Swarm and Evolutionary Computation 3 (2012) 1–14

Contents lists available at SciVerse ScienceDirect

Swarm and Evolutionary Computation

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

Local search based hybrid particle swarm optimization algorithm for multiobjective optimization

A.A. Mousa [a,c,*], M.A. El-Shorbagy [a], W.F. Abd-El-Wahed [b]

[a] *Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Egypt*

[b] *Faculty of Computers and Information, Menoufia University, Egypt*

[c] *Department of Mathematics, Faculty of Sciences, Taif University, Saudi Arabia*

ARTICLE INFO

Article history:
Received 9 December 2010
Received in revised form
29 November 2011
Accepted 29 November 2011
Available online 8 December 2011

Keywords:
Particle swarm optimization
Genetic algorithm
Local search
Constriction factor
Multiobjective optimization

ABSTRACT

In this paper, we propose a hybrid multiobjective evolutionary algorithm combining two heuristic optimization techniques. Our approach integrates the merits of both genetic algorithm (GA) and particle swarm optimization (PSO), and has two characteristic features. Firstly, the algorithm is initialized by a set of random particles which is flown through the search space. In order to get approximate nondominated solutions PND, an evolution of this particle is performed. Secondly, the local search (LS) scheme is implemented as a neighborhood search engine to improve the solution quality, where it intends to explore the less-crowded area in the current archive to possibly obtain more nondominated solutions. Finally, various kinds of multiobjective (MO) benchmark problems including the set of benchmark functions provided for CEC09 have been reported to stress the importance of hybridization algorithms in generating Pareto optimal sets for multiobjective optimization problems.

1. Introduction

Multiobjective optimization (MO) is an important research topic for both scientists and engineers. In MO, a set of nondominated solutions is usually produced instead of single recommended solution. According to the concept of nondominance, also referred to as Pareto optimality, a solution to an MO problem is nondominated, or Pareto optimal, if no objective can be improved without worsening at least one other objective [1].

Traditional MO methods attempt to find the set of nondominated solutions using mathematical programming. In the case of nonlinear problems, the weighting method and the ε-constraint method are the most commonly used techniques [2]. Both methods transform the MO problem into a single objective problem which can be solved using nonlinear optimization. With the weighting method, nondominated solutions are obtained if all weights are positive but not all Pareto optimal solutions can be found unless all objective functions as well as the feasible region are convex. Another disadvantage of this method is that many different sets of weights may produce the same solution, compromising the efficiency of the method. When the weights reflect the preferences of the decision maker (DM), the method gives the best-compromise solution, i.e. the solution which produces the highest utility to the DM. The ε-constraint method, on the other hand, does not require convexity but only leads to a nondominated solutions if certain specific conditions are satisfied [2].

Evolutionary Algorithms (EAs) are good candidates for (MOPs) due to their abilities, to search simultaneously for multiple Pareto optimal solutions and perform better global search of the search space [3]. Among existing evolutionary algorithms, the best-known branch is the GA. GA is a stochastic search procedure based on the mechanics of natural selection, genetics and evolution [4]. Since this type of algorithm simultaneously evaluates many points in the search space, it is more likely to find the global solution of a given problem. In addition, it uses only a simple scalar performance measure that does not require or use derivative information, so methods classified as GA are easy to use and implement. The area of MO using EAs has been explored for a long time. The first multiobjective GA implementation was called the vector evaluated genetic algorithm (VEGA) [5]. Since then, many EAs for solving MOPs have been developed [1].

Recently, a lot of emphasis has been laid on enhancing evolutionary algorithms to yield a computationally efficient and convergent procedure. In [6], a Multiobjective Self-adaptive Differential Evolution algorithm with objective-wise learning strategies (OW-MOSaDE) was presented. Suitable crossover parameter values and mutation strategies are learned for each objective separately in a multiobjective optimization problem. A hybrid Archive-based

* Corresponding author at: Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Egypt.
E-mail address: A_mousa15@yahoo.com (A.A. Mousa).

2210-6502/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.swevo.2011.11.005

Micro Genetic Algorithm (AMGA) was proposed in [7]. It is a combination of a classical gradient based single-objective optimization algorithm and an evolutionary multiobjective optimization algorithm. The gradient based optimizer is used for a fast local search and is a variant of the sequential quadratic programming method. Also it is used as the global optimizer. A scalarization scheme based on the weighted objectives is proposed which is designed to facilitate the simultaneous improvement of all the objectives. It utilizes reference points as constraints to enable the algorithm to solve non-convex optimization problems. The gradient based optimizer is used as the mutation operator of the evolutionary algorithm and a suitable scheme to switch between the genetic mutation and the gradient based mutation is proposed. In [8], DMOEA-DD was presented, which is an improvement of the dynamical multiobjective evolutionary algorithm DMOEA [9]. The domain decomposition technique is used to divide the feasible domain of the decision variables into several subdomains. In each subdomain, the DMOEA is used to search the Pareto optimal solutions and each subdomain can exchange the information by genetic operators. Multiobjective evolutionary programming (MOEP) using fuzzy rank-sum with diversified selection is introduced in [10]. It is different from classical MOEP in the sorting and selection steps. The rank-sum is used to divide every objective into 100 ranks and sum the corresponding rank for all objectives. The population is assigned the rank-sum according to its position in the search space. In [11], an augmented local search based EMO procedure was rigorously proposed, where the NSGAII method [12] is used as the EMO algorithm and is hybridized with an achievement scalarizing function (ASF) which is solved with any appropriate local search method. The local search is started from an offspring solution, which is considered as a reference point. The local search utilizes this reference point and minimizes the augmented ASF to obtain at least a locally Pareto optimal solution closest to the reference point.

More recently, based upon the interaction of individual entities called "particles" Kennedy and Eberhart [13,14] proposed a new heuristic algorithm called "particle swarm optimization" (PSO). The development of this algorithm follows from observations of social behaviors of animals, such as bird flocking and fish schooling. Compared with GA, PSO has some attractive characteristics [15,16]. It has memory, so knowledge of good solutions is retained by all the particles; whereas in GA, previous knowledge of the problem is discarded once the population changes. It has constructive cooperation between particles; that is, particles in the swarm share information among themselves. Enhancing the performance of PSO consists of three categories: extension of field searching space, adjustment the parameters, and hybrid with another technique [17,18].

Since PSO cannot be directly applied to multiobjective optimization, there are two issues to be considered when applying PSO to multiobjective optimization. The first one is how to select the global and local best particles to guide the search of a particle. The second is how to maintain good points found so far. In [19], a tournament niche technique is introduced to select the global best particle, and the local best particle is updated by the Pareto dominance criteria. In [20], the particles are clustered into groups, the global best particle of a particle is from its group and a weighted sum of the objectives is used to maintain its local best particle. In [21], the global best particle is selected from the nondominated solutions using a roulette wheel selection in which the density values are used as fitness. In [22], a multiobjective PSO was designed to tackle multiobjective mixed-model assembly line sequencing problems. To this end, a coding strategy and a local search are introduced. The global best particle is the nondominated solution in the archive with the highest crowding distance in the archive. In [23], a preference order, a generalization of Pareto dominance, is introduced to rank all the particles and thus to identify the global best particle. Three hybrid PSO algorithms were proposed in [24]. The fitness assignment is based on that of strength Pareto evolutionary algorithm 2 (SPEA2) [25]. The global best particle is selected from the external archive using a tournament selection, and the neighborhood best particle is selected as the one with lowest strength Pareto fitness. A multiple swam algorithm was proposed in [26]. Several components, such as cell-based rank density estimation, population growing and declining strategies, and local search, are designed to improve the algorithmic performance. In [27], an external archive is applied to maintain the nondominated solutions found so far, and a mutation operator is used to keep the population diversity. To choose a global best particle, the nondominated ones in sparse areas are emphasized. In [28], a fuzzy clustering-based PSO was presented where; a fuzzy clustering technique is applied to maintain the external archive. A self-adaptive mutation operator is also used to generate new trial solutions. A niching mechanism is designed to find the global best particle for each particle and thus to emphasize less explored areas. Finally, a fuzzy decision rule is used to assist decision making. In [29], a multiobjective comprehensive learning particle swarm optimizer (MOCLPSO) was proposed. MOCLPSO uses a learning strategy whereby all other particles' historical best information is used to update a particle's velocity. This strategy enables the diversity of the swarm to be preserved to discourage premature convergence. In [30], a two-local-best (lbest)-based multiobjective PSO (2LB-MOPSO) technique was proposed. Different from canonical multiobjective PSO, 2LB-MOPSO uses two local bests instead of one personal best and one global best to lead each particle. The two local bests are selected to be close to each other in order to enhance the local search ability of the algorithm. Compared to the canonical multiobjective PSO, 2LB-MOPSO shows great advantages in convergence speed and fine-searching ability. In [31], PSO is used in the MOEA/D framework. Each particle is responsible for solving one subproblem. A comprehensive survey of the state-of-the-art in MOPSO can be found in [16].

The aim of this paper is to introduce a hybrid multiobjective evolutionary algorithm. It combines two heuristic optimization techniques such that it integrates the merits of genetic algorithms and particle swarm optimization. In order to improve the solution quality, a local search scheme was implemented. Local search is a metaheuristic search method for solving computationally hard optimization problems [32]. It moves from solution to solution in the space of candidate solutions (the search space) until a solution deemed optimal is found or a time bound is elapsed. In this paper, the Modified local search MLS scheme is presented, which is a modification of Hooke and Jeeves method [33] to treat multiobjective optimization. Finally, various kinds of MO benchmark problems including the set of benchmark functions provided for CEC09 "special session and competition on multiobjective optimization" have been reported to illustrate the successful result in finding a Pareto optimal set.

The remainder of the paper is organized as follows. In Section 2, multiobjective optimization is described. Section 3, provides an overview of the PSO and GA. In Section 4, the proposed algorithm is presented. Experimental results are discussed in Section 5. Finally, Section 6 presents our conclusion and notes for future work.

2. Multiobjective optimization (MO)

A general minimization problem of M objectives can be mathematically stated as:

$$\left.\begin{array}{l}\text{Minimize} : \vec{f}\left(\vec{x}\right) = \left[f_i\left(\vec{x}\right), \quad i = 1, 2, \ldots, M\right] \\ \text{subject to the constraints} : g_j\left(\vec{x}\right) \leq 0, \quad j = 1, 2, \ldots, J\end{array}\right\} \tag{1}$$

given $\vec{x} = [x_1, x_2, \ldots, x_n]$, where n represents the dimension of the decision variable space, $f_i\left(\vec{x}\right)$ is the i-th objective function, $g_j\left(\vec{x}\right)$