Bayesian Optimization Algorithms for Multi-Objective Optimization

Marco Laumanns[1] and Jiri Ocenasek[2]

[1] ETH Zürich, Computer Engineering and Networks Laboratory, CH–8092 Zürich
laumanns@tik.ee.ethz.ch
http://www.tik.ee.ethz.ch/aroma
[2] VUT Brno, Faculty of Information Technology, Bozetechova 2, CZ - 612 66 Brno
ocenasek@fit.vutbr.cz
http://www.fit.vutbr.cz/~ocenasek

Abstract. In recent years, several researchers have concentrated on using probabilistic models in evolutionary algorithms. These Estimation Distribution Algorithms (EDA) incorporate methods for automated learning of correlations between variables of the encoded solutions. The process of sampling new individuals from a probabilistic model respects these mutual dependencies such that disruption of important building blocks is avoided, in comparison with classical recombination operators. The goal of this paper is to investigate the usefulness of this concept in multi-objective optimization, where the aim is to approximate the set of Pareto-optimal solutions. We integrate the model building and sampling techniques of a special EDA called Bayesian Optimization Algorithm, based on binary decision trees, into an evolutionary multi-objective optimizer using a special selection scheme. The behavior of the resulting Bayesian Multi-objective Optimization Algorithm (BMOA) is empirically investigated on the multi-objective knapsack problem.

1 Introduction

The Estimation of Distribution Algorithms (EDAs) [5, 8] also called probabilistic model-building evolutionary algorithms have attracted a growing interest during the last few years. Recombination and mutation operators used in standard EAs are replaced by probability estimation and sampling techniques to avoid the necessity to specify certain EA parameters, to avoid the disruption of building blocks and to enable solving of non-linear or even deceptive problems having considerable degree of epistasis.

In multi-objective optimization the usual goal is to find or to approximate the set of Pareto-optimal solutions. Research in the design of multi-objective evolutionary algorithms has mainly focused on the fitness assignment and selection part. In contrast, the variation operators could be used from the single objective case without modification, which gave them only little attention. Some studies indicate, however, that the existence of multiple objectives influences the success probabilities of mutations which in turn has consequences for the choice of

the mutation strength [7, 3]. For recombination it is unclear whether combining parents that are good in different objectives improve the search as they could create good compromise offspring [2], or whether they contain such incompatible features that a combination does not make sense, thus advocating mating restrictions. The fact that recombination generally is a "contracting" operator might also conflict with the goal to reach a broad distribution of Pareto-optimal solutions.

In this study we investigate the use of EDAs for multi-objective optimization problems to overcome the aforementioned difficulties when creating offspring from a set of diverse parents from different trade-off regions. The next section introduces the Bayesian Optimization Algorithm (BOA) as a special EDA based on the Bayesian network model. It summarizes disadvantages of Bayesian network and introduces BOAs based on decision trees. We derive our own incremental equation for construction of decision trees and demonstrate the algorithm for decision tree construction. In section 3 a new Multi-objective BOA is proposed. We derive design guidelines for a useful selection scheme in connection with the construction of the probabilistic model and develop a new operator based on ϵ-archives [4]. Section 4 demonstrates the applicability of the approach using the multi-objective knapsack problem and compares the results to other multi-objective evolutionary algorithms.

2 Bayesian Optimization Algorithm (BOA)

One of the most general probabilistic models for discrete variables used in EDAs is the Bayesian network (BN). It is able to encode any dependencies of variables that can obtain one out of a finite set of values.

2.1 Structure of Bayesian Network

A Bayesian network for the domain of possible chromosomes $X = (X_0, \ldots, X_{n-1})$ represents a joint probability over X. The BN representation consists of 2 parts, a set of conditional independence assertions and a set of local conditional distributions, that allow us to construct a global joint probability distribution of chromosome from the local distributions of genes.

The first part, the set of conditional independence assertions, is expressed by a dependency graph, where each gene corresponds to one node in the graph. If the probability of the value of a certain gene X_i is affected by value of other gene X_j, then we say that "X_i depends on X_j" or "X_j is a parent variable of X_i". This assertion is expressed by existence of edge (j, i) in the dependency graph. A set of all parent variables of X_i is denoted Π_i, it corresponds to the set of all starting nodes of edges ending in X_i.

In the example of Fig. 1 (left), genes X_0, X_2 are independent and the value of X_1 is affected by X_0 and X_2. Under this assertion we can write the probability of whole chromosome (X_0, X_1, X_2) as the product of local distributions:

$$p(X_0, X_1, X_2) = p(X_0) \cdot p(X_2) \cdot p(X_1|X_0, X_2) \tag{1}$$

Fig. 1. Example of dependency graph for $n = 3$ (left) and decision tree (right)

Now we will focus on the second part of BN representation – the set of local distributions. For simplicity let's consider binary genes. In the case of gene X_0 resp. X_2 from our example this local distribution is unconditional and can be estimated from the population by simple counting individuals where $X_0 = 1$ resp. $X_2 = 1$ and dividing it by the size of population N. In the case of gene X_1 the gene depends on X_0 and X_2, so the local distribution is conditional and can be expressed by the following table:

$$\begin{array}{r|l} X_0, X_2 & 00\ 01\ 10\ 11 \\ \hline p(X_1 = 1|X_0, X_2) & \ldots\ \ldots\ \ldots\ \ldots \end{array} \tag{2}$$

The dots in the second row denote values estimated from the population by

$$p(X_1 = 1|X_0, X_2) = m(X_1 = 1, X_0, X_2)/m(X_0, X_2).$$

With this Bayesian network we are able to determine the probability of each concrete chromosome $X = (x_0, x_1, x_2)$.

The most important and also most time-consuming part of EDA is the algorithm for construction of the probabilistic model from the population. Most methods for automated learning of probabilistic models have been adopted from the area of data mining. When a significant building block is detected among the solutions in the population, the information about dependency of its genes is added to the model.

The original BOA algorithm uses a hillclimbing algorithm to step-by-step improve the Bayesian network. It starts with independent genes (no edges are present between nodes of dependency graph), such that the local probabilities are unconditional. Then in each step the algorithm examines all possible edges and it adds the "best edge" to the network. By the term "best edge" we mean the edge which does not introduce any cycle in the dependency graph and which improves the score most. The quality of each edge is expressed by the Bayes-Dirichlet metric (BDe, see [1]). This equation measures the bias in the probability distribution of combinations of genes in the population. For further details on various types of EDAs see the exhaustive survey [5].

2.2 Binary Decision Trees Based BOA

The problem with the BN approach is that after introducing one more parent variable of X_i, the number of columns of the conditional distribution table of

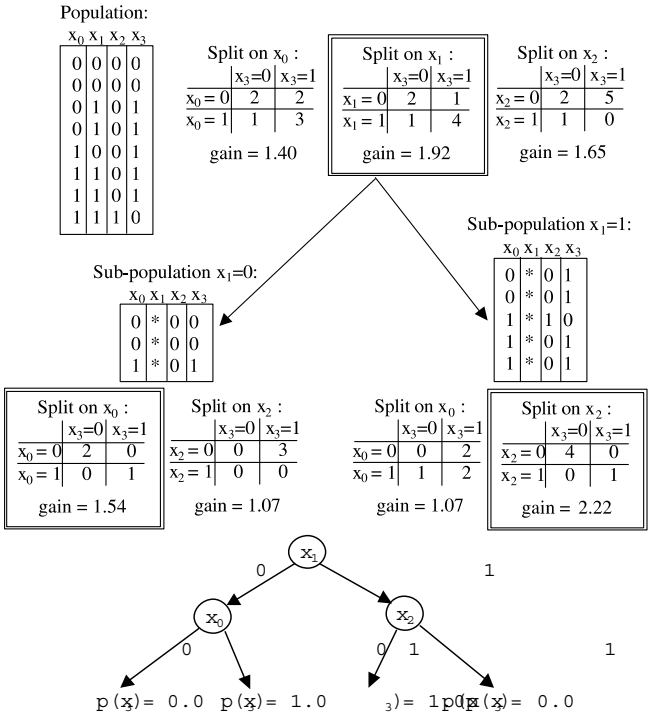

Fig. 2. Example of construction of the decision tree for variable $X_3, n = 4$, and final decision tree for variable X_3

X_i doubles, making the computation time of the BDe metric exponentially increasing with the number of parents. In previous versions of BOA the number of possible parents was limited to k, making the BDe computable in real time. A better divide-and-conquer approach is based on binary decision trees, firstly proposed for EDAs in [6].

The model is composed of the set of trees, one tree is for each variable. The dependence assertions are expressed by existence of splitting nodes and the local probabilities are stated in leaf nodes. A set of all parent variables of X_i denoted Π_i corresponds to the set of all decision nodes of i-th tree.

Decision trees are a more accurate model than Bayesian network. They allow us to describe the dependencies of alleles (gene values). Let us consider our first example from Fig. 1. Assume that if $X_0 = 1$, then the value of X_1 does not depend on X_2, but when $X_0 = 0$, then the value of X_1 depends on both X_0 and X_2. Because $p(X_1|10) = p(X_1|11) = p(X_1|1*)$, this would reduce the number of table columns in (2), as can be seen in the following table:

$$\begin{array}{r|l} X_0, X_2 & 00\ 01\ 1* \\ \hline p(X_1 = 1|X_0, X_2) & \ldots\ \ldots\ \ldots \end{array} \tag{3}$$

This situation can be expressed by a decision tree (see Fig. 1, right). Each variable which determines the X_1 value corresponds to one or more split nodes in the tree, each leaf determines $p(X_1)$ among the individuals fulfilling the split conditions on the path from the root.

A further advantage of decision trees lies in the low complexity of their building. The step of adding a new split node is easy to evaluate by the metric – it splits only one column in the local distribution table. From the Bayes-Dirichlet metrics (BDe) we derived the incremental equation for adding one new binary split:

$$Gain(X_i, X_j) = \frac{\sum_{r\in\{0,1\}} \sum_{s\in\{0,1\}} \Gamma(m_{r,s}+1) \cdot \Gamma(\sum_{r\in\{0,1\}} \sum_{s\in\{0,1\}} (m_{r,s}+1))}{\sum_{r\in\{0,1\}} \Gamma(\sum_{s\in\{0,1\}} (m_{r,s}+1)) \cdot \sum_{r\in\{0,1\}} \Gamma(\sum_{s\in\{0,1\}} (m_{r,s}+1))} \quad (4)$$

where X_i is the variable for which the tree is being constructed, X_j is the parent variable – possible split, and $m_{r,s}$ is the number of individuals having $X_i = r$ and $X_j = s$. Note that the splitting is done recursively, so $m_{r,s}$ is determined only from the subpopulation being splitted. We often use the logarithm of this metric, which avoids multiplication operations. Another method for construction of decision trees straight from the BDe metric can be found in [6]. Additionally this paper proposed a leaf-merge operator to obtain decision graphs instead of decision trees.

As result of model construction a set of decision trees is obtained, see Fig. 2. The dependencies are acyclic, so there exists a "topological" ordering of genes $o_0, o_1, \ldots, o_{n-1}$ such that parents Π_i are from the set $\{o_0, o_1, \ldots, o_{i-1}\}$. When generating a new binary chromosome, the independent gene X_{o_0} is generated first, by "flipping the coin" according to its single leaf node. Then, other genes are generated in the $o_1, \ldots, o_{n-1}$ order. The generation of X_i is driven by actual values of parent variables Π_i, the decision tree traversal ends up in one of the leaf nodes which describe the probability $p(X_i = 1)$.

3 Multi-objective BOA

For the design of a multi-objective BOA some important aspects have to be taken into account, some due to the existence of multiple objective, others from the necessity of the probabilistic model building techniques. Preliminary tests with a simple $(\mu + \lambda)$-strategy and fitness assignment based on the dominance grade have shown that a trivial multi-objective extension leads to poor performance. The population is likely to converge to an "easy to find" region of the Pareto set, as already noticed by [9], and duplicate solutions are produced repeatedly. The resulting loss of diversity leads to an insufficient approximation of the Pareto set and is especially harmful for building a useful probabilistic model. Therefore the following design requirements are essential:

1. Elitism (to preclude the problem of gradual worsening and enable convergence to the Pareto set)

Algorithm 1 $Select(A, P, \mu, \epsilon)$

Input: old parent set A, candidate set P, minimum size μ, approximation factor ϵ
Output: new parent set A'
for all $x \in P$ **do**
  $B := \{y \in A | \lfloor \frac{\log f_i(y)}{\log \epsilon} \rfloor = \lfloor \frac{\log f_i(x)}{\log \epsilon} \rfloor \quad \forall \text{ objective functions } i\}$
  if $B = \emptyset$ **then**
    $A := A \cup \{x\}$
  else if $\exists y \in B$ such that $x \succ y$ **then**
    $A := A \setminus B \cup \{x\}$
  end if
end for
$A' := \{y \in A | \nexists z \in A : z \succ y\}$
$D := A \setminus A'$
if $|A'| < \mu$ **then**
  Fill A' with $\mu - |A'|$ individuals $y \in D$ in increasing order of $|\{z \in A' \cup D | z \succ y\}|$
end if
Return: A'

Algorithm 2 $(\mu + \lambda, \epsilon)$-BMOA

$A := \emptyset$
while $|A| < \mu$ **do**
  Randomly create an individual x.
  $A := Select(A, \{x\}, \mu, \epsilon)$
end while
while Termination criteria not fulfilled **do**
  Create Bayesian Model M from A.
  Sample λ new individuals from M.
  Mutate these individuals and put them into B.
  $A := Select(A, B, \mu, \epsilon)$
end while

2. Diversity maintenance in objective space (to enable a good approximation of the whole Pareto set)
3. Diversity maintenance in decision space (to avoid redundancy and provide enough information to build a useful probabilistic model)

3.1 A New Selection Operator

From the existing selection/archiving operators in evolutionary multi-objective optimization, the ϵ-Archive [4] has been designed to meet the requirements 1 and 2 above. This method maintains a minimal set of solutions that ϵ-dominates all other solutions generated so far. However, as this set can become very small, the scheme has to be modified to provide enough decision space diversity. The new selection operator is described in Alg. 1. The idea is that now also dominated individuals are allowed to survive, depending on the number of individuals they are dominated by.

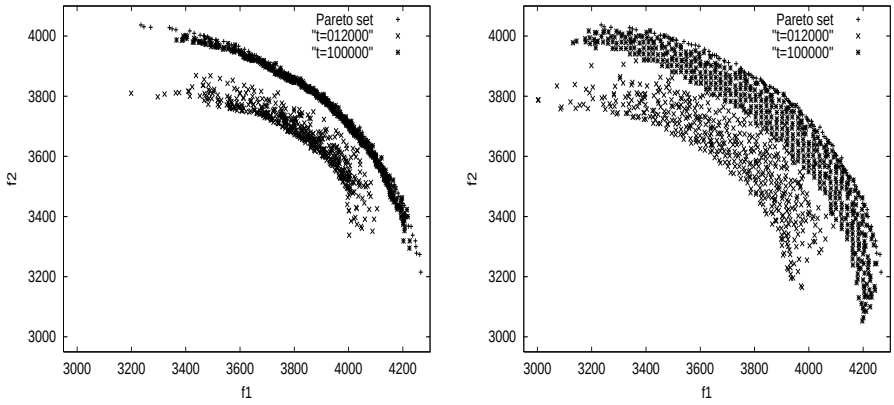

Fig. 3. Development of the population of $(500+500, \epsilon)$-BMOA on the KP-100-2 for $\epsilon = 10^{-6}$ (left) and $\epsilon = 0.005$ (right) after $t = 12000$ and $t = 100000$ function evaluations

3.2 The $(\mu + \lambda, \epsilon)$-BMOA

The combination of the selection operator (Alg. 1) and the variation based on the probabilistic model described in Section 2.2 results in a Bayesian Multi-objective Optimization Algorithm described in Alg. 2. In this $(\mu+\lambda, \epsilon)$-BMOA, μ denotes the (minimum) number of parents that survive to the next generation being the input to build the model, λ the number of samples from the model in one generation and ϵ the factor that determines the granularity of the approximation. As the Bayesian model M we used the set of decision trees described in section 2.2.

4 Experimental Results

In recent research activities in the field of multi-objective meta-heuristics the 0/1 knapsack problem has become a standard benchmark. Results of several comparative case studies are available in the literature, accompanied by test data through the Internet. The problem can be stated as KP-n-m:

$$\begin{array}{rl} \text{Maximize} & f_j(x) = \sum_{i=1}^{n} x_i \cdot p_{i,j} \\ \text{s.t.} & g_j(x) = \sum_{i=1}^{n} x_i \cdot w_{i,j} \leq W_j \\ & x_i \in \{0,1\}, 1 \leq i \leq n, 1 \leq j \leq m \end{array} \tag{5}$$

where $p_{i,j}$ and $w_{i,j}$ are the elements of the profit and the weight matrices, respectively, and W_j the j-th weight constraint. n denotes the number of binary decision variables and m the number of objectives and constraints. The representation of a solution as a bit string chromosome of length n is straightforward. Infeasible solutions are decoded using a greedy repair mechanism for the calculation of the objective values without changing the genotype of the individuals. The problem is NP-hard, and the exact Pareto set can only be computed for small instances.

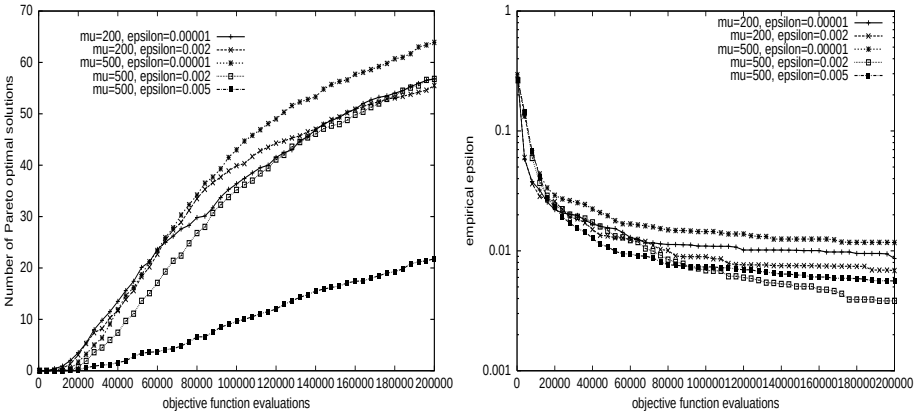

Fig. 4. Average number of Pareto-optimal solution contained in the population (left) and empirical approximation quality ϵ_{min} (right) for KP-100-2 for different μ and ϵ values

4.1 Results of $(\mu + \lambda, \epsilon)$-BMOA

In this section we report results of the BMOA on the knapsack problem (5) with $m = 2$ objectives to demonstrate the applicability of the approach. Each individual sampled from the model is additionally mutated by flipping each bit independently with probability $1/n$. Together with the results from [4] this guarantees that in the limit the non-dominated population members converge to a representative subset of the Pareto set of (5) in the sense that each Pareto-optimal point is ϵ-dominated by at least one population member.

Fig. 3 shows the development of the population for different ϵ values for a typical run. With a small ϵ, it can be seen that the population is more concentrated near the middle of the Pareto set compared to the larger ϵ value, where the population is distributed more evenly and broadly. Fig. 4 displays the number of Pareto-optimal solutions found and the empirical approximation quality ϵ_{min}[3] over time. It indicates how the algorithm can be tuned by different settings of the μ and ϵ values. For larger values of both parameters, more dominated individuals will be allowed in the population: Whereas a large ϵ value means that the individuals have to compete for less available "slots", a larger μ simply enlarges the storage. More individuals lead to a more accurate model estimation, but if the fraction of dominated individuals is large, a lot of sampling (and search) effort is wasted on exploring previously visited regions and thereby increasing the running time. The possibility to tune the approximation resolution via the ϵ value is an advantage compared to other existing strategies for diversity maintenance.

[3] $\epsilon_{min} := \text{Min}\{\epsilon \in \mathbb{R}^+ | \, \forall x \in \{0,1\}^n \, \exists y \in \mathbb{R} \text{ with } (1+\epsilon) f(y) \geq f(x)\}$

4.2 Comparison with Other MOEAs

In order to evaluate BMOA with respect to other MOEAs we use the results of the comparative case study from [10] and focus on the large instances of the knapsack problem with $n = 750$.

Table 1 compares BMOA to different algorithms using the *Coverage* ($\mathcal{C}$) and the *Hypervolume* ($\mathcal{S}$) metric. $\mathcal{C}(\mathcal{A}, \mathcal{B})$ denotes the relative number of non-dominated individuals contained in the population of algorithm B that are dominated by at least one individual from the population of algorithm A of a given point in time. $\mathcal{S}(\mathcal{A})$ denotes the relative volume of the objective space dominated by the solutions of algorithm A. The $(3000+3000, 10^{-6})$-BMOA is able to dominate more than half of the other algorithms' populations on nearly all instances, with the best results on the four-objective problem. The other algorithms are not able to dominate any of BMOA's non-dominated points, but they generally find a broader distribution as the hypervolume values indicate. Because of its relatively large population size, the BMOA proceeds much slower and it requires more CPU time due to the estimation of the probabilistic model.

Table 1. Results of the coverage measures $\mathcal{C}$(BMOA,*) (first entry per cell), $\mathcal{C}$(*,BMOA) (second entry) and of the hypervolume difference $\mathcal{S}$(*)-$\mathcal{S}$(BMOA) (third entry) to compare the $(3000 + 3000, 10^{-6})$-BMOA with NSGA-II, PESA, SPEA, and SPEA2 after $t = 480000$ function evaluations, median of 30 runs

*	NSGA-II	PESA	SPEA	SPEA2
KP-750-2	$0.71, 0.00, 0.006$	$0.71, 0.00, 0.008$	$0.52, 0.00, 0.009$	$0.58, 0.00, 0.013$
KP-750-3	$0.56, 0.00, 0.009$	$0.64, 0.00, 0.015$	$0.63, 0.00, 0.014$	$0.48, 0.00, 0.016$
KP-750-4	$0.72, 0.00, 0.010$	$0.97, 0.00, -0.003$	$0.99, 0.00, -0.003$	$0.80, 0.00, 0.008$

5 Conclusion

In this paper we discussed the use of Estimation of Distribution Algorithms for optimization problems involving multiple criteria. A Bayesian Multi-objective Optimization Algorithm $(\mu+\lambda, \epsilon)$-BMOA has been designed using a probabilistic model based on binary decision trees and a special selection scheme based on ϵ-archives. The convergence behavior of the algorithm can be tuned via the values of μ, the minimal population size to estimate the probabilistic model, and ϵ, the approximation factor.

The empirical tests on the 0/1 multi-objective knapsack problem show that the BMOA is able to find a good model of the Pareto set for the smaller instances. In order to find also the outer region of the Pareto set, large μ and ϵ values are required, which slows down the optimization process considerably. Further research could assess MBOA on other multi-objective combinatorial optimization problems with stronger variable interactions and on continuous problems.

From the decision-aid point of view it would be interesting to exploit the Bayesian model also outside the algorithm itself. The compact description of the model could assist a decision maker who can analyze the decision trees to get more insight into the structure of the Pareto set and to learn about correlations in the decision problem at hand.

Acknowledgments

This research has been carried out under the financial support of the Research intention no. CEZ: J22/98: 262200012 - "Research in information and control systems" (Ministry of Education, CZ), the research grant GA 102/02/0503 "Parallel system performance prediction and tuning" (Grant Agency of Czech Republic) and the Swiss National Science Foundation (SNF) under the ArOMA project 2100-057156.99/1.

References

1. D. Heckerman, D. Geiger, and M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. Technical report, Microsoft Research, Redmont, WA, 1994.
2. J. D. Knowles and D. W. Corne. M-PAES: A memetic algorithm for multiobjective optimization. In *Congress on Evolutionary Computation (CEC 2000)*, volume 1, pages 325–332, Piscataway, NJ, 2000. IEEE Press.
3. M. Laumanns, G. Rudolph, and H.-P. Schwefel. Mutation control and convergence in evolutionary multi-objective optimization. In *MENDEL 2001. 7th Int. Conf. on Soft Computing*, pages 24–29. Brno University of Technology, 2001.
4. M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3), 2002.
5. M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. IlliGAL Report No. 99018, 1999.
6. M. Pelikan, D. E. Goldberg, and K. Sastry. Bayesian optimization algorithm, decision graphs, and occams razor. IlliGAL Report No. 2000020, 2000.
7. G. Rudolph. On a multi-objective evolutionary algorithm and its convergence to the pareto set. In *IEEE Int'l Conf. on Evolutionary Computation (ICEC'98)*, pages 511–516, Piscataway, 1998. IEEE Press.
8. J. Schwarz and J. Ocenasek. Multiobjective bayesian optimization algorithm for combinatorial problems: Theory and practice. *Neural Network World*, 11(5):423–441, 2001.
9. D. Thierens and P. A. N. Bosman. Multi-objective mixture-based iterated density estimation evolutionary algorithms. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 663–670. Morgan Kaufmann, 2001.
10. E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation, and Control*, 2002.