Prediction-Based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization

Aimin Zhou[1], Yaochu Jin[2], Qingfu Zhang[1], Bernhard Sendhoff[2], and Edward Tsang[1]

[1] Department of Computer Science, University of Essex, Colchester, CO4 3SQ, U.K.
[2] Honda Research Institute Europe, Carl-Legien-Str. 30, 63073, Offenbach, Germany

Abstract. Optimization in changing environment is a challenging task, especially when multiple objectives are to be optimized simultaneously. The basic idea to address dynamic optimization problems is to utilize history information to guide future search. In this paper, two strategies for population re-initialization are introduced when a change in the environment is detected. The first strategy is to predict the new location of individuals from the location changes that have occurred in the history. The current population is then partially or completely replaced by the new individuals generated based on prediction. The second strategy is to perturb the current population with a Gaussian noise whose variance is estimated according to previous changes. The prediction based population re-initialization strategies, together with the random re-initialization method, are then compared on two bi-objective test problems. Conclusions on the different re-initialization strategies are drawn based on the preliminary empirical results.

1 Introduction

In this paper, we consider the following continuous dynamic multi-objective optimization problems (DMOP):

$$\begin{array}{ll} \text{minimize } \boldsymbol{F}(x,t) = (f_1(x,t), f_2(x,t), \ldots, f_m(x,t))^T, \\ \text{subject to} \qquad\qquad\qquad\quad x \in X, \end{array} \tag{1}$$

where $t = 0, 1, 2, \cdots$ represents time, $x = (x_1, \ldots, x_n)^T \in R^n$ is the decision variable vector and $X \subset R^n$ is the decision space. R^m is the objective space. $\boldsymbol{F} : (X, t) \rightarrow R^m$ consists of m real-valued objective functions $f_i(x,t)$ $(i = 1, 2, \ldots, m)$, each of which is continuous with respect to x over X. The Pareto front (PF) in the objective space and the Pareto set (PS) in the decision space change over time. The task of a dynamic multi-objective optimization algorithm is to trace the movement of the PF and PS with reasonable computational costs.

Inspired by the success of evolutionary algorithms on dynamic scalar optimization problems [1,2,3], research work on evolutionary dynamic multi-objective optimization (EDMO) has very recently been conducted by several researchers. In the following, we briefly review the current work on EDMO:

i **Test Problems:** Benchmarks are important for developing and testing algorithms for solving DMOPs. In [4], Jin and Sendhoff proposed a method for constructing

S. Obayashi et al. (Eds.): EMO 2007, LNCS 4403, pp. 832–846, 2007.

dynamic multi-objective test problems by aggregating different objectives of existing stationary multi-objective problems and changing the weights dynamically. Test problems in [5] and [6] are created by adding time-varying terms to the objectives in stationary MOP test problems.

ii **Algorithms:** Several attempts for solving DMOPs by evolutionary algorithms have been reported recently. Stationary multi-objective evolutionary algorithms such as NSGA-II [7], SPEA2 [8], MSOPS [9] and OMOEA-II [10] have been directly applied to DMOPs [6,11]. A few evolutionary algorithms for solving dynamic single objective optimization problems have also been extended to the case of multi-objective problems [12]. Several strategies have been proposed by extending stationary multi-objective evolutionary algorithms for tracking the movement of the PS [13,14,15] or uncertain objectives [16,17].

iii **Performance Indicators:** It is hard to measure the performance of algorithms for DMOPs for the following reasons. Firstly, the measure must be able to evaluate the quality of approximation of a solution set, which itself is not trivial. Secondly, the PS is changing over time. It is natural to draw the PF for stationary multi-objective optimization, but it is no longer practical to plot the changing PFs in dynamic environment. In [12], two convergence performance measures have been suggested. In [6], the generational distance with time was plotted to show the convergence. In addition, a distribution indicator, known as the PL-metric, has also been introduced [6].

Arguably, diversity maintenance is essential in dynamic scalar objective evolutionary optimization algorithms. It is, however, interesting to note that in multi-objective evolutionary algorithms, the diversity of population is inherently maintained due to the multi-objective nature. Thus, it is probably of greater importance to ensure that the population is able to follow the moving PF more quickly. To this end, a good guess of the new location of the changed PS is of great interest.

In this paper, we study how to generate an initial population close to a changed PF when a change is detected in a dynamic environment. Inspired by the prediction strategy in [14,15], we build prediction models to predict the location of the new PS based on the information collected from the previous search. Different to the method in [14,15] where only the new locations of two anchor points and the Closest-To-Ideal point are predicted, we predict the new locations of a number of Pareto solutions in the decision space once a change is detected. Individuals in the initial population for the changed problem are generated around these predicted points. In such a way, the changed PS and PF can be found more effectively by the algorithm.

Four methods for re-initialization have been studied and compared in this paper. They are 1) Random re-initialization method in which the initial populations are randomly generated in the search space; 2) Variation method in which the individuals in the current population are perturbed using a Gaussian noise whose variance is determined by changes in the history; 3) Prediction method in which the new trial solutions are generated around predicted locations; and 4) A naive hybrid method, in which half of population is generated by strategy 2 and half is created by strategy 3.