Modifications of Fitness Assignment Parts of Some Multi-Objective Evolutionary Algorithms

Engin Ufuk ERGUL*
Amasya University, Faculty of Technology, Department of Electrical and Electronics Engineering, Turkey,

Ilyas EMINOGLU
Ondokuz Mayis University, Faculty of Engineering, Department of Electrical and Electronics Engineering, Turkey

Article history
Received: 01.11.2014
Received in revised form: 28.12.2014
Accepted: 03.12.2014

Evolutionary algorithms are very popular methods to solve multi-objective optimization problems. In literature, there are many multi-objective evolutionary algorithm methods. It is possible to improve these methods. This paper proposes a line of refinements and modifications on the fitness assignment part of well known SPEA method. The density information generated by k-NN (k-th nearest neighbor method) and the concept of domination power are used to modify SPEA in order to deliver a much refinement fitness values in terms of uniqueness and homogeneity. Two modified methods are proposed (named SPEAmod1 and SPEAmod2). The main goal is to extract more information from a population (and is to deliver to the decision maker) by using proposed variants than the original method. The original method and its modifications are tested on four well-known test functions in the literature. The proposed modifications are shown to be superior to the original method. Finally, fitness assignment capability of original SPEA method is significantly improved.

Key words: evolutionary algorithms; multi-objective optimization; modification; fitness assignment

* Correspondence: engin.ergul@amasya.edu.tr
1. Introduction

In engineering area, multi-objective optimization is a significant subject. Many real-world optimization applications involve several conflicting objectives [1]. Evolutionary algorithms (EAs) are effective to solve the problems which have multiple objectives. While searching the optimal result/results of a problem, EAs assign fitness values to individuals/solutions in early steps of their algorithms. Detailed information about EAs can be found in [1,2].

A survey of multi-objective evolutionary algorithms (MOEAs) has been provided in [3]. A survey of recent developments on MOEAs can be found in [4]. In the literature, a lot of MOEAs have been published so far [5,6,7,8,9,10,11,12,13, and etc.]. Jensen [14] says that “The majority of these algorithms use fitness assignment based on Pareto domination: non-dominated sorting, dominance counting or identification of non-dominated solutions”.

Many existing methods also strongly indicate that fitness assignment is still an open question in MOEAs. Another interpretation of many existing FAMs (Fitness Assignment Methods) is that multiple objectives (or dimensionality) are reduced by FAMs to a single fitness value (or in a single dimension) and there is no single and simple way to make the dimensionality reduction [15].

Two improvements can be done in MOEAs: i) improvement in the fitness assignment mechanism, ii) improvement in the elitism mechanism. In this paper, the first one is applied for improving the performance of SPEA. The performance of SPEA algorithm is increased by modifying the fitness assignment mechanism.

The k-NN and the concept of domination power (of an individual) are used to modify SPEA method in order to deliver a much refinement fitness values in terms of uniqueness and homogeneity. Two modified methods are proposed out of SPEA [8]. The proposed modifications (or variants) are obtained by adding: i) the density information generated by k-NN, ii) the concept of domination power of an individual.

This paper is organized as follows. SPEA algorithm is briefly given in Section 2. Modification methods are presented in Section 3. Application the modification methods to SPEA is given in Section 4. Simulation results are presented in Section 5. Finally, some conclusions and future directions are presented in Section 6.

2. Strength pareto evolutionary algorithm (SPEA)

SPEA is an EA method used Pareto domination concept [8]. Two different populations are used in SPEA: The first population called as current population is used to create the offspring. The second population called as archive is used to maintain the evolutionary information of Pareto front. non-dominated solutions (or Pareto individuals) are copied to the archive at each generation. For each solution in the archive, a strength value is computed. Strength value of a solution is proportional to the number of solutions dominated by this solution.

The fitness value of a solution in the current population is computed by using the strength values of all external non-dominated solutions that dominate it. Additionally, a clustering method is used to keep diversity.

3. Modification methods

3.1. The density information generated by k-th nearest neighbor method

The density estimation method proposed by Zitzler et al.[10] is used for modifications. Zitzler et al.[10] suggested that “an adaptation of the k-th nearest neighbor method, where the density at any point is a (decreasing) function of the distance to the k-th nearest data point”. The authors of SPEA2 use the inverse of the distance to the k-NN for density estimation. Zitzler et al.[10] calculated the distances in
objective space for each solution $i$ to all solutions $j$ in the archive. After that, they stored distances in a list. Then, the list is sorted in an increasing order and the $k$-th component presents the distance indicated as $\sigma_i^k$ [10].

The authors of SPEA2 selected $k = \sqrt{N + N}$. ($N$: population size, $N$: archive size). Afterwards, the density of a solution is calculated by:

$$D(i) = \frac{1}{\sigma_i^k + 2}$$

(1)

### 3.2. Domination power concept

Domination powers of individuals in a population are calculated by using Algorithm-I.

**Algorithm-I (Domination Power Algorithm)**

**Step 1:** Assign dummy fitness to each individual inversely proportional to their sub-population number (obtained from MOGA [6] algorithm).

$$df(i) = \frac{1}{\text{sub_population}(j)}$$

(2)

$i = 1 \ldots n$, where $n$ is the total number of individuals and $j = 1 \ldots m$, and $m$ is the total number of sub-populations. A numerical example is given as (there are four sub-populations in Figure 2(b), {A,B,C,D},{E,F},{G},{H}):

- $df(A)=1.0000$, $df(B)=1.0000$, $df(C)=1.0000$, $df(D)=1.0000$
- $df(E)=0.5000$, $df(F)=0.5000$, $df(G)=0.3333$, $df(H)=0.2500$

**Step 2:** Calculate relative domination power of an individual. To do so, find out how many individuals are dominated by this particular individual and sum all the dummy fitness values of dominated individuals. This will determine:

$$rdp(i) = \sum_{k=1}^{r} df(k)$$

(3)

where $k \ldots r$ represents all individuals dominated by $ind_i$ and $k \ldots r \neq i$. A numerical example is given as:

- $rdp(A)=df(G)+df(H)=0.3333+0.25=0.5833$
- $rdp(B)=df(F)+df(G)+df(H)=0.5+0.3333+0.25=1.0833$
- $rdp(C)=df(E)+df(H)=0.5+0.25=0.7500$
- $rdp(D)=df(H)=0.2500$
- $rdp(E)=rdp(F)=rdp(G)=0.2500$
- $rdp(H)=0.0000$

**Step 3:** Repeat Step 2 for the rest of the individuals within the same sub-population. Sum the $rdp()$ of all individuals within the same sub-population and obtain total domination power for that sub-population as formulated below:

$$tdp(\text{sub_population}(j)) = \sum_{i}^{p} rdp(i)$$

(4)

where $i \ldots p$ are all individuals inside the sub-population($j$). A numerical example is given as:

- $tdp(\text{sub_population}_1)=rdp(A)+rdp(B)+rdp(C)+rdp(D)=0.5833+1.0833+0.75+0.25=2.6666$
- $tdp(\text{sub_population}_2)=rdp(E)+rdp(F)=0.25+0.25=0.5000$
- $tdp(\text{sub_population}_3)=rdp(G)=0.2500$
- $tdp(\text{sub_population}_4)=rdp(H)=0.0000$
**Step 4:** Calculate the *Domination Power* (DP) for the *i*-th individual with the following formula (where 1 is added to the nominator to avoid divide by zero):

\[
DP(i) = \frac{rdp(i)}{tdp(sub - population(j)) + 1}
\]  

A numerical example is given as:

- DP(A) = 0.5833/(2.6666+1) = 0.1591
- DP(B) = 1.0833/(2.6666+1) = 0.2955
- DP(C) = 0.7500/(2.6666+1) = 0.2045
- DP(D) = 0.2500/(2.6666+1) = 0.0682
- DP(E) = 0.2500/(0.5000+1) = 0.1667
- DP(F) = 0.2500/(0.2500+1) = 0.2000
- DP(G) = 0.2500/(0.0000+1) = 0.0000
- DP(H) = 0.0000/(0.0000+1) = 0.0000

**Step 5:** Repeat step 2 - step 4 until all individuals are ranked.

4. Application the modification methods to SPEA

4.1. The first modification of SPEA (SPEAmmod1)

In this modification, the final fitness values of individuals are calculated by adding the raw fitness values found by original SPEA to the density values calculated by using k-NN.

The final fitness value of an individual is computed as \((i=1,2,...,population\ size)\):

\[
F_{SPEAmmod1}(i) = F_{SPEA}(i) + D_{SPEAmmod1}(i)
\]  

An example of fitness assignment of this modification is shown in Figure 1. Ordering of individuals is: \(D > A, C > B > E > F > G > H\)

4.2. The second modification of SPEA (SPEAmmod2)

In this modification, the final fitness values of individuals are calculated by adding the raw fitness values found by original SPEA to the domination powers of individuals. The final fitness value of an individual is computed as \((i=1,2,...,population\ size)\):

\[
F_{SPEAmmod2}(i) = F_{SPEA}(i) + DP_{SPEAmmod2}(i)
\]  

An example of fitness assignment of this modification is shown in Figure 2. Ordering of individuals from the best one to the worst one is given as: \(D > A > C > B > E > F > G > H\).
5. Simulation results

SPEA and its modifications are simulated on four test functions (including ZDT1, ZDT2, ZDT3 and ZDT6) in the literature [8]. The single-point crossover and bit-wise mutation are applied. The crossover probability of $p_c = 0.9$ and a mutation probability of $p_m = 1/l$ (where $l$ is the string length) are used. Stochastic Universal Sampling (SUS) is employed as a fitness proportional selection operator.

The population of size is selected as 100 individuals and the external population size is selected as 25 individuals. 30 bits are used to code each decision variable. Two metrics (GD and $\Delta$ [1,11]) are used to evaluate the performance of algorithms. The first metric (GD-Generational Distance) measures the convergence to a known set of Pareto-optimal solutions [1,11]. The second metric ($\Delta$-Spread) measures the spread of the obtained solutions [1,11]. Afterwards, mean values of convergence (GD) and diversity ($\Delta$) metrics are calculated. This two metrics are desired to be small. SPEA and its modifications are run twenty times for each test function.

Table 1 presents the mean values of the GD for the 20 final populations. Table 2 presents the mean values of the $\Delta$ for the 20 final populations. To make an easy reading, the best results in Table 1 and Table 2 are shown in bold.

According to Table 1, SPEAmod1 has the best convergence values on ZDT1 and ZDT3 test functions, and SPEAmod2 has the best convergence values on ZDT2 and ZDT6 test functions. Therefore, it is noticed that SPEAmod1 is superior to original SPEA in terms of convergence metric on test set used in this paper.

According to Table 2, SPEAmod1 has the best diversity metric values on ZDT1, ZDT2 and ZDT3 test functions. Also, original SPEA has the best on ZDT6 test function. SPEAmod1 is superior to original SPEA in terms of diversity metric on three test problems out of four test problems. SPEAmod2 is the worst among three methods in terms of diversity.

Table 1. Mean values of the convergence metrics (GD).

|       | GD  | ZDT1   | ZDT2   | ZDT3   | ZDT6   |
|-------|-----|--------|--------|--------|--------|
| SPEA  | 0.1256 | 0.2432 | 0.1039 | 1.8338 |
| SPEAmod1 | 0.0695 | 0.2195 | **0.0750** | 1.7353 |
| SPEAmod2 | 0.1441 | **0.1879** | 0.1406 | **0.7070** |

According to Table 2, SPEAmod1 has the best diversity metric values on ZDT1, ZDT2 and ZDT3 test functions. Also, original SPEA has the best on ZDT6 test function. SPEAmod1 is superior to original SPEA in terms of diversity metric on three test problems out of four test problems. SPEAmod2 is the worst among three methods in terms of diversity.
6. Conclusions

The fitness assignment part of a well-known SPEA algorithm is computationally and structurally modified to generate more informative fitness values than the original one. And, two modified variants (SPEAmod1 and SPEAmod2) are obtained in order to deliver a much refinement fitness values in terms of uniqueness and homogeneity. The proposed modifications (or variants) are obtained by adding: i) the density information generated by k-NN, ii) the concept of domination power of an individual.

The main goal is to extract rich fitness (or the desirability level of individuals) information from a population (and is to deliver to the decision maker). The original methods and proposed ones are tested on four benchmarks. Generally speaking, modifications of SPEA are better than original SPEA.

Finally, fitness assignment capability of original SPEA method is significantly improved. As a future work, a structurally modified decision maker may be needed to be able to process a much more refined fitness values. Therefore, significant modifications are necessary to design and construct an intelligent (and as well as efficient) decision maker to process a more refinement set of fitness values.

7. References

[1] Deb K. Multiobjective Optimization Using Evolutionary Algorithms. Chichester, U.K.: Wiley, 2001.

[2] Coello, C.A.C, Lamont, G.B, Veldhuizen, D.A. Evolutionary Algorithms For Solving Multi-Objective Problems. Second Edition, Springer, New York, U.S.A., 2007.

[3] Coello C.A.C. Evolutionary multi-objective optimization: a historical view of the field. IEEE Computational Intelligence Magazine 2006; Vol.1, Issue 1: 28-36.

[4] Zhou A, Qu B.-Y, Li H, Zhao S.-Z, Suganthan P.N, Zhang Q. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 2011; Vol.1, Issue 1: 32-49.

[5] Schaffer J.D. Some experiments in machine learning using vector evaluated genetic algorithms. Ph.D Thesis, Vanderbilt University, Nashville, U.S.A., 1984.

[6] Fonseca C.M, Fleming P.J. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In Proceedings of the Fifth International Conference on Genetic Algorithms; July 1993; pp. 416-423.

[7] Srinivas N, Deb K. Multi-objective function optimization using non-dominated sorting genetic algorithms. Evolutionary Computation Journal 1994; 2(3): 221-248.

[8] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A
comparative case study and the strength pareto approach. IEEE Trans. on Evolutionary Computation 1999; Vol.3, No. 4: 257-271.

[9] Knowles J.D, Corne D.W. Approximating the nondominated front using the pareto archived evolution strategy. Evolutionary Computation 2000; Vol.8, No.2: 149-172.

[10] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength pareto evolutionary algorithm. TIK-Report 103, Swiss Federal Institute of Technology, Zurich, Switzerland, May 2001(Errata added Sept 27, 2001).

[11] Deb K, Pratap A, Agarwal S, and Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 2002; Vol.6, No.2: 182-197.

[12] Zitzler E, Künzli S. Indicator-Based selection in multiobjective search. In 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII); Birmingham, UK, September 2004; pp.832-842.

[13] Knowles, J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 2006; 10 (1): 50-66.

[14] Jensen M. T. Reducing the run–time complexity of multi-objective EAs: The NSGA-II and other algorithms. IEEE Transactions on Evolutionary Computation 2003; Vol.7, No.5: 503-515.

[15] Ergul, E.U, and Eminoglu, I. DOPGA: A new fitness assignment scheme for multi-objective evolutionary algorithms. International Journal of Systems Science 2014; 45(3): 407-426.