Chapter 6
A Particle Swarm Optimizer for Constrained Multiobjective Optimization

Wen Fung Leong
Kansas State University, USA

Yali Wu
Xi'an University of Technology, China

Gary G. Yen
Oklahoma State University, USA

ABSTRACT

Generally, constraint-handling techniques are designed for evolutionary algorithms to solve Constrained Multiobjective Optimization Problems (CMOPs). Most Multiojective Particle Swarm Optimization (MOPSO) designs adopt these existing constraint-handling techniques to deal with CMOPs. In this chapter, the authors present a constrained MOPSO in which the information related to particles' infeasibility and feasibility status is utilized effectively to guide the particles to search for feasible solutions and to improve the quality of the optimal solution found. The updating of personal best archive is based on the particles' Pareto ranks and their constraint violations. The infeasible global best archive is adopted to store infeasible nondominated solutions. The acceleration constants are adjusted depending on the personal bests' and selected global bests' infeasibility and feasibility statuses. The personal bests' feasibility statuses are integrated to estimate the mutation rate in the mutation procedure. The simulation results indicate that the proposed constrained MOPSO is highly competitive in solving selected benchmark problems.

INTRODUCTION

In real-world applications, most optimization problems are subject to various types of constraints. These problems are known as the constrained optimization problems (COPs) or constrained multiobjective optimization problems (CMOPs) if more than one objective function is involved. Comprehensive surveys (Michalewicz & Schoenauer, 1996; Mezura-Montes & CoellCoello, 2006) show

DOI: 10.4018/978-1-4666-6328-2.ch006

a variety of constraint handling techniques have been developed to address the deficiencies of evolutionary algorithms (EAs), in which, their original design are unable to deal with constraints in an effective manner. These techniques are mainly targeted at EAs, particularly genetic algorithms (GAs), to solve COPs (Runarsson & Yao, 2005; Takahama & Sakai, 2006; Cai & Wang, 2006; Oyama *et al*., 2007; Wang *et al*., 2007, 2008; Tessema & Yen, 2009) and CMOPs (Binh & Korn, 1997; Fonseca & Fleming, 1998; CoelloCoello & Christiansen, 1999; Deb *et al*., 2002; Jimenez *et al*., 2002; Kurpati *et al*., 2002; Chafekar *et al*., 2003; Ray & Won, 2005; Hingston *et al*., 2006; Geng *et al*., 2006; Zhang *et al*., 2006; Harada *et al*., 2007; Woldesenbet *et al*., 2009). During the past few years, due to the success of particle swarm optimization (PSO) in solving many unconstrained optimization problems, research on incorporating existing constraint handling techniques in PSO for solving COPs is steadily gaining attention (Parsopoulus & Vrahatis, 2002; Pulido & CoelloCoello, 2004; Zielinski & Laur, 2006; Lu & Chen, 2006; Liang & Suganthan, 2006; Wei & Wang, 2006; He & Wang, 2007; Cushman, 2007; Liu *et al*., 2008; Li *et al*., 2008;). Nevertheless, many real world problems are often multiobjective in nature. The ultimate goal is to develop multiobjective particle swarm optimization algorithms (MOPSOs) that effectively solve CMOPs. In addition to this perspective, the recent successes of MOPSOs in solving unconstrained MOPs have further motivated us to design a constrained MOPSO to solve CMOPs.

Considering a minimization problem, the general form of the CMOP with k objective functions is given as follows:

$$\text{Minimize } \mathbf{f}\left(\mathbf{x}\right) = \left[f_1\left(\mathbf{x}\right), f_2\left(\mathbf{x}\right), \ldots, f_k\left(\mathbf{x}\right)\right],$$
$$\mathbf{x} = \left[x_1, x_2, \ldots, x_n\right] \in \Re^n$$

subject to

$$g_j\left(\mathbf{x}\right) \le 0, \qquad j = 1, 2, \ldots, m; \tag{2a}$$

$$h_j\left(\mathbf{x}\right) = 0, \qquad j = m+1, \ldots, p; \tag{2b}$$

$$x_i^{\min} \le x_i \le x_i^{\max}, \quad i = 1, 2, \ldots, n \ , \tag{2c}$$

where $\mathbf{x}$ is the decision vector of n decision variables. Its upper ($x_i^{\max}$) and lower ($x_i^{\min}$) bounds in Equation (2c) define the search space, $S \subseteq \Re^n$. $g_j\left(\mathbf{x}\right)$ represents the *j*th inequality constraint, while $h_j\left(\mathbf{x}\right)$ represents the *j*th equality constraint. The inequality constraints that are equal to zero, i.e., $g_j\left(\mathbf{x}^*\right) = 0$, at the global optimum ($\mathbf{x}^*$) of a given problem are called *active constraints*. The feasible region ($F \subseteq S$) is defined by satisfying all constraints (Equations (2a)-(2b)). A solution in the feasible region ($\mathbf{x} \in F$) is called a feasible solution, otherwise it is considered an infeasible solution.

A general MOPSO algorithm consists of the five key procedures: 1) particles' flight (PSO equations), 2) particles' personal best (*pbest*) updating procedure, 3) particles' global best archive (*Gbest*) maintenance method, 4) particles' global best selection scheme, and 5) mutation operation. In the proposed design, we integrate the particles' dominance relationship, and their constraint violation information to each of these key procedures. The constraint violation information is formulated by two simple metrics that represent the particles' feasibility status individually and as a whole. The final goal is to solve the CMOPs by influencing the particles' search behavior in such that will lead them towards the feasible regions and the optimal Pareto front.

The remaining structure of this chapter is arranged as follows. A review of relevant works in this area is presented in *Literature Survey* section. The proposed constrained MOPSO (so called rank and constraint violation MOPSO or RCVMOPSO)