

A modified dual-population approach for solving multi-objective problems

Van Truong Vu
Le Quy Don Technical University.
 Ha Noi, Viet Nam
 Truongvv@mta.edu.vn

Lam Thu Bui
Le Quy Don Technical University
 Ha Noi, Viet Nam
 lam.bui07@gmail.com

Trung Thanh Nguyen
Liverpool John Moores University
 Liverpool, United Kingdom
 T.T.Nguyen@ljmu.ac.uk

Abstract—Maintaining the balance between convergence and diversity plays a vital role in multi-objective evolutionary algorithms (MOEAs). However, most MOEAs cannot reach a satisfying balance, especially when solving problems having complicated pareto optimal sets. In this paper, we present a modified cooperative co-evolution approach for achieving better convergence and diversity simultaneously (namely DPP2). In DPP2, while populations are trying to achieve both criteria, the priority being set for these criteria will be different. One population focuses on achieving better convergence (by using pareto-based ranking scheme), while the other is for ensuring the population diversity (by using the decomposition-based method). After that, we use a cooperation mechanism to integrate the two populations and create a new combined population with hopes of having both characteristics (i.e. converged and diverse). Performance of DPP2 is examined on the well-known benchmarks of multi-objective optimization problems (MOPs) using the hypervolume (HV), the generational distance (GD), the inverted generational distance (IGD) metrics. In comparison with the original version DPP algorithm, experimental results indicated that DPP2 can significantly outperform DPP on the benchmark problems with stable results.

Index Terms—Dual-population, Co-Evolution, Co-operative, convergence, diversity.

I. INTRODUCTION

Most real world problems are with more than one objective. These objectives are often conflicted with each other and their degree of importance might be the same. Therefore, it is difficult to find an exact solution (i.e. it is impossible to optimize all the objectives simultaneously). Instead, a set of optimal solutions (called pareto-optimal solutions) will be selected. Finding as many pareto-optimal solutions as possible plays a crucial role in multi-objective optimization. However, to get all such solutions is a time-consuming task. Instead, algorithms will try to find a set of solutions that satisfies simultaneously both criteria: as close as possible to the pareto-optimal front and as diverse as possible.

In pareto-based algorithms (e.g. NSGA-II and SPEA2), the convergence is considered first and the diversity second. In indicator-based algorithms (e.g. IBEA [1]), DNMOEA/HI [2], both convergence and diversity are simultaneously considered by using an indicator like hypervolume (HV). However, their biggest drawback is the large computational complexity (particularly when the number of objectives is large). Another way to balance convergence and diversity is to apply a

decomposition-based approach (e.g. MOEA/D [3] and MSOPS [4]) which decomposes a complex MOP into a number of sub-problems and solve them in a collaborative manner [5]. In other work, Bui et al [6] proposed a direction-based approach, call DMEA, to maintain a set of non-dominated solutions (the size of this set is depending on users).

Generally, it is difficult to solve MOPs (particularly with balancing convergence and diversity) if only one single algorithm is used. Another approach is that many algorithms are used in conjunction. In [7], authors proposed a multi-algorithm named MABNI, based on NSGA-II and IBEA for balancing convergence and diversity. In [8] authors presented a new hybrid learning algorithm based on a Co-evolutionary algorithm (CoEA) with dual populations for designing RBFNN models with feature selection. The first population aims to discover the most important input features of RBNNs, while the other tends to find the optimal RBFNN structure. In [9] a 2-population co-operative CoEA was presented where the first population is evolved using SPEA2 and the second one uses DEMO/GDE3 with the fitness sharing mechanism. Generally speaking, most CoEAs use co-operative mechanism between multiple populations to solve MOPs.

Inspired by the work done by Ke Li et.al. [10], where they used a dual-population co-operative co-evolution paradigm (DPP) to deal with convergence and diversity simultaneously. In their study, two populations are used. One population uses a pareto-based mechanism to maintain a repository of solutions with a satisfied convergence. The solutions of this population are randomly spread. The other population uses a decomposition-based mechanism to preserve diversity. In order to guarantee this trait, solutions in this population are uniformly populated. Finally, the authors used a restricted mating selection (RMS) mechanism to coordinate the interaction between two co-evolving populations. In RMS, there are two mating parents; one is selected in the first population and the other is selected in the second one. Each parent will be restricted by selecting from its neighboring sub-regions with a large probability. This may lead to the possibility that the one in the first population may not be found (due to its random spread), to address this situation, RMS uses an alternative one in the second population. As a result, both of mating parents are taken from the same population. It makes the co-evolutionary mechanism becomes no meaning anymore. That

is the point for this paper to contribute, in which another mating selection mechanism will be presented.

The rest of this paper is organized as follow. In Section 2, related works are presented including the preliminary concepts and the DPP algorithm. Afterward, the detail of proposed method is illustrated step by step in Section 3. Then, experimental results and discussions are given in Section 4. Finally, conclusions and future works are given in Section 5.

II. RELATED WORKS

A. Preliminary concepts

A multi-objective optimization problem (MOP) can be defined as follows:

Minimize:

$$F(x) = (f_1(x), \dots, f_m(x))^T \quad (1)$$

Subject to: $g_i(x) \leq 0; \forall i = 1, \dots, p$. $h_j(x) = 0; \forall j = 1, \dots, q$.

Where, a solution $x = (x_1, \dots, x_n) \in \Omega$ is the decision variable space or simply the decision space. $g_i(x)$ and $h_j(x)$ are called constraint functions. If any solution x satisfies all constraints and variable bounds, it is known as a feasible solution, otherwise, it is called an infeasible solution. There are m objective functions $F(x) = (f_1(x), \dots, f_m(x))^T$; $F: \Omega \rightarrow \mathbb{R}_+^m$. where \mathbb{R}_+^m is called the objective space. For each solution x in the decision variable space, there exists a point in the objective space.

Definition 1. A solution $x^{(1)}$ can *dominate* another solution $x^{(2)}$, denoted as $x^{(1)} \prec x^{(2)}$ if and only if: $\forall i \in \{1, \dots, m\} : f_i(x^{(1)}) \leq f_i(x^{(2)})$ and $\exists j \in \{1, \dots, m\} : f_j(x^{(1)}) < f_j(x^{(2)})$.

Definition 2. A feasible solution $x^* \in \Omega$ is a *pareto optimal solution* if $\nexists x \in \Omega$ such that $x \prec x^*$.

Definition 3. The set of all pareto optimal solutions is called the *pareto set* (PS), denoted as $PS = \{x^* \in \Omega \mid \nexists x \in \Omega, x \prec x^*\}$.

Definition 4. The set of all objective function values corresponding to the solutions in PS is called the *pareto front* (PF), denoted as $PF = \{F(x) \mid x \in PS\}$.

Definition 5. The ideal objective vector is $Z^* = (f_1^*, \dots, f_m^*)^T$. Where f_k^* is the minimum value of the k -th objective function, for all k in $\{1, \dots, m\}$.

Definition 6. The nadir objective vector is $Z^{nad} = (f_1^{nad}, \dots, f_m^{nad})^T$. Where f_k^{nad} is the maximum value of the k -th objective function, for all k in $\{1, \dots, m\}$.

B. The Dual-population Paradigm (DPP)

The general architecture of DPP [10] is given in Alg.1 and Fig.1. There are two co-evolving populations: the first population, denoted as A_p , is evolved using the pareto-based mechanism; the other one, named A_d , is evolved using the decomposition-based mechanism. These populations will evolve in parallel. At each generation, they will interact with each other through a restricted mating selection mechanism (RMS).

In RMS, two parents are selected to mate using the Cooperative mechanism. This way, the offspring can inherit all

Algorithm 1: DPP Algorithm

input : Maximum number of generations (M)
Neighborhood Size (T)
Population size (N)
output: Final Population P

- 1 $[A_p, A_d] = \text{initializePopulation}()$
- 2 $m \leftarrow 0$
- 3 **while** $m < M$ **do**
- 4 $\text{offspring}A_p \leftarrow \emptyset$
- 5 **for** $i \leftarrow 1$ **to** N **do**
- 6 $Q = \text{ReComMatSelection}(A_p, A_d, m, Bm)$
- 7 $Child = \text{CoOperativeMating}(Q)$
- 8 $\text{Mutate}(Child)$;
- 9 Update Sub-Region index for $Child$ in A_p and A_d respectively
- 10 Update Ideal point Z^* and nadir point Z^{nad}
- 11 Update A_p and A_d
- 12 $m++$;
- 13 **end**
- 14 **end**
- 15 Return $P \leftarrow A_p \cup A_d$

the good features (i.e. the convergence and diversity) from both parents. This offspring is used to update each of A_p and A_d , respectively, based on the corresponding archiving mechanism. Note that, this update step is performed for each sub-region in every generation. Besides, there are two cases in RMS process. The first, the selected sub-region in A_p does not contain any individual. In this case, RMS chooses an alternative one in the corresponding sub-region in A_d . In the second case, the selected sub-region in A_p contains more than one individuals, one of them is randomly selected.

III. THE PROPOSED METHOD

The general diagram of the modified dual-population approach (DPP2) is given in Fig.1 and the pseudo code of this algorithm is showed in Alg.2 A more detailed explanation of DPP will be showed in here.

At the first step, A_p and A_d (for simplicity, they have same size N) are randomly initialized. N solutions in A_d are evenly assigned to N sub-regions (according to N unit vectors). Later, in the process of evolution, each sub-region always has only one solution. This is to guarantee that A_d always has an even distribution (i.e. has diversity) in objective space. Whereas, N solutions in A_p will be randomly assigned to N sub-regions. This means, more than one solution can be in the same sub-region and also there are sub-regions that don't contain any solutions. Next, each solution specifies T closest neighborhood sub-regions based on the Euclidean distance between unit vectors.

As mentioned before. In [10], the authors used an RMS mechanism to select two mating parents. Ideally, one solution is selected from A_p and the other is chosen from A_d . However, there is no guarantee that each solution in A_p is associated with

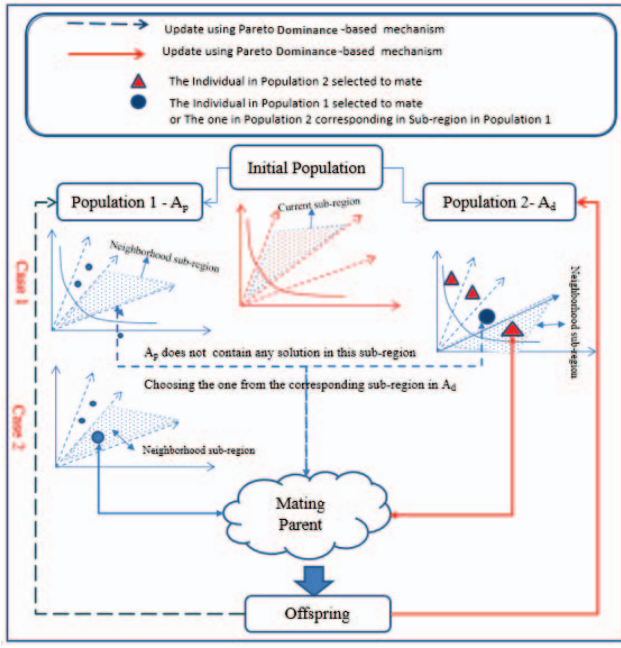


Fig. 1. Diagram of the DPP algorithm. In the first case, the selected neighborhood sub-region doesn't contain any solution (the alternative solution is selected from the corresponding sub-region in A_d), meanwhile, in the second case, this sub-region contains at least one solution (a random solution in this sub-region is selected).

a sub-region. Hence, when A_p does not contain any solution in the selected sub-region, RMS utilizes an alternative solution from the corresponding sub-region in A_d (see case 1 in Fig.1). At this time, the offspring is generated from parents in the same population. As a result, the offspring cannot take the advantages from both of populations. This might lead to an imbalance between diversity and convergence.

After the selection process, two mating parents (denoted to x_{r2} and x_{r3}) will be selected for the co-operative process. To generate new offspring from these mating parents, we borrow the reproduction idea from MOEA/D-DE [4].

One thing to be underlined here is that the new offspring need to be assigned to a certain sub-region. In this research, this offspring belongs to the sub-region which has the minimum Euclidian distance between its unit vector and the offspring's objective vector.

Finally, the new offspring is used to update each of A_p and A_d , respectively.

In general, the DPP2 has three main differences from the DPP:

First, when choosing one solution in A_p , instead of just selecting from a selected neighborhood sub-region we will select from all T neighborhood sub-regions. By doing this, the probability of finding one solution in A_p will be much higher than in RMS.

Second, in case all T neighborhood sub-regions do not contain any solution. Instead choosing alternative solution in A_d , we randomly select a solution in A_p . By this way, the offspring is generated from parents in the different populations

so they can take all advantages from both of parents (i.e. the diversity and the convergence).

Third, the update procedure of A_p is different from the original DPP. In particular, whenever a new offspring is generated, it will be stored in an offspring list (i.e. $offSpringAp$ in Alg.2) instead of being updated right away to A_p . After a generation finishes, $offSpringAp$ will be combined with A_p and we use the crowding distance sorting method (CDSM) in the combined population to select N best solutions for the new population. The reason for this approach is that the CDSM is a really time-consuming method. Assume that the maximum number of generation is $M=300,000$ and the population size is $N=300$ then there are 300,000 new offspring generated. Thus, the CDSM will be called 300,000 times in DPP. Meanwhile, 1000 is the number for DPP2. Apparently, with this new update mechanism, the computing time has been greatly reduced (i.e. N times).

In summary, based on the above discussion, we can conclude that DPP2, in principle, has many potentials in tackling this kind of problems.

IV. EXPERIMENTS

In the original paper, the authors compared the DPP algorithm with baseline algorithms (i.e. NSGA-II and MOEA/D-DE). In this research, we focus on comparing the proposed algorithm DPP2 with the original algorithms DPP.

Algorithm 2: DPP2 algorithm

input : Maximum number of generations (M)
Neighborhood Size (T)
Population size (N)

output: Final Population P

- 1 $[A_p, A_d] = initializePopulation()$
- 2 $W = InitializeUniformWeight()$
- 3 $B = InitializeNeighborhood()$
- 4 $Z^* = InitializeIdealPoint()$
- 5 $Z^{nad} = InitializeNadirPoint()$
- 6 $m \leftarrow 0$
- 7 **while** $m < M$ **do**
- 8 $offspringAp \leftarrow \emptyset$
- 9 **for** $i \leftarrow 1$ **to** N **do**
- 10 $Q = RMS2(A_p, A_d, m, Bm)$
- 11 $Child = CoOperativeMating(Q)$
- 12 Mutate($Child$);
- 13 Update Sub-Region index for $Child$
- 14 Update Ideal point Z^* and nadir point Z^{nad}
- 15 Update A^d
- 16 **Add Child to** $offspringAp$
- 17 $m++$;
- 18 **end**
- 19 $U = \mathbf{Union}(offspringAp, A_p)$
- 20 $A_p = \mathbf{crowdingDistanceSelection}(U)$
- 21 **end**

Algorithm 3: RMS2(A_p, A_d, m, B_m)

input : A_p (Pareto-based archive)
 A_d (Decomposition-based archive)
 m : current subregion index
 B_m : A Set contains neighborhood indexes of current subregion.
 T : the neighborhood size;
 N : the Population size
output: Two mating parent(Q).

```
1  $P_1 = \text{MatSelectionAp}()$ 
2  $P_2 = \text{MatSelectionAd}()$ 
3 Return  $Q = (P_1, P_2)$ 
```

Algorithm 4: MatSelectionAp(A_p, m, B_m)

input : A_p (Pareto-based archive)
 m : current subregion index
 B_m : A Set contains neighborhood indexes of current subregion.
 T : the neighborhood size;
 N : the Population size
output: A subregion index.

```
1 listNeighborAp  $\leftarrow \emptyset$ 
2 if rand < neighborhoodSelectionProbability then
3   // Select a sub-region index in  $A_p$ 
4   for  $i \leftarrow 0$  to  $T$  do
5     for  $j \leftarrow 0$  to  $N$  do
6       if  $A_p[j] \in B_m[i]$  then
7         Return  $j$ ;
8       end
9     end
10 end
11 else
12   Randomly select an index from  $\{1, 2, \dots, N\}$ 
13 end
```

Algorithm 5: MatSelectionAd(A_d, m, B_m)

input : A_d (Decomposition-based archive)
 m : current subregion index
 B_m : A Set contains neighborhood indexes of current subregion.
 T : the neighborhood size;
 N : the Population size
output: a subregion index.

```
1 if rand < neighborhoodSelectionProbability then
2   Randomly select an index from  $B_m$ 
3 end
4 else
5   Randomly select an index from  $\{1, 2, \dots, N\}$ 
6 end
```

A. Test problem

In order to evaluate the performance of the proposed method, both of algorithms are tested on ZDT [12], DTLZ-series [12], WFG [12] and UF [13].

B. Performance metrics

There are several performance metrics to measure various performance aspects of MOEAs such as generational distance (GD) [11], spacing metric (SP) [11], hypervolume (HV) [14] and inverted generational distance (IGD) [15], or stability [13]. GD evaluates the convergence; SP evaluates the uniformity; IGD, as well as HV evaluates both of the convergence and the diversity of a solution set. In this research, we will use IGD, GD and HV as key metrics. It is important to note that the larger HV value is, the better quality of a solution is. In contrast to HV, a low value of IGD and GD is desirable.

C. Parameters settings of MOEAs

The parameters of NSGA-II and MOEA/D-DE are set as Table I. Each algorithm is launched 20 times for each test instance. The population size is set $N=300$ and a maximum number of generation is set $M=300.000$.

TABLE I
THE PARAMETER SETTING OF THE MOEAs

MOEAs	Parameters settings
NSGA-II	$p_c=0.9, p_m=1/n$ variables; $\mu_c=20; \mu_m=20$
MOEA/D-DE	$p_m=1/n$ variables, $\mu_m=20, CR=1.0, F=0.5, \sigma=0.9, T=20, \text{"rand/1/bin"}; n$ replaced=2

D. Experimental results and discussions

The performance comparisons are presented from Table II to Table V via the mean and standard deviation values. The best mean metric value is highlighted.

In Table II, we conduct the comparison between DPP2 and DPP using the GD metric. In ZDT instances, DPP gives GD values with numerical precision of 10^{-1} , whereas the DPP reached to 10^{-5} . In UF instances, DPP2 give results better than DPP about 10^2 times with UF1, UF2, UF3 and 10^1 time with others. In WFG instances, in general, DPP2 achieves better GD metric values about 10^2 than DPP. Finally, with the DTLZ instances, DPP2 shows better performance than DPP about 10^3 times (except 10^1 in the case of DTLZ4).

From the results, we can see that DPP2 shows better performance than DPP in all instances in terms of convergence. In particular, DPP2 outperforms DPP in ZDT and DTLZ series.

Table III provides the performance comparisons of DPP2 and DPP using HV metric. Similar to the observation in Table II, DPP2 achieves better metric values in all of comparisons and DPP2 outperforms DPP in ZDT and WFG series.

TABLE II
GD. MEAN AND STANDARD DEVIATION

	DPP	DPP2
ZDT1	1.753648e - 01 _{1.5e-03}	3.816960e - 05 _{2.6e-06}
ZDT2	2.589376e - 01 _{1.7e-03}	2.886810e - 05 _{4.0e-07}
ZDT3	1.065891e - 01 _{1.4e-03}	5.289977e - 05 _{3.5e-07}
ZDT4	9.182252e + 00 _{1.1e-01}	3.467797e - 05 _{1.1e-06}
ZDT6	4.724667e - 01 _{1.1e-03}	2.671673e - 05 _{3.4e-08}
UF1	1.335208e - 01 _{2.1e-03}	1.720110e - 04 _{2.6e-04}
UF2	6.449769e - 02 _{1.1e-03}	6.337553e - 04 _{2.5e-04}
UF3	1.436763e - 01 _{2.2e-03}	6.135835e - 04 _{4.5e-04}
UF4	1.331551e - 02 _{6.9e-05}	3.750404e - 03 _{2.4e-04}
UF5	4.467954e - 01 _{2.7e-03}	2.016716e - 02 _{8.8e-03}
UF6	5.984329e - 01 _{5.5e-03}	1.146247e - 02 _{4.3e-03}
UF7	1.385920e - 01 _{1.4e-03}	4.490829e - 04 _{7.9e-04}
UF8	4.209743e - 01 _{6.9e-03}	4.377499e - 02 _{3.2e-02}
UF9	4.380817e - 01 _{9.3e-03}	1.214498e - 01 _{5.5e-02}
UF10	1.896500e + 00 _{3.2e-02}	8.642352e - 02 _{5.1e-02}
WFG1	3.018711e - 02 _{6.8e-05}	1.054068e - 04 _{1.8e-07}
WFG2	2.432470e - 02 _{6.5e-04}	3.164339e - 04 _{9.7e-08}
WFG3	1.745607e - 02 _{6.5e-04}	1.158201e - 04 _{7.2e-08}
WFG4	1.067246e - 02 _{1.7e-04}	4.390308e - 04 _{5.1e-06}
WFG5	1.716240e - 02 _{2.2e-04}	1.566477e - 03 _{2.9e-06}
WFG6	1.876775e - 02 _{4.1e-04}	1.779998e - 04 _{1.4e-07}
WFG7	1.358835e - 02 _{2.5e-04}	1.847863e - 04 _{2.7e-07}
WFG8	2.162389e - 02 _{3.7e-04}	5.499032e - 03 _{3.7e-03}
WFG9	2.171889e - 02 _{3.3e-04}	5.143616e - 04 _{3.3e-06}
DTLZ1	2.426916e + 01 _{4.9e-01}	3.945053e - 04 _{2.4e-06}
DTLZ2	4.921139e - 02 _{6.9e-04}	4.137693e - 04 _{7.0e-06}
DTLZ3	6.254630e + 01 _{9.1e-01}	6.590347e - 04 _{9.8e-06}
DTLZ4	5.080821e - 02 _{8.4e-04}	3.491099e - 03 _{2.9e-05}
DTLZ5	6.189454e - 02 _{1.5e-03}	1.495443e - 04 _{1.1e-06}
DTLZ6	6.427798e - 01 _{6.1e-03}	3.282463e - 04 _{2.2e-08}
DTLZ7	2.154503e - 01 _{2.1e-03}	7.528934e - 04 _{4.2e-05}

TABLE III
HV. MEAN AND STANDARD DEVIATION

	DPP	DPP2
ZDT1	3.996240e - 02 _{8.1e-02}	6.648521e - 01 _{4.4e-05}
ZDT2	1.791135e - 05 _{8.0e-05}	3.315724e - 01 _{4.0e-05}
ZDT3	7.759302e - 03 _{2.2e-02}	5.162230e - 01 _{9.9e-06}
ZDT4	0.000000e + 00 _{0.0e+00}	6.649724e - 01 _{8.8e-06}
ZDT6	1.513068e - 05 _{2.7e-05}	4.047281e - 01 _{2.2e-07}
UF1	6.400829e - 03 _{1.2e-02}	6.635530e - 01 _{2.4e-04}
UF2	1.910077e - 01 _{2.3e-02}	6.570243e - 01 _{3.2e-03}
UF3	5.080904e - 04 _{1.2e-03}	6.490813e - 01 _{1.6e-02}
UF4	1.197737e - 01 _{3.8e-03}	2.450716e - 01 _{5.5e-03}
UF5	0.000000e + 00 _{0.0e+00}	8.646766e - 02 _{8.3e-02}
UF6	0.000000e + 00 _{0.0e+00}	2.045398e - 01 _{1.0e-01}
UF7	2.768829e - 03 _{9.5e-03}	4.949293e - 01 _{2.6e-03}
UF8	0.000000e + 00 _{0.0e+00}	3.155399e - 01 _{1.6e-02}
UF9	0.000000e + 00 _{0.0e+00}	6.204633e - 01 _{5.8e-02}
UF10	0.000000e + 00 _{0.0e+00}	5.960499e - 02 _{2.7e-02}
WFG1	4.530206e - 02 _{1.8e-03}	6.348506e - 01 _{2.1e-04}
WFG2	3.850254e - 01 _{2.0e-02}	5.646771e - 01 _{1.2e-05}
WFG3	3.238837e - 01 _{1.2e-02}	4.980028e - 01 _{4.5e-06}
WFG4	1.309644e - 01 _{6.4e-03}	2.212260e - 01 _{1.1e-04}
WFG5	1.041140e - 01 _{6.2e-03}	1.981450e - 01 _{1.0e-05}
WFG6	9.195614e - 02 _{8.5e-03}	2.128722e - 01 _{4.2e-06}
WFG7	1.042489e - 01 _{6.6e-03}	2.128553e - 01 _{5.7e-06}
WFG8	7.370458e - 02 _{5.1e-03}	1.730381e - 01 _{2.6e-02}
WFG9	1.373612e - 01 _{1.6e-02}	2.438651e - 01 _{6.3e-05}
DTLZ1	0.000000e + 00 _{0.0e+00}	7.849109e - 01 _{3.6e-04}
DTLZ2	9.580321e - 02 _{1.5e-02}	4.184846e - 01 _{7.7e-04}
DTLZ3	0.000000e + 00 _{0.0e+00}	4.187471e - 01 _{1.0e-03}
DTLZ4	4.071083e - 02 _{1.5e-02}	4.118296e - 01 _{2.2e-02}
DTLZ5	3.329375e - 03 _{3.0e-03}	9.469711e - 02 _{5.7e-06}
DTLZ6	0.000000e + 00 _{0.0e+00}	9.566277e - 02 _{8.1e-07}
DTLZ7	8.433473e - 02 _{4.1e-02}	2.723934e - 01 _{2.1e-02}

In summary, from the results, we can see that DPP2 shows better performance more than DPP in all instances. These observations demonstrate the effective of the modified dual-population paradigm when taking into account all neighborhood sub-regions instead of using only random sub-region like

TABLE IV
IGD. MEAN AND STANDARD DEVIATION

	DPP	DPP2
ZDT1	1.467296e - 02 _{2.9e-03}	5.553788e - 05 _{1.7e-07}
ZDT2	2.518171e - 02 _{7.8e-03}	4.660187e - 05 _{4.9e-08}
ZDT3	1.386081e - 02 _{3.0e-03}	8.861649e - 05 _{5.4e-07}
ZDT4	9.770826e - 01 _{1.8e-01}	5.900098e - 05 _{3.3e-07}
ZDT6	2.702415e - 02 _{3.3e-03}	4.627856e - 05 _{3.9e-09}
UF1	2.757694e - 02 _{3.8e-03}	7.137613e - 05 _{7.6e-06}
UF2	1.315680e - 02 _{5.2e-04}	4.182316e - 04 _{1.6e-04}
UF3	2.729877e - 02 _{3.0e-03}	6.285715e - 04 _{8.8e-04}
UF4	5.606466e - 03 _{1.2e-04}	1.955271e - 03 _{1.4e-04}
UF5	8.764953e - 01 _{9.3e-02}	6.882732e - 02 _{1.5e-02}
UF6	1.139303e - 01 _{2.1e-02}	6.071352e - 03 _{6.1e-03}
UF7	2.919063e - 02 _{4.9e-03}	2.521310e - 04 _{3.7e-04}
UF8	2.162162e - 02 _{4.7e-03}	1.201758e - 03 _{2.0e-04}
UF9	1.876371e - 02 _{4.0e-03}	1.416785e - 03 _{7.5e-04}
UF10	1.002259e - 01 _{1.4e-02}	4.870078e - 03 _{6.2e-04}
WFG1	1.803620e - 02 _{1.1e-04}	2.678682e - 04 _{1.5e-05}
WFG2	1.139491e - 02 _{1.2e-03}	6.028959e - 04 _{9.1e-06}
WFG3	4.125034e - 03 _{3.6e-04}	5.478467e - 05 _{7.7e-08}
WFG4	1.905605e - 03 _{1.3e-04}	6.323277e - 05 _{3.9e-06}
WFG5	4.206134e - 03 _{2.8e-04}	9.326983e - 04 _{5.2e-07}
WFG6	5.724722e - 03 _{5.9e-04}	1.139314e - 05 _{9.8e-08}
WFG7	1.979212e - 03 _{1.8e-04}	4.054533e - 05 _{1.6e-08}
WFG8	6.701183e - 03 _{3.1e-04}	2.362629e - 03 _{1.1e-03}
WFG9	1.808642e - 03 _{2.7e-04}	4.074258e - 05 _{1.9e-07}
DTLZ1	2.953435e - 01 _{7.0e-02}	3.474254e - 04 _{1.1e-06}
DTLZ2	3.253742e - 03 _{1.7e-04}	4.306460e - 04 _{2.0e-06}
DTLZ3	2.822716e + 00 _{2.6e-01}	7.225804e - 04 _{4.3e-06}
DTLZ4	4.216476e - 03 _{1.7e-04}	8.112918e - 04 _{1.9e-04}
DTLZ5	7.508758e - 04 _{6.8e-05}	1.519963e - 05 _{1.2e-07}
DTLZ6	1.357017e - 02 _{3.6e-03}	3.453052e - 05 _{2.9e-08}
DTLZ7	2.453883e - 02 _{5.3e-03}	6.889806e - 03 _{7.7e-03}

in RMS mechanism.

V. CONCLUSIONS

In this paper, a modified dual-population based co-evolutionary approach (DPP2) is presented. In DPP2, two populations are simultaneously maintained, one based on the Pareto-based technique to achieve better convergence and the other based on the decomposition-based technique to preserve diversity. These populations interact with each other via a new restricted competitive mating selection mechanism (RMS2) and a co-operative mechanism. The performance of the proposed algorithm is compared with the original version DPP on the 4 series test instances via GD, HV, IGD and IGD+ metrics. The empirical results demonstrated the effectiveness of new co-operative co-evolutionary approach for achieving a balance between convergence and diversity on the test instances.

This is just our preliminary study, a series of other co-evolutionary variants with more better results on other benchmark problems will be tested and presented in the near future.

ACKNOWLEDGMENT

This work was supported by Newton Research Collaborations Programme grants funded by the UK BEIS via the Royal Academy of Engineering under grant numbers NRCP1516/1/134 and NRCP1617/6/125.

REFERENCES

- [1] Zitzler, E. and S. Knzli. "Indicator-based selection in multiobjective search". in International Conference on Parallel Problem Solving from Nature. 2004. Springer.

TABLE V
IGD+. MEDIAN AND INTERQUARTILE RANGE

	DPP	DPP2
ZDT1	3.981009e - 01 _{1.4e-01}	8.767672e - 04 _{3.5e-05}
ZDT2	5.900366e - 01 _{4.3e-01}	8.294902e - 04 _{2.1e-05}
ZDT3	3.790287e - 01 _{1.7e-01}	1.070006e - 03 _{9.2e-06}
ZDT4	3.311098e + 01 _{8.6e+00}	8.137764e - 04 _{5.5e-06}
ZDT6	4.999849e - 01 _{1.8e-05}	8.504027e - 04 _{1.6e-07}
UF1	8.862074e - 01 _{1.4e-01}	1.544448e - 03 _{9.4e-05}
UF2	3.877046e - 01 _{5.0e-02}	6.089317e - 03 _{2.5e-03}
UF3	8.433913e - 01 _{1.5e-01}	6.244716e - 03 _{1.2e-02}
UF4	1.761403e - 01 _{5.8e-03}	5.940100e - 02 _{4.9e-03}
UF5	4.144999e + 00 _{5.1e-01}	2.659879e - 01 _{4.4e-02}
UF6	3.754085e + 00 _{1.2e+00}	7.815053e - 02 _{9.8e-02}
UF7	9.182482e - 01 _{2.5e-01}	2.115369e - 03 _{9.9e-04}
UF8	2.255779e + 00 _{6.9e-01}	6.348187e - 02 _{8.6e-03}
UF9	1.817217e + 00 _{6.3e-01}	7.194544e - 02 _{9.6e-02}
UF10	9.921315e + 00 _{2.2e+00}	4.190577e - 01 _{7.6e-02}
WFG1	5.984753e - 01 _{4.5e-03}	1.181707e - 03 _{3.0e-05}
WFG2	1.112766e - 01 _{1.9e-02}	1.100278e - 03 _{1.9e-05}
WFG3	1.064674e - 01 _{1.9e-02}	9.339623e - 04 _{3.5e-06}
WFG4	5.804759e - 02 _{5.3e-03}	1.225275e - 03 _{8.8e-05}
WFG5	9.449718e - 02 _{7.2e-03}	2.673263e - 02 _{2.2e-05}
WFG6	8.699523e - 02 _{1.0e-02}	9.315293e - 04 _{3.3e-06}
WFG7	8.411806e - 02 _{1.3e-02}	1.139049e - 03 _{6.2e-06}
WFG8	1.361509e - 01 _{9.5e-03}	3.049303e - 02 _{3.1e-02}
WFG9	6.890068e - 02 _{1.5e-02}	1.268092e - 03 _{3.2e-05}
DTLZ1	3.068257e + 01 _{1.2e+01}	2.097064e - 02 _{1.2e-04}
DTLZ2	2.034966e - 01 _{2.7e-02}	1.944945e - 02 _{2.5e-04}
DTLZ3	1.753278e + 02 _{2.0e+01}	2.194771e - 02 _{2.8e-04}
DTLZ4	1.098057e - 01 _{1.6e-02}	1.448204e - 02 _{3.3e-03}
DTLZ5	2.033893e - 01 _{4.2e-02}	1.650981e - 03 _{1.6e-05}
DTLZ6	2.219135e + 00 _{1.3e+00}	1.474524e - 03 _{2.4e-06}
DTLZ7	2.529333e - 01 _{9.6e-02}	3.668727e - 02 _{4.9e-02}

- [2] Li, K., et al., "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm". Information Sciences, 2012. 182(1): p. 220-242.
- [3] Zhang, Q. and H. Li, MOEA/D: "A multiobjective evolutionary algorithm based on decomposition". IEEE transactions on evolutionary computation, 2007. 11(6): p. 712-731.
- [4] Li, H. and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II". IEEE transactions on evolutionary computation, 2009. 13(2): p. 284-302.
- [5] Liu, H.-L., F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems". IEEE transactions on evolutionary computation, 2014. 18(3): p. 450-455.
- [6] Bui, L.T., Liu, J., Bender, A., Barlow, M., Wesolkowski, Abbass, H.A., "DMEA: A Direction-based Multiobjective Evolutionary Algorithm", Memetic Computing Journal, Vol 3, No 4, pp 271-285, 2011.
- [7] Xie, D., et al., "A Multi-Algorithm Balancing Convergence and Diversity for Multi-Objective Optimization". J. Inf. Sci. Eng., 2013. 29(5): p. 811-834.
- [8] Tian, J., M. Li, and F. Chen, "Dual-population based co-evolutionary algorithm for designing RBFNN with feature selection". Expert Systems with Applications, 2010. 37(10): p. 6904-6918.
- [9] Zvoianu, A.-C., et al. "Efficient multi-objective optimization using 2-population co-operative co-evolution". in International Conference on Computer Aided Systems Theory. 2013. Springer.
- [10] Li, K., S. Kwong, and K. Deb, "A dual-population paradigm for evolutionary multiobjective optimization". Information Sciences, 2015. 309: p. 50-72.
- [11] Van Veldhuizen, D.A. and G.B. Lamont. "On measuring multiobjective evolutionary algorithm performance". in Evolutionary Computation, 2000. Proceedings of the 2000 Congress on. 2000. IEEE.
- [12] Huband, Simon, et al. "A review of multiobjective test problems and a scalable test problem toolkit." IEEE Transactions on Evolutionary Computation 10.5 (2006): 477-506.
- [13] Zhang, Qingfu, et al. "Multiobjective optimization test instances for the CEC 2009 special session and competition." University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report 264 (2008).
- [14] Zitzler, E. and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach". IEEE transactions on evolutionary computation, 1999. 3(4): p. 257-271.
- [15] Zhang, Q., et al., "Multiobjective optimization test instances for the CEC 2009 special session and competition". University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report, 2008. 264.
- [16] Bui, L.T., et al. "A dominance-based stability measure for multi-objective evolutionary algorithms". in Evolutionary Computation, 2009. CEC'09. IEEE Congress on. 2009. IEEE.