EMOPSO: A Multi-Objective Particle Swarm Optimizer with Emphasis on Efficiency

Gregorio Toscano-Pulido[1], Carlos A. Coello Coello[2], and Luis Vicente Santana-Quintero[2]

[1] Universidad Autónoma de Nuevo León, AP 34 - F
Cd. Universitaria, San Nicolás de los Garza, NL 66450, Mexico*
[2] CINVESTAV-IPN (Evolutionary Computation Group)
Depto. de Computación, Av. IPN No 2508
Col. San Pedro Zacatenco, México, D.F., 07360, Mexico
gtoscano@gmail.com
ccoello@cs.cinvestav.mx, lvspenny@hotmail.com

Abstract. This paper presents the Efficient Multi-Objective Particle Swarm Optimizer (EMOPSO), which is an improved version of a multi-objective evolutionary algorithm (MOEA) previously proposed by the authors. Throughout the paper, we provide several details of the design process that led us to EMOPSO. The main issues discussed are: the mechanism to maintain a set of well-distributed nondominated solutions, the turbulence operator that avoids premature convergence, the constraint-handling scheme, and the study of parameters that led us to propose a self-adaptation mechanism. The final algorithm is able to produce reasonably good approximations of the Pareto front of problems with up to 30 decision variables, while performing only 2,000 fitness function evaluations. As far as we know, this is the lowest number of evaluations reported so far for any multi-objective particle swarm optimizer. Our results are compared with respect to the NSGA-II in 12 test functions taken from the specialized literature.

1 Introduction

Particle swarm optimization (PSO) has been found to be a very effective engine for multi-objective optimization, and several multi-objective particle swarm optimizers (MOPSOs) have been proposed in the last few years [1]. Nevertheless, very few researchers have studied the basic mechanisms of a MOPSO, aiming to design a more efficient search engine, which achieves competitive performance at a low number of objective function evaluations (see for example [2,3]). This paper reports a detailed study of a MOPSO previously proposed by the authors in [4]. Such study led us to propose new mechanisms that produced a new MOPSO that only performs 2,000 fitness function evaluations, while solving test problems

* This author is currently affiliated to CINVESTAV-Tamaulipas Laboratorio de Tecnologías de la Información, Carretera a Monterrey Km. 6, Cd. Victoria, Tamps 87261, MEXICO.

S. Obayashi et al. (Eds.): EMO 2007, LNCS 4403, pp. 272–285, 2007.

of up to 30 decision variables. To the best of the authors' knowledge, this is the lowest number of fitness function evaluations ever reported for any MOPSO in the specialized literature. Our results are compared with respect to the NSGA-II [5], which is a multi-objective evolutionary algorithm (MOEA) representative of the state-of-the-art in the area.

2 Towards an Efficient MOPSO

In [4], we proposed the use of clustering techniques to improve the performance of a MOPSO. In order to improve the performance of our original algorithm, we performed several modifications to it. As a first step, we incorporated a mechanism to distribute the nondominated solutions obtained by the algorithm. Next, we used a turbulence operator, in order to avoid premature convergence. After that, and in order to maximize the subswarms performance, we performed an experiment to fix the number of subswarms to be adopted.Then, we incorporated a mechanism to handle constraints. Finally, we performed an empirical study of the influence of the C_1, C_2 and W parameters, and we proposed a simple methodology to self-adapt these parameters. Each of these componentes will be briefly described in the following subsections.

2.1 Handling Well-Distributed Solutions

The MOPSO proposed in [4] does not impose a bound on the total number of nondominated solutions that it can store. This makes it difficult for the decision maker to choose one of them and also complicates the definition of a baseline to perform fair comparisons with respect to other MOEAs. Researchers have proposed several mechanisms to reduce the nondominated solutions generated by a MOEA (most of them applicable to external archives): clusters [6], adaptive grids [7], crowding [5] and relaxed forms of Pareto dominance [8]. In our case, we implemented two mechanisms: (1) an adaptive grid and (2) a relaxed form of Pareto dominance (ε-dominance). These two mechanisms are described next:

- **Adaptive Grid:** Proposed by Knowles & Corne [7], the adaptive grid is really a space formed by hypercubes. Such hypercubes have as many components as objective functions has the problem to be solved. Each hypercube can be interpreted as a geographical region that contains an n number of individuals. The adaptive grid allows us to store nondominated solutions and to redistribute them when its maximum capacity is reached.
- **ϵ-dominance:** This is a relaxed form of dominance proposed by Laumanns et al. [8]. The so-called ε-Pareto set is an archiving strategy that maintains a subset of generated solutions. It guarantees convergence and diversity according to well-defined criteria, namely the value of the ε parameter, which defines the resolution of the grid to be adopted for the secondary population. The general idea of this mechanism is to divide objective function space into boxes of size ε. Each box can be interpreted as a geographical region that