Hybrid Approaches to Mutation in Genetic Search Algorithms

Drazen Draskovic, Veljko Milutinovic
Department of Computer Science and Engineering
University of Belgrade, School of Electrical Engineering
Belgrade, Serbia
drazen.draskovic@etf.bg.ac.rs, veljko.milutinovic@etf.bg.ac.rs

Abstract - **Genetic algorithms for Internet search have been classified in a recent paper of the same authors, as indicated in the reference section of this paper. The proposed new classification was based on a survey of the open literature and recognizes four basic approaches. This paper analyses these four approaches, and discusses their potentials, especially in the domain of hybridization, i.e., in the domain of generation of new hybrid approaches that take the best of one or more of the existing approaches. Two different hybrid approaches are described and their performance potentials are discussed in a way that opens up new avenues for future research.**

Keywords: genetic algorithms, hybrid algorithms

I. INTRODUCTION

Genetic search algorithms have been analyzed and classified in [1]. The main result of this analysis and classification is presented in Figure 1, which represents the starting point of the research described here.

In this work, under the term hybrid, we imply symbiotic or synergistic efforts that take the best of two or more approaches, and generate new solutions somewhere in the space in-between. The symbiotic approach implies hybridization on a coarse grain level, while synergistic approach implies hybridization on a fine grain level.

The starting point of this paper and the related research is the coursework at the University of Belgrade, in two different areas: Artificial Intelligence and Knowledge Search [2][3][4] [5][6][7]. Among the major goals of these two courses are the efforts to teach students how to think and create in domains of interdisciplinary and multidisciplinary research, with special emphasis on intelligent systems.

The rest of this paper is organized as follows: Section 2 describes the four approaches in the four leaves of the tree-like classification of Figure 1. Section 3 defines one possible hybrid approach. Section 4 shows how the two approaches can be utilized in the context of Wikipedia, using an example. Section 5 outlines directions for future work to compare the proposed approach, comparatively with the four basic approaches - this section defines requirements for a future simulator that is supposed to shed numerical light on the proposed approach. Section 6 opens some possible avenues of future research. Section 7 concludes the paper, by restating the contribution of this paper, by defining those to whom this work is of potential interest, and by emphasizing the major directions of future research.

II. CLASSIFICATION: CRITERIA AND TAXONOMY

This research is based on two basic classification criteria: (a) Using mutational databases, and (b) Doing concept oriented analysis. These criteria were selected both because they reflect some recent and successful research efforts, and also because they leave space for effective new approaches of the hybrid type.

The secondary classification criteria differ for two different branches of the basic classification: (a) In the database branch, mutation can be based on a random approach or on a targeted approach, and (b) In the analysis branch, mutation can be based on semantic analysis or on datamining oriented analysis.

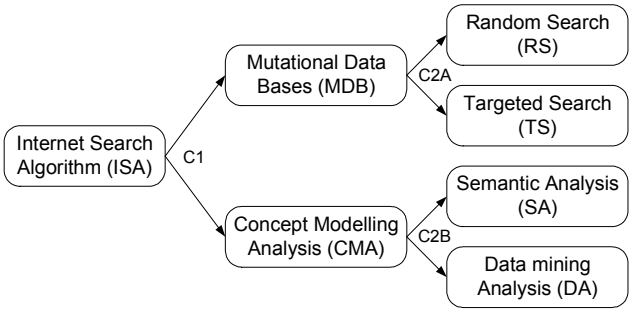

Figure 1. Classification of Internet Search Algorithms

Legend: C1 (criterion #1) = retrieval-oriented vs analysis-oriented
C2A (criterion #2, in the MDB path) = Random Search vs Targeted Search
C2B (criterion #2, in the CMA path) = Semantics-oriented vs Datamining-oriented

Figure 1 presents the proposed classification and defines the terms of interest. Further subclasses are also possible, in each one of the four major classes. Elaboration of further subclasses is a subject of a follow-up research. The proposed classification enables exiting approaches to be combined, with the goal of creating new research avenues based on hybrid solutions. In the hybrid domain, if only two approaches are combined, there are 6 different possibilities. This work tries to combine only the targeted mutation and the random mutation.

III. DEFINITION OF THE PROPOSED HYBRID APPROACH

Figure 2 represents a block diagram of the generalized structure of the random approach. Figure 3 represents a block diagram of the generalized structure of the targeted approach. Both the data flow and computation flow are described in an intuitive way, for easy understanding of the notions behind.

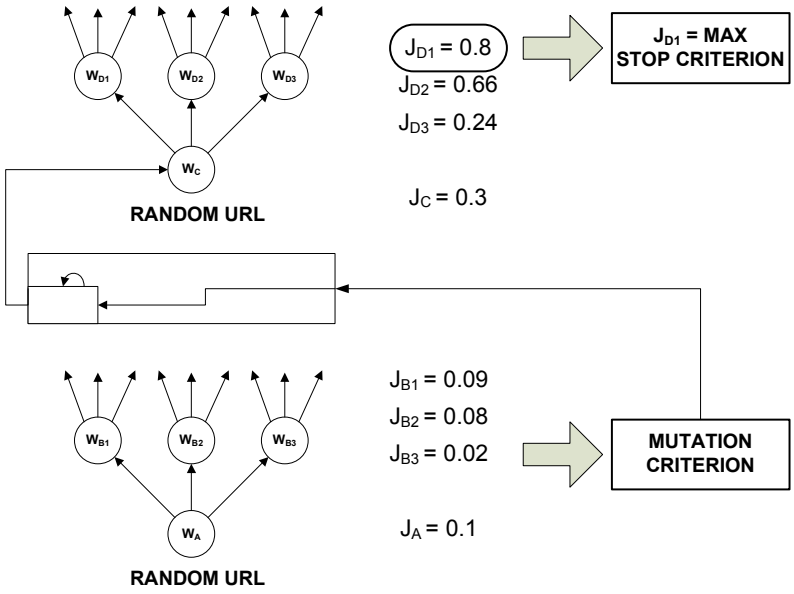

Figure 2. Random Search

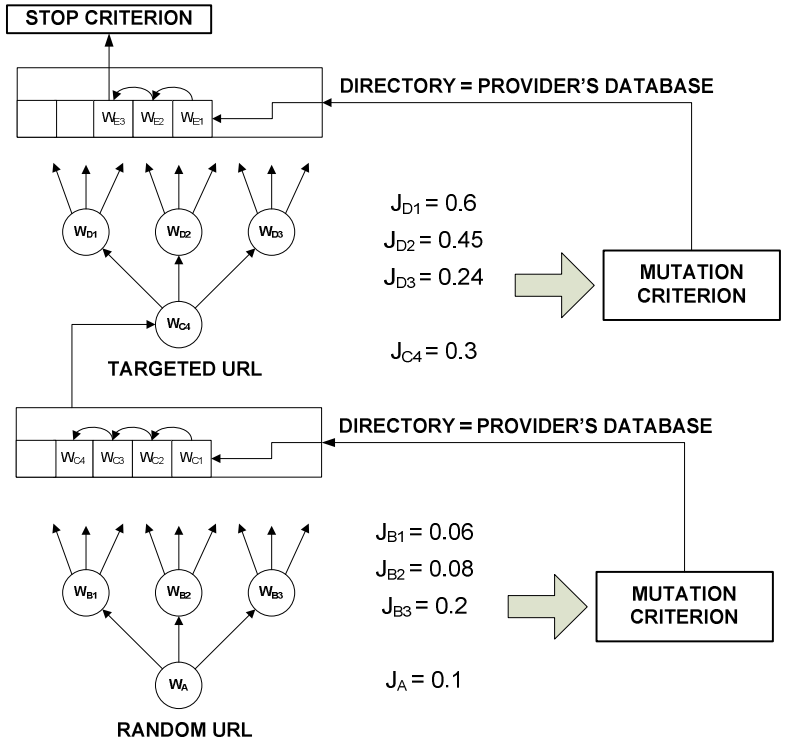

Figure 3. Targeted Search

Random search based on a mutational database was used in some early research in the field, with stress on scientific environments, and the detailed pseudo code of the approach (together with a definition of the fitness function based on Jackard's score [8]) reads as follows:

```
RANDOM SEARCH(key)
find_random_page_and_links(key)
selection_best_pages(first_page, links)
do
   find_page_in_mutational_db(key)
   selection_best_pages()
   mutation(population_and_random_pages)
while(stop_criterion==true)
return best_page
```

Targeted research with locality oriented mutation on system databases was used in some mission critical business oriented environments and is described with the following pseudo code (together with definitions of basic enabler resources):

```
TARGETED SEARCH(key)
find_random_page_and_links(key)
selection_best_pages(first_page, links)
goto directory(locality_type)
do
  find_transformed_keys_in_directory(key)
  selection_best_pages(directory)
  mutation(population_and_targeted_pages)
while(stop_criterion==true)
return best_page
```

The hybrid approach would implement both the random and the target search (only the resources which represent a minimal requirements without doubling the resources that appear both in the random and the targeted approach) and would include an additional resource that monitors the relevant parameters of the environment, like the Jackard's score, as indicated in Figure 4.

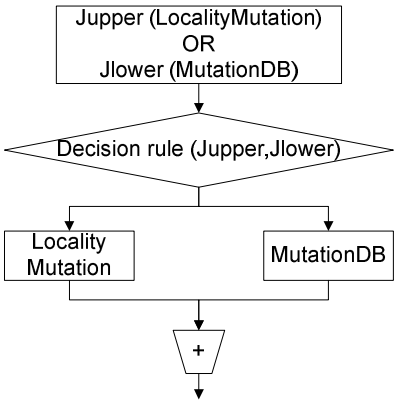

Figure 4. Criterion for mutation

The operation of the proposed hybrid approach implies the infrastructure of Figure 4, and can be defined as follows:

```
IF(JS > JSupper OR JS < JSlower)
THEN
  CALL_MUTATE_FUNCTION
ELSE
  CALL_BEST_FIRST_SEARCH_FUNCTION
WHERE
  CALL_MUTATE_FUNCTION is defined as:
IF(JS < JSlower) then
 CALL_DATABASE_MUTATION_FUNCTION
ELSE
 CALL_TARGETED_LOCALITY_MUTATION_FUNCTION
```

Since the locality mutation can be based on a number of locality related parameters (author locality, subject locality, IP locality, country locality, time locality, language locality, etc.), there are two different approaches that can be taken:

a) To select one of the possible locality type and to jump to a new web page which scores the best for that locality type (has the largest JS of all pages offered), and

b) To try all possible locality types, and to continue with the web page which scores the best for all locality types combined.

The hybrid algorithm decision rule phases:

1. Normally, the system follows the Best First Search (BFS) algorithm, which follows the links with the best Jackard's score (JS).

2. If the Jackard's score decreases below some critical value (Jlower), this is an indication that the BFS algorithm has come to a dead end, and the DM mutation is triggered.

3. If the Jackard's score increases above some critical value (Jupper), this is an indication that the BFS algorithm has come to some high quality finding, and that other even higher quality findings could exist in the nearby neighborhood, where the neighborhood could be defined via a number of different locality types, so the search has to trigger the targeted mutation according to one of the existing locality oriented mutations.

We opted for hybrid approach, because no implementation of the algorithm based on only one of four proposed classes [1] would give better results than the hybrid solution, which combines two classes. On the other hand, in contrast to the classical search algorithms in which we can utilize even some heuristic function, but in which we have to go through the search tree, in this hybrid algorithm we diminish the number of pages for which we calculate the score by utilizing crossover and mutation operations do not require passing through all the pages within the search tree and determining the score for each page (which represents the most expensive search operation), and thus we keep in population only the

pages with the best score, that is, at every moment of the operation of the algorithm we strive for the best solution.

IV. A WIKIPEDIA RELATED EXAMPLE

This section gives a simple example showing how the proposed hybrid approach can work in the Wikipedia environment.

For the example to follow, it is assumed that each national Wikipedia contains a number of domains and sub-domains, and each one can be treated as a mutational sub-database. This assumption is relevant for the database related mutation.

For the targeted locality mutation, a number of Wikipedia and XML properties [9] can be used as follows:

a) Author locality is determined by the XML field containing the author ID. If the author locality mutation is invoked, the ID of the author is determined and other web pages of the same author are fetched, and their JS is calculated. The one with the largest JS is selected to serve as the next web page to jump to (as the product of mutation).

b) Country locality can be defined as the Wikipedia system of a nearby country, in which case the next web page to fetch, after the mutation process is over, is the one with the highest JS among the web pages in the new Wikipedia system.

c) IP locality can be defined as the IP address of the Internet provider which hosts the presented page. In that case, the page which is fetched after the mutation process is completed is the one coming from another presentation of the same provider.

d) Subject locality implies that another subject is selected so that the conceptual difference of the old and new subjects is minimal, and the next page after mutation is one of the pages from the new subject group.

e) Time locality starts from the assumption that pages on the same subject generated at the same time may contain relevant information.

V. SIMULATOR REQUIREMENTS AND RESULTS

This section specifies the detailed project requirements for the team that has to implement the simulator to be used to compare the proposed hybrid approach and the existing four approaches. The text is divided into three sections: (a) Simulator functionalities, (b) Experimenting environment, and (c) Test procedures.

The major simulation parameters are:

a. The lower critical value of JS (Jlower), that triggers the switches to the DB mutation.

b. The upper critical value of JS (Jupper), that triggers the switches to the targeted mutation, according to one of the many possible locality mutations (URL locality, author locality, spatial locality, temporal locality, etc).

c. Selection of the appropriate locality mutation.

As far as the Jlower, the simulator should be able to try several values and to determine the optimal value for each particular application. As far as the Jupper, the simulator should be able to try different types of locality (author locality, IP locality, temporal locality, space locality, etc.) and to determine which one works best under given conditions.

A more detailed set of requirements for the implementation of the simulation environment is best defined after the preliminary simulation runs are performed and results are seen.

The basic step in simulator generation, Apache Lucene text search engine library [10] has been used for text analysis and Wikipedia pages index determination. One method for finding best set of documents related to the entered user keywords is described in [11]. The main purpose of the Lucene library is to retrieve pages containing at least one word from the query, and then the pages which contain the best result are obtained by hybrid genetic algorithm, as indicated in Figure 5, after a certain number of iterations.

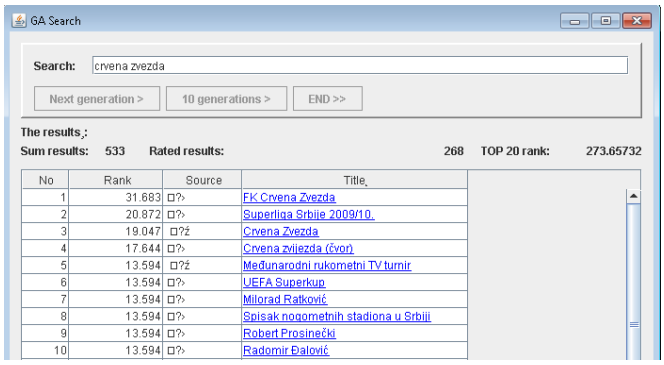

Figure 5. The simulator layout

At the beginning of the algorithm operation, the population is filled by random pages and the crossover operation is being performed in order to fetch all the pages containing links from or toward the population pages. After that, a provisional number of pages within the same category, (i.e. the same subdatabase like the population pages) has been selected, and then the mutation has been performed. After the mutation finalization, new pages, having good score, are added to the population, and the ones with the worst score have been removed. The results obtain in the course of obtaining notions in the Serbian articles subdatabase have been given in the Table 1. If the score is satisfactory, the algorithm returns the result, otherwise, there is the targeted locality mutation transition. The result of such an algorithm within the example of the Wikipedia articles Serbian subdatabase, and after utilization of country locality are given on Figure 6.

TABLE I. QUERY SEARCH RESULTS FOR "UNIVERSITY BELGRADE"

No.	Sub-database: http://sr.wikipedia.org	
	Web page (original name)	*Score*
1.	University of Belgrade	0.891
2.	University	0.494
3.	University of Novi Sad	0.427
4.	List of rectors of the University of Belgrade	0.393
5.	University of Nis	0.335
6.	School of Electrical Engineering, University of Belgrade	0.297
7.	List of universities in Serbia	0.245
8.	Dositej Obradovic	0.241
9.	University of Sidney	0.237
10.	University of Banja Luka	0.235

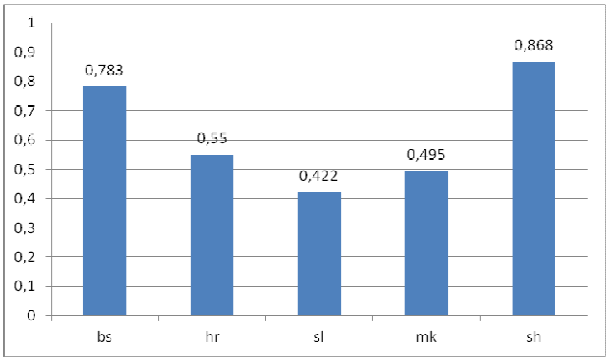

Figure 6. Query search results for country locality mutation

VI. OPEN AVENUES FOR FUTURE RESEARCH

When it comes to future research of the hybrid nature, the major challenge is to device and compare the remaining 5 hybrid approaches based on the 4 essential approaches from [1].

All the possible approaches can be compared analytically or by simulations, to see what benefits they bring. The benefits in the performance domain are obvious (combining the best characteristics of two different approaches can bring only benefits).

However, combining two approaches inevitably brings a complexity increase (both approaches plus selection/combining mechanisms require more transistor count than one isolated approaches). Consequently, the crucial issue here is the price/performance analysis.

VII. CONCLUSION

In this research, we have presented a detailed classification of algorithms for Internet search, using two basic classification criteria. The proposed classification allows the possibility of combining the defined classes of algorithms and creates opportunities for new "hybrid" algorithms. One such hybrid approach is introduced and presented using an example.

The newly open problems are: (a) To improve the performance of the proposed hybrid genetic algorithm and to find new algorithms for Web search [12], which give better results by combining the proposed classes; (b) To develop a tool that enables all examples with structured content to be compared; (c) To develop a tool that enables education of students on GA in general and MA in specific and (d) which is the main contribution of this paper, to develop original examples along all symbiotic and synergistic combinations that make sense.

The newly generated research avenues are best exploited if they build on the top of the existing research in [1]. Of the remaining five combinations of basic mutation approaches, some give hybrid algorithms that are not widely applicable, but do have a niche application of interest in some research areas.

ACKNOWLEDGMENT

This work was supported by the Ministry of Science, Republic of Serbia, grants TR32047 and III44008.

REFERENCES

[1] D. Draskovic, B. Nikolic and V. Milutinovic, "A Classification of Mutational Approaches for Genetic Search," IEEE International Conference on Industrial Technology, Athens, Greece, March 2012, pages 260-264

[2] http://www.acm.org/education/education/curric_vols/CC2005-March06Final.pdf, accessed on September 8th, 2010

[3] B. Nikolic, "Expert Systems," WUS Austria Educational Publishing and University of Belgrade, Classroom Textbook, Belgrade, June 2011, pp. 423-435

[4] V. Milutinovic, D. Cvetkovic and J. Mirkovic, "Genetic Search Based on Multiple Mutation Approaches," IEEE Computer, November 2000, vol. 33, issue: 9, pp. 118-119

[5] B. Furlan, V. Sivacki, D. Jovanovic and B. Nikolic, "Comparable Evaluation of Contemporary Corpus-Based and Knowledge-Bases Semantic Similarity Measures of Short Text," JITA, June 2011, vol. 1, no. 1, pp. 65-72

[6] V. Milutinovic, P. Knezevic, B. Radunovic, S. Casselman and J. Schewel, "Obelix Searches Internet Using Customer Data," IEEE Computer, July 2000, vol. 33, issue: 7, pp. 104-107

[7] J. Mirković, D. Cvetković, Lj. Nešić, N. Tomča, S. Cvetićanin, S. Slijepčević, V. Obradović, M. Mrkić, I. Čakulev, V. Milutinović and L. Kraus, "Genetic Algorithms for Intelligent Internet Search: A Survey and a Package for Experimenting with Various Locality Types," IEEE TCCA Newsletter, 1999, pp. 118-119

[8] V. Milutinovic, "Infrastructure for Electronic Business on the Internet," book, Springer, series: Multimedia Systems and Applications, Vol. 19, 1st edition, 2001, 464 p.

[9] Hu, M.; Lim, E.; Sun, A.; Lauw, H.W.; Vuong, B., "On Improving Wikipedia Search using Article Quality," WIDM '07 ACM International Workshop on Web information and data management, November 9, 2007, Lisboa, Portugal, on pages 145-152

[10] http://lucene.apache.org/core/, accessed on March 1st, 2012

[11] A. Al-Dallal, R. Shaker, "Genetic Algorithm in Web Search Using Inverted Index Representation," 5th IEEE GCC Conference & Exhibition, March 17-19, 2009, Kuwait City, on pages 1-5

[12] Z.Z. Nick, P. Themis, "Web Search using a Genetic Algorithm," Internet Computing IEEE, 2001, vol. 5, issue: 2, pp. 18-26