TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

School of Mathematics, Statistics and Computer Science
Computer Science

Object Detection using Neural Networks and Genetic Programming

Barret Chin and Mengjie Zhang

Technical Report CS-TR-07/03
November 2007

School of Mathematics, Statistics and Computer Science
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

School of Mathematics, Statistics and Computer Science
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Object Detection using Neural Networks and Genetic Programming

Barret Chin and Mengjie Zhang

Technical Report CS-TR-07/03
November 2007

Abstract

This paper describes a domain independent approach to the use of neural networks (NNs) and genetic programming (GP) for object detection problems. Instead of using high level features for a particular task, this approach uses domain independent pixel statistics for object detection. The paper first compares an NN method and a GP method on four image data sets providing object detection problems of increasing difficulty. The results show that the GP method performs better than the NN method on these problems but still produces a large number of false alarms on the difficult problem and computation cost is still high. To deal with these problems, we develop a new method called *GP-refine* that uses a two stage learning process. The results suggest that the new GP method further improves object detection performance on the difficult object detection task.

Keywords Object detection, genetic programming, neural networks, region refinement, feature selection

Author Information

Barret Chin is a Postgraduate student in Computer Science, and Mengjie Zhang is staff members in Computer Science. Both authors are with School of Mathematics, Statistics and Computer Science, Victoria University of Wellington, New Zealand.

Object Detection using Neural Networks and Genetic Programming

Barret Chin and Mengjie Zhang

School of Mathematics, Statistics and Computer Science
Victoria University of Wellington, P. O. Box 600, Wellington, 6140, New Zealand
Email: {chinbarr,mengjie}@mcs.vuw.ac.nz

Abstract. This paper describes a domain independent approach to the use of neural networks (NNs) and genetic programming (GP) for object detection problems. Instead of using high level features for a particular task, this approach uses domain independent pixel statistics for object detection. The paper first compares an NN method and a GP method on four image data sets providing object detection problems of increasing difficulty. The results show that the GP method performs better than the NN method on these problems but still produces a large number of false alarms on the difficult problem and computation cost is still high. To deal with these problems, we develop a new method called *GP-refine* that uses a two stage learning process. The results suggest that the new GP method further improves object detection performance on the difficult object detection task.

1 Introduction

Object detection is the task of finding objects of interest in large images. The process involves both sub-tasks of object classification and object localisation. Object classification refers to the task of distinguishing between images of different kinds of objects where each image contains only a single object image and the idea is to categorise these images into classes or groups. Object localisation refers to the task of determining the positions of all of these objects of interest in large images. Object localisation is often regarded as object detection, where all objects of interest need to be detected from the large images and object locations reported but the kinds of objects do not need to be distinguished. This is actually a binary classification problem, where one class is for the objects of interest and the other class is for the background, but the number of the objects of interest is usually very small compared with the number of pixels on the background. Object detection has many applications ranging from detecting clones in a set of satellite images to finding tumours in a set of X-ray images and finding a particular human face from a set of images containing human photographs.

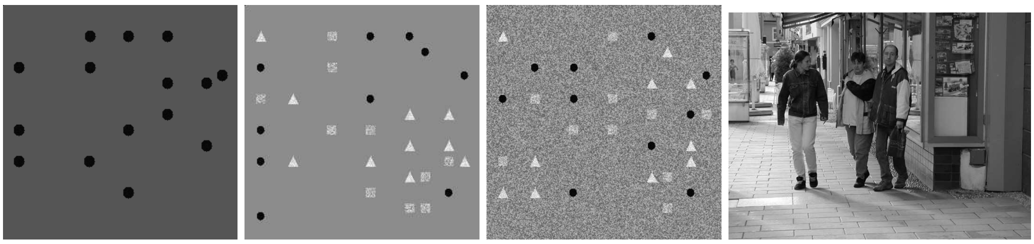
Neural networks (NNs) and genetic programming (GP) are two powerful paradigms in computational intelligence [1, 2]. Since the 1990s, NNs and GP have been used in many object classification and detection tasks [3–9]. In most existing approaches, high level features are used as inputs to NNs and genetic programs. While such approaches have achieved some success, they often involve a time consuming investigation of important specific features and a hand crafting of feature extraction programs. Another problem of using these two methods for object detection is that they often produce a large number of false alarms.

The goal of this paper is to investigate a domain independent approach to the use of NNs and GP for object detection problems. Instead of using raw pixels as inputs to neural networks and genetic programs, this approach uses domain independent pixel statistics for object detection to avoid the problem of too large architectures of neural networks and too large size of evolved programs. The two methods will be examined on four image data sets providing object detection problems of increasing difficulty. Specifically, we are interested in:

- Whether NNs and GP can achieve good performance on the four problems and which one is better on these problems.
- Whether and how the object detection performance can be improved using a set of good features automatically selected by GP.

The rest of the paper is organised as follows. Section 2 describes the four image data sets. Section 3 briefly describes the NN and GP methods for object detection and presents the results. Section 4 describes a new method using GP and presents the results with discussions. Section 5 draws the conclusions and gives future work directions.

2 Image Data Sets



No. of images: 10 Object size:14×14 (Trivial)	No. of images: 10 Object size:14×14 (Easy)	No. of images: 10 Object size:14×14 (Medium)	No. of images: 22 Object size:80×80 (Hard)

Fig. 1. Object Detection Problems

Four different data sets providing object detection problems of increasing difficulty were used in the experiments. Figure 1 shows example images for each data set. The three shapes data sets were generated to simulate a particular obstacle in object detection. Data set 1 (Trivial) is a trivial problem containing only one type of shape (black circles with very little Gaussian noise) against a uniform background. Data set 2 (Easy) introduces three different types of shape objects (black circles, grey squares and white triangles with Gaussian noise) on a relatively uniform background, so the detection program will need to regard all the three types of objects as a single type. This makes the detection problem harder than that in data set 1, but can be still considered easy. In data set 3 (Medium), the objects are still similar to those in data set 2, but the background is very noisy and in fact the intensities of the grey squares are very similar to the background, making the detection problem even harder. Data set 4 (Hard) consists of a set photographic images taken from the PASCAL Object Recognition Database collection [10]. The detection problem in this data set is very hard as there exist many different types of backgrounds as well as faces (objects) of varying shapes and sizes. We do not expect our approach to achieve perfect results, but would like to know how well our approaches can perform on such a difficult problem.

In the three shape data sets, 300 "objects" were cut out from the images in each data set, 150 for the objects of interest and 150 for background samples. For the hard face data set, 106 objects were cut out from the images where half were viewable faces and half were background samples.

3 The Baseline Approach: NNs vs GP

This section describes the overall baseline approach to object detection with NNs and GP first, then describes related aspects of the NN and GP methods, followed by the results with discussions.

3.1 Overall Baseline Object Detection Approach

The baseline object detection approach has a training process for learning a good classifier from the object cutout image examples and a testing process for object detection in the large images to detect objects of interest using the learned classifier as the detector. The approach is outlined as follows.

1. Assemble a database of images in which the locations and classes of all the objects of interest are manually determined. These full images are divided into two sets: a training set and a test set.
2. Determine an appropriate size of n×n square which covers a single object of interest and form the input field. A classification image data set is to be created by cutting out squares of size n×n from the training set. Each object cutout image either contains a single object or a background example.
3. Use either NNs or GP to learn a classifier that can well separate the object cutouts examples from the background examples in the classification data set.
4. The trained classifier (either a trained NN or an evolved genetic program) is then used as a detector, in a moving window template fashion, to locate the objects of interest in the full images in the test set. An object is reported based on the network activation values (the NN method) or the output value of the genetic program (GP method).
5. Measure the object detection performance by calculating the detection rate and false alarm rate (number of false alarms per object) in the test set.

3.2 The NN Method

In the NN method, we used the three-layer feed forward network architecture. The number of input nodes is the number of image features and the number of output nodes is 2 (one for object and the other for background). The number of nodes in the single hidden layer was determined based on an empirical search through experiments. The networks were trained by the back propagation algorithm [11] without momentum. The winner-takes-all strategy was applied to the activation values of the output nodes for classification: the class with the larger value is considered the class of the object.

Region Features. To meet the requirements of domain independence, we used low level pixel statistics as image features. As shown in figure 2, we consider six regions (the whole window, the central square, and the four rectilinear regions) from which the mean and standard deviation pixel statistics are extracted as the features. This gives a total number of 12 features.

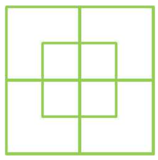


Fig. 2. Rectilinear regions.

Parameters. The network architectures and main parameter values used for the four data sets are shown in table 1. These parameter values were obtained by an empirical search via experiments. A network architecture of 12-8-2 means that there are 12, 8, and 2 nodes in the input, hidden and output layers, respectively. A critical error of 0.01 means that the network training will be terminated when the mean squared error reaches 0.01.

Table 1. Parameters used for NN training for the four databases.

	Trivial	Easy	Medium	Hard
Learning Rate	0.01	0.001	0.001	0.01
Critical Error	0.01	0.08	0.01	0.01
Random Range	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]
Net. Arch.	12-8-2	12-3-2	12-8-2	12-12-2

3.3 The GP Method

In the GP method, we used tree structures to represent genetic programs[12, 13]. The ramped half-and-half method was used for generating the programs in the initial population. The proportional selection mechanism and the reproduction, crossover and mutation operators were used in the learning process.

The Primitive Sets. The terminal set used in GP consists of the 12 features extracted from the six regions described earlier and a random constant. The function set consisted of the four standard arithmetic operators and a conditional operator: $\{+, -, *, /, if\}$. The $+$, $-$, and $*$ are the usual addition, subtraction and multiplication operators, where each operator has their usual meaning. However the / represents "protected" division which has the same meaning as the normal division except that divide by zero gives a value zero. Each of these operators takes two arguments. The `if` operator takes three arguments. The first argument, which can be any expression, constitutes the condition. If the first argument is positive, the `if` function returns its second argument; otherwise, it returns its third argument.

The Fitness Function. The GP method used the classification accuracy as the fitness function. A genetic program produces a floating point number where this output value determines whether an input field in the large test image contains an object or not. If the output of a genetic program is positive, then the input field is considered to contain an object; otherwise, it is considered background.

Parameters and Termination Criteria. The important parameter values used in this paper are shown in table 2. These parameter values were obtained via an empirical search to seek good results. The learning/evolutionary process is terminated at a pre-defined maximum number of generations unless a successful solution is found, in which case the evolution is terminated earlier.

Table 2. Parameters used for GP training for the four databases.

Parameter Kind	Parameter Name	Trivial	Easy	Medium	Hard
Search Parameters	Population Size	500	3330	500	3000
	Max Depth	6	6	6	6
	Max Generations	50	50	50	200
Genetic Parameters	Reproduction Rate	10%	10%	10%	10%
	Cross Rate	60%	60%	60%	60%
	Mutation Rate	30%	30%	30%	30%

For both the NN and GP methods, the experiments were repeated 20 times and the average results were presented in the next sub section.

3.4 Results and Discussion

Table 3. Object detection results using the baseline approach.

		Image data sets			
		Trivial	Easy	Medium	Hard
Best Detection Rate(%)		100	100	100	100
False Alarm Rate (%)	NN	0	66.67	253.33	3300
	GP	0	0	0	250

The object detection results for both the NN and GP methods are shown in table 3. Both methods achieved 100% detection rate on all the four data sets, meaning that all the objects of interest were successfully detected from the large images. They both achieved ideal performance on the Trivial data set, reflecting the fact that the detection problem in that data set is straightforward.

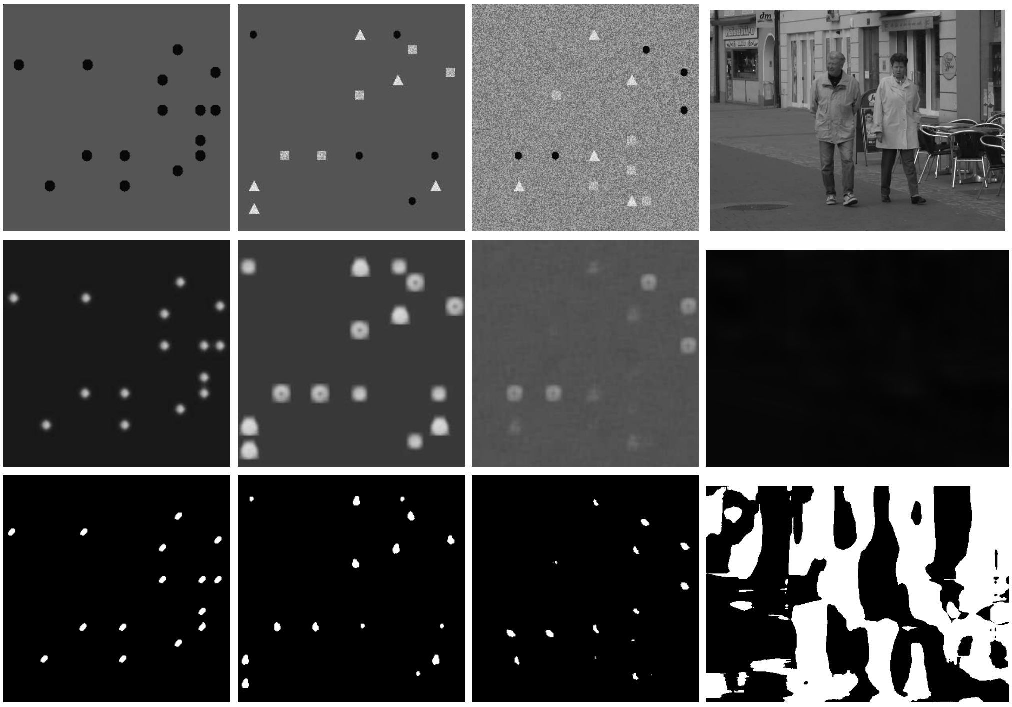


Fig. 3. GP Sweeping Maps using Pixel Statistics

On the other three data sets, while the NN method produced a large number of false alarms, the GP method did much better particularly for the Easy and the Medium data sets where ideal results were also achieved, suggesting the GP method is better than the NN method for these problems.

To further understand the differences of the detection performance, figure 3 shows the example images in the test set with their object sweeping maps for the two methods. The sweeping maps were produced by the object detection process. If there is no match between a square input field in an image and the template (either a trained neural network or an evolved genetic program), the centre of the input field will be black in the sweeping maps. A partial match corresponds to grey on the centre of the object, and a good match is close to white. The object sweeping mps can be used to get a qualitative indication of how accurate the object detection step is likely to be. In figure 3, the three rows are the original test images, the sweeping maps produced by the NN method, and sweeping maps generated by the GP methods. The four columns correspond to the four data sets respectively. The sweeping maps in the first three columns are highly consistent with the detection results. Those for the Hard face data set are a bit strange to human eyes: for the NN method, we can only see dark grey (very close to black) patches. In this case, many positions are either weakly considered objects (producing many false alarms) or the actual face objects are missing (false negatives). This sweeping map suggests that the NN method did badly for the Hard face data set. The sweeping map on this data set generated by

GP is also bad, but not as bad as that produced by the NN method. These maps confirm that the GP method is better than the NN method for these problems.

Although the GP method achieved better results than the NN method, there are still two key limitations. The first is the object detection performance for the Hard Faces data set still did not reach an acceptable level. The second is the computational costs for the evolutionary process and object detection were still quite high. The next section address these limitations by introducing an initial feature refinement phase to the GP method.

4 GP-Refiner

During the experiments of the baseline GP method, we found that the evolutionary process generated many genetic programs that do not use all of the 12 available features and that some features were chosen multiple times. This implies that some features are more important than others for classifying and detecting objects of interest from the background. Since each pair of features represents a local region, certain regions would be more important than others for a particular task. This would make sense: for example, a human could distinguish a face from an image if he/she found some particular regions/features such as eyes, noses and mouths.

Further inspection of the evolved programs generated by the baseline GP method reveals that there were some very complex combinations of the different features in the programs. We suspect that one of the reasons behind this is that the six regions used earlier were too abstract and the use of more regions could help the selection of local region features for object classification and detection.

Based on the two observations, we proposed a new method called *GP-Refiner* to address the limitations of the baseline GP method. We expect that the new method can achieve better performance on the Hard face data set and that the evolutionary learning can generate shorter program classifier/detector for object detection. This section describes the new method and results.

4.1 The GP-refine Method

The main idea of the new method is to use more local region features as terminals and introduce a "feature selection/refinement" phase into the evolutionary process for training the classifier in the GP method. So the new GP-refine method is a two-phase approach.

The first phase is called the *feature refinement* phase, where all the local region features are used as the terminal set and the GP evolution is performed over the classification data set just as in the baseline GP method. In this phase, the best programs at all generations are recorded and the statistical usage frequency of the individual features is reserved. Based on this information, top ten features are chosen to form a new terminal set. We hypothesised that these features are more important than other features and they can do a good job for object detection.

The second phase is the same as the evolutionary training phase in the baseline GP method except that only the refined features selected from the first phase are used in the new terminal set. Using a smaller terminal set, together with a small program depth limit allowed to grow during evolution, the training process testing process would be more efficient. Since the calculation of the "redundant" features from the input field in the large testing images will be omitted, the object detection process would also be more efficient.

The Features and Regions. As described earlier, this method require features that represent exclusive local regions. Therefore we introduced two new sets of local regions from which the features were extracted to form the terminal set for the feature refinement phase. Figure 4 show the two sets of regions. In both sets, mean and standard deviations of each local region will be used as the features, so a total number of 18 and 32 local region features will construct the terminals sets respectively.

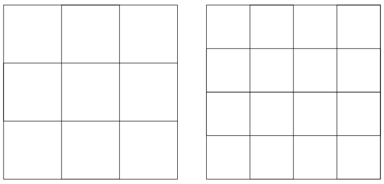


Fig. 4. Nine regions (left) and Sixteen regions (right).

Parameters and Termination Criteria. The GP system in the two phases used the same parameter values and termination criteria as the baseline GP method described in the previous section, except that in the second phase, the population size was 500 for all the four data sets, the maximum program depth was reduced to 5, and the number of maximum generations was reduced to 50 for all cases. The main consideration was that the terminal set in the second phase is smaller and good solutions could be found using a smaller population, program depth and number of generations.

4.2 Results and Discussion

Table 4 shows the average detection results of the new GP-refine method with the two sets of region features. Compared with the results (in table 3) obtained by the baseline GP method, the new GP-refine method achieved worse results with nine region features but achieved better overall results when 16 local regions were used in the feature refinement phase. While the results for the medium data set obtained by the new GP method with 16 regions were slightly worse than the baseline GP method, the difference was really small. In particular, it achieved much better detection results on the Hard face data set. This suggests that with sufficient number of local region features, the new GP-refine method can successfully select the important features for a particular task and achieve better results for object detection. We also observed that the computation costs for evolutionary learning and object detection testing were reduced by about

30–35% compared with the baseline GP method (details are not shown due to page limit). Another observation is that the new GP-refine method with the two local region sets achieved better results than the baseline NN method.

Table 4. Object detection results using the GP-Refine method.

		Image data sets			
		Trivial	Easy	Medium	Hard
Best Detection Rate(%)		100	100	100	100
False Alarm Rate (%)	9-regions	0	33.33	86.67	1050
	16-regions	0	0	6.67	50

To give an intuitive view of the object detection performance, the object sweeping maps produced by the new GP-refine method with the 9-regions (up row) and the 16-regions (down row) for the four data sets are shown in figure 5. Compared with the sweeping maps shown in figure 3, these sweeping maps clearly show that the new method with 16-regions achieved better detection performance, particularly for the Hard face data set.

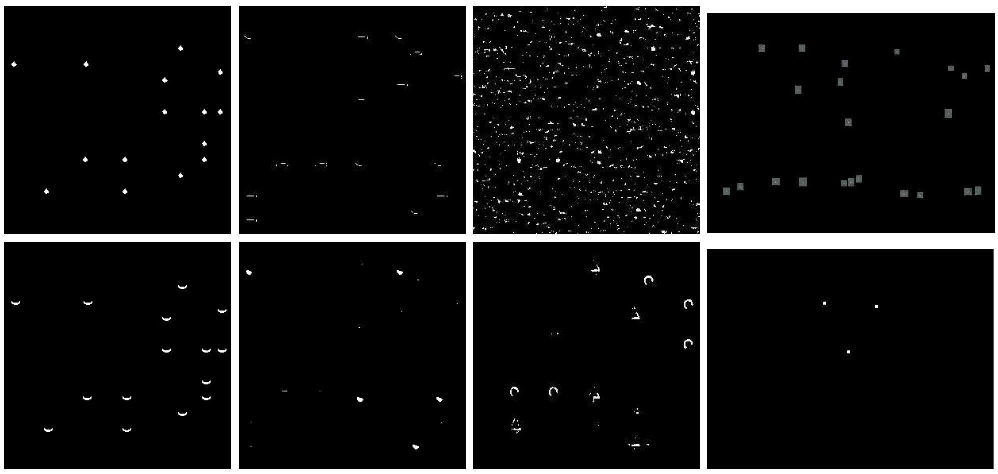


Fig. 5. Object sweeping maps generated by the GP-refine method: 9-regions (up); 16 regions (down).

5 Conclusions

The goal of this paper was to develop a domain independent approach to object detection using neural networks and genetic programming. The goal was successfully achieved by using domain independent low-level pixel statistics, investigating a baseline NN method and a GP method, and developing a new GP-refine method tested on four object detection problems of increasing difficulty. The results show that the baseline GP method performed better than the NN method on these problems but still produced a large number of false alarms on the difficult problem and computational cost was still high. The new GP-refine method achieved better object detection performance and the computational cost was also reduced over the baseline NN and GP methods.

There are a number of interesting points derived from this work, including which regions are more important than others for a particular task, and what types of features are more important for specific tasks. Our initial analyses show that the trained object classifier favoured the use of standard deviations over the mean features when the detection problem got more difficult, but further investigation is needed in the future.

References

1. Rumelhart, D.E., McClelland, J.L., the PDP research group: Parallel Distributed Processing, Explorations in the Microstructure of Cognition. The MIT Press, Cambridge, Massachusetts, London, England (1986) Volume 1: Foundations.
2. Koza, J.R.: Genetic programming : on the programming of computers by means of natural selection. Cambridge, Mass. : MIT Press, London, England (1992)
3. Shirvaikar, M.V., Trivedi, M.M.: A network filter to detect small targets in high clutter backgrounds. IEEE Transactions on Neural Networks **6** (1995) 252–257
4. Casasent, D.P., Neiberg, L.M.: Classifier and shift-invariant automatic target recognition neural networks. Neural Networks **8** (1995) 1117–1129
5. Lin, Y.Y., Liu, T.L., Fuh, C.S.: Fast object detection with occlusions. Technical report, Institute of Information Science, Academia Sinica and National Taiwan University, Department of CSIE (2004)
6. Roberts, M.E., Claridge, E.: Cooperative coevolution of image feature construction and object detection. Technical report, University of Birmingham, School of Computer Science (2004)
7. Howard, D., Roberts, S.C., Brankin, R.: Target detection in SAR imagery by genetic programming. Advances in Engineering Software **30** (1999) 303–311
8. Song, A., Loveard, T., Ciesielski:, V.: Towards genetic programming for texture classification. In: Proceedings of the 14th Australian Joint Conference on Artificial Intelligence, Springer Verlag (2001) 461–472
9. Zhang, M., Andreae, P., Pritchard, M.: Pixel statistics and false alarm area in genetic programming for object detection. In Cagnoni, S., ed.: Applications of Evolutionary Computing, Lecture Notes in Computer Science, LNCS Vol. 2611, Springer-Verlag (2003) 455–466
10. : The pascal object recognition database collection. http://www.pascal-network.org/challenges/VOC/databases.html (2007)
11. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In Rumelhart, D.E., McClelland, J.L., the PDP research group, eds.: Parallel distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations. The MIT Press, Cambridge, Massachusetts, London, England (1986)
12. Koza, J.R.: Genetic Programming II: Automatic Discovery of Reusable Programs. Cambridge, Mass. : MIT Press, London, England (1994)
13. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic Programming: An Introduction on the Automatic Evolution of computer programs and its Applications. San Francisco, Calif. : Morgan Kaufmann Publishers; Heidelburg : Dpunkt-verlag (1998) Subject: Genetic programming (Computer science); ISBN: 1-55860-510-X.