A Study on Evolutionary Multi-Objective Optimization with Fuzzy Approximation for Computational Expensive Problems

Alessandro G. Di Nuovo[1,2], Giuseppe Ascia[2], and Vincenzo Catania[2]

[1] Centre of Robotics and Neural Systems, Plymouth University, United Kingdom
[2] Dipartimento di Ingegneria Elettrica, Elettronica e Informatica, Universita´ degli Studi di Catania, Italy

Abstract. Recent progress in the development of Evolutionary Algorithms made them one of the most powerful and flexible optimization tools for dealing with Multi-Objective Optimization problems. Nowadays one challenge in applying MOEAs to real-world applications is that they usually need a large number of fitness evaluations before a satisfying result can be obtained. Several methods have been presented to tackle this problem and among these the use of approximate models within MOEA-based optimization methods proved to be beneficial whenever dealing with problems that need computationally expensive objective evaluations. In this paper we present a study on a general approach based on an inexpensive fuzzy function approximation strategy, that uses data collected during the evolution to build and refine an approximate model. When the model becomes reliable it is used to select only promising candidate solutions for real evaluation. Our approach is integrated with popular MOEAs and their performance are assessed by means of benchmark test problems. Numerical experiments, with a low budget of fitness evaluations, show improvement in efficiency while maintaining a good quality of solutions.

Keywords: Evolutionary Multi-objective Optimization, Expensive Optimization Problems, Fuzzy Function Approximation.

1 Introduction

Evolutionary algorithms (EAs) proved to be very powerful and flexible techniques for finding solutions to many real-world search and optimization problems. In fact they have been used in science and engineering as adaptive algorithms for solving practical problems and as computational models of natural evolutionary systems. In particular great effort was recently devoted to develop EAs to solve Multi-Objective Optimization (MOO) problems. Algorithms in this particular class of problems are named Multi-objective evolutionary algorithms (MOEAs) and they aim at finding a set of representative Pareto optimal solutions in a single run, see [16] for details and examples. Despite the great successes achieved, evolutionary algorithms have also encountered many challenges. For most evolutionary algorithms, a large number of fitness evaluations (performance calculations) are needed before a well acceptable solution can be found. In many real-world applications, fitness evaluation is not trivial. There are several situations in

C.A. Coello Coello et al. (Eds.): PPSN 2012, Part II, LNCS 7492, pp. 102–111, 2012.

which fitness evaluation becomes difficult and computationally efficient approximations of the fitness function have to be adopted. A detailed survey on proposals to speedup the evaluation of a single configuration can be found in [8]. The most popular models for fitness approximation are polynomials (often known as response surface methodology), the Kriging model, whereby Gaussian process models are parameterized by maximum likelihood estimation, most popular in the design and analysis of computer experiments, and artificial neural networks (ANNs). In particular in [8], it is stated that ANNs are recommended under the condition that a global model is targeted and that the dimension is high. The reason is that ANNs need a lower number of free parameters compared to polynomials or Gaussian models. As an example, in [6] an inverse neural network is used to map back from a desired point in the objective space (beyond the current Pareto surface) to an estimate of the decision parameters that would achieve it. The test function results presented look particularly promising, though fairly long runs (of 20,000 evaluations) are considered.

A multi-objective evolutionary algorithm, called ParEGO [9], was devised to obtain an efficient approximation of the Pareto-optimal set with a budget of a few hundred evaluations. The ParEGO algorithm is based on Kringing model, and it begins with solutions in a latin hypercube and updates a Gaussian process surrogate model of the search landscape after every function evaluation, which it uses to estimate the solution with the largest expected improvement. A recent work, [15], presents MOEA/D-EGO, that is based on Fuzzy clustering and Gaussian stochastic process modeling extends the ParEGO algorithm to generate many candidate solutions at the same time, in such a way it is possible to evaluate them all using parallel computing. In [12] an improved Archive-based Micro Genetic Algorithm (referred to as AMGA2) for constrained MOO is proposed. AMGA2 borrows and improves upon several concepts from existing MOEAs. Benchmarking and comparison demonstrate its improved search capability in the range between 5000 and 20000 function evaluations.

Recently, in [2] a MOEA with hierarchical fuzzy approximation was studied to speed-up the Design Space Exploration (DSE) of embedded computer architectures. The Evolutionary-Fuzzy methodology, named MOEA+FUZZY, exploits the knowledge of the embedded computer architecture with a hierarchical design of the fuzzy approximator system, this way, in comparisons with other MOEA for computational expensive optimization problems, like ParEGO, showed to save a great amount of time and also gives more accurate results.

In this work we present a study on a general implementation of the MOEA+FUZZY approach that can be applied to every optimization problem, using a general strategy to efficiently build a fuzzy function approximator. Details on MOEA+FUZZY approach are given in section 2. Our study aims to assess efficiency and performance of MOEAs combined with our +FUZZY approach when only a low budget of fitness evaluations is available. To this end we integrated proposed approach with popular MOEAs and tested four synthetic benchmarks. The setup of experiments is described in Section 3, while numerical results are presented in Section 4. Finally Section 5 gives our conclusion and directions for future work.

2 MOEA+FUZZY : Multi-Objective Evolutionary Optimization with Fuzzy Function Approximation

In this section we give a detailed presentation of our Evolutionary-Fuzzy strategy which has the ability to avoid the real evaluation of individuals that it foresees to be not good enough to belong to the Pareto-set and to give them fitness values according to a fast estimation of the objectives obtained by means of a Fuzzy System (FS). The main idea is that data collected for previously evaluated candidate solutions can be used during the evolution to build and refine an approximate model, and through it to avoid evaluating less promising candidate solutions. By doing so, expensive evaluations are only necessary for the most promising population members and the saving in computational costs is considerable.

The proposed approach could be informally described as follows: in a first phase the MOEA evolves normally; in the meanwhile the FS learns from real fitness function evaluations until it becomes expert and reliable. From this moment on the MOEA stops using the real function evaluation and uses the FS to estimate the objectives. From this moment on only if the estimated objective values are good enough to enter the Pareto-set will the associated individual be exactly evaluated. This avoids the insertion in the Pareto set of non-Pareto system individuals. It should be pointed out, however, that a “good” individual might be erroneously discharged due to the approximation and estimation error. At any rate, this could affect the overall quality of the solution found only after long runs as will be shown in Section 4. The reliability condition is essential in this flow. It assures that the approximator is reliable and that it can be used in place of the real function evaluation. To test reliability during the training phase the difference (approximation error) between the actual fitness function output and the predicted (approximated) fuzzy system output is evaluated. If this difference is below a user defined threshold and a minimum number of samples have been presented, the approximator is considered to be reliable. This strategy avoid to pre-set the number of samples needed by the approximator before the EA exploration starts, that is difficult when the objective function is not known.

The MOEA and the FS represent the main components of the proposed approach. Whereas the first one is used to select individuals to be explored, the second one is used to evaluate them. In our approach MOEAs could be chosen among ones presented in the literature, while the next subsection focus on fuzzy system generation strategy.

2.1 Strategy to Build a Fuzzy Approximation System during Evolutionary Process

The MOEA+Fuzzy approach uses a Fuzzy System (FS), which has been demonstrated to be a universal approximator [14]. In this work fuzzy systems are generated with a method that is based on the well-known Wang and Mendel method [13]. It consists of five steps:

- *Step 1* Divides the input and output space of the given numerical data into fuzzy regions;
- *Step 2* Generates fuzzy rules from the given data;

– *Step 3* Assigns a degree to each of the generated rules for the purpose of resolving conflicts among them (rule with higher degree wins);
– *Step 4* Creates a combined fuzzy rule base based on both the generated rules and, if there were any, linguistic rules previously provided by human experts;
– *Step 5* Determines a mapping from the input space to the output space based on the combined fuzzy rule base using a defuzzifying procedure.

The main advantages of this method are that allows to build rules step by step and that do not require a priori knowledge about function to be approximated. In addition from Step 1 to 5 it is evident that this method is simple and straightforward, in the sense that it is a one-pass build-up procedure that does not require time-consuming training. Our single-threaded implementation needs just about 10^{-2} seconds both to add a new rule and to perform an evaluation of an individual even a relatively big system, with thousands of fuzzy rules and tens of input variables, on an Intel Core i5 machine. In our implementation the output space could not be divided in Step 1, because we had no information about boundaries. For this reason we used Takagi-Sugeno fuzzy rules [11], in which each i-th rule has as consequent M real numbers s_{iz}, with $z \in [1, M]$, associated with all the M outputs. TS_j being the set of fuzzy sets associated with the variable x_j, the fuzzy rules R_i of the single fuzzy subsystem are defined as:

$$R_i : \text{if } x_1 \text{ is } S_{i1} \text{ and} \ldots \text{and } x_N \text{ is } S_{iN} \text{ then } y_{i1} = s_{i1}, \ldots, y_{im} = s_{iM}$$

where $S_{ij} \in TS_j$. Let α_{jk} be the degree of truth of the fuzzy set S_{jk} belonging to TS_j corresponding to the input value $\bar{x}_j$. If m_j is the index such that α_{jm_j} is the greatest of the α_{jk}, the rule R_i will contain the antecedent x_j is S_{jm_j}. After constructing the set of antecedents the consequent values y_{iz} equal to the values of the outputs are associated. The rule R_i is then assigned a degree equal to the product of the N highest degrees of truth associated with the fuzzy sets chosen S_{ij}. The rules generated in this way are "and" rules, i.e., rules in which the condition of the IF part must be met simultaneously in order for the result of the THEN part to occur. For the problem considered in this paper, i.e., generating fuzzy rules from numerical data, only "and" rules are required since the antecedents are different components of a single input vector. In this work fuzzy sets shape is Gaussian with normal distribution. Steps 2 to 4 are iterated with the MOEA: after every evaluation a fuzzy rule is created and inserted into the rule base, according to its degree in case of conflicts. More specifically, if the rule base already contains a rule with the same antecedents, the degrees associated with the existing rule are compared with that of the new rule and the one with the highest degree wins. In Step 5 the defuzzifying procedure to calculate the approximated output value $\hat{y}$ is the one suggested in [13]. According to this method the defuzzified output is determined as follows

$$\hat{y}_j = \frac{\sum_{r=1}^{K} m_r \bar{y}_{rz}}{\sum_{r=1}^{K} m_r} \tag{1}$$

where K is the number of rules in the fuzzy rule base, $\bar{y}_{rz}$ is the output estimated by the r-th rule for the z-th output and m_r is the degree of truth of the r-th rule. In our implementation the defuzzifying procedure and the shape of the fuzzy sets were chosen a priori. This choice proved to be effective as well as a more intelligent implementation, which could embed a selection procedure to choose the best defuzzifying function and shape to use online. The advantage of our implementation is a lesser computational requirement of the algorithm and a faster evaluation.

3 Experimental Setup

For implementation of the MOEA+Fuzzy strategy described above we used the PISA suite [3]. PISA stands for *Platform and programming language Independent interface for Search Algorithms* and allows to implement application-specific parts (representation, variation, objective function calculation) separately from the actual search strategy (fitness assignment, selection). Several multi-objective evolutionary algorithms as well as other well-known benchmark problems, such the widely used set of continuous test functions the ZTL [17] and DTLZ [5], are available for download at the PISA website [1]. Due to space restrictions, not all results can be presented here. Instead, we will focus on four test problems ZDT3, ZDT6, DTLZ3 and DTLZ6, that summarize main issues encountered in our tests. Table 1 lists the synthetic test problems chosen for this study. Test problem ZDT3 was selected because it is discontinuous, while DTLZ3 because it is multi-modal and difficult to solve. ZDT6 and DTLZ6 were selected because they involve a highly skewed distribution of points in the search space corresponding to a uniform distribution of points in the objective space, thus challenge an optimization algorithm's ability to find the global Pareto-optimal frontier. Detailed description of test problems can be found in their respective references. Using problems presented above we tested the proposed methodology integrating it with the 2 most popular MOEAs, SPEA2 [18] and NSGA-II [4], and a novel version of ϵ-constraint evolutionary algorithm ECEA [10]. In this work we tested two different set-ups of our approach in order to assess it after different ranges of real function evaluations:

1. +FUZZY$_1$. Fuzzy system has 9 sets for each input variable and reliability thresholds are distance of 1.0 and maximum of 5000 evaluations. The minimum number of evaluations is 200.
2. +FUZZY$_2$. Fuzzy system has 25 sets for each input variable and reliability thresholds are distance of 0.5 and maximum of 10000 evaluations. The minimum number of evaluations is 1000.

The first set-up is intended for a very low budget of real evaluations, from some hundreds to few thousand, while the second should perform better with longer runs.

Table 1. Test problems

Name	Variables	Objectives	Remarks
ZDT3	10	2	Discontinuous
ZDT6	10	2	Skewed
DTLZ3	12	3	Multi-modal
DTLZ6	12	3	Skewed

The performance measure we considered is the *Hypervolume* [19], that is the only one widely accepted and, thus, used in many recent similar works. This index measures the hypervolume of that portion of the objective space that is weakly dominated by the Pareto set to be evaluated. In order to measure this index the objective space must be bounded, then a bounding reference point that is (at least weakly) dominated by all points should be used. In this work we applied a standard linear normalization procedure, i.e. all values are normalized to the minimum and maximum value observed on the test problem. We took as bounding point vectors [1.1,1.1] and [1.1,1.1,1.1] for two and three objectives, respectively.

To present an uniform comparison between different problems, in Section 4 we show the percentage of a reference hypervolume covered by algorithms under investigation.

The reference hypervolume is calculated from a reference Pareto-set, that was obtained in following way: first, we combined all approximations sets generated by the algorithms under consideration after 50000 function evaluations (i.e. 500 generations with a population of 100 individuals), and then the dominated objective vectors are removed from this union. At last, the remaining points, which are not dominated by any of the approximations sets, form the reference set. The advantage of this approach is that the reference set weakly dominates all approximation sets under consideration.

Identical setting is used for all the algorithms. The parameter settings used for each algorithm are as follows: number of generations = 500; population size = 100; number of parents = 100; number of offsprings = 100; individual mutation probability = 1.0; individual recombination probability = 1.0; variable mutation probability = 1.0; variable swap probability = 0.5; variable recombination probability = 1.0; mutation distribution index = 20.0; recombination distribution index = 15.0.

4 Numerical Results

Using the experimental setup described in section 3, for each test problem, algorithms ran twenty times with different random seed. Median values for performance indicators are presented to represent the expected (mid-range) performance. For the analysis of multiple runs, we compute the quality measures of each individual run, and report the median and the standard deviation of these. Since the distribution of the algorithms we compare are not necessarily normal, we use the Kruskal-Wallis test [7] to indicate if there is a statistically significant difference between distributions. We recall that the significance level of a test is the maximum probability α, assuming the null hypothesis, which the statistic will be observed, i.e. the null hypothesis will be rejected in error when it is true. The lower the significance level the stronger the evidence. In this work we assume that the null hypothesis is rejected if $\alpha < 0.01$.

Table 2 presents median number of real function evaluations for MOEA+FUZZYs. As expected for ZDT problems MOEAs need less function evaluations to converge, for this reason we chose different pre-fixed amounts of real function evaluations for comparison reported in Table 3. To make an uniform comparison we calculated median values of +FUZZY algorithms taking into account all runs, even those with a number of function evaluations lower than maximum threshold selected. This means that absolute performance of MOEA+FUZZYs is sometimes slightly underestimated.

Table 2. Real function evaluations of MOEA+Fuzzy after 500 generations with a population of 100 individuals

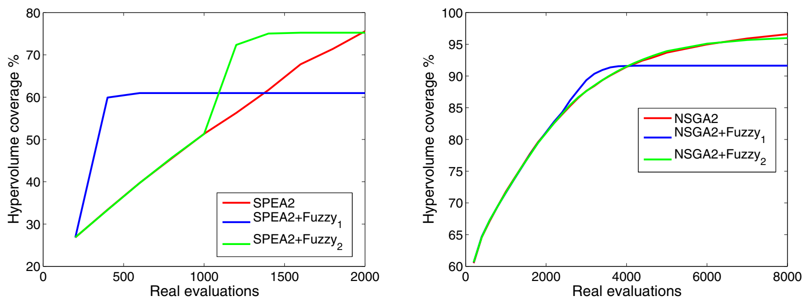
Algorithm	Test Problem / Real function evaluations of MOEA+Fuzzy							
	ZDT3		ZDT6		DTLZ3		DTLZ6	
	median	stddev	median	stddev	median	stddev	median	stddev
SPEA2+FUZZY$_1$	3095	3710	374	72	5354	60	3480	378
SPEA2+FUZZY$_2$	2246	509	1184	119	10571	112	10001	2107
NSGA-II+FUZZY$_1$	2387	4305	338	126	5256	75	3661	339
NSGA-II+FUZZY$_2$	2251	418	1218	131	10594	110	8549	1871
ECEA+FUZZY$_1$	297	38	298	32	10001	0	684	77
ECEA+FUZZY$_2$	1268	155	1294	203	10001	2	1969	2645

Table 3. Comparison of median hypervolume percentage covered after a fixed amount of real function evaluations

Algorithm	Test problem / Maximum number of real function evaluations*											
	ZDT3			ZDT6			DTLZ3			DTLZ6		
	400	1200	2000	400	1200	2000	2000	3000	5000	2000	3000	5000
SPEA2	81.26	95.90	99.30	32.80	58.61	77.43	99.38	99.52	99.64	81.06	87.65	93.68
+FUZZY$_1$	**94.27**	95.42	95.59	**58.08**	59.04	59.04	99.35	99.48	99.65	81.14	**89.34**	91.63
+FUZZY$_2$	81.25	**97.75**	99.00	34.19	**72.40**	74.61	99.35	99.48	99.64	81.00	87.61	93.91
NSGA-II	81.57	95.85	99.34	33.38	56.28	75.58	99.47	99.58	99.70	83.62	90.34	95.55
+FUZZY$_1$	**94.01**	96.56	96.65	**59.91**	60.96	60.96	99.46	99.57	99.71	83.91	**92.11**	93.73
+FUZZY$_2$	81.56	**97.66**	99.04	33.32	**72.36**	75.23	99.46	99.57	99.70	83.91	90.43	95.99
ECEA	74.48	76.16	76.39	28.87	32.94	33.59	97.36	97.40	97.53	62.48	62.72	68.24
+FUZZY$_1$	76.25	76.25	76.25	31.17	31.17	31.17	97.35	97.39	97.51	63.09	63.09	63.09
+FUZZY$_2$	74.85	76.25	77.47	28.53	33.08	33.90	97.35	97.39	97.54	63.04	63.11	63.15

* Median values of +FUZZY algorithms are reported taking into account also runs with a number of function evaluations lower than the threshold used for the comparison.
Results **in bold** are better than others with statistical significance level $\alpha < 0.01$, according to the Kruskal-Wallis test.

In particular, looking at Table 2 we remark that none of +FUZZY algorithms reached 2000 real evaluations in ZDT6 problem. From the results of the benchmark study, we can see that the SPEA2+FUZZYs and NSGA-II+FUZZYs perform comparably well in ranges considered, while ECEA has a slower convergence that impacts performance of ECEA+FUZZYs. Results in Table 3 show that fuzzy system in scenario +FUZZY$_1$ is able to speedup the convergence of MOEAs after a very low number of real fitness evaluations for ZDT problems. On the other hand +FUZZY$_2$ improvement is smaller, but its performance on longer runs is more reliable, as shown also in Table 4. Figure 1 shows two examples of hypervolume improvement during the evoluationary process. Speedup of +FUZZYs approach is evident in Figure 1(a). In DTLZ3 there is no significant improvement thanks to fuzzy approximation strategy, while in DTLZ6 our fuzzy approach help to improve only SPEA2 and NSGA-II evolution in the range between 2000 and 3000 as also shown in Figure 1(b).



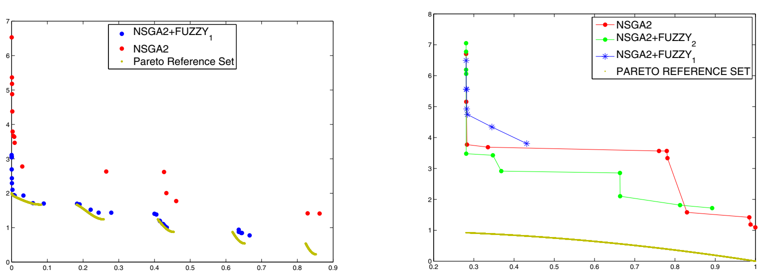
(a) (b)

Fig. 1. Hypervolume coverage: (a) ZDT6 - SPEA2 ; (b) DTLZ6 - NSGA-II

Table 4. Comparison of hypervolume percentage covered after 500 generations

Algorithm	Test Problem			
	ZDT3	ZDT6	DTLZ3	DTLZ6
SPEA2	**100**	**100**	**99.95**	**99.85**
SPEA2+FUZZY$_1$	96.67	60.96	99.72	93.74
SPEA2+FUZZY$_2$	99.15	75.23	99.82	97.98
NSGA-II	**100**	**100**	**99.96**	**99.85**
NSGA-II+FUZZY$_1$	95.68	59.04	99.65	91.63
NSGA-II+FUZZY$_2$	99.11	74.61	99.81	96.39
ECEA	**87.63**	**39.61**	98.22	**94.59**
ECEA+FUZZY$_1$	76.25	31.17	97.98	63.09
ECEA+FUZZY$_2$	77.47	33.90	97.99	63.32

Results **in bold** are better than others with statistical significance level $\alpha < 0.01$, according to the Kruskal-Wallis test.



(a) (b)

Fig. 2. Pareto set comparison: (a) ZDT3 after 400 real function evaluation ; (b) ZDT6 after 1200 real function evaluations

As expected +FUZZY$_1$ is more efficient that +FUZZY$_2$, but the second setup is able to achieve a better quality of solutions. Table 4 present results after 500 generations. MOEA+FUZZY solutions maintain a good quality even if they are not able to find the Pareto optimal set. ZDT6 is the only problem in which, despite the great efficiency improvement, +FUZZYs are not able to find a good approximation of the Pareto optimal set, this is because MOEAs converge slowly. However this problem could be solved using a higher minimum threshold for the fuzzy system learning strategy. Figure 2 presents two comparison: in (a) Pareto sets for ZDT3 problem are shown, demonstrating that NSGA-II+FUZZY$_1$ (as well as NSGA-II+FUZZY$_1$) is able to obtain a very good approximation of the Pareto optimal set after 400 real function evaluation; Pareto sets for ZDT6 problem are drawn in (b), where it is shown NSGA+FUZZY$_2$ outperform NSGA-II+FUZZY$_1$ and classic NSGA-II, but they are still quite far from the Pareto optimal set.

5 Conclusion and Future Work

In this paper we have presented an empirical study on use of fuzzy function approximation to speed up evolutionary multi-objective optimization. The methodology uses a MOEA for heuristic exploration of the search space and a fuzzy system to evaluate the candidate system individuals to be visited. Our methodology works in two phases: firstly all individuals are evaluated using computationally expensive evaluations and their results are used to train the fuzzy system until it becomes reliable; in the second phase the system is used to estimate fitness of all individuals and only promising individuals are actually evaluated to improve the accuracy of the fuzzy system.

Empirical results with low budgets of real evaluations (i.e. from hundreds to three thousand) encourage the use of a fuzzy system as approximate model to improve efficiency of MOEAs. This is because to the strategy used to build the fuzzy system, that allows generating an efficient fitness function approximator without any previous learning phase and knowledge of real function. Strengths of our approach are the inexpensive learning procedure, that could be easily integrated with every MOEA giving the opportunity to take advantage of novel algorithms, and the possibility to set-up the fuzzy system according to a maximum number of real function evaluations.

On the other hand fuzzy systems have the characteristic to allow to embed prior knowledge about the function to be approximated, this could be useful in problems where there is an expert that knows at least part of the behaviours of the objective function to be evaluated. This will be matter of our future work along with a study on improvement of fuzzy system learning and evaluation strategies in order to maximize approximation performance and, thus, tackle the problem of loss of solution quality in longer runs.

References

1. `http://www.tik.ee.ethz.ch/pisa/`
2. Ascia, G., Catania, V., Di Nuovo, A., Palesi, M., Patti, D.: Performance evaluation of efficient multi-objective evolutionary algorithms for design space exploration of embedded computer systems. Applied Soft Computing 11(1), 382–398 (2011)

3. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – A Platform and Programming Language Independent Interface for Search Algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)
4. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN VI. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
5. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, R., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, ch. 6, pp. 105–145. Springer (2005)
6. Gaspar-Cunha, A., Vieira, A.S.: A hybrid multi-objective evolutionary algorithm using an inverse neural network. In: Hybrid Metaheuristics, Workshop at ECAI 2004 (August 2004)
7. Gibbons, J.D.: Nonparametric Statistical Inference, 2nd edn. M. Dekker (1985)
8. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing Journal 9, 3–12 (2005)
9. Knowles, J.: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 10(1), 50–66 (2006)
10. Laumanns, M., Thiele, L., Zitzler, E.: An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. European Journal of Operational Research 169(3), 932–942 (2006)
11. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on System, Man and Cybernetics 15, 116–132 (1985)
12. Tiwari, S., Fadel, G., Deb, K.: Amga2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. Engineering Optimization 43(4), 377–401 (2011)
13. Wang, L.X., Mendel, J.M.: Generating fuzzy rules by learning from examples. IEEE Transactions on System, Man and Cybernetics 22, 1414–1427 (1992)
14. Zeng, K., Zhang, N.Y., Xu, W.L.: A comparative study on sufficient conditions for takagi-sugeno fuzzy systems as universal approximators. IEEE Transactions on Fuzzy Systems 8(6), 773–778 (2000)
15. Zhang, Q., Liu, W., Tsang, E.P.K., Virginas, B.: Expensive multiobjective optimization by moea/d with gaussian process model. IEEE Trans. Evolutionary Computation 14(3), 456–474 (2010)
16. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1(1), 32–49 (2011)
17. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2), 173–195 (2000)
18. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Proceedings of the EUROGEN 2001 Conference, Athens, Greece, September 19-21, pp. 95–100 (2002)
19. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions Evolutionary Computation 7(2), 117–132 (2003)