Computers & Operations Research 40 (2013) 282–302

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

BSTBGA: A hybrid genetic algorithm for constrained multi-objective optimization problems

Xiang Li*, Gang Du

School of Management, Tianjin University, Tianjin 300072, People's Republic of China

ARTICLE INFO

Available online 22 July 2012

Keywords:
Multi-objective optimization
Constrained multi-objective optimization
Inequality constraint
Constraint handling
Genetic algorithms
Boundary simulation method
Binary search method
Population diversity
Pareto optimum
Pareto set
Pareto front
Trie-tree
Rtrie-tree
Atrie-tree

ABSTRACT

Most of the existing multi-objective genetic algorithms were developed for unconstrained problems, even though most real-world problems are constrained. Based on the boundary simulation method and trie-tree data structure, this paper proposes a hybrid genetic algorithm to solve constrained multi-objective optimization problems (CMOPs). To validate our approach, a series of constrained multi-objective optimization problems are examined, and we compare the test results with those of the well-known NSGA-II algorithm, which is representative of the state of the art in this area. The numerical experiments indicate that the proposed method can clearly simulate the Pareto front for the problems under consideration.

1. Introduction

Many real-world problems involve the simultaneous optimization of several intrinsically conflicting objectives. For example, when designing a software system, we always hope to reduce development costs and to improve the performance, stability, scalability and re-usability of the final products. Generally, there is no single perfect solution that satisfies all the objectives simultaneously because they are intrinsically in conflict with each other. Mathematically, this type of problem can be formulated as a multi-objective optimization problem (MOP).

Traditional mathematical programming techniques have some limitations when solving MOPs. Most of them depend on the shape of the Pareto front and only generate one Pareto solution from each run. Thus, several runs (with different parameter settings) are generally required to generate a Pareto solution set; however, sometimes different parameter settings may generate similar results. In such circumstances, generating a Pareto solution set will be very computationally expensive.

Genetic algorithms (GAs) are a robust and efficient optimization technique based on the mechanism of natural selection and natural genetics [1]. One of the important features of GAs is that they are a population-based search technique. Instead of moving from one single point to another like traditional mathematical programming techniques, GAs always maintain and manipulate a solution set (population). This feature makes it possible to generate a Pareto solution set in a single run. Furthermore, GAs work on the function evaluation alone, which means no other information about the problem under consideration is required. Because of these advantages, GAs were recognized as potentially well-suited to MOPs. In addition, most of the existing multi-objective optimization techniques are based on GAs [24], and MOPs may be an area where GAs can distinguish themselves from other competitors [18].

However, GAs also have some limitations. GAs essentially are an unconstrained optimization technique. In other words, GAs do not have any explicit constraint-handling mechanism. When GAs are applied to constrained optimization problems, the traditional genetic search operators (e.g., crossover and mutation) may produce infeasible points (individuals). Therefore, a major research issue is how to handle the constraints when applying GAs to solve constrained optimization problems. In our previous research, we proposed a boundary simulation method to solve constrained single-objective optimization problems (CSOPs) [69]. In this paper, we combine the boundary simulation method with two specially designed trie-like data structures to form a hybrid genetic algorithm to solve constrained multi-objective optimization problems (CMOPs).

The structure of this paper is as follows. Section 2 provides some basic concepts and definitions. Section 3 reviews the literature, providing a brief introduction to status of current researches on this

* Corresponding author.
E-mail address: tju.som.li.xiang@gmail.com (X. Li).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.07.014

topic and briefly describing the boundary simulation method and the trie-tree data structure. Section 4 describes the proposed rtrie-tree data structure and the rtrie-based initialization and selection operators. Section 5 proposes an atrie-tree data structure and an atrie-based archive operator that can be used to efficiently maintain the archive (external population). Section 6 proposes a new hybrid genetic algorithm for CMOPs. In Section 7, the proposed method is applied to a series of test problems to investigate its feasibility and efficiency. Finally, our conclusions and some possible directions for future research are presented in Section 8.

2. Basic concepts and definitions

In this study, we concentrate on CMOPs which can be represented as follows:

$$\text{Minimize} \quad f(x) = (f_1(x), f_2(x), \ldots, f_k(x))$$
$$\text{Subject to} \quad \begin{cases} g_j(x) \le 0, & j = 1,2,\ldots,m, \\ x_i^l \le x_i \le x_i^u, & i = 1,2,\ldots,n, \end{cases} \tag{1}$$

where $f(x) = (f_1(x), f_2(x), \ldots f_k(x))$ is the objective functions, $g_j(x)$ is the j-th inequality constraint, $x = (x_1, x_2, \ldots, x_n)$ is the decision vector, x_i^l and x_i^u is the lower and upper bound of the decision variable x_i, respectively.

It is important to emphasize the following assumptions.

- The feasible region is connected.
- Only inequality constraints are involved.
- All decision variables have upper and lower bounds.

Generally, there are three types of constraints: linear inequality, nonlinear inequality and nonlinear equality constraints, as linear equality constraints can be easily converted to and added into other type constraints. In the remainder of this paper, "inequality constraints" will refer to linear and nonlinear inequality constraints, while "equality constraints" will only refer to nonlinear equality constraints.

For single-objective optimization problems (SOPs), the optimal solution is clearly defined; however, this is not the case for MOPs. Unlike SOPs, MOPs generally have not one but a set of compromise solutions that are equally good in some sense (called the Pareto solution). Consider the general model equation (1), with the following definitions.

Definition 1. The *feasible region* is defined by the following expression:

$$\mathcal{F} = \left\{ x = (x_1, x_2, \ldots, x_n) \,\middle|\, \begin{array}{ll} g_j(x) \le 0, & j = 1,2,\ldots,m, \\ x_i^l \le x_i \le x_i^u, & i = 1,2,\ldots,n. \end{array} \right\} \tag{2}$$

Definition 2. Given two vectors x and y, we say that $x \le y$ if $x_i \le y_i$ for $i = 1, \ldots, n$. If $x \le y$ and $x \ne y$ then x *dominates* y (denoted by $x \prec y$).

The classical Pareto dominance definition and related definitions are as follows.

Definition 3. We say that a vector of decision variables $x \in \mathcal{X}$ is *non-dominated* with respect to $\mathcal{X}$, if there does not exist another $x' \in \mathcal{X}$ such that $f(x') \prec f(x)$.

Definition 4. We say that a vector of decision variables $x^* \in \mathcal{F}$ is the *Pareto optimal*, if it is non-dominated with respect to $\mathcal{F}$.

Definition 5. The *Pareto optimal set* $\mathcal{P}^*$ is defined by

$$\mathcal{P}^* = \{x \in \mathcal{F} \,|\, x \text{ is the Pareto optimal}\}. \tag{3}$$

Definition 6. The *Pareto front* $\mathcal{PF}^*$ is defined by

$$\mathcal{PF}^* = \{f(x) \,|\, x \in \mathcal{P}^*\}. \tag{4}$$

Based on these classical definitions discussed above, we propose the following definition of *reasonable Pareto dominance*.

Definition 7. Given two vectors x and y, we say that x *reasonably dominates* y (denoted by $x \prec' y$) if the following conditions are satisfied.

- x and y do not dominate each other according to the classical Pareto dominance definition.
- There exists at least one i such that $f(y)_i - f(x)_i > \alpha_1 > 0$, and for other $i = 1, \ldots, k$, $|f(x)_i - f(y)_i| < \alpha_2$,

where α_1 and α_2 are the reasonable parameters that can be defined by decision makers according to the specific application. The physical meanings of the reasonable dominance definition and the reasonable parameters are clear. If x *reasonably dominates* y ($x \prec' y$), then x and y do not dominate each other; in at least one respect y is worse than x, while in all other respects x and y are similar. For example, in Fig. 1, a, b and c do not dominate each other according to the classical Pareto dominance definition; however, *a reasonably dominates b* and c; in addition, we say *a reasonably dominates b* and c in the y_2 and y_1 directions, respectively. Furthermore, all the points falling in the shallows are reasonably dominated by a.

In most cases, b and c should be *reasonably ignored* by decision makers, when comparing a, b and c, even though a does not dominate b and c according to the classical Pareto dominance definition. Furthermore, we can easily define the *reasonable Pareto optimal set* and the *reasonable Pareto front* based on the reasonable dominance definition. Due to limitations of space and their simplicity, these derivative definitions are omitted.

A Pareto optimal set is a set of solutions that are non-dominated with respect to each other. Moving from one Pareto optimal solution to another always implies making a certain sacrifice in some objectives to achieve a certain improvement in other objectives. The ultimate goal of a multi-objective optimization algorithm is to identify the solutions in the Pareto optimal set. However, for most MOPs, identifying the entire Pareto optimal set is practically impossible due to its size. A more practical computation strategy is to generate a non-dominated solution set to approximate the Pareto optimal set.

With these concerns in mind, a multi-objective optimization algorithm should achieve the following objectives.

- The resulting non-dominated solution set should converge at the Pareto front.

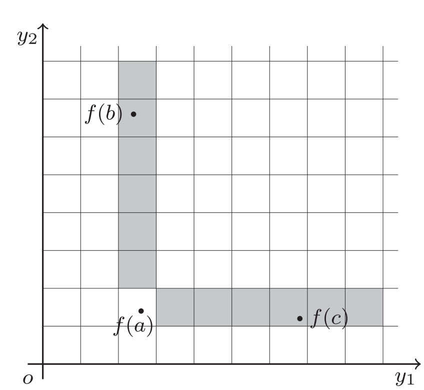

Fig. 1. Reasonable dominance relationship.