Strength Pareto Evolutionary Algorithm using Self-Organizing Data Analysis Techniques

Ionuț BALAN
“Ștefan Cel Mare” University, Suceava, Romania
ionutb@seap.usv.ro

Abstract—Multiobjective optimization is widely used in problems solving from a variety of areas. To solve such problems there was developed a set of algorithms, most of them based on evolutionary techniques. One of the algorithms from this class, which gives quite good results is SPEA2, method which is the basis of the proposed algorithm in this paper. Results from this paper are obtained by running these two algorithms on a flow-shop problem.

Keywords: multiobjective optimization, dominance, Pareto, evolutionary, classification, SPEA2.

I. INTRODUCTION

Multiobjective optimization or multiobjective programming[1], also known as multicriteria or multiatribute optimization, is the simultaneous determination of the minimum or maximum for two or more conflictual objectives.

Multicriteria optimization problem can be found in various areas: design of various products, finance, aircraft design, oil and gas, automotive and others. By solving optimization problems we should obtain unique solutions or a set of solutions, each being optimal in different situations for the objectives functions of the solved problem.

Multicriteria optimization requires the existence of two search spaces: a spaces of decision variables, a n-dimensional search space V, described by the decision vectors as \( x = (x_1, x_2, ..., x_n) \), and a m-dimensional space O, named objective space, described by the objectives vectors as \( y = (y_1, y_2, ..., y_m) \) [2].

The purpose of this paper is to present a new algorithm that uses population-based methods and unsupervised classification techniques to determine the corresponding solutions for multiobjective problem. Using this method we aim to achieve, in a reduced runtime, satisfactory results for multiobjective optimization problems.

II. EVOLUTIONARY ALGORITHMS USED IN MULTIOBJECTIVE OPTIMIZATION PROBLEM

Evolutionary algorithms are a metaheuristic, based on population, which simulates the processes of evolution inspired by nature [3]. These algorithms are one of those methods that solve problems using the random behaviors. These metaheuristics are considered, in the specialized literature, as some black box in the optimization process, or weak methods of optimization.

The first implementation of an evolutionary algorithm used in multiobjective optimization belonged to David Schaffer [4], who proposed the Vector Evaluated Genetic Algorithm (VEGA). In this algorithm the functions are optimized independently, for each of these being used subpopulations formed by a numbers of chromosomes equal to the ratio between the number of chromosomes from the initial population and the number of functions to be optimized.

After a long time, a new view on multiobjective optimization problems was made public by the algorithm MOGA (MultiObjective Genetic Algorithm) [5]. In this algorithm it was introduced the concept of nondominant level, in order to optimize, simultaneously, all objective functions. Population (points obtained as a result in a certain generation of the algorithm) is organized in nondominant levels, so the first level consists of non-dominant elements, the other levels including non-dominant elements that remain if we ignore the points taken into account until the current level. From these statements we can deduce that the points located on the same level of nondomination don’t dominate each other, they dominate any of the points from a higher level of nondomination. A point is considered superior to other points if nondomination level is lower.

Given a n-dimensional vector \( x = (x_1, ..., x_n) \), it is defined in (1) the set of vectors that dominates the vector \( x \):

\[
NQ^t = \{ x' \in V | f(x') \leq f(x) \}
\]

From (1) the non-domination level for each point, at a time \( t \), can be calculated with (2):

\[
r^t_x = 1 + |NQ^t_x|,
\]

where \( |NQ^t_x| \) is the number of vectors that dominate the solution \( x \) at the moment \( t \).

This algorithm has been a model for algorithms that occurred after his appearance. He was the first who treated, in same time, the functions of the optimization problems. Many researchers have proposed various methods to solve multiobjective optimization problems, all of them having MOGA as starting point. It is a reference which is compared to any other proposed multiobjective optimization algorithm.
Two other algorithms with very good results for multiobjective optimization are: NSGA-II (Non-dominated Sorting Genetic Algorithm) and SPEA2 (Strength Pareto Evolutionary Algorithm) [6,7]. These two algorithms use different methods for the selection of the individuals who have to survive from one generation to another, which leads to satisfactory results that can be taken as a benchmark for other algorithms that solve similar problems.

Another technique for solving multiobjective optimization problems using evolutionary algorithms is Single Front Genetic Algorithm (SFGA) [8]. This algorithm is interesting because it works with a single non-dominant solution front which leads, faster, to the desired solution and the results are comparable with those obtained with NSGA-II.

Maintaining a population with high degree of diversity is an important fact in multiobjectiv problems solved by genetic algorithms. By diversity we mean an even distribution of solutions in the obtained Pareto front [9]. Without preventing measures, the population tends to be composed by several points clustered in a small area, which could lead, eventually, to an inefficient approximation of Pareto front.

Many evolutionary methods used to solve multiobjective optimization problems try to maintain a degree of diversity in the Pareto front obtained at each generation of evolution, by incorporating the information about the density of the selection process. Thus, a higher density of points around a particular solution, reduces the chance to be selected for generating new individuals [10].

### III. USING UNSUPERVISED CLASSIFICATION TECHNIQUES IN OPTIMIZATION PROCESS

In the multiobjective optimization problems, elitism means ensuring the survival of the best solutions during search process (solutions that belong to the Pareto front). Elite is a solution obtained for a problem, that is not dominated by any other point in solution space. When we solve a multiobjective problem we are dealing with a lot of functions that can act as elite. In the multiobjective optimization solved with evolutionary algorithms it is used an external population to store the best solutions obtained over evolutionary generations. First, elitism was used to prevent the loss of the Pareto optimal solutions obtained during the evolution of populations. In this case we talk about a passive strategy used to store the elites, external population being considered as a secondary population that don’t have influence on the search process.

To improve the search process, it was introduced active elitism. Unlike passive elitism, the active elitism help to create new solutions by combining external population with the current population, or by applying specific operators only on the external population. The active elitism provide faster and more robust convergence than the passive elitism, the non-dominant front obtained approximating the true Pareto front.

In each of the evolutionary algorithms generation, the external population stores the best solution from the current population. Thus, in the composition of external populations can be stored solutions located on the nondominated levels, that are close to those that approximates the Pareto front. Strategies used to modify the composition of these external populations are influenced by the size of the Pareto front, in conjunction with the size of external population, the degree of convergence that is intended to be achieved and the criteria used to ensure diversity of the Pareto front.

When we use external lists to save elite, we should answer to a series of questions. The first question is: „What kind of solutions should be stored in this list of elites?”. Most methods prefer to store non-dominant solutions within a population, so the external population is changed every time when a new solution dominates a number of solutions that already exists in the list, by adding it in the list and by removing the dominated solutions from the list. However, these searches are quite expensive. Another question to be asked is: „which should be the size of this external list?”.

This question requires an answer especially because the number of Pareto optimal solution can greatly increase, which may hinder the resolution of certain problems. For example, SPEA [11] uses a method to reduce the size of the external list to the current population size (NP), taken when the number of non-dominant solutions is greater than NP. A final question is: „which of the solutions from the external list will be selected to be reintroduced into the population?”. One possible solution is to combine the individuals from the population with those from the external list, assigning each of them a certain value depending on its position in solutions space (fitness value) and selecting the first NP individuals according to the assigned fitness value. This method is adopted in SPEA2 [7], being a leading example for the algorithms that use external lists to determine the Pareto front.

Comparatively to SPEA, in SPEA2 there is a improvement of the calculation of fitness value assigned to each individual, taking into account the individuals who dominate the current individual and the number of individuals dominated by the dominant points.

Let P be the current population, E the external list, NP the size of population, NE the size of external list. To calculate, at a time t, the fitness values for each individual from the current population and from the external list (x from \( P_t \cup E_t \)) are required the following steps [9]:

- \( r(x,t)=\sum_{y\in P_t\cup E_t, y\neq x} s(y,t) \) where \( s(y,t) \) is the number of solutions from \( P_t \cup E_t \) dominated by solution \( y \);
- it’s calculated the density \( m(x,t)=(d^k_x + 1)^{-1} \) where \( d^k_x \) is the distance between the solution \( x \) and the \( k \)-th neighbour of the solution.
• for the solution obtained, it is assigned a fitness value \(f(x,t) = r(x,t) + m(x,t)\).

The external lists will be populated in every generation, first, with non-dominant solutions from \(Pt \cup Et\). If the number of solutions of this type exceeds the size of external list, it’s made a truncation of the list, eliminating iteratively, those elements with a maximum value for \(f_v\). However, if the number of non-dominant solutions obtained is smaller than the size of external list, it is going to fill it with dominated solutions, but there are taken into account only those with the smallest fitness value. In this way it is ensured the keeping of the optimal solutions obtained, but also, a pretty good degree of scattering for the solutions from the external list.

SPEA2 algorithm has a disadvantage: the calculations needed for determining the density of the points and, finally, those necessary to determine the fitness values assigned to each individual are quite expensive.

In the algorithm proposed in this paper we use a particular method for the selection of elites, the achieved goal being the decreasing of the runtime required to obtain the points that approximates the Pareto front for the multiobjective optimization problem considered. In this algorithm it is respected the principle of elites conservation by providing their evolution from one generation to another through a fixed size external population.

Similar to the SPEA2 algorithm, the proposed algorithm starts with the determination of the non-dominant level for each individual of the population. In this stage may occur some problems that could prevent the maintenance of diversity in the search process. For example, if we are dealing with a small search space, there is a high probability to obtain, as a result, two or more overlapping points (with the same coordinates). To remove this defect we prefer to keep, from this set of points obtained in a generation of the evolutionary process, only a single point which would survive. All the other points receive some fitness values which reduce their chances to survive.

The inclusion of the unsupervised classification in the proposed algorithm in this research, started from the idea to maintain a high level of diversity for the population. It comes when we must discriminate individuals who have the same level of non-domination, in order to ensure their survival or not. Items, to be considered in fitness values of a particular point calculating, will be the centers and the dispersions of the classes which include the points that dominate the considered point and the center and dispersion for the class of the studied point. A point will have a higher probability to be selected for survival in the next generation if it is at a greater distance from the centers of classes, distances weighted by the degree of dispersion of the points from each considered class. A positive influence on the fitness value assigned for the studied point is given by the high values obtained for the diversity degree on the considered classes. In order to obtain a lowest runtime for the proposed algorithm, the classification technique is used only when it is strictly necessary, ie, when the number of individuals, from the last non-domination level, which could populate the elites set is greater than the number of free positions from this set. There will participate in classification two sets of points: the points from the external population and the points from the current population. The number of classes depending by the number of individuals from the external population. We considered that this number of classes is calculated using (3).

\[
grno = \sqrt{n_{\text{external}}},
\]

where \(n_{\text{external}}\) represents the number of individuals from the external population.

Given \(E' = Pt \cup Et\). For the clustering of the \(E'\) set in \(\text{grno}\) disjoint subsets, it was used the algorithm based on the following pseudo-code:

- the kernel \(q'1, q'2, \ldots, q'\text{grno}\) is taken based the random points chosen from the set \(E'\)
- do
  - \(Cj = \emptyset, j=1,\text{grno}\)
  - \(*\) \(qj = q'j\) where \(j=1,\text{grno}\)
  - for \(*\) \(xi \in E'\)
    - \(s = \arg \min d(xi, qj), j=1,\text{grno}\)
    - \(Cs = Cs \cup \{xi\}\)
  - end for
  - for \(j=1,\text{grno}\)
    - \(*\) determination of \(c_j\) – the centroid of the class \(C_j\)
      - \(qj' = *\) nearest point of \(c_j\)
    - end for
  - until \(*\) qj'\(=qj\), where \(j=1,\text{grno}\)

The centroids \(c_j\) of the classes \(C_j\) have coordinates equal to the average for the coordinates of the elements that belong to each individual \(j\) class.

The degree of spread for the individuals from a class has a important role in the method proposed in this paper. When we have to choose a point from two points with similar level of non-domination is preferable to choose one who is in an area more isolated from the rest of the points already choosed. A particular case may arise when one of the items taken into account is part of a class that has a number of points selected for survival and the second is part of a class that has no point selected for survival. In this case, the second point is likely to survive. It can be considered isolated against the points selected for survival to next generation. However, referring to the degree of scattering, we consider that a individual who is closer to the more dispersed classes is preferably to a point that is closer to classes with a smaller scattering degree of points. The scattering degree of \(C_k\) class is calculated with (4):

\[
s \left( C_k \right) = \frac{1}{n_{\text{classes}}-1} \sum_{j=1}^{n_{\text{classes}}} d (c_k, c_j)
\]
The concept of classification means the creation of a set of points that have some similarities. Members of one class have similarities in some characteristics. In the multiobjective optimization, it is desirable to obtain, after each generation of the evolutionary process, a set of points with a high degree of diversity, which can lead to a final result which would approximate faithfully the Pareto front. To ensure a high degree of diversity in the SPEA-ISO algorithm, it has introduced a technique of classification, which will help us to determine the points that will survive from one generation to another. Basically, this method is used for a uniform distribution of the points situated in the objective space and which should be taken into account in the evolutionary process.

In the figure 1 is shown the proposed method for the multiobjective optimization and the SPEA2 algorithm. From this diagram we can see the procedures, common to the two methods, the only one difference occurring when the densities are estimated, the result of these estimates have an important role in the selection of the individuals that will survive in a new generation.

In SPEA2 algorithm we have a density estimation technique based on an adaptation of the k-th nearest neighbor, where the fitness value assigned to each point depends on the distance between the point and the k-th neighbour. Runtime required for calculating of point’s fitness is dominated by the density estimator. It’s complexity is $O(M^2 \cdot \log M)$ [12], where $M$ is the total number of chromosomes from the current and external population.

Regarding to the proposed method, which consists of replacing the density estimator used in SPEA2 with one based on a technique used to grouping data, we obtain a lower complexity compared to the previous case: $O(M^p \cdot n \cdot gr_{no} \cdot i)$ [13], where $n$ is the size of each chromosomes, and $i$ is the number of iteration required for the classification.

### IV. RESULTS

In this paper we will make a comparison between SPEA2 and the algorithm proposed above. In order to compare the obtained results for this two algorithms, in all versions implemented we have used the same input data (DD_SDST50_ta058) [14]. The input data are related to flow-shop scheduling problems that made a scheduling of 50 processes on 20 calculation unit.

![Diagram for SPEA2 and proposed algorithm (SPEA-ISO)](image)
processes (makespan), and the second is the sum of delays for each process referred to a predetermined time (due time).

Due to random behavior that occurs in genetic algorithms, algorithms that contributing to SPEA2 and SPEA-ISO implementation, five runs were performed and, as a final result, was extracted the Pareto front under five Pareto fronts (obtained from five runs) for each algorithm.

In figure 2 we have presented the running times obtained for each algorithm, for the conditions shown in table 1. In all that 6 cases I used 1000 generations.

Pareto fronts obtained by running that two algorithms for version $P=320$ $E=60$ are shown in figure 3.

From the figure 3 we can see a set of comparable results obtained for those two algorithms. Following the reunion of those sets of points, we can determine which of these results are superior. For this we extract, from the obtained set, the best solution that could approximate the Pareto front. The result can be seen in figure 4.

By combining the results obtained using those two algorithms described in this paper (SPEA2 and SPEA-ISO), followed by the selection of the best points from the resulting set, we note that the proposed algorithm gives us better results than SPEA2. Thus, only 4 points from 43 points were obtained by using SPEA2, the rest of points being obtained by using SPEA-ISO.

The figure 4 can help us to calculate the metrics used to compare two sets of points obtained for optimization problems, techniques proposed in [11].

Let $A, B \subseteq O$, two sets composed by non-dominant points. The $C$ metric described in (6) will assign to the ordered pair $(A,B)$ a value from $[0,1]$:

$$C(A,B) = \frac{\left| \{b \in B : \exists a \in A : a - b \} \right|}{|B|}$$

From $C(A,B)=1$ results that all vectors from $B$ are dominated by vectors from $A$, from $C(A,B)=0$ results that no point from $B$ is not dominated by any point from $A$. In order to compare two sets of points using this metric, we must consider both ordered pairs $(A,B)$ and $(B,A)$, because in some cases the expression $C(A,B) = 1 - C(B,A)$ is not true. The values obtained for this metric are presented in table 2.

To compare those two sets of points from figure 4 we can use other more characteristics:
- the degree of points scattering, using (4);
- the number of points obtained for the algorithms;
- the stretching area for that two sets of points obtained.

From the figure 3 we can see a set of comparable results obtained for those two algorithms. Following the reunion of those sets of points, we can determine which of these results are superior. For this we extract, from the obtained set, the best solution that could approximate the Pareto front. The result can be seen in figure 4.

By combining the results obtained using those two algorithms described in this paper (SPEA2 and SPEA-ISO), followed by the selection of the best points from the resulting set, we note that the proposed algorithm gives us better results than SPEA2. Thus, only 4 points from 43 points were obtained by using SPEA2, the rest of points being obtained by using SPEA-ISO.

The figure 4 can help us to calculate the metrics used to compare two sets of points obtained for optimization problems, techniques proposed in [11].

Let $A, B \subseteq O$, two sets composed by non-dominant points. The $C$ metric described in (6) will assign to the ordered pair $(A,B)$ a value from $[0,1]$:

$$C(A,B) = \frac{\left| \{b \in B : \exists a \in A : a - b \} \right|}{|B|}$$

From $C(A,B)=1$ results that all vectors from $B$ are dominated by vectors from $A$, from $C(A,B)=0$ results that no point from $B$ is not dominated by any point from $A$. In order to compare two sets of points using this metric, we must consider both ordered pairs $(A,B)$ and $(B,A)$, because in some cases the expression $C(A,B) = 1 - C(B,A)$ is not true. The values obtained for this metric are presented in table 2.

To compare those two sets of points from figure 4 we can use other more characteristics:
- the degree of points scattering, using (4);
- the number of points obtained for the algorithms;
- the stretching area for that two sets of points obtained.

<table>
<thead>
<tr>
<th>Run-time(s)</th>
<th>SPEA2</th>
<th>SPEA-ISO</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>724.26</td>
<td>37.70</td>
</tr>
<tr>
<td>300</td>
<td>320.15</td>
<td>32.34</td>
</tr>
<tr>
<td>400</td>
<td>98.12</td>
<td>21.12</td>
</tr>
<tr>
<td>500</td>
<td>14.99</td>
<td>1.97</td>
</tr>
</tbody>
</table>

Fig. 2. Run-times for the algorithms

<table>
<thead>
<tr>
<th>Makespan</th>
<th>SPEA2</th>
<th>SPEA-ISO</th>
</tr>
</thead>
<tbody>
<tr>
<td>3800</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3900</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4200</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4300</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Fig. 3. Pareto front SPEA2 vs. SPEA-ISO

<table>
<thead>
<tr>
<th>Table II. Values for C Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>C(SPEA2,SPEA-ISO)</td>
</tr>
<tr>
<td>0.025</td>
</tr>
</tbody>
</table>
The stretching area means the length (Euclidean distance) between the extreme points belonging to a set, the extreme points means those points which, at least one coordinate, represent the absolute minimum for the considered objective function. All results, required for this analysis, are presented in table 3.

A set of points could be considered superior to other sets if it has a higher degree of scattering points and the stretching area of the points obtained is higher. From this statement, we can conclude that, using SPEA-ISO algorithm’s calculations, there were obtained superior results to those obtained by using SPEA2. Taking into account the number of non-dominant points, obtained using that two algorithms, and the weight of selected points within each set to be part of the front shown in figure 4, we can say that SPEA-ISO is superior to SPEA2. So for SPEA-ISO, 97.5% (39/40) of points, will feed into the final Pareto front, while for SPEA2 only 13.33% (4/30) of points will be part of the final front.

To evaluate the quality of those two sets of non-dominant points, we can use the $S$ metric [11], which calculates the area (hyper-volume) of the region dominated by each set of points. This metric implies to use a reference point from the dominated space, to help calculate the desired areas. A disadvantage of this metric is given by the possibility of choosing of an inappropriate reference point, which can cause an incorrect calculation of areas related to the considered sets. To eliminate this disadvantage, we take into account a point, whose coordinates are higher to the maximum values achieved for that two objective functions, for both sets of non-dominant points. Reference point considered in our case was Pr(4250,18900). It will lead to the results presented in table 4.

From this metric we can see that the solution offered by SPEA-ISO is superior to that offered by SPEA2.

V. CONCLUSION

In a variety of areas, there are a multitude of problems, whose solution requires the achievement of one or more objectives. When we are dealing with a single objective, the situation is quite simple. Problems arise when we should optimize two or more functions.

Most times, these functions, that must be optimized, are contradictory, the optimization of one of them leading to removal of the optimum value for the other functions to be optimized.

Over time, was proposed a variety of methods to solve such a problems, many of them resorting to the use of genetic algorithms in generating of satisfactory solution. From the algorithms of this class, SPEA2 provides superior solutions [12], but a major drawback of this algorithm is given by the higher computational effort required to obtain results.

To eliminate the disadvantage mentioned above, this paper proposed an algorithm that uses a data clustering technique, a method that leads to superior results to those obtained with SPEA2, while the runtime is smaller than that obtained for the referenced algorithm.

REFERENCES


Distributed and Network-based Processing (EUROMICRO-PDP’02), 2002


Ionuț Balan received his MS and Ph.D. degrees in Computer Science from “Stefan cel Mare” University, Romania in 2009 and 2012 respectively. He is currently a Lecturer in “Stefan cel Mare” University, Faculty of Economics and Public Administration, Suceava. His main research interests include high performance computing, artificial intelligence, data mining, databases.