IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 1, 2020, accepted August 22, 2020, date of publication September 7, 2020, date of current version September 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022164

Decomposition-Based Multi-Objective Evolutionary Algorithm Design Under Two Algorithm Frameworks

LIE MENG PANG, (Member, IEEE), HISAO ISHIBUCHI, (Fellow, IEEE), AND KE SHANG, (Member, IEEE)

Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China

Corresponding author: Hisao Ishibuchi (hisao@sustech.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61876075, in part by the Guangdong Provincial Key Laboratory under Grant 2020B121201001, in part by the Program for Guangdong Introducing Innovative and Enterpreneurial Teams under Grant 2017ZT07X386, in part by the Shenzhen Science and Technology Program under Grant KQTD2016112514355531, and in part by the Program for University Key Laboratory of Guangdong Province under Grant 2017KSYS008.

ABSTRACT The development of efficient and effective evolutionary multi-objective optimization (EMO) algorithms has been an active research topic in the evolutionary computation community. Over the years, many EMO algorithms have been proposed. The existing EMO algorithms are mainly developed based on the final population framework. In the final population framework, the final population of an EMO algorithm is presented to the decision maker. Thus, it is required that the final population produced by an EMO algorithm is a good solution set. Recently, the use of solution selection framework was suggested for the design of EMO algorithms. This framework has an unbounded external archive to store all the examined solutions. A pre-specified number of solutions are selected from the archive as the final solutions presented to the decision maker. When the solution selection framework is used, EMO algorithms can be designed in a more flexible manner since the final population is not necessarily to be a good solution set. In this paper, we examine the design of MOEA/D under these two frameworks. We use an offline genetic algorithm-based hyper-heuristic method to find the optimal configuration of MOEA/D in each framework. The DTLZ and WFG test suites and their minus versions are used in our experiments. The experimental results suggest the possibility that a more flexible, robust and high-performance MOEA/D algorithm can be obtained when the solution selection framework is used.

INDEX TERMS Evolutionary multi-objective optimization, MOEA/D, final population framework, solution selection framework, hyper-heuristics.

I. INTRODUCTION

Multi-objective optimization problems are commonly found in many real-world applications [1]–[4]. The main goal of solving a multi-objective optimization problem is to optimize (either maximize or minimize) several objective functions simultaneously. However, the objective functions of a multi-objective optimization problem are conflicting in nature. Thus, it is not possible to obtain a single optimal solution that optimizes all the objective functions simultaneously. Usually a set of optimal solutions with different tradeoffs is obtained. The set of tradeoff optimal solutions is known as the Pareto optimal solution set. The Pareto optimal solutions form the Pareto front in the objective space.

Owing to the advantage of being able to search for a set of non-dominated solutions in a single run, evolutionary multi-objective optimization (EMO) algorithms have been a popular approach for solving multi-objective optimization problems. Over the years, various EMO algorithms such as SMS-EMOA, NSGA-III, HyPE, MOEA/D, and MOEA/D-HH have been proposed [5]. Since it is often impractical/difficult to include user preference a priori, most EMO algorithms in the literature are of the posteriori type [6]. That is, the main aim of these EMO algorithms is to find a set of well-distributed non-dominated solutions to approximate the Pareto front. Then, a decision

The associate editor coordinating the review of this manuscript and approving it for publication was Pavlos I. Lazaridis.

maker chooses a solution from the obtained solution set.

Based on generation update mechanisms, EMO algorithms can be classified into two categories. One is non-elitist algorithms (e.g., MOGA [7] and NSGA [8]) and the other is elitist algorithms (e.g., SPEA [9] and NSGA-II [10]). In non-elitist algorithms, the current population is entirely replaced with the offspring population. No solution in the current population can survive to the next generation. As a result, it is very difficult to design an efficient EMO algorithm based on the non-elitist framework.

In elitist algorithms, good solutions (called elite individuals) in the current population survive to the next generation. The most frequently-used mechanism for generation update is the $(\mu + \lambda)$-selection strategy [10]. In this mechanism, λ solutions are generated from the current population with μ solutions. Then, the best μ solutions are selected from the $(\mu + \lambda)$ solutions for the next generation. Some elitist EMO algorithms have an archive of a pre-specified size to store non-dominated solutions [11]–[13]. Different archiving methods such as the Pareto dominance-based archiving methods, decomposition-based archiving methods, and indicator-based archiving methods can be used in the archiving process [15]. The archive is updated at every generation. In these EMO algorithms, the current population or the archive at the final generation is presented to the decision maker.

Even though the elitist framework is much more efficient than the non-elitist framework, it still has a difficulty. As discussed in [14], good solutions can be deleted during the generation update phase. Since the population size (and the archive size) is pre-specified, some solutions must be discarded during the generation update phase. New solutions cannot be compared with those discarded solutions in previous generations. As a result, solutions in the final population (and the final archive) are not always non-dominated among all the examined solutions [15].

In both the non-elitist and elitist frameworks, solutions in the final generation are presented to the decision maker. Thus, EMO algorithms should be designed to have a set of well-distributed non-dominated solutions in the final generation. In this paper, both the non-elitist and elitist frameworks are referred to as the *final population framework* (since most of recent algorithms are based on the $(\mu + \lambda)$ selection mechanism).

Recently, it was proposed to use an unbounded external archive in the design of EMO algorithms [14]. The idea is to present a solution set selected from all the examined solutions to the decision maker. In this paper, this algorithm framework is referred to as the *solution selection framework*. Whereas this framework was used for performance evaluation of existing EMO algorithms in some studies [16], [17], it has not been used for the design of new EMO algorithms. Unlike the final population framework, the final population in the solution selection framework does not have to be a good solution set. The solution selection framework can use an EMO algorithm with a bounded internal archive together with an unbounded external archive (whereas we do not discuss such an EMO algorithm in this paper).

This paper aims to clearly demonstrate the flexibility of the solution selection framework in the design of EMO algorithms. We use Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [18] as a sample to empirically show the advantages of the solution selection framework. MOEA/D is a well-known and frequently-used EMO algorithm especially for many-objective optimization. A number of approaches have been proposed to further improve the performance of MOEA/D (e.g., with respect to the choice of a scalarizing function and the specification of weight vectors). In our former study [19], we demonstrated that the performance of MOEA/D is sensitive to the specification of the reference point and more robust performance is obtained by the solution selection framework.

Motivated by the promising experimental results in our former study about the reference point specification [19], we further investigate other components and parameters in MOEA/D in this paper. Our experiments are conducted using the two algorithm frameworks, as follows:

1. **Final population framework**. The output of the MOEA/D algorithm is the solutions in the final population.
2. **Solution selection framework**. The output of the MOEA/D algorithm is a solution set selected from all examined solutions in an unbounded external archive.

The examined MOEA/D components include a scalarizing function, a crossover operator, and a mutation operator. The examined parameters include the neighborhood size, the normalization parameter, the initial reference point and the final reference point in the dynamic reference point mechanism, the crossover rate, and the mutation rate. Since the parameter space is large and complex, we use an offline genetic algorithm-based hyper-heuristic method to search for the optimal configuration of MOEA/D under each algorithm framework for each test problem. Then, the obtained optimal algorithm configurations are compared and analyzed. As test problems, we use the three-objective DTLZ1-4 [20], WFG1-9 [21], and their minus versions [22]. We search for the best MOEA/D design for each of these 26 test problems under each algorithm framework.

The main contributions of this paper are listed as follows:

- A GA-based hyper-heuristic method is used to search for the optimal configurations of MOEA/D under two different algorithm design frameworks, namely the final population framework and the solution selection framework.
- A number of important components and parameters of MOEA/D are identified and form the algorithm design space for MOEA/D.
- Both the obtained MOEA/D configurations by the GA-based hyper-heuristic under the final population

framework and the solution selection framework are evaluated with two different scenarios.
- The experimental results demonstrate the usefulness of the solution selection framework in designing a flexible and high-performance MOEA/D algorithm.

This paper is organized as follows. First, we briefly explain multi-objective optimization and MOEA/D in Section II. Next, we explain our genetic algorithm-based hyper-heuristic method and the experiment settings in Section III. Then, we compare the experimental results under the two algorithm frameworks in Section IV. Finally, we conclude this paper in Section V.

II. BACKGROUND

A. MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Let us assume that we have the following M-objective minimization problem:

$$\text{Minimize } \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})) \quad \text{subject to } \boldsymbol{x} \in \mathbf{X}, \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_D)$ is a D-dimensional solution vector, $\mathbf{X}$ is the feasible region, and $f_i(\boldsymbol{x})$ is the i-th objective to be minimized ($i = 1, 2, \ldots, M$).

Definition 1 (Pareto Dominance Relation): A solution vector $\boldsymbol{x^A}$ is dominated by $\boldsymbol{x^B}$ if and only if $f_i(\boldsymbol{x^B}) \leq f_i(\boldsymbol{x^A})$ for all $i \in \{1, 2, \ldots, M\}$ and $f_j(\boldsymbol{x^B}) < f_j(\boldsymbol{x^A})$ for at least one index j.

Definition 2 (Pareto Optimal Solution): A solution $\boldsymbol{x^*}$ is a Pareto optimal solution if and only if it is not dominated by any other solution in $\mathbf{X}$.

The Pareto set (PS) consists of all Pareto optimal solutions. The projection of the Pareto set to the objective space is the Pareto front (PF).

B. MOEA/D

The basic idea of MOEA/D [18] is to decompose a multi-objective optimization problem into N single-objective subproblems using a set of predefined weight vectors $\boldsymbol{W} = \{\boldsymbol{w^1}, \boldsymbol{w^2}, \ldots, \boldsymbol{w^N}\}$ and a scalarizing function. Then, the N subproblems are evolved in a cooperative manner by exploiting the information from other subproblems during the search process.

A set of uniformly distributed weight vectors $\boldsymbol{W}$ is used in most MOEA/D implementations in the literature. Each weight vector $\boldsymbol{w^j} = \left(w_1^j, w_2^j, \ldots, w_M^j\right)$ must fulfil the following relation:

$$w_1^j + w_2^j + \cdots + w_M^j = 1, \tag{2}$$

where $w_i^j \geq 0$ ($i = 1, 2, \ldots, M$) and $j \in \{1, 2, \ldots, N\}$. In our study, the Das and Dennis method [23] is used to systematically generate the weight vectors. Since each subproblem j has a single individual $\boldsymbol{x^j}$ ($j \in \{1, 2, \ldots, N\}$), the population size equals to the number of subproblems (the number of weight vectors) in MOEA/D.

C. COMPONENTS AND PARAMETERS IN MOEA/D

MOEA/D includes a number of components and parameters which need to be specified. They are explained in this subsection.

1) SCALARIZING FUNCTIONS

It is known that a scalarizing function plays an essential role in MOEA/D. The scalarizing function is used to calculate the fitness value of each individual. In this paper, we consider five scalarizing functions: the weighted sum $\left(g^{\mathrm{WS}}\right)$, Tchebycheff $\left(g^{\mathrm{TCH}}\right)$, modified Tchebycheff (g^{MTCH}), penalty-based boundary intersection (PBI) $\left(g^{\mathrm{PBI}}\right)$, and inverted penalty-based boundary intersection (IPBI) (g^{IPBI}) functions. The five scalarizing functions are given as follows:

Weighted Sum (WS) [18]:

$$\text{Minimize } g^{\mathrm{WS}}(\boldsymbol{x}|\boldsymbol{w}) = w_1 f_1(\boldsymbol{x}) + \ldots + w_M f_M(\boldsymbol{x}), \tag{3}$$

Tchebycheff (TCH) [18]:

$$\text{Minimize } g^{\mathrm{TCH}}\left(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z}^*\right) = \max_{i=1,2,\ldots,M} \{w_i \cdot |z_i^* - f_i(\boldsymbol{x})|\}, \tag{4}$$

where $\boldsymbol{z^*} = \left(z_1^*, z_2^*, \ldots, z_M^*\right)$ is the reference point (which is served as the origin of the weight vectors). In our study, we consider a dynamic reference point mechanism proposed in [24]. In principle, the ideal point is used as the reference point. However, the ideal point is unknown for real-world problems. Thus, the common practice is to specify each element z_i^* of $\boldsymbol{z^*}$ by the minimum value of each objective $f_i(\boldsymbol{x})$ over all solutions examined so far, as shown in (5) and (6).

$$z_i^* = z_i^{min} - \epsilon_i, \quad \epsilon_i \geq 0, \tag{5}$$

$$z_i^{min} = \min\{f_i(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{S}\}, \ i = 1, 2, \ldots, M, \tag{6}$$

where ϵ_i is a non-negative number, and $\mathbb{S}$ consists of all examined solutions. The value of ϵ_i is set to zero in the original MOEA/D.

A dynamic reference point mechanism [24] has shown to be useful for improving the performance of MOEA/D. As discussed in [24], in the early stage of evolution, the estimated reference point is usually inaccurate since the population is not close to the Pareto front. The accuracy of the estimated reference point is gradually improved through the evolution. Based on these discussions, the following linearly decreasing formulation was proposed in [24]:

$$\epsilon_i = \left(\epsilon_i^{ini} - \epsilon_i^{end}\right)\left(\frac{T - t}{T - 1}\right) + \epsilon_i^{end}, \tag{7}$$

where T is the maximum generation number, t is the current generation index ($t = 1, 2, \ldots, T$), and ϵ_i^{ini} and ϵ_i^{end} are the initial and final settings of ϵ_i, respectively.

Modified Tchebycheff (MTCH) [25]:

$$\text{Minimize } g^{\mathrm{MTCH}}\left(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z}^*\right) = \max_{i=1,2,\ldots,M} \{\left|z_i^* - f_i(\boldsymbol{x})\right| / w_i\}, \tag{8}$$

where z^* is the reference point. In this paper, we use the formulation in (5)-(7) as in the Tchebycheff function. If $w_i = 0$, w_i is set to 10^{-6} to avoid division by zero.

Penalty-based Boundary Intersection (PBI) [18]:

$$\text{Minimize } g^{\text{PBI}}\left(\boldsymbol{x}|\boldsymbol{w}, z^*\right) = d_1 + \theta d_2, \tag{9}$$

where the penalty parameter θ is a user-definable non-negative real number. The commonly-used value for θ is 5. The two distances d_1 and d_2 are defined as

$$d_1 = \left|\left(\boldsymbol{f}\left(\boldsymbol{x}\right) - z^*\right)^T \boldsymbol{w}\right| / \left\|\boldsymbol{w}\right\|, \tag{10}$$

$$d_2 = \left\|\boldsymbol{f}\left(\boldsymbol{x}\right) - z^* - d_1\left(\boldsymbol{w}/\left\|\boldsymbol{w}\right\|\right)\right\|, \tag{11}$$

where z^* is the reference point specified by (5)-(7).

Inverted PBI (IPBI) function [26]:

$$\text{Maximize } g^{\text{IPBI}}\left(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z^N}\right) = d_1 - \theta d_2, \tag{12}$$

where θ is the penalty parameter. The two distances d_1 and d_2 are defined as

$$d_1 = \left|\left(\boldsymbol{z^N} - \boldsymbol{f}\left(\boldsymbol{x}\right)\right)^T \boldsymbol{w}\right| / \left\|\boldsymbol{w}\right\|, \tag{13}$$

$$d_2 = \left\|\boldsymbol{z^N} - \boldsymbol{f}\left(\boldsymbol{x}\right) - d_1\left(\boldsymbol{w}/\left\|\boldsymbol{w}\right\|\right)\right\|, \tag{14}$$

where $\boldsymbol{z^N} = \left(z_1^N, z_2^N, \ldots, z_M^N\right)$ is the estimated nadir point. Each element z_i^N of $\boldsymbol{z^N}$ is specified by the maximum value of each objective $f_i\left(\boldsymbol{x}\right)$ in the current population.

2) NORMALIZATION MECHANISM

In this paper, we consider a simple normalization mechanism [27] in MOEA/D to deal with problems with differently scaled objectives. The estimated ideal point $\boldsymbol{z^*}$(with $\epsilon_i = 0$ in (5)) and the estimated nadir point $\boldsymbol{z^N}$ are used to normalize the objective value z_i as follows:

$$z_i := \frac{z_i - z_i^*}{z_i^N - z_i^* + \varepsilon}, \tag{15}$$

where ε (which is referred to as the normalization parameter in this paper) is a positive real number used to prevent the denominator from becoming zero in the case of $z_i^N = z_i^*$.

3) NEIGHBORHOOD STRUCTURES

Neighborhood structure is an important feature of MOEA/D. Each subproblem has its own neighborhood which is defined by the Euclidean distance between weight vectors. For each subproblem, two parents are randomly selected from its neighborhood to generate a new solution. Then, the newly generated solution is compared with all solutions in its neighborhood (including the current solution of the current subproblem). All inferior neighbors are replaced with the newly generated solution. In the original MOEA/D, the same neighborhood is used for both mating and replacement.

In [28], the use of two different neighborhood structures for mating and replacement was investigated. For many-objective knapsack problems, it was shown that the search ability of MOEA/D can be improved by using a larger neighborhood structure for replacement than mating. In this paper, we also examine the use of two neighborhood structures.

4) GENETIC OPERATORS

In the literature, many EMO algorithms use the SBX crossover [29] and the polynomial mutation [30] for multi-objective continuous optimization problems. In addition to these commonly-used genetic operators, we also examine some other operators: the whole arithmetic crossover (WAX) [31], the local arithmetic crossover (LAX) [32], the Gaussian mutation [30], and the random mutation [6].

III. HYPER-HEURISTICS USING A GENETIC ALGORITHM

A. ALGORITHM DESIGN SPACE FOR MOEA/D

As explained in the previous section, MOEA/D has a number of components and parameters to be specified. In this paper, we search for an optimal configuration of MOEA/D under each algorithm framework (i.e., final population and solution selection) for each test problem. More specifically, we try to find the best combination of the components and parameters in Table 1. The domain of each component/parameter is shown in the column labeled "Domain" in Table 1. For example, one of {WS, TCH, PBI, IPBI, MTCH} is selected as a scalarizing function. For the penalty parameter θ in PBI and IPBI, a real number is specified in the closed interval [0, 10]. In this paper, the MOEA/D algorithm design means the choice of a possible value in the domain for each component/parameter in Table 1.

In our experiments, the population size for MOEA/D is fixed to 91, and the distribution index for the SBX crossover and the polynomial mutation is fixed to their default value 20.

B. GENETIC ALGORITHM-BASED HYPER-HEURISTIC

In this paper, we use a genetic algorithm-based hyper-heuristic method to find the optimal MOEA/D algorithm design for the final population framework and the solution selection framework.

We use the following parameter specifications in the hyper-heuristic method in our computational experiments:

Coding: 53-bit binary string,
Population size μ: 100,
Initial population: Random binary strings,
Termination condition: 100 generations,
Crossover: Uniform crossover with probability 1,
Mutation: Bit-flip mutation with probability 1/53,
Selection: Tournament selection with tournament size 3,
Generation update: $(\mu + \mu)$-selection strategy,
Fitness evaluation: Average hypervolume.

In our hyper-heuristic method, each component/parameter is represented by a binary substring. For example, the scalarizing function g is represented by a 3-bit substring

TABLE 1. Parameter space for MOEA/D.

Component/Parameter	Domain	Coding
g (Scalarizing function)	{WS, TCH, PBI, IPBI, MTCH}	3-bit
θ (only in PBI and IPBI)	[0,10]	10-bit
ϵ_i^{ini} (Reference point)	{0, 0.001, 0.005, 0.01, 0.05, 0.1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}	4-bit
ϵ_i^{end} (Reference point)	{-5, -4, -3, -2, -1, -0.1, -0.05, -0.01, -0.005, -0.001, 0, 0.001, 0.005, 0.01, 0.05, 0.1}	4-bit
T_{Mate} (Mating neighbors)	{5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%}	3-bit
T_{Rep} (Replacement neighbors)	{5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%}	3-bit
ε (Normalization)	{0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 5, 10, 15, 20, 25}	10-bit
Crossover	{SBX, WAX, LAX}	3-bit
P_c (Crossover rate)	[0,1]	5-bit
Mutation	{Polynomial, Gaussian, Random}	3-bit
P_m (Mutation rate)	[0,1]	5-bit

$S_g = s_1 s_2 s_3$. It is decoded as

$$\text{Decode}\left(S_g\right) = 1 + \text{round}\left(\frac{4}{7}\sum_{i=1}^{3} s_i 2^{3-i}\right), \qquad (16)$$

where round $(\cdot)$ means round to the nearest integer. The WS, TCH, PBI, IPBI and MTCH functions are represented by the decoded values 1, 2, 3, 4, and 5, respectively. The penalty parameter θ for the PBI and IPBI functions is represented by a 10-bit substring $S_\theta = s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10}$. The 10-bit substring S_θ is decoded as

$$\text{Decode}\left(S_\theta\right) = \frac{10}{1023}\sum_{i=1}^{10} s_i 2^{10-i}, \qquad (17)$$

where the 1024 real numbers in the interval [0, 10] are examined as the penalty parameter θ. Then, an MOEA/D algorithm is represented by a binary string of length 53, where the coding for each component/parameter is listed in the third column of Table 1.

The hypervolume (HV) indicator is used to evaluate each individual (i.e., the binary string of an MOEA/D configuration). The larger HV value indicates the better performance. In order to handle the stochastic nature of the MOEA/D algorithm, each MOEA/D configuration is applied to a test problem for five times. The average HV value over the five runs is used as the fitness value of each individual in the hyper-heuristic method.

In the final population framework, the HV value of the final population of each run of the MOEA/D configuration is calculated. In the solution selection framework, 91 solutions (which is the same as the population size of MOEA/D) are selected from non-dominated solutions among all the examined solutions in each run of the MOEA/D configuration. We use the distance-based solution subset selection method [33].

In the distance-based selection method, first, one of the extreme non-dominated solutions is randomly selected among the M extreme non-dominated solutions as the first solution. The second solution is the non-dominated solution with the largest distance from the first solution. Then, the non-dominated solution with the largest distance from the first and second solutions is selected as the third solution. The selection process is repeated until 91 solutions are selected. We use this method since it is computationally efficient (e.g., it is fast). Of course, we can use various HV-based solution subset methods [34]–[36]. Whereas they can find better solution subsets with higher HV values, they need much more computation time than the distance-based method. In our hyper-heuristic method, 10,000 MOEA/D configurations (100 individuals $\times$ 100 generations) are applied to the given test problem five times. The solution subset selection is performed 50,000 times in a single run of the hyper-heuristic method. Thus, we use the fast distance-based solution subset selection method in the hyper-heuristic method in this paper. The termination condition for each MOEA/D configuration is 10,000 solution evaluations.

HV calculation needs a reference point. Since it is assumed that we have no knowledge about the true Pareto front of the test problem during the algorithm design by the hyper-heuristic method, we combined all solution sets obtained by the five runs of all MOEA/D configurations in the current population of the hyper-heuristic method. Then, we select only the non-dominated solutions among them to form an approximated Pareto front. The ideal point and the nadir point are calculated using the approximated Pareto front. The objective space is normalized so that the calculated ideal and nadir points become $(0, 0, \ldots, 0)$ and $(1, 1, \ldots, 1)$. Then, the reference point $\boldsymbol{r}$ for HV calculation is specified as $\boldsymbol{r} = (r, r, \ldots, r)$ with $r = 1.1$ (i.e., a slightly worse point than the calculated nadir point in the normalized objective space).

IV. EXPERIMENTAL RESULTS

A. OBTAINED MOEA/D CONFIGURATIONS

Our computational experiments are performed for each of the three-objective DTLZ1-4 [20], WFG1-9 [21], and their minus versions [22] under each algorithm framework. The obtained MOEA/D configurations for the 26 test problems under the final population and solution selection frameworks are shown in Table 2 and Table 3, respectively.

In Table 2 and Table 3, we can see that the optimal MOEA/D configurations are totally different between the final population framework and the solution selection framework. As an example, when the final population framework is used, only the PBI and TCH functions are chosen as the scalarizing function for the DTLZ and WFG test suites (see Table 2). However, when the solution selection framework is used, the WS function is selected for DTLZ2, DTLZ3, WFG2, WFG3, and WFG6 (see Table 3).

TABLE 2. The MOEA/D configurations obtained by the Genetic algorithm-based hyper-heuristic method for each three-objective test problem under the final population (FP) framework.

Test problems ($M=3$)	g	θ (PBI/IPBI)	Reference Point ϵ_i^{ini}	Reference Point ϵ_i^{end}	ε (normalization)	Neighborhood T_{Mate}	Neighborhood T_{Rep}	*Crossover*	P_c	*Mutation*	P_m
DTLZ1	PBI	1.2219	0.005	0.005	15	30%	40%	SBX	0.9677	Polynomial	0.1290
DTLZ2	PBI	1.3099	0.001	0.005	1	15%	35%	SBX	1	Polynomial	0.0323
DTLZ3	PBI	0.9971	6	0.001	20	40%	40%	LAX	0.4194	Polynomial	0.0968
DTLZ4	PBI	1.5738	0.05	0.001	0.1	35%	40%	SBX	1	Random	0.0323
WFG1	TCH	-	0.05	0.1	0.1	25%	35%	SBX	1	Polynomial	0.0645
WFG2	PBI	3.4604	3	0.1	0.1	20%	40%	SBX	0.9677	Polynomial	0.1290
WFG3	TCH	-	0.1	0.05	5	5%	40%	-	0	Polynomial	0.2903
WFG4	PBI	1.3196	4	-0.01	0.0001	35%	40%	SBX	1	Polynomial	0.0323
WFG5	PBI	1.1437	3	0.005	0.001	35%	40%	-	0	Polynomial	0.0645
WFG6	PBI	2.5611	1	0.005	10	40%	40%	LAX	0.7419	Polynomial	0.0323
WFG7	PBI	2.0332	5	0.001	0.01	25%	30%	LAX	0.0968	Polynomial	0.0323
WFG8	PBI	1.3294	1	0.05	0.000001	15%	40%	SBX	0.9677	Polynomial	0.0323
WFG9	TCH	-	0.1	0.05	0.01	10%	40%	LAX	0.6129	Polynomial	0.2581
Minus-DTLZ1	IPBI	1.2219	0.005	-0.01	0.1	25%	35%	SBX	0.8710	Random	0.0323
Minus-DTLZ2	WS	-	-	-	0.00001	35%	40%	SBX	1	Polynomial	0.0645
Minus-DTLZ3	WS	-	-	-	25	35%	40%	SBX	1	Random	0.0323
Minus-DTLZ4	PBI	0.1369	1	0	0.001	10%	10%	LAX	0.3548	Gaussian	0.5161
Minus-WFG1	IPBI	2.5318	0.1	0.05	0.00001	10%	35%	SBX	0.8065	Polynomial	0.2581
Minus-WFG2	PBI	1.7204	0	0.01	0.1	25%	40%	SBX	1	Polynomial	0.0323
Minus-WFG3	IPBI	1.1046	0.05	-0.05	0.0001	20%	35%	SBX	1	Polynomial	0.0323
Minus-WFG4	WS	-	-	-	0.001	35%	40%	SBX	1	Polynomial	0.0645
Minus-WFG5	PBI	0.1369	0	-0.01	0.0001	40%	40%	SBX	1	Polynomial	0.0645
Minus-WFG6	WS	-	-	-	0.001	40%	40%	SBX	1	Polynomial	0.0323
Minus-WFG7	WS	-	-	-	0.001	35%	40%	SBX	1	Polynomial	0.0323
Minus-WFG8	PBI	0.1564	9	0.005	0.001	15%	40%	SBX	1	Polynomial	0.2581
Minus-WFG9	WS	-	-	-	0.0001	40%	40%	SBX	1	Polynomial	0.0645

TABLE 3. The MOEA/D configurations obtained by the genetic algorithm-based hyper-heuristic method for each three-objective test problem under the solution selection (SS) framework.

Test problems ($M=3$)	g	θ (PBI/IPBI)	Reference Point ϵ_i^{ini}	Reference Point ϵ_i^{end}	ε (normalization)	Neighborhood T_{Mate}	Neighborhood T_{Rep}	*Crossover*	P_c	*Mutation*	P_m
DTLZ1	PBI	0.5083	0.001	0.005	15	40%	40%	SBX	0.8387	Polynomial	0.1613
DTLZ2	WS	-	-	-	20	5%	35%	SBX	0.4516	Random	0.0968
DTLZ3	WS	-	-	-	0.001	40%	40%	SBX	0.9032	Polynomial	0.0968
DTLZ4	PBI	2.2287	1	-0.001	0.0001	40%	40%	SBX	1	Random	0.0323
WFG1	TCH	-	0.01	0.05	15	30%	40%	SBX	1	Polynomial	0.0323
WFG2	WS	-	-	-	15	5%	35%	LAX	0.9032	Polynomial	0.1613
WFG3	WS	-	-	-	25	10%	35%	LAX	0.4194	Polynomial	0.1935
WFG4	TCH	-	3	-0.05	0.00001	40%	40%	SBX	1	Polynomial	0.0323
WFG5	PBI	9.6090	8	0.005	0.0001	40%	40%	SBX	1	Polynomial	0.0645
WFG6	WS	-	-	-	15	25%	30%	LAX	0.6774	Polynomial	0.0323
WFG7	PBI	1.7791	6	0.1	1	10%	40%	LAX	0.5161	Polynomial	0.0323
WFG8	TCH	-	0.1	-0.005	25	10%	35%	LAX	0.9355	Polynomial	0.0968
WFG9	PBI	0.8309	0	0.1	25	10%	5%	LAX	0.6129	Polynomial	0.0968
Minus-DTLZ1	IPBI	1.9453	1	-0.05	0.00001	30%	40%	SBX	1	Random	0.0323
Minus-DTLZ2	IPBI	6.2757	7	-0.001	15	40%	35%	LAX	0.2581	Gaussian	0.5484
Minus-DTLZ3	PBI	0.2151	6	-0.01	0.1	30%	35%	SBX	0.9355	Random	0.0323
Minus-DTLZ4	WS	-	-	-	10	40%	30%	LAX	0.6774	Gaussian	0.8387
Minus-WFG1	TCH	-	0.1	0.1	0.0001	30%	40%	SBX	1	Polynomial	0.0645
Minus-WFG2	MTCH	-	1	-0.1	0.01	30%	30%	SBX	1	Polynomial	0.0323
Minus-WFG3	PBI	0.8602	4	-1	0.1	35%	35%	SBX	0.9355	Polynomial	0.0323
Minus-WFG4	PBI	0.3128	5	0	0.0001	30%	30%	SBX	0.8710	Polynomial	0.0645
Minus-WFG5	WS	-	-	-	0.00001	40%	40%	SBX	1	Polynomial	0.0323
Minus-WFG6	PBI	0.2151	10	0.005	0.0001	20%	35%	SBX	0.9677	Polynomial	0.0323
Minus-WFG7	TCH	-	6	-1	0.0001	30%	40%	LAX	0.2581	Gaussian	0.9032
Minus-WFG8	WS	-	-	-	5	20%	35%	SBX	1	Gaussian	0.5806
Minus-WFG9	WS	-	-	-	0.01	40%	40%	SBX	1	Polynomial	0.0323

Another example is the specification of the neighbourhood size. In the final population framework, the mating neighborhood size (T_{Mate}) is always smaller than or equal to the replacement neighborhood size (T_{Rep}). However, this is not always the case in the solution selection framework. As an example, for WFG9, the mating neighborhood size

TABLE 4. The mean hypervolume value over 31 runs of each MOEA/D on three-objective test problems using 10,000 solution evaluations under the final population framework. The best result is highlighted by yellow colour and the worst result is highlighted by red font.

Test problems (M=3)	Mean Hypervolume (10,000 solution evaluations)						
	MOEA/D-WS	MOEA/D-TCH	MOEA/D-PBI	MOEA/D-IPBI	MOEA/D-MTCH	Auto-MOEA/D-FP	Auto-MOEA/D-SS
DTLZ1	0.1969 –	0.6183 –	0.4582 –	0.0389 –	0.6418 =	0.6615	**0.6712** =
DTLZ2	0.2487 –	0.5303 –	0.5552 –	0.2487 –	0.5578 =	**0.5585**	0.2485 –
DTLZ3	**0.0056** =	0.0000 =	0.0000 =	0.0000 =	0.0000 =	0.0000	0.0000 =
DTLZ4	0.1275 –	0.2750 –	0.3295 –	0.1775 –	0.3618 –	**0.5573**	0.5519 –
WFG1	0.4069 –	0.4503 –	0.3473 –	0.4674 –	0.4768 –	0.5300	**0.5444** =
WFG2	0.7586 –	0.7819 –	0.7537 –	0.8012 –	0.7922 –	**0.8969**	0.8656 –
WFG3[1]	0.2389 –	0.4852 –	0.4346 –	0.4141 –	0.4861 –	**0.5146**	0.2265 –
WFG4	0.2488 –	0.4921 –	0.4573 –	0.4535 –	0.4948 –	0.5148	**0.5258** +
WFG5	0.2072 –	0.4637 –	0.4421 –	0.4200 –	0.4616 –	**0.4974**	0.4927 –
WFG6	0.2135 –	0.4692 –	0.4376 –	0.3426 –	0.4702 –	**0.4939**	0.2307 –
WFG7	0.2480 –	0.5027 –	0.3655 –	0.4194 –	0.4975 –	**0.5403**	0.5291 –
WFG8	0.1738 –	0.4416 –	0.4063 –	0.1863 –	0.4361 –	**0.4675**	0.4087 –
WFG9	0.1711 –	0.4484 –	0.3582 –	0.2999 –	0.4458 –	**0.4663**	0.0825 –
Minus-DTLZ1	0.0279 –	0.1953 –	0.1861 –	0.1309 –	0.1915 –	**0.2196**	0.2159 –
Minus-DTLZ2	0.5304 –	0.5161 –	0.5193 –	0.5304 –	0.5030 –	**0.5371**	0.3714 –
Minus-DTLZ3	0.5010 –	0.4772 –	0.4715 –	0.4951 –	0.4690 –	0.5252	**0.5343** +
Minus-DTLZ4	0.4710 –	0.5152 –	0.3849 –	0.4560 –	0.5030 –	**0.5374**	0.5290 –
Minus-WFG1	0.0190 –	0.0664 –	0.0498 –	0.0318 –	0.0902 –	**0.1083**	0.0991 –
Minus-WFG2	0.1548 –	0.2816 –	0.2614 –	0.2058 –	0.2786 –	**0.2824**	0.2634 –
Minus-WFG3	0.0240 –	0.1848 –	0.1886 –	0.0742 –	0.1818 –	**0.2160**	0.1880 –
Minus-WFG4	0.5113 –	0.4728 –	0.4906 –	0.5120 –	0.4648 –	**0.5281**	0.5251 –
Minus-WFG5	0.5108 –	0.4926 –	0.5029 –	0.5113 –	0.4745 –	0.5346	**0.5352** +
Minus-WFG6	0.5141–	0.4999 –	0.5078 –	0.5167 –	0.4770 –	0.5292	**0.5379** +
Minus-WFG7	0.5136 –	0.4624 –	0.4835 –	0.5146 –	0.4557 –	**0.5288**	0.0950 –
Minus-WFG8	0.5154 –	0.5028 –	0.5126 –	0.5179 –	0.4789 –	**0.5403**	0.5233 –
Minus-WFG9	0.5057 –	0.4780 –	0.4875 –	0.5071 –	0.4559 –	0.5217	**0.5220** =
+/−/=	0/25/1	0/25/1	0/25/1	0/25/1	0/23/3		4/18/4

(i.e., T_{Mate} = 10% of the population size) is larger than the replacement neighborhood size (i.e., T_{Rep} = 5% of the population size).

B. EVALUATION UNDER THE FINAL POPULATION FRAMEWORK

In this subsection, the performance of each MOEA/D configuration in Table 2 and Table 3 is evaluated under the final population framework. That is, the output of each algorithm is the solutions in the final population. For comparison purposes, we also perform the experiments for standard MOEA/D with the five scalarizing functions. For the standard MOEA/D, the following default parameters are used in the computational experiments:

Population size: 91,
Scalarizing functions: WS, TCH, MTCH, PBI, and IPBI,
Neighborhood size: 20,
Crossover: SBX with probability 1,
Distribution index for the SBX crossover: 20,
Mutation: Polynomial mutation with probability 1/D,
Distribution index for the polynomial mutation: 20.

As in [18], [22], and [26], the penalty parameter θ in the standard MOEA/D-PBI and MOEA/D-IPBI are set as 5 and 0.1, respectively. Our computational experiments are performed on the PlatEMO platform [37]. Each MOEA/D algorithm is independently run 31 times on each test problem.

In order to evaluate the performance of the MOEA/D, the HV indicator and the inverted generational distance (IGD) indicator are used. A larger HV value and a smaller IGD value indicate the algorithm has better performance. It should be noted that the HV indicator is also used to evaluate each MOEA/D configuration in the hyper-heuristic method (in Section III.B).

As we have explained in the previous section, a reference point is needed for calculating the HV value. In this section, the true Pareto front information of each test problem is used for evaluating the performance of MOEA/D. The true ideal and nadir points are used to normalize the objective space. The reference point $\boldsymbol{r}$ = (1.1, 1.1, 1.1) is used for the hypervolume calculation. For the IGD calculation, a reference point set is needed. In our experiments, about 10,000 reference points are uniformly sampled over the entire Pareto front of each test problem in PlatEMO [37] as the reference point set for the IGD calculation except for WFG3 (see the footnote of Table 4).

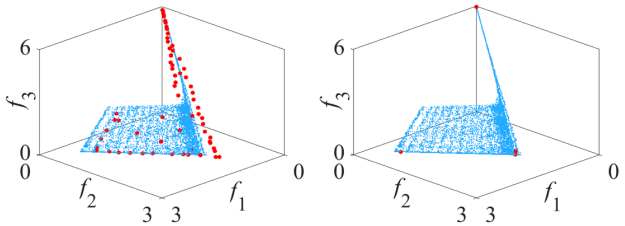
Table 4 shows the mean hypervolume value over 31 runs of each MOEA/D configuration for the three-objective test problems. The stopping condition for each MOEA/D

[1]Since the reference point set for WFG3 on the PlatEMO does not cover the flag region (the true Pareto front of the three-objective WFG3 has a flag region [38]), we generated the reference point set for the three-objective WFG3 problem by choosing non-dominated solutions from solution sets obtained by different EMO algorithms. We used NSGA-II, NSGA-III, MOEA/D-PBI, SMS-EMOA and SPEA2 with the population size 100 and 100,000 solution evaluations over 31 runs. A total of 7905 non-dominated solutions were obtained, which were used as the reference points for HV and IGD calculation of WFG3.

configuration is 10,000 solution evaluations. We use the terms "MOEA/D-WS", "MOEA/D-TCH", "MOEA/D-MTCH", "MOEA/D-PBI", and 'MOEA/D-IPBI" to denote the standard MOEA/D with the five scalarizing functions, respectively. The terms "Auto-MOEA/D-FP" and "Auto-MOEA/D-SS" are used to denote the obtained MOEA/D configurations in Table 2 (tuned under the final population framework) and Table 3 (tuned under the solution selection framework), respectively. Since Auto-MOEA/D-FP is tuned under the final population framework, it is expected that Auto-MOEA/D-FP has the best performance among all the algorithms. The Wilcoxon's rank sum test at a significant level of 5% is used to evaluate the statistical difference between Auto-MOEA/D-FP and each of the other six MOEA/D versions. The signs "+", "−", and "=" are used to indicate the compared MOEA/D version is statistically better than, worse than, or equivalent to Auto-MOEA/D-FP. The best and worst results among the seven MOEA/D versions are highlighted using yellow color and red color, respectively.

In Table 4, Auto-MOEA/D-FP shows high performance. Auto-MOEA/D-FP outperforms all the other six versions on almost all test problems. For some test problems, Auto-MOEA/D-FP is not the best in Table 4. However, the results obtained by Auto-MOEA/D-FP are very similar to the best results on those problems. From the experimental results, we can also see that the best results for almost all test problems are obtained by Auto-MOEA/D-FP and Auto-MOEA/D-SS. This observation shows that a tuning procedure is beneficial for improving the performance of MOEA/D.

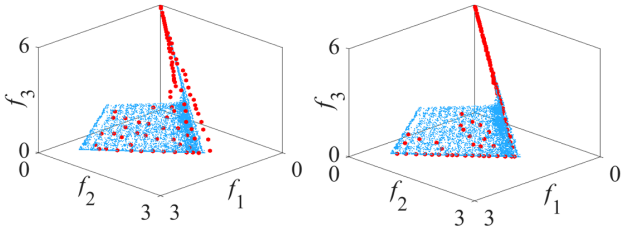
Even though Auto-MOEA/D-SS shows the best performance on some test problems, it also shows the worst performance on some other test problems. For example, it has the worst performance on WFG3. However, Auto-MOEA/D-SS has the best performance when it is evaluated under the solution selection framework (which will be shown in Table 6 in the next subsection). This observation suggests that the best algorithm configuration for the solution selection framework can be totally different from that for the final population framework.



(a) Auto-MOEA/D-FP (HV is 0.5149) (b) Auto-MOEA/D-SS (HV is 0.2183)

FIGURE 1. Solutions in the final population of a single run with the median HV value for WFG3 under the evaluation of the final population framework (with the stopping condition of 10,000 solution evaluations). The blue points are the Pareto front and the red points are the obtained solutions.

The solution sets obtained by Auto-MOEA/D-FP and Auto-MOEA/D-SS for WFG3 under the evaluation of the final population framework are shown in Fig. 1. That is, Fig. 1 shows the solutions in the final population of a single run of each algorithm. A single run (over 31 runs) with the median HV value (from Table 4) is selected. We can see that the final population of Auto-MOEA/D-SS is not a good solution set. That is, only a few solutions are obtained since many solutions are overlapping with each other. Fig. 2 shows the solution sets obtained by Auto-MOEA/D-FP and Auto-MOEA/D-SS for WFG3 under the evaluation of the solution selection framework. That is, Fig. 2 shows the selected solutions from all the examined solutions in the same single run as in Fig.1. We can see from the comparison between Fig. 1 and Fig. 2 that better solution sets can be obtained from the solution selection framework. We can also see that the obtained solution set by the solution selection framework in Fig. 2 (b) is the best with respect to the HV indicator among the four solution sets in Fig. 1 and Fig. 2, whereas the final population of the corresponding run in Fig. 1(b) is the worst.



(a) Auto-MOEA/D-FP (HV is 0.5275) (b) Auto-MOEA/D-SS (HV is 0.5499)

FIGURE 2. Selected solutions from all the examined solutions in a single run (corresponding to the same single run in Fig. 1) for WFG3 under the evaluation of the solution selection framework (with the stopping condition of 10,000 solution evaluations). The blue points are the Pareto front and the red points are the obtained solutions.

The performance of MOEA/D is also examined using a different performance indicator (i.e., IGD) and a different termination condition (i.e., 50,000 solution evaluations) under the final population framework. Table 5 shows the summary of statistical comparison results between Auto-MOEA/D-FP with the other six MOEA/D versions using the Wilcoxon rank sum test. The signs "+", "−", and "=" indicate the number of test instances on which the results of Auto-MOEA/D-FP are significantly better than, worse than, or equivalent to other MOEA/D versions.

We can see from Table 5 that Auto-MOEA/D-FP (which is designed for HV maximization under 10,000 solution evaluations) also shows high performance under different conditions. As an example, when Auto-MOEA/D-FP is evaluated using the HV indicator with 50,000 solution evaluations, its performance is significantly better than each of the other MOEA/D versions for at least 20 (out of 26) test problems. Auto-MOEA/D-FP also shows good performance when the IGD indicator is used for evaluation.

C. EVALUATION UNDER THE SOLUTION SELECTION FRAMEWORK

In this subsection, each MOEA/D configuration is independently run 31 times on each test problem under the

TABLE 5. Summary of statistical comparison results between Auto-MOEA/D-FP with the other MOEA/D versions under different conditions.

Auto-MOEA/D-FP vs.		MOEA/D-WS	MOEA/D-TCH	MOEA/D-PBI	MOEA/D-IPBI	MOEA/D-MTCH	Auto-MOEA/D-SS
Evaluation using HV with 50,000 solutions evaluations	+	23	25	21	23	22	20
	−	3	1	4	3	3	5
	=	0	0	1	0	1	1
Evaluation using IGD with 10,000 solutions evaluations	+	25	23	23	25	22	19
	−	1	3	3	1	2	4
	=	0	0	0	0	2	3
Evaluation using IGD with 50,000 solutions evaluations	+	22	22	16	24	19	20
	−	3	4	7	2	6	4
	=	1	0	3	0	1	2

solution selection framework. The same parameter settings as in Section IV.B are used in the experiments. Actually, the same 31 runs are used in Section IV.B and IV.C. That is, all the examined solutions are stored in an unbounded external archive in each run in Section IV.B where only the final population was used. In this subsection, the output of each MOEA/D algorithm is a set of selected solutions from the unbounded external archive.

Various solution subset selection methods can be used to select pre-specified numbers of solutions from the unbounded external archive [19]. We use greedy HV-based and IGD-based solution subset selection methods in our experiments to evaluate each MOEA/D version in this paper. A recently proposed lazy greedy inclusion technique [39] is used for the speed-up of greedy inclusion solution subset selection. The lazy greedy HV-based inclusion method [39] is used in the solution selection framework when the performance of MOEA/D is evaluated by the HV indicator. Likewise, the lazy greedy IGD-based inclusion method is used when the performance of MOEA/D is evaluated by the IGD indicator. It should be noted that the distance-based greedy solution subset selection method is used for the design of Auto-MOEA/D-SS by the hyper-heuristic method. This is because the solution subset selection method needs to be performed 50,000 times in a single run of the hyper-heuristic method (i.e., a very fast method needs to be used).

Table 6 shows the mean HV value over 31 runs of each MOEA/D version for each test problem under the solution selection framework. The termination condition in Table 6 is 10,000 solution evaluations. The Wilcoxon's rank sum test at a significant level of 5% is used to evaluate the statistical difference between Auto-MOEA/D-SS and each of the six versions of MOEA/D. In Table 6, the signs "+", "−", and "=" are used to indicate that the compared MOEA/D version is statistically better than, worse than, or equivalent to Auto-MOEA/D-SS. The best and worst results among the seven MOEA/D versions are highlighted using yellow color and red color, respectively.

Auto-MOEA/D-SS shows high performance on many test problems when it is evaluated under the solution selection framework in Table 6. Even though Auto-MOEA/D-SS does not show the best performance on some test problems, it shows very similar performance to the best results obtained on those problems.

Whereas Auto-MOEA/D-SS shows the worst performance on DTLZ2, WFG3, WFG9, Minus-DTLZ2 and Minus-WFG7 in Table 4 (evaluation under the final population framework), it shows the best performance on WFG3, WFG9, and Minus-WFG7, and comparable performance on DTLZ2 and Minus-DTLZ2 (i.e., the HV values are very similar to the best results on these two test problems) in Table 6. These observations clearly show that the optimal algorithm configuration for the solution selection framework can be totally different from that for the final population framework.

Moreover, we can see that better results are obtained in Table 6 than Table 4 for almost all cases. Actually, higher average HV values are obtained in 181 cases (out of 26 test problems $\times$ 7 algorithms $=$182 cases). This observation shows the usefulness of the solution selection framework with an unbounded external archive. Theoretically, the result of the solution selection framework is always better than or equal to that of the final population framework if we can select the best solution set from the examined solutions. Since the HV-based greedy selection method is used (i.e., since the greedy algorithm is not an exact optimization algorithm), better results are obtained in Table 4 than Table 6 for one case (among the 182 cases). This observation suggests that further improvement is needed for solution selection in future studies.

In Table 6, Auto-MOEA/D-SS shows high performance. However, if compared with the high performance of Auto-MOEA/D-FP in Table 4, the performance of Auto-MOEA/D-SS is not so dominant in Table 6. This is because the distance-based solution selection is used in the hyper-heuristic algorithm (for efficient calculation). If the HV-based solution selection method is used, the performance of Auto-MOEA/D-SS will be further improved. However, at the same time, much more computation time is needed.

In Table 6, we can also see that the difference in the average HV values between Auto-MOEA/D-FP and Auto-MOEA/D-SS is very small for all test problems (in comparison with the difference in Table 4) even when there exists a statistically significant difference. This may mean that the search for the best algorithm configuration is not easy under the solution selection framework since different configurations have similar performance. The search ability of the hyper-heuristic method will be improved by increasing the number of runs to evaluate each algorithm

TABLE 6. The mean hypervolume value over 31 runs of each MOEA/D on three-objective test problems using 10,000 solution evaluations under the solution selection framework (with the lazy greedy HV-based solution subset selection method). The best result is highlighted by yellow colour and the worst result is highlighted by red font.

Test problems (M=3)	10,000 solution evaluations (HV)						
	MOEA/D-WS	MOEA/D-TCH	MOEA/D-PBI	MOEA/D-IPBI	MOEA/D-MTCH	Auto-MOEA/D-FP	Auto-MOEA/D-SS
DTLZ1	0.4453 −	0.6455 =	0.4629 −	0.1016 −	0.6528 =	0.6688 =	**0.6950**
DTLZ2	0.3917 −	0.5658 +	0.5629 −	0.3919 −	0.5655 +	**0.5658** +	0.5634
DTLZ3	**0.0065** =	0.0000 =	0.0000 =	0.0000 =	0.0000 =	0.0000 =	0.0000
DTLZ4	0.2023 −	0.2821 −	0.3345 −	0.3503 −	0.3670 −	0.5652 −	**0.5660**
WFG1	0.4444 −	0.4911 −	0.3743 −	0.4944 −	0.5162 −	0.5490 =	**0.5638**
WFG2	0.7926 −	0.8275 −	0.7683 −	0.8259 −	0.8339 −	**0.9068** +	0.8978
WFG3	0.4367 −	0.5103 −	0.4565 −	0.4863 −	0.5169 −	0.5267 −	**0.5431**
WFG4	0.5241 −	0.5427 −	0.4813 −	0.5407 −	0.5451 −	0.5425 −	**0.5617**
WFG5	0.4866 −	0.5185 −	0.4628 −	0.5041 −	0.5180 −	0.5172 −	**0.5210**
WFG6	0.3383 −	0.5178 +	0.4596 −	0.3959 −	0.5158 =	**0.5242** =	0.5144
WFG7	0.4224 −	0.5549 −	0.4010 −	0.4799 −	0.5485 −	**0.5560** +	0.5550
WFG8	0.2842 −	0.4800 +	0.4269 −	0.2944−	0.4781 +	**0.4861** +	0.4626
WFG9	0.4471 −	0.4987 −	0.3970 −	0.4390 −	0.5013 −	0.4987 −	**0.5184**
Minus-DTLZ1	0.1563 −	0.2131 −	0.2156 −	0.2051 −	0.2176 −	0.2194 −	**0.2210**
Minus-DTLZ2	**0.5430** +	0.5400 +	0.5401 +	0.5428 +	0.5409 =	0.5416 +	0.5399
Minus-DTLZ3	0.5168 −	0.4963 −	0.4931 −	0.5072 −	0.5007 −	0.5387 =	**0.5401**
Minus-DTLZ4	0.4838 −	0.5407 −	0.4061 −	0.4682 −	**0.5418** +	0.5404 −	0.5409
Minus-WFG1	0.0386 −	0.0732 −	0.0553 −	0.0520 −	0.1102 −	0.1127 −	**0.1145**
Minus-WFG2	0.2199 −	0.2948 −	0.2787 −	0.2386 −	0.2956 −	0.2925 −	**0.2965**
Minus-WFG3	0.1314 −	0.2137 −	0.2184 −	0.1843 −	0.2170 −	0.2168 −	**0.2205**
Minus-WFG4	0.5389 =	0.5078 −	0.5208 −	0.5390 =	0.5237 −	**0.5417** +	0.5396
Minus-WFG5	0.5365 −	0.5235 −	0.5292 −	0.5362 −	0.5297 −	0.5382 −	**0.5393**
Minus-WFG6	0.5401 −	0.5337 −	0.5340 −	0.5410 −	0.5356 −	**0.5417** +	0.5415
Minus-WFG7	0.5402 −	0.4989 −	0.5181 −	0.5404 −	0.5188 −	0.5415 −	**0.5425**
Minus-WFG8	0.5419 =	0.5388 −	0.5388 −	0.5425 +	0.5392 −	**0.5427** +	0.5410
Minus-WFG9	0.5328 −	0.5150 −	0.5208 −	0.5334 −	0.5178 −	0.5354 =	**0.5355**
+/−/=	1/22/3	4/20/2	1/24/1	2/22/2	3/19/4	8/12/6	

configuration. However, this needs more computation time.

The performance of MOEA/D under the solution selection framework is also examined using a different performance indicator (i.e., IGD) and a different termination condition (i.e., 50,000 solution evaluations). Table 7 shows the summary of statistical comparison results between Auto-MOEA/D-SS with each of the other six MOEA/D versions using different evaluation conditions. The signs "+", "−", and "=" indicate the number of test instances on which the results of Auto-MOEA/D-SS are significantly better than, worse than, or equivalent to other MOEA/D versions.

In Table 7, Auto-MOEA/D-SS shows similar performance to Auto-MOEA/D-FP on average when the termination condition of 50,000 solution evaluations is used. Although statistical differences are observed in the performance between Auto-MOEA/D-SS and Auto-MOEA/D-FP, their average HV values are very similar as in Table 6. Auto-MOEA/D-SS outperformed the other five MOEA/D versions on average in Table 7 when the HV indicator is used for performance comparison.

When the IGD indicator is used in Table 7, the performance of Auto-MOEA/D-SS is not the best. For more than 15 test problems (out of 26 test problems), Auto-MOEA/D-SS is outperformed by Auto-MOEA/D-FP. One possible reason is that different configurations are needed to obtain good solution sets for different performance indicators. That is, the obtained configurations for the HV indicator are not always good for the IGD indicator. Another possible reason is that the optimization of Auto-MOEA/D-SS has not been fully completed as discussed for Table 6 (i.e., use of the distance-based selection method and similar performance of different configurations).

D. DISCUSSION

In general, offline automated algorithm design methods for evolutionary algorithms have the following limitations:

1) LARGE COMPUTATION LOAD RELATED TO FITNESS EVALUATION

Large computation load for evaluating each solution (i.e., each evolutionary algorithm implementation) are needed. Noisy evaluation of each solution due to the stochastic nature of evolutionary algorithm also poses a difficulty to the offline automated algorithm design methods. In order to handle the noisy evaluation issue, multiple runs are needed. Otherwise, it is very likely that a solution with a lucky run (e.g., lucky random initial solutions) will be chosen as the final solution. That is, the search by the hyper-heuristic algorithm is to choose a lucky solution (instead of a good algorithm implementation). However, it is difficult to perform many runs due to the large computation load.

TABLE 7. Summary of statistical comparison results between Auto-MOEA/D-SS with the other MOEA/D versions under different conditions.

Auto-MOEA/D-SS vs.		MOEA/D-WS	MOEA/D-TCH	MOEA/D-PBI	MOEA/D-IPBI	MOEA/D-MTCH	Auto-MOEA/D-FP
Evaluation using HV with 50,000 solutions evaluations	+	19	17	19	20	16	9
	−	4	5	5	6	5	10
	=	3	4	2	0	5	7
Evaluation using IGD with 10,000 solutions evaluations	+	18	12	18	19	8	5
	−	6	8	6	7	8	16
	=	2	6	2	0	10	5
Evaluation using IGD with 50,000 solutions evaluations	+	19	12	9	19	7	6
	−	7	14	9	7	15	17
	=	0	0	8	0	4	3

2) LARGE COMPUTATION LOAD RELATED TO OPTIMIZATION

By using various components and parameters of an evolutionary algorithm as decision variables in an offline automated algorithm design, the optimization problem by hyper-heuristic become small. Thus, it may be easy to find a good solution. However, the obtained good solution is not always a good algorithm implementation since many components and parameters are not optimized by hyper-heuristics. Thus, we need to appropriately choose only important (influential) components and parameters, and we need to appropriately specify other components and parameters in the hyper-heuristic algorithm. Both of these two tasks are not easy.

In addition to these difficulties, in the automated design of EMO algorithms, the following are difficult:

3) CHOICE OF A PERFORMANCE INDICATOR IN THE HYPER-HEURISTIC ALGORITHM

In order to evaluate each implementation of an EMO algorithm, a performance indicator is needed. The choice of a performance indicator and its specification (e.g., a reference point for hypervolume calculation) is not easy.

4) CALCULATION OF THE PERFORMANCE INDICATOR

When the hypervolume indicator is selected, its calculation is time-consuming for many-objective problems and large solution sets.

Moreover, when an unbounded external archive is used, the following are difficult:

5) SELECTION OF THE FINAL SOLUTION SET

We may need an efficient and effective solution selection method since the size of the unbounded external archive can be very large.

6) MEMORY SIZE FOR THE UNBOUNDED EXTERNAL ARCHIVE

If an original multi-objective problem has a large number of decision variables (e.g., a large-scale problem with one million decision variables), it may need some trick to store a huge number of solutions.

V. CONCLUSION

In this paper, we empirically demonstrated the usefulness and flexibility of the solution selection framework using the MOEA/D algorithm on 26 test problems. An offline genetic algorithm-based hyper-heuristic method was used to search for the optimal MOEA/D configurations for each test problem under the final population framework and the solution selection framework. The experimental results suggested the optimal configurations can be totally different under the two frameworks. Better solution sets were obtained from the solution selection framework with a greedy solution subset selection method for almost all test problems than the final population framework.

As shown in this paper (and some other studies [14], [19]), better solution sets are usually obtained from the solution selection framework than the final population framework. However, new algorithm design has not been studied under the solution selection framework in the literature. This is clearly a promising future research direction. Another important research direction is the development of a new efficient and effective solution subset selection method to facilitate the solution selection framework.

REFERENCES

[1] J. K. Rajesh, S. K. Gupta, G. P. Rangaiah, and A. K. Ray, "Multi-objective optimization of industrial hydrogen plants," *Chem. Eng. Sci.*, vol. 56, no. 3, pp. 999–1010, Feb. 2001.

[2] F. Altiparmak, M. Gen, L. Lin, and T. Paksoy, "A genetic algorithm approach for multi-objective optimization of supply chain networks," *Comput. Ind. Eng.*, vol. 51, no. 1, pp. 196–215, Sep. 2006.

[3] L. Sun, G. W. DePuy, and G. W. Evans, "Multi-objective optimization models for patient allocation during a pandemic influenza outbreak," *Comput. Oper. Res.*, vol. 51, pp. 350–359, Nov. 2014.

[4] W. Zhang, K. Cao, S. Liu, and B. Huang, "A multi-objective optimization approach for health-care facility location-allocation problems in highly developed cities such as Hong Kong," *Comput., Environ. Urban Syst.*, vol. 59, pp. 220–230, Sep. 2016.

[5] C. A. C. Coello, S. G. Brambila, J. F. Gamboa, M. G. C. Tapia, and R. H. Gómez, "Evolutionary multiobjective optimization: Open research areas and some challenges lying ahead," *Complex Intell. Syst.*, vol. 6, no. 2, pp. 221–236, Jul. 2020.

[6] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[7] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation discussion and generalization," in *Proc. 5th ICGA*, 1993, pp. 416–423.

[8] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Dec. 1994.

[9] 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, 1999.
[10] 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.
[11] C. A. Coello Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.
[12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Design, Optim. Control Appl. Ind. Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papailiou, and T. Fogarty, Eds., Athens, Greece, 2001, pp. 95–100.
[13] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proc. Congr. Evol. Comput. (CEC)*, Washington, DC, USA, 1999, pp. 98–105.
[14] H. Ishibuchi, L. M. Pang, and K. Shang, "A new framework of evolutionary multi-objective algorithms with an unbounded external archive," in *Proc. 24th Eur. Conf. Artif. Intell.*, Santiago de Compostela, Spain, vol. 2, Aug./Sep. 2020, pp. 1–8. [Online]. Available: https://doi.org/10.36227/techrxiv.11661276.v1
[15] M. Li and X. Yao, "An empirical investigation of the optimality and monotonicity properties of multiobjective archiving methods," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), vol. 11411. Cham, Switzerland: Springer, 2019, pp. 15–26.
[16] R. Tanabe, H. Ishibuchi, and A. Oyama, "Benchmarking multi-and many-objective evolutionary algorithms under two optimization scenarios," *IEEE Access*, vol. 5, pp. 19597–19619, 2017.
[17] R. Tanabe and H. Ishibuchi, "An analysis of control parameters of MOEA/D under two different optimization scenarios," *Appl. Soft Comput.*, vol. 70, pp. 22–40, Sep. 2018.
[18] 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.
[19] L. M. Pang, H. Ishibuchi, and K. Shang, "Algorithm configurations of MOEA/D with an unbounded external archive," 2020, *arXiv:2007.13352*. [Online]. Available: http://arxiv.org/abs/2007.13352
[20] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. Congr. Evol. Computation. (CEC)*, 2002, pp. 825–830.
[21] 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.
[22] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, Apr. 2017.
[23] 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, Aug. 1998.
[24] R. Wang, J. Xiong, H. Ishibuchi, G. Wu, and T. Zhang, "On the effect of reference point in MOEA/D for multi-objective optimization," *Appl. Soft Comput.*, vol. 58, pp. 25–34, Sep. 2017.
[25] 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.
[26] H. Sato, "Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization," in *Proc. Conf. Genetic Evol. Comput. (GECCO)*, 2014, pp. 645–652.
[27] H. Ishibuchi, K. Doi, and Y. Nojima, "On the effect of normalization in MOEA/D for multi-objective and many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 4, pp. 279–294, Dec. 2017.
[28] H. Ishibuchi, N. Akedo, and Y. Nojima, "Relation between neighborhood size and MOEA/D performance on many-objective problems," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Sheffield, U.K., 2013, pp. 459–474.
[29] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, Apr. 1995.
[30] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *Int. J. Artif. Intell. Soft Comput.*, vol. 4, no. 1, pp. 1–28, 2014.
[31] Z. Michalewicz, *Genetic Algorithms+ Data Structures= Evolution Programs*. Berlin, Germany: Springer, 1996.
[32] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*. London, U.K.: Springer, 2010.
[33] H. K. Singh, K. S. Bhattacharjee, and T. Ray, "Distance-based subset selection for benchmarking in evolutionary multi/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 904–912, Oct. 2019.
[34] K. Bringmann, T. Friedrich, and P. Klitzke, "Generic postprocessing via subset selection for hypervolume and epsilon-indicator," in *Proc. Int. Conf. Parallel Problem Solving Nature (PPSN)*, 2014, pp. 518–527.
[35] T. Kuhn, C. M. Fonseca, L. Paquete, S. Ruzika, M. M. Duarte, and J. R. Figueira, "Hypervolume subset selection in two dimensions: Formulations and algorithms," *Evol. Comput.*, vol. 24, no. 3, pp. 411–425, Sep. 2016.
[36] A. P. Guerreiro and C. M. Fonseca, "Computing and updating hypervolume contributions in up to four dimensions," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 449–463, Jun. 2018.
[37] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.
[38] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 807–813, Oct. 2016.
[39] W. Chen, H. Ishibuchi, and K. Shang, "Lazy greedy hypervolume subset selection from large candidate solution sets," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–9.

LIE MENG PANG (Member, IEEE) received the Bachelor of Engineering degree in electronic and telecommunication engineering and the Ph.D. degree in electronic engineering from the Faculty of Engineering, Universiti Malaysia Sarawak, in 2012 and 2018, respectively.

She is currently a Postdoctoral Researcher with the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), China. Her research interests include fuzzy systems and evolutionary computation.

HISAO ISHIBUCHI (Fellow, IEEE) received the B.S. and M.S. degrees in precision mechanics from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from Osaka Prefecture University, Sakai, Osaka, Japan, in 1992.

He was with Osaka Prefecture University from 1987 to 2017. Since 2017, he has been a Chair Professor with the Southern University of Science and Technology, China. His research interests include fuzzy rule-based classifiers, evolutionary multiobjective and many-objective optimization, memetic algorithms, and evolutionary games. He was the IEEE CIS Vice-President for Technical Activities from 2010 to 2013 and an AdCom Member of the IEEE CIS from 2014 to 2019. He served as the Editor-in-Chief for *IEEE Computational Intelligence Magazine* from 2014 to 2019.

KE SHANG (Member, IEEE) received the B.S. and Ph.D. degrees from Xi'an Jiaotong University, China, in 2009 and 2016, respectively.

From 2017 to 2019, he was a Postdoctoral Researcher with the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), China, where he is currently a Research Assistant Professor. His current research interests include evolutionary multiobjective optimization and its applications. He was a recipient of the GECCO 2018 Best Paper Award and the CEC 2019 First Runner-Up Conference Paper Award.