Elitist-Mutated Multi-Objective Particle Swarm Optimization for Engineering Design

M. Janga Reddy
Indian Institute of Technology Bombay, India

D. Nagesh Kumar
Indian Institute of Science, India

INTRODUCTION

Most of the real world problems are characterized by multiple goals, which often conflict and compete with one another. Multi-objective optimization problems (MOOPs) require the simultaneous optimization of several non-commensurable and often competitive/conflicting objectives. Because of the multiple conflicting objectives, it is not possible to find a single optimal solution, which will satisfy all the goals. Instead, the solution exists in the form of alternative trade-offs, also known as the non-dominated solutions. In the past, several researchers have used classical optimization techniques such as linear programming (LP), dynamic programming (DP) and non-linear programming (NLP) to solve the multi-objective problems by adopting weighted approach or constrained approach. These methods may face difficulties while generating optimal solutions for practical problems. For example, in the weighted sum method, the multiple objectives of the problem will be converted into a single objective optimization by adopting suitable weights to objectives. By using a single pair of fixed weights, only one point on Pareto front can be obtained. Therefore, if one would like to obtain the complete Pareto front, all possible Pareto solutions must first be derived. This requires the algorithms to be executed iteratively, so as to ensure that every weight combination has been evaluated. Obviously, it is impractical to reiterate the algorithms continually to exhaust all the weight combinations. Similarly, in the constraint method, it needs to reiterate the algorithm for a large number of times, which requires more computational effort. Also conventional methods may face problems, if optimal solution lies on non-convex or disconnected regions of the objective function space. Thus the classical methods are not ideal approaches to solve multi-objective optimization problems (MOOP). In developing an algorithm for solution of an MOOP, it should have an ability to "learn" from previous performance, to direct proper selection of weights for further evolutions. To achieve the above goals, multi-objective evolutionary algorithms (MOEAs) have been proposed and are suggested as effective means to handle these issues (Reddy & Kumar, 2007a). Due to their efficiency and easiness to handle non-linear relationships, ability to approximate the non-convex and disconnected Pareto optimal fronts of real-world problems, MOEAs are getting diverse applications in various fields. Apart from that, the specific advantage of MOEAs over the classical optimization techniques is that they generate a population of solutions in each iteration and offer a set of alternatives (Pareto optimal set) in a single run. Thus population based stochastic search techniques are becoming more popular to solve multi-objective optimization problems. This article first discusses the principles and issues in developing MOEAs, then presents an effective multi-objective optimization algorithm based on swarm intelligence principle.

BACKGROUND

Multi-Objective Problem

Definition: A general multi-objective optimization problem (MOOP) can be defined as, minimize a set of functions $f(x)$, subject to p inequality and q equality constraints.

DOI: 10.4018/978-1-4666-5888-2.ch346

H

$$Min.\ \ f(x) = \{ f_1(x)\, f_2(x) \ldots\ f_m(x) \}^T \qquad x \in D \tag{1}$$

where $x \in R^n$, $f_i : R^n \rightarrow R$ and

$$D = \begin{cases} x \in R^n : & l_i \le x \le u_i\,, \quad \forall\ i = 1, \ldots, n \\ & g_j(x) \ge 0\,, \quad \forall\ j = 1, \ldots, p \\ & h_k(x) = 0\,, \quad \forall\ k = 1, \ldots, q \end{cases} \tag{2}$$

where *m* is number of objectives; *D* is feasible search space; $x = \{x_1\ x_2 \ldots x_n\}^T$ is the set of *n*-dimensional decision variables (continuous, discrete or integer); *R* is the set of real numbers; R^n is *n*-dimensional hyperplane or space; l_i and u_i are lower and upper limits of *i*-th decision variable.

In MOOP, the desired goals are often conflicting against each other and it is not possible to satisfy all the goals at a time. Hence it gives a set of non-inferior solutions also known as *Pareto optimal* solutions. The Pareto optimal solution refers to a solution, around which there is no way of improving any objective without degrading at least one other objective (Deb et al., 2002).

Pareto Front: Pareto front is a set of nondominated solutions, being chosen as optimal, if no objective can be improved without sacrificing at least one other objective. On the other hand a solution *x** is referred to as dominated by another solution *x,* if and only if, *x* is equally good or better than *x** with respect to all objectives. The definition of Pareto optimality is very much useful in MOEAs to classify the population of solutions into dominated and non-dominated members, thereby helping in the selection of member solutions from one generation to next generation.

Main Issues in MOEA

Achieving a well-spread and global Pareto optimal front is the primary goal in solving MOOPs. In MOEAs, apart from finding a non-dominated solution in each generation, more computational effort is required for diversity preserving mechanisms. This computational complexity is directly related to the level of diversity and distribution, which the particular MOEA aims to obtain. The major issues in MOEA development are (Reddy & Kumar, 2007a): (i) how to guide the randomly distributed population towards the true Pareto optimal front; (ii) how to maintain good diversity in the generated non-dominated solutions. (iii) how to avoid the loss of obtained quality non-dominated solutions over the generations. These issues are depicted in Figure 1. A good MOEA should address all these issues.

Multiobjective Optimization Using Evolutionary Algorithms

During the last decade, a number of evolutionary algorithms (EAs) were suggested to solve multi-objective optimization problems. The first generation MOEAs, Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas & Deb, 1994) and Niched Pareto Genetic Algorithm (NPGA) (Horn et al., 1994) have received good recognition at the beginning in 1990s. A brief comparison of various MOEAs is presented in Zitzler et al. (2000). Recently elitist multi-objective evolutionary algorithms were found to be more efficient than those without elitism, since the elitism helps to preserve the best solutions in the past iterations and speeds up the convergence of the solution. Of them, the second generation MOEAs, Pareto-Archived Evolution Strategy (PAES) (Knowles & Corne, 2000), Strength-Pareto Evolutionary Algorithm (SPEA) (Zitzler & Thiele, 1999) and Non-dominated Sorting Genetic Algorithm-II (NSGA-II) (Deb et al., 2002) are popular due to their efficiency in producing better Pareto front. Later on, many developments have been proposed for MOEAs over the years.

Studies on Multiobjective Optimization Using PSO

The PSO algorithm has proven capabilities of quick convergence to optimal solution for single objective problems (Kumar & Reddy, 2007). The similarities of Particle Swarm Optimization (PSO) with EAs inspired the researchers to extend the algorithm to handle multiple objectives. The PSO algorithm maintains population of solutions, which allows exploration of different parts of the Pareto front simultaneously. By incorporating Pareto-dominance principle into PSO algorithm, various Multi-objective Particle Swarm Optimization (MOPSO) techniques are formulated. Ray and Liew (2002) proposed swarm metaphor approach, which uses