

Evolutionary Learning of Boolean Queries by Multiobjective Genetic Programming

Oscar Cordón, Enrique Herrera-Viedma, and María Luque

Dept. of Computer Science and A.I. E.T.S. de Ingeniería Informática
University of Granada. 18071 - Granada (Spain)
{ocordon,viedma}@decsai.ugr.es, mluque@fedro.ugr.es

Abstract. The performance of an information retrieval system is usually measured in terms of two different criteria, precision and recall. This way, the optimization of any of its components is a clear example of a multiobjective problem. However, although evolutionary algorithms have been widely applied in the information retrieval area, in all of these applications both criteria have been combined in a single scalar fitness function by means of a weighting scheme. In this paper, we will tackle with a usual information retrieval problem, the automatic derivation of Boolean queries, by incorporating a well known Pareto-based multiobjective evolutionary approach, MOGA, into a previous proposal of a genetic programming technique for this task.

1 Introduction

Information retrieval (IR) may be defined, in general, as the problem of the selection of documentary information from storage in response to search questions provided by a user [15]. Information retrieval systems (IRSs) are a kind of information systems that deal with data bases composed of information items—documents that may consist of textual, pictorial or vocal information—and process user queries trying to allow the user to access to relevant information in an appropriate time interval. Nowadays, the world wide web constitutes the main example of an IRS.

Most of the commercial IRSs are based on the Boolean IR model [17], based on the use of Boolean queries where the query terms are joined by the logical operators AND and OR. This way, the user needs to have a clear knowledge on how to connect the query terms together using the Boolean operators in order to build a query defining his information needs and allowing him to retrieve relevant documents.

The difficulty found by non-expert users to formulate these kinds of queries makes necessary the design of automatic methods for this task. The paradigm of Inductive Query by Example (IQBE) [4], where a query describing the information contents of a set of documents provided by a user is automatically derived, can be useful to assist the user in the query formulation process. Focusing on the Boolean IR model, the only existing approach is that of Smith and Smith [16], which is based on genetic programming (GP) [11]. As usual in the topic, this

approach is guided by a weighted fitness function combining the two common criteria to measure the performance of an IRS, precision and recall.

In this paper, we will propose a new IQBE algorithm to learn Boolean queries by extending Smith and Smith approach in order to transform it into a multi-objective evolutionary algorithm (EA) not based on a weighted fitness function [5]. This way, we will work as in [14], incorporating MOGA [9] Pareto-based evolutionary multiobjective components into GP. The experimental testbed will be based on one of the most known IR benchmarks, the Cranfield document collection [15, 1].

With this aim, this contribution is structured as follows. Section 2 is devoted to the preliminaries, including the basis of Boolean IRSs, the definition of both precision and recall criteria, and the basis of IQBE techniques. Then, Smith and Smith's proposal is reviewed in Section 3. Section 4 presents the adaptations made to include the Pareto-based multiobjective EA components in the latter algorithm while the experiments developed to test the new proposal are showed in Section 5. Finally, several concluding remarks are pointed out in Section 6.

2 Preliminaries

2.1 Boolean Information Retrieval Systems

An IRS is basically constituted of three main components: *documentary data base*, *query subsystem* and *matching or evaluation mechanism*, whose composition for Boolean IRSs are introduced as follows.

The documentary data base. This component stores the documents and the representation of their information contents. It is associated with the *indexer module*, which automatically generates a representation for each document by extracting the document contents. Textual document representation is typically based on index terms (that can be either single terms or sequences) which are the content identifiers of the documents.

In the Boolean retrieval model, the indexer module performs a binary indexing in the sense that a term in a document representation is either significant (appears at least once in it) or not (it does not appear in it at all). Let D be a set of documents and T be a set of unique and significant terms existing in them. The indexer module of the Boolean IRS defines an indexing function: $F : D \times T \rightarrow \{0, 1\}$, where $F(d, t)$ takes value 1 if term t appears in document d and 0 otherwise.

The query subsystem. It allows the users to formulate their queries and presents the relevant documents retrieved by the system to them. To do so, it includes a *query language*, that collects the rules to generate legitimate queries and procedures to select the relevant documents.

Boolean queries are expressed using a query language that is based on query terms and permits combinations of simple user requirements with logical operators AND, OR and NOT [17]. The result obtained from the processing of a query

is a set of documents that totally match with it, i.e., only two possibilities are considered for each document: to be or not to be relevant for the user's needs, represented by his query.

The matching mechanism. It evaluates the degree to which the document representations satisfy the requirements expressed in the query, the *retrieval status value* (RSV), and retrieves those documents that are judged to be relevant to it.

As said, the RSV has only two values associated, 0 and 1, in Boolean IRSs. In order to match a query, a document has to fulfill it completely, i.e., it has to include the positive query terms specified in the search expression and not to include those that have been specifically given in that negative way. In order to obtain the set of relevant documents for a query, it is represented as a parse tree and is evaluated from the leaves to the root. Each leaf is associated to the set of documents including (or not including) the corresponding (negative) query term. Then, the retrieved document sets in the inner nodes are computed by applying set arithmetic (with the AND operator being the set intersection and the OR operator standing for the set union). The final set of retrieved documents is that associated to the root when finishing the evaluation of the tree.

2.2 Evaluation of Information Retrieval Systems

There are several ways to measure the quality of an IRS, such as the system efficiency and effectiveness, and several subjective aspects related to the user satisfaction (see, for example, [1], chapter 3). Traditionally, the retrieval effectiveness—usually based on the document relevance with respect to the user's needs—is the most considered. There are different criteria to measure this aspect, with the *precision* and the *recall* being the most used.

Precision is the rate between the relevant documents retrieved by the IRS in response to a query and the total number of documents retrieved, whilst recall is the rate between the relevant documents retrieved and the total number of relevant documents to the query existing in the data base [17]. The mathematical expression of each of them is showed as follows:

$$P = \frac{\sum_d r_d \cdot f_d}{\sum_d f_d} \quad ; \quad R = \frac{\sum_d r_d \cdot f_d}{\sum_d r_d}$$

with $r_d \in \{0, 1\}$ being the relevance of document d for the user and $f_d \in \{0, 1\}$ being the retrieval of document d in the processing of the current query. Notice that both measures are defined in $[0, 1]$, with 1 the optimal value.

Notice that the only way to know all the relevant documents for a query existing in a documentary base (needed to compute the recall measure) is to evaluate them all one by one. Due to this and to the relevance subjectivity, there are several classical documentary bases available, each of them with a set of queries with known relevance judgments, that can be used to test the different new proposals in the field of IR [15, 1]. In this contribution, we will deal with the well known Cranfield collection.

As said, up to our knowledge, all the previous applications of machine learning techniques to any of the IRS components trying to optimize both criteria have considered a weighted combination of them. This is why the aim of our contribution is to provide a first step on the application of Pareto-based multi-objective EAs to IR in order to evolve a complete set of Pareto optimal solutions optimizing both criteria simultaneously.

2.3 Inductive Query by Example

IQBE was proposed in [4] as “a process in which searchers provide sample documents (examples) and the algorithms induce (or learn) the key concepts in order to find other relevant documents”. This way, IQBE is a process for assisting the users in the query formulation process performed by machine learning methods. It works by taking a set of relevant (and optionally, non relevant documents) provided by a user—that can be obtained from a preliminary query or from a browsing process in the documentary base—and applying an off-line learning process to automatically generate a query describing the user’s needs (as represented by the document set provided by him). The obtained query can then be run in other IRSs to obtain more relevant documents. This way, there is no need that the user interacts with the process as in other query refinement techniques such as relevance feedback [15].

There have been proposed IQBE proposals for the different existing IR models. As said, Smith and Smith [16] proposed the GP algorithm to derive Boolean queries that will be considered in this paper. On the other hand, all of the machine learning methods considered in Chen et al.’s paper [4] (regression trees, genetic algorithms and simulated annealing) dealt with the vector space model [15]. Moreover, there are several approaches for the derivation of weighted Boolean queries for fuzzy IRSs [3], such as the GP algorithm of Kraft et al. [12], the niching GA-P method [7] and the simulated annealing-GP hybrid [8]. For descriptions of some of the previous techniques based on EAs refer to [6].

3 The Smith and Smith’s Genetic Programming-based Inductive Query by Example Algorithm for Boolean Information Retrieval Systems

In [16], Smith and Smith proposed an IQBE to derive Boolean queries based on GP. Its components are described next:

Coding Scheme: The Boolean queries are encoded in expression trees, whose terminal nodes are query terms and whose inner nodes are the Boolean operators *AND*, *OR* or *NOT*.

The different expression trees are derived from the following grammar [16]:

$$\begin{aligned} \langle \text{QUERY} \rangle &::= \langle \text{TERM} \rangle \mid (\langle \text{QUERY} \rangle \langle \text{OPERATOR} \rangle \langle \text{QUERY} \rangle) \\ \langle \text{OPERATOR} \rangle &::= \text{AND} \mid \text{OR} \mid \text{NOT} \\ \langle \text{TERM} \rangle &::= t_1 \mid \dots \mid t_n \end{aligned}$$

Selection Scheme: Each generation is based on selecting two parents, with the best fitted one having a greater chance to be chosen, and generating two offspring from them. Both offspring are added to the current population¹.

Genetic Operators: The usual GP crossover is considered [11], which is based on randomly selecting one edge in each parent and exchanging both subtrees from these edges between the both parents. No mutation operator is considered².

Generation of the Initial Population: All the individuals in the first population are randomly generated. A pool is created with all the terms included in the set of relevant documents provided by the user, having those present in more documents a higher probability of being selected.

Fitness function: The following function combining precision and recall is used:

$$F = \frac{1}{\alpha \cdot (\frac{1}{P}) + \beta \cdot (\frac{1}{R})}$$

where α and β are the weighting factors. As can be seen, F has to be maximized. Moreover, when comparing two queries with the same F value, the shorter one is preferred.

4 Incorporating Pareto-based Multiobjective Components to the Smith and Smith's Algorithm

As said, the Pareto-based multiobjective EA considered to be incorporated to the basic Smith and Smith's GP algorithm in this first work has been Fonseca and Fleming's MOGA [9]. The selection scheme of MOGA is based on dividing the population in several ranked blocks and assigning a higher probability of selection to the blocks with a lower rank, taking into account that individuals in the same block will be equally preferable and thus will receive the same selection probability. The rank of an individual in the population (and consequently of his belonging block) will depend on the number of individuals dominating it.

Therefore, the selection scheme of our multiobjective GP involves the following four steps:

1. Each individual is assigned a rank equal to the number of individuals dominating it plus one (chromosomes encoding non-dominated solutions receive rank 1).
2. The population is increasingly sorted according to that rank.

¹ Our implementation differs in this point as we consider a classical generational scheme where the selection probabilities are assigned by the proportional scheme and the reproduction is performed by Baker's stochastic universal sampling [2].

² We do use a mutation operator which changes a randomly selected term or operator by a random one, or a randomly selected subtree by a randomly generated one.

3. Each individual is assigned a fitness value which depends on its ranking in the population. In this contribution, we consider the following assignment:

$$f(C_i) = \frac{1}{\text{rank}(C_i)}.$$
4. The fitness assignment of each equivalence class (group of individuals with the same rank, i.e., which are non dominated among them) is averaged among them, so that all of them finally receive the same fitness value.

Once the final fitness values have been computed, a usual selection mechanism is applied. In this contribution we consider the proportional assignment and Baker's stochastic universal sampling [2].

It is known that the MOGA selection scheme can cause a large selection pressure that might produce premature convergence. Fonseca and Fleming considered this issue and suggested to use a niching method to appropriately distribute the population in the Pareto [9]. However, as said in [5], one of the main weaknesses of MOGA is that sharing is performed in the objective space, thus making more difficult that two different Pareto solutions with the same objective function values can simultaneously coexist in the population.

For example, this is not a desirable characteristic in our case, as we are interested on obtaining as many queries with the same precision-recall values as possible. Fortunately, as Coello also mentions, there is no specific requirement in the MOGA algorithm to perform sharing in the objective space.

This way, in this paper we apply niching in the parameter (genotypic) space. To do so, we have to keep in mind that we are dealing with chromosomes encoding Boolean queries, and hence we need a metric capable of measuring distances between expression trees. In our case, this is put into effect by the so-called *edit* or *Levenshtein distance* [13], a text metric that computes the distance between two strings as the number of edit (delete, insert or change) steps needed to convert one into the other. In order to compute distances between trees with this metric, we apply it on the strings encoding the preorder representation of the trees.

Let $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_m)$ be the two tree preorder strings. The edit distance between them is recursively computed as follows³:

$$E((a_1, \dots, a_n), (b_1, \dots, b_m)) = \begin{cases} n, & \text{if } m = 0 \\ m, & \text{if } n = 0 \\ \min\{E((a_1, \dots, a_{n-1}), (b_1, \dots, b_m)) + 1, \\ E((a_1, \dots, a_n), (b_1, \dots, b_{m-1})) + 1, & \text{otherwise} \\ E((a_1, \dots, a_{n-1}), (b_1, \dots, b_{m-1})) + d(a_n, b_m)\}, \end{cases}$$

with $d(x, y) = 1$ if $x = y$, and 0 otherwise, being the character distance.

Once a valid metric for trees has been defined, it is easy to apply sharing by using the classical Goldberg and Richardson's sharing function [10]:

³ We should note that in this contribution we have been computed it iteratively by the corresponding Dynamic Programming algorithm.

$$F(C_i) = \frac{f(C_i)}{\sum_{j=1}^M Sh(d(C_i, C_j))} \quad ; \quad Sh(d) = \begin{cases} 1 - (\frac{d}{\sigma_{share}})^\gamma, & \text{if } d < \sigma_{share} \\ 0, & \text{otherwise} \end{cases}$$

with σ_{share} being the niche radius.

5 Experiments Developed and Analysis of Results

As said, the experimental study has been developed using the *Cranfield* collection, composed of 1400 documents about Aeronautics. The 1400 textual documents has been automatically indexed in the usual way by first extracting the non-stop words, thus obtaining a total number of 3857 different indexing terms, and then considering the binary indexing to generate the term weights in the document representations.

Among the 225 queries associated to the Cranfield collection, we have selected those presenting 20 or more relevant documents. The resulting seven queries (numbers 1, 2, 23, 73, 157, 220 and 225) have 29, 25, 33, 21, 40, 20 and 25 relevant documents associated, respectively.

Apart from our Pareto-based Multiobjective proposal, we have also run the basic Smith and Smith's algorithm with a typical setting for the weights in the fitness function $((\alpha, \beta) = (1.2, 0.8))$. Every algorithm have been run three times with different initializations during the same fixed number of fitness function evaluations (100000). The common parameter values considered are a maximum of 20 nodes for the trees, 0.8 and 0.2 for the crossover and mutation probabilities, respectively, and a population size of $M = 1600$ queries. The high value for the latter parameter is because it is well known that GP requires large population sizes to achieve good performance. Finally, the sharing function parameter γ takes value 2 and the niche radius σ_{share} has been experimentally set to 4 (a 20% of the maximum tree size).

The results obtained by the basic algorithm are showed in Table 1, where only the best of the three runs is reported. In this table, $\#q$ stands for the corresponding query number, Run for the number of the run where this result was derived, Sz for the generated query size, P and R for the precision and recall values, respectively, $\#rt$ for the number of documents retrieved by the query, and $\#rr$ for the number of relevant documents retrieved.

On the other hand, Table 2 shows several statistics corresponding to our multiobjective proposal. First, the number of non-dominated solutions found in the last population and the number of different queries (trees) existing among them are collected in columns $\#p$ and $\#dp$, respectively. Then, the distribution of the generated Pareto is reported as the value of two of the usual multiobjective EA metrics M_2 and M_3^* [18]. On the one hand, $M_2 \in [0, \#p]$ measures the distribution of the genotypes of the $\#p$ non-dominated solutions found⁴ (i.e.,

⁴ The value of neighborhood parameter σ considered in this metric has been set to the value of the niche radius σ_{share} .

Table 1. Results obtained by the basic Smith and Smith's IQBE algorithm

# <i>q</i>	<i>Run</i>	<i>Sz</i>	<i>P</i>	<i>R</i>	# <i>rr</i> /# <i>rt</i>
1	3	17	1.0	0.1379	4/4
2	1,3	19	1.0	0.16	4/4
23	1	17	1.0	0.1515	5/5
73	1	19	1.0	0.2857	6/6
157	1	15	1.0	0.15	6/6
220	1,2,3	19	1.0	0.15	3/3
225	2	13	1.0	0.2	5/5

the diversity of the solutions found). On the other hand, M_3^* estimates the range to which the Pareto front spreads out in the objective values. As our problem is composed of just two objectives, it is equal to the distance among the objective vectors of the two outer solutions (the two queries with maximum precision and maximum recall, respectively), whose associated data are showed in the remaining columns of the table.

In view of these results, the performance of our proposal is very significant. On the one hand, it overcomes the basic Smith and Smith's algorithm in all cases as the results of the latter when considering typical values for the weighted combination are dominated by the solutions in the Pareto front of the former. It seems that the use of a weighting scheme and the lack of a niching scheme make the basic algorithm not to perform appropriately. On the other hand, the main aim of this paper have been clearly fulfilled since the Pareto fronts obtained are very well distributed, as demonstrated by the high number of solutions included in them and the high values in the M_2 and M_3^* metrics. As an example, Figure 1 shows the Pareto front obtained in the first run of query 1.

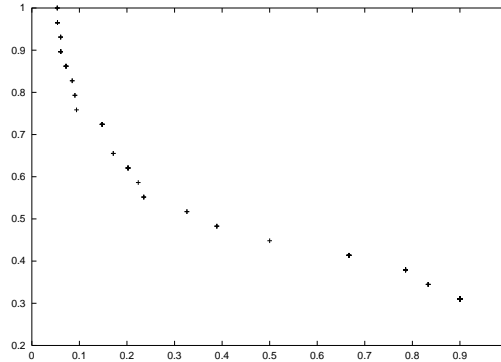
**Fig. 1.** Pareto front of the first run for query 1

Table 2. Results obtained by the proposed multiobjective IQBE algorithm

#q	Run	#p	#dp	M_2	M_3	P	R	Sz	#rr/#rt	P	R	Sz	#rr/#rt
1	1	134	74	62.7	1.292	1.0	0.2758	19	8/8	0.0534	1.0	19	29/543
	2	149	72	71.1	1.269	1.0	0.3448	19	10/10	0.0433	1.0	19	29/669
	3	126	62	59.8	1.294	1.0	0.2758	19	8/8	0.0475	1.0	19	29/610
2	1	118	62	54.7	1.269	1.0	0.36	19	9/9	0.0290	1.0	19	25/861
	2	559	478	274.1	1.301	1.0	0.28	19	7/7	0.0261	1.0	19	25/956
	3	411	337	202.1	1.299	1.0	0.28	19	7/7	0.0307	1.0	19	25/813
23	1	126	69	58.4	1.321	1.0	0.3030	19	10/10	0.0274	1.0	19	33/1204
	2	140	73	66.6	1.319	1.0	0.2121	19	7/7	0.0425	1.0	19	33/776
	3	118	62	54.7	1.269	1.0	0.2121	19	7/7	0.0464	1.0	19	33/710
73	1	101	48	44.1	1.180	1.0	0.5238	19	11/11	0.0820	1.0	19	21/256
	2	96	39	43.3	1.184	1.0	0.5238	19	11/11	0.0739	1.0	19	21/284
	3	218	145	101.7	1.210	1.0	0.4761	19	10/10	0.0573	1.0	19	21/366
157	1	274	185	133.8	1.301	1.0	0.25	19	10/10	0.0556	1.0	19	40/719
	2	131	56	61.8	1.310	1.0	0.225	19	9/9	0.0568	1.0	19	40/704
	3	119	59	56.8	1.316	1.0	0.225	19	9/9	0.0429	1.0	19	40/931
220	1	354	273	172.4	1.226	1.0	0.45	19	9/9	0.0465	1.0	19	20/430
	2	95	56	41.8	1.243	1.0	0.4	19	8/8	0.0543	1.0	19	20/368
	3	122	60	48.6	1.202	1.0	0.5	19	10/10	0.0549	1.0	19	20/364
225	1	136	60	62.3	1.248	1.0	0.4	19	10/10	0.0413	1.0	19	25/604
	2	373	314	182.8	1.271	1.0	0.36	19	9/9	0.0239	1.0	19	25/1042
	3	150	52	70.8	1.248	1.0	0.4	19	10/10	0.0421	1.0	19	25/593

6 Concluding Remarks

The automatic derivation of Boolean queries has been considered by incorporating the MOGA Pareto-based multiobjective evolutionary approach to an existing GP-based IQBE proposal. The proposed approach has performed appropriately in seven queries of the well known Cranfield collection in terms of absolute retrieval performance and of the quality of the obtained Paretos.

In our opinion, many different future works arise from this preliminary study. On the one hand, more advanced Pareto-based evolutionary multiobjective schemes (such as those considering an auxiliary population in order to better cover the Pareto front [5]) can be incorporated to the basic GP algorithm in order to improve the performance of the multiobjective EA proposed. On the other hand, preference information of the user on the kind of queries to be derived can be included in the Pareto-based selection scheme in the form of a goal vector whose values are adapted during the evolutionary process [9]. Finally, and more generically, Pareto-based evolutionary multiobjective optimization can be applied either to the automatic derivation of queries for other kinds of IR models (such as the extended Boolean ones tackled in the EAs proposed in [7, 8, 12]) or to other IR problems being solved by EAs [6], thus benefiting from the potential of these techniques in the problem solving.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval, Addison-Wesley (1999).
2. Baker, J.E.: Reducing bias and inefficiency in the selection algorithm, Proc. Second International Conference on Genetic Algorithms (ICGA'87), Hillsdale, NJ, (1987) 14–21.
3. Bordogna, G., Carrara, P., Pasi, G.: Fuzzy approaches to extend Boolean information retrieval, in: P. Bosc, J. Kacprzyk (Eds.), Fuzziness in Database Management Systems (1995) 231–274.
4. Chen, H.: A machine learning approach to inductive query by examples: an experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing, Journal of the American Society for Information Science **49:8** (1998) 693–705.
5. Coello, C.A.: A comprehensive survey of evolutionary-based multiobjective optimization techniques, Knowledge and Information Systems, **1:3** (1999) 269–308.
6. Cordon, O., Moya, F., Zarco, C.: A brief study on the application of genetic algorithms to information retrieval (in spanish), Proc. Fourth International Society for Knowledge Organization (ISKO) Conference (EOCONSID'99), Granada, Spain, (April, 1999) 179–186.
7. Cordon, O., Moya, F., Zarco, C.: A GA-P algorithm to automatically formulate extended Boolean queries for a fuzzy information retrieval system, Mathware & Soft Computing **7:2-3** (2000) 309–322.
8. Cordon, O., Moya, F., Zarco, C.: A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems, Soft Computing **6:5** (2002).
9. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, Discussion and Generalization, Proc. Fifth International Conference on Genetic Algorithms (ICGA'93), San Mateo, CA (July, 1993) 416–423.
10. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization, Proc. Second International Conference on Genetic Algorithms (ICGA'87), Hillsdale, NJ, (1987) 41–49.
11. Koza, J.: Genetic programming. On the programming of computers by means of natural selection, The MIT Press (1992).
12. Kraft, D.H., Petry, F.E., Buckles, B.P., Sadasivan, T.: Genetic algorithms for query optimization in information retrieval: relevance feedback, in: E. Sanchez, T. Shibata, L.A. Zadeh, Genetic Algorithms and Fuzzy Logic Systems, World Scientific (1997) 155–173.
13. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals, Sov. Phys. Dokl. **6** (1966) 705–710.
14. Rodríguez-Vazquez, K., Fonseca, C.M., Fleming, P.J.: Multiobjective genetic programming: A nonlinear system identification application, Late Breaking Papers at the Genetic Programming 1997 Conference, Stanford, CA (July, 1997) 207–212.
15. Salton, G., McGill, M.J.: Introduction to modern information retrieval, McGraw-Hill (1989).
16. Smith, M.P., Smith, M.: The use of genetic programming to build Boolean queries for text retrieval through relevance feedback, Journal of Information Science **23:6** (1997) 423–431.
17. van Rijsbergen, C.J.: Information Retrieval (2nd edition), Butterworth (1979).
18. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary Computations **8:2** (2000) 173–195.