Evolutionary Multi-modal Optimization with the Use of Multi-objective Techniques

Leszek Siwik and Rafał Dreżewski

AGH University of Science and Technology, Department of Computer Science,
Kraków, Poland
{siwik,drezew}@agh.edu.pl

Abstract. When evolutionary algorithms for solving multi-modal optimization problems are applied, the crucial issue to be solved is maintaining population diversity to avoid drifting and focusing individuals around single global optima. A lot of techniques have been used here so far. Simultaneously for last twenty years a lot of effort has been made in the area of evolutionary algorithms for multi-objective optimization. As the result at least several highly efficient algorithms have been proposed such as NSGAII or SPEA2. Obviously, also in this case maintaining of population diversity is crucial but this time, taking the specificity of optimization in the Pareto sense, there are built-in mechanisms to solve this issue effectively. If so, the idea arises of applying of state-of-the-art evolutionary multi-objective optimization algorithms for solving not original multi-modal (but single-objective) optimization task but rather its transformed into multi-objective problem form by introducing additional dispersion-oriented criteria. The goal of this paper is to present some further study in this area.

1 Motivation

One of the most important issue regarding multi-modal optimization is the ability for discovering not only the global but also (as many as possible) local optima (modes). When evolutionary solver is applied it is inseparably connected with keeping population dispersed and not focusing individuals around the global optima. Many techniques responsible for maintaining population diversity have been proposed so far. It is enough to call techniques based on modification of mechanism of selecting individuals for new generation (crowding model), modification of parent selection (fitness sharing, sexual selection), restricted application of selection and/or recombination mechanisms (grouping individuals into sub-populations, introducing environment with some topography etc.) [7] just to mention a few. Each of them however has its own shortcomings and it is not possible to point out a single diversity-maintaining technique giving evidently the best results and to be used in all (or at least in the majority of) cases. What is important their efficiency and the effectiveness depends often on the optimization algorithm used.

For the last thirty years evolutionary multi-objective optimization algorithms (EMOAs) have become more and more popular [4,11]. Historically, one tried

L. Rutkowski et al. (Eds.): ICAISC 2014, Part I, LNAI 8467, pp. 428–439, 2014.

to use classical EAs by combining all objectives in one single objective and repeating algorithm runs with different weights assigned to particular objectives to obtain different non-dominated solutions. The advantage of such an approach is its simplicity, however it is pretty unnatural, slow (since the EA has to be (re)run at least as many times as the number of solutions should be found) and—what is the most important—depending on the definitions of the objective functions (and their combination)—it often turns out that combining objectives with different weights results with the same solution, what makes this approach simply useless.

Also another techniques consisting in redefining multi-objective problem into single-objective one (and then (re)running single-objective algorithms to find consecutive non-dominated solutions, one in single algorithm's run) turned out to be useless in particular cases. It is enough to mention for instance ε—constrains technique which is useless in the case of concave problems.

That is why a lot of effort has been made to develop efficient and effective evolutionary (as general and population-based) algorithms for multi-objective optimization. It has been performed successfully and such algorithms as SPEA-II [20,19] or NSGA-II [14] are nowadays state-of-the-art EMOAs giving a really high-quality results in most cases. Also, agent-based multi-objective evolutionary algorithms (combining agent-based and evolutionary paradigms) were proposed and they proved to be quite effective in some cases (for example in multi-objective portfolio optimization problems) [5,6,8,9].

What is important, when the multi-objective optimization (and algorithms) (in the Pareto sense) are being considered as one of the most important difference in comparison to single objective optimization (algorithms) is the fact that the solution to be found is the whole set of non-dominated alternatives called the Pareto set (or the Pareto frontier in the objective space). The crucial here is the fact that using (weak) non-domination relation instead of simple mutual-comparisons as a mechanisms responsible for distinguishing "better" and "worse" alternatives—EMOAs are dedicated for looking for the whole set of solutions in one single run. One has to remember that the goal of the multi-objective optimization (in the Pareto sense) is to find (as-many-as-possible) non-dominated solutions dispersed over the whole Pareto frontier. Since EMOAs are population-based it is obviously the more so simple and natural but—what is crucial here—they have natural, built-in mechanisms for maintaining population diversity as well as the diversity of the solution itself.

The question thus arises if—in contrast to historical modifications of multi-objective optimization problems into single-objective one(s)—the way for obtaining high-quality solutions of multi-modal optimization tasks is converting multi-modal problems into multi-objective optimization problems by introducing additional objective responsible for maintaining population dispersed and then applying for solving such a modified problem one of the state-of-the-art efficient evolutionary multi-objective optimization algorithms.

Obviously such experiments have already been conducted. It is enough to mention here the work of M. Preuss, G. Rudolph and F. Tumakaka [12] but it still seems to be only a putting a toe into the water and the goal of this paper is to follow this research direction and to make some comparative assessment of several dispersing-oriented objectives introduced as a second objective while converting multi-modal single-objective optimization task into multi-objective optimization problem with the special attention paid to clustering method.

The computing experiments presented in this paper may be treated as preliminary results, planned to be adapted and ported to ParaPhrase[1] agent-based computing platform, which supplies hybrid CPU/GPU computing infrastructure via dedicated virtualisation tools.

2 The Idea of Transformation of Multi-modal into Multi-objective Optimization Problem

Typically, multi-objective (or multi-criteria) optimization problem (MOOP) is formulated as follows ([1,19,4]):

$$MOOP \equiv \begin{cases} Min/Max: \quad f_l(\bar{x}), \quad l = 1, 2 \ldots, L \\ Taking\ into\ consideration: \\ \quad g_j(\bar{x}) \geq 0, \quad j = 1, 2 \ldots, J \\ \quad h_k(\bar{x}) = 0, \quad k = 1, 2 \ldots, K \\ \quad x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2 \ldots, N \end{cases}$$

The set of constraints, both equalities $(h_k(\bar{x}))$, as well as inequalities $(g_j(\bar{x}))$, and constraints related to the decision variables, i.e. lower bounds $(x_i^{(L)})$ and upper bounds $(x_i^{(U)})$, define so called searching space—feasible alternatives $(\mathcal{D})$. Because of space limitation it is enough to say in this place that in the course of this paper multi-objective optimization in the Pareto sense is considered, so solving of defined problem means determining of all feasible and non-dominated alternatives from the set $(\mathcal{D})$. Such defined set is called Pareto set $(\mathcal{P})$ and in objective space it forms so called Pareto frontier $(\mathcal{PF})$.

Simultaneously, the multi-modal optimization task (assuming minimization) means determining of all $\boldsymbol{x}^+ \in D$ such as $\exists \epsilon > 0 \forall \boldsymbol{x} \in D \parallel \boldsymbol{x} - \boldsymbol{x}^+ \parallel < \epsilon \Rightarrow f(\boldsymbol{x}) \geq \boldsymbol{x}^+$ [2].

So, proposed transformation of multi-modal (but single-objective) into multi-objective optimization problem consists in formulating MOOP with original multi-modal function and dispersing oriented function as the second objective with preserving all original constraints and bounds of course.

$$MOOP \equiv \begin{cases} Min/Max: \quad f_m(\bar{x}), original\ multi-modal\ function \\ Min/Max: \quad f_d(\bar{x}), dispersing-oriented\ function \\ Taking\ into\ consideration: \\ \quad g_j(\bar{x}) \geq 0, \quad j = 1, 2 \ldots, J \\ \quad h_k(\bar{x}) = 0, \quad k = 1, 2 \ldots, K \\ \quad x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2 \ldots, N \end{cases}$$

[1] `http://paraphrase-ict.eu`

It can be said that such transformation unnecessarily complicates a problem to be solved because it makes multi-objective optimization problem from a single-objective one. However solving multi-modal single-objective problem (finding all global and local optima) is also not an easy task—there were lots niching techniques for evolutionary algorithms proposed and none of them is simple and perfect. Paradoxically converting such a problem into multi-objective one can lead to constructing simple and efficient techniques for evolutionary algorithms, especially that we utilize well established and very efficient evolutionary multi-objective algorithms.

3 Variants of Dispersion–Oriented Objective

During our experiments following variants of the second objective have been tested: fitness sharing, centroid method, weighted dispersion criteria and clustering.

Fitness sharing is classical niching technique consisting in (artificial) decreasing the value of fitness function according to the (higher) number of direct neighbors of given individual. Obviously there are some issues and decisions to be made (e.g. determining the radius of the neighborhood, determining the distance metrics and making a decision if it is calculated in the objective or in a decision variable space, determining how "density" is calculated and what is its influence on the fitness function value).

Discussion regarding above aspects can be found for instance in [4]. In its most popular version it is described according to the formula $f^{FS}(x_i) = \frac{f(x_i)}{m_i}$, where m_i is the sum of sharing function values defined as $m_i = \sum_{j=1}^{N} sh(d(x_i, x_j))$ and

$$f(x) = \begin{cases} 1 - \left(\frac{d(x_i,x_j)}{\sigma_{sh}}\right)^{\alpha} & ,x > 0 \\ 0 & ,x = 0 \end{cases} \tag{1}$$

where σ_{sh} is a radius of the niche and α parameter determines the shape of the fitness sharing function (usually equals 1).

Centroid based method is a simple in assumption and easy in implementation method for dispersing the population. The fitness value of the specimen is increased according to its (increasing) distance to the population center of gravity calculated as $\overrightarrow{\boldsymbol{x_c}} = \frac{\sum_{i=0}^{N} \overrightarrow{\boldsymbol{x_i}}}{N}$.

Weighted dispersion criteria technique tries to address one of the most significant problems observed in evolutionary multi-modal optimization: concentration of the whole population (which is usually intensifying over the course of time/iterations) around "strong" individuals, especially individuals located nearby the global optima. As a consequence of this phenomena the loss of the population diversity is observed and the chance for discovering (as many as possible) local optima is lower and lower. So the question is if it is not a good idea while introducing the second objective and converting multi-modal single objective problem into multi-objective optimization problem introducing the second criteria as a function which value would be inversely proportional to the value of the first criteria. In such a way strong individuals (from the first—crucial objective perspective) will not be able to "dominate" and to attract the rest of the population to their neighborhood. Simultaneously those individuals will not be lost by the population since they are "strong" as regards the first objective (so they won't be dominated in the Pareto domination relation). So assuming the first objective as a multi-modal function $F(x)$ with a global optima $M = F(x_{max})$ the second objective $S_{weighted}$ can be defined as $S_{weighted} = \alpha * (F(x_i)/F(x_{max}) * S(x_i)$, where: α is a weighting coefficient, $S(x_i)$ is the original value of dispersing function, $F(x_i)$ and $F(x_{max})$ are current and maximum values of the original (multi-modal) function (i.e. the first objective in fact).

One of interesting and (especially taking presented in section 4 selected preliminary results) promising technique is **clustering**. One of the fundamental question that can be considered is whether any of dispersion-oriented technique (i.e. the second objective after converting multi-modal into multi-objective optimization task) should be applied globally or "locally" i.e. within windows dividing the whole domain into sub-domain(s).When using clustering as a dispersion-oriented technique firstly all clusters are identified and then the fitness of individuals that are located outside or at the borders of the clusters is increased and the fitness of individuals that are located inside clusters is decreased proportionally to their distance from the center of the cluster.

Generally, research on clustering techniques and genetic algorithms was conducted in two areas: using evolutionary algorithms as a clustering technique [10,17,13,3] and using a clustering technique in evolutionary algorithm in order to find multiple solutions of multi-modal (but single criteria) problems [16,15]. We used clustering technique together with evolutionary algorithm as the mechanism of dispersing individuals over the solution space (as the second objective) during solving multi-modal problems converted into multi-objective ones.

For the purposes of making experiments unsupervised k-windows clustering algorithm has been implemented and used [18]. It is using a window(s)-based technique for determining possible clusters. Algorithm initializes a given number of 2-dimensional windows over the set of individuals. Then, it is moving on windows and enlarges them to cover existing clusters. Next, when all moving and enlarging operations have been performed—consolidation is being performed. All overlapping windows are either consolidated or skipped depending on the number of individuals belonging to the overlapped windows. In the consequence, the algorithm is able to reduce reasonably the (large) number of (possible) clusters identified originally at the beginning.

Algorithm consists of two crucial functions: movement and enlargement. The goal of movement function is setting the window as close to the center of the cluster as possible. Movement function is performed iteratively as long as the distance of the center of new window reaches the threshold value Θ_v (set experimentally).

The goal of enlargement operation is to improve the number of individuals belonging to the particular window. The window is being enlarged by Θ_e value in each dimension. Appropriate enlargement is the one assuring improving the number of individuals belonging to the given window with the number higher than Θ_c threshold value. If the number of new individuals belonging to the given window is smaller than Θ_e value then the last step of enlargement function is being withdrawn.

The crucial issue with using clusters is determining the number of clusters covering the whole population in the most appropriate way. In k-window algorithm it is determined by the algorithm itself during its work. To achieve that effectively, relatively the significant number of windows is needed at the beginning. After performing moving and enlarging operation pretty big number of windows are overlapping. So merging function is performed then. To do that—the number of "common" i.e. belonging to overlapped windows individuals is determined and then:

- if it is larger than the threshold value Θ_s windows are treated as parts of the same cluster and the smaller one is being removed;
- otherwise both windows are merged;
- if windows overlap but neither merging nor eliminating threshold is achieved, it is assumed that windows (their individuals) belong to different clusters.

```
Data: $a$, $\Theta_e$,$\Theta_m$,$\Theta_c$,$\Theta_v$,$k$
Result: clusters $c_{11}, c_{12}, \ldots$
begin
    $W \longleftarrow DetermineInitialWindows(k, a)$;
    for $w_j \in W$ do
        while The center or the size change do
            $movement(\Theta_v, w_j)$;
            $enlargement(\Theta_e, \Theta_c, \Theta_v, w_j)$;
        end
    end
    $merge(\Theta_m, \Theta_s, W)$
end
```

Algorithm 1. Unsupervised k-windows clustering algorithm

There is a pretty big number of parameters influencing significantly the behavior of the algorithm i.e.:

- the ratio between the initial number of windows and the number of individuals in population. It should be relatively high to spread windows among all clusters. During experiments it was set to 10%. (For the population with 1000 individuals it was set to 100 windows);
- the initial size of the window—it was determined experimentally;
- the minimum distance between windows at the beginning. It is important parameter to avoid overlapping windows during initialization;

```
Data: $k$,$a$
Result: a set $W$ of $k\ d-ranges$
begin
    initialize k d-ranges windows $w_{m1}, \ldots, w_{mk}$ each of size $a$;
    select k random points from the dataset and center the d-ranges at these points
end
```

Algorithm 2. DetermineInitialWindows

```
Data: $a$, $\Theta_v$,a d-range w
begin
    while The distance between m and the previous center of w is greater or equal to $\Theta_v$ do
        find the patterns that lie within the d-range w ;
        calculate the mean m of these patterns ;
        set the center of w equal to m ;
    end
end
```

Algorithm 3. Operation *movement*

```
Data: $\Theta_e$,$\Theta_v$,$\Theta_c$,a, d-range w
begin
    while The increase in number of patterns is $\geq \Theta_c\%$ across every $d_i$ do
        for Each coordinate $d_i$ do
            while The increase in number of patterns across $d_i$ is $\geq \Theta_c\%$ do
                enlarge $w$ across $d_i$
                $movement(\Theta_v, w)$
            end
        end
    end
end
```

Algorithm 4. Operation *enlargement*

- the movement threshold (Θ_v)—it defines the minimum distance between the new and the current gravity center of the window during its movement. When this value is not achieved movement operation is finished;
- the enlargement increase ratio (Θ_e)—it is a percentage ratio between the old and the new window size in consecutive steps of enlargement operation. During experiments it was set to 10% for each dimension respectively.
- enlargement stop ratio threshold (Θ_c)—the factor defining the minimum increase of the number of new individuals in the window when enlargement operation is performed. During experiments presented in this paper it was defined as $enlargement\_stop\_threshold = \frac{enlargement\_increase\_ratio}{init\_window\_population\_ratio}$
- merge ratio (Θ_s) is the minimum number of common individuals belonging to two windows to merge them. During experiments it was set to 80%;

```
Data: $\Theta_m$,$\Theta_s$,a set $W$ of $d-ranges$
begin
    for Each not marked $d-range$ $w_j \in W$ do
        mark $w_j$ with label $w_j$ ;
        if $\exists\ w_i \neq w_j \in W$ that overlaps with $w_j$ then
            compute the number of points $n$ that lie in the common part of
            windows ;
            if $n/ \mid w_i \mid \geq \Theta_s$ and $\mid w_i \mid < \mid w_j \mid$ then
                disregard $w_j$
            end
            if $0.5(n/ \mid w_j \mid + n/ \mid w_i \mid) \geq \Theta_m$ then
                mark all $w_j$ labeled d-ranges in W with label $w_j$
            end
        end
    end
end
```

Algorithm 5. Operation *merging*

- merge disregard ratio (Θ_m) is the minimum ratio of common individuals belonging to two windows to remove one of them (the smaller one). During experiments it was set to 90%.

4 Experimental Results

As a multi-modal benchmarks Michalewicz's, Rastrigin's and Schwefel's functions have been used. As a second (dispersion related) objective: fitness sharing, centroids and weighted centroids methods have been applied. As experimental tool jEMO framework has been used[2]. Because of the space limitations only a few experimental results are here presented.

First results obtained without clustering mechanism are presented. In table 1 there are listed the most important parameters of this experiment. As one may see in figure 2 transforming classical multi-modal optimization problem into multi-objective one and applying NSGA-II algorithm for solving such modified problem with centroids as a dispersion-oriented second objective allows for obtaining pretty promising results. They differ of course depending on particular parameters used but generally speaking results are promising.

For comparison in table 2 there are listed parameters of sample experiment where dispersion was applied "locally" i.e. within clusters discovered by described in section 3 k-window clustering algorithm. This time experiment was performed with the use of Michalewicz benchmark and typical obtained results are presented in figure 1. As one may see obtained results are also promising and encouraging for further research.

[2] `code.google.com/p/jemo/`

Table 1. Selected parameters taken in experiment 1

Parameter	Value
Original function	Rastrigin
Distribution function	Centroid
Optimization algorithm	NSGAII
Population size	1000
Number of generations	40
Mutation	Radial mutation
Mutation probability	0.5
Strong mutation probability	0.15
Domain control type	Move to domain border
Specimen repairing	None
Recombination	Radial crossover
Recombination probability	0.5
Domain control type	Move to border
Specimen repairing	None
Selection	Classical tournament
Tournament size ratio	80%
Tournament probability	0.8
Clustering	none

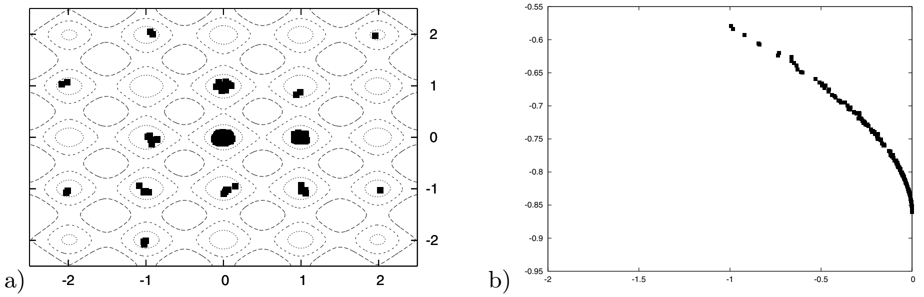

Fig. 1. Results obtained in experiment 1. Found solutions (a) and Pareto frontier (b).

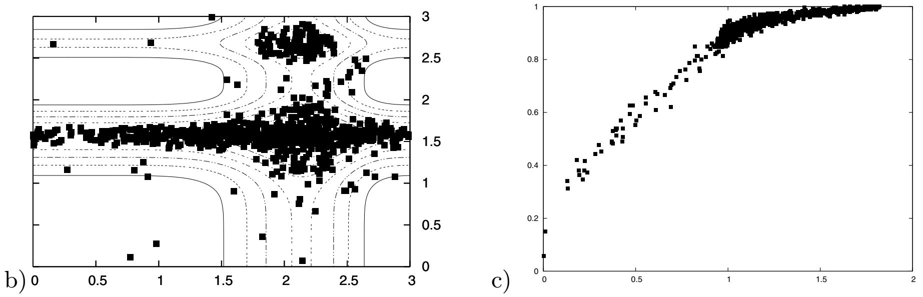

Fig. 2. Results obtained in experiment 2. Found solutions of: (a) multi-modal problem and (b) multi-objective problem

Table 2. Selected parameters taken in experiment 2

Parameter	**Value**
Original function	Michalewicz
Distribution function	Centroid
Optimization algorithm	NSGAII
Population size	1000
Number of generations	40
Mutation	Radial mutation
Mutation probability	0.5
Strong mutation probability	0.15
Domain control type	Move to domain border
Specimen repairing	None
Recombination	Radial crossover
Recombination probability	0.5
Domain control type	Move to border
Specimen repairing	None
Selection	Classical tournament
Tournament size ratio	80%
Tournament probability	0.8
Clustering	yes
Initial window's size	[0.4][0.4]
Initial number of windows	500
Movement threshold (Θ_v)	0.1
Enlargement increase step	0.08
Enlargement stop ratio threshold (Θ_c)	0.2
Merge ratio (Θ_S)	0.9
Merge disregard ratio (Θ_m)	1

5 Summary and Conclusions

When evolutionary algorithms for solving multi-modal optimization problems are applied the crucial issue to be solved is maintaining population diversity to avoid drifting and focusing individuals around single global optima. A lot of techniques have been proposed and used here so far.

Simultaneously, for the last twenty years a lot of effort has been made in the area of evolutionary algorithms for multi-objective optimization. As the result at least several highly efficient algorithms have been proposed such as NSGAII or SPEA2. Obviously, also in this case maintaining of population diversity is crucial but this time taking the specificity of optimization in the Pareto sense there are built-in mechanisms to solve this issue effectively.

If so, the idea arises of applying state-of-the-art evolutionary multi-objective optimization algorithms for solving not original multi-modal (but single-objective) optimization task but its transformed into multi-objective problem form by introducing additional dispersion-oriented criteria as it is discussed in section 2.

One of important issues is the definition of the dispersion-oriented criteria. In the course of this paper some of them, i.e. classical fitness sharing, centroids, weighted centroids have been discussed.

On the basis of some observations taken during experiments the idea of applying the second objective not globally but locally within some areas of concentration of individuals arose. To put this idea into practice k-window clustering algorithm has been implemented and applied and then dispersion-oriented mechanisms have been applied not globally but within formed windows.

Because of the space limitations it is impossible to present comprehensive review of obtained results especially that there are many parameters influencing the behavior and effectiveness of the proposed approach. Nevertheless it can be said for sure that preliminary results are promising and encourage for further research in this area.

Acknowledgments. The research presented in the paper was partially supported by the European Commission FP7 through the project *ParaPhrase: Parallel Patterns for Adaptive Heterogeneous Multicore Systems*, under contract no.: 288570 (`http://paraphrase-ict.eu`) and by Polish Ministry of Science and Higher Education under AGH University of Science and Technology Grant No. 11.11.230.015 (statutory project).

References

1. Abraham, A., Jain, L.C., Goldberg, R.: Evolutionary Multiobjective Optimization Theoretical Advances and Applications. Springer (2005)
2. Byrski, A., Dreżewski, R., Siwik, L., Kisiel-Dorohinicki, M.: Evolutionary multi-agent systems. The Knowledge Engineering Review (to be published, 2014)
3. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York (2006)
4. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons (2008)
5. Dreżewski, R., Obrocki, K., Siwik, L.: Agent-based co-operative co-evolutionary algorithms for multi-objective portfolio optimization. In: Brabazon, A., O'Neill, M., Maringer, D.G. (eds.) Natural Computing in Computational Finance. SCI, vol. 293, pp. 63–84. Springer, Heidelberg (2010)
6. Dreżewski, R., Sepielak, J.: Evolutionary system for generating investment strategies. In: Giacobini, M., et al. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 83–92. Springer, Heidelberg (2008)
7. Dreżewski, R., Siwik, L.: Techniques for maintaining population diversity in classical and agent-based multi-objective evolutionary algorithms. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007, Part II. LNCS, vol. 4488, pp. 904–911. Springer, Heidelberg (2007)
8. Dreżewski, R., Siwik, L.: Agent-based co-operative co-evolutionary algorithm for multi-objective optimization. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 388–397. Springer, Heidelberg (2008)

9. Drezewski, R., Siwik, L.: Co-evolutionary multi-agent system for portfolio optimization. In: Brabazon, A., O'Neill, M. (eds.) Natural Computing in Computational Finance. SCI, vol. 100, pp. 271–299. Springer, Heidelberg (2008)
10. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., de Carvalho, A.C.P.L.F.: A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 39(2) (2009)
11. Marler, R., Arora, J.: Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization 26(6) (2004)
12. Preuss, M., Rudolph, G., Tumakaka, F.: Solving multimodal problems via multiobjective techniques with application to phase equilibrium detection. In: IEEE Congress on Evolutionary Computation. IEEE (2007)
13. Sarafis, I.A., Trinder, P.W., Zalzala, A.: Towards effective subspace clustering with an evolutionary algorithm. In: Sarker, R., et al. (eds.) Proceedings of the 2003 Congress on Evolutionary Computation, vol. 2. IEEE Press (2003)
14. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation 2(3), 221–248 (1994)
15. Streichert, F., Stein, G., Ulmer, H., Zell, A.: A clustering based niching method for evolutionary algorithms. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 644–645. Springer, Heidelberg (2003)
16. Tasoulis, D.K., Plagianakos, V.P., Vrahatis, M.N.: Clustering in evolutionary algorithms to efficiently compute simultaneously local and global minima. In: Congress on Evolutionary Computation. IEEE (2005)
17. Tasoulis, D.K., Vrahatis, M.N.: The new window density function for efficient evolutionary unsupervised clustering. In: Congress on Evolutionary Computation. IEEE (2005)
18. Vrahatis, M.N., Boutsinas, B., Alevizos, P., Pavlides, G.: The new k-windows algorithm for improving the k-means clustering algorithm. J. Complex. 18(1) (March 2002)
19. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: methods and applications. PhD thesis, Swiss Federal Institute of Technology, Zurich (1999)
20. Zitzler, E.: Evolutionary algorithms, multiobjective optimization, and applications (September 2003)