

# Chapter 5

## A Multiobjective Particle Swarm Optimizer for Constrained Optimization

**Gary G. Yen**

*Oklahoma State University, USA*

**Wen-Fung Leong**

*Oklahoma State University, USA*

### **ABSTRACT**

*Constraint handling techniques are mainly designed for evolutionary algorithms to solve constrained multiobjective optimization problems (CMOPs). Most multiobjective particle swarm optimization (MOPSO) designs adopt these existing constraint handling techniques to deal with CMOPs. In the proposed constrained MOPSO, information related to particles' infeasibility and feasibility status is utilized effectively to guide the particles to search for feasible solutions and improve the quality of the optimal solution. This information is incorporated into the four main procedures of a standard MOPSO algorithm. The involved procedures include the updating of personal best archive based on the particles' Pareto ranks and their constraint violation values; the adoption of infeasible global best archives to store infeasible nondominated solutions; the adjustment of acceleration constants that depend on the personal bests' and selected global best's infeasibility and feasibility status; and the integration of personal bests' feasibility status to estimate the mutation rate in the mutation procedure. Simulation to investigate the proposed constrained MOPSO in solving the selected benchmark problems is conducted. The simulation results indicate that the proposed constrained MOPSO is highly competitive in solving most of the selected benchmark problems.*

DOI: 10.4018/978-1-4666-2479-5.ch005

## 1. INTRODUCTION

In real world applications, most optimization problems are subject to different 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 survey (Michalewicz & Schoenauer, 1996; Mezura-Montes & Coello Coello, 2006) shows a variety of constraint handling techniques have been developed to counter the deficiency 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; Wang *et al.*, 2007, 2008; Oyama *et al.*, 2007; Tessema & Yen, 2009) and CMOPs (Fonseca & Fleming, 1998; Coello & Christiansen, 1999; Binh & Korn, 1997; Deb *et al.*, 2002; Kurpati *et al.*, 2002; Hingston *et al.*, 2006; Jimenez *et al.*, 2002; Ray & Won, 2005; Harada *et al.*, 2007; Geng *et al.*, 2006; Zhang *et al.*, 2006; Chafekar, Xuan & Rasheed, 2003; Woldesenbet, Tessema, & Yen, 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 (Parsopoulos & Vrahatis, 2002; Zielinski & Laur, 2006; He & Wang, 2007; Pulido & Coello, 2004; Liu, Wang, & Li, 2008; Lu & Chen, 2006; Li, Li, & Yu, 2008; Liang & Suganthan, 2006; Cushman, 2007; Wei & Wang, 2006). 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:

$$\begin{aligned} \text{Minimize } \mathbf{f}(\mathbf{x}) &= [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})], \\ \mathbf{x} &= [x_1, x_2, \dots, x_n] \in \mathfrak{R}^n \end{aligned} \quad (1)$$

subject to

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m; \quad (2a)$$

$$h_j(\mathbf{x}) = 0, \quad j = m + 1, \dots, p; \quad (2b)$$

$$x_i^{\min} \leq x_i \leq x_i^{\max}, \quad i = 1, 2, \dots, n \quad (2c)$$

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,  $S \subseteq \mathfrak{R}^n$ .  $g_j(\mathbf{x})$  represents the  $j$ th inequality constraint, while  $h_j(\mathbf{x})$  represents the  $j$ th equality constraint. The inequality constraints that are equal to zero, i.e.,  $g_j(\mathbf{x}^*) = 0$ , 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 integrated the particles' dominance relationship, and their constraint violation information to each of these key procedures.

23 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/multiobjective-particle-swarm-optimizer-constrained/70642](http://www.igi-global.com/chapter/multiobjective-particle-swarm-optimizer-constrained/70642)

## Related Content

---

### On a Software Platform for MASIVE Simulations

Goran Trajkovski (2007). *An Imitation-Based Approach to Modeling Homogenous Agents Societies* (pp. 136-166).

[www.irma-international.org/chapter/software-platform-masive-simulations/5099](http://www.irma-international.org/chapter/software-platform-masive-simulations/5099)

### Meta-Monitoring Using an Adaptive Agent-Based System to Support Dependent People in Place

Nicolas Singer, Sylvie Trouilhetand Ali Rammal (2012). *Theoretical and Practical Frameworks for Agent-Based Systems* (pp. 66-78).

[www.irma-international.org/chapter/meta-monitoring-using-adaptive-agent/66024](http://www.irma-international.org/chapter/meta-monitoring-using-adaptive-agent/66024)

### Social Self-Regulation in Computer Mediated Communities: The Case of Wikipedia

Christopher Goldspink (2009). *International Journal of Agent Technologies and Systems* (pp. 19-33).

[www.irma-international.org/article/social-self-regulation-computer-mediated/1389](http://www.irma-international.org/article/social-self-regulation-computer-mediated/1389)

### Assessing the Effect of Aspect Refactoring on Multi-Agent Applications: A Dynamic Analysis

Mohamed Sedik Chebout, Farid Mokhatiand Mourad Badri (2015). *International Journal of Agent Technologies and Systems* (pp. 45-66).

[www.irma-international.org/article/assessing-the-effect-of-aspect-refactoring-on-multi-agent-applications/160817](http://www.irma-international.org/article/assessing-the-effect-of-aspect-refactoring-on-multi-agent-applications/160817)

### Embodied Conversational Virtual Patients

Patrick G. Kennyand Thomas D. Parsons (2011). *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices* (pp. 254-281).

[www.irma-international.org/chapter/embodied-conversational-virtual-patients/54641](http://www.irma-international.org/chapter/embodied-conversational-virtual-patients/54641)