Ensemble of Multimodal Genetic Algorithms for Design and Decision Making Support Problems

Evgenii Sopov, Eugene Semenkin and Ilia Panfilov
Department of Systems Analysis and Operations Research, Siberian State Aerospace University, Krasnoyarsk, Russia

Keywords: Genetic Algorithms, Ensemble Methods, Multimodal Optimization, Selective Hyper-Heuristic.

Abstract: Many problems of design and decision making support can be stated as optimization problems. For real-world problems, sometimes it is necessary to obtain many alternative solutions to the problem. In this case multimodal approach can be used. The goal of multimodal optimization (MMO) is to find all optima (global and local) or a representative subset of all optima. In recent years many efficient nature-inspired techniques have been proposed for real-valued MMO problems. At the same time, real-world design and decision making support problems may contain variables of many different types, including integer, rank, binary and others. In this case, the weakest representation (namely binary representation) is used. Unfortunately, there is a lack of efficient approaches for problems with binary representation. In this study, a novel approach based on a selective hyper-heuristic in a form of ensemble for designing multi-strategy genetic algorithm is proposed. The approach controls the interactions of many search techniques (different genetic algorithms for MMO) and leads to the self-configuring solving of problems with a priori unknown structure. The results of numerical experiments for benchmark problems from the CEC competition on MMO and for some real-world problems are presented and discussed.

1 INTRODUCTION

Decision making involves the choice of one or more alternatives from a list of options. The list of options usually contains both good and bad (more or less acceptable) solutions. The aim of rational decision making is to maximize some criterion that describes quality of the choice (Sen and Yang, 2012). Design problems can be formulated in the same way (Ray and Liew, 2002), the only difference is that alternatives are not defined beforehand. It is obviously that such problems can be stated as optimization problems. In this case, alternatives are considered as candidate-solutions, and quality criteria are considered as objectives. Additional requirements can be performed by constraints.

Many real-world design and decision making support problems are complex and bad-formalized, thus the quality criterion is usually considered as the "black-box" model. Moreover, alternatives are represented by complex structures that contain variable parameters of many different types, including categorical, integer, rank, binary and others. Such optimization problems require implying more advanced optimization techniques like evolutionary algorithms (EA).

The general EA scheme uses the conception of collective search based on the natural selection and nature-inspired (genetic and evolutionary) operations (Holland, 1975; Goldberg, 1989). All EAs in this study are assumed to be binary genetic algorithms (GAs).

From many practical points of view, the only solution to the problem can be not enough, even it is the optimal solution. For example, we need fallback solutions if the optimal solution can't be realized. Moreover, identification of many different (optimal and suboptimal) solutions is useful for better understanding of the problem.

Optimization problems that have more than one optimal solution (or there exists only one global optimum and several local optima in the feasible solution space) are called multimodal. The goal of multimodal optimization (MMO) is to find all optima (global and local) or a representative subset of all optima. EAs and GAs are efficient in the multimodal environment as they use a stochastic population-based search. At the same time, traditional EAs and GAs have a tendency to converge to the best-found

Sopov, E., Semenkin, E. and Panfilov, I.
Ensemble of Multimodal Genetic Algorithms for Design and Decision Making Support Problems.
DOI: 10.5220/0005976401600167
In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016) - Volume 1*, pages 160-167
ISBN: 978-989-758-198-4

optimum losing population diversity. In recent years MMO have become more popular, and many efficient nature-inspired MMO techniques were proposed. Almost all search algorithms are based on maintaining the population diversity, but differ in how the search space is explored and how optima basins are located and identified over a landscape. The majority of algorithms and the best results are obtained for real-valued MMO problems (Das et al., 2011). Unfortunately, there is a lack of efficient approaches for problems with binary representation. Existing techniques are usually based on general ideas of niching and fitness sharing. Heuristics from efficient real-valued MMO techniques cannot be directly applied to binary MMO algorithms because of dissimilar landscape features in the binary search space.

In this study, a novel approach based on a selective hyper-heuristic in a form of ensemble of MMO GA is proposed. Its main idea is to use many MMO techniques with different search strategies and adaptively control their interactions. Such an approach would lead to the self-configuring solving of problems with a priori unknown structure.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 describes the proposed approach. In Section 4 the results of numerical experiments are discussed. In the Conclusion the results and further research are discussed.

2 RELATED WORK

The problem of GA-based design and decision making support is well-studied (Sen and Yang, 2012; Kaklaukas, 2015). At the same time, many complex real-world problems are still a challenge for GAs and other nature-inspired techniques.

One of the ways for finding many efficient alternatives is multi-objective problem statement (Li et al., 2015). In this case, a set of Pareto-optimal solutions is obtained instead of the only optimal. The multi-objective statement needs more advanced GA-based techniques, and subsequent analysis of the obtained Pareto set approximation. The Pareto set can also contain very contrast solutions that are interesting from the mathematical point of view as they are still Pareto-optimal, but are not acceptable from the practical point of view.

Another way is MMO statement. Over the past decade interest for this field has increased. The recent approaches are focused on the goal of exploring the search space and finding many optima to the problem. Many efficient algorithms have been proposed. Good survey of widespread MMO techniques can be found in (Deb and Saha, 2010; Das et al., 2011; Liu et al., 2011). As we can see from many studies, there is no universal approach that is efficient for all MMO problems. Many researches design hybrid algorithms, which are generally based on a combination of search algorithms and some heuristic for a niching improvement. Another way is a combining many basic MMO algorithms to run them in parallel, migrate individuals and combine the results. In (Bessaou et al., 2000) an island model is applied, where islands are iteratively revised according to the genetic likeness of individuals. In (Yu and Suganthan, 2010) four MMO niching algorithms run in parallel to produce offspring, which are collected in a pool to produce a replacement step. In (Qu et al., 2012) the same scheme is realized using the clearing procedure.

The conception of designing MMO algorithms in the form of an ensemble seems to be promising. A selective hyper-heuristic (Burke et al., 2010) that includes many different MMO approaches (different search strategies) can deal with many different MMO problems. And such a hyper-heuristic can provide self-configuration due to the adaptive control of the interaction of single algorithms during the problem solving. This idea was implemented in (Sopov, 2015a). The approach has demonstrated good results with respect to multi-objective and non-stationary optimization. In this study, we will apply this concept to the MMO problem.

3 PROPOSED APPROACH

Heuristic and meta-heuristic search algorithms for complex optimization problems are well-studied and widely discussed (Bianchi et al., 2009; Boussaida et al., 2013). One of the applications of the heuristic search algorithms is the design of EAs. In other words, a heuristic is used to design a heuristic, and it is called hyper-heuristic (Ross, 2005; Burke et al., 2010; Maashi et al, 2015). There also exist examples of hyper-hyper-heuristics, which are the extension of the idea of hyper-heuristics to select or combine hyper-heuristics and generate new hyper-heuristics (Pillay, 2015).

In this study, we will present a hyper-heuristic for the design and control of the GA ensemble. The most important step of the multi-EA search is the interaction of component EAs. The general approach is the island model with random migrations of individuals. The majority of the proposed techniques are based on the "winner-take-all" concept. There also exist coevolutionary approaches that are usually based on a problem decomposition. In this study, we will combine many interaction methods.

The ensemble method can be also used in the field of EA. The main idea is to include different search strategies in the ensemble and to design effective control of algorithm interaction. Our hypothesis is that different EAs are able to deal with different features of the optimization problem, and the probability of all algorithms failing with the same challenge in the optimization process is low. Moreover, the interaction of algorithms can provide the ensemble with new options for optimization, which are absent in stand-alone algorithms.

The general structure of the self-configuring multi-strategy genetic algorithm proposed in (Sopov, 2015a) is called Self*GA (the star sign corresponds to the certain optimization problem.

The total population size (or the sum of populations of all component algorithms) is called the computational resource. The resource is distributed between algorithms, which run in parallel and independent over the predefined number of iterations (called the adaptation period). All algorithms have the same objective and use the same encoding (solution representation). All populations are initialized at random. After the distribution, each GA included in Self*GA has its own population which does not overlap with populations of other GAs. At the first iteration, all algorithms get an equal portion of the resource.

After the adaptation period, the performance of individual algorithms is estimated with respect to the objective of the optimization problem. After that, algorithms are compared and ranked. Search strategies with better performance increase their computational resource (the size of their populations).

We will discuss the design of a Self*GA for MMO problems that can be named SelfMMOGA.

At the first step, we need to define the set of individual algorithms included in the SelfMMOGA. In this study we use six basic techniques, which are well-studied and discussed (Singh and Deb, 2006; Das et al., 2011), and they can be used with binary representation with no modification. We have included the following component algorithms in the SelfMMOGA: Clearing (Alg1), Sharing (Alg2), Clustering (Alg3), Restricted Tournament Selection or RTS (Alg4), Deterministic Crowding (Alg5) and Probabilistic Crowding (Alg6).

The motivation of choosing certain algorithms is that if the SelfMMOGA performs well with basic techniques, we can develop the approach with more complex algorithms in further works.

The adaptation period is a parameter of the SelfMMOGA. Moreover, the value depends on the limitation of the computational resource (total number of fitness evaluations).

The key point of any coevolutionary scheme is the performance evaluation of a single algorithm. For MMO problems performance metrics should estimate how many optima were found and how the population is distributed over the search space. Unfortunately, good performance measures exist only for benchmark MMO problems, which contain knowledge of the optima. Performance measures for black-box MMO problems are still being discussed. Some good recommendations can be found in (Preuss and Wessing, 2013). In this study, the following criteria are used.

The first measure is called Basin Ratio (BR). The BR calculates the number of covered basins, which have been discovered by the population. It does not require knowledge of optima, but an approximation of basins is used. The BR can be calculated as

$$BR(pop) = \frac{l}{k}$$
$$l = \sum_{i=1}^{k} min\left\{1, \sum_{\substack{x \in pop \\ x \neq z_i}} b(x, z_i)\right\} \tag{1}$$
$$b(x,z) = \begin{cases} 1, if\ x \in basin(z) \\ 0, otherwise \end{cases}$$

where pop is the population, k is the number of identified basins by the total population, l is the indicator of basin coverage by a single algorithm, b is a function that indicates if an individual is in basin z.

To use the metric (1), we need to define how to identify basins in the search space and how to construct the function $b(x,z)$.

For continuous MMO problems, basins can be identified using different clustering procedures. In this study, for MMO problems with binary representation we use the following approach. We use the total population (the union of populations of all individual algorithms in the SelfMMOGA). For each solution, we consider a predefined number of its nearest neighbours (with respect to the Hamming distance). If the fitness of the solution is better than its neighbours fitness, it is denoted as a local optima and the centre of the basin. The number of neighbours is a tunable parameter.

The function $b(x,z)$ can be easily evaluated by defining if individual x is in a predefined radius of basin centre z. The radius is a tunable parameter. In this study, we define it as

$$radius = \frac{total\ population\ size}{k} \quad (2)$$

where k is the number of identified basins ($k = |Z|$).

The second measure is called Sum of Distances to Nearest Neighbour (*SDNN*). The *SDNN* penalizes the clustering of solutions. This indicator does not require knowledge of optima and basins. The *SDNN* can be calculated as

$$SDNN(pop) = \sum_{i=1}^{pop\ size} d_{nn}(x_i, pop) \quad (3)$$
$$d_{nn}(x_i, pop) = \min_{y \in pop \setminus \{x_i\}} \{dist(x_i, y)\}$$

where d_{nn} is the distance to the nearest neighbour, *dist* is the Hamming distance.

Finally, we combine the *BR* and the *SDNN* in an integrated criterion *K*:

$$K = \alpha \cdot BR(pop) + (1 - \alpha) \cdot \overline{SDNN}(pop) \quad (4)$$

where $\overline{SDNN}$ is a normalized value of $SDNN$, α defines weights of the BR and the *SDNN* in the sum ($\alpha \in [0,1]$).

At the coopearative stage, in many coevolutionary schemes, all individual algorithms begin each new adaptation period with the same starting points (such a migration scheme is called “the best displaces the worst”). For MMO problems, the best solutions are defined by discovered basins in the search space. As we already have evaluated the approximation of basins (*Z*), the solutions from *Z* are introduced in all populations replacing the most similar individuals.

4 EXPERIMENTAL RESULTS

To estimate the approach performance, we have used the following list of benchmark and real-world problems:

- Six binary MMO problems are from (Yu and Suganthan, 2010). These test functions are based on the unitation functions, and they are massively multimodal and deceptive.
- Eight real-valued MMO problems are from CEC’2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization (Li et al., 2013a).
- Fuzzy rule base classification system design using MMO GA.
- Designing loan portfolios for the Bank of Moscow.

4.1 Benchmark Problems

We have denoted the functions as in the source papers. Real-valued problems have been binarized using the standard binary encoding with 5 accuracy levels proposed in the CEC’13 competition rules. In all comparisons, all algorithms have equal maximum number of the objective evaluations, but may differ in population sizes.

The following criteria for estimating the performance of the SelfMMOGA over the benchmark problems are used for continuous problems:

- Peak Ratio (*PR*) measures the percentage of all optima found by the algorithm (5).
- Success Rate (*SR*) measures the percentage of successful runs (a successful run is defined as a run where all optima were found) out of all runs.

$$PR = \frac{|\{q \in Q \mid d_{nn}(q, pop) \le \varepsilon\}|}{k} \quad (5)$$

where $Q = \{q_1, q_2, \dots, q_k\}$ is a set of known optima, ε is accuracy level.

The maximum number of function evaluation and the accuracy level for the *PR* evaluation are the same as in CEC completion rules (Li et al., 2013a). The number of independent runs of the algorithm is 50.

In the case of binary problems, we cannot define the accuracy level in the *PR*, thus the exact points in the search space have to be found. This is a great challenge for search algorithms, thus we have substituted the *SR* measure with Peak Distance (*PD*). The *PD* indicator (6) calculates the average distance of known optima to the nearest individuals in the population (Preuss and Wessing, 2013).

$$PD = \frac{1}{k} \sum_{i=1}^{k} d_{nn}(q_i, pop) \quad (6)$$

The detailed results of estimating the performance of the SelfMMOGA with the pack of binary problems can be found in (Sopov, 2015b). We have compared the results with Ensemble of niching algorithms (ENA) proposed in (Yu and Suganthan, 2010). The experiments have shown that binary problems are not too complex for the SelfMMOGA and the ENA – there is no statistical significant difference in the results.

The results of estimating the performance of the SelfMMOGA with the pack of continuous problems are presented in Tables 1. Table 1 shows a comparison of results averaged over all problems with other techniques.

Table 1: Average PR and SR for each algorithm.

ε	SelfMMOGA		DE/nrand/1/bin		cDE/rand/1/bin		N-VMO		dADE/nrand/1		PNA-NSGAII	
	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR	PR	SR
1e-01	0.962	0.885	0.850	0.750	0.963	0.875	1.000	1.000	0.998	0.938	0.945	0.875
1e-02	0.953	0.845	0.848	0.750	0.929	0.810	1.000	1.000	0.993	0.828	0.910	0.750
1e-03	0.943	0.773	0.848	0.748	0.847	0.718	0.986	0.813	0.984	0.788	0.906	0.748
1e-04	0.907	0.737	0.846	0.750	0.729	0.623	0.946	0.750	0.972	0.740	0.896	0.745
1e-05	0.816	0.662	0.792	0.750	0.642	0.505	0.847	0.708	0.835	0.628	0.811	0.678
Average	0.916	0.780	0.837	0.750	0.822	0.706	0.956	0.854	0.956	0.784	0.893	0.759

We have compared the results of the SelfMMOGA runs with some efficient techniques from the competition. The techniques are DE/nrand/1/bin and Crowding DE/rand/1/bin (Li et al., 2013a), N-VMO (Molina et al., 2013), dADE/nrand/1 (Epitropakis et al., 2013), and PNA-NSGAII (Bandaru and Deb, 2013).

The settings for the SelfMMOGA are:

- Maximum number of function evaluation is 50000 (for cecF1-cecF5) and 200000 (for cecF6-cecF8);
- Total population size is 200;
- Adaptation period is 10 generations 25 times (for cecF1-cecF5) and 25 generations 40 times (cecF6-cecF8);
- All specific parameters of individual algorithms are self-tunable.

As we can see from Tables 1, the SelfMMOGA shows results comparable with popular and well-studied techniques. It yields to dADE/nrand/1 and N-VMO, but we should note that these algorithms are specially designed for continuous MMO problems, and have taken 2nd and 4th places (Li et al., 2013b), respectively, in the CEC competition. At the same time, the SelfMMOGA has very close average values to the best two algorithms, and outperforms PNA-NSGAII, CrowdingDE and DE, which have taken 7th, 8th and 9th places in the competition respectively.

In this study, we have included only basic MMO search techniques in the SelfMMOGA. Nevertheless, it performs well due to the effect of collective decision making in the ensemble. The key feature of the approach is that it operates in an automated, self-configuring way. Thus, the SelfMMOGA can be a good alternative for complex black-box MMO problems.

4.2 Real-world Problems

4.2.1 Designing Loan Portfolios for the Bank of Moscow

The problem of bank loan portfolio design is an optimization problem of maximizing the profit of the bank with some constraints on the amount of free liabilities, the amount of credit requested, periods of credits, credit interests and so on. Input data to the problem is a set of credit requests from loan borrowers. The bank portfolio is a subset of requests that are approved by the bank.

In this paper, the loan portfolio based on data presented by Krasnoyarsk department of the Bank of Moscow is discussed. The following profit model (optimization objective) is used (7):

$$\begin{gathered} \mathit{Profit(X)} = \sum_{j=1}^{N} k_j \cdot (1 + d_j \cdot t_j) \cdot x_j \rightarrow max \\ \mathit{Risk(X)} = \frac{1}{\sum_{j=1}^{N} x_j} \cdot \sum_{j=1}^{N} P_j \cdot x_j \leq \rho \\ \sum_{j=1}^{N} k_j \cdot x_j \leq F \\ X = (x_1, x_2, \dots, x_N), x_i \in \{0,1\} \end{gathered} \tag{7}$$

where F – the amount of free liabilities held by the Bank at a given time; N – the number of borrowers; k_j – the amount of credit requested by the j-th borrower $j=1,N$; t_j – the period for which the j-th borrower takes a loan; x_j – Boolean variable taking the value 1, if the k_j loan is issued, and 0 otherwise; d_j – interest (%) on j-th credit; P_j – probability of non-payment of loan and interest on the loan; ρ – limitation on the total riskiness of the loan portfolio.

As a candidate solution is binary vector, there is no need to encode it to chromosome. The fitness function is defined as the sum of the Profit and penalty functions for given constraints.

The initial information about credit requests and their characteristics are presented in Table 2.

The length of the chromosome is 50. The search space contains 2^{50} ($\approx 10^{15}$) different portfolios. The maximum number of the fitness evaluation is set to 10^{6} that is 10^{-9} % of the cardinality of the search space.

The results of the bank portfolio design (global and three local solutions) are presented in Table 3.

Table 2: Initial data for the loan portfolio design problem.

Request No.	Request amount	Loan rate (%)	Period	Riskiness
1	10 000 000	25	75	0.042
2	5 300 000	28	80	0.039
3	2400000	25	91	0.029
4	50 000 000	23	84	0.033
5	1 000 000	28	64	0.026
………				
48	9 000 000	27	86	0.024
49	22 000 000	29	91	0.016
50	350 000	27	69	0.026
Total sum of requests = 256 695 000				
The amount of free liabilities = 188 500 000				

Table 3: Results for the loan portfolio problem.

Solutions (the structure of the loan portfolio)	Profit of portfolio	Rest of free liabilities	Total riskiness
01111011111111110111101000111110111010000101010111	199734518.9	30000	0.0292
11011100111110110110011101110101111010100101011110	199691164	15000	0.0286
01011110101111110001110111011110110011111011001110	199668728.9	15000	0.028
00110011010001010110111000110110101111110101101111	199593407.3	10000	0.0276

As we can see from Table 3, solutions obtained with the SelfMMOGA have very close values of the profit, but have very different structures. Thus these portfolios can be used as alternative solutions or as additional information for the portfolio analysis.

The problem has been also solved using the brute-force search. The first best solution founded by the SelfMMOGA is the exact global solution to the problem.

4.2.2 Fuzzy Rule Base Classification System Design using MMO GA

Modern machine learning methods often use evolutionary computation techniques as a design tool, which is universal and can be applied for various structures. These evolutionary algorithms applied for machine learning problems are often called genetics-based machine learning algorithms. The fuzzy rule-based classification systems (FRBCSs) are effective approaches in machine learning, as they can provide easy-to-understand models for the end users (Ishibuchi, 2005).

Traditional GAs applied to the FRBCSs design have a tendency to converge to the best-found optimum losing population diversity. Such single best-found solution usually has very good accuracy, but may have a structure that is not convenient for human understanding and analysis. Thus there is a good idea to find many (or all) global and acceptable local optima which represent different solutions to the problem. In a case of the FRBCS, such optima, while saving comparable accuracy, may contain different rules in the rule base and/or different fuzzy term structures.

The number of rules in computational experiments was fixed and equal to 12. The FRBCS method, which have been implemented, is based on a simple rule base encoding into the GA chromosome. The chromosome contains fuzzy sets assigned to input variables in the premise part and class labels assigned to output variables in the conclusion part of each rule in the rule base. The number of fuzzy sets for granulation was fixed and equal to 5+1. Additional fuzzy term is the "Don't care" condition (corresponding input variable is ignored).

The fitness function includes two values: error on the training set and the complexity of the rule base. The complexity of the rule base was calculated as the ratio of number of non-empty fuzzy sets to the total number of possible fuzzy sets in the rule base. Including complexity of the rule base into the fitness function allows creating of simpler rule bases. The distance between two rule bases for the MMO GA was calculated as the number of different fuzzy sets for these rule bases. More detailed information can be found in (Sopov et al., 2015).

The computational experiments for the fuzzy classification were performed on 7 datasets from UCI and KEEL repositories (KEEL, 2015; ics.uci.edu, 2015).

Table 4: Classification Results for Test Sample.

Dataset	Standard GA	SelfMMOGA Solution 1	SelfMMOGA Solution 2	SelfMMOGA Solution 3
Australian	0.839	0.862	0.867	0.816
Banknote	0.947	0.892	0.867	0.862
Column 2c	0.773	0.789	0.768	0.751
Column 3c	0.668	0.741	0.674	0.619
Ionosphere	0.747	0.680	0.656	0.665
Liver	0.567	0.586	0.597	0.598
Seeds	0.874	0.793	0.691	0.621

The Table 4 contains the classification results for the test sample obtained with the standard GA and three best solutions obtained with the SelfMMOGA.

As we can see, for three datasets the standard GA allows finding most accurate solutions. However, the SelfMMOGA outperforms the standard GA on 4 datasets out of 7. Moreover, the best solution is not always the first one – for example, for datasets Australian and Liver, the best solution was second or even third. More detailed results and obtained rule bases description can be found in (Sopov et al., 2015).

Thus, using this method, several local optima have been found, and the researcher is able to select one of them. We suggest that the results can help the human experts in a field of the solving problem to obtain better (or may be very new) information about the problem features.

5 CONCLUSIONS

In this study, a selective hyper-heuristic for control of MMO GA ensemble (called SelfMMOGA) is proposed. It involves many different search strategies in the process of MMO problem solving and adaptively control their interactions.

The SelfMMOGA allows complex MMO problems to be dealt with, which are the black-box optimization problems (a priori information about the objective and its features are absents or cannot be introduces in the search process). We have included 6 basic MMO techniques in the SelfMMOGA realization to demonstrate that it performs well even with simple core algorithms. We have estimated the SelfMMOGA performance with a set of binary benchmark MMO problems and continuous benchmark MMO problems from CEC'2013 Special Session and Competition on Niching Methods for Multimodal Function Optimization. The proposed approach has demonstrated a performance comparable with other well-studied techniques. Experimental results show that the SelfMMOGA outperforms the average performance of its stand-alone algorithms. It means that it performs better on average than a randomly chosen technique. This feature is very important for complex black-box optimization, where the researcher has no possibility of defining a suitable search algorithm and of tuning its parameters.

We have also applied the SelfMMOGA for solving some real-world problems to demonstrate the effect of identifying many optima to the problem of design and decision making support.

In further works, we will investigate the SelfMMOGA using more advanced component techniques.

ACKNOWLEDGEMENTS

The research was supported by President of the Russian Federation grant (MK-3285.2015.9) and the Russian Foundation of Basic Research (contract №20 16-01-00767, dated 03.02.2016).

REFERENCES

Bandaru, S., Deb, K. (2013). A parameterless-niching-assisted bi-objective approach to multimodal optimization. *In Proc. 2013 IEEE Congress on Evolutionary Computation (CEC'13)*. pp. 95-102.

Bessaou, M., Petrowski, A., Siarry, P. (2000). Island Model Cooperating with Speciation for Multimodal Optimization. *Parallel Problem Solving from Nature PPSN VI, Lecture Notes in Computer Science, Volume 1917*. pp. 437-446.

Bianchi, L., Dorigo, M., Gambardella, L.M., Gutjahr, W.J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing, Volume 8, issue 2*. pp. 239–287.

Boussaida, I., Lepagnotb, J., Siarryb, P. (2013). A survey on optimization metaheuristics. *Information Sciences, Volume 237*. pp. 82–117.

Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Woodward, J.R. (2010). A Classification of Hyper-heuristic Approaches. *Handbook of Metaheuristics,* *Volume 146 of the series International Series in Operations Research & Management Science.* pp. 449-468.

Das, S., Maity, S., Qub, B.-Y., Suganthan, P.N. (2011). Real-parameter evolutionary multimodal optimization: a survey of the state-of-the art. *Swarm and Evolutionary Computation, 1.* pp. 71–88.

Deb, K., Saha, A. (2010). Finding Multiple Solutions for Multimodal Optimization Problems Using a Multi-Objective Evolutionary Approach. *In Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 2010. ACM, New York.* pp. 447-454.

Epitropakis, M.G., Li, X., Burke, E.K. (2013). A dynamic archive niching differential evolution algorithm for multimodal optimization. *In Proc. 2013 IEEE Congress on Evolutionary Computation (CEC'13).* pp. 79-86.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading. MA: Addison-Wesley.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

ics.uci.edu (2015). *UC Irvine Machine Learning Repository.* [online] Available at: http://archive.ics.uci.edu/ml/

Ishibuchi H. (2005). Hybridization of fuzzy GBML approaches for pattern classification problems. *In IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics, Volume 35, Issue 2.* pp. 359-365.

Kaklaukas, A. (2015). *Biometric and Intelligent Decision Making Support.* Intelligent Systems Reference Library, Vol. 81. 2015, XII. Springer-Verlag, Berlin.

KEEL (2015). *KEEL, Knowledge Extraction based on Evolutionary Learning.* [online] Available at: http://www.keel.es.

Li, B., Li. J., Tang, K., Yao, X. (2015). Many-Objective Evolutionary Algorithms: A Survey. *ACM Computing Surveys (CSUR), v.48 n.1.* pp. 1-35.

Li, X., Engelbrecht, A., Epitropakis, M.G. (2013a). Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization. *Evol. Comput. Mach. Learn. Group, RMIT University, Melbourne, VIC, Australia. Tech. Rep.*

Li, X., Engelbrecht, A., Epitropakis, M. (2013b). Results of the 2013 IEEE CEC Competition on Niching Methods for Multimodal Optimization. *Report presented at 2013 IEEE Congress on Evolutionary Computation Competition on: Niching Methods for Multimodal Optimization.*

Liu, Y., Ling, X., Shi, Zh., Lv, M., Fang. J., Zhang, L. (2011). A Survey on Particle Swarm Optimization Algorithms for Multimodal Function Optimization. *Journal of Software, Vol. 6, No. 12.* pp. 2449-2455.

Maashi M., Kendall, G., Özcan, E. (2015). Choice function based hyper-heuristics for multi-objective optimization. *Applied Soft Computing, Volume 28.* pp. 312–326.

Molina, D., Puris, A., Bello, R., Herrera, F. (2013). Variable mesh optimization for the 2013 CEC special session niching methods for multimodal optimization. *In Proc. 2013 IEEE Congress on Evolutionary Computation (CEC'13).* pp. 87-94.

Pillay, N. (2015). An Overview of Evolutionary Algorithms and Hyper Heuristics. *In 2015 IEEE Congress on Evolutionary Computation (IEEE CEC 2015), Sendai, Japan.* [online] Available at: http://www.cs.usm.maine.edu/~congdon/Conferences/CEC2015/Pillay.CEC2015.tutorial.pdf.

Preuss, M., Wessing, S. (2013). Measuring multimodal optimization solution sets with a view to multiobjective techniques. *EVOLVE – A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV. AISC, vol. 227, Springer, Heidelberg.* pp. 123–137.

Qu, B., Liang, J., Suganthan P.N., Chen, T. (2012). Ensemble of Clearing Differential Evolution for Multi-modal Optimization. *Advances in Swarm Intelligence Lecture Notes in Computer Science, Volume 7331.* pp. 350-357.

Ray, T., Liew K.M. (2002). A Swarm Metaphor for Multi-objective Design Optimization. *Engineering Optimization, 34.* pp. 141-153.

Ross, P. (2005). Hyper-Heuristics. *Search Methodologies.* pp. 529-556.

Sen, P., Yang, J.-B. (2012). *Multiple Criteria Decision Support in Engineering Design.* Springer Science & Business Media.

Singh, G., Deb, K. (2006). Comparison of multi-modal optimization algorithms based on evolutionary algorithms. *In Proceedings of the Genetic and Evolutionary Computation Conference, Seattle.* pp. 1305–1312.

Sopov, E., Stanovov, V., Semenkin, E. (2015). Multi-strategy Multimodal Genetic Algorithm for De-signing Fuzzy Rule Based Classifiers. *In Proceedings of 2015 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2015), Cape Town, South Africa.* pp.167-173.

Sopov, E. (2015a). A Self-configuring Metaheuristic for Control of Multi-Strategy Evolutionary Search. *ICSI-CCI 2015, Part III, LNCS 9142.* pp. 29-37.

Sopov, E. (2015b). Multi-strategy Genetic Algorithm for Multimodal Optimization. *In Proceedings of the 7th International Joint Conference on Computational Intelligence (IJCCI 2015) - Volume 1: ECTA, Portugal.* pp. 55-63.

Yu, E.L., Suganthan, P.N. (2010). Ensemble of niching algorithms. *Information Sciences, Vol. 180, No. 15.* pp. 2815-2833.