

Optimization Algorithms in Function of Binary Character Recognition

Petar Čisar¹, Sanja Maravić Čisar², Dane Subošić¹, Predrag Đikanović³, Slaviša Đukanović³

¹Academy of Criminalistic and Police Studies, Cara Dušana 196, 11080 Zemun, Serbia, petar.cisar@kpa.edu.rs, dane.subosic@kpa.edu.rs

²Subotica Tech, Marka Oreškovića 16, 24000 Subotica, Serbia, sanjam@vts.su.ac.rs

³Republic of Serbia, Ministry of Interior, Kneza Miloša 101, 11000 Belgrade, Serbia, predrag.djikanovic@mup.gov.rs, slavisa.djukanovic@mup.gov.rs

Abstract: The paper gives an analysis of some optimization algorithms in computer sciences and their implementation in solving the problem of binary character recognition. The performance of these algorithms is analyzed using the Optimization Algorithm Toolkit, with emphasis on determining the impact of parameter values. It is shown that these types of algorithms have shown a significant sensitivity as far as the choosing of the parameters was concerned which would have an effect on how the algorithm functions. In this sense, the existence of optimal value was found, as well as extremely unacceptable values.

Keywords: binary character recognition; Optimization Algorithm Toolkit; hill climbing; random search; clonal selection algorithm; parameter tuning

1 Introduction

In different areas of mathematics, statistics, empirical sciences, computer science, mathematical optimization makes up the selection of the prime components, according to given requirements from the series of possible alternatives [1]. An optimization problem involves finding the minimum and maximum of a real function by systematically changing the input values from within a pre-defined set and calculating the corresponding values of the function. Taking a wider approach, it can be stated that optimization comprises the detection of extreme values of a particular function concerning a specific domain, including various types of objective functions, as well as assorted types of domains.

Character recognition is a process by which a computer recognizes different characters (letters, numbers or symbols) and turns them into a digital form that a

computer can use; also known as optical character recognition (OCR). OCR is the mechanical or electronic transformation of scanned or photographed images of written or printed characters into a form that the computer can manipulate. It has been researched in detail and usually focuses on scanned documents. Since the foreground and background are easily segmented, high accuracy has been obtained for binary character recognition. In recent years, more and more OCR requests have come from camera-based and web images. In these images, characters usually have no clear foreground or clean background. Furthermore, degradations such as blurring, low resolution, luminance variance, complicated background, distortion, etc. often appear. All these factors contribute to a great challenge in the recognition of the character of natural scene images.

The majority of character recognition methods managed to improve traditional OCR. Weinman *et al.* [2] proposed a scene text recognition method. His emphasis was to combine lexicon information in order to improve scene text recognition result. Yokobayashi *et al.* [3] utilized a GAT (Global Affine Transformation) correlation method to improve the tolerance of binary character recognition to the distortions of scaling, rotation and shearing. GAT method was much time-consuming. De Campos *et al.* [4] utilized a bag-of-visual-word based method to recognize the character of a given natural scene. Six local descriptor-based character features were compared. Although some of them proved to be preferable to traditional OCR, the results still showed the difficulty in this field.

Besides the OCR applications, one can use binary images with great success with various applications given the lower computational complexity solutions, some embedded systems, autonomous robots, intelligent vehicles, using different algorithms such as hybrid evolutionary algorithm and so on.

The authors of this work will analyze the impact of parameters of several optimization algorithms on accuracy in solving the binary character recognition problem. An analysis of optimization algorithms is performed in the Optimization Algorithm Toolkit (OAT) environment. The OAT is a workbench and toolkit for developing, evaluating and experimenting with classical and state-of-the-art optimization algorithms on standard benchmark problem domains [5]. This open source software contains reference algorithm implementations, graphing, visualizations and other options. The OAT offers a functional computational intelligence library so as to investigate algorithms and problems at hand, while also making use of new problems and algorithms. A plain explorer and experimenter graphical user interface is placed on top of this in order to ensure a fundamental comprehension of the library's functionality. Non-technical access is ensured by the graphical user interface for the configuration and the visualization of existing methods on standard benchmark problem examples [5], [16].

For testing the quality of binary character recognition the following optimization algorithms from the OAT were used: parallel mutation hill climber and random search.

The paper consists of six sections. The introduction offers a terminological basis for the problem of character recognition, then in the second section an overview of hill climbing optimization is presented. The clonal selection algorithm is described in the third section, followed by the fourth section dealing with random search optimization. The results of the accuracy examination of analyzed optimization algorithms in binary character recognition and its various aspects in an adequate software environment are given in the fifth section. Finally, this paper closes with the conclusion based on the analyzed cases.

2 Hill Climbing

Hill climbing in mathematics is an optimization method categorized under local search. Further, hill climbing algorithm is known as the discrete optimization algorithm [6]. This algorithm belongs to the category of heuristic search as it uses a heuristic function to compute the next state. Hill climbing algorithm works on the basis of choosing the nearest neighbor as it computes a state that is better than other succeeding states at its current position. Therefore, it is also treated as a greedy algorithm.

Hill climbing is a suitable local optimum detection (a solution not improvable by considering a neighboring value) but there is no certainty to define the best possible solution (the global optimum) [17]. The characteristics that are provided only by the local optima can be improved by using restarts (repeated local search), or more complex iterative schemes (iterated local search), on memory (reactive search optimization and tabu search), or memory-less stochastic modifications (simulated annealing).

The implementation of this algorithm is rather straightforward, thus ensuring its popularity [17]. It is used widely in the area of artificial intelligence, for reaching a goal state from a starting node. While more advanced algorithms, including the simulated annealing and tabu search, might yield preferable results, with specific cases it is hill climbing that will prove to be a better choice. Hill climbing could frequently bring a better result as opposed to different algorithms in the case when there is only a limited period of time available for performing such a search (which is a common real situation). It is an anytime algorithm: it is able to offer a valid solution even if there is an interruption before the end.

Mathematical description [7] - Hill climbing tries to maximize (or minimize) a function $f(x)$, where x represent discrete states. Such states are usually signified by vertices in a graph, in which edges define nearness or similarity of a graph. Hill climbing traces the graph from vertex to vertex, it will consistently locally increase (or decrease) the value of f , until reaching a local maximum (or local minimum) x_m . Further, hill climbing can operate on a continuous space: should

that be the case, the algorithm is known as the gradient ascent (or gradient descent if the function is minimized).

Drawbacks occurring in the application of this algorithm are [7]:

- **Local maxima** - One of the problems regarding hill climbing is that the scope of detection is limited to local maxima. Assuming that the heuristic is not convex, the global maximum may not be reached. Various different local search algorithms will attempt to circumvent this issue (stochastic hill climbing, random walks and simulated annealing). A possible solution to the current problem of hill climbing is the implementation of random hill climbing search techniques.
- **Ridges** - A ridge can be defined as a curve in the search space leading to a maximum, while the ridge's orientation – in comparison with the available moves that are used to climb – ensures that every move leads to a smaller point. To sum it up, every point on a ridge appears to the algorithm as a local maximum, despite the fact that the given point may form part of a curve which leads to a better optimum.
- **Plateau** - A further problem that sometimes occurs with hill climbing is the issue of a plateau. One is faced with a plateau if the search space is flat - a path where the heuristics are situated in great proximity to each other. In the given cases, it may not be possible for the hill climber to determine the direction of the next step, this might end up wandering in a direction not leading to improvement.

The hill climbing algorithm is not a complete algorithm as it does not always give the result even if it exists [6]. The algorithm may fail if it finds any of these problems. It may choose a wrong path resulting in a dead end.

Hill climbing may be implemented with any problem of choice provided that the current state will allow for an accurate evaluation function. As instances for hill climbing one may list the problem of the traveling salesman, the eight-queen problem, circuit design, and a variety of other real-world problems. Inductive learning models have also implemented hill climbing. One specific instance is PALO, a probabilistic hill climbing system which provides a model of inductive and speed-up learning. Certain implementations of this system have been embedded into “explanation-based learning systems”, and “utility analysis” models [8].

Another field of application of hill climbing is robotics, namely to manage multiple-robot teams. A particular case would be the Parish algorithm, which enables scalable and efficient coordination in multi-robot systems. Their algorithm makes it possible for robots to opt for working alone or in teams by using hill-climbing. Robots executing Parish are thus “collectively hill-climbing according to local progress gradients, but stochastically make lateral or downward moves to help the system escape from local maxima” [9].

The procedure of parallel hill climbing is a hill climbing scheme initiated several instances at a time, this results in multiple hills being claimed in parallel.

It may be argued that the most wide-spread use of the stochastic hill climbing algorithm is by Forrest and Mitchell [10], who posited the Random Mutation Hill Climbing (RMHC) algorithm in a paper analyzing the behavior of the genetic algorithm on a deceptive class of bit-string optimization problems.

3 The Clonal Selection Algorithm¹

Artificial Immune Systems (AIS) are adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving [11].

As it is emphasized in [16], the clonal selection approach inspired the development of the AIS which executes optimization and pattern recognition problems. This initiates from the antigen controlled maturation of B-cells, with associated hyper mutation process. The B-cell is a type of white blood cell and, more specifically, a type of lymphocyte. These immune systems also implement the concept of memory cells to continue to provide excellent ways to solve the problem in question. De Castro and Timmis [11] focused on two vital characteristics of affinity maturation in B-cells. The first, being that the increase of B-cells is in proportion to the affinity of the antigen binding it. Therefore, the greater the affinity, the greater numbers of clones are created. Furthermore, the mutations resulted through the antibody of a B-cell are inversely proportional to the affinity of the antigen it binds. De Castro and Von Zuben [12] developed and frequently used the clonal selection based-AIS (CLONALG) by implementing these two features, which has been applied for executing pattern matching and multi-modal function optimization tasks.

The general principle of functioning of the clonal selection algorithm is presented in Figure 1 [16].

¹ Section 3 draws on material [16] published in Acta Polytechnica Hungarica. Vol. 11, No. 4, 2014

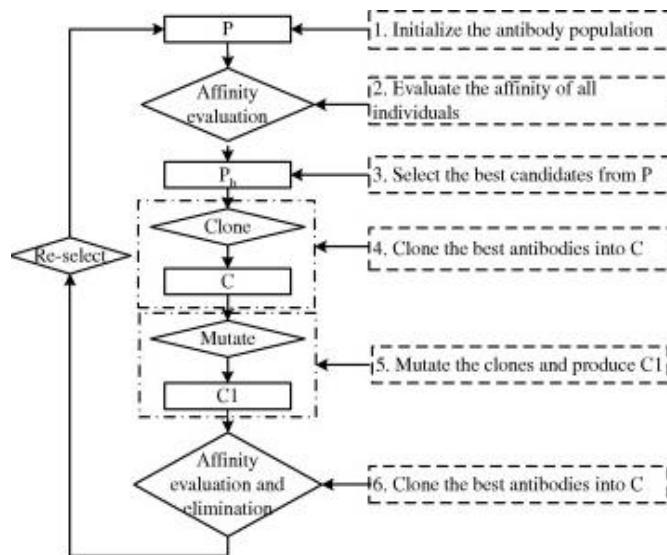


Figure 1

Steps of clonal selection method [13]

4 Random Search

Random Search (RS) represents an optimization method with the aim to detect the optimum by testing the objective function's value on a series of combinations of random values of the adjustable parameters. The random values are generated in compliance with certain boundaries as chosen by the user, moreover, the system excludes any combinations of parameter values failing to fulfill the constraints on the variables. In fact, it can be stated that the method can handle bounds on the adjustable parameters and fulfill constraints. For an infinite number of iterations it is guaranteed that the present method will detect the global optimum of the objective function. Generally, one is interested in processing a very large number of iterations [14]

Options for RS [14]:

- Number of iterations - this parameter is a positive integer to determine the number of parameter sets to be drawn before the algorithm stops. The default value is usually 100000.
- Random number generator - the parameter determines the random number generator, which is to be used with this method.
- Seed - the parameter is a positive integer value with the aim of determining the seed for the random number generator.

The name, random search, is attributed to Rastrigin [15] who made an early presentation of RS along with basic mathematical analysis.

Classification of RS [18]:

- Random jump - generates a huge number of data points assuming uniform distribution of decision variables and selects the best one.
- Random walk - generates the trial solution with sequential improvements using scalar step length and unit random vector.
- Random walk with direction exploitation – is the improved version of random walk search, the successful direction of generating trial solution is found out and steps are taken along this direction.

5 Binary Character Recognition – Results

The learning and memory acquisition of optimization algorithms within OAT is verified through its application to a binary character recognition problem (Fig. 2). The aim is to examine the accuracy of binary character recognition for the analyzed algorithms: random search and parallel mutation hill climber.

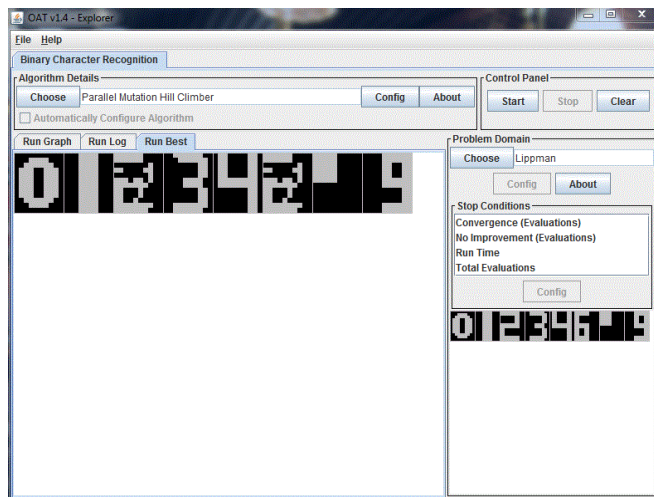


Figure 2

OAT environment for examination of binary character recognition

The parallel mutation hill climber algorithm (combined CLONALG – hill climber algorithm) - In the present instance, it can be supposed that the antigen population is signified by a set of eight binary characters to be learned. Each character is represented by a bitstring of length $L = 120$. The antibody configuration is

composed of 10 individuals, 8 of them in the memory set M. The initial parameters (set by OAT) are the following: mutate factor = 0.00833, population size = 16.

For different values of the population size (for the constant mutate factor), the output results (memory sets) are shown in Figure 3.

Similar to the previous case, with the choice of different values of the mutate factor, the resulting situation is shown in Figure 4. (the initial parameters being: mutate factor = 0.00833 (variable), population size = 30 (constant)).

Having in mind Figure 3 and Figure 4, two conclusions can be made. Through, increasing the number of population, with a fixed mutate factor, the quality of binary character recognition is also increased. In addition to this, the impact of the mutate factor (with a fixed population size) is not directly proportional. This practically means that it is necessary to examine several different values in order to obtain the best accuracy of recognition – with finding the optimal value.











Memory set	Population size
	Input patterns
	20 generations
	25 generations
	30 generations
	35 generations
	40 generations

Figure 3

Input and output patterns in function of the population size

Output	Mutate factor
	0.00833
	0.01
	0.02
	0.05





	0.055
	0.06
	0.07
	0.1

Figure 4

Output patterns in function of the mutation factor

After thoroughly analyzing the graphical results above, it can be concluded that the parallel mutation hill climber algorithm is very sensitive to the choice of parameters. This, in fact, means that the performance of this algorithm must definitely be checked with several different parameters, and based on the obtained, the best value can be chosen. This procedure is known in the literature as tuning parameters and directly affects the accuracy and the speed of convergence of the algorithm (e.g. solving time) to the solution. Also, it can be noticed that, due to the CLONALG component, this algorithm maintains a diverse set of local optimal solution, which can primarily be explained by the selection and reproduction schemes. Another important characteristic is the fact that CLONALG also considers the cell affinity, which corresponds to the fitness level of a given individual so as to determine the rate of mutation used for every member of the population.

In the case of a random search, the result is presented in Figure 5. For this type of algorithm, there are no configurable parameters and the quality of binary character recognition is very low.



Figure 5

Output patterns in the case of a random search

Conclusions

The practical part of the paper is focused on examining the impact of parameters of analyzed optimization algorithms on the accuracy of binary character recognition. In this sense, the existence of optimal value was found, as well as the extremely unacceptable values. This in fact means that great attention must be paid to the selection of parameter values (it is not to be disregarded which values are applied) and it is vital to test the behavior of the algorithm in practice with several different parameters. Moreover, it was confirmed that all of the optimization algorithms are not equally suitable for binary character recognition.

Namely, it has been shown that some of the analyzed algorithms did not generate an acceptable result.

References

- [1] The Nature of Mathematical Programming. Mathematical Programming Glossary. INFORMS Computing Society, at <http://glossary.computing.society.informs.org>
- [2] J. J. Weinman, E. Learned-Miller, A. R. Hanson: Scene Text Recognition Using Similarity and a Lexicon with Sparse Belief Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 31(10), pp. 1733-1746, 2009, doi:10.1109/TPAMI.2009.38
- [3] M. Yokobayashi, T. Wakahara: Binarization and Recognition of Degraded Characters Using a Maximum Separability Axis in Color Space and GAT Correlation. in *Proc. of 18th Int. Conf. on Pattern Recognition (ICPR2006)* Vol. 2, 2006, pp. 885-888, doi:10.1109/ICPR.2006.326
- [4] T. De Campos, M. Barnard, K. Mikolajczyk, J. Kittler, F. Yan, W. Christmas, D. Windridge: An Evaluation of Bags-of-Words and Spatio-Temporal Shapes for Action Recognition. in *Proc. WACV. 2011*, pp. 344-351
- [5] Optimization Algorithm Toolkit, at <http://optalgtoolkit.sourceforge.net/>
- [6] R. Agrawal: A Modified Hill Climbing Algorithm. *International Journal of Emerging trends in Engineering and Development*. Issue 2, Vol. 6, 2012, pp. 663-667
- [7] Hill Climbing Methods, at <http://www.scribd.com/doc/7290255/Hill-Climbing-Methods>
- [8] W. Cohen, R. Greiner, D. Schuurmans, W. Cohen: Probabilistic Hill-Climbing. In S. Hanson, T. Petsche, R. Rivest, M. Kearns eds. *Computational Learning Theory and Natural Learning Systems*. Vol. II, MITCogNet, Boston: 1994
- [9] B. P. Gerkey, S. Thrun: Parallel Stochastic Hill-climbing with Small Teams. in L. E. Parker et al. (eds.) *Multi-Robot Systems: From Swarms to Intelligent Automata*. Volume III, Springer, 2005, 65-77
- [10] S. Forrest, M. Mitchell: Relative Building-Block Fitness and the Building-Block Hypothesis. In D. Whitley (ed.) *Foundations of Genetic Algorithms 2*, Morgan Kaufmann, San Mateo, CA, 1993
- [11] L. N. de Castro, J. Timmis: *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer. 2002 Edition, pp. 57-58
- [12] L. N. de Castro, F. J. Von Zuben: Learning and Optimization using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation*. 2002 Jun, 6(3):239-251

- [13] I. Aydin, M. Karakose, E. Akin: Chaotic-based Hybrid Negative Selection Algorithm and its Applications in Fault and Anomaly Detection. *Expert Systems with Applications*. Volume 37, Issue 7, pp. 5285-5294, 2010
- [14] COPASI Development Team, at <http://www.copasi.org>
- [15] L. A. Rastrigin: The Convergence of the Random Search Method in the Extremal Control of a many Parameter system. *Automation and Remote Control*. 24 (10), 1963, pp. 1337-1342
- [16] P. Cisar, S. Maravic Cisar, B. Markoski: Implementation of Immunological Algorithms in Solving Optimization Problems. *Acta Polytechnica Hungarica*. Vol. 11, No. 4, 2014
- [17] Hill Climbing, at <http://stackoverflow.com/tags/hill-climbing/info>
- [18] Advanced Topics in Optimization, Direct and Indirect Search Methods, at http://nptel.ac.in/courses/Webcourse-contents/IISc-BANG/OPTIMIZATION METHODS/pdf/Module_8/M8L4slides.pdf