การรู้จักภาพโดยใช้ Genetic Algorithm

Pattern Recognition using Genetic Algorithm

สุรพงศ์ เอื้อวัฒนามงคล*
Surapong Auwatanamongkol,Ph.D

บทคัดย่อ

Genetic Algorithm (GA) มีประสิทธิภาพในการแก้ปัญหาการวิเคราะห์ทางเลือกที่ดีที่สุด และปัญหาปัญญาประดิษฐ์ ซึ่งมีการประยุกต์ใช้ในด้านต่าง ๆ รวมถึงการรู้จำรูปแบบ อย่างไรก็ตาม การใช้งาน GA ในด้านการรู้จำรูปแบบถูกจำกัดอยู่เพียงแต่ในการฝึกเครือข่ายใยประสาทเท่านั้น การวิจัยนี้ได้ศึกษาการใช้ GA ในการจับคู่รูปแบบโดยตรง โดยใช้ GA เพื่อหาคู่ที่ดีที่สุดระหว่างปม (nodes) ของรูปแบบ 2 รูปแบบ ในการนี้ สมการวัตถุประสงค์ถูกระบุในรูปของความแตกต่างทั้งหมดของมุมระหว่างขอบเคียงกันของรูปแบบทั้งสอง ในการทดลองเพื่อประเมินผลของแนวทางดังกล่าวพบว่าสามารถแบ่งกลุ่มรูปแบบต่าง ๆ ได้อย่างถูกต้องเป็นที่น่าพอใจ

* Assistant Professor, Department of Computer Science School of Applied Statistics National Institute of Development Administration Bangkok, Thailand 10240,
Email : surapong@as.nida.ac.th

Abstract

The Genetic Algorithm has proven quite effective in solving some optimization and Artificial Intelligence (AI) problems. It has been used in many application areas including Pattern Recognition. However, the applications of Genetic Algorithm to Pattern Recognition have so far concentrated primarily on training Neural Networks for Pattern Recognition (Montana 1989, Whitley 1992, Kitano 1994). The research in this paper is aimed at using the Genetic Algorithm to do pattern matching directly. The basic idea is to use the Genetic Algorithm to find the best match between the nodes of two patterns. The objective function can be defined in terms of the total difference in magnitude of angles between the corresponding edges of the two patterns. The experiments designed to evaluate the algorithm have shown very promising results and are highly accurate in classifying the input patterns.

1. Introduction

Pattern Recognition (PR) is a process whereby objects appearing in an input pattern are classified according to type. Applications of PR occur in many areas such as computer vision, face recognition, speech recognition, character recognition, signal classification and analysis, medical diagnosis etc. PR techniques have also been used as an important component of intelligent systems for data preprocessing and decision.

Following are some of the approaches that have been commonly used for PR (Schalkoff 1992):

1. Statistical Pattern Recognition (Devijver 1982)
2. Syntactic Pattern Recognition (Fu 1982)
3. Neural Pattern Recognition (Pao 1989)

Although the three approaches have shown some good classification capabilities, each has its own limitations. Statistical Pattern Recognition uses characteristic, elementary numerical description of objects, called features, to classify objects. The feature space of all possible patterns is divided into cluster space of objects using predefined decision rules. The pattern classification is performed based on discriminant functions and feature vectors of the pattern with respect to the cluster space. Statistical Pattern Recognition has difficulty expressing structural information of a pattern since structural information is irrelevant to this approach.

The Syntactic Pattern Recognition Approach tries to describe a complex pattern as a composition of simpler subpatterns or primitives. The primitives, or building blocks, are extracted from the input pattern using some structural rules and features of the pattern to distinguish which primitives the pattern consists of. Based on these extracted primitives and relations between them, a description word representing an object is formed. The type of object in the pattern can then be identified by syntactical analysis on this description word using description grammar for each class of objects. The major limitation of this approach is the difficulty in learning or deriving structural rules from a training set of patterns. This learning process may require significant human interaction.

In the Neural Network Approach, an artificial neural network, usually a feed-forward neuron network, is trained to perform like the human neural network. The network consists of neuron cells organized in a multi-layer platform. A neuron cell in one layer accepts input from neuron cells in the previous layer and produces output to the cells in the next layer. The training algorithm called Back-Propagation is normally used to adjust weight associated with each link connecting two cells residing in adjacent layers. The weight effects the input value presented to the cell and so the output of the cell. When presented with an input pattern the network can produce output which identifies the type of object in the input pattern. Although the neuron network can deliver highly accurate pattern recognition, the structural information of patterns is hidden inside the network and cannot be drawn out from its computation. Another concern regarding the neural network approach is the long duration time sometimes required to train the network.

Besides these three approaches, some alternative approaches have also been explored, e.g. graph matching and dynamic programming. The Genetic Algorithm is another approach that needs to be pursued for PR since it is capable of solving optimization and machine learning problems. The Genetic Algorithm should be a good candidate method to be used in determining which of the known patterns that best matches the input pattern.

2. Genetic Algorithm (GA)

A Genetic Algorithm mimics the biological evolutionary process in which new offspring can inherit good characteristics from their parents (Srinivas 1994, Mitchell 1996). The GA encodes a potential

solution as a string of 0's and 1's called a chromosome. With a set of initial chromosomes, the GA manipulates the most promising chromosomes in the set and creates the new population of improved chromosomes. The manipulation is a process of mating two chromosomes to create two offspring which inherit some of the good characteristics from their parents. The new population should contain solutions that are more promising than the previous generation.

Each chromosome is assigned a score or fitness function which is the target of the improvement. In order to generate improved offspring, we need criteria to select the good parents from the pool of chromosomes. The selection criteria should assure that the more promising parents should have more chance to be selected. Therefore the mating would yield improved offspring. One way to achieve this selection goal is to select parents based on their fitness values. The higher a parent's fitness value, the higher the chance of it being selected.

Once the parents have been selected, the next issue to consider is how the mating and evolutionary process are performed in order to generate offspring. Two basic operations are used for this purpose:

1. Crossover: This operation is performed on a pair of selected chromosomes and yields a pair of offspring. By picking a crossover point randomly (range 1 to L-1 where L is the length of the chromosome) the portions of two chromosomes around the crossover point are exchanged resulting in a pair of offspring. The crossover would take place only when a generated random number between 0 and 1 is less than or equal to a predetermined rate or constant P_c. Otherwise, the two parents remain unchanged and become the subjects of the second operation.
2. Mutation: This operation is performed on the offspring generated by the crossover operation. A randomly positioned piece in the offspring chromosome is changed from 0 to 1 or vice versa. The mutation is needed to allow the GA to escape from local optima since the offspring can be randomly changed and therefore be moved away from the local optima. As in the crossover operation, the rate of mutation can be controlled using another constant P_m. If a generated random number does not exceed P_m, then the mutation can take place. Otherwise, the offspring will remain unchanged. The output offspring from the mutation becomes a new member of the next population.

Following is an outline of a simple GA:

1. Initialize population of n candidate solutions by randomly generating n strings of chromosomes.
2. Evaluate the fitness value of each chromosome in the population
3. Repeat the following steps until there is no improvement on the average fitness value of the current population.
4. Repeat the following steps until n offspring have been created from the current population:
   - 4.1 select any pair of chromosomes from the current population
   - 4.2 perform crossover operation on the pair with probability of P_c
   - 4.3 perform mutation operation on the two offspring with probability of P_m
5. Replace the current population with the population of the n offspring
6. Evaluate the fitness value of each chromosome in the current population
7. Go to step 3.

3. Pattern Representation

Since this research is focused on the pattern recognition process, we assume that an input pattern has already been preprocessed (Gonzalez 1993). The preprocessing should include the segmentation of the pattern into a set of links or segments. Each link constitutes a straight line connecting two nodes (deviational points) in the pattern. Each link of the pattern can be represented by the XY-coordinates of its two end nodes. However, the XY-coordinates of the nodes are dependent on a scaling factor and degree of rotation imposed on the input pattern. This means that if the pattern is enlarged and/or rotated, the XY-coordinates of nodes also change. The changes complicate the process of patttern recognition, therefore, a new pattern representation is needed to eliminate the scaling and rotation dependencies. We propose a new scheme to represent a pattern, that eliminates the dependencies and is suitable for our proposed pattern recognition algorithm. The scheme consists of the following steps:

1. Between any pair of nodes that are not directly connected, introduce a new link that connects the two nodes but does not cross over any original links. The new links will be called augmented links. For instance, the augmented link AC and BD (dotted lines) are introduced into the pattern shown in figure 1.

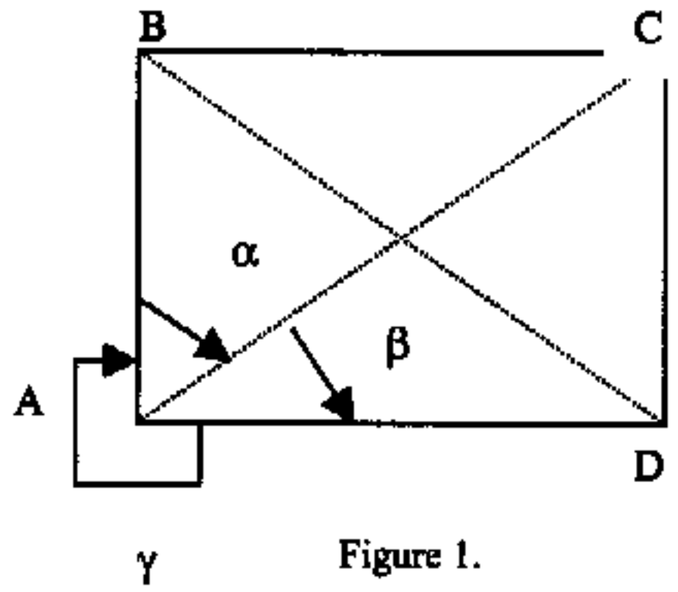

Figure 1.

2. Each original or augmented link connecting to a node will be represented by the degree of the angle spanning between the link and its adjacent link in a clockwise direction. For instance, the link AB, AC and AD are represented by the degree of the angles α, β and γ respectively.

The new representation uses the degrees of angles spanning between two adjacent links to convey structural information of a pattern. The angles are not subjected to changes due to the scaling and rotation imposed on the pattern. The introduction of augmented links also gives us some extra structural information about the pattern, e.g. which node indirectly connects with others through an augmented link. This information is quite helpful to distinguish one patttern from another. For instance, consider the two patterns, a square and a rectangle, shown in figure 2. Without augmented links the two patterns would have the same representation in term of the degrees of angles.

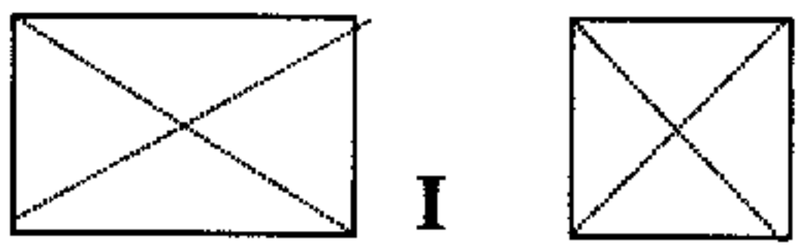

Figure 2

With the presence of augmented links in the two patterns, some angles associated with matched links will no longer have the same size and therefore one of the patterns can now be distinguished from the other.

4. Genetic Algorithm for solving PR problems

In order to use Genetic Algorithm to solve PR problems, we need to view a PR problem as a problem of finding the best match between nodes in a given input pattern and nodes in known patterns. The best matching solution should give the maximum value of fitness function which is particularly designed to reflect the similarities between the two patterns. The pattern among the known ones with the best fit will be recognized as the one that best matches the input pattern.

Before we examine a suitable fitness function for the PR problem, we should look at how each potential solution for the match can be encoded. We can view any potential solution of the PR problem as a 1-to-1 mapping between nodes in the two patterns. The mapping is restricted to 1-to-1 function since many-to-many mapping would create a cluster of nodes that are mapped onto a node or cluster in the other pattern. The cluster forms an equivalent class of nodes which contradicts the definition of the nodes (nodes must be deviational points of the pattern)

Each candidate solution can be encoded as a string of length N of integer numbers between 0 and M where M and N are the numbers of nodes in the known and input patterns respectively. The numbers between 0 and M-1 correspond to the mappings onto node numbers between 0 and M-1 in the known pattern respectively, while the number M is a special mapping number used to indicate that the corresponding node in the input pattern is not mapped onto any node in the known pattern.

5. The Fitness Function for PR

As mentioned earlier, the fitness function for PR should indicate the degree of similarity between the two patterns. This degree of similarity can be measured by the average similarity degree between links in the input pattern and corresponding ones, if they exist in the known pattern. Consider the parts of patterns shown in figure 3.

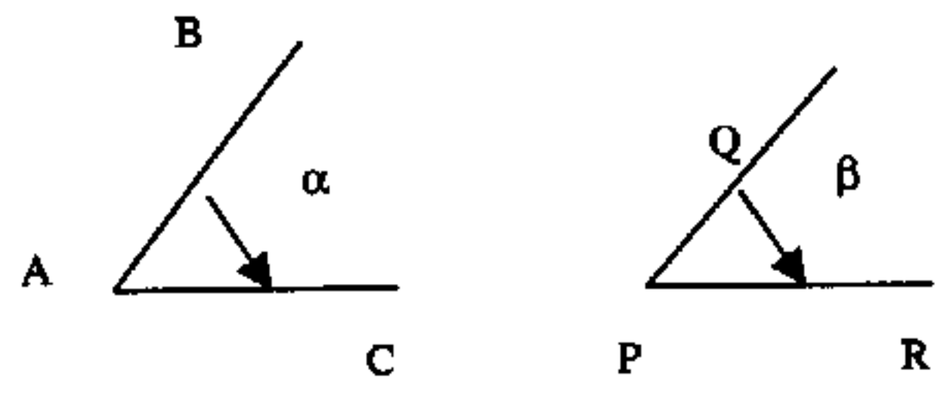

Figure 3

Suppose A is a node in the input pattern and is mapped onto node P in the known pattern. If there exists a link PQ in the known pattern where B is mapped onto Q and both links, AB and PQ, are the same type (original or augmented), then AB is said to be mapped onto PQ. Furthermore, if the next adjacent link of AB (clockwise direction), AC in this case, is also mapped onto the next adjacent link of PQ, i.e. C is mapped onto R and AC and PR are the same type, the similarity degree of AB and PQ with respect to A can be computed by a similarity function defined over the absolute difference value between the magnitudes of the angles BAC and QPR, $\Delta = |\alpha - \beta|$.

This similarity function should show the function value decreasing as the difference Δ increases. The function could have a maximum value of 1.0 when Δ is equal to 0 and decrease to zero when Δ approaches 2π or 360 degrees. Figure 4 illustrates an example of similarity function f and its linear approximation which is used for ease of evaluation. The linear approximation of f contains two threshold values for Δ, a and b.

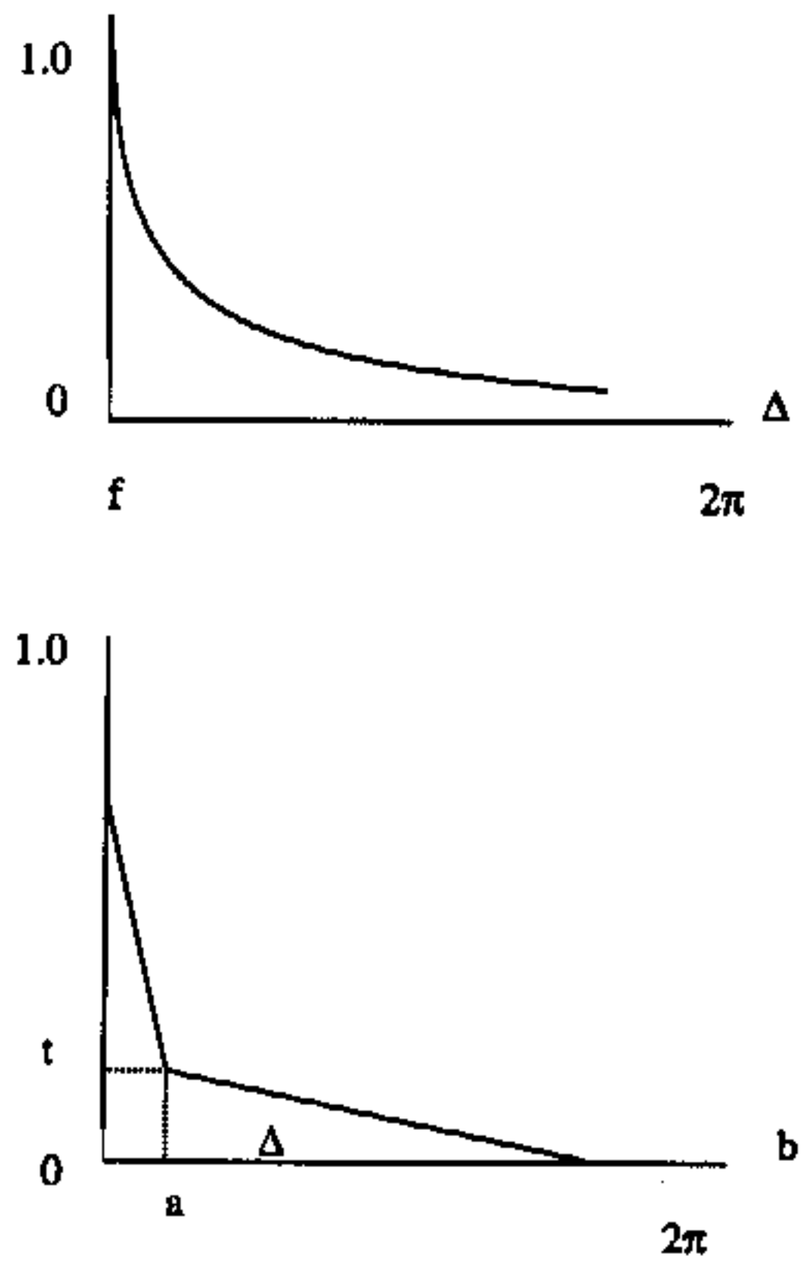

Figure 4

To find the degree of similarities between the two patterns, it is reasonable to have the fitness function evaluated on the original input pattern (without augmented links), then compliment the result by adding it with the value of the fitness function evaluated on the augmented input pattern (with augmented links). The complimentary value is intended to distinguish some similar patterns by exploiting the extra structural information inferred by the augmented links. The degree of similarity between the two patterns can then be defined as a sum of two parts as follows:

Σ Similarity degrees of all links with respect to all nodes in the original input pattern / (2 * number of links in the original input pattern) +Σ Similarity degrees of all links with respect to all nodes in the augmented input pattern / (2 * number of links in the augmented input pattern)

Notice that each link will be evaluated twice in each part by the similarity function, one with respect to each end node of the link. Hence the factor 2 is needed as the denominator of each part to get the average value. Each part has the value of at most 1.0 and therefore the degree of similarity between the two patterns would range from 0.0 to 2.0. It is also possible to multiply both parts with their weighting factors, e.g. 0.5 and 0.5 respectively, so the total value will not exceed 1.0. The weighting factors can also be adjusted to give more weight to one part than the other.

6. The Crossover Operator

Once each potential solution or chromosome can be represented as a string of numbers (ranging from 0 to M), the crossover operator to be performed between two chromosomes simply picks a crossover point randomly then exchanges the portions of the two chromosomes around the crossover point. However, if either of the new offspring is not a valid chromosome, i.e. the chromosome does not constitute a 1-to-1 mapping between the two patterns, then a new crossover point must be randomly generated until both new offspring contain valid mappings. If no such crossover point exists, the crossover operation will not be performed on the parent chromosomes and so the new offspring are simply the copies of the parents.

7. The Mutation Operator

The mutation to be performed on the chromosome is quite straightforward. It begins with the selection of which numbers in the chromosome to be changed. For each number to be changed, a random number is generated to become the new value of the number. Since the mapping represented by the chromosome is 1-to-1, the random number must be generated from the set of numbers ranging from 0 to M

and does not exist in the chromosome except the number M which can occur in the chromosome as many times as needed.

8. Experiments on the genetic algorithm for PR

In order to evaluate the effectiveness of the proposed genetic algorithm for PR, a testing set was created. Patterns of characters A-Z and 0-9 were chosen to be the known patterns in the testing set. Each pattern in the testing set was plotted on 4x8 matrix. The patterns were assumed to be preprocessed and augmented. Nodes in the patterns were represented by their XY coordinates. Pattern original and augmented links were defined by their two end nodes. These XY coordinate representations, once input, were then converted into angular representations as defined in section 3. The conversion would make the representations to be independent from the scaling and rotation factors, as well as to be suited to the algorithm. Input patterns to be tested on the algorithm were also input in XY coordinates and then converted into angular representations.

9. Population of the Potential Solutions or Chromosomes

Two important issues concerning the population of the potential solutions or chromosomes need to be addressed before we can perform the experiments. The first issue relates to population size and the second issue to the initialization of the population. The decisions on the size of population and its initialization algorithm have a significant impact on the performance of the genetic algorithm. Too small a population size could lead to local optima or early convergences while too large a population would exhaust processing time and memory space. An appropriate initialization algorithm would contribute to the randomness in the population concentration and yield a higher chance that the global optima would be found during the course of the genetic algorithm.

Consider a node in the input pattern. We need at least M+1 chromosomes in the initial population so that the particular node could have all possible mapping values assigned to it (0 to M). This will assure that any possible mapping values for the particular node will be included in the initial population. The other numbers inside each of the M+1 chromosomes could be randomly generated as long as the result chromosomes represent 1-to-1 mappings. Since the input pattern has N nodes in the input pattern, at least N * (M+1) chromosomes would be needed in the initial population.

However, the number of chromosomes would not guarantee global optima convergences especially when the number of nodes in patterns are large. To study how the population size effects the

global optima convergence of the algorithm, a multiplier S, is used to scale up the initial population size. Hence,

Initial population size = S * N * (M + 1)

To generate the initial population with this size, the process of generating N * (M + 1) chromosomes needs to be repeated S times. Hence, the initial population would be comprised of S different sets of N * (M + 1) chromosomes.

10. Selection Schemes and Terminating Conditions

Besides parameters such as population size, crossover and mutation rates etc., a selection scheme plays a major role in achieving successful convergence to the global optima. A poor selection scheme can cause premature convergence to a local optima. Several selection schemes had been proposed and explored (Goldberg 1989, 1991, Mitchell 1996). To study the effect of the selection scheme on the algorithm, we chose a few simple selection schemes to experimented with.

The selection schemes are:

1) Fitness-Proportionate Selection
2) Tournament Selection
3) Tournament Selection with some modifications

The fitness-proportionate selection is one of the very first selection schemes that were used for GA. The expected number of times an individual in a population will be selected to reproduce during the selection process is equal to the individual's fitness value divided by the average fitness value of the population. Roulette wheel sampling is the most common method used to implement the selection algorithm. The selection can be implemented as follows:

1) Sum the fitness value of all individuals in the population and call the sum D
2) Repeat the following N times where N is the size of the population
   a) Choose a random integer R between 0 and D.
   b) Loop through the individuals in the population, summing their fitness values until the sum is greater than or equal to R. The individual whose expected value puts the sum over the value R is the one to be selected.

The major problem with fitness-proportionate selection is premature convergence. Typically, early in the search the fitness variance in the population is high and a small number of individuals are much fitter than the others. These individuals and their descendents would have a high chance of being selected and

may prevent the GA from exploring other portions of the population that might lead to the global optima solution.

The Tournament selection scheme tries to avoid premature convergence by reducing pressure to select only individuals with high fitness, especially at the earlier stage of the search. The selection scheme would give a chance for less fit individuals to be selected and reproduce their offspring. The selection scheme is quite simple and suitable for parallel implementation. To select an individual to be one of the parents, the following steps are made:

1) Choose two individuals at random from the population
2) Generate a random number r between 0 and 1
3) If r < K (where K is a parameter, e.g. 0.75), the fitter of the two individuals is selected to be a parent, otherwise the less fit individual is selected.
4) Return the two individuals to the original population so they can be selected again.

However, the tournament selection scheme introduces one extra parameter K which can effect the performance of the GA. At the earlier stage of the search, the fitness variance is high, therefore, it is appropriate to relax the selection pressure by having a small K parameter. When the variance gets smaller at the later stage of the search, the selection pressure should be strengthened to allow fitter individuals to reproduce. Hence the parameter K should be increased as the search proceeds. This leads to the idea of varying the value of the K parameter during the search in the third selection scheme. This parameter K would be assigned with an initial value, e.g. 0.7, and then be increased by a small amount, e.g. 0.02, for every iteration until its value reaches a threshold value, e.g. 0.9, where the parameter value becomes unchanged.

When the GA stops the search it should be ensured that the global optima has been examined or reached. One way to ensure such a condition is to have the maximum fitness value remain unchanged for some number of iterations before the search is to be terminated. The threshold for the number of iterations can be fixed for all patterns or varied depending on the size of the patterns. The appropriate value of the threshold could be a subject for further study.

11. Experiment Results

As mentioned earlier, the experiments would be divided into three suites, one for each selection scheme. Each suite is comprised of testing sets with different values of parameters, i.e. population size, crossover rate and mutation rate. An approximation similarity function, shown in figure 4, was used for the

experiments. Each testing set consists of 36 input patterns (0-9 and A-Z). ,Each testing pattern was matched against the same 36 known patterns for 5 trials. Hence, 180 test runs would be needed for each testing set. During the test runs the percentage of correct classifications were recorded. The results of correct classifications for all testing sets are shown in tables 1 and 2.

Fitness Proportionate Scheme

CR \ MR	0.05	0.1
0.6	93.89	95.56
0.8	92.22	92.78

Tournament Scheme with fixed K

CR \ MR	0.05	0.1
0.6	92.78	90.56
0.8	95.00	90.56

Tournament Scheme with variable K

CR \ MR	0.05	0.1
0.6	95.56	100.00
0.8	93.33	98.33

Table 1

Notes :

1. CR = Crossover Rate, MR = Mutation Rate

2. Population Factor S is fixed at 1.00 P where P is the maximum value between the number of nodes in the input pattern and its counterpart known pattern.

S / Selection	0.5 P	1.0 P	2.0 P
Fitness Proportionate	92.78	95.56	98.89
Tournament fixed K	86.56	90.56	96.11
Tournament Variable K	91.67	100.00	100.00

Table 2

Notes :

1. CR is fixed at 0.6 and
2. MR is fixed at 0.1.

From the results of the experiments several conclusions can be drawn. They are:

1) Among the three selection schemes, the tournament with variable K performed best in recognizing the patterns. The GA using the selection scheme achieved almost 100 % accuracy in classifying the patterns when crossover and mutation rates are moderate, e.g. 0.6 and 0.1 respectively.

2) The size of the population has a significant impact on the performance of the algorithm. The GA performs better when population size gets larger. The appropriate population size depends very much on the number of nodes in the input pattern and its counterpart pattern. In our experiment, the appropriate population size for both the input pattern and known pattern with no more than 15 nodes would require the population factor S between 1.00 P and 2.00 P. These appropriate values of S were derived by incrementing the factor values until the best performances were achieved. In most cases, the tournament selection scheme with variable K required the smallest population among the three schemes to achieve the correct classifications. The scheme to select the best value of S should be a subject for further investigation.

3) All test runs were terminated when maximum fitness figures remained unchanged for some number of iterations. Regardless of which selection scheme was used, the numbers of iterations can be bounded by O (P log P) in most cases.

4) The variables, a, b and t, of the approximation similarity function show some influence on the algorithm performance. The values of 20 and 90 degrees were used for a and b respectively

while t was fixed at 0.3. These values of a, b and t can vary to some extent without effecting the performance of the algorithm.

5) Recognition errors mostly occur in input patterns with large numbers of nodes (such as 8) and in patterns with similar shapes such as zero and letter o, 6 and 9. The errors were probably caused by premature convergence to local optima. However, these errors were reduced when the population size was increased. Input patterns with degrees of angles different from those in the corresponding known patterns were also tested. The algorithm can still recognize the input pattern correctly despite some differences between the input and its corresponding known patterns.

Although the GA performanced well in recognizing patterns in our experiments, it still needs to be seen whether it can perform well on more complex or larger patterns. Some optimizations could be applied to the algorithm when used with larger or more complex patterns. For instances, all known patterns can be matched against the input pattern concurrently. Any matchings with known patterns that yield very small fits, i.e. smaller than a threshold for some number of iterations, should be frozen or terminated while the others that show promising fit values could proceed with high priority. This could help the concurrent searches concentrate their work on the more promising prospects despite some small risk that a global optima might be missed. Another method to improve the accuracy of the algorithm is to repeat any matchings in a number of trials. This should result in a higher possibility that the global optima would be reached by at least one of the trials.

12. Future works and Conclusions

The direct application of the GA to PR has been explored in this research. The aim is to introduce an alternative method for solving PR problems. The results from the experiments conducted on the algorithm are quite encouraging. The algorithm achieved very high accuracy on English and numeric character recognition (100% correct classification in some parameter settings). This accuracy is comparable to, or even better than, those of some other approaches such as Statistical PR, Syntactical PR and Neural PR. Futhermore, the PR using GA does not need a long training process such as that required by others approaches. The GA is also suitable for parallel implementation, a prerequisite for recognizing large complex patterns.

Nevertheless, more work needs to be done, especially on the tests for large complex patterns. It is possible to use GA to select subpatterns that best form a larger pattern based on some predefined syntax

rules. This is to allow a higher or more abstract level of PR to be performed using the outcomes from the lower levels of pattern classification. Some optimizations and parallel implementations of the algorithm should also be pursued in order to enhance the performance of the algorithm.

Bibliography

Devijver, P. and Kittler, J. (1982). *Pattern Recognition : A Statistical Approach*, Prentice-Hall, Englewood Cliffs, N.J.

Fu, K.S. (1982). *Syntactic Pattern Recognition and Application*, Prentice-Hall, Englewood Cliffs, N.J.

Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley.

Goldberg, D.E. and Deb K. (1991). "A comparative analysis of selection schemes used in genetic algorithms", *Foundations of Genetic Algorithms*, Morgan Kaufmann.

Gonzalez, Rafael C. and Woods, Richard E. (1993). *Digital Image Processing*, Addison-Wesley.

Kitano, H. (1994). "Neurogenetic Learning : An integrated method of designing and training neural networks using genetic algorithms", *Physica D 75*, p. 225-238.

Mitchell, Melanie (1996). *An Introduction to Genetic Algorithms*, MIT Press.

Montana, D.J. and Davis, L.D. (1989). "Training feedforward networks using genetic algorithms", *Proceedings of the International Joint Conference on Artificial Intelligence.*

Pao, Y.H. (1989). *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley.

Schalkoff, Robert (1992). *Pattern Recognition, Statistical, Structural and Neural Approaches*, John Wiley & Sons.

Srinivas M. and Patnaik, Lalit M. (1994). "Genetic Algorithms : A survey", *Computer*, June 1994, p.17-26.

Whitley L.D. and Schaffer J.D eds. (1992). *COGANN-92 : International Workshop on Combinations of Genetic Algorithms and Neural Networks.*