Applied Soft Computing 35 (2015) 161–174

Contents lists available at ScienceDirect

Applied Soft Computing

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

A double-module immune algorithm for multi-objective optimization problems

Zhengping Liang, Ruizhen Song, Qiuzhen Lin*, Zhihua Du, Jianyong Chen, Zhong Ming, Jianping Yu

College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, PR China

ARTICLE INFO

Article history:
Received 22 July 2014
Received in revised form 16 May 2015
Accepted 20 June 2015
Available online 29 June 2015

Keywords:
Multi-objective optimization
Double-module framework
Immune algorithm
Differential evolution

ABSTRACT

Multi-objective optimization problems (MOPs) have become a research hotspot, as they are commonly encountered in scientific and engineering applications. When solving some complex MOPs, it is quite difficult to locate the entire Pareto-optimal front. To better settle this problem, a novel double-module immune algorithm named DMMO is presented, where two evolutionary modules are embedded to simultaneously improve the convergence speed and population diversity. The first module is designed to optimize each objective independently by using a sub-population composed with the competitive individuals in this objective. Differential evolution crossover is performed here to enhance the corresponding objective. The second one follows the traditional procedures of immune algorithm, where proportional cloning, recombination and hyper-mutation operators are operated to concurrently strengthen the multiple objectives. The performance of DMMO is validated by 16 benchmark problems, and further compared with several multi-objective algorithms, such as NSGA-II, SPEA2, SMSEMOA, MOEA/D, SMPSO, NNIA and MIMO. Experimental studies indicate that DMMO performs better than the compared targets on most of test problems and the advantages of double modules in DMMO are also analyzed.

1. Introduction

Over the last decades, multi-objective optimization problems (MOPs) have attracted a great interest of researchers, which are motivated by the real-world engineering problems, such as job shop scheduling [1,2], water distribution network design [3,4], antenna design [5] and power supply management [6]. For example, the objectives of makespan, total workload, and critical workload in job shop scheduling are all required to be minimized, while the network cost and total head loss in pipes are preferred to be optimized in the design of water distribution network. Obviously, unlike single-objective optimization problems that only seek for one optimal value, MOPs bring new challenges as they need to simultaneously optimize several conflicting objectives. Thus, the output of MOPs is generally a number of equally-optimal solutions termed Pareto-optimal solutions when considering all the objectives. All Pareto-optimal solutions compose the Pareto-optimal set (PS), whose projection in the objective space is termed Pareto-optimal front (PF). Thus, the aim of MOPs is to achieve a subset of PS that is distributed uniformly along the true PF [7,8], which can be provided to the decision maker as the available choices for various practical cases.

Traditionally, multiple objectives are simply aggregated into a single objective optimization problem and then several runs of optimization algorithm are executed in order to find a set of optimal solutions [9]. The representative algorithms include the weighting method, the constraint approach, goal programming and the minmax formulation [10,11]. However, it is pretty difficult to achieve a satisfactory result in limited time using these traditional approaches as it needs to explore a huge solution space in MOPs by multiple runs for gathering a set of optimal solutions. Thus, evolutionary algorithms (EAs) are presented for solving MOPs, which have been demonstrated to be an effective method for MOPs as the population-based searching nature of EAs can obtain multiple Pareto-optimal solutions in a single run [7,8]. Since the first multi-objective EA (MOEA) named vector evaluated genetic algorithm (VEGA) was proposed by Schaffer [12], various MOEAs were developed afterward [7,8,13–16], among which a fast non-dominated sorting genetic algorithm (NSGA-II) [7] and an improved strength Pareto EA (SPEA2) [8] are acknowledged as the two state-of-the-art MOEAs. A fast non-dominated sorting approach and elitism strategy are proposed in NSGA-II, while a new fitness

* Corresponding author. Tel.: +86 75526713281.
E-mail address: qiuzhlin@szu.edu.cn (Q. Lin).

http://dx.doi.org/10.1016/j.asoc.2015.06.022
1568-4946/

assignment strategy and an enhanced archive truncation technique are presented in SPEA2. To further enhance the performance, various promising approaches are designed afterward, such as IBEA [17] and SMSEMOA [18] that embed the quality indicators into the selection procedure, ParEGO [19] that builds a Gaussian processes model of the search landscape to solve expensive MOPs, and MO-CMA-ES [20] that extends the covariance matrix adaptation evolution strategy (CMA-ES) to tackle MOPs. More recently, many competitive MOEAs have been reported with better performance [21–29]. For example, a neighborhood knowledge-based EA for MOPs (NKBA) [21] exploits the neighborhood knowledge systematically for acquiring more thorough local search. A diversity maintain strategy [22] is designed for MOPs to compute a solution density estimation of the archive through a binary space partitioning tree. A novel adaptive local search approach is embedded into MOEAs [23] for accelerating the convergence speed. With more and more MOEAs proposed, the performance for solving MOPs is constantly improved.

On the other hand, other nature-inspired algorithms such as particle swarm optimization (PSO) [30,31], ant colony algorithm [32,33] and artificial immune algorithm [34,35] are also investigated to solve MOPs. Among them, artificial immune algorithm is designed by mimicking the information processing procedures of artificial immune system (AIS), which has attracted much attention and been successfully applied in many research fields, such as data mining, computer security, optimization, and fault diagnosis [36,37]. Especially for tackling MOPs, artificial immune algorithm has demonstrated to be very competitive as experimentally studied in [38–41]. A more detail introduction of multi-objective immune algorithms is presented in Section 2.3.

It is recently found that single-objective optimization process can be effectively embedded into multi-objective algorithms for enhancing their performance. For example, MOEA/D decomposes MOPs into a set of single-objective aggregation problems based on the predefined weight vectors and then assigns each individual to optimize the corresponding subproblem [27]; CMPSO utilizes PSO to optimize each objective using one sub-population and then employs the shared external archive for information exchange between all the objectives [30]; the membrane computing algorithm proposed in Refs. [42,43] uses several cells to optimize each objective and then employs other cells to enhance all the objectives simultaneously. Their promising performance motivates us to study the possibility of embedding single-objective optimization process into artificial immune algorithm, which may further enhance the performance of our previous algorithms [39,40]. Therefore, a double-module immune algorithm for MOPs (DMMO) is accordingly designed, which integrates a single-objective optimization module into the traditional immune algorithm and makes two evolutionary modules cooperatively evolved to solve MOPs. Although the idea to embed single-objective optimization process for MOPs has been investigated in MOEA/D [27], CMPSO [30] and the membrane computing algorithm [42,43], our framework with two evolutionary modules makes DMMO totally different from MOEA/D and CMPSO as they only adopt a single evolutionary module to evolve the population. For the membrane computing algorithm in Refs. [42,43], DMMO is essentially different as they are designed based on different evolutionary computing frameworks with various evolutionary operators. Our first module is aimed at improving each objective independently while the second one follows the traditional immune algorithm to strengthen the multiple objectives simultaneously. The cooperation of two modules is expected to accelerate the convergence speed and maintain the population diversity, which is validated by the experimental studies. When compared with various multi-objective algorithms, *e.g.*, NSGA-II [7], SPEA2 [8], SMSEMOA [18], MOEA/D [27], SMPSO [31], NNIA [38] and MIMO [40], the experimental results show that DMMO performs better on most of test problems.

The rest of this paper is organized as follows. Section 2 introduces the related background, such as MOPs, AIS and some relevant immune algorithms. In Section 3, the proposed DMMO algorithm is described in detail. Section 4 gives the experimental comparisons of DMMO with other algorithms, which validate the superior performance of DMMO. Moreover, the advantages of double modules in DMMO are also experimentally analyzed. At last, further discussions of the double-module framework are given in Section 5 and conclusions are drawn in Section 6.

2. Background

2.1. Multi-objective optimization problems

Multi-objective optimization problems for minimization can be mathematically described by

$$\text{Minimize}\, f(x) = \{f_1(x), f_2(x), \ldots, f_m(x)\} \quad (1)$$

where x is a decision vector as represented by $x = \{x_1, x_2, \ldots, x_n\} \in \Omega$, n and m are, respectively, the dimensions of the decision and objective vectors, and Ω is the feasible region in decision space. To distinguish the superiority of each solution, the definitions of Pareto optimality are important, which are described as follows [44].

Definition 1. *Pareto domination.* A decision vector $x_A \in \Omega$ is said to dominate another decision vector $x_B \in \Omega$ (noted as $x_A \succ x_B$) if and only if the following conditions are true.

$$(\forall i \in \{1, 2, \ldots, m\} : f_i(x_A) \le f_i(x_B)) \wedge (\exists j \in \{1, 2, \ldots, m\} : f_j(x_A) < f_j(x_B))$$

Definition 2. *Pareto optimal.* A decision vector $x^* \in \Omega$ is called a Pareto-optimal or non-dominated solution if and only if there does not exist another decision vector $x \in \Omega$ that can dominate x^*, which can be described as follows.

$$\neg\exists x \succ x^*, \quad x \in \Omega$$

Definition 3. *Pareto-optimal Set (PS).* All Pareto-optimal vectors compose a PS, which can be defined by

$$PS = \{x | \neg y \in \Omega : y \succ x\}$$

Definition4. Pareto-optimal Front (PF). The corresponding map of PS on the objective space is called PF, which is represented by

$$PF = \{f(x) = \{f_1(x), f_2(x), \ldots, f_m(x) | x \in PS\}$$

2.2. Artificial immune system

Artificial immune system is an interesting bio-inspired intelligent approach that simulates the information processing procedures of biologic immune system [45,46]. When foreign antigens are detected in biologic immune system, its B-cell is adapted correspondingly to eliminate the intruders, which is realized by the processes known as clonal selection and affinity maturation through hyper-mutation. Antibodies that can better recognize an antigen will be selected to proliferate by cloning, which is known as the process of clonal selection. Then, hyper-mutation implements an affinity maturation process proportional to the fitness values in order to generate the matured population. At last, some antibodies with better affinities will be remained as memory cells to prevent the re-intrusion of the previous antigens. This information processing principle gives some inspirations to design artificial