Swarm and Evolutionary Computation 22 (2015) 1–14

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

A genetic algorithm for unconstrained multi-objective optimization

Qiang Long [a], Changzhi Wu [b,*], Tingwen Huang [c], Xiangyu Wang [b,d]

[a] School of Science, Southwest University of Science and Technology, Mianyang, China
[b] Australasian Joint Research Centre for Building Information Modelling, School of Built Environment, Curtin University, Australia
[c] Texas A&M University at Qatar, Doha, P.O.Box 23874, Qatar
[d] Department of Housing and Interior Design, Kyung Hee University, Seoul, South Korea

ARTICLE INFO

Article history:
Received 29 August 2014
Received in revised form
18 November 2014
Accepted 19 January 2015
Available online 28 January 2015

Keywords:
Genetic algorithm
Optimal sequence method
Multi-objective optimization
Numerical performance evaluation

ABSTRACT

In this paper, we propose a genetic algorithm for unconstrained multi-objective optimization. Multi-objective genetic algorithm (MOGA) is a direct method for multi-objective optimization problems. Compared to the traditional multi-objective optimization method whose aim is to find a single Pareto solution, MOGA tends to find a representation of the whole Pareto frontier. During the process of solving multi-objective optimization problems using genetic algorithm, one needs to synthetically consider the fitness, diversity and elitism of solutions. In this paper, more specifically, the optimal sequence method is altered to evaluate the fitness; cell-based density and Pareto-based ranking are combined to achieve diversity; and the elitism of solutions is maintained by greedy selection. To compare the proposed method with others, a numerical performance evaluation system is developed. We test the proposed method by some well known multi-objective benchmarks and compare its results with other MOGASs'; the result show that the proposed method is robust and efficient.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

In this paper, we consider the following multi-objective optimization problem:

$$\text{(MOP)}\quad \begin{cases} \text{Minimize} & F(x) \\ \text{Subject to} & x \in X, \end{cases} \tag{1}$$

where $F(x) = (f_1(x), f_2(x), \ldots, f_p(x))^T$ is a vector-valued function, $X = \{x \in \mathbb{R}^n : l_b \le x \le u_b\} \subset \mathbb{R}^n$ is a box set, l_b and u_b are lower and upper bounds, respectively. We suppose that $f_i(x), i = 1, 2, \ldots, p$ are Lipschitz continuous but not necessarily differentiable.

Multi-objective optimization has extensive applications in engineering and management [2,28,29]. Most of the optimization problems appearing in real-world applications have multiple objectives; they can be modeled as multi-objective optimization problems. However, due to the theoretical and computational challenges, it is not easy to solve multi-objective optimization problems. Therefore, multi-objective optimization attracts lots of researches over the last few decades.

So far, there are two types of methods to solve multi-objective optimization problems: indirect and direct methods. The indirect method converts multiple objectives into a single one. One strategy is to combine the multiple objective functions using the utility theory or the weighted sum method. The difficulties for such methods are the selection of utility function or proper weights so as to satisfy the decision-maker's preferences, and furthermore, the greatest deficiency of the (linear) weighted sum method is that we cannot obtain the concave part of the Pareto frontier. Another indirect method is to formulate the multiple objectives, except one, as constraints. However, it is not easy to determine the upper bounds of these objectives. On the one hand, small upper bounds could exclude some Pareto solutions; on the other hand, large upper bounds could enlarge the objective function value space which leads to some sub-Pareto solution. Additionally, indirect method can only obtain a single Pareto solution in each run. However, in practical applications, decision-makers often prefer a number of Pareto solutions so that they can choose one strategy according to their preferences.

Direct methods devote themselves to explore the entire set of Pareto solutions or a representative subset. However, it is extremely hard or impossible to obtain the entire set of Pareto solutions for most multi-objective optimization problems, except some simple cases. Therefore, stepping back to a representative subset is preferred. Genetic algorithm (GA), as a population-based algorithm, is a good choice to achieve this goal. The generic single-objective genetic algorithm can be modified to find a set of multiple non-dominated solutions in a single run. The ability of the genetic algorithm to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems. The crossover operator of the genetic

* Corresponding author.
E-mail address: C.Wu@exchange.curtin.edu.au (C. Wu).

http://dx.doi.org/10.1016/j.swevo.2015.01.002
2210-6502/© 2015 Elsevier B.V. All rights reserved.

algorithm can exploit structures of good solutions with respect to different objectives, which in return, creates new non-dominated solutions in unexplored parts of the Pareto frontier. In addition, multi-objective genetic algorithm does not require user to prioritize, scale, or weight objectives. Therefore, the genetic algorithm is one of the most popular metaheuristic approaches for solving multi-objective optimization problems [18,24,36].

The first multi-objective optimization method based on the genetic algorithm, called the vector evaluated GA (or VEGA), was proposed by Schaffer [35]. Afterwards, several multi-objective evolutionary algorithms were developed, such as Multi-objective Genetic Algorithm (MOGA) [6], Niched Pareto Genetic Algorithm (WBGA) [15], Weight-based Genetic Algorithm (WBGA) [13], Random Weighted Genetic Algorithm (RWGA) [31], Nondominated Sorting Genetic Algorithm (NSGA) [38], Strength Pareto Evolutionary Algorithm (SPEA) [52], improved SPEA (SPEA2) [51], Pareto-Archived Evolution Strategy (PAES) [21], Pareto Envelope-based Selection Algorithm (PESA) [3], Nondominated Sorting Genetic Algorithm-II (NSGA-II) [5], Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [1,46,47] and Indicator-Based Evolutionary Algorithm (IBEA) [34].

There are three basic issues [50] in solving multi-objective optimization problems using the genetic algorithm:

1. *Fitness:* Solutions whose objective values are close to the real Pareto frontier should be selected as parents of the next generation. This gives rise to the task of ranking candidate solutions in each generation.
2. *Diversity:* The obtained subset of Pareto solutions should distribute uniformly over the real Pareto frontier. This reveals the true tradeoff of the multi-objective optimization problem for decision-makers.
3. *Elitism:* The best candidate solution is always kept to the next generation in solving single-objective optimization problems using the genetic algorithm. We extend this idea to the multi-objective case. However, this extension is not straightforward since a large number of candidate solutions are involved in multi-objective genetic algorithm.

The aim of this paper is to introduce new techniques to tackle the issues mentioned above. The rest of the paper is organized as follows. In Section 2, we review some basic definitions of multi-objective optimization and the process of genetic algorithm. In Section 3, we propose an improved genetic algorithm for multi-objective optimization problems. In Section 4, an evaluation system for numerical performance of multi-objective genetic algorithms is developed. Some simulation studies are carried out in Section 5. Section 6 concludes the paper.

2. Preliminaries

In this section, we first review some definitions and theorems in the multi-objective optimization, and then introduce the general procedure of genetic algorithm.

2.1. Definitions in multi-objective optimization

First of all, we present the following notations which are often used in vector optimization. Given two vectors

$$x = (x_1, x_2, \ldots, x_n)^T \quad \text{and} \quad y = (y_1, y_2, \ldots, y_n)^T \in \mathbb{R}^n,$$

then

- $x = y \Leftrightarrow x_i = y_i$ for all $i = 1, 2, \ldots, n$;
- $x < y \Leftrightarrow x_i < y_i$ for all $i = 1, 2, \ldots, n$;
- $x \le y \Leftrightarrow x_i \le y_i$ for all $i = 1, 2, \ldots, n$, and there is at least one $i \in \{1, 2, \ldots, n\}$ such that $x_i < y_i$, i.e., $x \ne y$.
- $x \leqq y \Leftrightarrow x_i \le y_i$ for all $i = 1, 2, \ldots, n$.

" $>$ ", " $\ge$ " and "$\geqq$" can be defined similarly. In this paper, we call $x \le y$ x dominates y or y is dominated by x.

Definition 2.1. Suppose that $x \subseteq \mathbb{R}^n$ and $x^* \in X$. If $x^* \leqq x$ for any $x \in X$, then x^* is called an *absolute optimal point* of X.

Absolute optimal point is an ideal point but it may not exist.

Definition 2.2. Let $x \in \mathbb{R}^n$ and $x^* \in X$. If there is no $x \in X$ such that

$$x \le x^* (\text{or } x < x^*),$$

then x^* is called an efficient point (or weakly efficient point).

The sets of absolute optimal points, efficient points and weakly efficient points of X are denoted as X_{ab}, X_{ep} and X_{wp}, respectively. For the problem MOP, $X \subseteq \mathbb{R}^n$ is called the *decision variable space* and its image set $F(X) = \{y \in \mathbb{R}^p | y = F(x), x \in X\} \subset \mathbb{R}^p$ is called the *objective function value space*.

Definition 2.3. Suppose that $x^* \in X$. If

$$F(x^*) \leqq F(x),$$

for any $x \in X$, x^* is called an *absolute optimal solution* of the problem MOP. The set of absolute optimal solution is denoted as S_{as}.

The concept of the absolute optimal solution is a direct extension of that for single-objective optimization. It is the ideal solution but may not exist for most cases.

Definition 2.4. Suppose that $x^* \in X$. If there is no $x \in X$ such that

$$F(x) \le F(x^*) (\text{or } F(x) < F(x^*)),$$

i.e., $F(x^*)$ is an efficient point (or weakly efficient point) of the objective function value space $F(X)$, then x^* is called an *efficient solution* (or *weakly efficient solution*) of the problem MOP. The sets of efficient solutions and weakly efficient solutions are denoted as S_{es} and S_{ws}, respectively.

Another name of the efficient solution is *Pareto solution*, which was introduced by T.C. Koopmans in 1951 [22]. The meaning of Pareto solution is that, if $x^* \in S_{es}$, then there is no feasible solution $x \in X$, such that any $f_i(x)$ of $F(x)$ is not worse than that of $F(x^*)$. In other words, x^* is the best solution in the sense of " $\le$ ". Another intuitive interpretation of Pareto solution is that it cannot be improved with respect to any objective without worsening at least one of the other objectives. The set of Pareto solutions is denoted by $\mathcal{P}^*$. Its image set $F(\mathcal{P}^*)$ is called the *Pareto frontier*, denoted by $\mathcal{PF}^*$.

2.2. Genetic algorithm

Genetic algorithm is one of the most important evolutionary algorithms. It was introduced by John Holland in 1960s, and then developed by his students and colleagues at the University of Michigan between 1960s and 1970s [14]. Over the last two decades, the genetic algorithm was increasingly enriched by plenty of literatures, such as [9,10,12,19]. Nowadays various genetic algorithms are applied in different areas; for example, mathematical programming, combinational optimization, automatic control and image processing.

Suppose that $P(t)$ and $O(t)$ represent parents and offspring of the tth generation, respectively. Then, the general structure of genetic algorithm can be described in the following pseudocode.

General structure of genetic algorithm.

Download English Version:

https://daneshyari.com/en/article/493670

Download Persian Version:

https://daneshyari.com/article/493670

Daneshyari.com