cannot be found in the neighborhood sub-regions we borrow from the A_d.

C. THE COMPETITIVE CO-EVOLUTIONARY MECHANISM (COMPETITIVE PROCESS)

In this step, two offspring solutions $Child_{Ap}$ and $Child_{Ad}$ in each population are selected to participate in tournaments. Fig.7 gives an intuitive explanation of this mechanism. Specifically, $Child_{Ap}$ competes against $Child_{Ad}$ using the Pareto-based rule (i.e. *CompeteDominance* method in Algorithm 1), a winner is selected to update A_p. Meanwhile, $Child_{Ad}$ competes against $Child_{Ap}$ using the decomposition-based rule (i.e. *CompeteDecomposition* method in Algorithm 1). We would like to highlight the benefits of competitive co-evolution by considering two possible cases:

(+) If two winning solutions belong to two different populations. It means we have a solution which has good convergence and another with good diversity. This is what we expected.

(+) If two winning solutions belong to the same population (e.g. A_p). It means, one solution in A_p has the convergence better than its competitor (this is normal). Along with that, the remaining solution in A_p is more diverse than its contestant in A_d. It is interesting to note that, in this case, we have one mating solution which has good convergence and the other which not only good convergence but also excellent diversity. Therefore, the offspring can get both of good traits. This is very different from the cooperative co-evolution in DPP.

D. UPDATE POPULATION

The update mechanism in each population will be different. In research [31] the authors only updated Offspring to the nearest sub-region. This is to ensure the population diversity, but the probability that this solution will replace the sub-region is rather small (because it only compares to only one sub-region, while there are some other sub-regions having much worse solutions). This may lead to the possibility that convergence will decrease. To improve this disadvantage, we used the updated idea of the MOEA/D algorithm for both populations. Specifically, we will iterate through all solutions in the neighborhood sub-regions in question and update them on a limited number of times (this helps to avoid having many similar solutions as well as speeding up the convergence of the population).

Specifically, the *UpdateAd (winner2, A_d)* method in algorithm 1 is used in the same way as the MOEA/D-DE algorithm; whereas the *UpdateAp (winner1, A_p)* method is shown in the algorithm 5.

It can be easy to see that, the way to update A_p is different from the one in the NSGA-II algorithm. We update A_p as soon as the winner dominates a solution in A_p and conducting *Ranking* and *CrowdingDistanceSelection* methods every time updating offspring. Meanwhile, the NSGA-II uses a list to store all of the offspring and performing ranking when the loop is finished.

Algorithm 5 UpdateAp (winner1, A_p)

input : *LimitedNum*: The limited number of updated times
N: the Population size
output: A_p after updated

isNonDominte= False ; *Flag* = False;
for $i \leftarrow 0$ **to** N **do**
  if *winner*1 *dominate* $A_p[j]$ **then**
    Update A_p[j] by winner1;
    Flag = True;
    Num++;
    if *Num==LimitedNum* **then**
      Break;
    end
    else if *winner1 and $A_p[j]$ are nondominated* **then**
      isNonDominte = True;
    end
    ;
  end
end
if *isNonDominte = True and Flag = False* **then**
  Add *winner1* to A_p
  A_p = **crowdingDistanceSelection**(A_p)
end
else
  Randomly select an index from $\{1, 2, \ldots, N\}$
end

It is highlighted that the new offspring requires being assigned to a certain sub-region. In this paper, to determine the suitable sub-region, the authors first measure the distance between its unit vector and the offspring's objective vector. Which sub-region has the minimum distance will be selected to contain the offspring. It is also noted that the scaling of each objective function differs from each other. It means the value of objective functions can vary from low to high one. This leads to a situation that in the calculation of the Euclid distance, the low objective value is of no importance. Therefore, it is essential to standardize the objective functions in the same range of values. Here we perform the standardization [2] to the interval [0, 1] as follows:

$$f_i^{norm} = \frac{f_i - Z_i^*}{Z_i^{nad} - Z_i^*} \tag{5}$$

where i $\in \{1, 2, \ldots, m\}$; m is the number of objective functions.

The above analysis shows that basically A_p is similar to NSGA-II; and A_d is similar to MOEA/D in terms of how they work, but the details of the implementation are quite different. In the experimental results, we will further analyze these differences.

IV. EXPERIMENTAL STUDIES

In this step, we will perform experiments with the proposed algorithm to clarify some following problems. First, we compare the proposed method with some baseline algorithms (i.e. NSGA-II and MOEA/D-DE) and the state-of-the-art algorithms (ED/DPP in [31] and DPP2 in [32]). Via the comparison results, we could see how good the performance of the proposed method when comparing to the others. Next, we develop a variant named *DPPCP-Variant1* and compare it to the DPPCP to know the effects of competitiveness. In order to know the effects of the NBSM mechanism, we create two variants named *DPPCP-Variant2* and *DPPCP-Variant3* and compare them to DPPCP. Finally, to know the interaction between two co-evolving populations, we create two other variants named *DPPCP-Ap* and *DPPCP-Ad*. After that, we conduct three test cases between NSGA-II and A_p; MOEAD and A_d; and *DPPCP-Ap* and *DPPCP-Ad*.

A. TEST PROBLEMS

In this paper, 32 test instances (ZDT1 to ZDT6, UF1 to UF10, WFG1 to WFG9 and DTLZ1 to DTLZ7) are used as benchmark problems. Among these, UF1 to UF7 [38] and WFG1 to WFG9 [36] and ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 [40] are the bi-objective test problems; and UF8, to UF10 and DTLZ1 to DTLZ7 are tri-objective benchmarks. More detailed properties of DTLZ problems are summarized in Table 1.

B. PERFORMANCE METRICS

When measuring the performance of MOEAs, two common factors considered are convergence (the closeness between the obtained solutions set and the true Pareto optimal front) and diversity (the spread and distribution of solutions on the Pareto front). There exists a number of performance metrics to evaluate these factors such as generational distance (GD) [10], spacing metric (SP) [10], hypervolume (HV) [11] and inverted generational distance (IGD) [12], inverted generational distance plus (IGD+) or stability [13]. Fig.8 shows the ability of each metric. The GD and SP metrics evaluate the convergence and uniformity respectively. Meanwhile, the IGD as well as the HV metrics measure are not only the convergence but also the diversity of a solution set.

In this paper, the IGD and HV are chosen as the main metrics. It is worth highlighting that the quality of a solution depends on the HV value. The greater the HV value is, the better the solution is. Besides, the lower the value of the IGD is, the better.

TABLE 1. The DTLZ series test instances.

Name	Variable number	Objective number	Geometry
DTLZ1	7	3	Linear
DTLZ2	12	3	Concave
DTLZ3	12	3	Concave
DTLZ4	12	3	Concave
DTLZ5	12	3	-
DTLZ6	12	3	-
DTLZ7	22	3	Disconnected

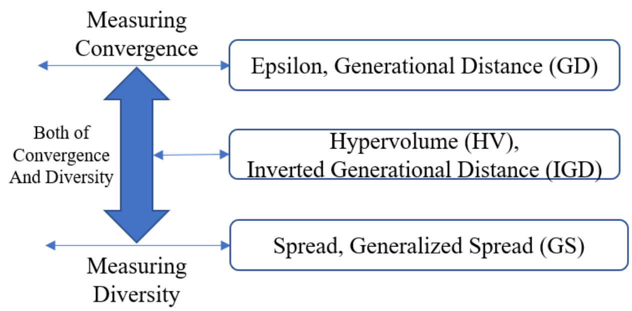


FIGURE 8. Several performance metrics used in MOEAs.

TABLE 2. The parameter setting of the MOEAs.

MOEAs	Parameters settings
NSGA-II	pc=0.9, pm=1/$n_{variables}$; $\mu_c = 20$; $\mu_m = 20$
MOEA/D-DE	pm=1/ $n_{variables}$, $\mu_m = 20$, CR=1.0, F=0.5, $\sigma = 0.9$, T=20, $'rand/1/bin'$; $n_{replaced}$=2

C. PARAMETERS SETTINGS OF MOEAs

Given in table 2 are the parameters of the NSGA-II and MOEA/D-DE. In each test trial, every algorithm is independently run 20 times. The population size (N) is set to 300 and the termination criterion of an algorithm is a predefined number of generation (M), which is constantly set to 300.000.

D. DPPCP AGAINST BASELINE ALGORITHMS

As mentioned above, the DPPCP uses two populations, one based on the Pareto mechanism (using the NSGA-II

TABLE 3. Performance comparisons between the DPPCP with baseline algorithms using HV metric.

	NSGAII	MOEAD	DPPCP
ZDT1	$6.647858e-01_{5.3e-05}$	$6.648386e-01_{4.7e-05}$	$6.655793e-01_{0.0e+00}$
ZDT2	$3.314961e-01_{3.3e-05}$	$3.315780e-01_{5.1e-05}$	$3.321892e-01_{0.0e+00}$
ZDT3	$5.168567e-01_{1.2e-05}$	$5.162200e-01_{1.1e-05}$	$5.170450e-01_{0.0e+00}$
ZDT4	$6.648548e-01_{2.2e-05}$	$6.649686e-01_{9.8e-06}$	$6.655913e-01_{0.0e+00}$
ZDT6	$4.031456e-01_{1.4e-04}$	$4.047282e-01_{1.7e-08}$	$4.053136e-01_{0.0e+00}$
UF1	$5.484432e-01_{1.9e-02}$	$6.636184e-01_{1.3e-04}$	$6.639659e-01_{0.0e+00}$
UF	$6.396909e-01_{3.6e-03}$	$6.572483e-01_{2.8e-03}$	$6.616692e-01_{0.0e+00}$
UF3	$4.571061e-01_{3.4e-02}$	$6.569119e-01_{9.5e-03}$	$5.965543e-01_{0.0e+00}$
UF4	$2.726602e-01_{3.2e-04}$	$2.458777e-01_{5.8e-03}$	$2.551797e-01_{0.0e+00}$
UF5	$2.374327e-01_{2.8e-02}$	$3.632481e-02_{6.5e-02}$	$8.999985e-02_{0.0e+00}$
UF6	$2.778372e-01_{4.1e-02}$	$2.024907e-01_{7.9e-02}$	$1.725421e-01_{0.0e+00}$
UF7	$4.195309e-01_{6.1e-02}$	$4.952861e-01_{2.4e-03}$	$4.954061e-01_{0.0e+00}$
UF8	$1.069511e-01_{2.2e-02}$	$3.295556e-01_{2.3e-02}$	$3.623309e-01_{0.0e+00}$
UF9	$4.296275e-01_{1.2e-01}$	$6.014756e-01_{6.2e-02}$	$5.565798e-01_{0.0e+00}$
UF10	$5.292133e-04_{1.1e-03}$	$5.944409e-02_{2.7e-02}$	$9.315613e-02_{0.0e+00}$
WFG1	$6.336915e-01_{3.8e-04}$	$6.347114e-01_{2.2e-04}$	$6.370891e-01_{0.0e+00}$
WFG2	$5.652985e-01_{1.2e-05}$	$5.646742e-01_{1.3e-05}$	$5.654082e-01_{0.0e+00}$
WFG3	$4.978645e-01_{4.7e-05}$	$4.980009e-01_{6.1e-06}$	$4.987931e-01_{0.0e+00}$
WFG4	$2.214379e-01_{8.9e-05}$	$2.212084e-01_{1.1e-04}$	$2.220795e-01_{0.0e+00}$
WFG5	$1.982352e-01_{8.4e-05}$	$1.988567e-01_{2.2e-03}$	$1.989817e-01_{0.0e+00}$
WFG6	$2.104230e-01_{3.2e-03}$	$2.128723e-01_{4.5e-06}$	$2.136358e-01_{0.0e+00}$
WFG7	$2.129812e-01_{4.4e-05}$	$2.128570e-01_{8.1e-06}$	$2.136157e-01_{0.0e+00}$
WFG8	$1.676837e-01_{2.3e-02}$	$1.704603e-01_{2.4e-02}$	$2.114787e-01_{0.0e+00}$
WFG9	$2.433160e-01_{5.0e-04}$	$2.438831e-01_{5.5e-05}$	$2.449073e-01_{0.0e+00}$
DTLZ1	$7.952908e-01_{2.2e-03}$	$7.849171e-01_{2.9e-04}$	$8.027772e-01_{0.0e+00}$
DTLZ2	$4.145143e-01_{2.7e-03}$	$4.185581e-01_{8.5e-04}$	$4.294453e-01_{0.0e+00}$
DTLZ3	$4.237451e-01_{2.2e-03}$	$4.192297e-01_{9.9e-04}$	$4.298513e-01_{0.0e+00}$
DTLZ4	$4.135101e-01_{1.9e-03}$	$4.121623e-01_{2.2e-02}$	$4.245304e-01_{0.0e+00}$
DTLZ5	$9.540642e-02_{3.0e-05}$	$9.469685e-02_{7.7e-06}$	$9.579286e-02_{0.0e+00}$
DTLZ6	$6.407350e-02_{1.0e-02}$	$9.566239e-02_{7.8e-07}$	$9.678412e-02_{0.0e+00}$
DTLZ7	$3.120551e-01_{1.2e-03}$	$2.770128e-01_{1.6e-02}$	$3.103702e-01_{0.0e+00}$

TABLE 4. Performance comparisons between the DPPCP with baseline algorithms using IGD metric.

	NSGAII	MOEAD	DPPCP
ZDT1	$5.788071e-05_{4.8e-06}$	$5.556843e-05_{3.3e-07}$	$3.093510e-05_{0.0e+00}$
ZDT2	$5.968199e-05_{2.9e-06}$	$4.660528e-05_{4.7e-08}$	$3.241247e-05_{0.0e+00}$
ZDT3	$4.146423e-05_{1.9e-06}$	$8.838159e-05_{7.9e-07}$	$2.623904e-05_{0.0e+00}$
ZDT4	$5.699439e-05_{2.2e-06}$	$5.906846e-05_{5.0e-07}$	$3.104248e-05_{0.0e+00}$
ZDT6	$7.378607e-05_{4.2e-06}$	$4.628025e-05_{6.2e-09}$	$3.135962e-05_{0.0e+00}$
UF1	$3.546947e-03_{6.0e-04}$	$6.903693e-05_{5.0e-06}$	$5.686509e-05_{0.0e+00}$
UF	$1.066904e-03_{2.9e-04}$	$3.608501e-04_{2.4e-04}$	$1.700001e-04_{0.0e+00}$
UF3	$7.154636e-03_{1.8e-03}$	$1.702418e-04_{1.5e-04}$	$1.820430e-03_{0.0e+00}$
UF4	$1.358224e-03_{2.4e-05}$	$1.940253e-03_{2.3e-04}$	$1.676999e-03_{0.0e+00}$
UF5	$4.394656e-02_{8.3e-03}$	$6.607502e-02_{1.7e-02}$	$1.547979e-01_{0.0e+00}$
UF6	$8.797878e-03_{3.8e-03}$	$3.479782e-03_{8.8e-03}$	$2.268674e-02_{0.0e+00}$
UF7	$1.859278e-03_{1.7e-03}$	$1.085036e-04_{2.4e-05}$	$1.185823e-04_{0.0e+00}$
UF8	$2.981191e-03_{1.7e-04}$	$9.841127e-04_{4.3e-04}$	$8.438084e-04_{0.0e+00}$
UF9	$2.732983e-03_{2.0e-03}$	$2.165702e-03_{1.5e-03}$	$2.269156e-03_{0.0e+00}$
UF10	$5.161469e-03_{3.7e-03}$	$4.986122e-03_{6.2e-04}$	$4.938679e-03_{0.0e+00}$
WFG1	$3.200666e-04_{2.3e-05}$	$2.702204e-04_{2.7e-05}$	$7.811839e-05_{0.0e+00}$
WFG2	$1.174109e-04_{1.1e-05}$	$6.057198e-04_{6.6e-06}$	$8.474260e-05_{0.0e+00}$
WFG3	$6.512669e-05_{3.3e-06}$	$5.476262e-05_{9.3e-08}$	$3.343249e-05_{0.0e+00}$
WFG4	$5.717426e-05_{2.8e-06}$	$6.238275e-05_{1.3e-06}$	$3.390037e-05_{0.0e+00}$
WFG5	$9.330129e-04_{5.7e-07}$	$9.337936e-04_{3.8e-07}$	$9.303355e-04_{0.0e+00}$
WFG6	$1.122812e-04_{6.5e-05}$	$9.139109e-05_{1.4e-07}$	$5.394634e-05_{0.0e+00}$
WFG7	$3.877603e-05_{1.7e-06}$	$4.054086e-05_{1.9e-08}$	$2.255773e-05_{0.0e+00}$
WFG8	$2.747423e-03_{2.2e-03}$	$3.174261e-03_{2.4e-03}$	$8.095268e-04_{0.0e+00}$
WFG9	$4.337451e-05_{5.0e-06}$	$4.071753e-05_{1.1e-07}$	$2.269694e-05_{0.0e+00}$
DTLZ1	$3.201110e-04_{1.9e-05}$	$3.474762e-04_{1.3e-06}$	$2.526425e-04_{0.0e+00}$
DTLZ2	$4.355620e-04_{2.4e-05}$	$4.306742e-04_{3.0e-06}$	$3.343429e-04_{0.0e+00}$
DTLZ3	$6.953162e-04_{2.1e-05}$	$7.211632e-04_{6.2e-06}$	$5.305681e-04_{0.0e+00}$
DTLZ4	$7.724318e-04_{1.2e-04}$	$7.911537e-04_{1.3e-04}$	$5.417136e-04_{0.0e+00}$
DTLZ5	$6.185971e-06_{4.8e-07}$	$1.516055e-05_{1.1e-07}$	$3.674228e-06_{0.0e+00}$
DTLZ6	$3.789467e-04_{2.9e-04}$	$3.453845e-05_{3.1e-08}$	$8.785615e-06_{0.0e+00}$
DTLZ7	$1.204211e-03_{6.4e-05}$	$2.612999e-03_{1.7e-04}$	$1.181486e-03_{0.0e+00}$

algorithm), the other based on the decomposition mechanism (using MOEAD/DE algorithm). In order to assess the effectiveness of using the co-evolutionary mechanism, we compare the proposed algorithm with these two baseline algorithms (i.e. NSGA-II and MOEAD/DE). Table 3 and Table 4 provide the performance comparisons of DPPCP, MOEA/D-DE and NSGA-II on 32 test instances, with respect to the IGD and the HV metrics, respectively. Based on experimental results, we can see that DPPCP achieves a better outcome than both NSGA-II and MOEA-D/DE. It wins in 26 out of 32 comparisons using the HV and the IGD metrics. It is worth noting that although NSGA-II is the worst among

TABLE 5. Performance comparisons between the DPPCP with state-of-art algorithms using HV metric.

	ED/DPP	DPP2	DPPCP
ZDT1	$6.503543e-01_{1.9e-02}$	$6.648341e-01_{4.6e-05}$	$6.655793e-01_{0.0e+00}$
ZDT2	$3.062317e-01_{7.8e-02}$	$3.315689e-01_{3.8e-05}$	$3.321892e-01_{0.0e+00}$
ZDT3	$5.137847e-01_{2.3e-03}$	$5.162233e-01_{9.3e-06}$	$5.170450e-01_{0.0e+00}$
ZDT4	$0.000000e+00_{0.0e+00}$	$6.649716e-01_{1.0e-05}$	$6.655913e-01_{0.0e+00}$
ZDT6	$4.052969e-01_{1.9e-05}$	$4.047282e-01_{5.4e-08}$	$4.053136e-01_{0.0e+00}$
UF1	$6.095147e-01_{1.5e-02}$	$6.635853e-01_{1.4e-04}$	$6.639659e-01_{0.0e+00}$
UF	$6.071246e-01_{9.4e-03}$	$6.568236e-01_{1.6e-03}$	$6.616692e-01_{0.0e+00}$
UF3	$4.299732e-01_{5.4e-02}$	$6.521115e-01_{1.5e-02}$	$5.965543e-01_{0.0e+00}$
UF4	$2.384583e-01_{7.3e-03}$	$2.438856e-01_{4.2e-03}$	$2.551797e-01_{0.0e+00}$
UF5	$1.642984e-03_{6.7e-03}$	$1.208023e-01_{9.1e-02}$	$8.999985e-02_{0.0e+00}$
UF6	$9.514643e-02_{5.0e-02}$	$2.001687e-01_{8.5e-02}$	$1.725421e-01_{0.0e+00}$
UF7	$4.701857e-01_{8.9e-03}$	$4.958239e-01_{8.2e-04}$	$4.954061e-01_{0.0e+00}$
UF8	$2.564091e-01_{3.3e-02}$	$3.271188e-01_{1.6e-02}$	$3.623309e-01_{0.0e+00}$
UF9	$5.107049e-01_{3.8e-02}$	$5.646465e-01_{3.1e-02}$	$5.565798e-01_{0.0e+00}$
UF10	$0.000000e+00_{0.0e+00}$	$7.045457e-02_{1.4e-02}$	$9.315613e-02_{0.0e+00}$
WFG1	$4.500378e-01_{6.6e-02}$	$6.347178e-01_{3.6e-04}$	$6.370891e-01_{0.0e+00}$
WFG2	$5.635474e-01_{7.7e-04}$	$5.646761e-01_{1.2e-05}$	$5.654082e-01_{0.0e+00}$
WFG3	$4.922166e-01_{2.6e-03}$	$4.979994e-01_{4.2e-06}$	$4.987931e-01_{0.0e+00}$
WFG4	$2.117675e-01_{1.3e-03}$	$2.211018e-01_{1.4e-04}$	$2.220795e-01_{0.0e+00}$
WFG5	$1.995414e-01_{2.1e-03}$	$1.987193e-01_{2.8e-03}$	$1.989817e-01_{0.0e+00}$
WFG6	$2.109626e-01_{1.5e-03}$	$2.128725e-01_{3.8e-06}$	$2.136358e-01_{0.0e+00}$
WFG7	$2.129709e-01_{1.8e-04}$	$2.128533e-01_{6.8e-06}$	$2.136157e-01_{0.0e+00}$
WFG8	$1.576003e-01_{1.8e-02}$	$1.732010e-01_{2.6e-02}$	$2.114787e-01_{0.0e+00}$
WFG9	$2.412725e-01_{1.6e-04}$	$2.438372e-01_{4.6e-05}$	$2.449073e-01_{0.0e+00}$
DTLZ1	$3.981344e-02_{1.6e-01}$	$7.848863e-01_{1.7e-04}$	$8.027772e-01_{0.0e+00}$
DTLZ2	$4.325969e-01_{1.5e-03}$	$4.185189e-01_{9.8e-04}$	$4.294453e-01_{0.0e+00}$
DTLZ3	$8.816880e-02_{1.7e-01}$	$4.182285e-01_{1.1e-03}$	$4.298513e-01_{0.0e+00}$
DTLZ4	$4.198804e-01_{1.5e-03}$	$4.065588e-01_{3.0e-02}$	$4.245304e-01_{0.0e+00}$
DTLZ5	$8.724304e-02_{2.3e-03}$	$9.469878e-02_{9.7e-06}$	$9.579286e-02_{0.0e+00}$
DTLZ6	$9.656330e-02_{1.2e-05}$	$9.566300e-02_{8.0e-07}$	$9.678412e-02_{0.0e+00}$
DTLZ7	$3.129314e-01_{2.5e-03}$	$2.800935e-01_{6.8e-03}$	$3.103702e-01_{0.0e+00}$

TABLE 6. Performance comparisons between the DPPCP with state-of-art algorithms using IGD metric.

	ED/DPP	DPP2	DPPCP
ZDT1	$3.945162e-04_{6.4e-04}$	$5.554228e-05_{2.2e-07}$	$3.093510e-05_{0.0e+00}$
ZDT2	$4.602254e-05_{3.6e-05}$	$4.661050e-05_{7.3e-08}$	$3.241247e-05_{0.0e+00}$
ZDT3	$6.907624e-05_{8.2e-05}$	$8.860561e-05_{8.5e-07}$	$2.623904e-05_{0.0e+00}$
ZDT4	$3.273414e-01_{2.5e-01}$	$5.866687e-05_{4.8e-07}$	$3.104248e-05_{0.0e+00}$
ZDT6	$3.147239e-05_{8.0e-07}$	$4.628070e-05_{6.9e-09}$	$3.135962e-05_{0.0e+00}$
UF1	$1.219677e-03_{4.7e-04}$	$6.864305e-05_{3.0e-06}$	$5.686509e-05_{0.0e+00}$
UF	$1.983361e-03_{6.9e-04}$	$4.221505e-04_{1.3e-04}$	$1.700001e-04_{0.0e+00}$
UF3	$6.117055e-03_{1.7e-03}$	$2.379576e-04_{5.5e-04}$	$1.820430e-03_{0.0e+00}$
UF4	$2.055704e-03_{3.4e-04}$	$1.955396e-03_{2.4e-04}$	$1.676999e-03_{0.0e+00}$
UF5	$1.319009e-01_{4.5e-02}$	$6.085614e-02_{2.8e-02}$	$1.547979e-01_{0.0e+00}$
UF6	$1.023728e-02_{2.6e-03}$	$6.511947e-03_{4.9e-03}$	$2.268674e-02_{0.0e+00}$
UF7	$7.577647e-04_{4.1e-04}$	$1.073177e-04_{3.7e-05}$	$1.185823e-04_{0.0e+00}$
UF8	$1.105829e-03_{3.2e-04}$	$1.033388e-03_{2.1e-04}$	$8.438084e-04_{0.0e+00}$
UF9	$2.296300e-03_{2.4e-04}$	$2.186628e-03_{1.5e-04}$	$2.269156e-03_{0.0e+00}$
UF10	$1.254666e-02_{4.0e-03}$	$4.860551e-03_{5.9e-04}$	$4.938679e-03_{0.0e+00}$
WFG1	$4.091125e-03_{2.0e-03}$	$2.672370e-04_{1.8e-05}$	$7.811839e-05_{0.0e+00}$
WFG2	$3.898765e-04_{1.9e-04}$	$6.061874e-04_{2.3e-05}$	$8.474260e-05_{0.0e+00}$
WFG3	$1.837856e-04_{8.8e-05}$	$5.480337e-05_{3.4e-08}$	$3.343249e-05_{0.0e+00}$
WFG4	$2.109915e-04_{2.9e-05}$	$6.344580e-05_{1.9e-06}$	$3.390037e-05_{0.0e+00}$
WFG5	$9.304804e-04_{2.1e-06}$	$9.328236e-04_{8.8e-07}$	$9.303355e-04_{0.0e+00}$
WFG6	$1.249467e-04_{9.1e-05}$	$9.143875e-05_{2.0e-07}$	$5.394634e-05_{0.0e+00}$
WFG7	$2.832908e-05_{3.6e-06}$	$4.054432e-05_{2.5e-08}$	$2.255773e-05_{0.0e+00}$
WFG8	$3.479606e-03_{9.6e-04}$	$3.172940e-03_{2.8e-03}$	$8.095268e-04_{0.0e+00}$
WFG9	$5.823859e-05_{3.0e-06}$	$4.076749e-05_{3.5e-07}$	$2.269694e-05_{0.0e+00}$
DTLZ1	$2.274694e-02_{1.4e-02}$	$3.471784e-04_{1.4e-06}$	$2.526425e-04_{0.0e+00}$
DTLZ2	$3.282544e-04_{9.9e-06}$	$4.301280e-04_{1.9e-06}$	$3.343429e-04_{0.0e+00}$
DTLZ3	$1.647616e-01_{3.3e-01}$	$7.229934e-04_{7.5e-06}$	$5.305681e-04_{0.0e+00}$
DTLZ4	$4.906612e-04_{2.2e-05}$	$8.454214e-04_{2.2e-04}$	$5.417136e-04_{0.0e+00}$
DTLZ5	$2.934037e-05_{6.4e-06}$	$1.518117e-05_{1.8e-07}$	$3.674228e-06_{0.0e+00}$
DTLZ6	$1.138830e-05_{1.1e-06}$	$3.454072e-05_{2.2e-08}$	$8.785615e-06_{0.0e+00}$
DTLZ7	$1.091168e-03_{5.5e-05}$	$2.612434e-03_{2.2e-04}$	$1.181486e-03_{0.0e+00}$

three candidates, it achieves the best IGD metric values on the UF4 and the UF5. Meanwhile, MOEA/D-DE obtains the best IGD metric values on the UF3, UF6, UF9, and WFG5. By contrast, DPPCP shows a poor result on the UF5 test instance. However, DPPCP shows better performance than the baseline algorithm on all the ZDT and DTLZ instances. These results indicate the effectiveness of DPPCP for achieving both convergence and diversity criteria.

E. DPPCP AGAINST STATE-OF-ART ALGORITHMS

In [31], the authors compared the DPP algorithm with some state-of-the-art algorithms (e.g. MOEA/D-FRRMAB [41],

TABLE 7. Performance comparisons between the DPPCP with DPPCP-Variant1 using HV metric.

	DPPCP-Variant1	DPPCP
ZDT1	$6.630161e-01_{7.2e-04}$	$6.655793e-01_{0.0e+00}$
ZDT2	$3.308929e-01_{2.9e-04}$	$3.321892e-01_{0.0e+00}$
ZDT3	$5.146396e-01_{6.9e-04}$	$5.170450e-01_{0.0e+00}$
ZDT4	$6.636102e-01_{2.7e-04}$	$6.655913e-01_{0.0e+00}$
ZDT6	$4.045542e-01_{1.9e-05}$	$4.053136e-01_{0.0e+00}$
UF1	$6.507534e-01_{1.6e-02}$	$6.639659e-01_{0.0e+00}$
UF	$6.571386e-01_{1.9e-03}$	$6.616692e-01_{0.0e+00}$
UF3	$5.920556e-01_{1.8e-02}$	$5.965543e-01_{0.0e+00}$
UF4	$2.470220e-01_{4.8e-03}$	$2.551797e-01_{0.0e+00}$
UF5	$1.429759e-01_{9.4e-02}$	$8.999985e-02_{0.0e+00}$
UF6	$2.563506e-01_{5.3e-02}$	$1.725421e-01_{0.0e+00}$
UF7	$4.927444e-01_{3.7e-03}$	$4.954061e-01_{0.0e+00}$
UF8	$3.124623e-01_{1.6e-02}$	$3.623309e-01_{0.0e+00}$
UF9	$6.119489e-01_{5.9e-02}$	$5.565798e-01_{0.0e+00}$
UF10	$5.908046e-02_{1.7e-02}$	$9.315613e-02_{0.0e+00}$
WFG1	$5.860027e-01_{7.6e-02}$	$6.370891e-01_{0.0e+00}$
WFG2	$5.636535e-01_{3.4e-04}$	$5.654082e-01_{0.0e+00}$
WFG3	$4.961655e-01_{4.1e-04}$	$4.987931e-01_{0.0e+00}$
WFG4	$2.194508e-01_{5.5e-04}$	$2.220795e-01_{0.0e+00}$
WFG5	$1.974580e-01_{2.2e-04}$	$1.989817e-01_{0.0e+00}$
WFG6	$2.113252e-01_{2.7e-04}$	$2.136358e-01_{0.0e+00}$
WFG7	$2.115485e-01_{2.1e-04}$	$2.136157e-01_{0.0e+00}$
WFG8	$1.716974e-01_{2.4e-02}$	$2.114787e-01_{0.0e+00}$
WFG9	$2.414078e-01_{6.9e-04}$	$2.449073e-01_{0.0e+00}$
DTLZ1	$7.868666e-01_{1.2e-03}$	$8.027772e-01_{0.0e+00}$
DTLZ2	$4.063181e-01_{1.5e-03}$	$4.294453e-01_{0.0e+00}$
DTLZ3	$4.069326e-01_{1.5e-03}$	$4.298513e-01_{0.0e+00}$
DTLZ4	$4.027093e-01_{2.5e-03}$	$4.245304e-01_{0.0e+00}$
DTLZ5	$9.467247e-02_{3.2e-05}$	$9.579286e-02_{0.0e+00}$
DTLZ6	$9.562214e-02_{8.9e-06}$	$9.678412e-02_{0.0e+00}$
DTLZ7	$2.873680e-01_{6.2e-03}$	$3.103702e-01_{0.0e+00}$

TABLE 8. Performance comparisons between the DPPCP with DPPCP-Variant1 using IGD metric.

	DPPCP-Variant1	DPPCP
ZDT1	$1.300515e-04_{2.0e-05}$	$3.093510e-05_{0.0e+00}$
ZDT2	$9.765445e-05_{2.2e-05}$	$3.241247e-05_{0.0e+00}$
ZDT3	$1.731261e-04_{2.4e-05}$	$2.623904e-05_{0.0e+00}$
ZDT4	$1.363398e-04_{1.4e-05}$	$3.104248e-05_{0.0e+00}$
ZDT6	$5.354539e-05_{5.6e-07}$	$3.135962e-05_{0.0e+00}$
UF1	$6.941814e-04_{8.8e-04}$	$5.686509e-05_{0.0e+00}$
UF	$3.218592e-04_{1.0e-04}$	$1.700001e-04_{0.0e+00}$
UF3	$2.212806e-03_{6.5e-04}$	$1.820430e-03_{0.0e+00}$
UF4	$2.022856e-03_{1.5e-04}$	$1.676999e-03_{0.0e+00}$
UF5	$6.183181e-02_{1.6e-02}$	$1.547979e-01_{0.0e+00}$
UF6	$5.570435e-03_{4.3e-03}$	$2.268674e-02_{0.0e+00}$
UF7	$3.427245e-04_{4.4e-04}$	$1.185823e-04_{0.0e+00}$
UF8	$1.005022e-03_{1.5e-04}$	$8.438084e-04_{0.0e+00}$
UF9	$1.387835e-03_{6.8e-04}$	$2.269156e-03_{0.0e+00}$
UF10	$4.734256e-03_{4.8e-04}$	$4.938679e-03_{0.0e+00}$
WFG1	$1.711538e-03_{1.9e-03}$	$7.811839e-05_{0.0e+00}$
WFG2	$1.185953e-03_{8.2e-05}$	$8.474260e-05_{0.0e+00}$
WFG3	$1.448185e-04_{2.6e-05}$	$3.343249e-05_{0.0e+00}$
WFG4	$1.497142e-04_{2.8e-05}$	$3.390037e-05_{0.0e+00}$
WFG5	$9.367849e-04_{2.1e-06}$	$9.303355e-04_{0.0e+00}$
WFG6	$2.140030e-04_{3.1e-05}$	$5.394634e-05_{0.0e+00}$
WFG7	$8.367173e-05_{1.1e-05}$	$2.255773e-05_{0.0e+00}$
WFG8	$2.419494e-03_{1.0e-03}$	$8.095268e-04_{0.0e+00}$
WFG9	$9.602084e-05_{1.4e-05}$	$2.269694e-05_{0.0e+00}$
DTLZ1	$3.483237e-04_{6.9e-06}$	$2.526425e-04_{0.0e+00}$
DTLZ2	$4.540029e-04_{7.2e-06}$	$3.343429e-04_{0.0e+00}$
DTLZ3	$7.371303e-04_{1.6e-05}$	$5.305681e-04_{0.0e+00}$
DTLZ4	$6.460156e-04_{7.3e-05}$	$5.417136e-04_{0.0e+00}$
DTLZ5	$1.263027e-05_{1.2e-06}$	$3.674228e-06_{0.0e+00}$
DTLZ6	$3.071575e-05_{4.2e-07}$	$8.785615e-06_{0.0e+00}$
DTLZ7	$3.071591e-03_{3.0e-03}$	$1.181486e-03_{0.0e+00}$

MOEA/D-RMS [11], MOEA/D-M2M [11], D^2 MOPSO [42] and HyPE [9]) with competitive results. Therefore, DPP can be considered a state of the art algorithm. In this study, we focus on comparing the DPPCP with the original algorithm DPP (named ED/DPP) and the DPP2 algorithm [32], a modified version of this algorithm.

The results in Table 5 and Table 6 show that the proposed method (DPPCP) a is clearly better ED/DPP and DPP2 (it gives better metric value in 24 out of 32 comparisons). In ZDT instances, DPPCP give results better than ED/DPP in all instances, especially, in ZDT4 instance, DPPCP outperforms ED/DPP about 10.000 times. In UF instances, ED/DPP achieves better performance on UF5 and UF6 instances. However, DPPCP obtains better IGD metric values in other UF instances, even with UF1 it is better about 100 times. Similar to WFG instances, DPPCP achieves better metric values in all of the comparisons (except WFG5). These results show that the competitive co-evolution model proposed in this paper outperforms the co-operative co-evolution methods in ([31], [32]).

F. EFFECTS OF COMPETITIVENESS

To verify the effect of the competitive co-evolutionary approach, we have developed a variant (denoted as *DPPCP-Variant1*). In *DPPCP-Variant1*, there is no interaction between the two populations, except for a connection in the selection stage (NBSM), where some individuals on the A_d side can be borrowed from the A_p. Two offspring $Child_{Ap}$ and $Child_{Ad}$ will be used to update the population immediately instead of being used to compete in the DPPCP. The result is a combination of output from each population.

The performance comparisons are shown in Table 7 and Table 8 via the mean and standard deviation values. For each row in the table, we highlight the best value in bold.

In Table 7, we conduct the comparison between DPPCP and *DPPCP-Variant1* using the HV metric. The DPPCP attains better metric values in all of the comparisons (except UF5, UF6, and UF9). Especially in Table 8, The DPPCP's results are about ten times as good as DPP's are with ZDT1, ZDT3, ZDT4, UF1, WFG3, WFG4, DTLZ5, DTLZ6 and about 100 times with WFG1, WFG2.

It can be seen that DPPCP shows better performance than *DPPCP-Variant1* in most instances. Especially, DPPCP outperforms *DPPCP-Variant1* in the WFG series. Based on the results, it can be clear to see the advantage of the competitive co-evolutionary approach. It helps to achieve better results on both criteria (i.e. convergence and diversity).

G. EFFECTS OF THE NBSM MECHANISM

To further understand the effects of the NBSM mechanism, we extend this mechanism to two other variants as follows:

1. DPPCP-Variant2: This variant is different from DPPCP in that it chooses two sub-regions in the A_p. If the sub-regions do not contain any solution, it randomly selects from the A_p instead of borrowing from the A_d such as DPPCP. This experiment aims to show the importance of selecting solutions in the neighborhood sub-regions.

2. DPPCP-Variant3: In A_p, instead of carefully selecting two mating parents from all neighborhoods of the current

TABLE 9. Performance comparisons between the DPPCP with DPPCP-Variant2 and DPPCP-Variant3 using IGD metric.

	DPPCP	DPPCP-Variant3	DPPCP-Variant2
ZDT1	**$6.655793e-01_{0.0e+00}$**	$6.655222e-01_{3.5e-05}$	$6.654682e-01_{8.0e-05}$
ZDT2	$3.321892e-01_{0.0e+00}$	$3.321594e-01_{4.2e-05}$	**$3.322645e-01_{4.6e-05}$**
ZDT3	$5.170450e-01_{0.0e+00}$	**$5.170514e-01_{4.4e-06}$**	$5.170115e-01_{2.3e-05}$
ZDT4	$6.655913e-01_{0.0e+00}$	**$6.655955e-01_{1.2e-05}$**	$6.655771e-01_{1.7e-05}$
ZDT6	$4.053136e-01_{0.0e+00}$	**$4.053195e-01_{1.0e-05}$**	$4.053177e-01_{2.0e-05}$
UF1	**$6.639659e-01_{0.0e+00}$**	$6.638367e-01_{1.7e-04}$	$6.611297e-01_{1.3e-02}$
UF	**$6.616692e-01_{0.0e+00}$**	$6.584050e-01_{2.3e-03}$	$6.585740e-01_{1.8e-03}$
UF3	$5.965543e-01_{0.0e+00}$	**$6.489444e-01_{1.2e-02}$**	$6.341751e-01_{4.1e-02}$
UF4	**$2.551797e-01_{0.0e+00}$**	$2.522927e-01_{5.0e-03}$	$2.527116e-01_{3.6e-03}$
UF5	$8.999985e-02_{0.0e+00}$	$1.511017e-01_{9.1e-02}$	**$1.851661e-01_{1.0e-01}$**
UF6	$1.725421e-01_{0.0e+00}$	**$1.954255e-01_{1.0e-01}$**	$1.720137e-01_{8.3e-02}$
UF7	**$4.954061e-01_{0.0e+00}$**	$4.472362e-01_{1.0e-01}$	$4.918281e-01_{8.7e-03}$
UF8	**$3.623309e-01_{0.0e+00}$**	$3.418781e-01_{2.7e-02}$	$3.032140e-01_{8.0e-02}$
UF9	$5.565798e-01_{0.0e+00}$	$5.540199e-01_{7.7e-03}$	**$5.885163e-01_{5.4e-02}$**
UF10	**$9.315613e-02_{0.0e+00}$**	$6.069509e-02_{1.6e-02}$	$8.027631e-02_{4.1e-02}$
WFG1	**$6.370891e-01_{0.0e+00}$**	$6.353138e-01_{3.2e-04}$	$6.353696e-01_{1.4e-04}$
WFG2	$5.654082e-01_{0.0e+00}$	**$5.654134e-01_{6.5e-06}$**	$5.654117e-01_{4.9e-06}$
WFG3	**$4.987931e-01_{0.0e+00}$**	$4.987877e-01_{8.3e-06}$	$4.987841e-01_{1.7e-05}$
WFG4	$2.220795e-01_{0.0e+00}$	$2.220849e-01_{1.1e-04}$	**$2.221528e-01_{4.8e-05}$**
WFG5	$1.989817e-01_{0.0e+00}$	**$1.997983e-01_{2.6e-03}$**	$1.993798e-01_{1.8e-03}$
WFG6	$2.136358e-01_{0.0e+00}$	**$2.136419e-01_{1.0e-05}$**	$2.136229e-01_{9.6e-06}$
WFG7	$2.136157e-01_{0.0e+00}$	**$2.136398e-01_{9.0e-06}$**	$2.136212e-01_{8.9e-06}$
WFG8	**$2.114787e-01_{0.0e+00}$**	$1.950397e-01_{2.6e-02}$	$1.762630e-01_{2.7e-02}$
WFG9	**$2.449073e-01_{0.0e+00}$**	$2.446959e-01_{8.0e-05}$	$2.447730e-01_{1.6e-04}$
DTLZ1	$8.027772e-01_{0.0e+00}$	**$8.034714e-01_{6.8e-04}$**	$7.993489e-01_{8.5e-04}$
DTLZ2	$4.294453e-01_{0.0e+00}$	**$4.318317e-01_{1.2e-03}$**	$4.275582e-01_{8.8e-04}$
DTLZ3	$4.298513e-01_{0.0e+00}$	**$4.337437e-01_{1.2e-03}$**	$4.282988e-01_{8.1e-04}$
DTLZ4	$4.245304e-01_{0.0e+00}$	**$4.270484e-01_{6.5e-04}$**	$4.241378e-01_{7.9e-04}$
DTLZ5	**$9.579286e-02_{0.0e+00}$**	$9.578658e-02_{1.1e-05}$	$9.574149e-02_{1.1e-05}$
DTLZ6	**$9.678412e-02_{0.0e+00}$**	$9.676609e-02_{6.0e-06}$	$9.676292e-02_{7.1e-06}$
DTLZ7	**$3.103702e-01_{0.0e+00}$**	$2.811700e-01_{4.6e-02}$	$2.462745e-01_{3.2e-02}$

TABLE 10. Performance comparisons between the DPPCP with DPPCP-Variant2 and DPPCP-Variant3 using SPREAD metric.

	DPPCP	DPPCP-Variant3	DPPCP-Variant2
ZDT1	**$5.090732e-01_{0.0e+00}$**	$5.147352e-01_{1.8e-02}$	$5.096334e-01_{1.4e-02}$
ZDT2	$5.234142e-01_{0.0e+00}$	**$4.927756e-01_{1.3e-02}$**	$4.992091e-01_{2.2e-02}$
ZDT3	**$8.496801e-01_{0.0e+00}$**	$8.649409e-01_{3.5e-03}$	$8.585498e-01_{5.6e-03}$
ZDT4	**$4.978226e-01_{0.0e+00}$**	$5.190670e-01_{1.3e-02}$	$5.083129e-01_{1.4e-02}$
ZDT6	$1.262919e+00_{0.0e+00}$	$1.100262e+00_{3.3e-01}$	**$7.020149e-01_{3.2e-01}$**
UF1	$4.395728e-01_{0.0e+00}$	**$4.341900e-01_{1.9e-02}$**	$4.813619e-01_{9.4e-02}$
UF	$5.609735e-01_{0.0e+00}$	$5.422028e-01_{2.4e-02}$	**$5.267486e-01_{2.3e-02}$**
UF3	$9.158526e-01_{0.0e+00}$	**$8.435704e-01_{2.0e-01}$**	$8.540161e-01_{2.0e-01}$
UF4	**$5.607462e-01_{0.0e+00}$**	$6.460472e-01_{6.7e-02}$	$6.239098e-01_{6.3e-02}$
UF5	**$1.116620e+00_{0.0e+00}$**	$1.210485e+00_{2.2e-01}$	$1.219553e+00_{1.7e-01}$
UF6	**$1.000035e+00_{0.0e+00}$**	$1.329176e+00_{1.8e-01}$	$1.217140e+00_{1.6e-01}$
UF7	$9.278187e-01_{0.0e+00}$	$6.441274e-01_{3.1e-01}$	**$5.959498e-01_{1.7e-01}$**
UF8	**$8.112851e-01_{0.0e+00}$**	$8.631439e-01_{4.4e-02}$	$8.516223e-01_{1.1e-01}$
UF9	$1.172136e+00_{0.0e+00}$	$1.055870e+00_{1.0e-01}$	**$1.016694e+00_{1.4e-01}$**
UF10	$1.113429e+00_{0.0e+00}$	$1.139191e+00_{2.0e-01}$	**$9.984497e-01_{1.4e-01}$**
WFG1	**$5.910268e-01_{0.0e+00}$**	$5.980603e-01_{8.6e-03}$	$6.014965e-01_{1.4e-02}$
WFG2	**$9.265058e-01_{0.0e+00}$**	$9.275593e-01_{3.8e-03}$	$9.266749e-01_{3.2e-03}$
WFG3	$5.067168e-01_{0.0e+00}$	$5.092385e-01_{9.3e-03}$	**$5.061781e-01_{1.5e-02}$**
WFG4	**$5.102968e-01_{0.0e+00}$**	$5.259370e-01_{8.2e-03}$	$5.184175e-01_{1.5e-02}$
WFG5	**$5.124313e-01_{0.0e+00}$**	$5.297967e-01_{1.2e-02}$	$5.311470e-01_{1.1e-02}$
WFG6	**$5.025690e-01_{0.0e+00}$**	$5.081665e-01_{1.7e-02}$	$5.058109e-01_{1.5e-02}$
WFG7	$5.117605e-01_{0.0e+00}$	**$5.070984e-01_{1.2e-02}$**	$5.085303e-01_{1.4e-02}$
WFG8	**$5.850444e-01_{0.0e+00}$**	$6.678814e-01_{9.4e-02}$	$6.883508e-01_{7.6e-02}$
WFG9	**$5.152843e-01_{0.0e+00}$**	$5.435935e-01_{1.6e-02}$	$5.384225e-01_{1.1e-02}$
DTLZ1	**$7.480045e-01_{0.0e+00}$**	$7.700924e-01_{1.7e-02}$	$7.612776e-01_{1.5e-02}$
DTLZ2	**$6.909910e-01_{0.0e+00}$**	$7.058144e-01_{2.1e-02}$	$6.936715e-01_{1.4e-02}$
DTLZ3	$7.258689e-01_{0.0e+00}$	$7.074477e-01_{1.5e-02}$	**$6.992068e-01_{1.7e-02}$**
DTLZ4	$7.042373e-01_{0.0e+00}$	$7.078723e-01_{1.9e-02}$	**$6.894604e-01_{1.2e-02}$**
DTLZ5	**$6.133811e-01_{0.0e+00}$**	$6.366189e-01_{1.3e-02}$	$6.351343e-01_{1.7e-02}$
DTLZ6	**$6.074105e-01_{0.0e+00}$**	$6.360939e-01_{1.2e-02}$	$6.328897e-01_{9.2e-03}$
DTLZ7	$9.165426e-01_{0.0e+00}$	$8.179126e-01_{1.1e-01}$	**$7.521809e-01_{1.1e-01}$**

sub-region, this variant randomly selects two neighborhood sub-regions regardless of whether they contain any solution or not. If two sub-regions do not contain any solution, they borrow from two sub-regions in the A_d respectively. This experiment aims to show the importance of searching for neighborhood solutions in the whole neighborhood sub-regions.

The performance comparisons between DPPCP with two variants, regarding the IGD and the SPREAD metrics, are presented in Table 9 and Table 10. It is clear that DPPCP is

TABLE 11. Performance comparisons between NSGAII with *Ap* using HV metric.

	NSGAII	Ap
ZDT1	$6.647712e-01_{5.6e-05}$	**$6.650548e-01_{3.2e-05}$**
ZDT2	$3.314915e-01_{2.8e-05}$	**$3.317714e-01_{1.2e-05}$**
ZDT3	$5.168546e-01_{1.3e-05}$	**$5.169530e-01_{2.6e-06}$**
ZDT4	$6.648566e-01_{2.0e-05}$	**$6.651099e-01_{3.9e-06}$**
ZDT6	$4.031377e-01_{1.2e-04}$	**$4.047653e-01_{6.9e-06}$**
UF1	$5.419462e-01_{2.1e-02}$	**$6.639871e-01_{8.5e-05}$**
UF	$6.409438e-01_{3.6e-03}$	**$6.602307e-01_{9.2e-04}$**
UF3	$4.692771e-01_{1.8e-02}$	**$6.531597e-01_{1.2e-02}$**
UF4	**$2.726242e-01_{3.8e-04}$**	$2.538280e-01_{4.0e-03}$
UF5	**$2.395064e-01_{3.2e-02}$**	$1.533705e-01_{9.1e-02}$
UF6	**$2.702487e-01_{4.2e-02}$**	$1.845366e-01_{7.1e-02}$
UF7	$4.238320e-01_{6.3e-02}$	**$4.944873e-01_{3.7e-03}$**
UF8	$1.065488e-01_{2.3e-02}$	**$3.201665e-01_{2.2e-02}$**
UF9	$4.384664e-01_{1.2e-01}$	**$4.968106e-01_{6.8e-02}$**
UF10	$8.641870e-04_{1.4e-03}$	**$3.287506e-02_{2.3e-02}$**
WFG1	$6.337007e-01_{3.9e-04}$	**$6.354226e-01_{1.9e-04}$**
WFG2	$5.653015e-01_{1.1e-05}$	**$5.653629e-01_{2.6e-06}$**
WFG3	$4.978586e-01_{4.8e-05}$	**$4.982742e-01_{5.1e-06}$**
WFG4	$2.214275e-01_{1.0e-04}$	**$2.218771e-01_{3.7e-06}$**
WFG5	$1.982329e-01_{9.5e-05}$	**$1.986686e-01_{7.8e-06}$**
WFG6	$2.097628e-01_{4.0e-03}$	**$2.133093e-01_{2.2e-06}$**
WFG7	$2.129740e-01_{5.5e-05}$	**$2.133029e-01_{1.6e-06}$**
WFG8	**$1.752138e-01_{2.6e-02}$**	$1.734960e-01_{2.6e-02}$
WFG9	$2.435342e-01_{4.9e-04}$	**$2.444773e-01_{1.5e-04}$**
DTLZ1	**$7.951600e-01_{2.9e-03}$**	$7.902245e-01_{1.8e-03}$
DTLZ2	$4.146214e-01_{2.9e-03}$	**$4.210202e-01_{1.9e-03}$**
DTLZ3	**$4.231073e-01_{2.8e-03}$**	$4.225103e-01_{2.4e-03}$
DTLZ4	$4.132746e-01_{1.8e-03}$	**$4.148104e-01_{2.7e-03}$**
DTLZ5	$9.541452e-02_{2.5e-05}$	**$9.562316e-02_{6.0e-06}$**
DTLZ6	$6.790182e-02_{1.2e-02}$	**$9.656647e-02_{1.0e-05}$**
DTLZ7	**$3.121336e-01_{1.4e-03}$**	$2.459266e-01_{3.4e-02}$

TABLE 12. Performance comparisons between NSGAII with Ap using IGD metric.

	NSGAII	Ap
ZDT1	$5.919099e-05_{2.6e-06}$	**$4.484072e-05_{2.0e-07}$**
ZDT2	$6.035183e-05_{2.2e-06}$	**$4.586843e-05_{2.9e-07}$**
ZDT3	$4.190027e-05_{1.5e-06}$	**$3.235044e-05_{4.4e-07}$**
ZDT4	$5.677679e-05_{1.1e-06}$	**$4.468699e-05_{2.6e-07}$**
ZDT6	$7.329672e-05_{2.5e-06}$	**$4.395604e-05_{1.6e-07}$**
UF1	$3.531346e-03_{6.7e-04}$	**$5.458855e-05_{1.5e-06}$**
UF	$1.014096e-03_{2.9e-04}$	**$2.503008e-04_{5.8e-05}$**
UF3	$7.093487e-03_{1.6e-03}$	**$3.015407e-04_{2.9e-04}$**
UF4	**$1.351779e-03_{1.3e-05}$**	$1.721485e-03_{8.0e-05}$
UF5	**$4.420051e-02_{4.3e-03}$**	$1.027762e-01_{4.5e-02}$
UF6	**$8.820306e-03_{2.6e-03}$**	$1.716127e-02_{5.2e-03}$
UF7	$3.519909e-03_{3.8e-03}$	**$4.313322e-04_{5.1e-04}$**
UF8	$2.991707e-03_{8.8e-05}$	**$1.032012e-03_{1.4e-04}$**
UF9	$2.324010e-03_{1.1e-03}$	**$2.173429e-03_{5.4e-04}$**
UF10	**$5.264139e-03_{2.0e-03}$**	$5.715941e-03_{9.0e-04}$
WFG1	$3.146505e-04_{2.1e-05}$	**$2.353121e-04_{1.7e-05}$**
WFG2	$1.185735e-04_{8.1e-06}$	**$9.394695e-05_{5.0e-06}$**
WFG3	$6.509974e-05_{2.8e-06}$	**$4.879745e-05_{8.5e-07}$**
WFG4	$5.599029e-05_{3.1e-06}$	**$4.188075e-05_{5.4e-07}$**
WFG5	$9.334401e-04_{1.2e-06}$	**$9.309544e-04_{1.1e-07}$**
WFG6	$1.654294e-04_{1.3e-04}$	**$7.376991e-05_{3.9e-06}$**
WFG7	$3.933206e-05_{3.0e-06}$	**$2.845054e-05_{2.6e-07}$**
WFG8	**$2.389023e-03_{1.3e-03}$**	$2.778718e-03_{1.4e-03}$
WFG9	$4.201996e-05_{2.8e-06}$	**$2.995207e-05_{1.9e-07}$**
DTLZ1	$3.231009e-04_{9.7e-06}$	**$3.214582e-04_{1.3e-05}$**
DTLZ2	$4.433946e-04_{1.7e-05}$	**$4.132936e-04_{9.8e-06}$**
DTLZ3	$6.935231e-04_{1.5e-05}$	**$6.602071e-04_{1.7e-05}$**
DTLZ4	$7.956023e-04_{1.2e-04}$	**$7.878465e-04_{4.7e-05}$**
DTLZ5	$6.212032e-06_{2.9e-07}$	**$4.629317e-06_{7.3e-08}$**
DTLZ6	$2.898050e-04_{1.5e-04}$	**$1.134035e-05_{3.0e-07}$**
DTLZ7	**$1.210533e-03_{4.6e-05}$**	$1.846060e-02_{7.2e-03}$

the best candidate: it obtains better metric values in 20 out of 32 comparisons. On the contrary, DPPCP-Variant2 is the worst among them with IGD metric. Meanwhile, DPPCP-Variant3 obtains the poor spread.

In short, our proposed NBSM mechanism, which fully utilizes the guidance information of the neighborhood, is effective.

TABLE 13. Performance comparisons between MOEAD/DE with Ad using HV metric.

	MOEAD/DE	Ad
ZDT1	$6.648280e-01_{5.4e-05}$	**$6.649500e-01_{3.2e-05}$**
ZDT2	$3.315831e-01_{5.4e-05}$	**$3.316895e-01_{1.0e-05}$**
ZDT3	$5.162168e-01_{1.4e-05}$	**$5.162376e-01_{3.0e-06}$**
ZDT4	$6.649704e-01_{8.3e-06}$	**$6.649978e-01_{4.4e-06}$**
ZDT6	**$4.047282e-01_{1.5e-08}$**	$4.047278e-01_{1.3e-06}$
UF1	$6.635979e-01_{1.3e-04}$	**$6.639067e-01_{1.6e-04}$**
UF	$6.565669e-01_{2.8e-03}$	**$6.602800e-01_{6.3e-04}$**
UF3	**$6.578853e-01_{4.4e-03}$**	$6.528604e-01_{1.0e-02}$
UF4	$2.472456e-01_{5.4e-03}$	**$2.530931e-01_{5.1e-03}$**
UF5	$4.976365e-02_{5.8e-02}$	**$1.901213e-01_{1.2e-01}$**
UF6	**$2.059300e-01_{8.3e-02}$**	$1.557399e-01_{9.6e-02}$
UF7	$4.945292e-01_{3.3e-03}$	**$4.947838e-01_{3.1e-03}$**
UF8	$3.286710e-01_{2.2e-02}$	**$3.347479e-01_{3.0e-02}$**
UF9	**$5.973945e-01_{6.0e-02}$**	$5.653599e-01_{7.9e-03}$
UF10	$7.037545e-02_{1.9e-02}$	**$7.481132e-02_{3.5e-02}$**
WFG1	$6.347123e-01_{2.2e-04}$	**$6.349902e-01_{2.2e-04}$**
WFG2	$5.646696e-01_{1.5e-05}$	**$5.646973e-01_{2.0e-06}$**
WFG3	$4.980009e-01_{4.5e-06}$	**$4.980084e-01_{2.8e-06}$**
WFG4	$2.212079e-01_{8.0e-05}$	**$2.213991e-01_{9.9e-06}$**
WFG5	$1.987543e-01_{2.0e-03}$	**$1.989529e-01_{2.6e-03}$**
WFG6	$2.128719e-01_{4.6e-06}$	**$2.128776e-01_{3.4e-06}$**
WFG7	$2.128584e-01_{8.0e-06}$	**$2.128669e-01_{3.0e-06}$**
WFG8	**$1.732240e-01_{2.6e-02}$**	$1.677706e-01_{2.3e-02}$
WFG9	$2.439037e-01_{6.4e-05}$	**$2.439922e-01_{1.7e-04}$**
DTLZ1	$7.848600e-01_{2.6e-04}$	**$7.849389e-01_{2.5e-04}$**
DTLZ2	**$4.187079e-01_{9.6e-04}$**	$4.186242e-01_{7.6e-04}$
DTLZ3	**$4.193613e-01_{9.2e-04}$**	$4.193516e-01_{1.2e-03}$
DTLZ4	**$4.169853e-01_{8.7e-04}$**	$4.157544e-01_{1.1e-03}$
DTLZ5	$9.469885e-02_{7.5e-06}$	**$9.472012e-02_{5.0e-06}$**
DTLZ6	**$9.566261e-02_{6.8e-07}$**	$9.566209e-02_{6.4e-07}$
DTLZ7	**$2.801997e-01_{6.6e-03}$**	$2.316691e-01_{2.6e-02}$

TABLE 14. Performance comparisons between MOEAD/DE with Ad using IGD metric.

	MOEAD	Ad
ZDT1	$5.561091e-05_{2.2e-07}$	**$5.528320e-05_{1.7e-07}$**
ZDT2	**$4.661063e-05_{4.8e-08}$**	$4.662861e-05_{8.1e-09}$
ZDT3	$8.897932e-05_{1.7e-06}$	**$8.797083e-05_{9.4e-08}$**
ZDT4	$5.903217e-05_{4.4e-07}$	**$5.892080e-05_{1.5e-07}$**
ZDT6	$4.627950e-05_{5.4e-09}$	**$4.627584e-05_{1.6e-09}$**
UF1	$7.056534e-05_{5.6e-06}$	**$6.249016e-05_{2.8e-06}$**
UF	$4.789866e-04_{1.5e-04}$	**$2.519000e-04_{3.9e-05}$**
UF3	**$1.852946e-04_{9.2e-05}$**	$2.948503e-04_{2.2e-04}$
UF4	$1.919477e-03_{1.2e-04}$	**$1.765559e-03_{1.1e-04}$**
UF5	**$6.953943e-02_{1.0e-02}$**	$9.583202e-02_{5.3e-02}$
UF6	**$7.166399e-03_{8.6e-03}$**	$2.003117e-02_{7.4e-03}$
UF7	**$3.171871e-04_{4.7e-04}$**	$3.210059e-04_{4.0e-04}$
UF8	**$1.082474e-03_{2.5e-04}$**	$1.122969e-03_{3.3e-04}$
UF9	**$1.706092e-03_{7.8e-04}$**	$2.168509e-03_{7.8e-05}$
UF10	**$5.162630e-03_{4.9e-04}$**	$5.195688e-03_{6.8e-04}$
WFG1	$2.771777e-04_{1.4e-05}$	**$2.574595e-04_{1.9e-05}$**
WFG2	**$6.031492e-04_{8.5e-06}$**	$6.081285e-04_{1.1e-06}$
WFG3	$5.477941e-05_{6.2e-08}$	**$5.475384e-05_{3.2e-08}$**
WFG4	$6.262985e-05_{7.2e-07}$	**$6.154841e-05_{2.9e-07}$**
WFG5	$9.156851e-04_{5.7e-05}$	**$9.095219e-04_{7.3e-05}$**
WFG6	**$9.134833e-05_{1.0e-07}$**	$9.137324e-05_{6.0e-08}$
WFG7	$4.053872e-05_{1.2e-08}$	**$4.053413e-05_{2.3e-08}$**
WFG8	**$2.601146e-03_{1.3e-03}$**	$2.976396e-03_{1.2e-03}$
WFG9	$4.064367e-05_{1.4e-07}$	**$4.055560e-05_{2.0e-07}$**
DTLZ1	$3.475079e-04_{1.1e-06}$	**$3.473201e-04_{8.9e-07}$**
DTLZ2	$4.309118e-04_{2.2e-06}$	**$4.308753e-04_{1.9e-06}$**
DTLZ3	$7.219248e-04_{4.6e-06}$	**$7.210165e-04_{3.3e-06}$**
DTLZ4	**$7.737441e-04_{5.9e-05}$**	$8.445142e-04_{7.7e-05}$
DTLZ5	$1.517558e-05_{6.4e-08}$	**$1.514454e-05_{5.4e-08}$**
DTLZ6	**$3.453149e-05_{2.9e-08}$**	$3.454897e-05_{1.9e-08}$
DTLZ7	**$4.015304e-03_{4.3e-03}$**	$2.044364e-02_{7.1e-03}$

H. INTERACTION BETWEEN TWO CO-EVOLVING POPULATIONS

In this section, we clarify the effect of using dual populations. Specifically, we consider two main points:

TABLE 15. Performance comparisons between DPPCP with *DPPCP-Ap* and *DPPCP-Ad* using the HV metric.

	DPPCP-Ap	DPPCP-Ad	DPPCP
ZDT1	$6.650573e-01_{3.4e-05}$	$6.649439e-01_{3.5e-05}$	$6.654682e-01_{8.0e-05}$
ZDT2	$3.317733e-01_{1.2e-05}$	$3.316889e-01_{1.1e-05}$	$3.322645e-01_{4.6e-05}$
ZDT3	$5.169542e-01_{2.8e-06}$	$5.162387e-01_{3.6e-06}$	$5.170115e-01_{2.3e-05}$
ZDT4	$6.651104e-01_{5.3e-06}$	$6.649971e-01_{5.5e-06}$	$6.655771e-01_{1.7e-05}$
ZDT6	$4.047643e-01_{6.5e-06}$	$4.047280e-01_{8.9e-07}$	$4.053177e-01_{2.0e-05}$
UF1	$6.639747e-01_{7.9e-05}$	$6.639469e-01_{1.4e-04}$	$6.611297e-01_{1.3e-02}$
UF	$6.599954e-01_{1.3e-03}$	$6.601949e-01_{9.0e-04}$	$6.585740e-01_{1.8e-03}$
UF3	$6.521510e-01_{1.2e-02}$	$6.512283e-01_{1.2e-02}$	$6.341751e-01_{4.1e-02}$
UF4	$2.541054e-01_{3.9e-03}$	$2.555352e-01_{5.5e-03}$	$2.527116e-01_{3.6e-03}$
UF5	$1.690427e-01_{1.0e-01}$	$1.736297e-01_{1.2e-01}$	$1.851661e-01_{1.0e-01}$
UF6	$1.857803e-01_{8.4e-02}$	$1.660392e-01_{1.0e-01}$	$1.720137e-01_{8.3e-02}$
UF7	$4.834219e-01_{5.2e-02}$	$4.951684e-01_{2.6e-03}$	$4.918281e-01_{8.7e-03}$
UF8	$3.158450e-01_{2.6e-02}$	$3.255501e-01_{4.0e-02}$	$3.032140e-01_{8.0e-02}$
UF9	$4.965614e-01_{6.8e-02}$	$5.620449e-01_{9.7e-03}$	$5.885163e-01_{5.4e-02}$
UF10	$3.846684e-02_{2.6e-02}$	$8.202228e-02_{3.5e-02}$	$8.027631e-02_{4.1e-02}$
WFG1	$6.353914e-01_{1.7e-04}$	$6.350033e-01_{2.3e-04}$	$6.353696e-01_{1.4e-04}$
WFG2	$5.653623e-01_{2.9e-06}$	$5.646976e-01_{2.3e-06}$	$5.654117e-01_{4.9e-06}$
WFG3	$4.982741e-01_{4.9e-06}$	$4.980092e-01_{3.6e-06}$	$4.987841e-01_{1.7e-05}$
WFG4	$2.218733e-01_{7.9e-06}$	$2.214039e-01_{1.0e-05}$	$2.221528e-01_{4.8e-05}$
WFG5	$1.990736e-01_{1.8e-03}$	$1.993133e-01_{2.9e-03}$	$1.993798e-01_{1.8e-03}$
WFG6	$2.133075e-01_{3.9e-06}$	$2.128778e-01_{3.0e-06}$	$2.136229e-01_{9.6e-06}$
WFG7	$2.133019e-01_{3.1e-06}$	$2.128669e-01_{3.4e-06}$	$2.136212e-01_{8.9e-06}$
WFG8	$1.680864e-01_{2.2e-02}$	$1.731327e-01_{2.6e-02}$	$1.762630e-01_{2.7e-02}$
WFG9	$2.444569e-01_{1.5e-04}$	$2.439633e-01_{1.5e-04}$	$2.447730e-01_{1.6e-04}$
DTLZ1	$7.910644e-01_{2.2e-03}$	$7.848877e-01_{2.4e-04}$	$7.993489e-01_{8.5e-04}$
DTLZ2	$4.200114e-01_{2.3e-03}$	$4.184426e-01_{8.9e-04}$	$4.275582e-01_{8.8e-04}$
DTLZ3	$4.221261e-01_{2.2e-03}$	$4.190582e-01_{1.0e-03}$	$4.282988e-01_{8.1e-04}$
DTLZ4	$4.145095e-01_{2.2e-03}$	$4.159602e-01_{1.1e-03}$	$4.241378e-01_{7.9e-04}$
DTLZ5	$9.562550e-02_{6.0e-06}$	$9.472234e-02_{6.1e-06}$	$9.574149e-02_{1.1e-05}$
DTLZ6	$9.656628e-02_{1.0e-05}$	$9.566211e-02_{5.6e-07}$	$9.676292e-02_{7.1e-06}$
DTLZ7	$2.407067e-01_{3.4e-02}$	$2.339938e-01_{2.6e-02}$	$2.462745e-01_{3.2e-02}$

TABLE 16. Performance comparisons between DPPCP with DPPCP-Ap and DPPCP-Ad using IGD metric.

	DPPCP-Ap	DPPCP-Ad	DPPCP
ZDT1	$4.485633e-05_{2.2e-07}$	$5.530559e-05_{1.6e-07}$	$3.187123e-05_{6.9e-07}$
ZDT2	$4.583104e-05_{2.6e-07}$	$4.662961e-05_{7.6e-09}$	$3.175499e-05_{5.7e-07}$
ZDT3	$3.215338e-05_{5.4e-07}$	$8.807805e-05_{2.4e-07}$	$2.664059e-05_{4.9e-07}$
ZDT4	$4.468264e-05_{2.6e-07}$	$5.901455e-05_{3.2e-07}$	$3.138571e-05_{4.9e-07}$
ZDT6	$4.397428e-05_{1.8e-07}$	$4.627621e-05_{1.6e-09}$	$3.142337e-05_{7.2e-07}$
UF1	$5.465684e-05_{1.3e-06}$	$6.231947e-05_{2.6e-06}$	$3.484854e-04_{1.3e-03}$
UF	$2.682161e-04_{7.3e-05}$	$2.575379e-04_{5.1e-05}$	$3.626042e-04_{1.1e-04}$
UF3	$3.703678e-04_{4.5e-04}$	$3.300163e-04_{2.7e-04}$	$7.198866e-04_{1.1e-03}$
UF4	$1.728170e-03_{8.3e-05}$	$1.719490e-03_{1.2e-04}$	$1.792191e-03_{9.6e-05}$
UF5	$9.393395e-02_{4.9e-02}$	$9.658430e-02_{5.1e-02}$	$9.217221e-02_{4.4e-02}$
UF6	$1.702495e-02_{4.9e-03}$	$1.887109e-02_{7.1e-03}$	$1.715472e-02_{5.8e-03}$
UF7	$1.034829e-03_{3.2e-03}$	$3.049188e-04_{3.6e-04}$	$4.723165e-04_{7.4e-04}$
UF8	$1.068130e-03_{1.6e-04}$	$1.199492e-03_{4.7e-04}$	$1.427242e-03_{1.6e-03}$
UF9	$2.177674e-03_{5.4e-04}$	$2.194444e-03_{1.0e-04}$	$1.860454e-03_{6.2e-04}$
UF10	$5.692229e-03_{7.7e-04}$	$4.920016e-03_{7.8e-04}$	$4.756647e-03_{8.1e-04}$
WFG1	$2.378909e-04_{1.5e-05}$	$2.564765e-04_{2.0e-05}$	$2.642254e-04_{9.2e-06}$
WFG2	$9.553416e-05_{4.9e-06}$	$6.077109e-04_{1.0e-06}$	$8.645268e-05_{5.0e-06}$
WFG3	$4.854589e-05_{8.6e-07}$	$5.473958e-05_{5.1e-08}$	$3.496741e-05_{5.4e-07}$
WFG4	$4.186748e-05_{5.1e-07}$	$6.152974e-05_{2.5e-07}$	$3.327079e-05_{6.0e-07}$
WFG5	$9.194347e-04_{5.2e-05}$	$8.987365e-04_{8.3e-05}$	$9.190031e-04_{5.1e-05}$
WFG6	$7.328672e-05_{3.7e-06}$	$9.137049e-05_{4.6e-08}$	$5.268868e-05_{2.4e-06}$
WFG7	$2.842715e-05_{2.2e-07}$	$4.053810e-05_{1.8e-08}$	$2.202683e-05_{3.0e-07}$
WFG8	$2.972886e-03_{1.2e-03}$	$2.690646e-03_{1.3e-03}$	$2.447889e-03_{1.3e-03}$
WFG9	$2.997811e-05_{2.6e-07}$	$4.058226e-05_{1.7e-07}$	$2.311125e-05_{6.5e-07}$
DTLZ1	$3.199129e-04_{1.3e-05}$	$3.470590e-04_{9.9e-07}$	$2.666947e-04_{5.0e-06}$
DTLZ2	$4.126831e-04_{8.9e-06}$	$4.307759e-04_{2.0e-06}$	$3.401370e-04_{5.6e-06}$
DTLZ3	$6.701955e-04_{1.9e-05}$	$7.209664e-04_{3.7e-06}$	$5.485917e-04_{8.3e-06}$
DTLZ4	$7.959802e-04_{5.7e-05}$	$8.381398e-04_{6.4e-05}$	$5.614028e-04_{5.8e-05}$
DTLZ5	$4.629690e-06_{7.8e-08}$	$1.514716e-05_{5.3e-08}$	$3.817541e-06_{1.1e-07}$
DTLZ6	$1.127554e-05_{2.9e-07}$	$3.453401e-05_{3.1e-08}$	$9.029561e-06_{3.2e-07}$
DTLZ7	$1.923883e-02_{7.1e-03}$	$2.005448e-02_{6.9e-03}$	$1.836414e-02_{6.8e-03}$

1. The effect of using competitive co-evolution on each population.
2. The effect of the interaction between two populations.

Specifically, we first compare A_p with the NSGA-II algorithm and A_d with the MOEA/D-DE algorithm. As discussed above, the algorithms used in A_p and A_d differ from baseline algorithms (i.e. NSGA-II and MOEA/D) at three main points: (a) the mating parent selection mechanism (i.e. NBSM); (b) the way to generate offspring (i.e. competitive method); and (c) how to update Offspring to populations. Through this experiment, we will know whether co-evolution helps individual populations to evolve better than independent evolution.

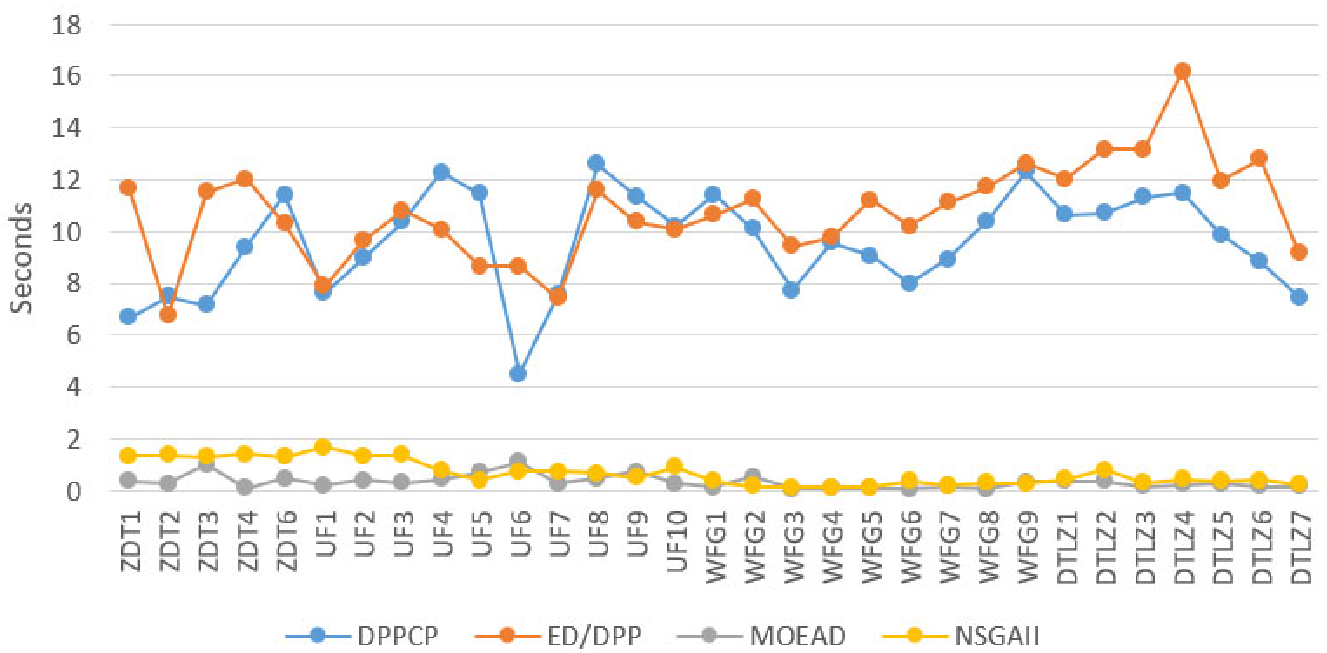


FIGURE 9. CPU time comparisons between for algorithms on different test instances (with the number of generations is 3000).

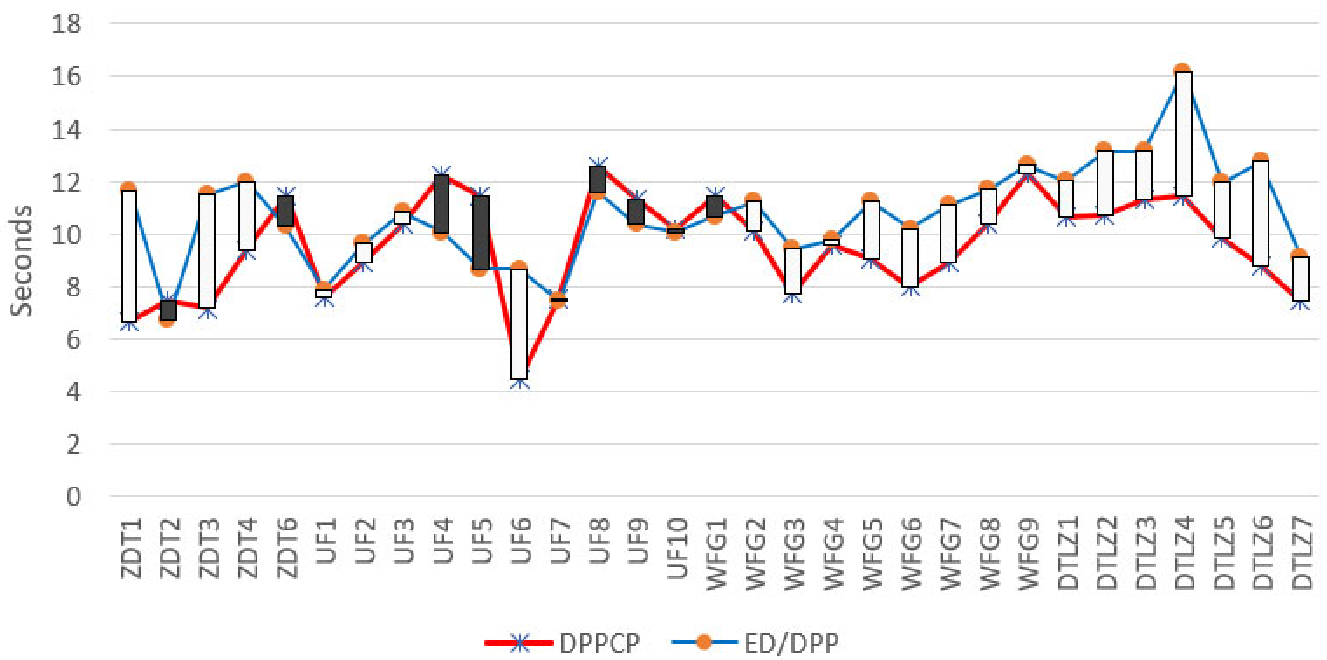


FIGURE 10. CPU time comparisons between for algorithms on different test instances (with the number of generations is 3000).

To implement this comparison, we create two new variants of the DPPCP algorithm: *DPPCP-Ad* and *DPPCP-Ap*. These variants are very similar to the DPPCP algorithm, except that at the last step they only get the results done by the A_p (for *DPPCP-Ap*) and by the A_d (for *DPPCP-Ad*).

The performance of NSGA-II and *DPPCP-Ap* are presented in Table 11, Table 12; and Table 13, Table 14 show the results of comparisons between MOEA/D-DE and DPPCP-Ad. It is clear that *DPPCP-Ap* and *DPPCP-Ad* give better results than NSGA-II and MOEA/D-DE respectively. *DPPCP-Ap* wins in 25 out of 31 comparisons, *DPPCP-Ad*

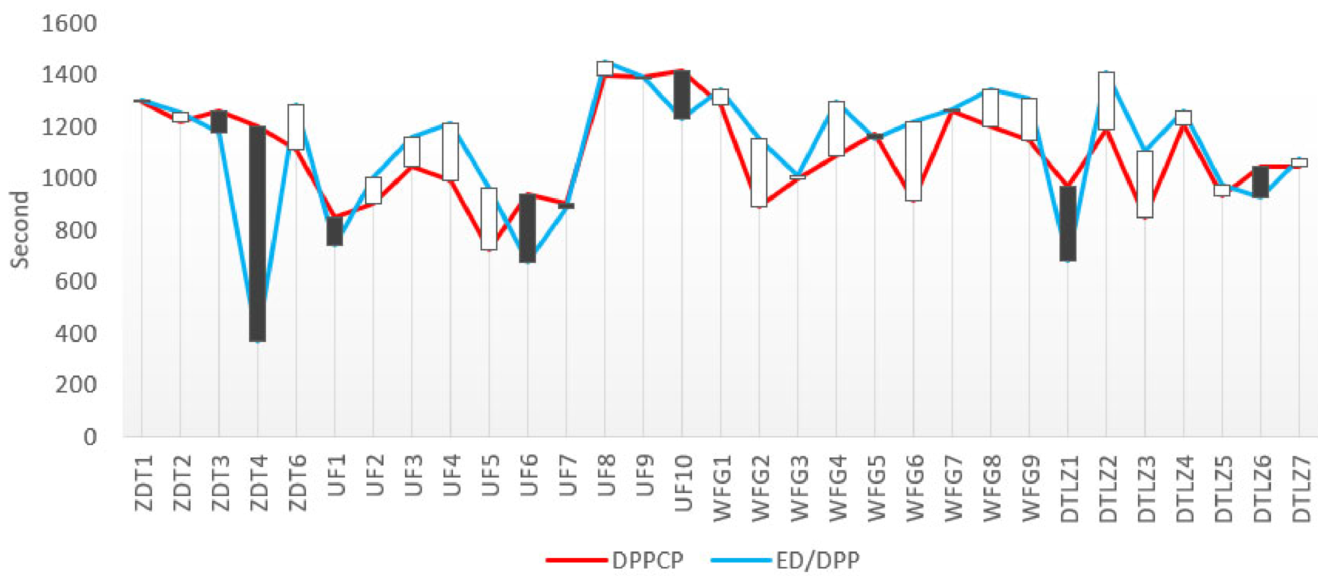


FIGURE 11. CPU time comparisons between DPPCP and ED/DPP on different test instances (with the number of generations is 300.000).

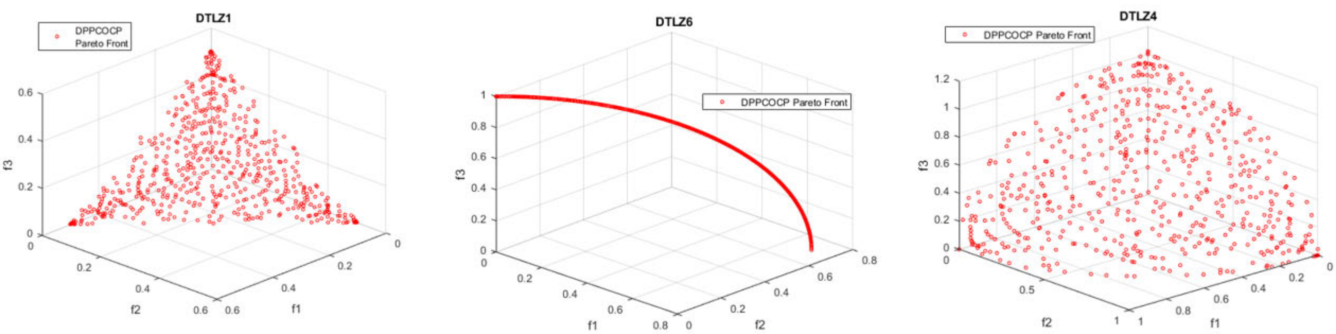


FIGURE 12. Plots of final solutions found by DPPCP algorithm on DTLZ test instances.

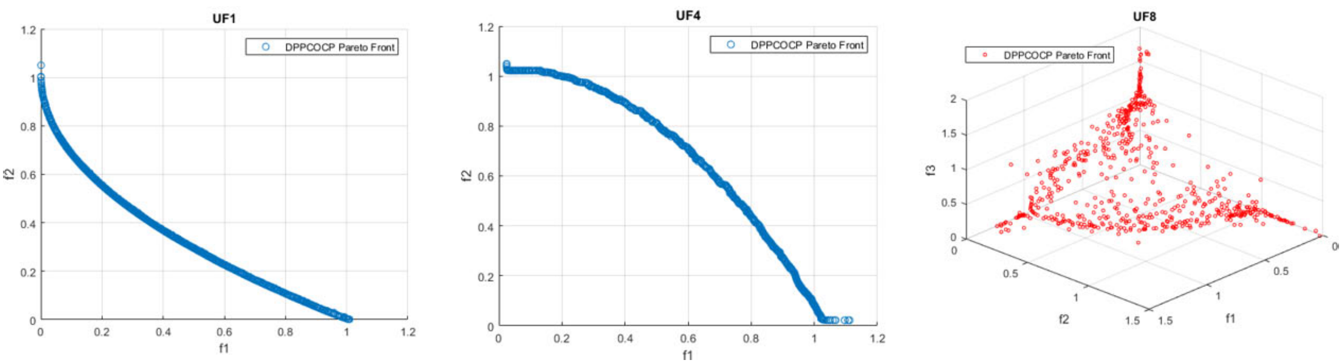


FIGURE 13. Plots of final solutions found by DPPCP algorithm on UF test instances.

obtains better in 22 out of 31 comparisons using the IGD metric. Through experimental results, we realize that the effectiveness of baseline algorithms is enhanced by utilizing a competitive co-evolutionary approach.

We continue comparing the results of each independent population (i.e. A_p and A_d) using co-evolutionary mechanisms with the result of combining both dual populations. By this comparison, we would like to examine whether or not

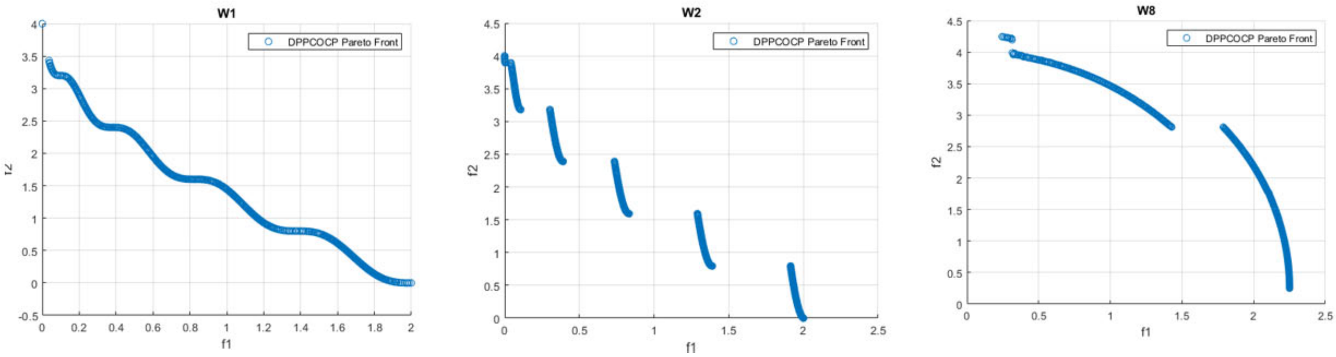


FIGURE 14. Plots of final solutions found by DPPCP algorithm on WFG test instances.

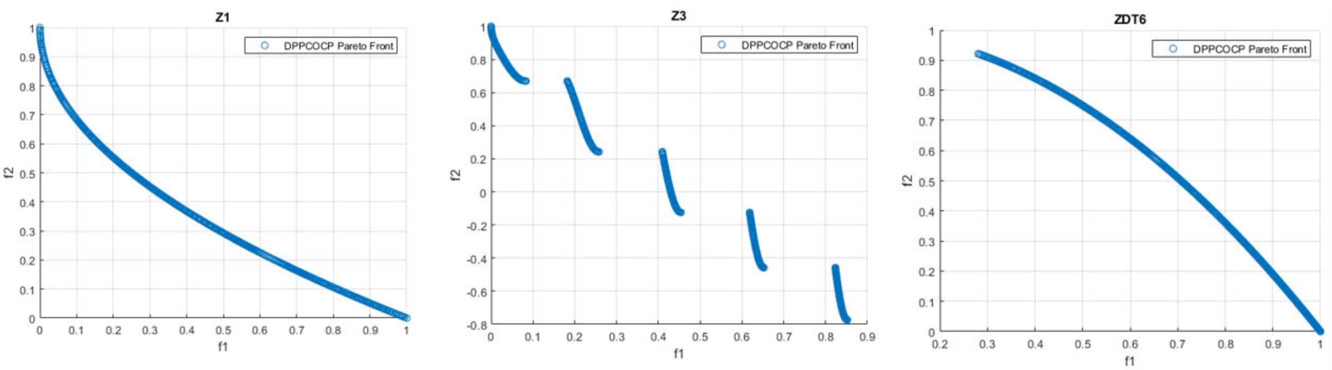


FIGURE 15. Plots of final solutions found by DPPCP algorithm on ZDT test instancess.

the use of dual populations combines the quintessence of both populations.

Table 15 and Table 16 shows the results of comparisons between DPPCP, *DPPCP-Ap*, and *DPPCP-Ad*. It is clear that DPPCP achieves better values in most instances. This shows that thanks to the co-evolution mechanism, with interactions between solutions in two populations that the final population can get the advantages from both populations. It can be said that this population is likely to be able to balance both convergence and diversity.

The final solutions obtained by the DPPCP algorithm and the true PF on the DTLZ, UF, WFG and ZDT series are plotted in Fig.12-15. From these figures, we find that the proposed algorithm can find the approximation set that covers entirely the true PF.

I. CPU TIME COMPARISON

To compare the runtime of the algorithms, we analysed the CPU time cost of the proposed algorithm (DPPCP) with two baselines (NSGA-II and MOEA/D) and the co-evolution algorithms ED/DPP algorithm. We conducted comparisons on 31 problems. To get the most accurate assessment, all algorithms are implemented in *jMetal5* (an integrated JAVA framework). It can be downloaded from http://jmetal.github.io/jMetal/. We run multithreading with 8 cores on computers configured as Intel Xeon E5-2620, 16gb Ram. Experimental results are shown in Fig.9-11. We experimented with two different generation parameters which are 3000 (Fig.9-10) and 300,000 (Fig.11).

Suppose that M is the objective number; N is the population size and T is the neighbour size. The time complexity of MOEA/D in one generation (iteration) is only *O(NTM)*, where M, T≪N. Meanwhile, $O(MN^2)$ is the time complexity of NSGA-II algorithm. The DPPCP and ED/DPP algorithms maintain two coevolving populations. The main running steps of these two algorithms are similar to the MOEA/D algorithm. Thus, the complexity in these main steps is still *O(NTM)*. However, small steps in the algorithms often have to be processed twice for two populations so the calculation time will take more than baseline algorithms. Results with CPU time show this clearly. As you can see in Fig.9, the MOEA/D algorithm costs the least CPU time, followed by the NSGA-II algorithm. These two baseline algorithms take less time than the two DPPCP and ED/DPP algorithms. Fig.10 shows the comparison of the two co-evolution algorithms. The white boxes indicate that the DPPCP algorithm runs faster and vice versa with the black boxes. It is evident that with a loop count of 3000, the DPPCP algorithm runs faster than ED/DPP in most test cases. The result is similar for these two algorithms when the number of iterations increases to 300,000, as shown in Fig.11. The explanation for this result may be in the step of updating the child solution into the Ad population. While DPPCP updates with a limited number (K<T) of neighbourhood solutions (the time complexity is *O(K)*), in ED/DPP the author proceeds through all sub-regions to calculate distances and find the nearest sub-region and compare the child solution with solution in this sub-region to update (the time complexity is *O(NM)*). This is the main reason leading to the difference in CPU time between these two algorithms.

V. CONCLUSION

In this paper, we presented a competitive co-evolutionary approach (DPPCP) for balancing between convergence and diversity in MOEAs. Specifically, we use dual-population competitive co-evolutionary approach with a pair of populations evolved in parallel. One uses the Pareto-based mechanism to obtain a better convergence and the other one uses the decomposition-based technique to maintain the diversity. These populations interact with each other via a new neighborhood based selection mechanism (NBSMS) and a competitive mechanism. We have evaluated the proposed model on four sets of benchmark problems. The performance of the DPPCP is compared with the baseline algorithms, the original version DPP and some variants using the HV and IGD metrics. The empirical outcomes show that the DPPCP model is better on test instances. By comparing DPPCP with baseline algorithms, the empirical results pointed out the efficacy of the new competitive co-evolutionary approach in balancing diversity and convergence in solving MOPs. This is our first study of applying competitive co-evolution to multi-objective optimization problems. The proposed method can still be improved and expanded in several aspects such as how to choose a final population that not only ensures diversity but also approximates the Pareto optimal solution set. Besides, in an extension of this work, we also plan to apply the approach to finding optimal parameters as well as features for machine learning models.

REFERENCES

[1] L. T. Bui, Ed., *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*. Hershey, PA, USA: IGI Global, 2008.

[2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. Hoboken, NJ, USA: Wiley, 2001.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "Improving the strength Pareto evolutionary algorithm," in *Proc. Evol. Methods Design, Optim. Control Appl. Ind. Problems (EUROGEN)*, 2001.

[5] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 2004, pp. 832–842.

[6] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[7] L. T. Bui, J. Liu, A. Bender, M. Barlow, S. Wesolkowski, and H. A. Abbass, "DMEA: A direction-based multiobjective evolutionary algorithm," *Memetic Comput.*, vol. 3, no. 4, pp. 271–285, Dec. 2011.

[8] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Inf. Sci.*, vol. 182, no. 1, pp. 220–242, Jan. 2012.

[9] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.

[10] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, Sep. 2007.

[11] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[12] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[13] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[14] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, Apr. 2016.

[15] D. Xie, L. Ding, Y. Hu, S. Wang, C. Xie, and L. Jiang, "A multi-algorithm balancing convergence and diversity for multi-objective optimization," *J. Inf. Sci. Eng.*, vol. 29, no. 5, pp. 811–834, 2013.

[16] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.

[17] Q. Zhao and T. Higuchi, "Evolutionary learning of nearest-neighbor MLP," *IEEE Trans. Neural Netw.*, vol. 7, no. 3, pp. 762–767, May 1996.

[18] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 1994, pp. 249–257.

[19] H.-F. Wang and Y.-Y. Chen, "A coevolutionary algorithm for the flexible delivery and pickup problem with time windows," *Int. J. Prod. Econ.*, vol. 141, no. 1, pp. 4–13, Jan. 2013.

[20] J. Tian, M. Li, and F. Chen, "Dual-population based coevolutionary algorithm for designing RBFNN with feature selection," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 6904–6918, Oct. 2010.

[21] A.-C. Zăvoianu, E. Lughofer, W. Amrhein, and E. P. Klement, "Efficient multi-objective optimization using 2-population cooperative coevolution," in *Proc. Int. Conf. Comput. Aided Syst. Theory*, 2013, pp. 251–258.

[22] S. Luke, Ed., *Essentials of Metaheuristics*. 2009.

[23] C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, "A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design," *Eur. J. Oper. Res.*, vol. 202, no. 1, pp. 42–54, Apr. 2010.

[24] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009.

[25] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 303–315, Apr. 2019.

[26] L. Zhang, X. Bi, and Y. Wang, "Adaptive truncation technique for constrained multi-objective optimization," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 11, pp. 5489–5511, 2019.

[27] Y. Liu, X. Li, and Q. Hao, "A new constrained multi-objective optimization problems algorithm based on group-sorting," in *Proc. Genet. Evol. Comput. Conf. Companion (GECCO)*, 2019, pp. 221–222.

[28] J. Ou, J. Zheng, G. Ruan, Y. Hu, J. Zou, M. Li, S. Yang, and X. Tan, "A Pareto-based evolutionary algorithm using decomposition and truncation for dynamic multi-objective optimization," *Appl. Soft Comput.*, vol. 85, Dec. 2019, Art. no. 105673.

[29] C. Bao, L. Xu, and E. D. Goodman, "A new dominance-relation metric balancing convergence and diversity in multi- and many-objective optimization," *Expert Syst. Appl.*, vol. 134, pp. 14–27, Nov. 2019.

[30] H. Seada, M. Abouhawwash, and K. Deb, "Multiphase balance of diversity and convergence in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 503–513, Jun. 2019.

[31] K. Li, S. Kwong, and K. Deb, "A dual-population paradigm for evolutionary multiobjective optimization," *Inf. Sci.*, vol. 309, pp. 50–72, Jul. 2015.

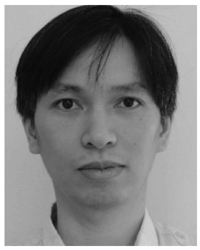
[32] V. T. Vu, L. T. Bui, and T. T. Nguyen, "A modified dual-population approach for solving multi-objective problems," in *Proc. 21st Asia Pacific Symp. Intell. Evol. Syst. (IES)*, Nov. 2017, pp. 89–94.

[33] K. Li, K. Deb, and Q. Zhang, "Evolutionary multiobjective optimization with hybrid selection principles," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 900–907.
[34] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994.
[35] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
[36] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.
[37] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.
[38] Q. Zhang, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Univ. Essex, Colchester, U.K., Nanyang Technol. Univ., Singapore, Tech. Rep., 2008.
[39] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
[40] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
[41] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.* vol. 18, no. 1, pp. 114–130, Feb. 2014.
[42] N. A. Moubayed, A. Petrovski, and J. McCall, "D^2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evol. Comput.*, vol. 22, no. 1, pp. 47–77, 2014.

VAN TRUONG VU received the B.S. degree in geomatics and the M.S. degree in information systems from Le Quy Don Technical University, Hanoi, Vietnam, in 2010 and 2014, respectively, where he is currently pursuing the Ph.D. degree in information technology.

His research interests include the evolutionary computations specialized with evolutionary multi-objective optimization, machine learning, and GIS and remote sensing.

LAM THU BUI received the Ph.D. degree in computer science from the University of New South Wales (UNSW), Australia, in 2007. He was a Postdoctoral Training with UNSW, from 2007 until 2009. He has been involved with academics, including teaching and research, since 1998. He is currently an Associate Professor with Le Quy Don Technical University, Hanoi, Vietnam. He is researching in the field of evolutionary computation, specialized with evolutionary multi-objective optimization. He was the co-editor of the book *Multi-objective Optimization in Computational Intelligence: Theory and Practice* (Hershey, PA: IGI Global Information Science Reference Series, 2008). He was a member of the Evolutionary Computation Technical Committee, the IEEE Computational Intelligence Society. He has been a member of the program committees of several conferences and workshops in the field of evolutionary computing, such as the IEEE Congress on Evolutionary Computation and the Genetic and Evolutionary Computation Conference.



TRUNG THANH NGUYEN has an international standing in operational research for logistics/transport. He is currently a Reader in operational research (OR) with Liverpool John Moores University, and also the Co-Director with the Liverpool Offshore and Marine Research Institute. He has led more than 20 research projects in transport/logistics, most with close industry collaborations. He has published about 50 peer-reviewed articles. All of his journal articles are in leading journals (ranked 1st-20th in their fields). He co-organized six leading conferences, was a TPC member of more than 30 international conferences, edited eight books, and gave speeches to many conferences/events.