Information Sciences 367–368 (2016) 529–549

Contents lists available at ScienceDirect

Information Sciences

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

Self-adaptive multi-objective evolutionary algorithm based on decomposition for large-scale problems: A case study on reservoir flood control operation

Yutao Qi [a,*], Liang Bao [b], Xiaoliang Ma [a], Qiguang Miao [a], Xiaodong Li [c]

[a] School of Computer Science and Technology, Xidian University, Xi'an, 710071, China
[b] School of Software, Xidian University, Xi'an, 710071, China
[c] School of Computer Science and Information Technology, RMIT University, Melbourne, Australia

A R T I C L E I N F O

Article history:
Received 23 June 2015
Revised 6 November 2015
Accepted 6 June 2016
Available online 7 June 2016

Keywords:
Large-scale multi-objective optimization problem
Genetic operator selection
Neighborhood size selection
Reservoir flood control operation

A B S T R A C T

Large-scale multi-objective optimization problems (LS-MOP) are complex problems with a large number of decision variables. Due to its high-dimensional decision space, LS-MOP poses a significant challenge to multi-objective optimization methods including multi-objective evolutionary algorithms (MOEAs). Following the algorithmic framework of multi-objective evolutionary algorithm based on decomposition (MOEA/D), an enhanced algorithm with adaptive neighborhood size and genetic operator selection, named self-adaptive MOEA/D (SaMOEA/D), is developed for solving LS-MOP in this work. Learning from the search history, each scalar optimization subproblem in SaMOEA/D varies its neighborhood size and selects a genetic operator adaptively. The former determines the size of the search scope, while the latter determines the search behavior and as a result the newly generated solution. Experimental results on 20 LS-MOP benchmarks have demonstrated that SaMOEA/D outperforms or performs similarly to the other four state-of-the-art MOEAs. The effectiveness of the self-adaptive strategies has also been experimentally verified. Furthermore, SaMOEA/D and the comparing algorithms are then applied to solve a challenging real-world problem, the multi-objective reservoir flood control operation problem. Optimization results illustrate the superiority of SaMOEA/D.

1. Introduction

A multi-objective optimization problem (MOP) [1] can be described as:

$$\begin{array}{ll} \text{Minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to:} & \mathbf{x} \in \Omega \end{array} \tag{1}$$

where $\mathbf{F}: \Omega \to \mathbb{R}^m$ consists of m real-valued conflicting objective functions. $\Omega \subset \mathbb{R}^n$ indicates the feasible region of decision vector. $\mathbf{x} \in \Omega$ is called decision vector.

Let $\mathbf{x}_A, \mathbf{x}_B \in \Omega$. $\mathbf{x}_A$ dominates $\mathbf{x}_B$ (denoted as $\mathbf{x}_A \prec \mathbf{x}_B$) if and only if $f_i(\mathbf{x}_A) \le f_i(\mathbf{x}_B)$, $\forall i \in \{1, \cdots, m\}$ and $\mathbf{F}(\mathbf{x}_A) \ne \mathbf{F}(\mathbf{x}_B)$. A solution $\mathbf{x}^* \in \Omega$ is called a Pareto optimal solution if and only if there is no solution that can dominate it. The set of

* Corresponding author. Tel.: +8613572932245.
E-mail address: ytqi@xidian.edu.cn (Y. Qi).

http://dx.doi.org/10.1016/j.ins.2016.06.005
0020-0255/

all Pareto optimal solutions is defined as Pareto optimal set (PS). $PS = \{\mathbf{x}^* \mid \neg\exists \mathbf{x} \in \Omega, \mathbf{x} \prec \mathbf{x}^*\}$. And the corresponding Pareto optimal set on the objective space is called Pareto optimal front (PF). $PF = \{\mathbf{F}(\mathbf{x}) \mid \mathbf{x} \in PS\}$. Usually, it is very time-consuming or even impossible to obtain the complete PF of the target MOP. Therefore, multi-objective optimization algorithms for MOP aim at finding a finite number of Pareto optimal solutions whose images in the objective space evenly scattered along the entire PF.

Due to the capability of generating a set of representative and approximate solutions in a single run, evolutionary algorithms (EAs) have been recognized as competent solvers for MOP. In recent years, many multi-objective evolutionary algorithms (MOEAs) for solving MOP have been developed. According to the different selection mechanisms employed, MOEAs can be divided into three main categories, the Pareto-dominance-based MOEAs [2,3], the indicator-based MOEAs [4,5] and the decomposition-based MOEAs [6]. Most recently, MOEAs that combines Pareto-dominance-based and decomposition-based selection mechanisms began to appear [7], which can be classified as a new category.

Among existing MOEAs, the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [6] has been regarded as one of the most efficient algorithms, especially for solving complex MOP. Due to its simplicity and efficiency, MOEA/D has attracted increasing research interests and developed into a general framework for MOEAs. Research works on MOEA/D generally falls into several aspects. For example, combining MOEA/D with other nature inspired meta-heuristics [8–11], investigating on the decomposition approaches [12–14], refining the weight vectors for scalar optimization subproblems [15–17], changing the offspring reproduction methods in MOEA/D [18–20], applying MOEA/D on benchmark and real-world problems [21–24], and so on. In MOEA/D, the target MOP is decomposed into a set of scalar optimization subproblems by using conventional aggregation approaches. Then, an evolutionary algorithm is employed to optimize the decomposed subproblems simultaneously. Each subproblem is optimized in MOEA/D by using information largely from its neighbouring subproblems. In other words, neighboring subproblems collaborate with each other and improve the quality of the population. Therefore, it is a key issue for the MOEA/D algorithm to determine which subproblems should be cooperative partners and how one subproblem can collaborate with another. These are what we will study in this paper.

Large-scale multi-objective optimization problems (LS-MOP) are complex MOPs with a large number of decision variables. Because of its high-dimensional decision space, LS-MOP poses a significant challenge to multi-objective optimization methods including MOEAs. When dealing with large-scale optimization problems, the divide-and-conquer paradigm is commonly used. Its basic idea is to decompose the original large-scale problem into a series of smaller and simpler subproblems which are easier to solve. With such a decomposition, the whole problem can be solved by optimizing the individual subproblems independently [25]. One of the major issue in applying the divide-and-conquer paradigm is the choice of a good decomposition strategy. At present, there are two types decomposition methods for LS-MOP. One of which performs decomposition in the decision space by decomposing decision vectors into smaller components [26], while the other one performs decomposition in the objective space by decomposing a MOP into scalar optimization subproblems [6] or simple multiobjective subproblems [27].

The decomposition method based on the decision space is the most widely used one, it has achieved significant success in solving large-scale single-objective optimization problems (LS-SOP) [25]. Approaches using such decomposition method are commonly referred to as cooperative co-evolutionary (CC) algorithms. In these CC algorithms, the decision variables are grouped into several subcomponents by using various methods. As the interdependence between variables can significantly affect the performance of optimizers, interacting variables are expected to be optimized in the same subcomponent. After the subcomponents which have few or no interdependencies on each other are identified, they undergo optimization using evolutionary optimizers. The CC algorithm has been proved to be effective for solving large-scale single-objective optimization problems, however, its performance decreases sharply when dealing with problems with strong interdependence between decision variables. Unfortunately, linkages between decision variables are stronger in a MOP than in a single-objective one, because two independent decision variables on one objective might be interdependent on another objective. Therefore, CC algorithms may not perform well on LS-MOP as they have been on LS-SOP. To our knowledge, few works have been done on solving LS-MOP by using CC algorithms.

The MOEA/D [6] and MOEA/D-M2M [27] provide another decomposition method based on the objective space. By utilizing the regularity of MOP, they decompose the target MOP into several scalar optimization subproblems or simple multi-objective subproblems, and then solve these subproblems simultaneously by using evolutionary optimizers. Such decomposition method has been proved to be more efficient on solving MOP. However, in MOEA/D and MOEA/D-M2M, the decomposed subproblems are optimized by using the same evolutionary optimizers, which degrades the performances of the algorithm on solving LS-MOP because each subproblem has its own peculiarity. To remedy this, this paper proposes that those optimizers for decomposed subproblems can be endowed with self-adaptability by taking advantage of information learned from the evolutionary history.

For solving single-objective optimization problems, it is a commonly used and efficient technique to enhance the adaptivity of an evolutionary algorithm by taking advantage of information learned from the evolutionary history of the population to further improve the performance. However, it is not an easy job for the Pareto-dominance-based MOEAs and the indicator-based MOEAs to measure such fitness improvement [20]. Since MOEA/D decomposes the target MOP into scalar optimization subproblems, the information of evolutionary experiences can be easily obtained from the improvements of subproblems' objective functions. In this work, the information abstracted from evolutionary history of the population is utilized to improve the adaptivity of MOEA/D. An adaptive strategy that considers the adaptiveness of both neighborhood size and genetic operators is developed and integrated with MOEA/D to enhance its performance on solving LS-MOP.