A Novel Algorithm for Non-dominated Hypervolume-based Multiobjective Optimization

Ke Li, Jinhua Zheng, Miqing Li, Cong Zhou, Hui Lv
Institute of Information Engineering,
Xiangtan University, 411105, Hunan, China
JerryI00@yahoo.com.cn

Abstract—Hypervolume indicator is a commonly accepted quality measure to assess the set of non-dominated solutions obtained by an evolutionary multiobjective optimization algorithm. Recently, an emerging trend in the design of evolutionary multiobjective optimization algorithms is to directly optimize a quality indicator. In this paper, we propose a hypervolume-based evolutionary algorithm for multiobjective optimization. There are two main contributions of our approach, on one hand, a unique fitness assignment strategy is proposed, on the other hand, we design a slicing based method to calculate the exclusive hypervolume of each individual for environmental selection. From an extensive comparative study with three other MOEAs on a number of two and three objective test problems, it is observed that the proposed algorithm has good performance in convergence and distribution.

Keywords — Evolutionary computation, Fitness assignment, Slicing objectives, Hypervolume indicator.

I. INTRODUCTION

In the past twenty years, there has been a growing interest in the studies of multiobjective optimization problems (MOPs). Many evolutionary algorithms have been developed, such as NSGA-II [1], SPEA2 [2], PAES [3] and so on. However, a new focus in the recent research was to apply the quality indicator during the operation of multiobjective evolutionary algorithms. Quality indicators are functions which map the solutions sets to real numbers. This mapped real number can be used to evaluate a set of non-dominated solutions obtained by an evolutionary multiobjective algorithm. Among those existed various quality indicators, hypervolume indicator is the only known unary indicator [6] that can assess the quality of a solutions set by the single value of its hypervolume.

The hypervolume indicator was first introduced by Zitzler et al. in [5] where it was denoted as 'size of the space covered' and also some other terms such as 'S-metric [7]', 'hyperarea metric [8]' and 'Lebesgue measure [9]' were used. As is mentioned in [6], up to the present, according to the investigations, hypervolume indicator is the only known indicator which is compliant with the concept of Pareto dominance. It is said that whenever a set of solutions is better than another set, its hypervolume indicator value is higher than the latter one. Most of the recent proposed indicator based algorithms use hypervolume as the underlying indicator. The first algorithm that incorporates the hypervolume indicator was ESP which was proposed by Huband et al. [12]. Zitzler et al. proposed a framework of indicator based evolutionary algorithm (IBEA) which use two different quality indicators as generation update strategy [13]. Igel et al. use the hypervolume

indicator into the CMA-ES for multiobjective optimization [14]. Most recently, Beume et al. [15] used the *S*-metric as a selection operator in multiobjective optimization evolutionary algorithm in their steady state optimizer SMS-EMOA. Different from the previous ones, in the selection operation, they would abandon the worst solution, which contributes the least hypervolume in the worst ranked front, from the current and offspring population. As is reported in [11], the search ability of the hypervolume-based algorithms such as SMS-EMOA and IBEA are compared with that of other two popular EMO algorithms such as SPEA2 and NSGA-II in the high dimensional space.

Despite of the ideal experimental results and commonly accepted aforementioned advantages of the hypervolume-based algorithm, it inevitably has some drawbacks, too. As is mentioned in [16], the solutions obtained by a steady state MOEA may be not as good as other MEOEAs under some circumstances. What's more, incorporating the quality indicator into EMOA as the selection operator brings the high-cost in calculation which grows exponentially with the increased number of objectives.

In this paper, we proposed a novel hypervolume-based evolutionary algorithm (denoted as HvEA) to solve multiobjective optimization problems. More specifically, the main contributions of our work can be summarized as the following two issues:

1) Proposing a unique fitness assignment strategy which is based on the density information of each individual.

2) Designing a slicing based method to calculate the exclusive hypervolume value of each individual in the three dimensional space.

The remainder of this paper is organized as follows: In Section II, some related definitions of hypervolume indicator are given at first. Next, the straightforward implementation of HvEA is presented and some difficulties of our algorithm are discussed separately. The performance of our algorithm is examined in Section IV through computational experiments. Section V concludes the paper and outlines future work.

II. RELATED DEFINITION OF HYPERVOLUME INDICATOR

Without loss of generality, a decision vector a is said to dominate a vector b is defined as $\forall i, f_i(a) \leq f_i(b) \wedge \exists j, f_j(a) < f_j(b)$

with $i, j \in \{1, \cdots, n\}$, $f : \mathbb{R}^m \to \mathbb{R}^n$ and $a, b \in \mathbb{R}^m$, where n is the number of objectives and m is the number of variables. The Pareto optimal solutions set is made up of the non-dominated

decision vectors in $\mathbb{R}^m$. And its corresponding image under *f* is the so called Pareto front that can be found in the objective space.

The common goal of hypervolume-based algorithms is to maximize the hypervolume indicator I_{Hv} of a set of non-dominated solutions. Let *Leb* denotes the Lebesgue measure, then the hypervolume indicator $I_{Hv}(S)$ of a solution set $S \subseteq \mathbb{R}^m$ can be defined as the hypervolume of the space which is dominated by the set *S* and is bounded by a reference point $x_{ref} = (r_1, \cdots, r_n) \in \mathbb{R}^n$:

$$I_{Hv}(S) = Leb\left(\bigcup_{x \in S} \left[f_1(x), r_1 \right] \times \left[f_2(x), r_2 \right] \times \cdots \times \left[f_n(x), r_n \right] \right)$$

where $\left[f_1(x), r_1 \right] \times \left[f_2(x), r_2 \right] \times \cdots \times \left[f_n(x), r_n \right]$ is the *n* dimensional hypercuboid consisting of all points which are bounded by the point *a* and the reference point.

As to a specific solution *p*, the hypervolume that is contributed by p exclusively can be defined as the exclusive hypervolume of *p* (denoted as *ExcHv*)

$$ExcHv = I_{Hv}\left(S \cup \{ p \} \right) - I_{Hv}\left(p \right)$$

where *S* represents a solution set.

III. OUR PROPOSED ALGORITHM

In this section, we describe an evolutionary algorithm named HvEA (Hypervolume-based Evolutionary Algorithm) for multiobjective optimization problems.

The main loop of HvEA is given by Algorithm 1. It is composed of a standard structure of evolutionary algorithm and is formed by the successive application of mating selection, variation, and environmental selection. As to mating selection, binary tournament selection is proposed here. The variation operator consists of a real-parameter SBX crossover operator and variable-wise polynomial mutation operator to generate *N* offspring. At last, in order to create a new population, pick out the most promising *N* solutions from the mixed population of parent and offspring population, we use environmental selection to create a new population.

Algorithm 1: HvEA main loop

Require: reference set $x_{ref} \subseteq IR^n$, population size *N*, number of generations g_{max}
1: initialize population *P* at random
2: $g = 0$
3: **while** $g \leq g_{max}$ **do**
4:     $P' = \textbf{\textit{MatingSelection}}(P, x_{ref}, N)$
5:     $P'' = \textbf{\textit{VariationOperator}}(P', N)$
6:     $P = \textbf{\textit{EnviromentalSelection}}(P' \cup P'', R, N)$
7:     $g = g + 1$
8: **end while**

Different from the existing algorithms, there are two major contributions as is mentioned previously in our proposed method. In the remainder of this section, we devote two separate subsections for describing them.

A. Fitness Assignment

In our method, we propose a generalized fitness assignment strategy which is similar to the strategy used in SPEA2. The fitness function *F(i)* consists two separate parts, which are the raw fitness value *R(i)* and the density value *D(i)* of each individual. It can be defined as

$$F(i) = R(i) + D(i)$$

where the raw fitness value *R(i)* is the same as what have been defined in SPEA2. The density value *D(i)* is introduced as follows.

Definition I. In the objective space, the influence factor of the ith individual to the individual *y* is defined as:

$$\varphi\left(l_{i \to y} \right) : \partial \to \partial$$

where $l_{i \to y}$ is the Euclidean distance from the individual *i* to individual *y*. In here $\varphi\left(l_{i \to y} \right)$ is a decreasing function and it can perform in various forms, such as parabolic line, square wave function, Gaussian function. In this paper we use the standard normal distribution function.

Definition II. Let an individual $y \in \mathbb{R}^n$, the density information of *y* is defined as the sum of the influence factors to it:

$$density(y) = \sum_{i=1}^{N} \varphi\left(l\left(i, y \right) \right)$$

For a certain individual, this method calculating density information has fully considered the influence made by other individuals in different distances. Namely, the closer, the more influential; the farther, the less, even no influence. According to the above formula of the influence factor, with the increasing of the distance between two individuals, the influence to each other will weaken. Consequently, individuals will have no impact on each other when the distance reaches a certain level. Therefore, to improve the efficiency of density calculation, we ignore those individuals which are less influential. Here we only take $\lfloor \sqrt{N} \rfloor$ of the nearest individuals around individual y into account.

Afterwards, the density value *D(i)* corresponding to *i* is defined by:

$$D(i) = 1 \Big/ density(i) + 2$$

In the denominator, two is added to ensure that its value is greater than zero and *D(i)* < 1.

It is clear that the fitness value of non-dominated individual is less than 1. Having bigger fitness of an individual shows its greater density, but also means it has smaller capability of survival and reproduction. The density value calculation of each individual is of complexity $O(\lfloor \sqrt{N} \rfloor)$, then the total complexity is $O\left(\left(\lfloor \sqrt{N} \rfloor \right)^2 \right)$.

B. Calculation of exclusive hypervolume

As is mentioned in previous sections, to compute the whole hypervolume of a set of points is usually a waste of time. The running time complexity grows exponentially with the number of objectives, even for the best algorithm [17] to calculate the

hypervolume indicator. For the sake of saving time, there is an urgent to develop a dedicated algorithm to directly calculate all values of the exclusive hypervolume.

For the case of two-dimensional space, the computation for the exclusive hypervolume of each point is similar to that has been discussed in [10]. The running time complexity of the computation procedure in this case is governed by the sorting algorithm that is $O(nlogn)$.

For the case of three objectives, we design a novel slicing based method which calculates the exclusive hypervolume in $O(n^3)$. The main frame structure is shown in Algorithm 2 and some of its details are described as follows.

Algorithm 2: Calculate the hypervolume indicator

Require: sort the input solution set *ps* on ascending order of the 1st objective value

1: ***IHV*** (*ps*, *p*, *nobj*)
2: choose the appropriate points and store them into array *prior_level*, *prior_dominated*, *next_level* separately
3: **if** (*prior_dominated* == ∅ ☐ *prior_level* == ∅)
4:    *ExcHv* = ***Sub_proc*_1** (*ps*, *nobj*, *N*)
5: **else if** (*prior_dominated* == ∅ ☐ *prior_level*! = ∅)
6:    *ExcHv* = ***Sub_proc*_2** (*ps*, *nobj*, *N*)
7: **else if** (*prior_dominated* != ∅ ☐ *prior_level* == ∅)
8:    *ExcHv* = ***Sub_proc*_3** (*ps*, *nobj*, *N*)
9: **else**
10:    *ExcHv* = ***Sub_proc*_4** (*ps*, *nobj*, *N*)
11: **return** *ExcHv*

Here, *p* denotes the current examined point. At the beginning, the input solutions are pre-sorted on ascending order of the values at the 1st objective. In this algorithm, four dedicated arrays are proposed to store specific points.

1) *prior_dominated*[]: This array is used to store the points which are prior to *p* and dominated by it.

2) *prior_level*[]: This array is used to store the points which are prior to *p* and nondominated with it.

3) *next_level*[]: This array is used to store the points which follow *p* and nondominated with it.

4) *same_cur*[]: This array is used to store the points which share the same value with *p* at the 1st objective.

At first, the algorithm searches forward and backward to find the fixed points to store into the arrays *prior_level*[], *prior_dominated*[], and *next_level*[] respectively. After that, we begin to calculate the exclusive hypervolume. We divide the points to be processed into the following four cases.

1) Case 1: There are no points prior to *p* that nondominated and dominated by it.

2) Case 2: There are only some points prior to *p* that dominated by it.

3) Case 3: There are only some points prior to *p* that nondominated with it.

4) Case 4: There are both points prior to *p* that dominated and nondominated with it.

It is obvious that Case 1 is similar to Case 4 and Case 2 is similar to Case 3. Therefore, only Case 1 (shown in algorithm 3) and Case 3 (shown in algorithm 4) are discussed here.

Algorithm 3: Calculate the hypervolume indicator in case 1

Require: solution set *ps* size *N*, number of objectives *nobj*

1: ***Sub_proc*_1** (*ps*, *nobj*, *N*)
2: store the appropriate points into array *same_cur*
3: **if** (*next_level* == ∅)
4:    u_1 = *GeUppertBound* ($obj_1(p)$)
5:    *slice_depth* = $|obj_1(p) - u_1|$
6:    $obj_2(ref)$ = *GetLowerBound* ($obj_2(p)$)
7:    $obj_3(ref)$ = *GeUppertBound* ($obj_3(p)$)
8:    *cur_area* = $|(obj_2(p) - obj_2(ref))| * |(obj_3(p) - obj_3(ref))|$
9:    *ExcHv* = *slice_depth* * *cur_area*
10: **else**
11:    **for** (*i* = 0; *i* < *N*; *i* += *j*)
12:       u_1 = *GeUppertBound* ($obj_1(next\_level[i])$)
13:       *slice_depth* = $|obj_1(next\_level[i]) - u_1|$
14:       $obj_2(ref)$ = *GetLowerBound* ($obj_2(next\_level[i])$)
15:       $obj_3(ref)$ = *GeUppertBound* ($obj_3(next\_level[i])$)
16:       *cur_area* = $|(obj_2(next\_level[i]) - obj_2(ref))| *$
                    $|(obj_3(next\_level[i]) - obj_3(ref))|$
17:       *ExcHv* = *slice_depth* * *cur_area*
18: **return** *ExcHv*

Algorithm 4: Calculate the hypervolume indicator in case 3

Require: solution set *ps* size *N*, number of objectives *nobj*

1: ***Sub_proc*_3** (*ps*, *nobj*, *N*)
2: store the appropriate points into array *same_cur*
3: **if** (*next_level* == ∅)
4:    u_1 = *GeUppertBound* ($obj_1(p)$)
5:    *slice_depth* = $|obj_1(p) - u_1|$
6:    $obj_2(ref)$ = *GetLowerBound* ($obj_2(p)$)
7:    $obj_3(ref)$ = *GeUppertBound* ($obj_3(p)$)
8:    store the appropriate points that are needed to be deleted into array *removal*
9:    calculate the area of removal part (denoted as *cut_area*)
10:    *total_area* = $|(obj_2(p) - obj_2(ref))| * |(obj_3(p) - obj_3(ref))|$
11:    *cur_area* = *total_area* – *cut_area*
12:    *ExcHv* = *slice_depth* * *cur_area*
13: **else**
14:    **for** (*i* = 0; *i* < *N*; *i* += *j*)
15:       u_1 = *GeUppertBound* ($obj_1(next\_level[i])$)
16:       *slice_depth* = $|obj_1(next\_level[i]) - u_1|$
17:       $obj_2(ref)$ = *GetLowerBound* ($obj_2(next\_level[i])$)
18:       $obj_3(ref)$ = *GeUppertBound* ($obj_3(next\_level[i])$)
19:       store the appropriate points that are needed to be deleted into array *removal*
20:       calculate the area of removal part (denoted as *cut_area*)
21:       *total_area* = $(obj_2(next\_level[i]) - obj_2(ref)) *$
                    $(obj_3(next\_level[i]) - obj_3(ref))$
22:       *cur_area* = *total_area* – *cut_area*
23:       *ExcHv* = *slice_depth* * *cur_area*
24: **return** *ExcHv*

After the above pre-treatment, the procedure to calculate the exclusive hypervolume can be achieved in the following four

steps:

Step 1: Determine the depth (denoted as *slice_depth*) of the current slice.

Step 2: Determine the shape of the current slice.

Step 3: Reconfirm the temporary reference point.

Step 4: Calculate the exclusive hypervolume of *p*.

At the beginning of calculating the exclusive hypervolume, we store the appropriate points into the array *same_cur*[]. If array *next_level*[] is empty, we only need to process the points prior to *p*, otherwise the points after *p* are also needed to be processed. Since only the non-dominated points have contributions to the total hypervolume indicator, the task of *Step* 2 is to scan the points of the set one after another in order to find the appropriate points to organize the shape of the slice. It is obvious that the time complexity of this process is bounded by $O(N)$. When processing Case 2 or Case 4, we have to determine the part that will be deleted. The will-be deleted part is organized by some of points that are prior to *p* and dominated by it. In *Step* 3, the temporary reference point is composed of the largest lower bound at the 2nd objective and the lowest upper bound at the 3rd objective. In conclusion, to calculate the exclusive hypervolume of one point costs $O(N^2)$ time in the worst condition.

In order to illustrate the above process of calculating the exclusive hypervolume, Fig. 1 (given in Appendix A) gives a simple example. The input solution set is as follows:

$S = \{(1, 8, 7), (2, 6, 3), (4, 5, 8), (5, 2, 5), (7, 3, 2), (10, 1, 9)\}$

The initial reference point is chosen as (12, 12, 12). The current examined point is chosen as (5, 2, 5).

IV. SIMULATION RESULTS ON TEST PROBLEMS

A. Settings

In order to validate the proposed HvEA was tested on several test problems. We compared our algorithm with three well established EMOAs: *ε*-MOEA [4], NSGA-II, SPEA2. As is referred above, we invoke the same variation operator and the same parameter settings used in previous three algorithms [1][2][4]. We have selected first the following bi-objective unconstrained problems: ZDT1 to ZDT4 and ZDT6, which are all defined in [19]. Then we applied functions DTLZ1 to DTLZ3, which are all defined in [20]. 20,000 function evaluations are calculated for the ZDT functions and 30,000 on all DTLZ functions except for DTLZ3, where 100,000 function evaluations are used. The population size is set to 100 for all functions except for DTLZ3, where 200 individuals are applied.

To assess the performances of these algorithms on the test problems, the following three different aspects are normally taken into account:

1) Convergence. The distance of the obtained non-dominated set to the Pareto optimal front;

2) Uniformity. The uniformity of the obtained non-dominated solutions set;

3) Spread. The distribution extent of the obtained non-dominated solutions set.

In this paper, four different performance metrics are invoked here. The generation distance (*GD*) [21], related with the first aspect, is used to evaluate the closeness to the Pareto front; for the second aspect, the uniform assessment (*UA*) [22] is used to assess the diversity in the solutions obtained; and then, the last aspect, the spread assessment (*SA*) [23] is used to assess the spread of the obtained solution set; both of the convergence and uniformity of the solutions found can be evaluated by the *S*-metric (hypervolume) [5]. The reference point which is used for the calculation of *S*-metric is chosen as $(0.7, 0.7, 0.7)^T$, for DTLZ1 and as $(1.1, 1.1, 1.1)^T$ for the other functions [4].

B. Two-objective Test Problems

The experiment results for two-objective case are summarized in Table I and II (shown in Appendix B). For each problem, we have carried out ten independent runs, and the tables include the average and standard deviation.

For convenient comparison, the best results for convergence, and *S*-metric measures that published in [15] are copied into Table 1 and 2. The performances of algorithms in each metric for different test problems are ranked and the best results are shown in bold. The challenge of ZDT1, ZDT2 and ZDT3 lies on the high-dimensionality of these problems. Many MOEAs have achieved very good results on these problems. The results of HvEA are ranked best on both convergence and *S*-metric measures. As to the uniformity measure, HvEA get the best result on function ZDT1. The spread of solutions of HvEA is the best on function ZDT3, but are both worse than SPEA2 and NSGA-II on functions ZDT1 and ZDT2. ZDT4 is a hard optimization problem with 21^9 local Pareto fronts that end to mislead the optimization algorithm. HvEA get the most convergent solutions on this function. NSGA-II generated the most extensive solutions on this function. With the test problem ZDT6, there are two major difficulties. The first one is the thin density of solutions towards the Pareto front and the second one lies on non-uniform spread of solutions along the front. Here, HvEA is ranked best on all of these measures.

C. Three-objective Test Problems

Now we refer to the three-objective test problems. The results are depicted in Table III and IV (shown in Appendix B). For each problem, we have carried out 5 independent runs, and the table includes the average and standard deviation.

The true Pareto front of DTLZ1 lies on the linear hyper-plane: $\sum_{m=1}^{M} f_m = 0.5$. The difficulty in this problem is converging to the hyper-plane, the search space contains (11^k-1) local Pareto optimal fronts. From Table 3 and 4, we can see that HvEA is ranked best on *GD* and *S*-metric measures and is second on *UA* and *SA* measures. Next, we consider the three-objective test problem of DTLZ2, the Pareto optimal front satisfying $\sum_{i=1}^{M} f_i^2 = 1$. HvEA is ranked best on *GD* and *S*-metric, is second on *UA* and third on *SA*. DTLZ3 is a hard optimization problem with 3^9 local Pareto fronts, whereas the optimal front is equivalent to that of DTLZ2. HvEA has the best convergence. The *UA* and *SA* measures are both ranked second.

In conclusion, the outstanding results on *GD* show that the incorporation of hypervolume indicator as a selection operator makes our algorithm to be a better convergent one than the examined well established algorithms. Since there is no explicit strategy for preserving the boundary solutions, the spread of solutions obtained by HvEA is not as good as those of SPEA2 and NSGA-II. Besides, the excellent result on *S*-metric is a very encouraging result even though good performance seems to be natural, for the hypervolume indicator is incorporated as the selection operator.

V. CONCLUSIONS

In this paper, we propose HvEA (Hypervolume-based Evolutionary Algorithm) to solve multiobjective optimization problems. It incorporates a unique fitness assignment strategy which is based on the density information of each individual for mating selection. A novel slicing based method to calculate the exclusive hypervolume which is used for environmental selection is proposed here. HvEA is compared to three well established techniques like SPEA2, NSGA-II and ε-MOEA. A distinguishing convergence and *S*-metric measures are received from the computational experiments.

By now, computing the hypervolume indicator is still time-consuming for more than three objectives. Thus greatly hampers HvEA to be applicable to higher dimensional problems. This can be an encouraging issue for future research. As is proposed in [18], in the hypervolume-based algorithms the actual hypervolume is not so important. Instead, the rankings of solutions which are according to the hypervolume indicator are more significant. Following this issue, we may design an estimate algorithm that just calculates the approximation of the hypervolume of each individual and remove the solutions with smaller approximation. This may make our algorithm effective for more than three objectives.

REFERENCES

[1] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA--II, *IEEE Transactions on Evolutionary Computation*, 6(2): 182-197, 2002

[2] Eckart Zitzler, Marco Laumanns and Lothar Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-Report 103, 2001

[3] Joshua Knowles and David Corne. Approximating the nondominated front using the Pareto archive evolutionary strategy. *Evolutionary Computation*, 8(2): 149-172, 2000.

[4] Kalyanmoy Deb, Manikanth Mohan, Shikhar Mishra: Evaluating the epsilon-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation*, 13(4):501-525 (2005)

[5] 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.

[6] E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

[7] J. Knowles and D. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 100–116, Apr. 2003.

[8] D. A. V. Veldhuizen. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Graduate School of Engineering, Air Force Institute of Technology, Air University, June 1999.

[9] M. Fleischer, "The measure of Pareto optima: Applications to multiobjective Metaheuristics". In C. M. Fonseca et al., editor, *Evolutionary Multi-objective Optimization 2003* (*EMO 2003*), volume 2632 of LNCS, pp. 519–533. Springer-Verlag, 2003.

[10] M. Emmerich, N. Beume, and B. Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In Conference on *Evolutionary Multi-Criterion Optimization 2005* (*EMO 2005*), volume 3410 of LNCS, pages 62–76. Springer, 2005.

[11] T. Wagner, N. Beume, and B. Naujoks. Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. In S. Obayashi et al., editors, Conference on *Evolutionary Multi-Criterion Optimization 2007* (*EMO 2007*), volume 4403 of LNCS, pages 742–756, Berlin Heidelberg, Germany, 2007. Springer. extended version published as internal report of Sonderforschungsbereich 531 Computational Intelligence CI-217/06, Universität Dortmund, September 2006.

[12] S. Huband, P. Hingston, L. While, L. Barone, An evolution strategy with probabilistic mutation for multiobjective optimization, In 2003 *IEEE Congress on Evolutionary Computation* (*CEC 2003*), vol. 4, IEEE Press, Piscataway, NJ, 2003, pp. 2284–2291.

[13] E. Zitzler, S. Kűnzli, Indicator-based selection in multiobjective search, in: *Parallel Problem Solving from Nature* (*PPSN 2004*), Springer, Berlin, 2004, pp. 832–842.

[14] C. Igel, N. Hansen, and S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

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

[16] L. Bradstreet, L. Barone, and L. While. Maximising Hypervolume for Selection in Multi-objective Evolutionary Algorithms. In 2006 *IEEE Congress on Evolutionary Computation* (*CEC 2006*), pages 6208–6215, Vancouver, BC, Canada, 2006. IEEE.

[17] L. While, P. Hingston, L. Barone, and S. Huband. A Faster Algorithm for Calculating Hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1):29–38, Feb. 2006.

[18] J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. TIK Report 286, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, November 2008.

[19] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173-195, 2000

[20] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. In A. Abraham, R. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter 6, pages 105–145. Springer, 2005.

[21] Van Veldhuizen David A. and Gary B. Lamont. Evolutionary Computation and Convergence to a Pareto Front, In John R. Koza, editor, Late Breaking Papers at the Genetic Programming Conference, 1998. 221-228

[22] Miqing Li, Jinhua Zheng and Guixia Xiao. Uniformity Assessment for Evolutionary Multi-Objective Optimization, in Proceedings of *IEEE Congress on Evolutionary Computation* (*CEC 2008*), 625-632, 2008

[23] Miqing Li and Jinhua Zheng. Spread Assessment for Evolutionary Multi-Objective Optimization, *Evolutionary Multi-Criterion Optimization, 5th International Conference 2009* (*EMO 2009*), Nantes, France 216-230, 2009

APPENDIX A

Fig. 1 gives a simple example to illustrate the process of calculating the exclusive hypervolume of a specific point. In this figure, we choose the 4th point as the current examined point *p*. The polyhedron enclosed by *p* and the reference point (denoted as *Ref*) is sliced into three smaller polyhedrons according to the dashed lines. Then we project these polyhedrons into two-dimensional space separately. The shadowed polygons are the parts that need to be calculated. The red circle is the initial reference point and the green one is the reconfirmed temporary reference point.
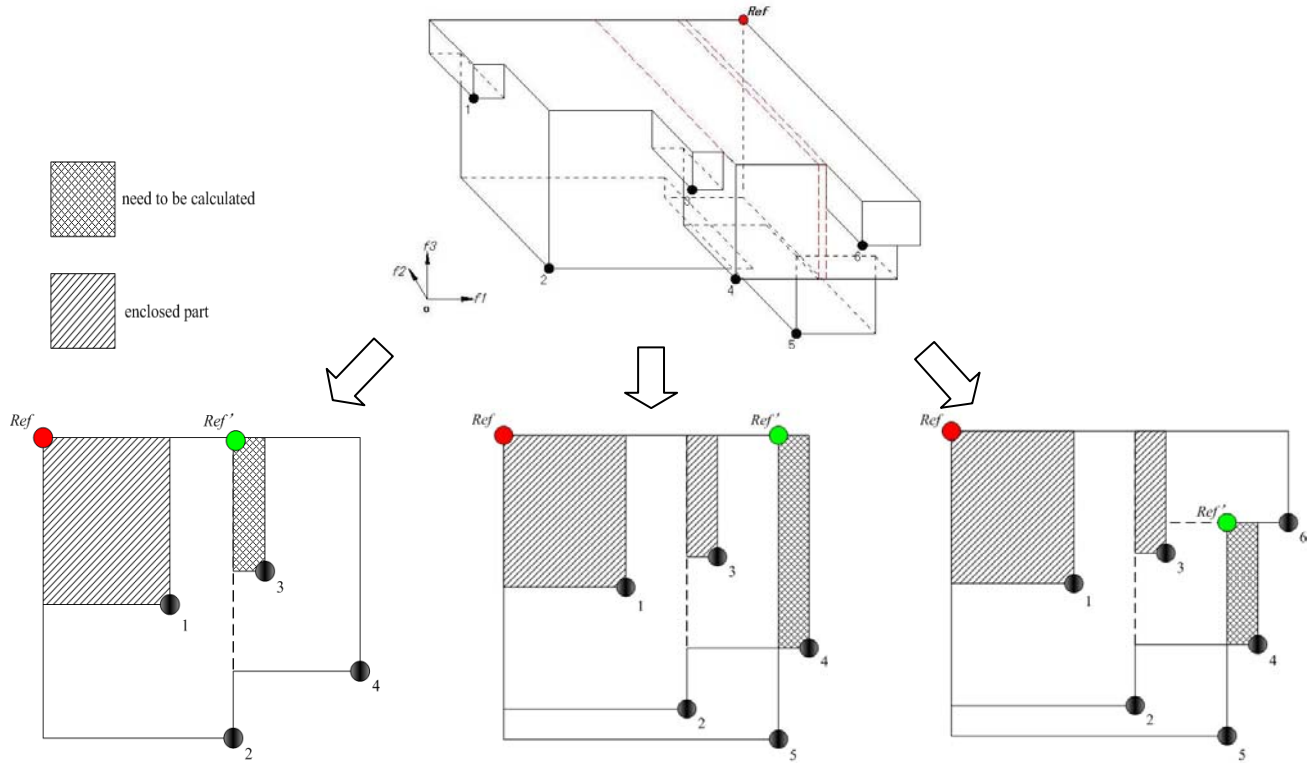
Fig. 1 A simple example for six points

The empirical results are summarized in the following four tables.

Table I   COVERGENCE AND UNIFORMITY MEASURE FOR 2-OBJECTIVE TEST PROBLEMS

Test problems	MOEAs	*GD*			*UA*		
		Average	Std.Dev	Rank	Average	Std.Dev	Rank
ZDT1	NSGA-II	0.00054898	6.62e-05	3	0.44177108	5.31e-05	2
	SPEA2	0.00100589	12.06e-05	4	0.75664550	10.12e-05	4
	ε-MOEA	0.00039545	1.22e-05	2	0.6163923	2.12e-05	3
	HvEA	**0.00016414**	4.21e-05	1	**0.76011216**	3.13e-05	1
ZDT2	NSGA-II	0.00037851	1.88e-05	2	0.43541380	1.76e-05	4
	SPEA2	0.00082852	11.38e-05	4	**0.76688318**	12.72e-05	1
	ε-MOEA	0.00046448	2.47e-05	3	0.7243562	1.98e-05	3
	HvEA	**0.000156531**	1.56e-05	1	0.73665959	2.23e-05	2
ZDT3	NSGA-II	0.00232321	13.95e-05	3	0.33308437	9.23e-05	4
	SPEA2	0.00260542	15.46e-05	4	**0.72215054**	12.72e-05	1
	ε-MOEA	0.00175135	7.45e-05	2	0.7023855	5.23e-05	3
	HvEA	**0.000540217**	17.21e-05	1	0.70727224	13.17e-05	2
ZDT4	NSGA-II	0.00639002	0.0043	3	0.43457665	0.0062	4
	SPEA2	0.00769278	0.0025	4	0.72280722	0.0073	2
	ε-MOEA	0.00259063	0.0006	2	0.7135789	0.0013	3
	HvEA	**0.000256040**	0.0012	1	**0.73833501**	0.0008	1
ZDT6	NSGA-II	0.07896111	0.0067	4	0.51667780	0.0095	4
	SPEA2	0.00573584	0.0009	2	0.70857258	0.0010	2
	ε-MOEA	0.06792800	0.0118	3	0.6798456	0.0098	3
	HvEA	**0.001512960**	0.0095	1	**0.72680862**	0.0089	1

Table II   SPREAD AND *S*-METRIC MEASURE FOR 2-OBJECTIVE TEST PROBLEMS

Test problems	MOEAs	*SA*			*HV*		
		Average	Std.Dev	Rank	Average	Std.Dev	Rank
ZDT1	NSGA-II	**2.360414**	5.21e-05	1	0.8701	3.85e-04	4
	SPEA2	2.353484	9.11e-05	2	0.8708	1.86e-04	2
	ε-MOEA	2.330971	2.16e-05	4	0.8702	8.25e-05	3
	HvEA	2.346017	3.13e-05	3	**0.8750**	2.45e-05	1
ZDT2	NSGA-II	**1.706483**	2.08e-05	1	0.5372	3.01e-04	4
	SPEA2	1.703973	9.32e-05	2	0.5374	2.61e-04	3
	ε-MOEA	1.692582	1.65e-05	4	0.5383	6.39e-05	2
	HvEA	1.695270	1.05e-05	3	**0.5823**	4.15e-05	1
ZDT3	NSGA-II	2.026019	12.42e-05	3	1.3285	1.72e-04	3
	SPEA2	2.019513	16.12e-05	4	1.3276	2.54e-04	4
	ε-MOEA	2.039567	6.35e-05	2	1.3287	1.31e-04	2
	HvEA	**2.056663**	15.23e-05	1	**2.7712**	1.53e-4	1
ZDT4	NSGA-II	2.347478	0.0051	4	0.8613	0.00640	2
	SPEA2	2.362302	0.0046	3	0.8609	0.00536	3
	ε-MOEA	2.363267	0.0013	2	0.8509	0.01537	4
	HvEA	**2.378865**	0.0010	1	**0.8671**	0.01235	1
ZDT6	NSGA-II	1.847677	0.0057	2	0.3959	0.00894	4
	SPEA2	1.846541	0.0007	3	0.4968	0.00117	2
	ε-MOEA	1.845432	0.00951	4	0.4112	0.01573	3
	HvEA	**1.849740**	0.00752	1	**0.8507**	0.05341	1

Table III   COVERGENCE AND UNIFORMITY MEASURE FOR 3-OBJECTIVE TEST PROBLEMS

Test problems	MOEAs	*GD*			*UA*		
		Average	Std.Dev	Rank	Average	Std.Dev	Rank
DTLZ1	NSGA-II	0.06791356	1.38e-03	4	0.27106877	1.23e-04	4
	SPEA2	0.0033377	9.17e-03	3	0.75209725	5.32e-03	3
	ε-MOEA	0.00245	9.52e-05	2	**0.8143912**	12.34e-05	1
	HvEA	**0.0021354**	2.35e-03	1	0.7624552	3.25e-03	2
DTLZ2	NSGA-II	0.0098249	2.72e-05	4	0.32723246	5.24e-05	4
	SPEA2	0.0096292	3.45e-05	3	**0.81348710**	7.53e-05	1
	ε-MOEA	0.0073462	6.35e-05	2	0.7199123	4.24e-05	3
	HvEA	**0.0062536**	5.16e-04	1	0.7324523	9.92e-05	2
DTLZ3	NSGA-II	0.25543011	1.06e-03	4	0.29579150	2.46e-04	4
	SPEA2	0.14616599	9.03e-03	3	**0.84215285**	8.85e-03	1
	ε-MOEA	0.0122290	2.23e-04	2	0.5192874	2.37e-04	3
	HvEA	**0.0092467**	3.25e-03	1	0.5242596	2.64e-03	2

Table IV   SPREAD AND *S*-METRIC MEASURE FOR 2-OBJECTIVE TEST PROBLEMS

Test problems	MOEAs	*SA*			*HV*		
		Average	Std.Dev	Rank	Average	Std.Dev	Rank
DTLZ1	NSGA-II	3.442371	3.52e-04	4	0.313222	5.89e-04	3
	SPEA2	**4.241223**	6.15e-03	1	0.315981	6.98e-04	2
	ε-MOEA	4.127042	10.35e-05	3	0.298487	NC	4
	HvEA	4.209586	2.54e-03	2	**0.328475**	5.25e-05	1
DTLZ2	NSGA-II	3.063681	4.29e-05	2	0.557843	1.02e-03	3
	SPEA2	**3.176612**	6.75e-05	1	0.711212	7.82e-05	2
	ε-MOEA	2.828597	5.04e-05	4	NC	NC	NC
	HvEA	2.985684	8.97e-05	3	**0.782251**	5.23e-04	1
DTLZ3	NSGA-II	3.067451	3.27e-04	3	0.719123	3.24e-05	3
	SPEA2	**3.174212**	7.89e-03	1	0.748125	3.64e-05	2
	ε-MOEA	2.827452	3.27e-04	4	NC	NC	NC
	HvEA	3.096782	2.25e-03	2	**0.769462**	4.72e-05	1