Applied Soft Computing 30 (2015) 238–248

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A new decomposition based evolutionary algorithm with uniform designs for many-objective optimization

Cai Dai [a,b,*], Yuping Wang [b]

[a] College of Computer Science, Shaanxi Normal University, Xi'an 710062, China
[b] School of Computer Science and Technology, Xidian University, Xi'an 710071, China

A R T I C L E I N F O

Article history:
Received 27 July 2014
Received in revised form 27 January 2015
Accepted 29 January 2015
Available online 7 February 2015

Keywords:
Multi-objective optimization
Decomposition
Uniform design
Weight vector
Many-objective optimization problems

A B S T R A C T

For many-objective optimization problems, how to get a set of solutions with good convergence and diversity is a difficult and challenging work. In this paper, a new decomposition based evolutionary algorithm with uniform designs is proposed to achieve the goal. The proposed algorithm adopts the uniform design method to set the weight vectors which are uniformly distributed over the design space, and the size of the weight vectors neither increases nonlinearly with the number of objectives nor considers a formulaic setting. A crossover operator based on the uniform design method is constructed to enhance the search capacity of the proposed algorithm. Moreover, in order to improve the convergence performance of the algorithm, a sub-population strategy is used to optimize each sub-problem. Comparing with some efficient state-of-the-art algorithms, e.g., NSGAII-CE, MOEA/D and HypE, on six benchmark functions, the proposed algorithm is able to find a set of solutions with better diversity and convergence.

1. Introduction

Multi-objective evolutionary algorithms (MOEAs) are a kind of effective method for solving multi-objective problems because they can handle a set of solutions in parallel. In the last twenty years, there are many well-known MOEAs [1–5] that are proposed, most of these MOEAs are based on Pareto dominance. Such Pareto dominance-based algorithms usually deal well with two or three objectives problems but their searching and selecting ability are often severely degraded with the increased number of objectives [6,50,51]. This is explained by the fact that, as the number of objectives increases, the proportion of non-dominated elements in the population grows, being increasingly difficult to discriminate among solutions using only the dominance relation [51]; if the number of solutions is constant, the size of non-dominance area of solutions will increase with the increase of the number of objectives, these will make the Pareto dominance-based fitness evaluation generate very weak selection pressure toward the Pareto front (PF). Therefore, how to enhance the selection pressure toward the PF and maintain the diversity of obtained solutions are critical for the many-objective optimization algorithms.

Currently, the methods for dealing with many-objective problems can be divided into three categories. The first category uses an indicator function, such as the hypervolume [7–9], as the fitness function. This kind of algorithm is also referred to as IBEAs (indicator-based evolutionary algorithms), and their high search ability has been shown in the literature [10]. Recently, Bader and Zitzler [11] proposed a fast hypervolume-based many-objective optimization algorithm (HypE) which uses Monte Carlo simulation to quickly approximate the exact hypervolume values. However, one of their main drawbacks is the computation time for the hypervolume calculation which exponentially increases with the number of objectives [43], and even if the hypervolume values are calculated by Monte Carlo approximations, its running time is more than 10 h after 50,000 objective function evaluations for seven-objective problems [44]. This limits the application of hypervolume indicator-based evolutionary algorithms to many-objective optimization problems.

The second category takes advantages of solution ranking methods. Specifically, solution ranking methods are used to discriminate among solutions in order to enhance the selection pressure toward the PF, which makes sure the solutions are able to converge to the PF. At present, numerous approaches have been proposed to rank solutions for many-objective problems. Bentley and Wakefield [12] proposed ranking composition methods which extract the separated fitness of every solution into a list of fitness values for each objective. Kokolo and Hajime [13] proposed a relaxed form of dominance (RFD) to deal with what they called dominant resistant

* Corresponding author at: College of Computer Science, Shaanxi Normal University, Xi'an 710062, China. Tel.: +86 13720727536.
E-mail addresses: daicai8403@hotmail.com (C. Dai), ywang@xidian.edu.cn (Y. Wang).

http://dx.doi.org/10.1016/j.asoc.2015.01.062
1568-4946/

solutions, i.e., solutions that are extremely inferior to others in at least one objective. Farina and Amato [14] proposed a dominance relation which takes into consideration the number of objectives where a solution is better, equal and worse than another solution. Sato et al. [15] proposed a method to strength or weaken the selection process by expanding or contracting the solutions' dominance area (CE).

The third category utilizes the scalarizing functions to deal with the many-objective problems. According to the literatures [16–18], scalarizing function-based algorithms could deal better with many-objective problems than Pareto dominance-based algorithms. The main advantage of scalarizing function-based algorithms is that their fitness evaluation can be easily calculated. The representative MOEA in this category is MOEA/D [19] (multi-objective evolutionary algorithm based on decomposition), which makes use of traditional aggregation methods to convert a MOP into a number of single objective optimization sub-problems, and simultaneously optimizes each sub-problem in one single run. Each sub-problem is optimized by using information from its several neighboring sub-problems, which makes MOEA/D have a good performance. MOEA/D works well on a wide range of multi-objective problems with many objectives, discrete decision variables and complicated Pareto sets [20–22]. In MOEA/D, weight vectors play a very important role, they directly determine the distribution of obtained solutions and affect the convergence of obtained solutions. In MOEA/D [19], the uniformity of the used weighted vectors determines the uniformity of the obtained non-dominated optimal solutions; however, the used weighted vectors in MOEA/D are not very uniform and the size N of these weighted vectors should satisfy the restriction $N = C_{H+m-1}^{m}$ (where m is the number of objectives and H is an integer). Thus N cannot be freely assigned and it will increase nonlinearly with the increase of m, which restricts the application of MOEA/D to a certain extent in many-objective optimization problems. Therefore, for many-objective problems, how to set weight vectors is a very difficult but critical task, and it is necessary to consider an efficient and simple method to product the weight vectors [19,23]. Hughes [24] also considers a similar idea to set the weight vectors.

Uniform design (UD) which is proposed by Fang and Wang [25] represents a combination of number theory and numerical analysis. The UD method has been successfully implemented in science, engineering and industries [26–31]. The literature [32] has shown that the uniform design performs better at estimating nonlinear problems than other designs. The foremost goal of the UD method is to find a set of points that are uniformly distributed over the design space, and the set has a small discrepancy. The UD method has been used in MOEAs to generate the weight vectors, for example, Leung and Wang [28] use the UD method to generate multiple weight vectors which are uniformly scattered points on a unit hypercube and each point on the unit hypercube yields a weight vector; the literature [22] uses the UD method to yield weight vectors and design a uniform design multi-objective evolutionary algorithm based on decomposition for many-objective optimization problems, but the algorithm only tests five-objective problems.

Because the computation time for the discrepancy of a set of weight vectors exponentially increases with the number of objectives and the weight vectors, which restricts the application of the UD to many-objective optimization problems. In this paper, we use the inverted generational distance (IGD) [33] to approximate the discrepancy, then we use the UD method to generate a set of points which are uniformly distributed on a unit sphere, and the points are the weight vectors. In addition, a sub-population strategy is used to enhance the local search ability of the proposed algorithm. We make each sub-problem have a sub-population, and each sub-problem uses the information provided by its corresponding sub-population to improve the convergence performance. Then, a selection strategy based on decomposition and the sub-population strategy is designed to help crossover operators carry out the global search and local search. Moreover, a crossover operator based on the UD method is constructed to improve the search capacity. Based on all these, a new decomposition based evolutionary algorithm with uniform design, UDEA/D, is designed for many-objective optimization problems. The experiments demonstrate that UDEA/D can significantly outperform MOEA/D, NSGAII-CE (NSGAII based on contracting or expanding the solutions' dominance area) and HypE on a set of test instances.

The rest of this paper is organized as follows: Section 2 introduces the main concepts of the multi-objective optimization; Section 3 describes two related uniform design methods; Section 4 presents a crossover operator based on a uniform design method; Section 5 presents a new many-objective evolutionary algorithm; while Section 6 shows the experiment results of the proposed algorithm and the related analysis; finally, Section 7 draws the conclusions and proposes the future work.

2. Multi-objective optimization

A multi-objective optimization problem can be formulated as follows [34]:

$$\begin{cases} \min F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ \text{s.t.}\, g_i(x) \le 0,\ i = 1, 2, \ldots, e_1 \\ h_j(x) - 0,\ j = 1, 2, \ldots, e_2 \end{cases} \tag{1}$$

where $x = (x_1, \ldots, x_n) \in X \subset R^n$ is called decision variable and X is n-dimensional decision space. $f_i(x)(i = 1, \ldots, m)$ is the ith objective to be minimized, $g_i(x)(i = 1, 2, \ldots, e_1)$ defines ith inequality constraint and $h_j(x)(j = 1, 2, \ldots, e_2)$ defines jth equality constraint. Furthermore, all the constraints determine the set of feasible solutions which are denoted by Ω. To be specific, we try to find a feasible solution $x \in \Omega$ minimizing each objective function $f_i(x)(i = 1, \ldots, m)$ in F. In the following, four important definitions [35] for multi-objective problems are given.

Definition 1 *(Pareto dominance).* Pareto dominance between solutions $x,z \in \Omega$ is defined as follow. If

$$\begin{aligned} &\forall_i \in \{1, 2, \ldots, m\} f_i(x) \le f_i(x) \\ &\wedge \exists_i \in \{1, 2, \ldots, m\} f_i(x) < f_i(z) \end{aligned} \tag{2}$$

are satisfied, x dominates (Pareto dominate) z (denoted $x \succ z$).

Definition 2 *(Pareto optimal).* A solution vector x is said to be Pareto optimal with respect to Ω, if $\nexists z \in \Omega : z \succ x$.

Definition 3 *(Pareto optimal set (PS)).* The set of Pareto optimal solutions (PS) is defined as:

$$\mathrm{PS} = \{x \in \Omega | \nexists z \in \Omega : z \succ x\} \tag{3}$$

Definition 4 *(Pareto optimal front).* The Pareto optimal front (PF) is defined as:

$$\mathrm{PF} = \{F(x) | x \in \mathrm{PS}\} \tag{4}$$

3. Uniform design

In this section, two uniform design methods are briefly introduced. The main goal of a uniform design is to sample a small set of points from a given closed and bounded set $G \subset R^M$ such that the sampled points are uniformly scattered on G. In the following, we consider only two specific cases of G and describe the main features of uniform design. For more details, we refer to [25].