ICIC Express Letters ICIC International ©2010 ISSN 1881-803X
Volume **4**, Number **3**, **June 2010** pp. **1–6**

A Novel Slicing Based Algorithm to Calculate Hypervolume for Multi-Objective Optimization Problems

Ke Li[1], Jinhua Zheng[1], Miqing Li[1], Cong Zhou[1] and Hui Lv[2]
[1]College of Information Engineering
Xiangtan University
P. O. Box 9014, Xiangtan University, China
keli.genius@gmail.com, jhzheng@xtu.edu.cn, limit1008@126.com, cleverchou23@yahoo.com.cn

[2]College of Mathematical and Computational Science
Xiangtan University
P. O. Box 9014, Xiangtan University, China
lvyishan_email@126.com

Received November 2009; accepted February 2010

ABSTRACT. *Hypervolume indicator is a commonly accepted quality measure for the Pareto optimal approximation set. But the calculation of hypervolume indicator is rather difficult, which greatly hampers its applications. Here we propose a slicing-based computation method (MHSO) to calculate hypervolume. MHSO processes objective space and points together. It recursively projects the set of points into fewer dimensions and incorporates a heuristic method to extract non-dominated points from the whole set which are used to calculate the contributed hypervolume in two-dimensional plane. This can enable the time complexity of hypervolume calculation achieve O(nlogn) in three-dimensional case. The time-complexity of our proposed MHSO achieves* $O(n^{d-2}logn)$ *which is better than the original HSO's* $O(n^{d-1})$*. Two different types of test sets are utilized to compare the efficiency both algorithms. Experimental results confirm that MHSO will enable the use of hypervolume with larger population and more objectives.*

Keywords: Multi-Objective Optimization, Hypervolume, Slicing Objectives, Time Complexity.

1. **Introduction.**

In the past ten years, several performance assessments have been emerging in the literature to evaluate the quality of the observed solutions set. Among these, one metric called hypervolume has received more and more attentions in recent years. Hypervolume is also called hypervolume indicator which was first proposed and employed in papers [2-4]. As is investigated in paper [2], hypervolume is the only unary metric of which they are aware that is capable of assessing that a set of solutions *S* is not worse than another set *S'*. In paper [5], Fleischer has proved that a set of solutions are Pareto optima only when its hypervolume is maximized, vice versa. On top of that, comparing to the other metrics, hypervolume has also been subjected to several theoretical investigations in papers [3,5,6]. Hypervolume has some unfavorable properties too: the precision of hypervolume depends on the choice of the reference point, and it is sensitive to the relative scaling of the objectives and to the presence or absence of extreme points in a front. Most recently, hypervolume has also been proposed as a diversity mechanism in evolutionary multiobjective algorithms, for example using it within an archiving strategy or as selection criterion [9,10].

However, calculating hypervolume exactly is very expensive in previously studied algorithms. For problems with more than three objectives, the computational cost may be too expensive to facilitate the use of hypervolume.

The principal contribution of this paper is a novel slicing based algorithm to calculate hypervolume, denoted as MHSO. MHSO processes objectives and points together. It recursively projects points into fewer dimensions. And then slices through the hypervolume are made repeatedly in fewer and fewer objectives. A heuristic method to extract non-dominated points is incorporated, where the extracted points are used to calculate the contributed hypervolume in two-dimensional plane. Comparing to the fastest algorithm HSO reported in paper [7], the computational complexity of MHSO would achieve to $O(nlogn)$ in three-dimensional case and $O(n^{d-2}logn)$ time complexity would achieved in d-dimensional case. Moreover, we show that MHSO is significantly faster than HSO, by two and three orders of magnitude over the selective test fronts in three to eight objectives. Thus MHSO broaden the utility of hypervolume to problems with more objectives and allows the evaluation of much bigger fronts for such problems.

The remainder of this paper is organized as follows. In the next section, a mathematical definition of hypervolume is provided. Then, our MHSO algorithm is proposed, with its advantages of the improved computational complexity and convenient implementation. After that, the results of experiments confirm the utility of MHSO. At last, this paper concludes with a discussion of the proposed technique and outlines some directions for future work in this area.

2. **Problem Statement and Preliminaries.** In multiobjective optimization problems, we aim to find a set of optimal trade-off solutions known as the Pareto optimal set [18-20]. We only take minimal optimization problems into account in this paper. Vectors in the set are partially ordered according to the component-wise order. Given two vectors a and b we say that vector a weakly dominates vector b (in notion: $a \preceq b$) if $a_i \leq b_i$ for all $i \in \{1, \cdots, k\}$. If $a \preceq b$ holds and additionally $a \neq b$, then we say that vector a dominates vector b (in notion: $a \prec b$). Set S is called non-dominated if and only if all vectors in S are mutually non-dominated. Vector a is Pareto optimal if and only if a is non-dominated with respect to all possible vectors in the set. The set of all Pareto optimal vectors is called Pareto front.

The hypervolume $Hv(P)$ of a solution set $P \subseteq S$ can be defined as the hypervolume of the space that is dominated by the set P and is bounded by a reference point $r = (r_1, r_2, \cdots, r_d)$:

$$Hv(P) = Leb\left(\bigcup_{\vec{x} \in P} [\, f_1(\vec{x}), r_1 \,] \times [\, f_2(\vec{x}), r_2 \,] \times \cdots \times [\, f_d(\vec{x}), r_d \,] \right)$$

where $Leb(P)$ is the Lebesgue measure of a set P and $[\, f_1(\vec{x}), r_1 \,] \times [\, f_2(\vec{x}), r_2 \,] \times \cdots \times [\, f_d(\vec{x}), r_d \,]$ is the d-dimensional hyper-cuboid consisting of all points that are weakly dominated by the point $\vec{x}$ but not weakly dominated by the reference point r.

3. **Proposed MHSO Algorithm.** Given *n* mutually non-dominated points in *d* objectives, our proposed MHSO algorithm is directly based upon the idea proposed by Zitzler and Knowles [2-4] but implemented in different ways. On the one hand, we use a projecting idea which projects *d*-dimensional space to (*d*-1)-dimensional space recursively until the number of objectives is decreased to three. This operation aims to reduce the dimensions directly so as to ease the calculations. On the other hand, a heuristic method is employed to extract non-dominated points from the whole solutions set to calculate the contributed hypervolume in two-dimensional plane. This process could enable the computational complexity of hypervolume calculation achieve $O(nlogn)$ in three-dimensional case.

The pseudo-code of **MHSO** is given in Algorithm1. It is the skeleton of the whole calculation procedure. In Algorithm1, function **Sort** is a quick sort procedure which is used to sort the points on descending order by their values at the last objective (without loss of generality). As for line 10, function **Operate3D** (the pseudo-code is given in Algorithm2) is used to process the three-dimensional case. There are three main operations in this function. Specifically, function **CheckDominated** is used to check whether the vectors in temporary array temp[][] are dominated by the remaining points of the set in the previous two-dimensional space. The pseudo-code of Algorithm3 gives the process of function **CalculateArea** which is used to calculate the area of points in the current temp[][]. Function **FilterNondominated** (the pseudo-code is given in Algorithm4) is to extract non-dominated points of the set in the previous two-dimensional space. Besides, function **Truncation** is used to eliminate the useless points, whose maximum depth is attained. Here the maximum depth is defined as the difference between the current examined point and the reference point at the slicing-based objective.

Algorithm 1. The skeleton of **MHSO**

```
1: MHSO (ps, nobj)
2:    while (n)
3:       if (nobj > 3)
4:          tempVolume = MHSO (ps, nobj - 1)
5:             Sort (ps)
6:             store the first vector p1 in the set to a array temp[][]
7:             get the lowest upper bound ul of vector p1
8:             slice_depth[] = |p1 - ul|
9:             if (nobj == 3)
10:             tempVolume = Operate3D (ps)
11:            tempVolume = silce_depth[] * tempVolume
12:            volume += tempVolume
13:            n = Truncation (ps)
```

Algorithm 2. Operation in the three-dimensional space

```
Operater3D (ps)
   store the vectors which have the same value with p1 at the last objective to array
   temp[][]
   flag = CheckDominated ()
   if (flag == 1)
      tempVolume = CalculateArea ()
```

else
  FilterNondominated ()
  tempVolume = **CalculateArea** ()

Algorithm3. Calculate the area in two-dimensional plane

CalculateArea (*ps*)
  Sort the points of *ps* on descending order by values at the 1st objective
  Area = 0
  for ($i = 1$; $i \le N$; i++)
    Area += $|obj_2(p_i) - obj_2(Ref)| \times |obj_1(p_i) - obj_1(p_{i-1})|$
return *Area*

Algorithm 4. Extract the non-dominated vectors in the set

FilterNondominated (*ps*)
  sort the points of *ps* to a sequence $p_1, p_2, \ldots, p_n$ on ascending order by values at the 1st objective
  $NDSet = \{p_1\}$
  $sentinel = p_1$
  for ($i = 2$; $i \le n$; i++)
    if ($obj_2(p_i) > obj_2(sentinel)$)
      continue
    else
      $NDSet = NDSet \cup \{p_i\}$
      $sentinel = p_i$
return *NDSet*

3. **Performance.** In order to study the efficiency of our algorithm, we compared the performance of MHSO with the famous While's HSO [7]. We evaluated them on two different types of non-dominated points set instances: one is randomly generated, the other are samples taken from three distinct Pareto optimal sets of the problems from the DTLZ test suite [16]: they are DTLZ-1, DTLZ-2 and DTLZ-7 separately. All of these data are available from [17]. We choose the worst value of each objective to form a reference point. In order to test the efficiency of proposed heuristic method, we firstly compare the performances of MHSO and HSO on three different DTLZ sets (namely DTLZ-1, DTLZ-2, and DTLZ-7) in three-dimensional case. From FIGURE.1 we can observe that MHSO indeed outperforms HSO in all test instances, with speed-up factors more than double. FIGURE.2-9 show the comparison results in more than four objectives. From these figures we can observe that the performances of MHSO are superior to that of HSO in all test cases. In most situations, the process speed of MHSO is almost two times faster than that of HSO. But when dealing with the eight-dimensional cases, the advantage of MHSO is not as remarkably as before. Besides, if we look carefully, we can observe that both algorithms process more points when dealing with the points set which is randomly generated. As for the DTLZ test suites, both algorithms meet some obstacles when dealing with DTLZ-2 points set which are extracted from a spherical hyper-plane.

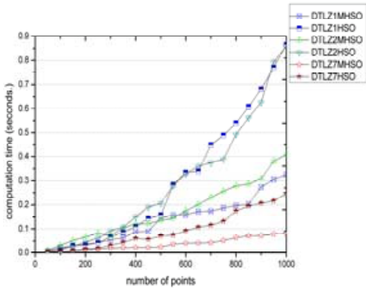

FIGURE 1. 3-dimensional case

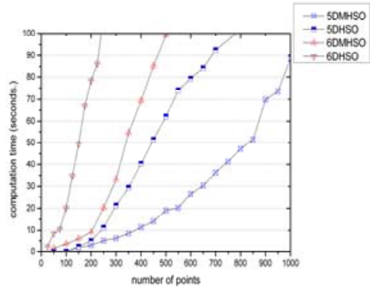

FIGURE 2. DTLZ-1 in 5, 6 objectives

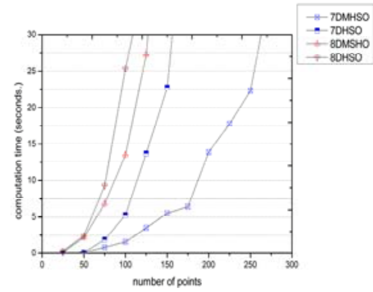

FIGURE 3. DTLZ-1 in 7, 8 objectives

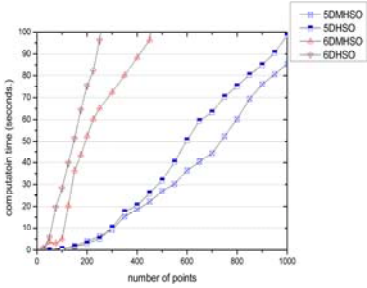

FIGURE 4. DTLZ-7 in 5, 6 objectives

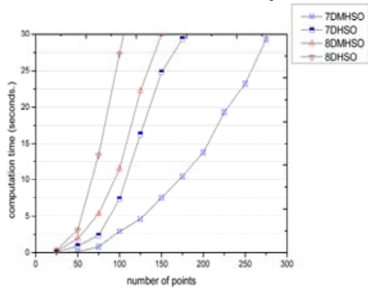

FIGURE 5. DTLZ-7 in 7, 8 objectives

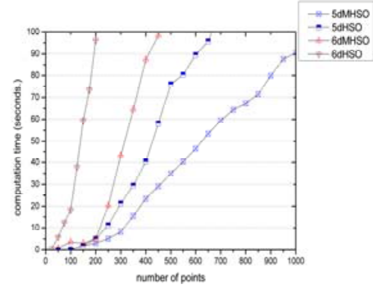

FIGURE 6. DTLZ-2 in 5, 6 objectives

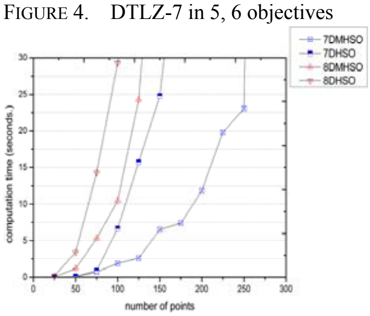

FIGURE 7. DTLZ-2 in 7, 8 objectives

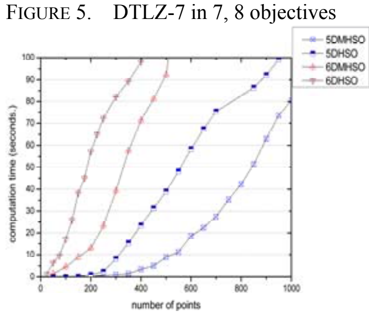

FIGURE 8. Random in 5, 6 objectives

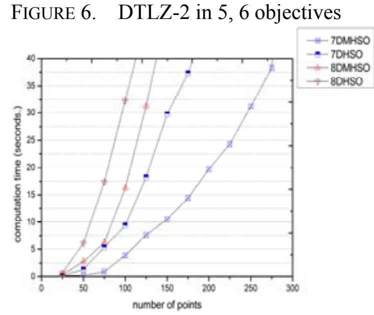

FIGURE 9. Random in 7, 8 objectives

4. **Conclusions.** Hypervolume is a popular metric for evaluating the performance of multiobjective optimization algorithms. This article presented an improved slicing based algorithm called MHSO. We use a projecting idea which recursively project points into fewer dimensions. In three-dimensional case, we incorporate a heuristic operation to extract non-dominated points from the two-dimensional plane, which could enable the worst case time complexity achieve $O(n\log n)$. The total running time complexity in d-dimensional case is bounded by $O(n^{d-2}\log n)$. Experimental results indicate that our proposed algorithm clearly outperforms HSO in all test cases. Thus, MHSO further increases the utility of hypervolume to calculate reasonable sized sets in almost any likely number of objectives.

Acknowledgment. The work was supported by the National Natural Science Foundation of China (60773047), the Natural Science Foundation of Hunan Province (05JJ30125).

REFERENCES

[1] E. Zitzler, Evolutionary algorithms for multiobjective optimization: Methods and applications, *Ph.D. dissertation*, Swiss Federal Institute Technology (ETH) Zurich, Switzerland, 1999.

[2] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca, Performance assessment of multiobjective optimizers: An analysis and review, *IEEE Transactions on Evolutionary Computation*, vol.7, no.2, pp.117-132, 2003.

[3] E. Zitzler and L. Thiele, Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation*, vol.3, no.4, pp.257-271, 1999.

[4] 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, 2003.

[5] M. Fleischer, The measure of Pareto optima: Applications to multiobjective Metaheuristics, *Proc. of 2nd Conference on Evolutionary Multi-Criterion Optimization*, Milan, Italy, pp.519-533, 2003.

[6] L. While, A New Analysis of the LebMeasure Algorithm for Calculating Hypervolume. *Proc. of 3rd Conference on Evolutionary Multi-Criterion Optimization,* Guanajuato, Mexico, pp.326-340, 2005.

[7] L. While, P. Hingston, L. Barone, and S. Huband, A Faster Algorithm for Calculating Hypervolume, *IEEE Transactions on Evolutionary Computation*, vol.10, no.1, pp.29–38, 2006.

[8] D. A. V. Veldhuizen, Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations, P*h.D. dissertation*, Air Force Institute of Technology, Air University, 1999.

[9] J. Knowles, D. Corne, and M. Fleischer, Bounded archiving using the lebesgue measure, *Proc. of the 2003 IEEE International Congress on Evolutionary Computation*, Newport Beach, CA, pp. 2490-2498.

[10] N. Beume, B. Naujoks and M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, *European Journal of Operational Research*, vol.181, no.3, pp.1653-1669, 2007.

[11] J. Wu and S. Azam, Metrics for quality assessment of a multiobjective design optimization solution set, *Journal of Mechanical Design*, vol.123, no.1, pp.18-25, 2001.

[12] N. Beume and G. Rudolph, Faster *S*-metric calculation by considering dominated hypervolume as klee's measure problem, *Proc. of 2nd IASTED Conference on Computational Intelligence*, Anaheim, pp. 231-236, 2006.

[13] M. H. Overmars and C. K. Yap, New upper bounds in Klee's measure problem, *SIAM Journal on Computing,* vol.20, no.6, pp.1034-1045, 1991.

[14] Carlos M. Fonseca, Luis Paquete and Manuel Lopez-Ibanez. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator, *Proc. of 2006 IEEE International Congress on Evolutionary Computation*, pp. 3973-3979, Vancouver, BC, Canada, 2006.

[15] L. While, L. Bradstreet, L. Barone and P. Hingston, Heuristics for optimising the calculation of the hypervolume for multi-objective optimisation problems, *Proc. of 2005 IEEE International Congress on Evolutionary Computation*, vol.3, pp. 2225-2232, München, Germany, 2005.

[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, Scalable multi-objective optimization test problems, *Proc. of 2002 IEEE International Congress on Evolutionary Computation*, vol.1, pp.825-830, Honolulu, Hawaii, 2002.

[17] http://wfg.csse.uwa.edu.au/Hypervolume

[18] Shi-Zheng Zhao and Ponnuthurai Nagaratnam Suganthan, Multi-Objective Evolutionary Algorithm with Ensemble of External Archives. *International Journal of Innovative Computing, Information and Control*, vo. 6, no. 4, pp. 1713-1726, 2010.

[19] Chun-an Liu, New Evolutionary Algorithm for Multi-objective Constrained Optimization. *ICIC Express Letters, An International Journal of Research and Surveys*. vol. 2, no. 4, pp. 339-344, 2008.

[20] Chun-an Liu and Yuping Wang, A New Evolutionary Algorithm for Multi-objective Optimization Problems. *ICIC Express Letters, An International Journal of Research and Surveys*. vol. 1, no. 1, pp. 93-98, 2007.