A Simple and Fast Hypervolume Indicator-based Multiobjective Evolutionary Algorithm

Siwei Jiang, Jie Zhang, Yew-Soon Ong, Allan Nengsheng Zhang and Puay Siew Tan

***Abstract*—To find diversified solutions converging to true Pareto fronts (PFs), hypervolume indicator-based algorithms have been established as effective approaches in multiobjective evolutionary algorithms (MOEAs). However, the bottleneck of hypervolume indicator-based MOEAs is the high time complexity for measuring the exact hypervolume contributions of different solutions. To cope with this problem, in this paper, a simple and Fast hyperVolume indicator-based MOEA (FV-MOEA) is proposed to quickly update the exact hypervolume contributions of different solutions. The core idea of FV-MOEA is that the hypervolume contribution of a solution is only associated with partial solutions rather than the whole solution set. Thus, the time cost of FV-MOEA can be greatly reduced by deleting irrelevant solutions. Experimental studies on 44 benchmark multiobjective optimization problems (MOPs) with 2–5 objectives in platform jMetal demonstrate that FV-MOEA not only reports higher hypervolumes than the five classical MOEAs (NSGAII, SPEA2, MOEA/D, IBEA and SMS-EMOA), but also obtains significant speedup compared to other hypervolume indicator-based MOEAs.**

***Index Terms*—Multiobjective Evolutionary Algorithms, Pareto Dominance-based, Scalarizing Function-based, Indicator-based, Hypervolume, jMetal.**

I. Introduction

Many real-world problems can be formulated as multiobjective optimization problems (MOPs), which involve several conflicting objectives to be optimized simultaneously [1–43]. A minimization of MOP can be stated as follows:

$$\begin{array}{rcl} min\ \ F(x) & = & (f_1(x),\ldots,f_d(x)) \\ s.t.\ \ G(x) & \leq & 0,\ H(x)\ =\ 0,\ x\ \in\ \Omega \end{array} \quad (1)$$

where $x = (x_1, \ldots, x_m)$, Ω is the decision (*variable*) *space*, R^d is the *objective space*, and $F : \Omega \rightarrow R^d$ consists of d real-valued objective functions with constraints $G(x) \leq 0, H(x) = 0$. The feasible solution space is $\Omega = \Pi_{i=1}^{m}[L_i, U_i]$, and L_i, U_i are the lower and upper bound of x_i, respectively.

Multiobjective evolutionary algorithms (MOEAs) have been well established as effective approaches to deal with MOPs [1–43]. Based on various *acceptance rules* for selecting offspring solutions, the classical MOEAs can be generally divided to three groups: 1) Pareto dominance-based approaches (*e.g.*, NSGAII [6], SPEA2 [7]), 2) Scalarizing function-based methods (*e.g.*, MOEA/D [8–12]) and 3) Indicator-based algorithms (*e.g.*, IBEA [15], SMS-EMOA [16, 39–41]).

Pareto dominance-based approaches utilize the Pareto dominance concept together with the crowding distance (NSGAII [6]) or clustering methods (SPEA2 [7]) to select offspring. On the other hand, scalarizing function-based methods calculate solution fitness using predefined weight vectors and perform solution selection in different subproblems (MOEA/D [8–12]). In essence, the heuristic selection schemes in these MOEAs can be considered as approximated performance indicators to measure the quality of solutions.

In contrast to the two mentioned groups of MOEAs, indicator-based algorithms directly use performance indicators (*e.g.*, hypervolume, epsilon metric $I^1_{\epsilon+}$) to select offspring [13–16, 31–33, 36–41]. To date, several hypervolume indicator-based MOEAs have been proposed to deal with MOPs. For instance, SMS-EMOA [16, 39–41] is designed as a steady-state MOEA to measure exact hypervolume contributions of different solutions. The superiority of SMS-EMOA over the two above groups of MOEAs has been verified by a plethora of research studies [16, 37–41]. However, due to the high time complexity of hypervolume calculation, it is unpractical to apply SMS-EMOA to high-dimensional MOPs as well as a large number of solutions. To alleviate this problem, Zitzler *et al.* [15] proposed a general indicator-based evolutionary algorithm (IBEA) that approximates hypervolume contributions by aggregating the hypervolume differences of pairwise solutions. In addition, Bringmann *et al.* [31–33, 36] and Bader *et al.* [34, 35] adopted the Monte Carlo sampling method to estimate hypervolumes. On the other hand, Ishibuchi *et al.* [13, 14] transformed hypervolume contributions as a form of distances between solutions and predefined weight vectors. Although the time cost is reduced, the accuracy of hypervolumes in [13–15, 31–36] is compromised.

In view of this problem, we make an attempt to enhance the efficiency of hypervolume indicator-based MOEAs without sacrificing their effectiveness. In this paper, a simple and **F**ast hyper**V**olume indicator-based MOEA (FV-MOEA) is proposed to quickly update the exact hypervolume contributions of different solutions. In comparison to SMS-EMOA using the whole solution set, FV-MOEA only involves part of solutions. Thus, the time cost for calculating hypervolume contributions

Manuscript received August 18, 2013; revised April 22, 2014; accepted Oct 26, 2014. This work is partially supported under the A*Star-TSRP funding, Singapore Institute of Manufacturing Technology-Nanyang Technological University (SIMTech-NTU) Joint Laboratory and Collaborative research Programme on Complex Systems, and the Computational Intelligence Research Laboratory at NTU.

Siwei Jiang is with Singapore Institute of Manufacturing Technology (SIMTech), Singapore 638075 (e-mail: jiangsw@simtech.a-star.edu.sg).

Jie Zhang and Yew-Soon Ong are with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore 639798 and SIMTech-NTU Joint Laboratory on Complex Systems (e-mail: zhangj@ntu.edu.sg, asysong@ntu.edu.sg).

Allan Nengsheng Zhang and Puay Siew Tan are with Singapore Institute of Manufacturing Technology (SIMTech), Singapore 638075 and SIMTech-NTU Joint Laboratory on Complex Systems (e-mail: nzhang@simtech.a-star.edu.sg, pstan@simtech.a-star.edu.sg).

in FV-MOEA can be greatly saved by deleting irrelevant solutions. In addition, when a solution is removed from a population, FV-MOEA transfers its hypervolume contribution to others. This is beneficial to reduce the time cost of recalculating hypervolume contributions when the population is changed. Furthermore, different from the studies in [13–15, 31–36] that only find approximated hypervolumes, FV-MOEA measures exact hypervolumes. Experimental studies on 44 MOPs with 2–5 objectives in jMetal [4, 5] demonstrate that FV-MOEA not only reports higher hypervolumes than the five classical MOEAs, but also obtains significant speedup compared to other hypervolume indicator-based MOEAs.

In summary, the core contributions of the proposed FV-MOEA are highlighted as follows.

- A simple and fast method for measuring hypervolume contributions is introduced. To the best of our knowledge, this paper serves as the first attempt that reduces the computational burden of calculating exact hypervolume contributions by means of deleting irrelevant solutions and transferring hypervolume contributions.
- A batch model for selecting offspring in hypervolume indicator-based MOEAs is proposed. This mechanism can greatly save the time cost of recalculating hypervolume contributions when a population is changed.
- Experimental studies demonstrate that FV-MOEA is significantly better than the five classical MOEAs (*i.e.*, NSGAII, SPEA2, MOEA/D, IBEA and SMS-EMOA) in term of hypervolume and time cost.

The rest of this paper is organized as follows. The related work of hypervolume calculation is surveyed in Section II. Section III introduces the fast hypervolume method and the proposed FV-MOEA in detail. Experimental studies on 44 MOPs with 2–5 objectives are presented in Section IV. Finally, the conclusions and future work are given in Section V.

II. Related Work

In the last four decades, multiobjective evolutionary algorithms (MOEAs) have gained popularity for solving multiobjective optimization problems (MOPs) [1–43]. Among the classical MOEAs, SMS-EMOA belonging to the indicator-based MOEAs has exhibited superior performance compared to the other two groups of MOEAs (*i.e.*, Pareto dominance-based approaches and scalarizing function-based methods), especially on high-dimensional MOPs [16, 37–41]. Nevertheless, the bottleneck of SMS-EMOA is the high time complexity for computing the exact hypervolume contributions of various solutions [15, 16, 31–36, 39–44].

To alleviate the computational burden of hypervolume calculation, various approaches have been proposed to either fast compute exact hypervolumes [4, 5, 45–56] or obtain approximated hypervolumes [13–15, 31–36]. Specifically, the methods in [4, 5, 45–56] are briefly discussed as follows.

- **HSO**: Hypervolume by slicing objectives (HSO) [4, 5, 45] reduces a d-dimensional hypervolume into several $(d-1)$-dimensional hypervolumes by slicing one objective each time, and then it sums up the hypervolumes of all slices. The worst-case time complexity of HSO is $\mathcal{O}(n^{d-1})$, where n is the size of solutions, d is the number of objectives. In addition, two improved versions of HSO [46, 47] adopt an incremental strategy and an iterative schema to speed up hypervolume calculation.
- **FPL**: Fonseca, Paquete and López-Ibánez (FPL) [48] prunes recursion trees for avoiding repeated domination checks and the recalculation of partial hypervolumes. FPL's worst-case time complexity is $\mathcal{O}(n^{d-2}\log n)$.
- **HOY**: Hypervolume by Overmars and Yap (HOY) [49–51] converts a hypervolume calculation problem to a Klee's measure problem, which has the worst-case time complexity of $\mathcal{O}(n\log n + n^{\frac{d}{2}}\log n)$. In addition, an improved version of HOY [52] recursively divides a hypervolume space into trellises. Since calculating hypervolumes within the trellises is trivial, HOY's worst-case time complexity reduces to $\mathcal{O}(n\log n + n^{\frac{d}{2}})$. Another improved version of HOY [53] arrives at the time complexity of $\mathcal{O}(n^{\frac{d}{2}})$ in 4-dimensional hypervolume calculation by adopting a fast dimension-sweep method.
- **QHV**: Quick hypervolume (QHV) [54] recursively divides a hypervolume space into several regions according to pivot solutions. When solutions are uniformly distributed on a hyper-sphere or hyper-plane, the time complexity of QHV is $\mathcal{O}(dn^{1.1}\log^{d-2} n)$.
- **WFG**: Walking Fish Group (WFG) [55, 56] proposes an efficient algorithm to calculate the exclusive hypervolume of a solution, where most of other solutions limited within this solution are dominated and will not participate into the hypervolume calculation. While *et al.* [55, 56] prove that the worst-case time complexity of WFG is $\mathcal{O}(2^{n-1})$.

Among the above methods, WFG has been generally accepted as the fastest one for calculating exact hypervolumes on MOPs with different geometrical shapes [37, 55, 56].

On the other hand, the following approaches [13–15, 31–36] have been designed to approximate hypervolumes.

- **General indicator-based methods**: Zitzler *et al.* [15] designed a general indicator-based evolutionary algorithm (IBEA) to roughly estimate the hypervolume contributions of a solution by aggregating the pairwise hypervolume differences between the target solution and other solutions.
- **Monte Carlo sampling methods**: Bader *et al.* [34, 35] proposed a hypervolume-based search algorithm (HypE), where the hypervolume contribution of a solution is treated as the ratio between the size of sampling points dominated by the target solution and the size of the whole sampling points. In addition, the similar idea has been adopted in [31–33, 36] for estimating hypervolumes.
- **Achievement scalarizing function-based methods**: Ishibuchi *et al.* [13, 14] proposed an achievement scalarizing function-based method, which approximates the hypervolume contribution of a solution as the mean value of distances between the target solution and weight vectors that are predefined by users.

Although the above approaches [13–15, 31–36] can save the time cost of hypervolume indicator-based MOEAs, their performance to solve various MOPs is compromised.

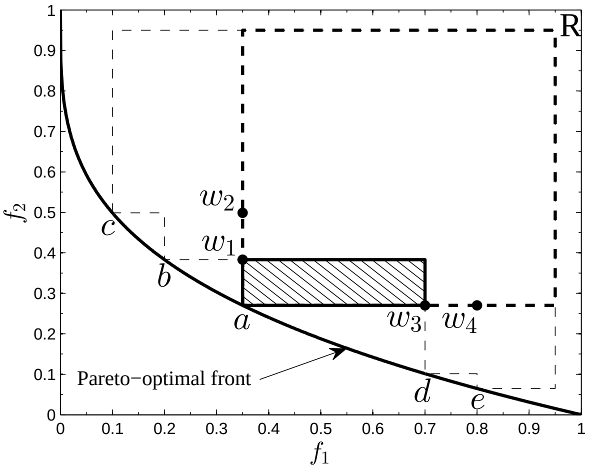


Fig. 1. Hypervolume contribution for one solution $v^1(a, S, R)$

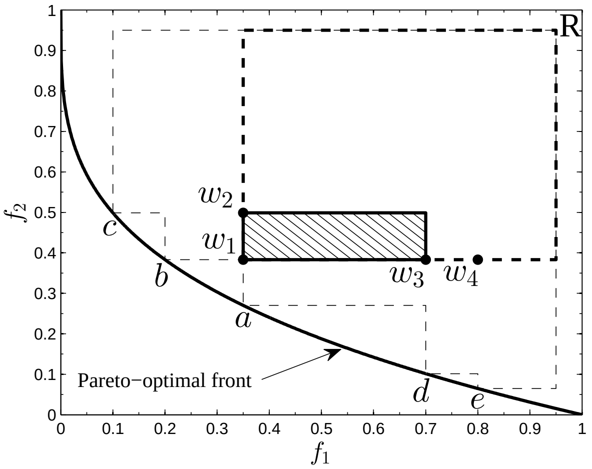


Fig. 2. Hypervolume contribution for two solutions $v^2(\{a, b\}, S, R)$

In contrast to the approximation approaches [13–15, 31–36], the proposed FV-MOEA is designed to calculate exact hypervolume contributions for different solutions. On the other hand, the previous studies in [4, 5, 45–56] focus on calculating exact hypervolumes on a fixed solution set. The special design of FV-MOEA is suitable to deal with the dynamic changes of inserting and deleting solutions in population-based MOEAs, by means of quickly transferring hypervolume contributions among solutions.

III. Fast Hypervolume indicator-based MOEA

In this section, a simple and **F**ast hyper**V**olume indicator-based multiobjective evolutionary algorithm (FV-MOEA) is proposed to quickly update the exact hypervolume contributions of different solutions. For the sake of readability, the general procedure of FV-MOEA is outlined as follows.

- Population initialization. The first population of individual solutions is random generated and evaluated by the conflicting multiobjective fitness functions.
- Offspring reproduction. The offspring solutions are produced based on a selection method and evolutionary operators (*i.e.*, binary tournament, SBX and polynomial mutation [6, 7, 15, 16]), which are commonly used in the classical MOEAs.
- Solution selection. Based on the proposed fast hypervolume method, FV-MOEA updates hypervolume contributions and selects offspring with a batch model.

Since FV-MOEA does not involve special initialization or reproduction methods, we will put concentration on the third element (*i.e.*, solution selection) in the following sections.

A. Hypervolume Contribution for One Solution

To calculate the hypervolume contribution for one solution, we design a *nondominated-worse* function to delete irrelevant solutions, which can largely save the time cost for computing the hypervolume of the whole solution set.

Taking Fig. 1 as an illustrative example, suppose a nondominated solution set $S = \{a, b, c, d, e\}$ is on the 2-dimensional Pareto-optimal front, and the reference set is $R = \{r\}$ that only involves one solution. The hypervolume of set S with respect to R, denoted as $\text{HV}(S, R)$, is the area $cbadeRc$ enclosed into the dash discontinuous line [4, 5, 13–15, 31–36, 42, 43, 45–56]. Then, a straightforward way to calculate the hypervolume contribution of solution a is proposed in SMS-EMOA [16, 39–41], taking the form of:

$$v^1(a, S, R) = \text{HV}(S, R) - \text{HV}(S - a, R), \qquad (2)$$

where $a \in S$ and $v^1(a, S, R)$ is the rectangle filled with slash lines in Fig. 1.

In general, the time cost of Eq. 2 is high, because it needs to calculate the hypervolume for all solutions $\text{HV}(S, R)$ and the hypervolume of the set excluding one solution $\text{HV}(S - a, R)$. For instance, suppose HSO[1] [4, 5, 45] is adopted to compute hypervolumes, the time complexity is $\mathcal{O}(n^{d-1})$ and $|S| = n$ (see Section II). This means that the time cost of hypervolume calculation increases significantly as the size of solution set $|S|$ and dimension of MOPs d grow. Another limitation of Eq. 2 is that we have to recalculate the two terms (*i.e.*, $\text{HV}(S, R)$ and $\text{HV}(S - a, R)$) when some solutions are removed from or added into set S.

To cope with the above two limitations, in this paper, a fast hypervolume method is proposed to compute $v^1(a, S, R)$ by removing irrelevant solutions. At first, we define a *worse* function to find the worse objective values from two solutions. For instance, to minimize a bi-objective MOP in Fig. 1, the worse solutions w_1, w_2, w_3, w_4 have the maximum objective values between a and b, c, d, e, respectively. Then, the *nondominated-worse* function for solution $a \in S$ is defined as:

$$W = \text{nondominated-worse}(a, S - a). \qquad (3)$$

In Fig. 1, $W = \text{nondominated}(\{w_1, w_2, w_3, w_4\}) = \{w_1, w_3\}$ since w_2, w_4 are dominated by w_1, w_3, respectively.

[1] HSO is implemented in the commonly used platform jMetal [4, 5].

```
Input : $S$, nondominated solution set; $R$, reference set; $N$, required size;
Output: $S$, solution set after hypervolume selection;
1  for $s_i \in S$ do
2      $W =$ nondominated-worse$(s_i, S - \{s_i\})$
3      $v^1(s_i, S, R) =$ HV$(s_i, R) -$ HV$(W, R)$
4  while $|S| > N$ do
5      $j = \min_{s_i \in S} v^1(s_i, S, R)$
6      for $s_k \in S, s_k \neq s_j$ do
7          $w =$ worse$(s_j, s_k)$
8          $W =$ nondominated-worse$(w, S - \{s_j, s_k\})$
9          $v^2(\{s_j, s_k\}, S, R) =$ HV$(w, R) -$ HV$(W, R)$
10         $v^1(s_k, S, R) += v^2(\{s_j, s_k\}, S, R)$
11     $S = S - s_j$
12 Output nondominated solution set $S$
```

Algorithm 1: The procedure of FastHypervolume.

Based on the above definitions, an alternative way to calculate $v^1(a, S, R)$ is proposed as:

$$v^1(a, S, R) = \text{HV}(a, R) - \text{HV}(W, R). \tag{4}$$

In contrast to Eq. 2, the time cost of Eq. 4 is relatively low, because it only needs to calculate the dominated volume of one solution HV$(a, R) = \prod_{i=1}^{d} |a_i - r_i|$ and the term HV(W, R) is only associated with partial solutions (*i.e.*, solutions $\{b, d\} \in S$ in Fig. 1) [55, 56]. In addition, the methods discussed in Section II, including HSO [4, 5, 45–47], FPL [48], HOY [49–53], QHV [54], WFG [55, 56], can be adopted to calculate the hypervolume for solutions (*i.e.*, HV(W, R)). In Section IV, the popular WFG is selected to calculate hypervolumes in experimental studies.

B. Hypervolume Contribution for Two Solutions

By using the *nondominated-worse* function in Eq. 3, we can quickly obtain the hypervolume contribution of two solutions. Taking Fig. 2 as an illustrative example, the worse solution $w_1 =$ worse(a, b) has the maximum objective values of solutions a and b. Then, the nondominated worse solutions for w_1 is found as:

$$W = \text{nondominated-worse}(w_1, S - \{a, b\}), \tag{5}$$

where $W =$ nondominated$(\{w_2, w_3, w_4\}) = \{w_2, w_3\}$ in Fig. 2. The hypervolume contribution of the two solutions $\{a, b\}$ is the rectangle with slash lines in Fig. 2, which is calculated by the following method:

$$v^2(\{a, b\}, S, R) = \text{HV}(w_1, R) - \text{HV}(W, R). \tag{6}$$

Based on Eq. 6, the hypervolume contribution of two solutions can be transferred to one solution when another solution is deleted from the solution set. For instance, suppose that solution a is removed from solution set S and $S' = S - a = \{b, c, d, e\}$. The new hypervolume contribution of solution b can be quickly updated as:

$$v^1(b, S', R) = v^1(b, S, R) + v^2(\{a, b\}, S, R). \tag{7}$$

```
Input : NP, population size; Max_FES, maximum function evaluations; $b$, batch size;
Output: $P$, the nondominated population;
1  $P_g =$ Initialization(), $g = 0$
2  $t = 0$
3  while $t <=$ Max_FES do
4      $Q_{g+1} = \phi$
5      for $i = 1$ $to$ $b$ do
6          $x_{i,g+1} =$ Generate$(P_g)$
7          $Q_{g+1} = Q_{g+1} \bigcup \{F(x_{i,g+1})\}$
8          $t$++
9      $Q_{g+1} = P_g \bigcup Q_{g+1}$
       // Select Offspring from Nondominated Fronts
10     $\{\mathcal{F}_1, \mathcal{F}_2, \cdots\} =$ nondominated-sort$(Q_{g+1})$
11     $P_{g+1} = \phi$, $i = 1$
12     while $|P_{g+1}| + |\mathcal{F}_i| < NP$ do
13         $P_{g+1} = P_{g+1} \bigcup \mathcal{F}_i$
14         $i$++
       // Select Offspring by FastHypervolume Alg. 1
15     $R =$ ConstructReferenceSet$(Q_{g+1})$
16     $\mathcal{F}_i =$ FastHypervolume$(\mathcal{F}_i, R,$ NP $- |P_{g+1}|)$
17     $P_{g+1} = P_{g+1} \bigcup \mathcal{F}_i$
18     $g$++
19 Output nondominated population $P_g$
```

Algorithm 2: The procedure of FV-MOEA.

In Fig. 2, the new hypervolume contribution of solution b, termed as $v^1(b, S', R)$, is the sum of its original hypervolume contribution $v^1(b, S, R)$ (*i.e.*, rectangle area bw_1w_2b) and the hypervolume contribution of two solutions $v^2(\{a, b\}, S, R)$. It is worth noting that this special mechanism overcomes the limitation of Eq. 2 for recalculating hypervolume contributions when some changes are detected in set S.

C. Pseudocode of Solution Selection by Fast Hypervolume

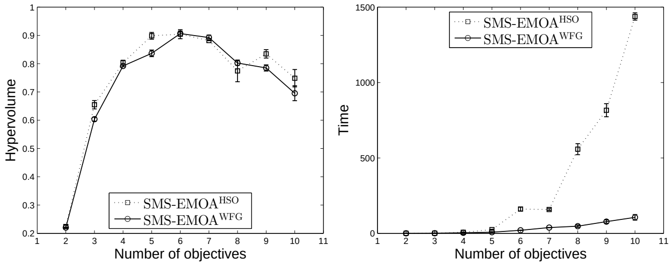
The pseudocode of the fast hypervolume method to select solutions with a batch model is described in Alg. 1.

In Alg. 1, Lines 1-3 calculate each solution's hypervolume contribution $v^1(s_i, S, R), s_i \in S$ by Eq. 4. Line 5 finds the solution s_j that has the smallest hypervolume contribution. Lines 7-9 obtain the hypervolume contribution of two solutions $v^2(\{s_j, s_k\}, S, R)$ by Eq. 6, where $s_k \neq s_j, s_k \in S$. Then, Line 10 updates the hypervolume contribution of s_k by Eq. 7. In Line 11, solution s_j is deleted from set S. The procedure of Lines 4-11 repeats until the number of solutions in S reaches the predefined size N. It is worth noting that hypervolume contribution for one solution by Eq. 4 ($v^1(s_i, S, R), s_i \in S$) is not recalled in Lines 4-11. Finally, the solution set after hypervolume selection is output in Line 12.

D. Pseudocode of the Proposed FV-MOEA

Based on the solution selection of Alg. 1, the pseudocode of the proposed FV-MOEA is given in Alg. 2.

In Alg. 2, Lines 1-2 random initialize the population P_0. In Lines 5-8, the offspring solutions Q_{g+1} is produced based



(a) Hypervolume (b) Time (s)

Fig. 3. The median and IQR of **Hypervolume** (a) and **Time** (b) derived by SMS-EMOA$^{\text{HSO}}$ and SMS-EMOA$^{\text{WFG}}$ with population size NP = 50 on $\{2, \cdots, 10\}$–dimensional DTLZ2 over 5 independent runs.

on a selection method and evolutionary operators (*i.e.*, binary tournament, SBX and polynomial mutation). Line 10 finds the fronts $\{\mathcal{F}_1, \mathcal{F}_2, \cdots\}$ by the nondominated-sort method in NSGAII [6]. Lines 11-14 select solutions from nondominated fronts into P_{g+1} until the solution size in front $\mathcal{F}_i$ is larger than NP$-|P_{g+1}|$. In Line 15, the reference set R is constructed by using the extreme objective values found in Q_{g+1}[2]. Line 16 uses the fast hypervolume method in Alg. 1 for selecting NP $-$ $|P_{g+1}|$ solutions from front $\mathcal{F}_i$. When the termination criterion is satisfied (*i.e.*, $t >$ Max_FES), the nondominated population P_g is output in Line 19.

IV. Experimental Results and Discussions

The experiments are conducted on jMetal 4.3 [4, 5], which is a Java-based framework that is aimed at facilitating the development of metaheuristics for solving MOPs[3].

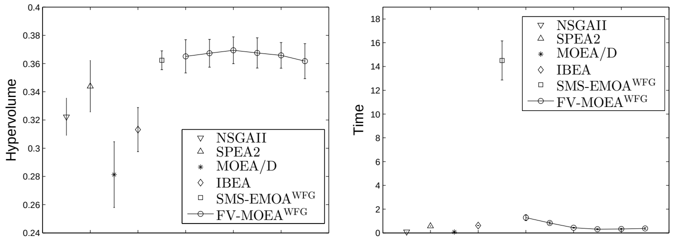
A. Benchmark Problems and Experimental Settings

The benchmark MOPs include 44 testing instances: 5 MOPs in ZDT family problems ($d = 2$) [57], 7 MOPs in DTLZ family problems ($d = 3$) [58], 18 MOPs in WFG family problems[4] [59]. In addition, 14 high-dimensional DTLZ family problems with $d = 4, 5$ objectives [58] are also involved.

The testing environment is described as follows: the operation system is Ubuntu kernel Linux 2.6.38-16-server GNOME 2.32.1, CPU is Intel(R) Xeon(R) E31270@3.40GHz×8, the memory is 16G, and jMetal is running with JRE 1.7.

The major experimental settings are outlined as follows.

1) Population size: In MOEA/D [8], the population size is decided by the number of weight vectors C_{H+d-1}^{d-1} (d is the objective number, H is a predefined integer). NP $= 50, 55, 56, 70$ for $d = 2, 3, 4, 5$ objectives ($H = 49, 9, 5, 4$), respectively. Other algorithms have the same population size as MOEA/D on various MOPs.
2) Maximum function evaluations: Max_FES = $15,000$.
3) Independent run times: Runs = 20.
4) Selection method in FV-MOEA: Binary tournament [5].
5) Evolutionary operators in NSGAII [6], SPEA2 [7], IBEA [15], SMS-EMOA [16] and FV-MOEA:
   - SBX: $p_c = 0.9$, $\eta_c = 20$.
   - Polynomial mutation: $p_m = 1/n$, $\eta_m = 20$.
6) DE operator in MOEA/D [8]: $CR = 1.0, F = 0.4$.
7) Batch size in FV-MOEA: $b = 0.2 \times$ NP.



(a) Hypervolume (b) Time (s)

Fig. 4. The median and IQR of **Hypervolume** (a) and **Time** (b) by classical MOEAs and FV-MOEA$^{\text{WFG}}$ with $b = \{0.05, 0.1, 0.2, 1.0, 2.0, 5.0\} \times$ NP on 30 low-dimensional MOPs ($d = 2, 3$) over 20 independent runs.

TABLE I
Hypervolume and **Time** Median and IQR by Classical MOEAs and FV-MOEA$^{\text{WFG}}$ with $b = \{0.05, 0.1, 0.2, 1.0, 2.0, 5.0\} \times$ NP on 30 Low-dimensional MOPs over 20 Independent Runs.

	Hypervolume	Time (s)
NSGAII	3.223E-01±2.63E-02	9.387E-02±1.23E-03
SPEA2	3.439E-01±3.63E-02	5.714E-01±1.77E-02
MOEA/D	2.813E-01±4.66E-02	7.585E-02±2.91E-03
IBEA	3.132E-01±3.13E-02	6.220E-01±1.13E-02
SMS-EMOA$^{\text{WFG}}$	3.623E-01±1.33E-02	1.451E+01±3.29E-00
FV-MOEA$^{\text{WFG}}$ $b = 0.05 \times$ NP	3.699E-01±2.35E-02	1.284E-00±5.05E-01
FV-MOEA$^{\text{WFG}}$ $b = 0.1 \times$ NP	3.694E-01±1.97E-02	8.326E-01±1.75E-01
FV-MOEA$^{\text{WFG}}$ $b = 0.2 \times$ NP	3.694E-01±1.90E-02	4.370E-01±7.79E-02
FV-MOEA$^{\text{WFG}}$ $b = 1.0 \times$ NP	3.675E-01±2.14E-02	3.112E-01±1.15E-01
FV-MOEA$^{\text{WFG}}$ $b = 2.0 \times$ NP	3.658E-01±1.82E-02	3.327E-01±1.21E-01
FV-MOEA$^{\text{WFG}}$ $b = 5.0 \times$ NP	3.617E-01±2.48E-02	3.719E-01±1.36E-01

Other parameters are set as the default values in jMetal [4, 5].

All the algorithms are evaluated based on hypervolume [34, 35, 42, 43, 60]. In addition, the time cost of various algorithms is also investigated. The high hypervolume and low time cost are desirable. The obtained results are compared using median values and interquartile ranges (IQR). In order to obtain statistically sound conclusion, the Wilcoxon rank sum test with 95% confidence level is conducted on experimental results.

B. Impact of WFG on SMS-EMOA

As mentioned in Section II, several methods have been proposed to calculate exact hypervolumes, including HSO [4, 5, 45–47], FPL [48], HOY [49–53], QHV [54], WFG [55, 56]. They can be adopted to calculate hypervolumes in MOEAs. For instance, SMS-EMOA in jMetal [4, 5] utilizes HSO to compute hypervolume contributions (called SMS-EMOA$^{\text{HSO}}$). In this section, another version of SMS-EMOA is designed with the fastest method (WFG), denoted as SMS-EMOA$^{\text{WFG}}$. Both of them are tested on $\{2, \cdots, 10\}$-dimensional DTLZ2 problems with NP = 50 over 5 independent runs. Fig. 3 shows their results in terms of hypervolume and time cost.

From Fig. 3(a), both SMS-EMOA$^{\text{HSO}}$ and SMS-EMOA$^{\text{WFG}}$ report the similar hypervolumes on $\{2, \cdots, 10\}$-dimensional DTLZ2 problems. On the other hand, Fig. 3(b) shows that SMS-EMOA$^{\text{WFG}}$ is more efficient than SMS-EMOA$^{\text{HSO}}$,

[2] The reference set R involves one solution that can be simply found by constructing a vector of worst objective values from Q_{g+1}.

[3] http://jmetal.sourceforge.net

[4] WFG1-9 with 2 objectives and WFG1-9-3D with 3 objectives.

TABLE II
HYPERVOLUME STATISTICAL RESULTS BY CLASSICAL MOEAS AND FV-MOEA$^{\mathrm{WFG}}$ ON 30 LOW-DIMENSIONAL MOPS ($d = 2, 3$).

	SPEA2	MOEA/D	IBEA	SMS-EMOA$^{\mathrm{WFG}}$	FV-MOEA$^{\mathrm{WFG}}$
NSGAII	19 6 5	6 5 19	17 6 7	22 5 3	24 4 2
SPEA2		2 4 24	14 4 12	23 5 2	26 3 1
MOEAD			22 4 4	25 1 4	25 2 3
IBEA				23 5 2	24 4 2
SMS-EMOA$^{\mathrm{WFG}}$					9 17 4

TABLE IV
TIME STATISTICAL RESULTS BY CLASSICAL MOEAS AND FV-MOEA$^{\mathrm{WFG}}$ ON 30 LOW-DIMENSIONAL MOPS ($d = 2, 3$).

	SPEA2	MOEA/D	IBEA	SMS-EMOA$^{\mathrm{WFG}}$	FV-MOEA$^{\mathrm{WFG}}$
NSGAII	0 0 30	27 3 0	0 0 30	0 0 30	0 0 30
SPEA2		30 0 0	7 3 20	0 0 30	21 7 2
MOEAD			0 0 30	0 0 30	0 0 30
IBEA				0 0 30	23 2 5
SMS-EMOA$^{\mathrm{WFG}}$					30 0 0

which is consistent with the results in [37, 55, 56]. In the following sections, both SMS-EMOA and FV-MOEA will utilize WFG to calculate hypervolumes. They are termed as SMS-EMOA$^{\mathrm{WFG}}$ and FV-MOEA$^{\mathrm{WFG}}$, respectively.

C. Effect on Batch Size

In FV-MOEA$^{\mathrm{WFG}}$, the parameter b is designed for defining the batch size of offspring solutions (see Alg. 2). To investigate the influence of this parameter, FV-MOEA$^{\mathrm{WFG}}$ is tested with $b = \{0.05, 0.1, 0.2, 1.0, 2.0, 5.0\} \times \mathrm{NP}$. The median and IQR of hypervolume and time cost are evaluated on 30 low-dimensional MOPs over 20 independent runs. Table I and Fig. 4 show the mean values on 30 MOPs by the the five classical MOEAs and the proposed FV-MOEA$^{\mathrm{WFG}}$.

From Table I and Fig. 4(a), FV-MOEA$^{\mathrm{WFG}}$ obtains higher mean hypervolumes than the five classical MOEAs except $b = 5.0 \times \mathrm{NP}$. The hypervolume of FV-MOEA$^{\mathrm{WFG}}$ decreases as the batch size increases. The possible reason is that the chance of generating high quality solutions becomes smaller when FV-MOEA$^{\mathrm{WFG}}$ produces more offspring at each generation. For instance, the value of $b = 1$ represents a steady-state FV-MOEA$^{\mathrm{WFG}}$ that inserts only one solution into the population at each generation, which is beneficial for selecting high quality solutions in the searing process of MOEAs [61]. On the other hand, the extreme case $b = \mathrm{Max\_FES}$ means that all offspring are reproduced based on the random initial population, which brings up offspring with low quality solutions.

The time cost (seconds) in Table I and Fig. 4(b) show that SMS-EMOA$^{\mathrm{WFG}}$ is much computationally expensive, and the other five algorithms remain at a relatively low level of time cost. Although both SMS-EMOA$^{\mathrm{WFG}}$ and FV-MOEA$^{\mathrm{WFG}}$ select offspring based on hypervolume contributions, the proposed fast hypervolume method in Alg. 1 enables FV-MOEA$^{\mathrm{WFG}}$ to quickly update hypervolume contributions of different solutions. In addition, the time cost of FV-MOEA$^{\mathrm{WFG}}$ decreases first and then increases as the batch size grows. By considering the hypervolume and time cost, the tradeoff of batch size in FV-MOEA$^{\mathrm{WFG}}$ is set as $b = 0.2 \times \mathrm{NP}$ and tested in the following sections.

D. Results of Low-dimensional MOPs

In this section, the five classical MOEAs (NSGAI-I [6], SPEA2 [7], MOEA/D [8], IBEA [15] and SMS-EMOA$^{\mathrm{WFG}}$ [16]) and the proposed FV-MOEA$^{\mathrm{WFG}}$ are tested on 30 low-dimensional MOPs ($d = 2, 3$) in terms of hypervolume and time cost.

Table III reports the detailed experimental results, where each tuple tabulates the median and IQR of hypervolume over 20 independent runs on 30 low-dimensional MOPs with maximum 15,000 function evaluations (FES). Table II shows the win/tie/lose (*w/t/l*) hypervolume statistical results under the Wilcoxon rank sum test with 95% confidence level. Each tuple *w/t/l* means that the algorithm at the corresponding column wins on w MOPs, ties on t MOPs, and loses on l MOPs, compared to the algorithm at the corresponding row.

In Table II, the *w/t/l* values of hypervolumes between FV-MOEA$^{\mathrm{WFG}}$ and NSGAII, SPEA2, MOEA/D are 24/4/2, 26/3/1, 25/2/3, respectively. In addition, both IBEA and SMS-EMOA$^{\mathrm{WFG}}$ report superior hypervolume results than NSGAII, SPEA2 and MOEA/D (as shown in Table II). These results indicate that the three hypervolume indicator-based algorithms (IBEA, SMS-EMOA$^{\mathrm{WFG}}$ and FV-MOEA$^{\mathrm{WFG}}$) are better than the Pareto dominance-based approaches (NSGAII and SPEA2) and the scalarizing function-based methods (MOEA/D) for solving low-dimensional MOPs, which is consistent with the results in [16, 37–41]. On the other hand, the *w/t/l* values of hypervolumes between SMS-EMOA$^{\mathrm{WFG}}$, FV-MOEA$^{\mathrm{WFG}}$ and IBEA are 23/5/2, 24/4/2, respectively. The reason is that the approximate hypervolume estimation in IBEA is insufficient compared to SMS-EMOA$^{\mathrm{WFG}}$ and FV-MOEA$^{\mathrm{WFG}}$ that are able to measure the exact hypervolume contributions for various solutions. In particular, the *w/t/l* values of hypervolumes between FV-MOEA$^{\mathrm{WFG}}$ and SMS-EMOA$^{\mathrm{WFG}}$ are 9/17/4. The mean hypervolumes of NSGAII, SPEA2, MOEA/D, IBEA, SMS-EMOA$^{\mathrm{WFG}}$ and FV-MOEA$^{\mathrm{WFG}}$ are 0.3223, 0.3439, 0.2813, 0.3132, 0.3623, 0.3694, respectively. The results in Tables II-V indicate that FV-MOEA$^{\mathrm{WFG}}$ is significantly better than the five classical MOEAs to obtain high hypervolumes on low-dimensional MOPs.

Tables IV shows the time cost statistical comparison results of six MOEAs over 20 independent runs on 30 low-dimensional MOPs, and Table V reports the detailed time cost results. From Tables IV, the *w/t/l* values between the hypervolume indicator-based algorithms (IBEA, SMS-EMOA$^{\mathrm{WFG}}$ and FV-MOEA$^{\mathrm{WFG}}$) and NSGAII, MOEA/D are 0/0/30. This means that the computational cost of hypervolume indicator-based algorithms is larger than other three classical MOEAs except SPEA2 on low-dimensional MOPs. Among the hypervolume indicator-based algorithms, the *w/t/l* values of time cost between FV-MOEA$^{\mathrm{WFG}}$ and IBEA, SMS-EMOA$^{\mathrm{WFG}}$ are 23/2/5, 30/0/0, respectively. The mean time cost of IBEA, SMS-EMOA$^{\mathrm{WFG}}$ and FV-MOEA$^{\mathrm{WFG}}$ are 0.622, 14.51, 0.437 seconds, respectively. It demonstrates that FV-MOEA$^{\mathrm{WFG}}$ is faster than IBEA and SMS-EMOA$^{\mathrm{WFG}}$ on low-dimensional MOPs. This is because the proposed fast hypervolume method in FV-MOEA$^{\mathrm{WFG}}$ significantly reduces the time cost for selecting offspring.

TABLE III
HYPERVOLUME MEDIAN AND IQR BY CLASSICAL MOEAS AND FV-MOEA$^{\text{WFG}}$ ON 30 LOW-DIMENSIONAL MOPS ($d = 2, 3$) OVER 20 INDEPENDENT RUNS WITH 15,000 FES.

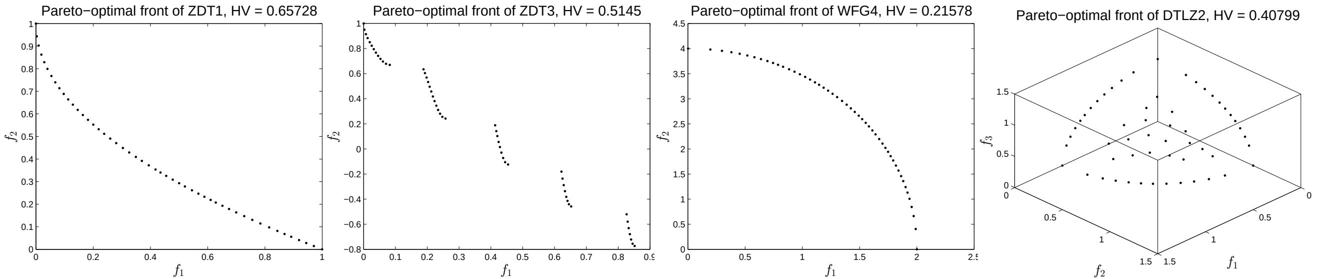
MOPs	NSGAII	SPEA2	MOEA/D	IBEA	SMS-EMOA$^{\text{WFG}}$	FV-MOEA$^{\text{WFG}}$
ZDT1	6.520E-01±8.05E-04 –	6.541E-01±1.00E-03 –	5.380E-01±5.82E-02 –	6.565E-01±2.95E-04 –	6.572E-01±5.57E-05≈	6.572E-01±7.27E-05
ZDT2	3.190E-01±1.54E-03 –	3.207E-01±1.23E-03 –	2.367E-01±2.52E-02 –	3.214E-01±4.67E-04 –	3.240E-01±1.27E-04 –	3.242E-01±4.73E-05
ZDT3	5.124E-01±3.42E-04 –	5.112E-01±1.04E-03 –	3.468E-01±1.01E-01 –	5.080E-01±4.10E-04 –	5.142E-01±2.26E-04 –	5.144E-01±3.11E-03
ZDT4	6.409E-01±1.60E-02 –	6.227E-01±4.07E-02 –	6.529E-02±2.10E-01 –	1.687E-01±1.59E-01 –	6.312E-01±3.60E-02 –	6.504E-01±7.14E-03
ZDT6	3.821E-01±4.14E-03 –	3.724E-01±5.80E-03 –	3.962E-01±1.71E-05 +	3.877E-01±1.13E-03 –	3.913E-01±1.52E-03 –	3.949E-01±3.09E-04
DTLZ1	8.871E-03±3.27E-01 –	6.823E-01±4.49E-01 –	6.749E-01±7.33E-01 –	2.818E-02±6.29E-02 –	7.638E-01±5.43E-03 –	7.662E-01±3.78E-03
DTLZ2	3.378E-01±9.60E-03 –	3.803E-01±3.54E-03 –	2.854E-01±1.88E-02 –	3.943E-01±1.40E-03 –	4.078E-01±6.69E-04≈	4.076E-01±2.81E-04
DTLZ3	0.000E-00±0.00E-00≈	0.000E-00±0.00E-00≈	0.000E-00±0.00E-00≈	0.000E-00±0.00E-00≈	0.000E-00±0.00E-00≈	0.000E-00±0.00E-00
DTLZ4	3.431E-01±9.80E-03≈	2.035E-01±1.65E-01 –	2.794E-01±4.76E-02≈	2.013E-01±2.02E-01 –	2.051E-01±5.14E-02≈	2.051E-01±1.96E-01
DTLZ5	8.977E-02±4.21E-04 –	9.070E-02±3.32E-04 –	7.096E-02±7.34E-04 –	8.867E-02±3.05E-04 –	9.202E-02±1.06E-04≈	9.199E-02±1.37E-04
DTLZ6	0.000E-00±0.00E-00 –	0.000E-00±0.00E-00 –	7.056E-02±7.74E-06 +	5.692E-02±1.33E-02 +	0.000E-00±0.00E-00 –	3.069E-02±2.61E-02
DTLZ7	2.530E-01±8.17E-03 +	2.680E-01±1.53E-02 +	8.947E-02±1.07E-01 –	2.001E-01±6.97E-02≈	1.986E-01±4.67E-02 +	1.861E-01±1.34E-02
WFG1	3.459E-01±1.92E-01 +	2.944E-01±1.77E-01≈	1.095E-01±3.83E-03 –	2.645E-01±2.26E-01≈	1.886E-01±1.15E-01 –	2.756E-01±1.26E-01
WFG2	5.597E-01±1.27E-03≈	5.595E-01±8.66E-04 –	5.445E-01±4.20E-03 –	5.562E-01±6.71E-04 –	5.606E-01±2.42E-03≈	5.604E-01±1.98E-03
WFG3	4.365E-01±1.05E-03 –	4.382E-01±5.56E-04 –	4.369E-01±5.01E-04 –	4.390E-01±2.64E-04 –	4.395E-01±2.49E-04≈	4.395E-01±2.79E-04
WFG4	2.114E-01±1.32E-03 –	2.126E-01±5.46E-04 –	1.876E-01±3.70E-03 –	2.116E-01±6.03E-04 –	2.157E-01±1.60E-04 +	2.156E-01±1.89E-04
WFG5	1.891E-01±1.49E-03 –	1.912E-01±4.22E-04 –	1.887E-01±5.51E-04 –	1.909E-01±1.95E-04 –	1.932E-01±8.79E-05 +	1.931E-01±7.95E-05
WFG6	1.904E-01±1.35E-02≈	1.924E-01±2.34E-02≈	1.974E-01±1.59E-03 +	1.950E-01±8.76E-03≈	1.898E-01±2.71E-02≈	1.930E-01±2.68E-02
WFG7	2.026E-01±9.87E-04 –	2.046E-01±8.22E-04 –	1.986E-01±6.97E-04 –	2.034E-01±5.90E-04 –	2.074E-01±4.59E-05 +	2.074E-01±6.48E-05
WFG8	1.400E-01±2.56E-03 –	1.433E-01±1.26E-03 –	1.387E-01±3.09E-03 –	1.391E-01±2.65E-03 –	1.458E-01±1.90E-03≈	1.462E-01±2.06E-03
WFG9	2.265E-01±3.48E-03 –	2.295E-01±3.11E-03 –	2.247E-01±1.04E-03 –	2.332E-01±1.06E-03 –	2.350E-01±3.78E-03≈	2.360E-01±1.72E-03
WFG1-3D	6.285E-01±9.56E-02 –	5.632E-01±1.35E-01 –	2.149E-01±6.80E-03 –	5.574E-01±1.37E-01 –	8.164E-01±5.66E-02≈	8.923E-01±1.25E-01
WFG2-3D	8.742E-01±8.16E-03 –	8.951E-01±3.36E-03 –	8.737E-01±6.08E-03 –	8.892E-01±7.83E-03 –	9.194E-01±1.21E-03≈	9.197E-01±1.08E-03
WFG3-3D	2.960E-01±4.22E-03 –	2.768E-01±1.12E-02 –	2.319E-01±1.05E-02 –	3.229E-01±1.04E-03 +	3.206E-01±8.56E-04≈	3.209E-01±8.64E-04
WFG4-3D	3.346E-01±1.48E-02 –	3.619E-01±8.92E-03 –	3.121E-01±9.72E-03 –	3.936E-01±1.20E-03 –	4.058E-01±5.47E-04 –	4.065E-01±6.80E-04
WFG5-3D	3.098E-01±9.21E-03 –	3.435E-01±3.63E-03 –	3.258E-01±3.97E-03 –	3.582E-01±1.09E-03 –	3.704E-01±7.83E-04≈	3.704E-01±5.33E-04
WFG6-3D	3.255E-01±2.75E-02 –	3.687E-01±1.73E-02 –	3.305E-01±8.42E-03 –	3.840E-01±2.56E-02 –	3.951E-01±3.59E-02≈	3.941E-01±2.13E-02
WFG7-3D	3.218E-01±1.40E-02 –	3.504E-01±7.86E-03 –	3.264E-01±1.11E-02 –	3.897E-01±1.20E-03 –	4.026E-01±9.65E-04 –	4.034E-01±5.51E-04
WFG8-3D	2.157E-01±1.10E-02 –	2.353E-01±6.87E-03 –	2.011E-01±1.65E-02 –	2.687E-01±8.25E-03 –	2.851E-01±5.32E-03≈	2.862E-01±9.15E-03
WFG9-3D	3.218E-01±9.31E-03 –	3.493E-01±4.33E-03 –	3.410E-01±5.27E-03 –	3.869E-01±2.96E-03 –	3.930E-01±4.61E-03≈	3.937E-01±2.75E-03
Mean	3.223E-01±2.63E-02	3.439E-01±3.63E-02	2.813E-01±4.66E-02	3.132E-01±3.13E-02	3.623E-01±1.33E-02	3.694E-01±1.90E-02

+ , ≈ and – represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively.

TABLE V
TIME MEDIAN AND IQR BY CLASSICAL MOEAS AND FV-MOEA$^{\text{WFG}}$ ON 30 LOW-DIMENSIONAL MOPS ($d = 2, 3$) OVER 20 INDEPENDENT RUNS WITH 15,000 FES.

MOPs	NSGAII	SPEA2	MOEA/D	IBEA	SMS-EMOA$^{\text{WFG}}$	FV-MOEA$^{\text{WFG}}$
ZDT1	1.420E-01±1.00E-03+	5.325E-01±6.25E-03 –	8.550E-02±7.75E-03+	6.225E-01±1.02E-02 –	3.272E-00±6.07E-01 –	3.045E-01±6.63E-02
ZDT2	1.435E-01±5.75E-03+	4.960E-01±2.10E-02 –	7.900E-02±4.50E-03+	6.230E-01±1.48E-02 –	2.626E-00±5.02E-01 –	2.770E-01±3.42E-02
ZDT3	1.430E-01±1.50E-03+	5.340E-01±7.25E-03 –	8.450E-02±2.25E-03+	6.280E-01±6.75E-03 –	3.115E-00±7.63E-01 –	2.910E-01±3.50E-02
ZDT4	8.700E-02±1.00E-03+	3.365E-01±1.15E-02 –	7.050E-02±6.75E-03+	4.970E-01±6.50E-03 –	1.383E-00±3.57E-01 –	1.875E-01±4.62E-02
ZDT6	8.500E-02±3.75E-03+	3.535E-01±8.75E-03 –	6.600E-02±2.25E-03+	5.100E-01±9.00E-03 –	2.080E-00±3.88E-01 –	2.020E-01±2.90E-02
DTLZ1	8.100E-02±2.00E-03+	4.885E-01±1.45E-02 +	6.600E-02±1.25E-03+	6.765E-01±1.25E-02 –	1.336E+01±2.73E-00 –	5.320E-01±5.43E-02
DTLZ2	9.700E-02±1.25E-03+	7.400E-01±6.50E-03≈	7.100E-02±2.00E-03+	7.235E-01±1.15E-02 +	3.247E+01±7.35E-00 –	7.635E-01±6.78E-02
DTLZ3	1.030E-01±2.00E-03+	4.760E-01±1.92E-02 –	7.700E-02±3.50E-03+	7.255E-01±1.10E-02 –	3.454E-00±7.80E-01 –	3.115E-01±8.05E-02
DTLZ4	1.110E-01±1.00E-03+	5.830E-01±1.69E-01≈	8.300E-02±2.00E-03+	7.270E-01±6.20E-02 –	1.729E+01±1.69E+01 –	3.980E-01±3.94E-01
DTLZ5	8.850E-02±1.00E-03+	5.930E-01±4.50E-03 –	7.100E-02±5.00E-04+	6.995E-01±1.12E-02 –	1.579E+01±3.34E-00 –	3.245E-01±4.83E-02
DTLZ6	1.240E-01±2.50E-04+	6.425E-01±1.02E-02 –	7.500E-02±2.00E-03+	7.215E-01±1.52E-02 –	1.363E+01±1.80E-00 –	5.385E-01±1.02E-01
DTLZ7	1.310E-01±2.00E-03+	7.130E-01±1.20E-02 –	7.800E-02±4.00E-03+	7.800E-01±9.50E-03 –	2.116E+01±4.22E-00 –	6.235E-01±1.53E-01
WFG1	8.400E-02±2.50E-03+	4.280E-01±1.83E-02 –	8.100E-02±1.25E-03+	5.080E-01±1.00E-02 –	3.655E-00±5.82E-01 –	2.470E-01±7.15E-02
WFG2	6.900E-02±1.00E-03+	4.015E-01±1.27E-02 –	6.700E-02±2.00E-03+	4.960E-01±8.25E-03 –	2.952E-00±9.37E-01 –	2.250E-01±8.77E-02
WFG3	6.500E-02±0.00E-00+	4.650E-01±1.02E-02 –	6.500E-02±4.75E-03+	4.830E-01±1.05E-02 –	4.667E-00±9.20E-01 –	2.565E-01±5.87E-02
WFG4	7.300E-02±2.50E-04+	4.610E-01±1.00E-02 –	7.200E-02±3.00E-03+	4.915E-01±5.75E-03 –	4.947E-00±1.23E-00 –	2.620E-01±3.40E-02
WFG5	6.600E-02±1.00E-03+	5.230E-01±1.20E-02 –	6.500E-02±3.00E-03+	4.820E-01±3.25E-03 –	6.559E-00±1.91E-00 –	3.100E-01±6.05E-02
WFG6	6.600E-02±2.50E-04+	4.445E-01±8.25E-03 –	6.500E-02±0.00E-00+	4.865E-01±8.50E-03 –	4.441E-00±6.57E-01 –	2.615E-01±5.85E-02
WFG7	7.400E-02±2.50E-04+	4.890E-01±7.00E-03 –	7.400E-02±2.25E-03+	4.925E-01±1.20E-02 –	5.491E-00±7.28E-01 –	2.790E-01±4.52E-02
WFG8	8.300E-02±1.00E-03+	3.910E-01±7.00E-03 –	8.000E-02±1.00E-03+	5.045E-01±8.25E-03 –	1.557E-00±4.26E-01 –	1.930E-01±3.03E-02
WFG9	9.100E-02±0.00E-00+	5.135E-01±5.75E-03 –	9.000E-02±1.00E-03+	5.080E-01±8.50E-03 –	5.351E-00±7.22E-01 –	2.900E-01±7.82E-02
WFG1-3D	9.800E-02±1.00E-03+	6.130E-01±2.10E-02≈	8.800E-02±3.00E-03+	7.170E-01±9.00E-03 –	1.938E+01±2.54E-00 –	6.200E-01±6.25E-02
WFG2-3D	8.300E-02±1.00E-03+	6.305E-01±1.15E-02 +	7.200E-02±3.50E-03+	6.895E-01±1.12E-02≈	2.566E+01±3.35E-00 –	6.635E-01±7.37E-02
WFG3-3D	7.600E-02±1.00E-03+	7.820E-01±1.25E-02 –	6.800E-02±6.00E-03+	6.705E-01±7.75E-03 –	2.079E+01±3.84E-00 –	4.415E-01±9.75E-02
WFG4-3D	9.000E-02±5.00E-04+	7.450E-01±7.00E-03≈	7.800E-02±3.25E-03+	6.980E-01±6.50E-03 +	3.554E+01±6.71E-00 –	7.625E-01±7.80E-02
WFG5-3D	8.300E-02±2.50E-04+	7.510E-01±1.45E-02≈	7.000E-02±2.25E-03+	6.910E-01±7.25E-03 +	3.749E+01±4.58E-00 –	7.520E-01±9.78E-02
WFG6-3D	8.200E-02±1.00E-03+	6.800E-01±9.75E-03≈	7.100E-02±1.25E-03+	6.890E-01±1.22E-02≈	2.748E+01±4.32E-00 –	6.710E-01±7.02E-02
WFG7-3D	9.100E-02±1.25E-03+	8.185E-01±1.57E-02≈	8.000E-02±4.25E-03+	6.950E-01±1.23E-02 +	4.254E+01±1.22E+01 –	8.110E-01±6.85E-02
WFG8-3D	9.700E-02±1.00E-03+	6.220E-01±2.92E-02 –	8.700E-02±5.50E-03+	7.050E-01±6.25E-03 –	1.386E+01±2.00E-00 –	5.265E-01±6.65E-02
WFG9-3D	1.090E-01±1.25E-03+	8.935E-01±2.67E-02 –	9.600E-02±1.25E-03+	7.180E-01±1.10E-02 +	4.335E+01±1.13E+01 –	7.845E-01±8.65E-02
Mean	9.387E-02±1.23E-03	5.714E-01±1.77E-02	7.585E-02±2.91E-03	6.220E-01±1.13E-02	1.451E+01±3.29E-00	4.370E-01±7.79E-02

+ , ≈ and – represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively.



(a) ZDT1 (b) ZDT3 (c) WFG4 (d) DTLZ2

Fig. 5. The best Pareto-optimal fronts on ZDT1, ZDT3, WFG4, DTLZ2 derived by FV-MOEA$^{\text{WFG}}$ over 20 independent runs.

TABLE VI
HYPERVOLUME STATISTICAL RESULTS BY CLASSICAL MOEAS AND FV-MOEA$^{\text{WFG}}$ ON 14 HIGH-DIMENSIONAL MOPS ($d = 4, 5$).

	SPEA2	MOEA/D	IBEA	SMS-EMOA$^{\text{WFG}}$	FV-MOEA$^{\text{WFG}}$
NSGAII	2 7 5	9 1 4	9 2 3	11 3 0	12 2 0
SPEA2		11 0 3	12 1 1	12 1 1	12 1 1
MOEAD			7 3 4	7 4 3	9 3 2
IBEA				6 3 5	7 5 2
SMS-EMOA$^{\text{WFG}}$					6 6 2

TABLE VIII
TIME STATISTICAL RESULTS BY CLASSICAL MOEAS AND FV-MOEA$^{\text{WFG}}$ ON 14 HIGH-DIMENSIONAL MOPS ($d = 4, 5$).

	SPEA2	MOEA/D	IBEA	SMS-EMOA$^{\text{WFG}}$	FV-MOEA$^{\text{WFG}}$
NSGAII	0 0 14	14 0 0	0 0 14	0 0 14	0 0 14
SPEA2		14 0 0	4 0 10	0 0 14	1 1 12
MOEAD			0 0 14	0 0 14	0 0 14
IBEA				0 0 14	2 0 12
SMS-EMOA$^{\text{WFG}}$					14 0 0

To verify the quality of optimal solution sets, Fig. 5 shows the best Pareto-optimal fronts found by FV-MOEA$^{\text{WFG}}$ over 20 runs on the four representative MOPs (ZDT1, ZDT3, WFG4 and DTLZ2). For the discrete Pareto front (PF) of ZDT3, FV-MOEA$^{\text{WFG}}$ arrives at the highest hypervolume among the six algorithms (see Table III) and obtains well distributed solutions in Fig. 5(b). The result of ZDT1 in Fig. 5(a) shows that solutions are scattered to the two extreme objective values on the convex PF, which is consistent with the results in [11, 12, 16]. On the other hand, Figs. 5(c-d) show that solutions are assembled to the central region on concave PFs (WFG4 and DTLZ2), which also shares the same conclusion in [11, 12, 16]. These results indicate that FV-MOEA$^{\text{WFG}}$ is able to deal with PFs with different geometrical characteristics.

E. Results of High-dimensional MOPs

To further evaluate FV-MOEA$^{\text{WFG}}$, in this section, 14 high-dimensional MOPs are selected as testing instances. Tables VI-IX summarize the median and IQR results of the five classical MOEAs and the proposed FV-MOEA$^{\text{WFG}}$ over 20 independent runs on 14 high-dimensional MOPs ($d = 4, 5$) with 15,000 FES in terms of hypervolume and time cost.

From Table VI, the *w/t/l* values of hypervolumes between MOEA/D and NSGAII, SPEA2 are 9/1/4, 11/0/3, respectively. It indicates that the scalarizing function-based method (MOEA/D) is more suitable to deal with high-dimensional MOPs than the two Pareto dominance-based approaches (NSGAII and SPEA2). This is consistent with the studies in [8, 9, 13, 14]. In addition, as shown in Table VI, the indicator-based algorithms (IBEA, SMS-EMOA$^{\text{WFG}}$ and FV-MOEA$^{\text{WFG}}$) arrive at higher hypervolumes than the other three MOEAs (NSGAII, SPEA2 and MOEA/D). On the other hand, the *w/t/l* values of hypervolumes between SMS-EMOA$^{\text{WFG}}$, FV-MOEA$^{\text{WFG}}$ and IBEA are 6/3/5, 7/5/2, respectively. These results are similar to those of low-dimensional MOPs in Tables II-III. The reason is that IBEA is not designed to measure the exact hypervolume contributions of different solutions. Furthermore, the *w/t/l* values of hypervolumes between FV-MOEA$^{\text{WFG}}$ and IBEA, SMS-EMOA$^{\text{WFG}}$ are 7/5/2, 6/6/2, respectively. These results indicate that FV-MOEA$^{\text{WFG}}$ is the best among the three hypervolume indicator-based algorithms for solving high-dimensional MOPs.

In Table VIII, the *w/t/l* values of time cost between FV-MOEA$^{\text{WFG}}$ and IBEA, SMS-EMOA$^{\text{WFG}}$ are 2/0/12, 14/0/0, respectively. The mean time cost of IBEA, SMS-EMOA$^{\text{WFG}}$ and FV-MOEA$^{\text{WFG}}$ in Table IX are 1.196, 214.3 and 3.423 seconds, respectively. This indicates that SMS-EMOA$^{\text{WFG}}$ needs more computational resource than the other two hypervolume indicator-based approaches, whereas the time cost of IBEA and FV-MOEA$^{\text{WFG}}$ remain at a relatively lower level.

F. Influence of Population Size and MOPs Dimension

Among hypervolume indicator-based algorithms, IBEA only obtains approximated hypervolume estimations, whereas SMS-EMOA$^{\text{WFG}}$ and FV-MOEA$^{\text{WFG}}$ are able to find exact hypervolume contributions for different solutions. Due to this reason, in this section, we only compare SMS-EMOA$^{\text{WFG}}$ and FV-MOEA$^{\text{WFG}}$ with different population size and dimensional settings on DTLZ2 problems. The maximum function evaluations is Max_FES $= 20 \times$ NP and the independent runs is set as Runs $= 5$.

Fig. 6 reports the hypervolume and time cost by SMS-EMOA$^{\text{WFG}}$ and FV-MOEA$^{\text{WFG}}$ with population size NP $= \{25, 50, 75, 100, 125, 150\}$ on $\{2, 3, 4, 5\}$–dimensional DTLZ2 over 5 independent runs. The results in Figs. 6(a-d) show that both algorithms arrive at the similar hypervolumes on different testing scenarios. On the other hand, Figs. 6(e-h) report that the time cost of SMS-EMOA$^{\text{WFG}}$ becomes significantly larger than those of FV-MOEA$^{\text{WFG}}$ as the population size grows.

TABLE VII
HYPERVOLUME MEDIAN AND IQR BY CLASSICAL MOEAS AND FV-MOEA$^{\text{WFG}}$ ON 14 HIGH-DIMENSIONAL MOPS ($d = 4, 5$) OVER 20 INDEPENDENT RUNS WITH 15,000 FES.

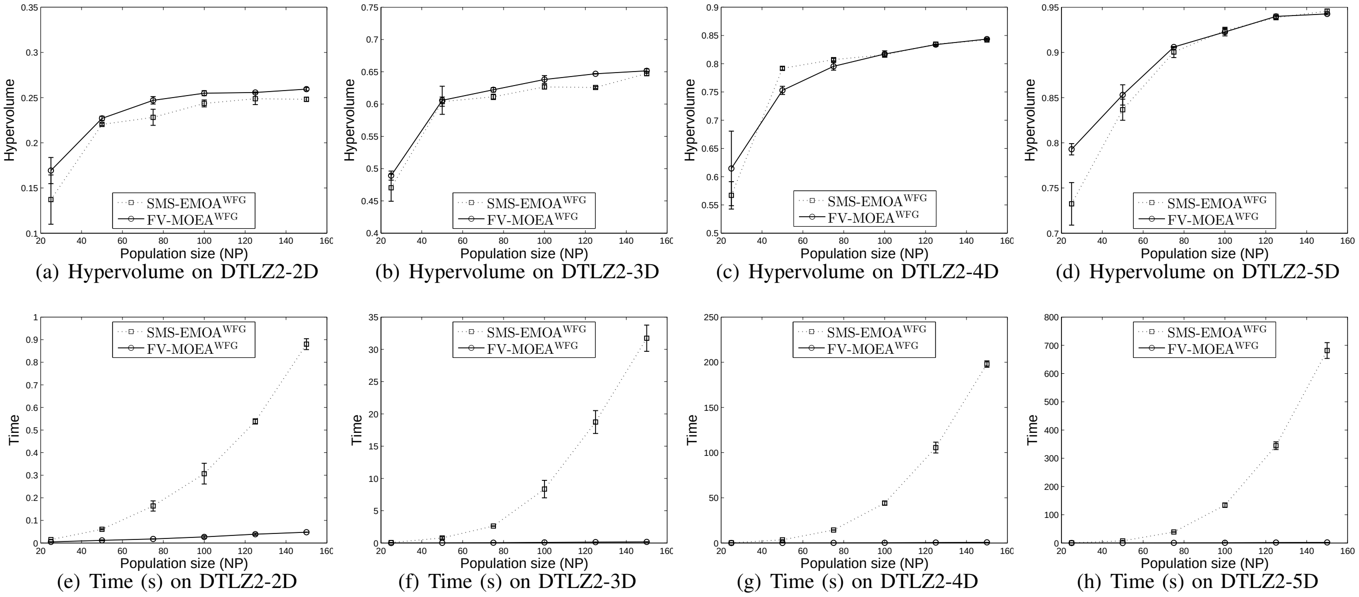
MOPs	NSGAII	SPEA2	MOEA/D	IBEA	SMS-EMOA$^{\text{WFG}}$	FV-MOEA$^{\text{WFG}}$
DTLZ1D4	0.000E-00±0.00E-00 –	0.000E-00±0.00E-00 –	8.226E-01±1.47E-01 –	1.097E-01±1.70E-01 –	8.866E-01±4.22E-03≈	8.880E-01±1.31E-03
DTLZ2D4	5.885E-01±1.69E-02 –	6.262E-01±1.71E-02 –	4.364E-01±4.96E-02 –	7.004E-01±8.11E-04 –	7.093E-01±4.38E-04 –	7.097E-01±2.30E-04
DTLZ3D4	7.846E-01±1.17E-01 –	6.207E-01±7.21E-01 –	9.772E-01±6.49E-02 –	9.957E-01±8.62E-04≈	9.936E-01±1.19E-03≈	9.939E-01±3.48E-03
DTLZ4D4	4.879E-01±2.00E-02≈	2.929E-01±2.26E-01≈	5.062E-01±3.44E-02≈	4.575E-01±2.00E-01≈	4.745E-01±2.42E-01≈	3.713E-01±2.89E-01
DTLZ5D4	7.832E-01±2.67E-03 –	7.547E-01±1.77E-02 –	7.773E-01±1.45E-03 –	7.765E-01±6.05E-03 –	7.939E-01±6.37E-04 –	7.942E-01±3.44E-04
DTLZ6D4	3.840E-01±6.39E-02 –	3.884E-01±1.02E-01 –	9.292E-01±1.23E-03 +	9.371E-01±2.18E-03 +	8.888E-01±7.65E-03 –	9.157E-01±6.98E-03
DTLZ7D4	2.905E-01±1.68E-02≈	3.163E-01±1.50E-02 +	6.531E-02±4.42E-02 –	2.586E-01±9.58E-02≈	2.503E-01±6.48E-02≈	2.506E-01±6.47E-02
DTLZ1D5	0.000E-00±0.00E-00 –	0.000E-00±0.00E-00 –	9.264E-01±7.55E-03 +	8.359E-02±1.52E-01 –	8.917E-01±1.59E-02 +	8.770E-01±2.02E-02
DTLZ2D5	6.436E-01±7.44E-02 –	6.308E-01±9.62E-02 –	6.929E-01±4.10E-02 –	8.852E-01±3.63E-04 –	8.888E-01±3.49E-04 –	8.893E-01±3.63E-04
DTLZ3D5	0.000E-00±0.00E-00 –	0.000E-00±0.00E-00 –	9.992E-01±6.07E-01≈	9.994E-01±2.47E-03≈	9.974E-01±5.97E-03 –	9.994E-01±2.19E-03
DTLZ4D5	8.916E-01±7.12E-02 –	8.609E-01±9.13E-02 –	9.686E-01±1.67E-03≈	9.348E-01±8.17E-02 –	9.294E-01±3.88E-01≈	9.497E-01±3.20E-02
DTLZ5D5	7.759E-01±6.73E-03 –	6.946E-01±2.97E-02 –	7.821E-01±3.34E-03 –	7.735E-01±3.21E-02 –	8.050E-01±5.59E-04 +	8.041E-01±1.13E-03
DTLZ6D5	2.299E-01±9.13E-02 –	1.532E-01±6.30E-02 –	9.397E-01±1.24E-02 –	9.482E-01±1.87E-02≈	9.322E-01±6.28E-03 –	9.517E-01±4.06E-03
DTLZ7D5	3.648E-01±3.84E-02 –	3.375E-01±4.19E-02 –	7.287E-03±2.30E-02 –	4.216E-01±2.18E-02 +	3.981E-01±5.06E-02≈	3.920E-01±1.03E-02
Mean	4.446E-01±3.71E-02	4.055E-01±1.01E-01	7.022E-01±7.41E-02	6.630E-01±5.61E-02	7.743E-01±5.63E-02	7.705E-01±3.12E-02

+ , ≈ and – represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively.

TABLE IX
TIME MEDIAN AND IQR BY CLASSICAL MOEAS AND FV-MOEA$^{\text{WFG}}$ ON 14 HIGH-DIMENSIONAL MOPS ($d = 4, 5$) OVER 20 INDEPENDENT RUNS WITH 15,000 FES.

MOPs	NSGAII	SPEA2	MOEA/D	IBEA	SMS-EMOA$^{\text{WFG}}$	FV-MOEA$^{\text{WFG}}$
DTLZ1D4	9.200E-02±6.00E-03+	6.355E-01±5.20E-02 +	6.900E-02±1.25E-03+	8.250E-01±2.67E-02+	6.745E+01±4.31E-00 –	2.027E-00±1.07E-01
DTLZ2D4	1.130E-01±1.27E-02+	9.215E-01±4.77E-02 +	8.000E-02±1.22E-02+	8.645E-01±2.00E-02+	1.223E+02±3.50E-00 –	2.479E-00±9.07E-02
DTLZ3D4	1.165E-01±1.32E-02+	6.765E-01±2.32E-02 +	8.300E-02±1.32E-02+	8.735E-01±2.95E-02+	3.518E+01±4.31E-00 –	1.548E-00±1.31E-01
DTLZ4D4	1.380E-01±9.75E-03+	7.960E-01±2.05E-01≈	1.005E-01±8.50E-03+	9.035E-01±2.12E-02–	4.031E+01±3.04E+01–	6.495E-01±5.57E-01
DTLZ5D4	1.090E-01±9.50E-03+	9.610E-01±6.13E-02 –	7.750E-02±2.25E-03+	8.550E-01±3.80E-02–	2.457E+01±1.92E-00 –	6.405E-01±4.50E-02
DTLZ6D4	1.375E-01±1.25E-02+	9.460E-01±7.38E-02 +	8.100E-02±7.75E-03+	8.900E-01±4.83E-02+	5.746E+01±1.79E-00 –	1.341E-00±5.45E-02
DTLZ7D4	1.450E-01±1.33E-02+	9.025E-01±4.20E-02 +	8.100E-02±3.75E-03+	9.350E-01±4.80E-02+	6.068E+01±6.74E-00 –	1.409E-00±9.98E-02
DTLZ1D5	1.130E-01±8.25E-03+	1.006E-00±4.27E-02 +	7.400E-02±1.20E-02+	1.470E-00±5.20E-02+	4.303E+02±3.78E+01–	5.890E-00±1.23E-00
DTLZ2D5	1.395E-01±8.00E-03+	1.262E-00±7.75E-02 +	8.600E-02±4.75E-03+	1.512E-00±9.05E-02+	8.997E+02±5.02E+01–	1.182E+01±3.30E-01
DTLZ3D5	1.420E-01±4.50E-03+	1.102E-00±6.40E-02 +	9.000E-02±8.00E-03+	1.510E-00±1.06E-01+	2.799E+02±2.71E+01–	6.248E-00±9.30E-01
DTLZ4D5	1.810E-01±1.08E-02+	1.329E-00±7.58E-02 +	1.180E-01±7.75E-03+	1.535E-00±6.53E-02+	1.879E+02±4.25E+02–	2.502E-00±4.82E-00
DTLZ5D5	1.340E-01±4.50E-03+	1.353E-00±3.63E-02 +	8.400E-02±7.75E-03+	1.493E-00±5.77E-02+	1.329E+02±2.19E+01–	2.578E-00±1.54E-01
DTLZ6D5	1.600E-01±5.25E-03+	1.541E-00±4.63E-02 +	8.900E-02±1.23E-02+	1.508E-00±3.37E-02+	3.570E+02±1.42E+01–	4.529E-00±2.60E-01
DTLZ7D5	1.715E-01±1.13E-02+	1.246E-00±2.97E-02 +	8.650E-02±6.50E-03+	1.568E-00±6.57E-02+	3.045E+02±1.97E+01–	4.263E-00±3.48E-01
Mean	1.351E-01±9.25E-03	1.048E-00±6.27E-02	8.568E-02±7.71E-03	1.196E-00±5.02E-02	2.143E+02±4.63E+01	3.423E-00±6.54E-01

+ , ≈ and – represent previous algorithm statistically significant better, similar and worse than the last algorithm, respectively.



(a) Hypervolume on DTLZ2-2D (b) Hypervolume on DTLZ2-3D (c) Hypervolume on DTLZ2-4D (d) Hypervolume on DTLZ2-5D

(e) Time (s) on DTLZ2-2D (f) Time (s) on DTLZ2-3D (g) Time (s) on DTLZ2-4D (h) Time (s) on DTLZ2-5D

Fig. 6. The median and IQR of **Hypervolume** (a-d) and **Time** (e-h) derived by SMS-EMOA$^{\text{WFG}}$ and FV-MOEA$^{\text{WFG}}$ with population size NP $= \{25, 50, 75, 100, 125, 150\}$ on $\{2, 3, 4, 5\}$–dimensional DTLZ2 over 5 independent runs.

TABLE X
TIME MEDIAN AND IQR BY SMS-EMOA$^{\text{WFG}}$ AND FV-MOEA$^{\text{WFG}}$ WITH POPULATION SIZE NP $= \{25, 50, 75, 100, 125, 150\}$ ON $\{2, 3, 4, 5\}$–DIMENSIONAL DTLZ2 OVER 5 INDEPENDENT RUNS.

		SMS-EMOA$^{\text{WFG}}$	FV-MOEA$^{\text{WFG}}$	$\frac{\text{SMS-EMOA}^{\text{WFG}}}{\text{FV-MOEA}^{\text{WFG}}}$
DTLZ2-2D	NP = 25	1.500E-02±0.00E-00	5.000E-03±1.00E-03	3.0
	NP = 50	6.100E-02±5.00E-03	1.200E-02±4.00E-03	5.1
	NP = 75	1.640E-01±4.50E-02	1.800E-02±2.00E-03	9.1
	NP = 100	3.070E-01±9.20E-02	2.700E-02±1.20E-02	11.4
	NP = 125	5.380E-01±2.30E-02	3.900E-02±1.00E-02	13.8
	NP = 150	8.800E-01±4.80E-02	4.800E-02±1.00E-03	18.3
DTLZ2-3D	NP = 25	7.100E-02±6.00E-03	9.000E-03±2.00E-03	7.9
	NP = 50	7.880E-01±3.44E-01	2.900E-02±3.00E-03	27.2
	NP = 75	2.630E-00±6.20E-02	5.600E-02±1.10E-02	47.0
	NP = 100	8.354E-00±2.71E-00	9.100E-02±1.20E-02	91.8
	NP = 125	1.874E+01±3.55E-00	1.380E-01±1.20E-02	135.8
	NP = 150	3.172E+01±4.07E-00	1.800E-01±5.10E-02	176.2
DTLZ2-4D	NP = 25	3.040E-01±6.50E-02	2.800E-02±1.10E-02	10.9
	NP = 50	3.551E-00±2.54E-01	9.400E-02±5.00E-03	37.8
	NP = 75	1.442E+01±2.92E-01	2.350E-01±2.20E-02	61.4
	NP = 100	4.417E+01±4.45E-00	3.120E-01±1.40E-02	141.6
	NP = 125	1.056E+02±1.20E+01	5.510E-01±7.20E-02	191.7
	NP = 150	1.980E+02±7.03E-00	7.370E-01±5.70E-02	268.7
DTLZ2-5D	NP = 25	7.020E-01±2.32E-01	4.900E-02±1.40E-02	14.3
	NP = 50	7.664E-00±7.47E-01	2.480E-01±2.50E-02	30.9
	NP = 75	3.916E+01±1.48E-00	5.860E-01±2.30E-02	66.8
	NP = 100	1.340E+02±1.54E+01	1.016E-00±7.90E-02	131.9
	NP = 125	3.446E+02±2.74E+01	1.745E-00±8.00E-02	197.5
	NP = 150	6.814E+02±5.62E+01	2.244E-00±1.71E-01	303.7

Table X shows the detailed time cost of two algorithms on various testing scenarios. The last column calculates the ratio of time cost between SMS-EMOA$^{\text{WFG}}$ and FV-MOEA$^{\text{WFG}}$. As shown in Table X, the ratio increases as the population size grows. In particular, the ratio increases from 14.3 to 303.7 on 5–dimensional DTLZ2 when the population size grows from 25 to 150. Other similar results can be found on DTLZ2 with 2–4 objectives. In addition, the ratio becomes large as the MOPs dimension increases. For instance, the ratios for NP = 150 are found as 18.3, 176.2, 268.7, 303.7 on 2, 3, 4, 5–dimensional DTLZ2 problems, respectively.

In summary, the results of Fig. 6 and Table X show that FV-MOEA$^{\text{WFG}}$ is able to find competitive hypervolumes compared to SMS-EMOA$^{\text{WFG}}$. The interesting thing is that FV-MOEA$^{\text{WFG}}$ is able to obtain significant speedup compared to SMS-EMOA$^{\text{WFG}}$ when the population size increases as well as the dimension of MOPs grows.

V. CONCLUSION AND FUTURE WORK

To find high quality of solutions in indicator-based multi-objective evolutionary algorithms (MOEAs), hypervolume is a critical performance metric to perform solution selection. However, the high time complexity of calculating exact hypervolume contributions is cumbersome for applying it to high-dimensional multiobjective optimization problems (MOPs). In this paper, a simple and **F**ast hyper**V**olume indicator-based MOEA (FV-MOEA) is proposed to quickly update exact hypervolume contributions for different solutions. The core idea of FV-MOEA is that the hypervolume contribution of a solution is only associated with partial solutions rather than the whole solution set. Experimental studies on 44 benchmark MOPs with 2–5 objectives on jMetal confirm its superior performance over the five classical MOEAs (NSGAII, SPEA2, MOEA/D, IBEA and SMS-EMOA) in terms of hypervolume and time cost.

In future, we plan to combine the ideas from different groups of MOEAs (*i.e.*, fusion of scalarizing function-based methods and hypervolume indicator-based approaches) for solving MOPs with nonlinear Pareto sets (PSs). In particular, the scalarizing function-based methods concentrate on converging solutions along predefined weight vectors. With the help of the fast hypervolume method proposed in this paper, the fused approach is able to jump out of local Pareto-optimal fronts and obtain highly diversified solutions converging to the true Pareto fronts (PFs). On the other hand, to enhance the search ability of MOEAs, we will adapt the choice of selecting various evolutionary operators and adjusting control parameters. Last but not least, we will extend the proposed fast hypervolume method to surrogated-assisted MOEAs, and test their performance on dealing with computationally expensive MOPs.

The Java source codes of the proposed FV-MOEA is available at http://trust.sce.ntu.edu.sg/~sjiang1/.

REFERENCES

[1] Coello C.C.A. Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36, 2006.

[2] H. Abbass, A. Bender, S. Gaidow, and P. Whitbread. Computational red teaming: Past, present and future. *IEEE Computational Intelligence Magazine*, 6(1):30–42, 2011.

[3] A. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 2011.

[4] J.J. Durillo, A.J. Nebro, and E. Alba. The jmetal framework for multi-objective optimization: Design and architecture. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.

[5] J.J. Durillo and A.J. Nebro. jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[7] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. *Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland*, (103), 2001.

[8] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[9] H. Li and Q. Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.

[10] E.J. Hughes. MSOPS-II: A general-purpose many-objective optimiser. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 3944–3951, 2007.

[11] S. Jiang, J. Zhang, and Y.S. Ong. Asymmetric pareto-adaptive scheme for multiobjective optimization. In *Proceedings of Australasian Joint Conference on Artificial Intelligence (AI)*, pages 351–360. Springer, 2011.

[12] S. Jiang, J. Zhang, and YewSoon. Ong. Multiobjective optimization by decomposition with pareto-adaptive weight vectors. In *Proceedings of IEEE Conference on Natural Computation (ICNC)*, volume 3, pages 1260–1264, 2011.

[13] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima. Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 530–537, 2009.

[14] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima. Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, pages 527–534, 2010.

[15] E. Zitzler and S. Kúnzli. Indicator-based selection in multiobjective search. In *Proceedings of Parallel Problem Solving from Nature (PPSN)*, pages 832–842. Springer, 2004.

[16] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[17] S. Jiang, J. Zhang, and Y.S. Ong. A multiagent evolutionary framework based on trust for multiobjective optimization. In *Proceedings of Autonomous Agents and Multiagent Systems (AAMAS)*, pages 299–306, 2012.

[18] C.W. Seah, Y.S. Ong, I.W. Tsang, and S. Jiang. Pareto rank learning in multi-objective evolutionary algorithms. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2012.

[19] G.G. Yen and W.F. Leong. Dynamic multiple swarms in multiobjective particle swarm optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(4):890–911, 2009.

[20] 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*, 34(3):315–326, 2004.

[21] J.B. Yang, J. Liu, D.L. Xu, J. Wang, and H. Wang. Optimization models for training belief-rule-based systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37(4):569–585, 2007.

[22] Y. Wang, Z. Cai, G. Guo, and Y. Zhou. Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(3):560–575, 2007.

[23] D. Liu, K.C. Tan, C.K. Goh, and W.K. Ho. A multiobjective memetic algorithm based on particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(1):42–50, 2007.

[24] W.F. Leong and G.G. Yen. PSO-based multiobjective optimization with dynamic population size and adaptive local archives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(5):1270–1293, 2008.

[25] K.C. Tan, T.H. Lee, D. Khoo, and E.F. Khor. A multiobjective evolutionary algorithm toolbox for computer-aided multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 31(4):537–556, 2001.

[26] B.B. Li and L. Wang. A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(3):576–591, 2007.

[27] X. Zou, Y. Chen, M. Liu, and L. Kang. A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(5):1402–1412, 2008.

[28] D. Buche, P. Stoll, R. Dornberger, and P. Koumoutsakos. Multiobjective evolutionary algorithm for the optimization of noisy combustion processes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 32(4):460–473, 2002.

[29] Y. Jin and B. Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(3):397–415, 2008.

[30] Y.W. Leung and Y. Wang. Multiobjective programming using uniform design and genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 30(3):293–304, 2000.

[31] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. In *Proceedings of Algorithms and Computation*, pages 436–447. Springer, 2008.

[32] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Computer Science*, 425:104–116, 2012.

[33] K. Bringmann and T. Friedrich. The maximum hypervolume set yields near-optimal approximation. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, pages 511–518, 2010.

[34] J. Bader and E. Zitzler. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

[35] D. Brockhoff, J. Bader, L. Thiele, and E. Zitzler. Directed multiobjective optimization based on the weighted hypervolume indicator. *Journal of Multi-Criteria Decision Analysis*, 20(5-6):291–317, 2013.

[36] T. Voß, T. Friedrich, K. Bringmann, and C. Igel. Scaling up indicator-based MOEAs by approximating the least hypervolume contributor: a preliminary study. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, pages 1975–1978, 2010.

[37] C. Priester, K. Narukawa, and T. Rodemann. A comparison of different algorithms for the calculation of dominated hypervolumes. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, pages 655–662, 2013.

[38] K. Narukawa and T. Rodemann. Examining the performance of evolutionary many-objective optimization algorithms on a real-world application. In *Proceedings of IEEE International Conference on Genetic and Evolutionary Computing (ICGEC)*, pages 316–319, 2012.

[39] M. Emmerich, N. Beume, and B. Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In *Proceedings of Evolutionary Multi-Criterion Optimization (EMO)*, pages 62–76. Springer, 2005.

[40] B. Naujoks, N. Beume, and M. Emmerich. Multi-objective optimisation using S-metric selection: Application to three-dimensional solution spaces. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 1282–1289, 2005.

[41] N. Beume, M. Laumanns, and G. Rudolph. Convergence rates of SMS-EMOA on continuous bi-objective problem classes. In *Proceedings of Workshop on Foundations of Genetic Algorithms (FOGA)*, pages 243–252, 2011.

[42] S. Jiang, Y.-S. Ong, J. Zhang, and L. Feng. Consistencies or contradictions of performance metrics in multiobjective optimization? *IEEE Transactions on Cybernetics*, pages 1–14, 2014.

[43] S. Jiang, J. Zhang, and Y.-S. Ong. Multiobjective optimization based on reputation. *Information Sciences*, 286:125–146, 2014.

[44] H. Ishibuchi, M. Yamane, and Y. Nojima. Effects of duplicated objectives in many-objective optimization problems on the search behavior of hypervolume-based evolutionary algorithms. In *Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, pages 25–32, 2013.

[45] J. Knowles. *Local-search and hybrid evolutionary algorithms for Pareto optimization*. PhD thesis, Department of Computer Science, University of Reading, 2002.

[46] L. Bradstreet, L. While, and L. Barone. A fast incremental hypervolume algorithm. *IEEE Transactions on Evolutionary Computation*, 12(6):714–723, 2008.

[47] Lucas Bradstreet, Lyndon While, and Luigi Barone. A fast many-objective hypervolume algorithm using iterated incremental calculations. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.

[48] C.M. Fonseca, L. Paquete, and M. López-Ibánez. An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 1157–1163, 2006.

[49] M.H. Overmars and C.K. Yap. New upper bounds in Klee's measure problem. In *Proceedings of Annual Symposium on Foundations of Computer Science (FOCS)*, pages 550–556, 1988.

[50] H. Gazit. New upper bounds in Klee's measure problem. *SIAM Journal on Computing*, 20(6):1034–1045, 1991.

[51] N. Beume. S-metric calculation by considering dominated hypervolume as Klee's measure problem. *Evolutionary Computation*, 17(4):477–492, 2009.

[52] N. Beume and G. Rudolph. Faster S-metric calculation by considering dominated hypervolume as Klee's measure problem. *Tech. Rep., Sonderforschungsbereich 531 Computational Intelligence, University at Dortmund*, 2006.

[53] A.P. Guerreiro, C.M. Fonseca, and M.T. Emmerich. A fast dimension-sweep algorithm for the hypervolume indicator in four dimensions. In *Proceedings of Canadian Conference on Computational Geometry (CCCG)*, pages 77–82, 2012.

[54] L. Russo and A. Francisco. Quick hypervolume. *arXiv 1207.4598*, 2012.

[55] L. While, L. Bradstreet, and L. Barone. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95, 2012.

[56] L. While and L. Bradstreet. Applying the WFG algorithm to calculate incremental hypervolumes. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2012.

[57] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[58] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multiobjective optimization. *Evolutionary Multiobjective Optimization*, pages 105–145, 2005.

[59] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

[60] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[61] J.J. Durillo, A.J. Nebro, F. Luna, and E. Alba. On the effect of the steady-state selection scheme in multi-objective genetic algorithms. In *Proceedings of Evolutionary Multi-Criterion Optimization (EMO)*, pages 183–197. Springer, 2009.

Siwei Jiang received the M.S. and Ph.D. degrees in computer science from the China University of Geosciences (CUG), Wuhan, China, in 2006 and 2011, respectively. He received another Ph.D. degree in School of Comouter Engineering, Nanyang Technological University (NTU), Singapore, in 2014.

He is currently working as a research fellow in Singapore Institute of Manufacturing Technology (SIMTech), Singapore. His research interests include multiagent evolutionary algorithms, reputation systems and vehicle routing problems.

Jie Zhang received the Ph.D. degree from the University of Waterloo, Waterloo, Canada, in 2009.

He is currently an Assistant Professor at the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. His research interests include artificial intelligence and multiagent systems, trust modeling and incentive mechanisms, and mobile and vehicular ad hoc networks.

Yew-Soon Ong received the B.S. and M.S. degrees in Electrical and Electronics Engineering from Nanyang Technological University (NTU), Singapore, in 1998 and 1999, respectively, and the Ph.D. degree on artificial intelligence in complex design from the Computational Engineering and Design Center, University of Southampton, Southampton, U.K., in 2002.

He is currently an Associate Professor, Director of the Center for Computational Intelligence at the School of Computer Engineering, NTU, and co-Director of the SIMTECH-NTU Joint Lab on Complex Systems. His research interest in computational intelligence spans across memetic computation, evolutionary design, machine learning and agent-based systems. Dr. Ong is the Founding Technical Editor-in-Chief of the Memetic Computing Journal, the Chief Editor of the Springer book series on studies in adaptation, learning, and optimization, and an Associate Editor of the IEEE Computational Intelligence Magazine, IEEE Transactions on Evolutionary Computation, IEEE Transactions on Neural Network & Learning Systems, IEEE Transactions on Cybernetics and others.

Allan Nengsheng Zhang is a Senior Scientist with Singapore Institute of Manufacturing Technology, A*STAR, Singapore. He has more than 20 years experiences in knowledge-based systems and enterprise information systems development. His research interests include knowledge management, data mining, machine learning, artificial intelligence, computer security, software engineering, software development methodology and standard, and enterprise information systems. He and his team are currently working toward research in manufacturing system analyses including data mining, supply chain information management, supply chain risk management using Complex Systems approach, multi-objective vehicle routing problems, and urban last mile logistics.