Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ϵ-Dominance

Margarita Reyes Sierra and Carlos A. Coello Coello

CINVESTAV-IPN (Evolutionary Computation Group),
Electrical Eng. Department, Computer Science Dept.,
Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07300, México
mreyes@computacion.cs.cinvestav.mx
ccoello@cs.cinvestav.mx

Abstract. In this paper, we propose a new Multi-Objective Particle Swarm Optimizer, which is based on Pareto dominance and the use of a crowding factor to filter out the list of available leaders. We also propose the use of different mutation (or *turbulence*) operators which act on different subdivisions of the swarm. Finally, the proposed approach also incorporates the ϵ-dominance concept to fix the size of the set of final solutions produced by the algorithm. Our approach is compared against five state-of-the-art algorithms, including three PSO-based approaches recently proposed. The results indicate that the proposed approach is highly competitive, being able to approximate the front even in cases where all the other PSO-based approaches fail.

1 Introduction

Kennedy and Eberhart [1] initially proposed the swarm strategy for optimization. The particle swarm optimization (PSO) algorithm is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. In PSO, individuals, referred to as particles, are "flown" through hyperdimensional search space. Changes to the position of the particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals. A swarm consists of a set of particles, where each particle represents a potential solution. The position of each particle is changed according to its own experience and that of its neighbors. Let $\boldsymbol{x}_i(t)$ denote the position of particle p_i, at time step t. The position of p_i is then changed by adding a velocity $\boldsymbol{v}_i(t)$ to its current position, i.e.: $\boldsymbol{x}_i(t) = \boldsymbol{x}_i(t-1) + \boldsymbol{v}_i(t)$. The velocity vector drives the optimization process and reflects the socially exchanged information. In the global best version (used here) of PSO, the social knowledge used to drive the movement of particles includes the position of the best particle from the entire swarm ($gbest$) and its history of experiences in terms of its own best solution thus far ($pbest$). In this case, the velocity vector changes in the following way: $\boldsymbol{v}_i(t) = W\boldsymbol{v}_i(t-1) + C_1 r_1(\boldsymbol{x}_{pbest_i} - \boldsymbol{x}_i(t)) + C_2 r_2(\boldsymbol{x}_{gbest} - \boldsymbol{x}_i(t))$, where W is the inertia weight, C_1 and C_2 are the learning factors (usually defined as constants), and $r_1, r_2 \in [0,1]$ are random values. The successful application of PSO in many single objective optimization problems reflects the effectiveness of PSO. However, in order to handle multiple objectives, PSO must be obviously modified. In most

C. A. Coello Coello et al. (Eds.): EMO 2005, LNCS 3410, pp. 505–519, 2005.
© Springer-Verlag Berlin Heidelberg 2005

approaches (which will be generically called MOPSOs, for Multiple-Objective Particle Swarm Optimizers), the major modifications to the basic PSO algorithm are the selection process of $pbest$ and $gbest$ [2, 3]. In this paper, we present a new proposal which is based on Pareto dominance and the use of a crowding factor for the selection of leaders. We also incorporate mutation operators (taken from the evolutionary algorithms literature) and the concept of ϵ-dominance. This paper is organized as follows. The previous related work is reviewed in Section 2. In Section 3, we describe our proposed approach. The obtained results and discussion are presented in Sections 4 and 5, respectively. Finally, the conclusions and future work are described in Section 6.

2 Related Work

There have been several proposals to extend PSO to handle multiple objectives. We will review next the most representative of them:

Ray and Liew [4]: This algorithm uses Pareto dominance and combines concepts of evolutionary techniques with the particle swarm. The approach uses crowding to maintain diversity and a multilevel sieve to handle constraints.

Hu and Eberhart [5]: In this algorithm, only one objective is optimized at a time using a scheme similar to lexicographic ordering. In further work, Hu et al. [6] adopted a secondary population (called "extended memory") and introduced some further improvements to their dynamic neighborhood PSO approach.

Fieldsend and Singh [7]: This approach uses an unconstrained elite archive (in which a special data structure called "dominated tree" is adopted) to store the nondominated individuals found along the search process. The archive interacts with the primary population in order to define local guides. This approach also uses a "turbulence" (or mutation) operator.

Coello et al. [2]: This approach uses a global repository in which every particle deposits its flight experiences. Additionally, the updates to the repository are performed considering a geographically-based system defined in terms of the objective function values of each individual; this repository is used by the particles to identify a leader that will guide the search. It also uses a mutation operator that acts both on the particles of the swarm, and on the range of each design variable of the problem to be solved. In more recent work, Toscano and Coello [8] adopted clustering techniques in order to divide the population of particles into several swarms in order to have a better distribution of solutions in decision variable space. In each sub-swarm, a PSO algorithm is executed and, at some point, the different sub-swarms exchange information: the leaders of each swarm are migrated to a different swarm to variate the selection pressure.

Mostaghim and Teich [3]: They proposed a sigma method in which the best local guides for each particle are adopted to improve the convergence and diversity of a PSO approach used for multiobjective optimization. They also use a "turbulence" operator. In further work, the authors [9] studied the influence of ϵ-dominance [10] on MOPSO methods. ϵ-dominance is compared with existing clustering techniques for fixing the archive size and