Expert Systems with Applications 41 (2014) 6274–6290

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A new Probe Guided Mutation operator and its application for solving the cardinality constrained portfolio optimization problem

K. Liagkouras *, K. Metaxiotis

Decision Support Systems Laboratory, Department of Informatics, University of Piraeus, 80, Karaoli & Dimitriou Str., 18534 Piraeus, Greece

ARTICLE INFO

Keywords:
Multiobjective optimization
Evolutionary algorithms
Mutation
Portfolio optimization
Cardinality constrained

ABSTRACT

This paper revisits the classical Polynomial Mutation (PLM) operator and proposes a new probe guided version of the PLM operator designed to be used in conjunction with Multiobjective Evolutionary Algorithms (MOEAs). The proposed Probe Guided Mutation (PGM) operator is validated by using data sets from six different stock markets. The performance of the proposed PGM operator is assessed in comparison with the one of the classical PLM with the assistance of the Non-dominated Sorting Genetic Algorithm II (NSGAII) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2). The evaluation of the performance is based on three performance metrics, namely Hypervolume, Spread and Epsilon indicator. The experimental results reveal that the proposed PGM operator outperforms with confidence the performance of the classical PLM operator for all performance metrics when applied to the solution of the cardinality constrained portfolio optimization problem (CCPOP). We also calculate the True Efficient Frontier (TEF) of the CCPOP by formulating the CCPOP as a Mixed Integer Quadratic Program (MIQP) and we compare the relevant results with the approximate efficient frontiers that are generated by the proposed PGM operator. The results confirm that the PGM operator generates near optimal solutions that lie very close or in certain cases overlap with the TEF.

1. Introduction

Evolutionary algorithms (EAs) have been increasingly applied over the past years for the solution of optimization problems with multiple objectives. The typical Multiobjective Evolutionary Algorithm (MOEA) utilizes three basic operators: selection, crossover and mutation (Metaxiotis & Liagkouras, 2012). However, the available literature regarding the variation operators for evolutionary multiobjective optimization remains relatively small. In particular, the mutation operator has received little attention and the majority of MOEAs make use of the Polynomial Mutation (PLM) operator proposed by Deb and Goyal (1996). Later, Deb and Tiwari (2008) proposed a highly disruptive version of the Polynomial Mutation that has been utilized in the latest version of NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002) and SPEA2 (Zitzler, Laumanns, & Thiele, 2001). This paper proposes a new version of the highly disruptive Polynomial Mutation (PLM) named Probe Guided Mutation (PGM) operator that produces better results. More recently, Da Ronco and Benini (2013) presented a Shrink-Mutation operator for MOEAs that belongs to the Gaussian mutations category. However, as the authors admit the PLM operator when applied to IBEA algorithm gives better results than the Shrink-Mutation operator in terms of convergence towards the True Pareto Front (Da Ronco & Benini, 2013). Das, Mallipeddi, and Maity (2013) presented a *p*-best mutation strategy for Evolutionary Programming (EP). EP relies mainly on its mutation operator for function optimization. Shortly the *p*-best mutation operator entails that any one of the *p* top-ranked population members according to fitness value is selected randomly for mutation. Tang and Tseng (2013) presented a new mutation operator called ADM for real coded Genetic Algorithms (GAs). According to the authors, the ADM mutation operator enhances the abilities of GAs in searching global optima as well as in speeding convergence by integrating a local directional search and adaptive random search strategies.

The majority of the most recent mutation operators have been developed for differential evolution (DE) algorithms. Thus, Zhou, Li, and Gao (2013) proposed a new mutation operator called intersect mutation differential evolution (IMDE) algorithm. Alguliev, Aliguliyev, and Isazade (2012) proposed a new DE algorithm based on self-adaptive mutation and crossover (DESAMC). The proposed method dynamically adapts scale factor and crossover rate. Gong, Cai, and Liang (2014) present a ranking-based mutation operator that makes the DE algorithm to converge faster.

* Corresponding author. Tel.: +30 6948612347; fax: +30 210 4142264.
E-mail addresses: k_liagkouras@hotmail.com, kliagk@unipi.gr (K. Liagkouras).

http://dx.doi.org/10.1016/j.eswa.2014.03.051
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

Although the considerable amount of the ongoing work related to mutation operators and their importance for the performance of the entire algorithm, we noticed that the study of mutation operators in the context of MOEAs remains relatively rare. The PLM operator remains undoubtedly the mutation operator of choice when it comes to MOEAs. The motivation of this study is to build on the existing PLM and present a mechanism (the PGM) that allows the better exploration of solution space and is able to generate near optimal solutions that lie very close to the True Efficient Frontier (TEF).

The remainder of the paper is organized as follows. In Section 2, a description of the highly disruptive Polynomial Mutation (PLM) is given and in Section 3 the proposed Probe Guided Mutation (PGM) and the formulation of the cardinality constrained portfolio optimization problem (CCPOP) are presented. In Section 4 the implementation of the cardinality constraint and lower and upper bound to the MOEA are presented. The parameters setup is presented in Section 5.1 and in Section 5.2 we formulate the CCPOP as a Mixed Integer Quadratic Program (MIQP) and we extract the True Efficient Frontier for each one of the examined problems with the assistance of CPLEX 12.5. Section 6 presents the performance metrics. In Section 7 we test the performance of the proposed PGM by using data sets from six different stock markets for the solution of the CCPOP. In Section 8 the results are analyzed and finally, Section 9 concludes the paper.

2. Polynomial Mutation

Mutation operators are being used as variation mechanisms to change the offspring genes. They assist to the better exploration of the search space. For MOEAs solving Multiobjective Problems (MOPs), Deb and Goyal (1996) proposed a variation mechanism called Polynomial Mutation (PLM). This operator was later improved by Deb and Tiwari (2008).

In Polynomial Mutation as introduced by Deb and Tiwari (2008) each decision variable x_i, can take values in the interval: $x_i^{(L)} \leqslant x_i \leqslant x_i^{(U)}$, $i = 1, 2, \ldots, n$. Where $x_i^{(L)}$ and $x_i^{(U)}$ stand respectively for the lower and upper bounds for the decision variable *i*. Moreover, each decision variable has a probability P_m to be perturbed. For each decision variable, a random value *rand* is drawn. If $rand \leqslant P_m$ then using the algorithm described in Fig. 1, a mutated variable obtains its new value. If random value is $r \leqslant 0.5$ it samples to the left hand side (region between X_{Low} and X_i), otherwise if $r > 0.5$ it samples to the right hand side (region between X_i and X_{Upper}). The algorithm also calculates the δ_q value to be used in getting the variable its new value. According to Deb and Tiwari (2008) one of the main advantages of Polynomial Mutation is that it allows us to sample the entire search space of the decision variable even though the value to be mutated is close to one of the boundaries $(X_{Low} - X_{Upper})$.

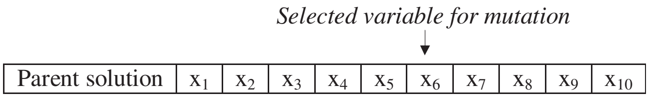
Moreover, because PLM allows big jumps in the search space of the decision variable, the optimization process has better chances of escaping from local optima and can modify a solution when on the boundary. However, high disruption levels might not be good for achieving smooth approximation of the Pareto front. For instance, if a solution is near an optimal solution then large jumps in the decision space might not be an efficient way of discovering other optimal solutions too. The highly disruptive PLM has been applied in the latest version of NSGAII (Deb et al., 2002), SPEA2 (Zitzler et al., 2001) and a java implementation of PLM algorithm provided by the jMetal framework (Durillo & Nebro, 2011).

3. Probe Guided Mutation (PGM)

The Probe Guided Mutation (PGM) operator, as its name reveals has been developed in order to facilitate the more efficient exploration of the search space. We will start analyzing PGM mechanism by recalling Fig. 1, as the first step is common for both methods. In particular, *if* $rand \leqslant P_m$ then a decision variable is selected to be mutated. Suppose a hypothetical solution vector $\mathbf{x} = (x_1, x_2, \ldots, x_{10})$ which satisfies the variables bounds $x_i^{(L)} \leqslant x_i \leqslant x_i^{(U)}$, $i = 1, 2, \ldots, 10$. Also suppose that a random number $rand \leqslant P_m$ occurs for the 6th decision variable. That means that the 6th decision variable of the parent solution should be mutated.

```
Begin
mutation_probability = 1/n; (where n is the number of decision variables)
η_m = distribution index;

 for i=0 to N; (where N is the population size)
     for z=0 to n;
         X_p = getValue(z);
         X_l = getLowerBound(z);
         X_u = getUpperBound(z);

         rand → [0, 1];
         if (rand <= mutation_probability) then

             $\delta_1 = \frac{X_p - X_l}{X_u - X_l}$      $\delta_2 = \frac{X_u - X_p}{X_u - X_l}$

             r → [0, 1];

                 if (r<= 0.5) then
                     $\delta_q = [2r + (1 - 2r)(1 - \delta_1)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}}$
                 else
                     $\delta_q = 1 - [2(1 - r) + 2(r - 0.5)(1 - \delta_2)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}}$
                 end if

                 $X_c = X_p + \delta_q(X_u - X_l)$
                     if ( X_c < X_l ) then
                         X_c = X_l ;
                     endif
                     if ( X_c > X_u ) then
                         X_c = X_u ;
                     endif

             Child_Solution = Parent_Solution.setValue(z , X_c);
         endif
     endfor
endfor
```

Fig. 1. Polynomial Mutation (PLM) Pseudo code.



Parent solution	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}

As shown in Fig. 1 that illustrates the Polynomial Mutation (PLM) operator, if random value is $r \leqslant 0.5$ it samples to the left hand side (region between X_{Low} and X_i), otherwise if $r > 0.5$ it samples to the right hand side (region between X_i and X_{Upper}). In PGM at this particular point, as shown in Fig. 2 we follow a different methodology. Specifically, instead of generating a random number $r \in [0, 1]$, we generate two random numbers, $r_L \in [0, 0.5]$ to sample the left hand side and a random number $r_R \in (0.5, 1]$ to sample the right hand side. Then, we calculate parameter δ_q^L and δ_q^R as shown in Fig. 2 and their values make possible the calculation of the X_C^L and X_C^R, where X_C^L stands for the child decision variable that samples the region between X_{Low} and X_p. Similarly X_C^R stands for the child decision variable that samples the region between X_p and X_{Upper}. Please notice that X_p is the parent decision variable.

So far by implementing the PGM we obtained two child decision variables, one that samples in the left hand side of the parent