Multi-objective Evolutionary Algorithms in the Automatic Learning of Boolean Queries: A Comparative Study

A.G. Lopez-Herrera[1], E. Herrera-Viedma[2], F. Herrera[2], C. Porcel[3], and S. Alonso[4]

[1] Dept. Computer Sciences, University of Jaén, Spain
aglopez@ujaen.es
[2] Dept. Computer Sciences and A.I, University of Granada, Spain
{viedma,Herrera}@decsai.ugr.es
[3] Dept. Computer Sciences and N.A., University of Córdoba, Spain
carlos.porcel@uco.es
[4] Software Engineering Dept., University of Granada, Spain
salonso@decsai.ugr.es

Abstract. The performance of Information Retrieval Systems (IRSs) is usually measured using two different criteria, precision and recall. In such a way, the problem of tuning an IRS may be considered as a multi-objective optimization problem. In this contribution, we focus on the automatic learning of Boolean queries in IRSs by means of multi-objective evolutionary techniques. We present a comparative study of four multi-objective evolutionary optimization techniques of general-purpose (NSGA-II, SPEA2 and two MOGLS) to learn Boolean queries.

Keywords: Information Retrieval Systems, Genetic Programming, Inductive Query By Example, Multi-objective Evolutionary Algorithms, Query Learning.

1 Introduction

Information Retrieval (IR) may be defined as the problem of selecting documentary information from storage in response to searches provided by a user in form of queries [2], [24]. IRSs deal with documentary databases containing textual, pictorial or vocal information and they process user queries to allow the user to access relevant information in an appropriate time interval.

The Boolean IR model [26] is frequently used to build queries in the IRSs. However, it presents some limitations: a Boolean query is defined by a set of terms joined by the logical operators AND, OR and NOT, but to build Boolean queries is not usually easy neither very intuitive [2]. This problem becomes a more serious issue if the users do not have previous experience with the model. A possible solution to overcome this problem is to build automatic aid tools to assist users to express their information needs by means of Boolean queries. Inductive Query By Example (IQBE) [5], where a query describing the information contexts of a set of key documents provided by the user is automatically derived or learned, is an useful paradigm to assist users to express queries.

O. Castillo et al. (Eds.): Theor. Adv. and Appl. of Fuzzy Logic, ASC 42, pp. 71–80, 2007.
springerlink.com

Some approaches based on the IQBE paradigm to learn queries have been proposed in the specialized literature [8], [9], [10], [18], [20], [25]. They are based on a kind of Evolutionary Algorithms (EA) [1] as it is Genetic Programming (GP) [19], where queries are represented as expression syntax trees and adapted using selection, crossing and mutation methods. They are usually guided by a fitness function that combines precision and recall [7] (the traditional performance measures in IR) in an unique objective. One of the main characteristics of this approach is that it only provides a single solution query in each run (the one that maximizes the fitness function). However, there exist a kind of EAs specially designed for multi-objective problems, MOEAs, which are able to obtain different solutions to the problem in a single run [6]. As multi-objective problems are characterized by the fact that several objectives have to be simultaneously optimized, there is not usually a single best solution solving the problem, that is, being better than the remainder with respect to every objective, but there exist a set of solutions which are superior to the remainder when all the objectives are considered. This set is called the Pareto set. These solutions are known as non-dominated solutions [4], while the remainder are known as dominated solutions. Since none of the Pareto set solutions is absolutely better than the other non-dominated solutions, all of them are equally acceptable regarding the satisfaction of all the objectives.

Recently an IQBE MOEA in the context of automatic learning of Boolean queries in IRSs has been proposed [9]. This IQBE MOEA is based on the first version of Strength Pareto Evolutionary Algorithm (SPEA) [30] and uses GP concepts to extend, toward the multi-objective context, the single-objective Boolean IQBE EA proposal of Smith and Smith [25]. However, there exist other MOEAs which usually improve the performance of SPEA [12], [13], [15], [29].

In this work, a comparative study of the performance of four of the currently most successful MOEAs applied to the automatic learning of Boolean queries will be done. The studied MOEAs are: the second version of Non-dominated Sorting Genetic Algorithm (NSGA-II) [12], the second version of Strength Pareto Evolutionary Algorithm (SPEA2) [29], and two MOGLS (MOEAs using Local Search): MOGLS-I [13] and MOGLS-J [15]. All of them are adapted to use GP concepts and optimize both objectives (precision and recall) simultaneously, extending the Smith and Smith IQBE EA [25] proposal to the multi-objective context. Experimental results show that NSGA-II, adapted with GP concepts, obtain the best performance in the context of automatic learning of Boolean queries.

To do this, this paper is structured as follows. In Section 2 IRSs' foundations and the IQBE paradigm are drawn. Section 3 describes the four MOEAs with GP concepts which are used. In Section 4 the experimentation framework is described, and finally Section 5 presents some conclusions.

2 Preliminaries

In this section, we introduce the foundations of the IRSs, their components and procedure of evaluation and a brief description of the IQBE paradigm.

2.1 Information Retrieval Systems

An IRS is basically composed by three main components [3]:

The documentary database: This component stores the documents and the representation of their contents. Textual documents representation is typically based on index terms (that can be either single terms or sequences), which work as content identifiers for the documents. We assume that the database is built like in usual IRSs [2], [24]. Therefore, IRS-user interaction is unnecessary because it is built automatically. The database stores a finite set of documents $D=\{d_1,\cdots,d_m\}$, a finite set of index terms $T=\{t_1,\cdots,t_l\}$, and the representation R_{d_j} of each document d_j characterized by a numeric indexing function $F: D \times T \rightarrow [0,1]$ such that $R_{d_j}=\sum_{i=1}^{l} F(d_j,t_i)/t_i$ is the representation of d_j in fuzzy sets notation. Using numeric values in (0,1), F can weight index terms according to their significance in describing the content of a document in order to improve the retrieval of documents.

The query subsystem: It allows users to formulate their information needs (queries) and presents the relevant documents which are retrieved by the system. To do this, each query is expressed as a combination of index terms which are connected by the Boolean operators AND ($\wedge$), OR ($\vee$), and NOT ($\neg$).

The matching mechanism: It evaluates the degrees (the Retrieval Status Value (RSV)) to which the document representation satisfy the requirements expressed in the query, and it retrieves the documents that are judged to be relevant. To evaluate Boolean queries, the matching function uses a constructive bottom-up process based on the separability criterion [27]. This process includes two steps:

- Firstly, the documents are evaluated according to their relevance only to the terms of the query. In this step, a partial relevance degree is assigned to each document with respect to every term in the query.
- Secondly, the documents are evaluated according to their relevance to the Boolean combination of the terms (their partial relevance degree), and so on, working in a bottom-up fashion until the whole query is processed. In this step, a total relevance degree is assigned to each document that is used to rank the documents from the most relevant one to the less relevant.

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 user satisfaction [2]. Traditionally, the retrieval effectiveness is based on the document relevance with respect to the users needs. There are different criteria to measure this aspect, but precision and recall are the most used. Precision is the ratio between the relevant documents retrieved by the IRS in response to a query and the total number of documents retrieved, whilst recall is the ratio between the number of relevant documents retrieved and the total number of relevant documents for the query that exist in the database [26]. The mathematical expression of each of them is:

$$P=\frac{D_{rr}}{D_{tr}},\ R=\frac{D_{rr}}{D_{rt}} \tag{1}$$

where D_{rr} is the number of relevant documents retrieved, D_{tr} is the total number of documents retrieved and D_{rt} is the total number of relevant documents for the query which exist in the database. P and R are defined in [0, 1], being 1 the optimal value.

We notice that the only way to know all the relevant documents existing for a query in the database (value used in the R measure) is to evaluate all documents. Due to this fact and tacking into account that relevance is subjective, there are some classic documentary databases (TREC, CACM, Cranfield) available, each one with a set of queries for which the relevance judgments are known, so that they can be used to verify the new proposals in the field of the IR [2], [22]. In this contribution, we use the Cranfield collection.

2.3 The IQBE Paradigm

The IQBE paradigm was proposed by Chen [5] as a process in which users provide documents (examples) and an algorithm induces (or it learns) the key concepts of the examples with the purpose of finding other and equally relevant documents. In this way, IQBE can be seen as a technique to assist users in the query building process by using automatic learning methods.

It works taking a set of relevant documents (and optionally non-relevant documents) provided by the user (they can be obtained from a preliminary query or from a browsing process through the documentary database) and applying an automatic learning process to generate a query that describes the user information needs (represented by the previous set of documents). The query that is obtained can be executed in other IRSs to obtain new relevant documents. In this way, it is not necessary for the user to interact with the IR process which is mandatory in other techniques for query refinement as the relevance feedback [22].

Several IQBE techniques for different IR models have been proposed [7]. The most used IQBE models are based on GP concepts, with queries being represented by expression syntax trees and the algorithms are articulated on the basis of the classic operators: cross, mutation and selection.

3 Structure of the MOEAs with GP Concepts

3.1 Components

The four studied MOEAs with GP (MOEAs-GP) concepts share the following components:

- *Codification scheme*: Boolean queries are encoded in expression syntax trees, whose terminal nodes are query terms and whose inner nodes are the Boolean operators AND, OR and NOT.
- *Crossover operator*: subtrees are randomly selected and crossover in two randomly selected queries.
- *Mutation operator*: changes a randomly selected term or operator in a randomly selected tree.
- *Initial population*: all individuals of the first generation are generated in a random way. The population is created including all the terms in the relevant documents

provided by the user. Those that appear in more relevant documents will have greater probability of being selected.
- *Objectives to optimize*: precision and recall.
- *Local search*: it is based on the Crossover Hill-Climbing (XHC) algorithm defined in [21] and adapted to expression syntax trees. This XHC operator uses hill-climbing as the move accepting criterion of the search and uses crossover as the move operator. XHC maintains a pair of predecessors and repeatedly performs crossover on this pair until some number of offspring, n_{off} , is reached. Then, the best offspring is selected and it replaces the current solution only if it is better. The process iterates n_{it} times and returns the final current solution. This XHC requieres values for n_{off} and n_{it}, and a starting pair of parents.

3.2 NSGA-II-GP

NSGA-II [12] is a very complete algorithm since, not only incorporates a strategy of preservation of an elite population, but in addition, it uses an explicit mechanism to preserve diversity.

NSGA-II works with a population of offsprings Q_t, which is created using a predecesor population P_t. Both populations (Q_t and P_t) are combined to form a unique population R_t, with a size $2{\cdot}M$, that is examined in order to extract the front of the Pareto. Then, an arrangement on the non-dominated individuals is done to classify the R_t population. Although this implies a greater effort compared with the arrangement of the set Q_t, it allows a global verification of the non-dominated solutions that as much belong to the population of offsprings a the one of the predecesors.

Once the arrangement of the non-dominated individuals finishes, the new generation (population) is formed with solutions of the different non-dominated fronts, taking then alternatively from each of the fronts. It begins with the best front of non-dominated individuals and continues with the solutions of the second one, later with third one, etc.

Since the R_t size is $2{\cdot}M$, it is possible that some of the front solutions have to be eliminated to form the new population.

In the last states of the execution, it is usual that the majority of the solutions are in the best front of not-domintad solutions. It is also probable the size of the best front of the combined population R_t be bigger than M. It is then, when the previous algorithm assures the selection a diverse set of solutions of this front by means of the method of niches. When the whole population converges to the Pareto-optimal frontier, the algorithm continues, so that the best distribution between the solutions is assured.

3.3 SPEA2-GP

SPEA2 [29] introduces elitism by explicitly maintaining an external population. This population stores a fixed number of the non-dominated solutions found from the beginning of the simulation.

In each generation, the new non-dominated solutions are compared with the existing external population and the resulting non-dominated solutions are preserved. In addition, SPEA2 uses these elite solutions in the genetic operations with the current population to guide the population towards good regions in the search space.

The algorithm begins with a randomly created population P_0 of size M and an external population $\underline{P}_0$ (initially empty) which has a maximum capacity $\underline{M}$. In each generation t, the best non-dominated solutions (belonging to the best non-dominated front) of the populations P_t and $\underline{P}_t$ are copied in the external population $\underline{P}_{t+1}$. If the size of $\underline{P}_{t+1}$ exceeds M, then $\underline{P}_{t+1}$ is reduced by means of a truncate operator; on the other hand, $\underline{P}_{t+1}$ is filled up with dominated solutions from P_t and $\underline{P}_t$. This truncate operator is used to maintain the diversity of the solutions.

From $\underline{P}_{t+1}$, a pool of individuals is obtained applying a binary tournament selection operator with replacement. These individuals are crossed and mutated to obtain the new generation P_{t+1}.

3.4 MOGLS-GP-I

MOGLS is an hybrid approach that combines concepts of MOEAs and Local Search for improving the current population. Several alternatives have been proponed in the specialized literature [13], [14], [15], [16]. In this paper we have chosen to study the performance of two of them, MOGLS-GP-I [13] and MOGLS-GP-J [15]. MOGLS-GP-I, is an adapted version of the first Ishibuchi proposal [13] which includes GP concepts. This approach has great part of its effort in obtaining the most extended Pareto front possible. To do so, it associates each individual of the population with weighting vector that points the direction, in the objective space, with which that individual was generated. These directions are later considered when generating new individuals.

This algorithm maintains elitism using two sets of solutions: the current population and a provisional population of non-dominated solutions. The algorithm begins generating an initial population of N_{pop} individuals, it evaluates them and updates the provisional population with the non-dominated individuals. Next, N_{pop} - N_{elite} predecessor solutions are obtained considering their direction vectors. After the crossing and mutation processes, the population will be completed with N_{elite} non-dominated individuals of the provisional population. Next, a local search (XHC) is applied to all N_{pop} individuals of the current population. In this stage, the direction vector of each individual will guide the local search process. Finally, the following generation is obtained with the N_{pop} improved individual of the actual population.

3.5 MOGLS-GP-J

In this subsection we describe an adapted version of the first Jaszkiewicz proposal [15] which use GP concepts. We call it MOGLS-GP-J. This MOGLS implements the idea of simultaneous optimization of all weighted Tchebycheff or all weighted linear utility functions by random choice of the utility function optimized in each iteration. The general idea of this MOGLS is similar to that used by Ishibuchi [13]. The main difference is in the the way that the solutions are selected for recombination. A Current Set of solutions (*CS*) is used. *CS* is initially filled up with *S* random solutions. In each iteration, the algorithm randomly draws an utility function *u*. From *CS*, *k* different solutions (with the best *u* evaluations) are selected to form a Temporary Population (*TP*). A crossover operador is applied to two randomly selected parents from the *TP*, and the new resulting solution *x* is locally optimized using the XHC local search operator. A Potentially Efficient (*PE*) set, with non-dominated solutions is updated with the new solution x. The random selection of utility functions may be seen as a mechanism that introduces some additional diversification.

3.6 Evaluation of MOEAs

In multi-objective optimization problems, the definition of the quality concept is substantially more complex than in single-objective ones, since the processes of optimization imply several different objectives. In the specialized literature several quantitative measures have been proposed [6], [11], [17], [28]. The most used is the C measure, whose expression:

$$C(A,B) = \frac{\left|\{a \in A; \exists b \in B : b \succ a\}\right|}{|A|} \tag{2}$$

measures the ratio of individuals of the Pareto A that are dominated by individuals of the Pareto B. A value of 1 indicates that all individuals of the Pareto A are dominated by individuals of the Pareto B; on the other hand, a value of 0 indicates that none of the individuals of A is dominated by individuals of B.

4 Experimental Study

The experimental study has been developed using the Cranfield collection, componed by 1398 documents about Aeronautics. The 1398 documents have been automatically indexed in the usual way, removing the stop-words, and obtaining 3857 different index terms in total. A *tf·idf* scheme[1] [22] has been used to represent the relevant index terms in the documents. Cranfield provides 225 queries, of which, those that have 20 o more relevant documents have been selected. The seven resulting queries (#1, #2, #23, #73, #157, #220 and #225) have 29, 25, 33, 21, 40, 20, 25 relevant documents associated respectively. Our MOEA-GP approaches based on the IQBE paradigm generate a set of queries from a relevant and a non-relevant documents sets. To do so, it is necessary to consider sufficiently representative number of positive examples (relevant documents), so queries with more relevant documents associated have been selected.

The studied MOEAs in this contribution have been run 30 times for each query (a total of 840 runs) with different initializations for each selected query during the same fixed number of fitness function evaluations (50.000) in a 1.5GHz Pentium Mobile computer with 2Gb of RAM. The common parameter values considered are a maximum of 19 nodes for trees, 0.8 arid 0.2 for crossover and mutation probabilities, respectively and population size of M = 800 queries. Additionally, MOGLS-GP-I and MOGLS-GP-J use an elite population size = 200, local search probability = 0.3, n_{off} = 3 and n_{it} = 10; MOGLS-GP-J use an initial population S = 1600; and SPEA2-GP use an elite population size = 200.

From each run a pareto set is obtained. The four pareto sets obtained by each run and query are compared with the performance C measure (in Table 1, average results

[1] To do so, we use the classical Salton's SMART IR [23].

of the *C* measure for each pair of MOEAs and query are showed). The experimental results show that NSGA-II, with GP concepts (NSGA-II-GP), is the IQBE MOEA technique that achieves a better performance achieves, i.e., it achieves better non-dominated solutions sets (view values in bold type-style in Table 1) in the process of learning Boolean queries in IRSs, than the other studied IQBE MOEAs.

Table 1. Average results of the *C* measure for each queries and pair of MOEAs-GP studied. W=NSGA-II, X=SPEA2, Y=MOGLS-GP-I and Z=MOGLS-GP-J.

Query	C(W,X) / C(X,W)	C(W,Y) / C(Y,W)	C(W,Z) / C(Z,W)
#1	**0.908** / 0.031	**0.803** / 0.129	**0.861** / 0.071
#2	**0.973** / 0.013	**0.902** / 0.037	**0.936** / 0.025
#23	**0.917** / 0.022	**0.895** / 0.030	**0.922** / 0.017
#73	**0.952** / 0.026	**0.943** / 0.044	**0.955** / 0.036
#157	**0.934** / 0.002	**0.925** / 0.002	**0.933** / 0.005
#220	**0.924** / 0.019	**0.917** / 0.036	**0.920** / 0.027
#225	**0.940** / 0.007	**0.909** / 0.007	**0.927** / 0.007
Query	**C(X,Y) / C(Y,X)**	**C(X,Z) / C(Z,X)**	**C(Y,Z) / C(Z,Y)**
#1	0.067 / 0.888	0.157 / 0.700	0.704 / 0.240
#2	0.068 / 0.867	0.176 / 0.681	0.675 / 0.204
#23	0.053 / 0.852	0.207 / 0.639	0.713 / 0.179
#73	0.033 / 0.869	0.156 / 0.715	0.641 / 0.207
#157	0.064 / 0.846	0.158 / 0.739	0.639 / 0.234
#220	0.054 / 0.877	0.196 / 0.693	0.698 / 0.229
#225	0.037 / 0.865	0.150 / 0.679	0.719 / 0.164

5 Conclusions

In this contribution he have presented a comparative study of performance, in the Boolean IR models context, of four of the most currently successful MOEAs in the specialized literature has been done. The studied MOEAs have been applied on the automatic learning of Boolean queries problem. The original proposals [12], [13], [15], [29] have been adapted to use GP concepts. All of them extend the Smith and Smith's IQBE EA propose [25] to work in a multi-objective context. The experimental results show that NSGA-II, with GP concepts (NSGA-II-GP), is the best IQBE MOEA technique, i.e., it achieves better non-dominated solutions sets in the process of learning Boolean queries in IRSs, than the other studied IQBE MOEAs.

In future works, we will perform a more exhaustive comparative study, using additional IQBE MOEAs and bigger database collections like TRECs.

Acknowledgments. This work has been supported by the projects: GLIRS-II, Ref. TIC-2003-07977 and Proyecto de Excelencia de la Junta de Andalucia SAINFOWEB, Cod. 00602.

References

1. T. Back, D. Fogel, and Z. Michalewicz, Handbook of evolutionary computation, IOP Publishing and Oxford University Press, 1997.
2. R. Baeza-Yates and B. Ribeiro-Neto, Modern information retrieval, Addison-Wesley, 1999.

3. G. Bordogna, P. Carrara, and G. Pasi, Fuzzy sets and possibility theory in database management systems, ch. Fuzzy Approaches to Extend Boolean Information Retrieval, pp. 231-274, Bosc, P. and Kacpryk, J., Eds. (Springer Verlag), 1995.
4. V. Chankong and Y.Y. Haimes, Multiobjective decision making theory and methodology, North-Holland, 1983.
5. H. Chen, G. Shankaranarayanan, L. She, and A. Iyer, A machine learning approach to inductive query by example: An experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing, Journal of the American Society for Information Science 49 (1998), no. 8, 693-705.
6. C.A. Coello, D.A. Van Veldhuizen, and G.B. Lamant, Evolutionary algorithms for solving multi-objective problems, Kluwer Academic Publishers, 2002.
7. O. Cordón, E. Herrera-Viedma, C. López-Pujalte, C. Zarco and M. Luque, A review on the application of evolutionary computation to information retrieval, International Journal of Approximate Reasoning 34 (2003), 241-264.
8. O. Cordon, E. Herrera-Viedma, and M. Luque, Evolutionary learning of boolean queries by multiobjective genetic programming, Proceding of PPSN-VII, Granada, Spain, LNCS 2439, Springer, 2002, pp. 710-719.
9. O. Cordón, E. Herrera-Viedma, and M. Luque, Improving the learning of boolean queries by means a multiobjective IQBE evolutionary algorithm, Information Processing & Management 42 (2006), no. 3, 615-632.
10. O. Cordón, F. Moya, and C. Zarco, A GA-P algorithm to automatically formulate extended boolean queries for a fuzzy information retrieval system, Mathware & Soft Computing 7 (2000), no. 2-3, 309-322.
11. K. Deb, Multi-objective optimization using evolutionary algorithms, Wiley, 2001.
12. K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002), 182-197.
13. H Ishibuchi and T. Murata, A multi-objective genetic local search algorithm and its application to flowshop scheduling, IEEE Transactions on Systems, Man and Cibernetics, Part C: Applications and Reviews 28 (1998), no. 3, 392-403.
14. H. Ishibuchi, T. Yoshida, and T. Murata, Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, IEEE Transactions on Evolutionary Computation 7 (2003), no. 2, 204-223.
15. A. Jaszkiewicz, Genetic local search for mult-objective combinational optimization, European Journal of Operational Research 137 (2002), 50-71.
16. A. Jaszkiewicz, Do multiple-objective metaheuristics deliver on their promises? A computacional experiment on the set-covering problem, IEEE Transactions on Evolutionary Computation 7 (2003), no. 2, 133-143.
17. J. Knowles, L. Thiele, and E. Zitzler, A tutorial on the performance assessment of stochastic multiobjective optimizers, 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, feb 2006, revised version.
18. R. Korfhage and J. Yang, Query modifications using genetic algorithms in vector space models, International Journal of Expert Systems 7 (1994), no. 2, 165-191.
19. J. Koza, Genetic programming. on the programming of computers by means of natural selection, The MIT Press, 1992.
20. D.H. Kraft, F.E. Petry, B.P. Buckles, and T. Sadasivan, Genetic algorithms and fuzzy logic systems, ch. Genetic algorithms for query optimization in information retrieval: relevance feedback, pp. 155-173, Sanchez, E. and Shibata, T. and Zadeh, L.A., Eds., (World Scientific), 1997.

21. M. Lozano, F. Herrera, N. Krasnogor, and D. Molina, Real-coded memetic algorithms with crossover hill-climbing, Evolutionary Computation 12 (2004), no. 3, 272-302.
22. G. Salton, Automatic text processing: The transformation, analysis and retrieval of information by computer, Addison-Wesley, 1989.
23. G. Salton, The smart retrieval system. experiments in automatic document processing., Englewood Cliffs: Prentince-Hall, 1997.
24. G. Salton and M.J. McGill, An introduction to modern information retrieval, McGraw-Hill, 1983.
25. M. P. Smith and M. Smith, The use of genetic programming to build boolean queries for text retrieval through relevance feedback, Journal of Information Science 23 (1997), no. 6, 423-431.
26. C.J. van Rijsbergen, Information retrieval, Butterworth, 1979.
27. W.G. Waller and D.H. Kraft, A mathematical model of a weighted boolean retrieval system, Information Processing and Management 15 (1979), 235-245.
28. E. Zitzler, K. Deb, and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary Computation 8 (2000), no. 2, 173-271.
29. E. Zitzler, M. Laumanns, and L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems (EUROGEN 2001) (K. Giannakoglou et al. (Eds.), ed.), International Center for Numerical Methods in Engineering (CIMNE), 2002, pp. 95-100.
30. E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (1999), no. 4, 257-271.