Introduction of a Sectioned Genetic Algorithm for Large Scale Problems

Zacharias Detorakis
Inst. for Language and Speech Processing
6 Artemidos & Epidavrou Str.
Paradissos Amaroussiou, 151 25, Greece
++30 210-6875363
zdetor@ilsp.gr

George Tambouratzis
Inst. for Language and Speech Processing
6 Artemidos & Epidavrou Str.
Paradissos Amaroussiou, 151 25, Greece
++30 210-6875411
giorg_t@ilsp.gr

ABSTRACT

The sectioned genetic algorithm (hereafter denoted as sectioned GA), which is presented in this paper, represents a modification of the standard GA and deals with large scale problems (i.e. problems involving pattern spaces with high dimensionalities). Instead of increasing the size of the population searching the pattern space when the problem dimensionality increases, the sectioned GA approach divides each individual into smaller parts (sections) and subsequently applies the genetic operators on each of these parts. Results from the application of sectioned GA on the problem of automatic morphological analysis are also presented in this article. Morphological analysis is by nature a large scale problem since a great number of words need to be segmented into stems and suffixes. The proposed system improves the segmentation accuracy substantially in comparison to standard GA algorithms.

Keywords

Genetic algorithms, parallel distributed algorithms, stemming, input space dimensionality, masks.

1. INTRODUCTION

Genetic Algorithms (GAs) [5] are a widely used technique for solving problems whose solutions cannot be expressed in an analytical form. Morphological analysis, aiming at the identification of stems and suffixes from a list of words, is such a problem.

Word segmentation is a key element in several information retrieval systems that search in huge databases of texts for documents relevant to a query. A suffix rarely introduces new meaning to a specific word. Therefore, when processing a query, it would be useful to include the stems of the words to broaden the search results, without deviating from the user's initial request.

Bionetics'07, December 10-13, 2007, Budapest, Hungary.

Several methods have been proposed to deal with word segmentation. These methods fall into two large categories, depending on whether they are rule-based or not. Rule-based methods utilize a series of rules to examine whether certain suffixes, from a predefined list, are suitable for the word being processed [9], [10]. These methods are language dependent, since different rules apply for different languages, and thus lack generality.

Systems that are not rule-based produce a number of solutions and subsequently choose the best one according to certain fitness criteria. Typical examples are Goldsmith's Linguistica [6], where the criterion used is the minimum description length, and the AMP [12], where the final choice is made based on the frequencies of stems and suffixes.

The method described in the present article also belongs to the category of non rule-based systems. It proposes a certain objective function that is shown by experimental results to be more efficient than other more general criteria that have been used (namely MDL – Minimum Description Length). Moreover the method addresses the task of simultaneously processing large corpora without experiencing loss in the precision accomplished or incurring an extreme increase in the execution times.

2. METHOD DESCRIPTION

The algorithm is based on maintaining an evolving population of N individuals, each corresponding to a single possible solution. The elements of an individual correspond to words extracted from a corpus (*test set*), and each word is represented by a segmentation boundary, i.e. the length of its proposed stem. Segmentation boundaries are integers ranging from zero to the length of the given word [7].

After initializing the population, the GA enters a continuous loop of applications of the genetic operators that ends only when certain termination criteria are met. During initialization, random numbers are chosen for every element of each individual. There are, however, three constraints, dictated by the Greek Language, regarding the values that randomly-generated segmentation boundaries are allowed take:

i. The segmentation boundary cannot take a zero value or a value that is equal to the word length. This guarantees that

none of the two constituents (i.e. the stem or the suffix) will be null.

ii. Certain two-letter combinations (i.e. "αι", "ει", "οι", "αυ", "ευ", "ου" and "υι") cannot be separated by a segmentation boundary [13].

iii. A potential suffix should begin with a vowel letter.

These constraints apply whenever a new random variable is chosen for the element of an individual. After the GA has settled, the segmentation boundaries provided by a specific individual propose a suitable solution.

The basic steps of a typical GA are depicted in Fig. 1.

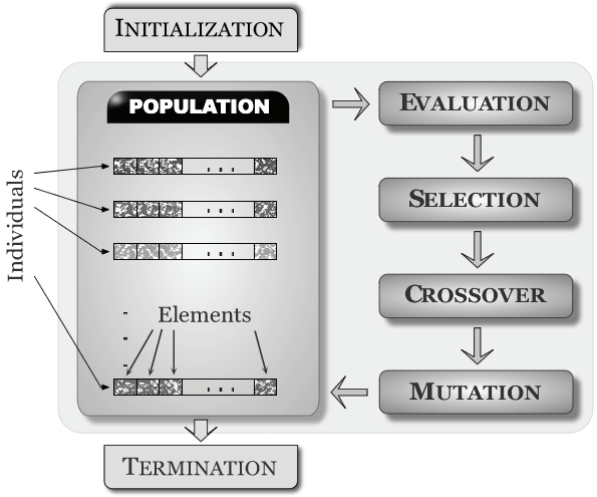

Fig. 1. Flow chart of a typical GA

2.1 Evaluation

After a population has been formed, it is evaluated according to a fitness function to associate fitness values to the individuals, indicating their quality as possible solutions to the problem. This fitness function defines the environment in which the population develops and encompasses all the knowledge that derives from the study of the problem. The function proposed here is based on an a-priori given set of correctly segmented words and is a variation of the function originally proposed in [3].

The fitness function proposed in [3] calculated a histogram that recorded suffixes as well as their frequencies of appearance in the a-priori given set, referred to as *training set*. When evaluating an individual from the GA population, a similar histogram was formed based on the proposed segmentation of the words from the corpus. The expected macroscopic resemblance of those two histograms was used by the system as a fitness function to guide them towards the optimum solution. That fitness function produced better results than the MDL when tested for a number of different corpora.

The variation of the fitness function that is used in this paper lies in the way that the training set is utilized. Each word from the corpus to be processed is associated with the word from the set of correctly segmented words that mostly resembles it with respect to the matching last letters of the two words. This way, a revised and more suitable set is created containing entries from the original a-priori set, which now serves as the new training set.

The mathematical formula of the fitness function is expressed in (1):

$$Grade_{hist} = \sum_{i=1}^{s} \left[f\_train_i - \left| f\_train_i - \tilde{f}\_test_i \right| \right] \quad (1)$$

where s is the number of suffixes present in the training set, f_train_i is the frequency of the i[th] suffix in the training set and $\tilde{f}\_test_i$ is the normalized frequency of the same i[th] suffix in the test set. The normalized frequency is given by:

$$\tilde{f}\_test_i = f\_test_i \Big/ \sum_{i=1}^{s} f\_test_i \quad (2)$$

were f_test_i is the frequency of the i[th] suffix as recorded in the corresponding histogram.

It is obvious from (1) and (2), that only suffixes existing in the training set participate in the evaluation. If a suffix recognized in the training set is not present in the test set, then its corresponding frequency $\tilde{f}\_test_i$ is set to zero. The reason for using the normalized values instead of the originals is that a large test set (larger than the training set) will most likely encompass a greater range of suffixes. Therefore the original frequencies will span over a larger ensemble, leading to smaller individual values.

2.2 Selection

When all individuals have been evaluated, pairs of parents need to be selected in order to combine and create offspring. N pairs of parents are selected, with N being the number of individuals in the population, and each pair produces two complementary offspring [11].

In nature, fitter individuals are more probable to recombine and pass on their good characteristics to the next generation. The probability Pr_k of an individual being selected as parent is therefore linked in the GA to its evaluation grade as follows:

$$Pr_k = Grade_k \,/\, \sum_{j=1}^{N} Grade_j \quad (3)$$

According to this probability and using a roulette wheel selection scheme, the N pairs are formed.

2.3 Crossover

Each pair of parents produces two complementary offspring by recombination of the parents' elements. A uniform crossover scheme [8] is used where the elements are swapped individually between the parents. When an element from one parent is transferred to one offspring, then the corresponding element of the other parent will pass on to the second offspring.

Uniform crossover is more appropriate for the particular problem because there is no apparent link between neighboring elements and therefore no reason for swapping them in groups. In fact, experimental results have indicated that using other techniques like one-point or two-point crossover results in a slower progress through the iterations.

The 2N individual offspring need to be reduced to N to substitute the parent generation on a one-to-one basis. In fact, because the best individual of a generation is directly transferred to the next generation, the offspring need to be reduced to N-1. The process of preserving the best individual is called elitism [1]. To implement the reduction, all offspring are evaluated according to the fitness function and then sorted in descending order of fitness. The N-1 first individuals of this sorted list will substitute the previous parents in the new generation.

2.4 Mutation

Before the substitution is made, the offspring are mutated by changing the values of a very small percentage of their characteristics. Mutation is intended to introduce to randomly selected elements new values that probably did not exist in the original population, thereby allowing a broader search toward an optimum solution. The new values for the elements chosen to be mutated need not only be different from their previous ones but also be in compliance with the three constraints mentioned earlier.

Though mutation addresses the need to escape from local optima, it shouldn't be applied to a large extent because in that case it might lead to a purely random search in the pattern space [14]. For the present application, mutation probability has been fixed to 0.02%.

After mutation is complete, the offspring substitute their parents and, together with the best individual of the previous generation, they form the new population. This marks the start of a new iteration that follows the exact same steps. After a certain number of iterations, the algorithm comes to an end with the best individual of that last population representing the final solution.

3. INTRODUCTION OF THE SECTIONED GA APPROACH

The population of a GA has two dimensions: the individual's size, that is the number of elements of the individual, and the population size, that is the number of individuals constituting the population. These two dimensions are closely linked since, as the size of an individual grows, so must the population size. Otherwise, by keeping the population size fixed while the individuals grow, the GA would fail to approach the ideal solution before settling, since it would sample a very small portion of the pattern space.

To avoid premature convergence and to accelerate the algorithm, the parallel distributed GA model (pdGA) [2] divides the population in several isolated subpopulations (referred to as islands), which, for most of the time, run independently from one another. A certain migration rate is set so that individuals from one subpopulation can interact with other subpopulations at fixed time intervals and exchange their genetic characteristics.

The number of elements constituting the individual equals the number of distinct words in the corpus and is therefore defined by the problem. Since large corpora contain a great number of distinct words, both dimensions of the corresponding populations will have to take very large values.

The solution proposed in this paper is intended to keep the one dimension that is not outwardly defined (i.e. the population size) small, without degrading the performance of the GA. In contrast to pdGA, this approach uses one population instead of many sub-populations and always operates on that. If a pdGA were to be applied to the specific problem, it would either have to split the existing population into smaller subpopulations or create several populations of the same size as the existing one. In the first case, taking into account the large size of each individual, the smaller subpopulations would converge even faster than the one in the standard GA, potentially resulting in sub-optimal solutions. In the second case, the pdGA would increase the size of the overall population, prolonging the execution time, which is exactly what the GA variant proposed in this paper attempts to avoid.

3.1 Static Masks

All individuals of the population are segmented into smaller identically-sized *sections* with the help of a group of mutually exclusive masks that collectively cover the entire set of elements. These masks hide all other elements except for the ones included in their associated section. Fig. 2 illustrates an example of an individual comprising 12 elements that is segmented in 4 parts, each of which consists of 3 elements.

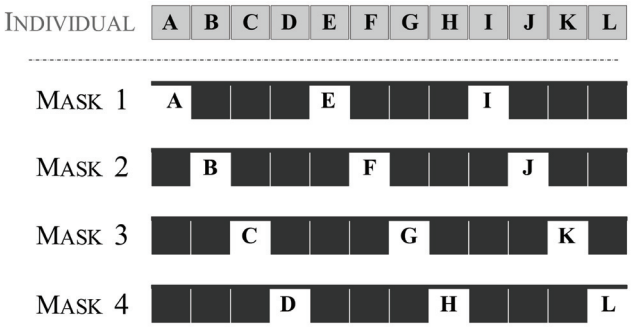

Fig. 2. Segmentation of an individual in four sections by a group of static masks

After creating the group of masks, the genetic algorithm performs its operations on the smaller sections instead of on the whole individuals. In the example depicted in Fig. 2, each of the smaller sections has 3 instead of 12 elements and therefore the population size doesn't have to be large.

When processing a mask, all hidden elements of the individuals do not participate in the GA steps and are not affected by the genetic operators. That way, each individual is evaluated according only to the limited number of elements that are visible following the mask application. Also, the offspring substitute only the particular elements in the parents' individuals for each mask, so crossover and mutation affect just the reduced section of the individual.

Every mask is processed for a certain number *m* of iterations (this being set to 10 in our experiments), before the GA shifts to the next mask. After all masks have been processed, all the elements of the individuals have been updated, so a new circle of *m* iterations is initiated where the masks are yet again applied in succession to the population. Since the individuals in each population are sorted in descending fitness order, it is guaranteed that the best individuals according to one mask will have the highest fitness in accordance to all masks.

3.2 Dynamic Masks

In the approach described up to this point, the masks are created at the beginning of the GA simulation and remain static throughout all iterations. Therefore the elements don't have the opportunity to interact with other elements that belong to different masks. The introduction of dynamic masks overcomes this obstacle, by allowing for a recombination of the elements in the smaller sections after the completion of a predefined number of iterations. Figure 3 illustrates the notion of the dynamic masks that are updated every *m* iterations.

It should be noted that the masks depicted in Fig. 2 are static. Hence, the same spatial pattern could be used in each mask, being simply displaced by a different amount. On the contrary, in Fig. 3, the pattern of each mask is random (under the condition that all elements are covered exactly once), with the patterns of masks being altered at regular intervals.

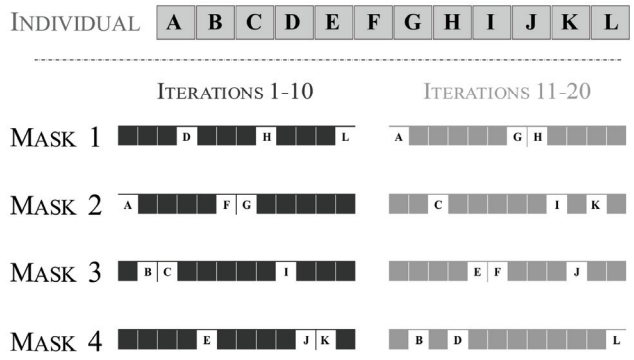

Fig. 3. Example of application of dynamic masks updated every 10 iterations

The use of masks (either static or dynamic) has an additional beneficial effect on the implementation of the sectioned GA. Different masks from a group may run in parallel in different processors (PCs) and only need to interact after a certain number of *m* iterations to change the mask(s) they are assigned to and share the solutions found so far with the other processors. In this intuitive manner, the GA takes a distributed form that can further reduce the execution time.

4. EXPERIMENTS

The sets of words used in the GA for training and testing are extracted from various Greek corpora, corresponding to a number of registers such as the literary and journalistic registers. Each word is inserted to the set once, at the point of its first appearance, and is thereafter ignored whenever encountered in the texts. Since words are inserted to the set in order of appearance, less frequent words are more likely to be situated near the end of the set.

The ideal segmentation, used as reference, is that provided by the ILSP morphological lexicon [4], which has been created to a large extent manually by linguists, over a period spanning several years.

4.1 Standard versus Sectioned GA

This series of experiments examines the method proposed in this paper for the problem at hand against the traditional GA approach. For this reason both versions of the GA – standard and sectioned – were applied to various sizes of corpora and the results were compared. For the sectioned GAs the sections created by the static masks have a size of 1,000, while the population size is 250 individuals in both the sectioned and standard GAs.

Figure 4 illustrates comparatively the performance of the two types of GAs. A more detailed presentation of the results is provided by Table I that presents the initial and final values of segmentation precision in all cases as well as the execution time required for 50 iterations.

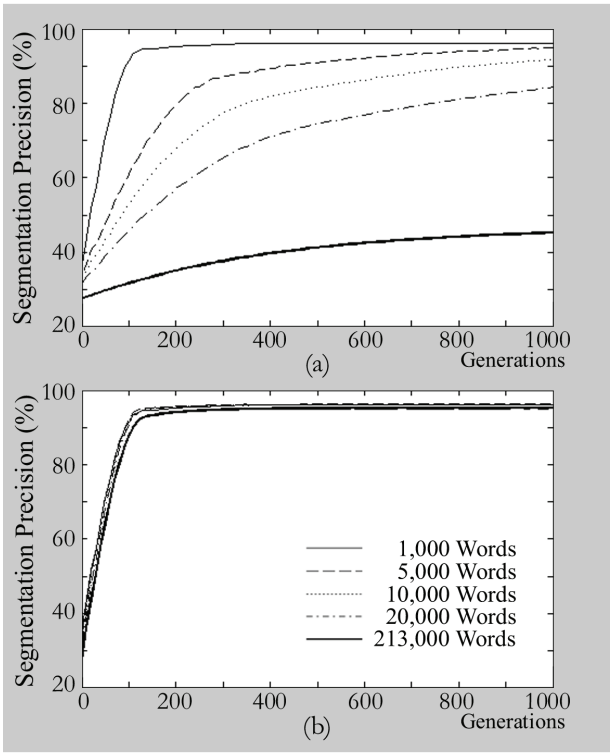

Fig. 4. Segmentation precision of (a) the standard GA and (b) the sectioned GA for various sizes of individuals.

The algorithm execution times depicted in Table I have been obtained using a PC with a single Intel Pentium processor operating at a frequency of 3.4 GHz. The initial precision is that of the best individual of the first population, which has been created by randomly choosing segmentation boundaries that adhere to the constraints described in Section 2.

Table I. Detailed Comparative Results of the Standard and the Sectioned Genetic Algorithms

Section size	GA-type	Initial Precision	Final Precision	Execution Time (secs)
1,000	Standard GA	36.0	96.0	113.7
	Sectioned GA	36.0	96.0	115.5
5,000	Standard GA	33.6	94.8	406.3
	Sectioned GA	33.6	96.3	356.4

10,000	Standard GA	32.8	91.8	769.1
	Sectioned GA	32.8	96.1	668.5
20,000	Standard GA	31.4	84.2	1517.6
	Sectioned GA	31.4	96.1	1323.3
213,000	Standard GA	27.5	45.2	18480.3
	Sectioned GA	27.5	95.2	16665.5

Figure 4a illustrates the fact that the individuals' size (i.e. the number of words to be processed) affects not only the performance of the system but also the number of iterations needed for the GA to settle. This observation is even more evident as the individuals' size increases, and, in the extreme case of 213,000 words, the segmentation precision reaches less than 50% of the precision accomplished with the sectioned GA. Even though, in the experiments conducted with more than 1,000 words, the standard GA has not fully settled in the first 1,000 iterations, data from further iterations indicate that the maximum precision accomplished after settling is always below that of the sectioned GA.

In contrast to the simple GA, the sectioned GA seems to be independent of the individual's size (Fig. 4b) and thus is much more robust. The experimental data for one prove that regardless of the size, every sectioned GA has settled to a final segmentation solution after 500 iterations. It is therefore safe to set a termination point after the completion of 1,000 iterations. Moreover, Table I indicates that the final segmentation precision accomplished by the sectioned GA is virtually the same for all numbers of words.

The small drop of 1% that occurs in the case of 213,000 words is attributed to the nature of the words being processed rather than to the algorithm itself. When processing larger sized sets, less frequent words are almost certainly included in a greater percentage. This is a result of the way in which the test set is created with the insertion of words according to their order of appearance in the texts. Less frequent words tend to have a more difficult morphology and are harder to analyze, thus leading to a small decline in the segmentation accuracy.

The success of the sectioned GA resides in that it deals with smaller groups, instead of trying to find appropriate values for all variables simultaneously. It is always easier to determine the relation between a small amount of parameters (in our experiments 1,000 words) rather than larger sets of parameters (for instance the full set of 213,000 words). Moreover, the individual size poses no real limitations to the proposed method since a potential increase would only be translated into the formation of additional sections (and the generation of the corresponding masks).

The last column of Table I indicates that the execution time of the sectioned GA is slightly smaller, even when executed on a single processor. This represents an additional bonus to the fewer iterations needed for the sectioned GA to settle, in comparison to the standard GA.

Even if the population size grew (in accordance to the rate of growth of the individuals), the results of the standard GA would still be poor. Figure 5a illustrates the performance of a standard GA with individuals of 10,000 elements for various population sizes. As a comparison, Figure 5b depicts the results from the same experiments when the sectioned GA is used.

The growth of the population size helps the standard GA to settle after a smaller number of iterations and achieve results equal to those of the sectioned GA. However, there seems to be a limit as to how fast the standard GA will settle and moreover the growth of the population comes at the expense of substantially longer execution times.

On the other hand, the sectioned GA seems to be independent of the population size (as well as of the size of the individuals) and performs similarly in all cases. This supports the initial assumption that when using the proposed structure for the GA even small population sizes can be adequate.

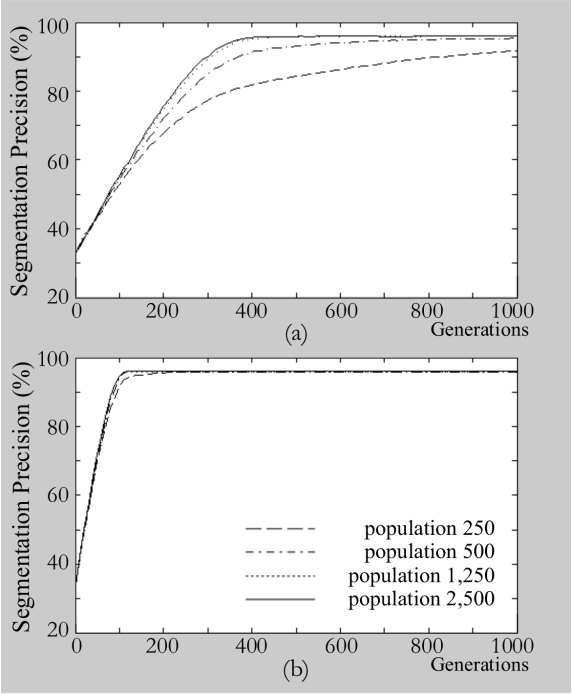

Fig. 5. Performance of (a) the standard GA and (b) the sectioned GA for various population sizes.

4.2 Static versus Dynamic Masks

As mentioned in paragraph 3, when a sectioned GA operates on static masks, parameters from one sub-section don't have the chance to interact with parameters from other sub-sections. For the application on morphological analyses and the particular objective function used, this might not pose such an important problem, since every word is independent of its environment.

However, when using dynamic masks the sub-sections are periodically updated following the completion of a certain number of iterations. The elements that constitute each sub-section are chosen randomly and thus after a large number of

iterations each particular word will have combined with almost all words in the training set, thus allowing the optimisation of the final solution. To compare the effectiveness of using static versus dynamic masks, a set of simulations has been performed, where the size of the smaller sections in the experiments conducted remains 1,000 in both cases.

Figure 6 indicates that the segmentation error in the case of 213,000 words is reduced by 11% when dynamic masks are applied (the error falling from 4.84% to 4.29%). Even though the problem parameters are independent of each other, the dynamic mask still enhances the GA's performance.

Apart from the recombination between different elements that promotes the exchange of genetic material, dynamic masks introduce a certain degree of randomness much like the migration in the pdGA, which makes the sectioned GA more stochastic and thus more likely to locate the optimal solution.

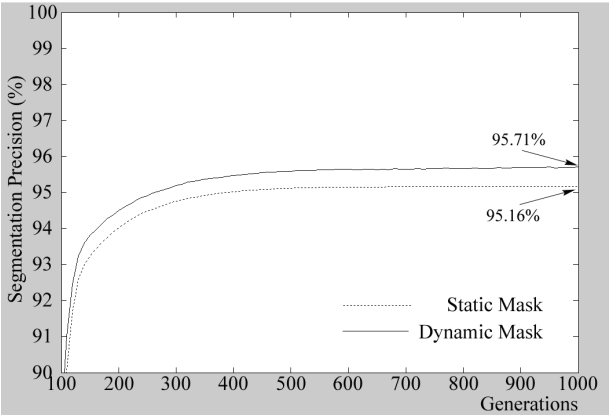

Fig. 6. Segmentation precision of sectioned GAs when utilizing a static and a dynamic mask.

5. CONCLUSIONS

The experiments reported in this article indicate that the introduction of a sectioned structure on the individuals leads to superior performance in relation not only to the algorithm's speed but also to the achieved segmentation precision. This sectioned structure can be readily implemented in GA applications involving a great number of variables, in order to maintain a fairly small population without affecting the system's performance. In fact, experimental data prove that the GA might perform even better.

Moreover, it has been shown that the concept of GAs can be successfully applied to the task of word segmentation. The system generates a segmentation accuracy of up to 96%, which is more than adequate for an application focusing on data mining. Even though the system described is language-dependent in terms of a small number of explicitly provided linguistic rules, as the segmentation boundaries adhere to restrictions that have been extracted from the study of the Greek language, results achieved even in the absence of those restrictions indicate that the system could be satisfactorily applied to other languages

6. ACKNOWLEDGMENTS

The authors would like to thank Ms. M. Vassiliou of the ILSP for her valuable help providing insights on the study & characteristics of the Greek language. This research has been supported by the PENED programme 03ED97, funded by the Greek Secretariat for Research and Technology.

7. REFERENCES

[1] Ahn, C.W., and Ramakrishna, R.S. Elitism-Based Compact Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*, *7*, 4 (2003), 376-385.

[2] Cantú-Paz, E. A survey of Parallel Genetic Algorithms. *Calculateurs Paralleles, 10*, 2 (1998), 141-171.

[3] Detorakis, Z., and Tambouratzis, G. Implementation of a Multi-Objective Genetic Algorithm on Word Segmentation in Modern Greek. *In Proc. of the 11th IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2007)*, Mallorca, Spain, 29-31 August, ISBN 978-0-88986-693-5.

[4] Gavrilidou, M. The ILSP morphological lexicon and morpho-syntactic tagger. *Internal report. Athens: Institute for Language & Speech Processing (in Greek)*, 1996.

[5] Goldberg, D. E. *Genetic algorithms in search, optimisation and machine learning*, Addison-Wesley, Boston, 1989.

[6] Goldsmith, J. Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics, 27*, 2 (2001), 153-193.

[7] Kazakov, D., and Manandhar, S. A hybrid approach to word segmentation. *In Proc. of the 8th International Workshop on Inductive Logic Programming (ILP-98)*, Madison, Wisconsin, USA (July 22-24), Springer, 125-134.

[8] Lin G. and Yao X. Analysing crossover operators by search step size. *In Proc. of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, Indianapolis, USA (April 13-16), IEEE Press, New York, 1997, 107-110.

[9] Ntais, G. *Development of a Stemmer for the Greek Language*, Master Thesis, Department of Computer and Systems Sciences, KTH-Stockholm University, 2006.

[10] Porter, M. F. *An algorithm for suffix stripping*, Program, *14,* 3 (1980), 130-137.

[11] Srinivas, M., and Patnaik, L.M. Genetic Algorithms: A Survey. *IEEE Computer, 27*, 6 (1994), 17-26.

[12] Tambouratzis, G., and Carayannis, G. Automatic Corpora – based Stemming in Greek. *Literary and Linguistic Computing*, *16*, 4 (2001), 445-466.

[13] Triantafilidis, M. *Modern Greek Grammar*, Institute M. Triantafilidis, 1941.

[14] Vasconcelos, J.A., Ramirez, J.A., Takahashi, R.H.C., and Saldanha, R.R. Improvements in genetic algorithms. *IEEE Transactions on Magnetics, 37*, 5, part 1 (2001), 3414 - 3417.