Evolutionary Approach to Multiparty Multiobjective Optimization Problems with Common Pareto Optimal Solutions

Wenjie Liu*, Wenjian Luo†, Xin Lin*, Miqing Li‡, and Shengxiang Yang§
*School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, Anhui, China
†School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, Guangdong, China
‡Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA),
School of Computer Science, University of Birmingham, B15 2TT, UK
§the School of Computer Science and Informatics, the De Montfort University, Leicester LE1 9BH, UK
Email: lwj1217@mail.ustc.edu.cn, luowenjian@hit.edu.cn, iskcal@mail.ustc.edu.cn, m.li.8@cs.bham.ac.uk, syang@dmu.ac.uk

***Abstract*—Some real-world optimization problems involve multiple decision makers holding different positions, each of whom has multiple conflicting objectives. These problems are defined as multiparty multiobjective optimization problems (MPMOPs). Although evolutionary multiobjective optimization has been widely studied for many years, little attention has been paid to multiparty multiobjective optimization in the field of evolutionary computation. In this paper, a class of MPMOPs, that is, MPMOPs having common Pareto optimal solutions, is addressed. A benchmark for MPMOPs, obtained by modifying an existing dynamic multiobjective optimization benchmark, is provided, and a multiparty multiobjective evolutionary algorithm to find the common Pareto optimal set is proposed. The results of experiments conducted using the benchmark show that the proposed multiparty multiobjective evolutionary algorithm is effective.**

***Index Terms*—Multiobjective optimization, evolutionary computation, multiparty multiobjective optimization**

I. Introduction

A. Motivation

In business, scientific, and social political decision making, multiple decision makers (DMs) holding different positions, e.g., working in different departments/units, are frequently involved, and each DM, having a specific perspective of the same problem, may have multiple conflicting objectives. Such problems are called multiparty multiobjective optimization problems (MPMOPs).

In real life, the purchase of a family car, where a husband and wife may consider different objectives, is a good example of an MPMOP. One person (e.g., the husband) may consider two objectives, (f_{H1} and f_{H2}), where f_{H1} represents the price and f_{H2} represents the acceleration performance. The husband pays considerable attention to a lower price and better acceleration performance but is not concerned about other perspectives. At the same time, the second person (e.g., the wife) considers two different main objectives (f_{W1} and f_{W2}), where f_{W1} represents the aesthetic level and f_{W2} represents the interior space. The wife wants the car to be beautiful and the interior space of the car to be large. Therefore, we need to find the cars that are Pareto optimal for both parties.

Another example related to MPMOPs is the problem of scheduling reservoir water resources. This problem is frequently decided by multiple departments, where each department may consider the solutions from a different perspective, e.g. the electricity sector and the water sector. Assume that x is a decision variable, such as water storage or water release. The electricity sector always considers reservoir scheduling from the perspective of hydroelectric power generation, including the objectives (f_{E1} and f_{E2}) [1], where

(1) f_{E1} is hydropower generation, which should be maximized. The aim is to utilize hydropower fully to minimize the residual load after the hydropower output deducted, which reduces the pollution caused by thermal power generation.
(2) f_{E2} is the peak-to-valley difference of the residual load, which should be minimized. Hydropower generation is used to adjust the residual load. Thus, the fluctuation of thermal power generation is not excessive.

The water sector considers mainly the safety factors of the reservoir [2] and the demands for water supply. Thus, the reservoir scheduling is considered from the viewpoint of two objectives (f_{W1} and f_{W2}), where

(1) f_{W1} is the safety goal, which should be maximized. It requires that the water discharge from the reservoir be increased; that is, a reduction in the water level of the reservoir benefits dam safety and flood prevention safety.
(2) f_{W2} is the water supply goal, which should be maximized. It requires that the water discharge be reduced and the reservoir storage be increased to help meet the long-term water demands (such as the industrial water supply, agricultural irrigation, and urban water supply).

As discussed above, the electricity sector considers the

This work is partly supported by the National Natural Science Foundation of China (No. 61573327). *(Corresponding author: Wenjian Luo.)*

problem of reservoir water storage and discharge from the perspective of hydropower generation, whereas the water sector controls the reservoir water storage capacity and water discharge from the viewpoint of safety and water supply. Certain conflicts exist between the two objectives for each sector, and their scheduling schemes differ because of their different perspectives.

Multiobjective evolutionary optimization has been studied for many years [3], [4]. Thus far, many multiobjective evolutionary algorithms (MOEAs) have been proposed, e.g., NSGA-II [5], SPEA2 [6], PAES [7], MOEA/D [8], and NSGA-III [9]. However, few studies have been conducted on MPMOPs in the field of evolutionary computation.

Existing MOEAs cannot be applied to MPMOPs directly. A simple example is as follows. Suppose a problem in which two DMs are involved, each of whom has two minimization objectives. Given two solutions, x_1 and x_2, the objective values of the first DM are $F_1(x_1) = (1.0, 2.0)$ and $F_1(x_2) = (11, 21)$ of the second DM are $F_2(x_1) = (3.1, 4.0)$ and $F_2(x_2) = (3.0, 4.1)$. If we consider these four objectives of the two DMs together, we may have the objective values

$$(1.0, 2.0, 3.1, 4.0) \textit{ for } x_1$$

and

$$(11, 21, 3.0, 4.1) \textit{ for } x_2.$$

Evidently, if we consider these four objectives together, x_1 and x_2 are equally good. However, for the first DM, x_1 is better than x_2, whereas for the second, they are equally good. The reason is that, whereas the solutions for four objectives do not dominate each other, in the case of two objectives they may have a dominance relationship. In fact, from the viewpoint of multiparty multiobjective optimization (MPMO), x_1 is better than x_2.

To the best of our knowledge, multiparty multiobjective negotiations may be the issue that most closely resembles MPMO. However, they differ from each other. A review of multiparty multiobjective negotiations is presented in Section II-C.

B. Contributions

In this paper, a class of MPMOPs, where the intersection of the Pareto optimal sets (PSs) of each party is not empty, is addressed. Thus, the objective is to find **the set of the common Pareto optimal solutions** of the multiobjective optimization problems (MOPs) of all DMs. Thereby, complex negotiations are avoided, because all solutions in the common PS are non-inferior solutions for all DMs.

In this paper, a set of MPMO benchmark, which is based on the dynamic multiobjective optimization (DMO) benchmark [10], is given. We propose an evolutionary-based algorithm to solve MPMOPs. Our experimental results demonstrate that the proposed algorithm is more effective than a baseline algorithm straightforwardly modified from a typical MOEA.

C. Organization

The rest of this paper is organized as follows. Section II introduces the definition of MOPs and a review of typical MOEAs and multiparty multiobjective negotiations. Section III describes the benchmark. The algorithm to solve MPMOPs is presented in Section IV. Our experiments and results are described in Section V. Finally, Section VI concludes this paper.

II. Review of Multiobjective Optimization

In this section, we first introduce the definition of MOPs. Then, typical MOEAs and multiparty multiobjective negotiations are reviewed.

A. Multiobjective Optimization

The procedure of solving an MOP consists of optimizing more than one conflicting objective. Here, we take the minimization objective as an example. It can be defined as [11]

$$\begin{aligned} &Min\ F(x) = (f_1(x), f_2(x) ..., f_m(x)), \\ &Subject\ to \begin{cases} g_c(x) \le 0, c = 1, ..., c_g, \\ h_c(x) = 0, c = c_g + 1, ..., c_g + c_h, \\ x \in [x_{min}, x_{max}]^d, \end{cases} \end{aligned} \tag{1}$$

where $g_c(x)$ represents the inequality constraints and $h_c(x)$ represents the equality constraints; c_g and c_h are the numbers of the inequality constraints and equality constraints, respectively. $x = (x_1, x_2, \ldots, x_d)$ is a d-dimensional vector, which represents the decision variables, and x_{min} and x_{max} denote the lower and upper bounds, respectively.

A decision vector x dominates a second decision vector y under the following two conditions [12]. 1) For each objective $i \in \{1, \ldots, m\}$, $f_i(x) \le f_i(y)$. 2) There exists at least one objective $j \in \{1, \ldots, m\}$ that satisfies $f_j(x) < f_j(y)$. In general, that x *dominates* y is denoted by $x \prec y$.

Based on the domination definition, a decision vector $x^* \in \Omega$ is **Pareto optimal** if and only if there exists no decision vector $x \in \Omega$ that can dominate x^*. The **PS** is defined as the collection of all Pareto optimal solutions; i.e., $PS = \{x^* \in \Omega | \nexists x \in \Omega, x \prec x^*\}$.

Meanwhile, the PS on the objective space constitutes the Pareto optimal front (**PF**), which can be defined as $PF = \{f = (f_1(x^*), \ldots, f_m(x^*)) | x^* \in PS\}$.

B. Multiobjective Evolutionary Algorithms (MOEAs)

MOEAs can find an optimal solution set in a run. In the last few decades, many MOEAs have been proposed to solve MOPs [3], [4].

NSGA-II [5] is one of the representative MOEAs based on Pareto dominance, and many variants of it have been proposed. For example, RD-NSGA-II [13] uses the reference direction to guide the search process of NSGA-II. NSGA-III [9], adapted from NSGA-II, distributes the solutions more widely and uniformly in many-objective optimization where more than three objectives are involved.

MOEA/D [8] is an additional representative MOEA, which decomposes an MOP into N scalar quantum subproblems and simultaneously optimizes N subproblems by using the solutions of adjacent subproblems. Further, a new version of MOEA/D based on differential evolution, i.e., MOEA/D-DE, was proposed in [14]. UMOEA/D, proposed by Tan *et al.* [15], modifies the MOEA/D for many-objective optimization problems, and MOEA/D-AWA by Qi *et al.* [16] adjusts the weight vector to be effective in complex Pareto fronts. The idea of cross-entropy has also been applied in MOEA/D [17].

In addition, Zhang *et al.* proposed a competitive mechanism based multiobjective particle swarm optimizer (CMOPSO) [18], which updates the swarm of solutions based on particle pairwise competitions in each generation. Each particle learns from the winner of the competitions. Based on a fuzzy consistent matrix to guide global particle search, the evolution weight and learning factor are adjusted adaptively in the optimization process in adaptive multiobjective particle swarm optimization (AMOPSO) [19]. Based on the strength Pareto evolutionary algorithm (SPEA) [20], SPEA/R was proposed by Jiang and Yang [21], who designed a reference direction-based density estimator. In addition, the rotation grid-based evolutionary algorithm (RGridEA) [22] uses rotary grids to split the objective space to enhance the distribution performance, and thus, it is suitable for many-objective problems. Pareto dominance has been reconsidered in many-objective evolutionary algorithms, such as θ-dominance [23], generalized Pareto optimality (GPO) [24], and the k-optimal algorithm [25].

C. Multiparty Multiobjective Negotiations

The objective interests of multiple DMs are frequently in conflict. Multiparty negotiation is a process that helps multiple parties reach an agreement after communication among the parties or arbitration by a third party. Multiparty negotiation is frequently used in resource allocation, for solving conflicts of interest between buyers and sellers in the market, etc.

The arbitration of a third party is required to find the optimal (or acceptable) solutions of some negotiation problems. In the study in [26], reservoir flood control problems were investigated, where two parties, i.e., the Electrical Administrative Bureau of Northeastern and the Committee of Songliao Basin, which have different goals of concerning the reservoir's water levels, are requested to provide their solutions. The third party (the arbitrator) makes the decision by choosing the solution which is closest to the ideal state. Similarly, to find a compromise solution for multiple parties, third-party arbitration has been applied in the transfer of the water across an inter-basin region [27] and in groundwater management [28].

In some negotiation problems, each party has an objective and the objectives of multiple parties are combined to form an MOP. For example, in the study on the problem of resource allocation presented in [29], each party has a different preference concerning multiple resources and desires that the resources allocated to it be close to the ideal state. The objective of each party is constituted of the weighted sum of the differences between the resources allocated in the actual and the ideal state. All parties' goals are combined as a final objective, and a genetic algorithm (GA) with a new operation named *trade* is adopted to find the optimal resource allocation states. Rubenstein-Montano and Malaga [30] also linearly combined these goals, alternately evolving resource allocation and weight to generate the optimal allocation. In the study in [31], a GA is used to generate negotiated solutions that meet the goals of multiparties, which are greater than a predefined threshold.

Besides third-party arbitration, automatic learning is a method applied in negotiations where a party needs to interact with another party and the opponent's preference message is not clear. Negotiation is frequently applied in cases where buyers and sellers may have opposite interests and each party considers the price, negotiation period, and other information as negotiated issues (attributes). None of the parties knows the opponents preference. In the study in [32], each party, which has its own population, takes its own maximum payoff as its objective, and this goal comprises a linear combination of the above issues. Before each negotiation round, a GA is used to find an optimal solution to be the proposal that interacts with the opponent party. In the process of the GA, so that the proposal will be more easily accepted by the opponent, the fitness of the solution is related to the previous proposal of the opponent, which includes the preference message of the opponent. Hence, the GA can learn the preference of the other party. The best solution is selected as the offer and is sent to the opponent in the negotiation round. As long as the offer's payoff calculated by the opponent is greater than or equal to the offer which the opponent proposed previously, the offer will be accepted, and the negotiation process ends. Similar studies have been reported in [33], [34]. In the negotiation process, each party can generate optimal strategies for interaction, as well as optimal proposals. Its own maximum payoff is still the goal of each party in the negotiation, but the decision variables consist mainly of the negotiation strategy rather than other attributes. Gwak and Sim [35] proposed a co-evolution mechanism that adopts estimation of distribution algorithms (EDAs) with dynamic diversity control and local neighborhood search to generate optimal negotiation strategies. Each co-evolved population represents the individuals preferred by one party. During the negotiation process, the individuals in the two populations are randomly paired one-to-one to find whether each individual is accepted by the opponent. The fitness of an individual is based on the strategy and on whether its conditions are accepted by the opponent. In the study in [36], [37], a co-evolution mechanism that adopts a GA to generate optimal negotiation strategies was investigated. The learning of bargaining strategies for each party by a GA was also reported in [38]. In [39], Sim and An considered unity, success rate, and negotiation speed as the objectives; however, they are merged into a single objective.

In addition, in some studies, although the opponent's preference message is known, each party may regard its own payoff and the opponent's payoff as multiple objectives. Typically, for the methods in [40], [41], each party uses MOEAs to generate the PS, where its payoff and the opponent's payoff are considered to be two objectives. Based on the principle that each party should maximize its own benefit, only one solution will be selected from the PS to negotiate with the opponent. Moreover, complex negotiations are often needed to find the final solution accepted by all parties.

Similarly to multiparty multiobjective negotiations, MPMO handles problems involving multiple DMs. However, in contrast to multiparty multiobjective negotiations, MPMO considers all the information of the objectives of all parties to be available (or public), and thus, it does not need to consider the complex negotiation process, as well as the negotiation strategies, and its objective is to present the common PS of all parties. In this study, we assumed that the intersection of the PSs of each party is not empty. Again, it should be noted that the obtained solutions are Pareto equivalent for all parties.

III. Benchmark

To the best of our knowledge, no public benchmark of MPMOPs with common PSs exists. In this study, the benchmark MPMOPs were derived from the test functions of the CEC'2018 Competition on Dynamic Multiobjective Optimization [10].

In recent years, evolutionary DMO has attracted considerable research attention [42], [43]. DMO problems (DMOPs) are a type of optimization problem where the objective functions and/or constraints change with time, and then, the PSs and PFs of the problems may also change dynamically [44]. The objectives of a DMOP in each environment can be regarded as the MOP of one party. Therefore, an MPMOP can comprise a group of environments from a DMOP, corresponding to a group of MOPs, and each MOP could be regarded as the objectives of one party, respectively. Thus, the objective of the corresponding MPMOP is to find the common PS of the group of MOPs.

The benchmark is given in appendix. See Tables A-1 and A-2 in appendix for details.

IV. Proposed Algorithm

To solve MPMOPs, we propose an evolutionary algorithm, OptMPNDS, which is based on multiparty non-dominated sorting (MPNDS). The pseudo-code of the optimizer is given in Algorithm 1, which is described as follows.

First, the population P_0 is initialized with size N. In addition, t is set as 0 and Q_t is initialized as $\emptyset$. At each generation t, the individuals in Q_t and P_t are merged and stored in R_t. Second, for each DM i, non-dominated sorting is selected to sort the individuals in R_t; the sorting results are stored in $\mathcal{L}^i$. Here, the function *non-dominated-sorting*(...) in Step 6 is the same as the non-dominated sorting method proposed in [45]. Third, to find the solutions that are non-dominated for multiple parties, MPNDS is used to obtain the new sorting results. Fourth, individuals are ranked in order according to the crowding distance in each front. Fifth, the individuals in $\mathcal{F}_1$ to $\mathcal{F}_{z-1}$ are stored in the next generation, P_{t+1}. Evidently, the number of individuals in P_{t+1} is no greater than N. The remaining $N-|P_{t+1}|$ individuals with higher rankings are selected from $\mathcal{F}_z$. Then, t is incremented by 1, and the offspring Q_t of P_t is created by both recombination and mutation operators. The above steps are executed until the termination condition is satisfied. Finally, in the last generation's population P_t, the individuals that are multiparty Pareto optimal are selected and saved in CPS. Here, CPS means the set of the common Pareto optimal solutions.

The main steps of MPNDS are shown in Algorithm 2 and explained as follows.

1) Each individual may be located on different Pareto levels for different DMs. Each individual's maximum level in all DMs is denoted by j, and the individual is stored in $Max\mathcal{L}_j$. For example, individual x lies in the 1st level for the first party and the 2nd level for two other parties. Then, $x \in Max\mathcal{L}_2$.
2) To obtain the multiparty Pareto optimal individuals in the population, the individuals in $\mathcal{L}_j^1 \cap ... \cap \mathcal{L}_j^M$ are stored in $\mathcal{F}_z$. These individuals show a good and balanced performance for each party, and therefore, in general, they are better than others in any $\mathcal{L}_k^i$ ($k > j$, $i \in \{1, \ldots, M\}$).
3) If $\mathcal{F}_z$ is empty, the individuals of $\mathcal{L}_j^1 \cup ... \cup \mathcal{L}_j^M$ are incorporated in S. Each party's previous individuals that do not intersect with other parties in the same layer have more advantages than the following layer's individuals, and therefore, they are stored in S. Then, the value of j is incremented by 1 and the current front $\mathcal{F}_z$ is recalculated as the intersection of S and $Max\mathcal{L}_j$. The individuals in $\mathcal{F}_z$ are then deleted from S.
4) If $\mathcal{F}_z$ is not empty, z is incremented by 1 and the individuals in $\mathcal{F}_z$ are deleted from $\mathcal{L}_j^i$ ($i \in \{1, \ldots, M\}$).
5) Steps 2–4 are repeated until the termination condition is satisfied.

The crowding distance is used to represent the density of individuals. The method of calculating crowding distance is provided in [5]. It is noteworthy that the crowding distance is calculated in the objective space of all parties. Finally, it is noted that the sorting results based on MPNDS and the crowding distance are also used in the tournament selection.

V. Experiments

A. Compared Algorithms

OptMPNDS was compared with one baseline algorithm, which adopts an MOEA to optimize all the objectives from all parties. For convenience, we call this baseline algorithm OptAll.

The pseudo-code of OptAll is shown in Algorithm 3. In this algorithm, first, the objective functions from F_1 to F_M are combined and taken as evolutionary objectives. Then, the population is initialized and an MOEA is selected to optimize the combined objective functions, which is equivalent to solving a general MOP with many objectives. In this study, NSGA-II [5] was adopted. After it has consumed the maximum allowed number of function evaluations, the PS is obtained by the MOEA, and stored in CPS. Finally, the

Algorithm 1 OptMPNDS

Input: N, $F(x) = (F_1(x), ..., F_M(x))$
Output: CPS

Initialize population P_0 with size N;
$t = 0$, $Q_t = \emptyset$;
while the terminated condition is not satisfied **do**
    $R_t = P_t \cup Q_t$;
    for $i \in \{1, ..., M\}$ **do**
        $\mathcal{L}^i$ = *Non-Dominated-Sorting*(R_t, F_i);
    end for
    $\mathcal{F}$ = *MPNDS*$(N, \mathcal{L}^1, ..., \mathcal{L}^M)$
    Sort $\mathcal{F}$ by crowding distance for each level descent;
    $P_{t+1} = \emptyset$, $z = 1$;
    repeat
        $P_{t+1} = P_{t+1} \bigcup \mathcal{F}_z$;
        $z = z + 1$;
    until $|P_{t+1} \bigcup \mathcal{F}_z| > N$
    $P_{t+1} = P_{t+1} \bigcup \mathcal{F}_z(1 : N - |P_{t+1}|)$;
    $t = t + 1$;
    Create offspring Q_t of P_t;
end while
CPS = multiparty Pareto optimal individuals in P_t;

Algorithm 2 Multiparty non-dominated sorting (MPNDS)

Input: $N, \mathcal{L}^1, ..., \mathcal{L}^M$
Output: $\mathcal{F}$

$j = 1$, $z = 1$, $\mathcal{F}_z = \emptyset$, $S = \emptyset$, $Max\mathcal{L} = \emptyset$;
for $i \in \{1, ..., N\}$ **do**
    Find max level j from $(\mathcal{L}^1, ..., \mathcal{L}^M)$ for individual x_i;
    Store x_i in $Max\mathcal{L}_j$;
end for
while $\sum_{k=1}^{k=z} |\mathcal{F}_k| < N$ **do**
    $\mathcal{F}_z = \mathcal{L}_j^1 \bigcap ... \bigcap \mathcal{L}_j^M$;
    if $\mathcal{F}_z = \emptyset$ **then**
        $S = S \bigcup \mathcal{L}_j^1 \bigcup ... \bigcup \mathcal{L}_j^M$;
        $j = j + 1$;
        $\mathcal{F}_z = S \bigcap Max\mathcal{L}_j$;
        $S = S - \mathcal{F}_z$;
    end if
    if $\mathcal{F}_z \neq \emptyset$ **then**
        for $i \in \{1, ..., M\}$ **do**
            $\mathcal{L}_j^i = \mathcal{L}_j^i - \mathcal{F}_z$;
        end for
        $z = z + 1$, $\mathcal{F}_z = \emptyset$;
    end if
end while

Algorithm 3 OptAll

Input: N, $F = (F_1, ..., F_M)$
Output: CPS

Combine objective functions $F_{ALL} = (F_1; ...; F_M)$;
Initialize population P_0 with size N;
CPS = *MOEA*(P_0, F_{ALL});
for $i \in \{1, ..., M\}$ **do**
    Filter dominated individuals in CPS by F_i;
end for

dominated solutions are removed from CPS according to each party's objectives. The individuals remaining in CPS constitute the final output.

B. Experimental Settings

Details of the MPMOP test problems are shown in Tables A-1 and A-2. The number of the decision variables of all the problems was set to 10, 30, and 50.

In OptMPNDS and OptAll, simulated binary crossover (SBX) and polynomial mutation were used [46]. According to [5], the distribution indexes of SBX and the polynomial mutation were set as 20. The crossover probability was set as 1.0 and the mutation rate as $1/d$, where d is the dimension of decision variables. Code implementations of SBX and polynomial mutation refer to "PlatEMO" [47].

The population size for both algorithms was set to 100. The maximum allowed number of function evaluations was set to $1000*d*M$ for all problems, where d represents the dimension of decision variables and M represents the number of parties. All the algorithms were run 30 times independently for each test problem.

C. Performance Metrics

We used the inverted generational distance (IGD) [48] and generational distance (GD) [49] to evaluate the algorithms. These two indicators consider the convergence, uniformity, and spread performance of the solutions to evaluate the MOPs.

IGD is defined as

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}, \tag{2}$$

where P^* represents the actual PF and P is the PF obtained by the algorithms. $d(v, P)$ represents the minimum distance between v from P^* and points from P.

GD is defined as

$$GD(P^*, P) = \frac{\sqrt{\sum_{v \in P} d^2(v, P^*)}}{|P|}, \tag{3}$$

where the $d(v, P^*)$ represents the minimum distance between v from P and points from P^*.

Because there are M DMs, all DMs' objectives should be considered when $d(v, S)$ is calculated. $d(v, S)$ is defined as

$$d(v, S) = \min_{s \in S}(\sum_{i=1}^{i=M} \sqrt{(v_{i1} - s_{i1})^2 + \cdots + (v_{im} - s_{im})^2}), \tag{4}$$

where S represents P^* for IGD and P for GD, respectively, $(v_{i1}, \ldots, v_{im})$ means the m objectives of the i-th DM for solution v, and $(s_{i1}, \ldots, s_{im})$ means the same for solution s.

Both indicators represent the distance between the actual optimal solutions and the solutions obtained by the algorithms, and therefore, the smaller their value, the better the performance of the algorithm.

Besides IGD and GD, the number of solutions in the final common PS obtained by the algorithms is also adopted for the comparisons. This index is denoted by SN. The larger the SN value, the better the performance of the algorithm.

D. Experimental Results

The results of MPMOP1 to MPMOP11 are shown in Tables I, II, and III. The dimension of the decision space is set as 10, 30, and 50, respectively, in the three tables. For each table, the mean and standard deviation values of IGD, GD, and SN are reported. The best results of each problem solved by two algorithms are shown in bold font. The number of best results on all problems, named nbr, is shown at the bottom of each table. The sign "—" means that an algorithm does not obtain any common Pareto solution in one or more independent runs, and no value is recorded for the IGD and GD measures.

As we can see in the three tables, OptMPNDS performs better by modifying the non-dominated sorting. In Table I, it can be seen that, according to the metrics IGD and GD, the performance of OptMPNDS is the better on all problems. In Table II, it is clear that the performance of OptMPNDS is the better for 8 problems and that of OptAll is the better for 3 problems in terms of IGD, whereas the performance of OptMPNDS is the better for 10 problems in terms of GD. In Table III, it can be seen that OptMPNDS outperforms OptAll in 9 problems in terms of IGD and 10 problems in terms of GD.

In terms of SN, compared with OptAll, OptMPNDS always generates more solutions in the final common PS for all problems.

In summary, OptMPNDS finds more multiparty common optimal solutions on MPMOPs and produces more satisfactory results.

VI. Conclusion and Future Work

In this paper, MPMOPs with common PSs are addressed. We propose an evolutionary algorithm named OptMPNDS to solve MPMOPs, which is based on MPNDS, an algorithm modified from the traditional Pareto dominance. Our experimental results show that OptMPNDS achieves a good convergence effect on MPMOPs.

In the future, we may consider applying our method in dynamic MOPs. Considering some near future environments, which are frequently predictable, we could determine the common PS for the current environment and several upcoming environments by using the proposed method. Meanwhile, in this paper, we assumed the common PS is not empty. We will consider the case where the common PS is empty in the future. Finally, we will consider the MPMOPs with preference in the future, where the preference could be expressed by $\widehat{g}$-*donminance* [50] or others.

Appendix

The benchmark is described in Tables A-1 and A-2, which is based on the test functions of the CEC'2018 Competition on Dynamic Multiobjective Optimization [10].

In Table A-1, F_1 and F_2, both of which have two or three conflicting objectives, are two MOPs of two DMs, respectively. In Table A-2, F_1, F_2, and F_3, which are extended from Table A-1, are three MOPs, i.e., three parties. All the objective functions should be minimized.

The PSs for each party in MPMOPs are also listed in the corresponding tables. The common PS of the MPMOP is the intersection of the PSs of all involved MOPs. In this study, we approximately calculated the common PS. First, we took a series of discrete points for each party's PS. Second, when the distances between the points from different parties were less than 10^{-4}, we selected these points in one party and stored them as the common PS.

Acknowledgment

The authors would like to thank Jiang *et al.* for providing the benchmark of DMOPs [10] and Tian *et al.* [47] for providing "PlatEMO." The source code of this work is available at https://github.com/lwjie595/OptMPNDS.

References

[1] X. Ge, L. Zhang, and C. Wang, "Mixed integer model for multi-objective short-term cascaded hydropower optimal scheduling," *Power System Protection and Control*, vol. 41, no. 4, pp. 55–60, 2013.

[2] G. Fu, "A fuzzy optimization method for multicriteria decision making: An application to reservoir flood control operation," *Expert Systems with Applications*, vol. 34, no. 1, pp. 145–149, 2008.

[3] A. Zhou, B. Qu, H. Li, S. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[4] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2016.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[6] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK-Report*, vol. 103, pp. 182–197, 2001.

[7] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1. IEEE, 1999, pp. 98–105.

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

[9] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[10] S. Jiang, S. Yang, X. Yao, K. C. Tan, M. Kaiser, and N. Krasnogor, "Benchmark Functions for the CEC'2018 Competition on Dynamic Multiobjective Optimization," Newcastle University, Tech. Rep., 2018.

[11] H. Geng, M. Zhang, L. Huang, and X. Wang, "Infeasible elitists and stochastic ranking selection in constrained evolutionary multi-objective optimization," in *Proceedings of Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 2006, pp. 336–344.

[12] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer Science & Business Media, 1999, vol. 12.

TABLE I: Mean and standard deviation of inverted generational distance (IGD), generational distance (GD), and the number of solutions in the final common Pareto optimal set (SN) for $d = 10$

Problems	IGD		GD		SN	
	OptMPNDS	OptAll	OptMPNDS	OptAll	OptMPNDS	OptAll
MPMOP1	**2.7894E-05±1.8206E-05**	—	**3.8657E-04±1.0962E-04**	—	**92.53±11.14**	0.03±0.18
MPMOP2	**2.7364E-04±1.0637E-03**	6.5179E-02±7.5853E-02	**1.9701E-03±3.6896E-04**	2.7435E-02±9.6293E-03	**89.27±7.14**	6.30±1.37
MPMOP3	**2.5420E-02±9.2706E-03**	3.6974E-02±1.1352E-02	**3.5816E-03±1.6171E-03**	8.4763E-03±3.6664E-03	**100.00±0.00**	30.20±1.52
MPMOP4	**5.7572E-02±7.0402E-03**	4.7859E-01±8.3774E-02	**1.9118E-02±2.4296E-02**	2.3447E-01±6.4889E-02	**100.00±0.00**	19.70±4.25
MPMOP5	**6.2492E-02±1.1398E-02**	1.5786E-01±3.3413E-02	**8.1808E-02±1.2760E-02**	1.3622E-01±2.4067E-02	**100.00±0.00**	29.90±3.74
MPMOP6	**1.9709E-02±2.4359E-03**	7.6749E-01±2.0057E-01	**1.4156E-02±4.4596E-02**	1.2537E+00±5.4241E-01	**100.00±0.00**	6.60±1.87
MPMOP7	**4.5667E-06±4.1010E-06**	—	**3.9392E-04±1.4739E-04**	—	**99.53±1.25**	0.00±0.00
MPMOP8	**1.2262E-02±4.9124E-02**	7.2167E-01±2.6127E-01	**2.7334E-03±7.2968E-04**	1.8833E-02±3.2047E-02	**85.67±8.04**	2.60±1.16
MPMOP9	**8.2055E-02±1.1601E-02**	6.1508E-01±9.2426E-02	**3.6453E-02±5.3268E-02**	2.9442E-01±8.7200E-02	**100.00±0.00**	21.80±4.12
MPMOP10	**5.7619E-02±6.3695E-03**	3.2451E-01±8.2224E-02	**2.5512E-02±1.9876E-03**	1.3481E-01±3.3844E-02	**100.00±0.00**	10.00±2.03
MPMOP11	**1.8343E-02±9.0481E-04**	1.5923E+00±5.8241E-01	**9.9348E-04±1.8189E-04**	3.1056E+00±1.7653E+00	**100.00±0.00**	4.97±2.33
nbr	**11**	0	**11**	0	**11**	0

TABLE II: Mean and standard deviation of inverted generational distance (IGD), generational distance (GD), and the number of solutions in the final common Pareto optimal set (SN) for $d = 30$

Problems	IGD		GD		SN	
	OptMPNDS	OptAll	OptMPNDS	OptAll	OptMPNDS	OptAll
MPMOP1	**1.6166E-05±5.9224E-06**	8.2886E-03±2.8462E-03	**1.4895E-04±3.5328E-05**	8.5851E-03±2.8688E-03	**99.47±1.50**	2.00±0.00
MPMOP2	**2.4442E-03±1.3105E-02**	5.6115E-02±5.8392E-02	**5.4909E-04±9.1160E-05**	2.3879E-02±2.2528E-02	**94.00±5.68**	5.47±1.14
MPMOP3	2.4422E-01±1.5082E-01	**2.1359E-01±6.1893E-02**	**2.3656E-02±1.1274E-02**	4.1957E-02±2.0581E-02	**94.87±8.76**	25.67±5.45
MPMOP4	**5.3974E-02±9.7020E-03**	1.4187E+00±7.8597E-01	**2.4825E-02±4.8948E-02**	9.0690E-01±6.5261E-01	**100.00±0.00**	14.87±9.87
MPMOP5	**4.1739E-02±4.2176E-03**	3.8524E-01±8.7829E-02	**1.2526E-02±8.9261E-04**	2.2637E-01±1.6706E-01	**100.00±0.00**	5.73±1.72
MPMOP6	**1.5277E-02±9.1020E-04**	2.1889E+00±1.5142E+00	**1.2830E-03±1.8572E-04**	4.8311E+00±2.7628E+00	**100.00±0.00**	3.30±1.09
MPMOP7	**5.1282E-06±2.9116E-06**	—	**7.0512E-05±4.4645E-05**	—	**99.80±0.76**	1.10±0.66
MPMOP8	1.7650E-01±1.4797E-01	**1.4400E-01±1.1676E-01**	**4.4631E-04±2.2156E-04**	4.5138E-02±4.3458E-02	**98.47±3.25**	5.20±1.10
MPMOP9	**7.8395E-02±9.6008E-03**	2.1323E+00±9.0104E-01	**8.0474E-03±5.6064E-03**	1.5716E+00±4.2154E-01	**100.00±0.00**	12.83±3.87
MPMOP10	4.3365E+00±2.7402E+00	**6.4002E-01±2.2718E-01**	1.1043E+00±7.8363E-01	**1.6547E-01±1.7244E-01**	**37.30±33.27**	2.83±0.79
MPMOP11	**1.7898E-02±7.7425E-04**	3.0805E+00±2.6421E+00	**7.1128E-04±2.2673E-04**	9.2080E+00±7.7316E+00	**100.00±0.00**	2.63±1.22
nbr	**8**	3	**10**	1	**11**	0

TABLE III: Mean and standard deviation of inverted generational distance (IGD), generational distance (GD), and the number of solutions in the final common Pareto optimal set (SN) for $d = 50$

Problems	IGD		GD		SN	
	OptMPNDS	OptAll	OptMPNDS	OptAll	OptMPNDS	OptAll
MPMOP1	**1.8747E-05±6.0160E-06**	6.3026E-03±1.1210E-03	**1.0978E-04±1.7849E-05**	5.3650E-03±1.0718E-03	**99.80±0.66**	2.00±0.00
MPMOP2	**2.7957E-02±4.8711E-02**	7.6744E-02±7.2050E-02	**3.4810E-04±6.2528E-05**	2.1345E-02±1.9497E-02	**96.87±4.34**	5.17±1.02
MPMOP3	5.1823E-01±2.0427E-01	**4.2675E-01±1.0779E-01**	**5.8021E-02±2.8762E-02**	6.4503E-02±2.6333E-02	**87.70±14.24**	23.53±5.20
MPMOP4	**5.4883E-02±9.0317E-03**	2.0794E+00±1.8092E+00	**1.1611E-02±3.0415E-02**	1.1314E+00±1.0174E+00	**100.00±0.00**	19.33±16.69
MPMOP5	**3.2240E-02±2.9652E-03**	4.9932E-01±1.5816E-01	**6.8471E-03±4.7592E-04**	3.5955E-01±5.4693E-01	**100.00±0.00**	3.47±1.01
MPMOP6	**1.4232E-02±6.2776E-04**	2.6863E+00±2.5453E+00	**1.0587E-03±2.0233E-04**	5.5870E+00±4.5832E+00	**100.00±0.00**	3.43±1.79
MPMOP7	**7.3476E-06±2.4889E-06**	—	**4.6983E-05±2.1547E-05**	—	**99.33±1.63**	1.33±0.61
MPMOP8	**2.4295E-01±1.3750E-01**	2.8748E-01±1.6227E-01	**1.8611E-04±1.3263E-04**	5.3030E-02±5.7344E-02	**98.47±3.52**	4.27±0.87
MPMOP9	**7.3962E-02±1.0786E-02**	3.9340E+00±2.5023E+00	**9.8420E-03±1.2792E-02**	2.0262E+00±1.1542E+00	**100.00±0.00**	15.23±6.18
MPMOP10	1.1150E+01±1.2323E+00	**8.2081E-01±3.0717E-01**	2.5768E+00±5.9650E-01	**1.7001E-01±2.0396E-01**	**21.27±7.54**	2.13±0.63
MPMOP11	**1.7885E-02±9.2098E-04**	5.0912E+00±6.7435E+00	**6.7693E-04±1.8141E-04**	1.4051E+01±1.3402E+01	**100.00±0.00**	3.17±1.32
nbr	**9**	2	**10**	1	**11**	0

[13] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2007, pp. 781–788.

[14] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[15] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D+ uniform design: A new version of MOEA/D for optimization problems with many objectives," *Computers & Operations Research*, vol. 40, no. 6, pp. 1648–1660, 2013.

[16] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.

[17] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Information Sciences*, vol. 282, pp. 363–387, 2014.

[18] X. Zhang, X. Zheng, R. Cheng, J. Qiu, and Y. Jin, "A competitive mechanism based multi-objective particle swarm optimizer with fast convergence," *Information Sciences*, vol. 427, pp. 63–76, 2018.

[19] Z. Geng, Z. Wang, Q. Zhu, and Y. Han, "Multi-objective operation optimization of ethylene cracking furnace based on AMOPSO algorithm," *Chemical Engineering Science*, vol. 153, pp. 21–33, 2016.

[20] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[21] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 329–346, 2017.

[22] J. Zou, L. Fu, J. Zheng, S. Yang, G. Yu, and Y. Hu, "A many-objective evolutionary algorithm based on rotated grid," *Applied Soft Computing*, vol. 67, pp. 596–609, 2018.

TABLE A-1: Test problems with two decision makers

Problems	Objective functions	Variable bounds	Pareto optimal set of each subproblem
MPMOP1	$F_1 = (g_{11}(x,1), g_{12}(x,1)),\ F_2 = (g_{11}(x,2), g_{12}(x,2))$ $g_{11}(x,t) = g_3(x)(\frac{1+t}{x_1}),\ g_{12}(x,t) = g_3(x)(\frac{x_1}{1+t})$ $g_3(x) = 1 + \sum_{i=2}^{n}\Big(x_i - \frac{1}{1+e^{\alpha_t(x_1-2.5)}}\Big)^2,\ \alpha_t = 5\cos(0.5\pi t)$	$x_1 \in [1,4]$ $x_i \in [0,1]$ $i = 2,\ldots,d$	$1 \le x_1 \le 4$ $x_i = \frac{1}{1+e^{\alpha_t(x_1-2.5)}}$ $i = 2,\ldots,d$
MPMOP2	$F_1 = (g_{21}(x,0), g_{22}(x,0)),\ F_2 = (g_{21}(x,3), g_{22}(x,3))$ $g_{21}(x,t) = g_3(x)(x_1 + 0.1\sin(3\pi x_1))$ $g_{22}(x,t) = g_3(x)(1 - x_1 + 0.1\sin(3\pi x_1))^{\alpha_t}$ $g_3(x) = 1 + \sum_{i=2}^{n}\Big(x_i - \frac{G(t)\sin(4\pi x_1^{\beta_t})}{1+\lvert G(t)\rvert}\Big)^2$ $\alpha_t = 2.25 + 2\cos(2\pi t), \beta_t = 1, G(t) = \sin(0.5\pi t)$	$x_1 \in [0,1]$ $x_i \in [-1,1]$ $i = 2,\ldots,d$	$0 \le x_1 \le 1$ $x_i = \frac{G(t)\sin(4\pi x_1^{\beta_t})}{1+\lvert G(t)\rvert}$ $i = 2,\ldots,d$
MPMOP3	$F_1 = (g_{31}(x,0), g_{32}(x,0)),\ F_2 = (g_{31}(x,\pi/2), g_{32}(x,\pi/2))$ $g_{31}(x,t) = g_3(x)(x_1 + k(t)),\ g_{32}(x,t) = g_3(x)(1 - x_1 + k(t))$ $g_3(x) = 1 + \sum_{i=2}^{n}(x_i - \cos(4t + x_1 + x_{i-1}))^2$ $k(t) = \max\{0, (\frac{1}{2N_t} + 0.1)\sin(2N_t\pi x_1)\},\ N_t = 1 + \lfloor 10 * \lvert\sin(0.5\pi t)\rvert\rfloor$	$x_1 \in [0,1]$ $x_i \in [-1,1]$ $i = 2,\ldots,d$	$x_1 \in \bigcup_{i=1}^{N_t}[\frac{2i-1}{2N_t}, \frac{i}{N_t}]\bigcup\{0\}$ $x_i = \cos(4t + x_1 + x_{i-1})$ $i = 2,\ldots,d$
MPMOP4	$F_1 = (g_{41}(x,0), g_{42}(x,0), g_{43}(x,0)),\ F_2 = (g_{41}(x,1), g_{42}(x,1), g_{43}(x,1))$ $g_{41}(x,t) = g_4(x)[\sin(0.5\pi x_1)]^{H(t)}$ $g_{42}(x,t) = g_4(x)[\sin(0.5\pi x_2)\cos(0.5\pi x_1)]^{H(t)}$ $g_{43}(x,t) = g_4(x)[\cos(0.5\pi x_2)\cos(0.5\pi x_1)]^{H(t)}$ $g_4(x) = 1 + \sum_{i=3}^{n}(x_i - \frac{\sin(2\pi(x_1+x_2))}{1+\lvert G(t)\rvert})^2$ $H(t) = 2.25 + 2\cos(0.5\pi t), G(t) = \sin(0.5\pi t)$	$x_{i=1,2} \in [0,1]$ $x_i \in [-1,1]$ $i = 3,\ldots,d$	$0 \le x_{i=1,2} \le 1$ $x_i = \frac{\sin(2\pi(x_1+x_2))}{1+\lvert G(t)\rvert}$ $i = 3,\ldots,d$
MPMOP5	$F_1 = (g_{51}(x,0), g_{52}(x,0), g_{53}(x,0))$ $F_2 = (g_{51}(x,1.5), g_{52}(x,1.5), g_{53}(x,1.5))$ $g_{51}(x,t) = g_4(x)\sin(y_1),\ g_{52}(x,t) = g_4(x)\sin(y_2)\cos(y_1)$ $g_{53}(x,t) = g_4(x)\cos(y_2)\cos(y_1), y_{i=1,2} = \frac{\pi}{6}G_t + (\frac{\pi}{2} - \frac{\pi}{3}G_t)x_i$ $g_4(x) = 1 + \sum_{i=3}^{n}(x_i - 0.5G(t)x_1)^2, G(t) = \lvert\sin(0.5\pi t)\rvert$	$x_i \in [0,1]$ $i = 1,\ldots,d$	$0 \le x_{i=1,2} \le 1$ $x_i = 0.5G(t)x_1$ $i = 3,\ldots,d$
MPMOP6	$F_1 = (g_{61}(x,0), g_{62}(x,0), g_{63}(x,0)),\ F_2 = (g_{61}(x,1), g_{62}(x,1), g_{63}(x,1))$ $g_{61}(x,t) = g_4(x)\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $g_{62}(x,t) = g_4(x)\cos(0.5\pi x_1)\sin(0.5\pi x_2), g_{63}(x,t) = g_4(x)\sin(0.5\pi x_1)$ $g_4(x) = 1 + \sum_{i=3}^{n}(x_i - \sin(tx_1))^2 + \Big\lvert\prod_{j=1}^{2}\sin(\lfloor k_t(2x_j - r)\rfloor * \pi/2)\Big\rvert$ $k_t = \lfloor 10sin(\pi t)\rfloor, r = 1 - mod(k_t, 2)$	$x_{i=1,2} \in [0,1]$ $x_i \in [-1,1]$ $i = 3,\ldots,d$	$\{(x1, x2) \in [0,1]^2\lvert$ $\prod_{j=1}^{2} mod(\lvert\lfloor k_t(2x_j - r)\rfloor\rvert, 2) = 0\}$ $x_i = \sin(tx_1)$ $i = 3,\ldots,d$

[23] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2015.

[24] C. Zhu, L. Xu, and E. D. Goodman, "Generalization of Pareto-optimality for many-objective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 299–315, 2015.

[25] M. Farina and P. Amato, "A fuzzy definition of "optimality" for many-criteria optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 34, no. 3, pp. 315–326, 2004.

[26] C. Chuntian and K. Chau, "Three-person multi-objective conflict decision in reservoir flood control," *European Journal of Operational Research*, vol. 142, no. 3, pp. 625–631, 2002.

[27] C. Zhang, G. Wang, Y. Peng, G. Tang, and G. Liang, "A negotiation-based multi-objective, multi-party decision-making model for inter-basin water transfer scheme optimization," *Water Resources Management*, vol. 26, no. 14, pp. 4029–4038, 2012.

[28] G. Wang, L. Huang, and G. Liang, "Application of a multi-person and multi-objective decision-making model in groundwater resources management," *Journal of Hydrologic Engineering*, vol. 17, no. 3, pp. 389–393, 2011.

[29] B. Rubenstein-Montano and R. Malaga, "A weighted sum genetic algorithm to support multiple-party multiple-objective negotiations," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 366–377, 2002.

[30] ——, "A co-evolutionary approach to strategy design for decision makers in complex negotiation situations," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. IEEE, 2000, pp. 9–17.

TABLE A-2: Test problems with three decision makers

Problems	Objective functions	Variable bounds	Pareto optimal set of each subproblem
MPMOP7	$F_1 = (g_{11}(x,0), g_{12}(x,0))$ $F_2 = (g_{11}(x,1), g_{12}(x,1))$ $F_3 = (g_{11}(x,2), g_{12}(x,2))$	$x_1 \in [1,4]$ $x_i \in [0,1]$ $i = 2,\ldots,d$	$1 \le x_1 \le 4$ $x_i = \dfrac{1}{1+e^{\alpha_t(x_1-2.5)}},\ i = 2,\ldots,d$
MPMOP8	$F_1 = (g_{21}(x,0), g_{22}(x,0))$ $F_2 = (g_{21}(x,1), g_{22}(x,1))$ $F_3 = (g_{21}(x,3), g_{22}(x,3))$	$x_1 \in [0,1]$ $x_i \in [-1,1]$ $i = 2,\ldots,d$	$0 \le x_1 \le 1$ $x_i = \dfrac{G(t)\sin(4\pi x_1^{\beta_t})}{1+\lvert G(t)\rvert},\ i = 2,\ldots,d$
MPMOP9	$F_1 = (g_{41}(x,0), g_{42}(x,0), g_{43}(x,0))$ $F_2 = (g_{41}(x,0.5), g_{42}(x,0.5), g_{43}(x,0.5))$ $F_3 = (g_{41}(x,1), g_{42}(x,1), g_{43}(x,1))$	$x_{i=1,2} \in [0,1]$ $x_i \in [-1,1]$ $i = 3,\ldots,d$	$0 \le x_{i=1,2} \le 1$ $x_i = \dfrac{\sin(2\pi(x_1+x_2))}{1+\lvert G(t)\rvert},\ i = 3,\ldots,d$
MPMOP10	$F_1 = (g_{51}(x,0), g_{52}(x,0), g_{53}(x,0))$ $F_2 = (g_{51}(x,1), g_{52}(x,1), g_{53}(x,1))$ $F_3 = (g_{51}(x,1.5), g_{52}(x,1.5), g_{53}(x,1.5))$	$x_i \in [0,1]$ $i = 1,\ldots,d$	$0 \le x_{i=1,2} \le 1$ $x_i = 0.5G(t)x_1,\ i = 3,\ldots,d$
MPMOP11	$F_1 = (g_{61}(x,0), g_{62}(x,0), g_{63}(x,0))$ $F_2 = (g_{61}(x,1), g_{62}(x,1), g_{63}(x,1))$ $F_3 = (g_{61}(x,1.5), g_{62}(x,1.5), g_{63}(x,1.5))$	$x_{i=1,2} \in [0,1]$ $x_i \in [-1,1]$ $i = 3,\ldots,d$	$\{(x1,x2) \in [0,1]^2 \mid$ $\prod_{j=1}^{2} mod(\lvert\lfloor k_t(2x_j - r)\rfloor\rvert, 2) = 0\}$ $x_i = \sin(tx_1),\ i = 3,\ldots,d$

[31] D. Zhai, Y. Wu, J. Lu, and F. Yan, "A fuzzy negotiation model with genetic algorithms," in *Integration and Innovation Orient to E-Society Volume 1*. Springer, 2007, pp. 35–43.

[32] R. Y. Lau, M. Tang, O. Wong, S. W. Milliner, and Y.-P. P. Chen, "An evolutionary learning approach for adaptive negotiation agents," *International Journal of Intelligent Systems*, vol. 21, no. 1, pp. 41–72, 2006.

[33] R. Y. Lau, M. Tang, and O. Wong, "Towards genetically optimised responsive negotiation agents," in *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. IEEE, 2004, pp. 295–301.

[34] R. Y. Lau, "Towards genetically optimised multi-agent multi-issue negotiations," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. IEEE, 2005, pp. 35–45.

[35] J. Gwak and K. M. Sim, "An augmented EDA with dynamic diversity control and local neighborhood search for coevolution of optimal negotiation strategies," *Applied Intelligence*, vol. 38, no. 4, pp. 600–619, 2013.

[36] J. Gwak, K. M. Sim, and M. Jeon, "Novel dynamic diversity controlling EAs for coevolving optimal negotiation strategies," *Information Sciences*, vol. 273, pp. 1–32, 2014.

[37] J. Gwak and K. M. Sim, "Coevolving near-optimal strategies for negotiation with incomplete information using a diversity controlling GA," in *Proceedings of 2010 IEEE International Conference on Automation and Logistics*. IEEE, 2010, pp. 624–631.

[38] J. R. Oliver, "A machine-learning approach to automated negotiation and prospects for electronic commerce," *Journal of Management Information Systems*, vol. 13, no. 3, pp. 83–112, 1996.

[39] K. M. Sim and B. An, "Evolving best-response strategies for market-driven agents using aggregative fitness GA," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 3, pp. 284–298, 2009.

[40] J. Zheng, C. X. Ling, Z. Shi, and Y. Xie, "Some discussions about MOGAs: individual relations, non-dominated set, and application on automatic negotiation," in *Proceedings of Congress on Evolutionary Computation*, vol. 1. IEEE, 2004, pp. 706–712.

[41] R. Beheshti and A. Rahmani, "A multi-objective genetic algorithm method to support multi-agent negotiations," in *Proceedings of 2009 Second International Conference on Future Information Technology and Management Engineering*. IEEE, 2009, pp. 596–599.

[42] J. Eaton, S. Yang, and M. Gongora, "Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2980–2992, 2017.

[43] Z. Yang, Y. Jin, and K. Hao, "A bio-inspired self-learning coevolutionary dynamic multiobjective optimization algorithm for internet of things services," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 675–688, 2018.

[44] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.

[45] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2014.

[46] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*. ACM, 2007, pp. 1187–1194.

[47] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[48] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "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*, vol. 264, 2008.

[49] D. A. Van Veldhuizen and G. B. Lamont, "On measuring multiobjective evolutionary algorithm performance," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1. IEEE, 2000, pp. 204–211.

[50] W. Luo, L. Shi, X. Lin, and C. A. C. Coello, "The $\widehat{g}$-dominance relation for preference-based evolutionary multi-objective optimization," in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation*. IEEE, 2019, pp. 2418–2425.