An Efficient Multi-objective Evolutionary Algorithm: OMOEA-II

Sanyou Zeng[1,2], Shuzhen Yao[1], Lishan Kang[3], and Yong Liu[1]

[1] Dept. of Computer Science, China University of GeoSciences,
430074 Wuhan, Hubei, P.R.China
yliu@u-aizu.ac.jp, sanyou-zeng@263.net

[2] Dept. of Computer Science, Zhuzhou Institute of Technology,
412008 Zhuzhou, Hunan, P.R.China

[3] State Key Laboratory of Software Engineering, Wuhan University
430072 Wuhan, Hubei, P.R.China

Abstract. An improved orthogonal multi-objective evolutionary algorithm (OMOEA), called OMOEA-II, is proposed in this paper. Two new crossovers used in OMOEA-II are orthogonal crossover and linear crossover. By using these two crossover operators, only small orthogonal array rather than large orthogonal array is needed for exploiting optimal in the global space. Such reduction in orthogonal array can avoid exponential creation of solutions of OMOEA and improve the performance in robusticity without degrading precision and distribution of solutions. Experimental results show that OMOEA-II can solve problems with high dimensions and large number of local Pareto-optimal fronts better than some existing algorithms recently reported in the literatures.

Keywords: evolutionary algorithms; multi-objective optimization; Pareto optimal set.

1 Introduction

Almost every real-world problem involves simultaneous optimization of several incommensurable and often competing objectives. Evolutionary algorithms have the ability to find multiple Pareto-optimal solutions in one single simulation run. They have often been used to solve multi-objective problems. Such as vector evaluated genetic algorithm (VEGA)[1], Hajela and Lins genetic algorithm(HLGA) [2], pareto-based ranking procedure(FFGA) [3], niched Pareto genetic algorithm (NPGA) [4], pareto archived evolution strategy (PAES)[5], nondominated sorting genetic algorithm (NSGA-II)[6], strength pareto evolutionary algorithm (SPEA2) [7], rMOGAxs [8], and generalized regression GA (GRGA) [9].

Orthogonal design method [10] is developed to sample a small, but representative set of combinations for experimentation to obtain good combination. Leung and Zhang incorporated orthogonal design in genetic algorithm for single objective problems[11][12], found such method was more robust and statistically

C. A. Coello Coello et al. (Eds.): EMO 2005, LNCS 3410, pp. 108–119, 2005.

sound. In [13], orthogonal design method is used in multi-objective evolutionary algorithm and developed algorithm was called OMOEA. It was showed that OMOEA could find good solutions. But OMOEA degraded its performance on both precision and distribution of the yielded solutions for problems with strong interaction between variables, and when the number of objectives increases, the solutions yielded by OMOEA increased exponentially.

In this paper, an improved version of OMOEA (OMOEA-II) is proposed. Orthogonal design method is nested in crossover operator to select better genes as offsprings, and consequently, enhances the performance of OMOEA. Both orthogonal crossover and linear crossover are used in OMOEA-II. By combining two crossover operators, faster convergence and better solutions are obtained.

The rest of this paper is organized as follows. Section 2 briefly describes multi-objective optimization problem and orthogonal design method. Section 3 presents the proposed OMOEA-II. Section 4 shows experiment results and discussions. Finally, Section 5 concludes with a summary of the paper.

2 Preliminary

2.1 Problem Definition

Definition 1. ***(Multi-objective Optimization Problem(MOP))*** *A general MOP includes a set of N parameters (decision variables), a set of K objective functions, and a set of L constraints. Objective functions and constraints are functions of the decision variables. The optimization goal is to*

$$\begin{array}{ll}
\mathit{minimize} & \boldsymbol{y} = \boldsymbol{f}(x) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_K(\boldsymbol{x})) \\
\mathit{subject\ to} & \boldsymbol{e}(\boldsymbol{x}) = (e_1(\boldsymbol{x}), e_2(\boldsymbol{x}), ..., e_L(\boldsymbol{x})) \leq \boldsymbol{0} \\
\mathit{where} & \boldsymbol{x} = (x_1, x_2, ..., x_N) \in \mathcal{X} \\
& \mathcal{X} = \{(x_1, x_2, ..., x_N) | l_i \leq x_i \leq u_i, i = 1, 2, ..., N\} \\
& \boldsymbol{z} = (z_1, z_2, ..., z_N) \\
& \boldsymbol{l} = (l_1, l_2, ..., l_N) \\
& \boldsymbol{u} = (u_1, u_2, ..., u_N) \\
& \boldsymbol{y} = (y_1, y_2, ..., y_K) \in \mathcal{Y}
\end{array} \tag{1}$$

*where $\boldsymbol{x}$ is the **decision vector**, $\boldsymbol{y}$ is the **objective vector**, $\mathcal{X}$ denotes the decision space, $\boldsymbol{z}$ is the center of the decision space, $\boldsymbol{l}$ and $\boldsymbol{u}$ are the upper bound and lower bound of the decision space, and $\mathcal{Y}$ is called the objective space.*

2.2 Orthogonal Design Methods

An example was introduced in [14] to explain the basic concept of experimental design methods. The yield of a vegetable depends on: 1) the temperature, 2) the amount of fertilizer, and 3) the pH value of the soil. These three quantities are called the factors of the experiment. Each factor has three possible levels shown in Table 1. To find the best combination of levels for a maximum yield, we can do one experiment for each combination, and then select the best one. In the