Genetic Based DNA Algorithm to Solve Distribution Problem

ABSTRACT:

In this paper a genetic based DNA algorithm is proposed to deal with the distribution problem. DNA interons , Hot and Cold spots features are used to improve the performance of standard genetic algorithm. This modified algorithm can solve single or multiobjective problems. The distribution problem is one of the important economic problems. The correct solution of this problem will save time and cost. This proposed algorithm is implemented on a distribution problem. The results are compared with those yield from genetic algorithm and neural network.

Keywords

Genetic algorithm, interons, exons, hot and cold spots, building blocks, distribution problem.

Introduction:

Genetic algorithm is one of the most interesting heuristic search techniques. It depends basically on three operations; selection, crossover and mutation. The outcome of the three operations is a new population for the next generation. Repeating these operations until the termination condition is reached. The selected chromosome consists of specific bit patterns which contribute to its high fitness value. According to Annie S.wu. and Robert K. Lendsay [1] these patterns are called building blocks so that the better solutions have the ability to keep their building blocks in the next generation. Through the convergence process, good solutions (due to fitness) may be disrupted during the crossover operations. This consequently may result in losing these solutions through the generation upgrading. This problem increases the instability of the genetic algorithm. Moreover, the random mutation shows similar influences according to the positions and number of applied mutations.

In this paper, a new algorithm is proposed to decrease the crossover and mutation drawbacks by adding a protection to each chromosome parts and depending on a partially controlled mutation based on DNA features. These changes increase the stability of the genetic algorithm and produces better performance .

This paper is organized as follows: Section 1 introduces an introduction to the proposed algorithm. The based DNA features are presented in section 2 . The core of the proposed solution algorithm is described in section 3. A test case for distribution problem is solved using the proposed algorithm in section 4. The results are analyzed and compared to those of genetic and neural network algorithms in section 5. Finally section 6 summarizes the conclusions.

2. DNA and Based DNA features

DNA (deoxyribonucleic acid) is the primary genetic marital in all living organisms. It is a polymer which constructed from a series of monomers. Monomers which form the building of blocks of nucleic acid are called nucleotides. Each nucleotide consists of a sugar ring, a phosphate and a nucleobase. There are four types of nucleobases Adenine (*A*), Guanine (*G*), Thymine (*T*) and Cytosine (*C*). Single stranded DNA is a linear chain of nucleotides. Thus, single strand DNA is normally written according to the sequence of the nucleobases. For example, a single stranded DNA is written as $5^{'}$-ATCG-3'. Two respective single strands DNA can combine with each other at a specific condition. This attachment is subject to Watson –Crick complementary base pairing which means *A* is paired with *T* whereas *C* is paired with *G* to form a double helix of DNA. Since Adelman [2] introduced his algorithm to solve the directed Hamiltonian path problem using DNA coding strands, many researchers solved some NP hard problems like Knapsack problem, maximal clique problem and shortest path problem. But due to the drawbacks of biological operations and the high cost of this algorithm, some researchers go to the simulation of DNA operations or improving genetic algorithm using some features of DNA because it is well known that gene is a part of the hole DNA[3].

2.1 Interons and Exons

Living organisms need protein to survive, in human there are 20 kinds of amino acids which are the units of protein .Each amino acid has his private code or codes as genes in DNA, this genetic code is represented by a triplets of {A, T, C, G} so that we have (4^3=64) genetic code for these amino acids which are called coding areas or (Exons) [4], [5]. In biology much of DNA in higher organisms (like human about 90% of human DNA) does not represent any amino acid codes. They link between DNA codes and protect them from destruction in the process of evolution they are growing with evolutionary process. These parts are called non-coding areas or (Interons).

2.2 Hot spots and Cold spots

In Neuhauser and Krone [6], there are hot spots and cold spots in the DNA sequence, i.e the nucleotide bases in the cold spots mutates more slowly than those in the hot spots for an example nucleotides C and G mutates faster than nucleotides A and T .Given that a mutation occurs in the new individual, a site is selected at random from some distribution:

$$f=(f_1+f_2+......+f_M)=1 \qquad (1)$$

This allows to model hot spots for mutation process.

3.GA based DNA

3.1 Effect of Interons

There is no doubt that any algorithm of selection will select solutions of higher fitness so that, the improvement we are looking for could be produced by the crossover and mutation processes. Involving the idea of interons in GA by inserting some segments between the building blocks of each chromosome, this will produce the following:

1-Increasing the probability of making the crossover operation in between the building blocks rather than within them as shown in Fig.1[5]:

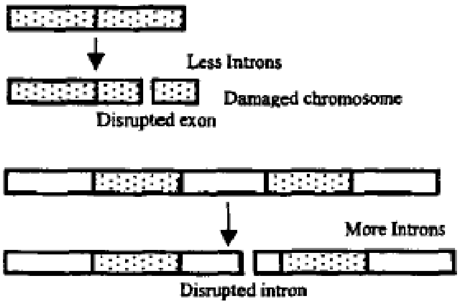

Fig.1 Crossover before and after adding interons [4]

2-Separating the building blocks and moving them apart from each other which reduces the hitchhiking effect.

3-Increasing the amount of interons during GA search gives various crossover points because of change in chromosome length.

4-Increasing the stability of GA during protecting the building blocks in each chromosome.

Based on Jili Tao and Ning Wang [7], GA with too large mutation probability becomes a random search and too small mutation probability subject the algorithm to run into a local minimum. According to hot and cold spots they considered if the chromosome was of length L so the bits between 1and $L/2$ is the set of low bit position (cold spots)and the rest is the set of high bit position (hot spots) .At the beginning of evolution larger probabilities of mutation are assigned in the hot spots to explore larger feasible region, towards the end of evolution the mutation probabilities in the hot spots are decreased to prevent better solutions from disruption .Accordingly there are two kinds of mutation probability p_{mh} and p_{ml} which are described as:

$$P_{mh}=a_1+b_1/1+exp[aa(g-g_o)] \qquad (2)$$

$$P_{ml}=a_1+b_1/1+exp[-aa(g-g_o)] \qquad (3)$$

Where a_1 is the initial probability of mutation of p_{ml} , b_1 is the range of transmutability , g is the evolution generation, g_o decides the generation where great mutation probability occurs, aa is the speed of change and G is the whole number of generations. Fig.2 illustrates the changing curves with evolution generations of p_{mh} and p_{ml} .

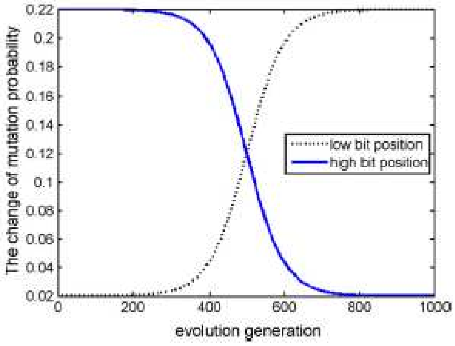

Fig.2 The curves of p_{mh} and p_{ml} [6]

Constants are taken as:

a_1=0.02 , b_1=0.2, $g_o=G/2$, and $aa=20/G$.

3.2 The Proposed algorithm

The following steps illustrate the proposed algorithm to apply GA with the new modifications:

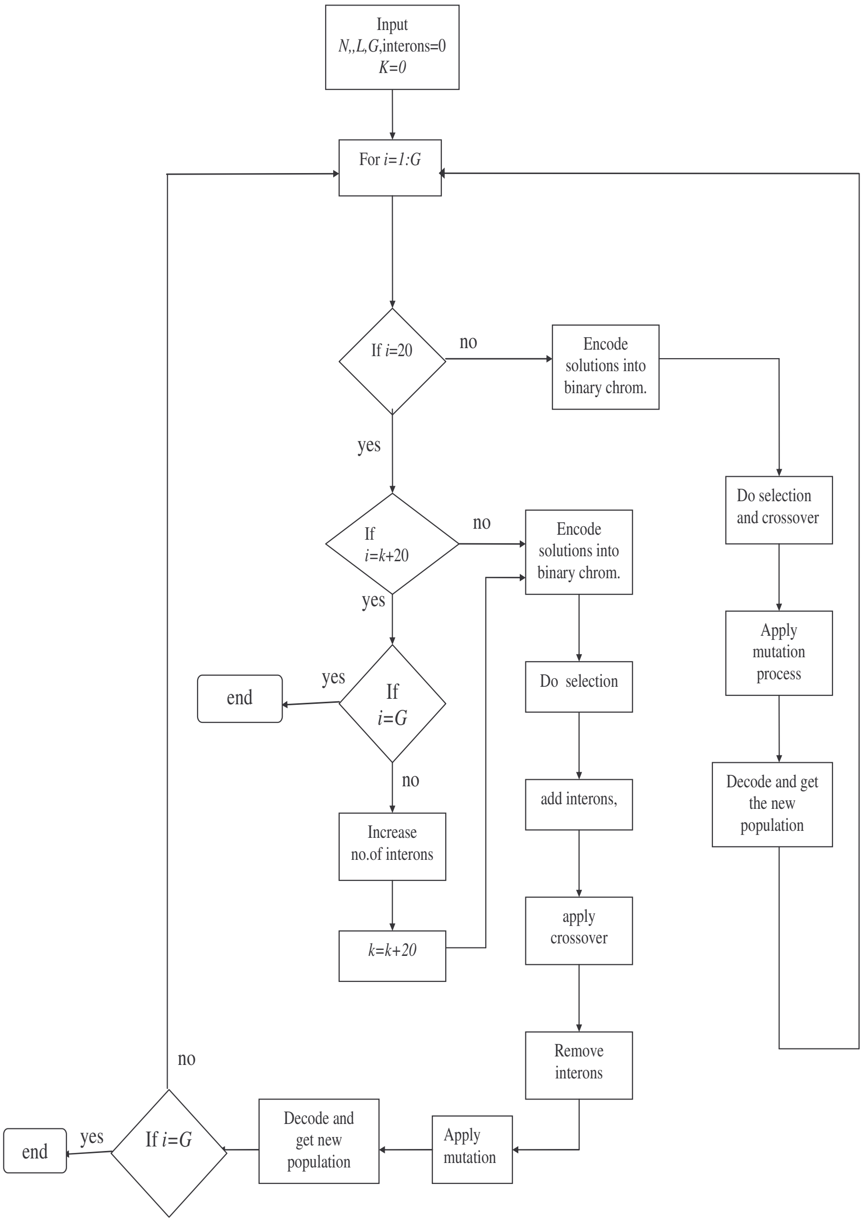

Fig.3 Flow chart of the proposed algorithm.

Step 1

Start with random population of constant size N and each chromosomes of length L *b*inary digits and without interons .

Step 2

apply selection and crossover then calculate the mutation of probability from (2),(3).Produce L decimal fraction between [0,1] and compare them with the calculated mutation probability. If the mutation probability is larger than the corresponding fraction the mutation takes place.

Step 3

After the first 20 generations , insert the interons between the chosen building blocks after selection and before applying the crossover operation .Then apply the mutation operation of step 2.

Step 4

Remove the interons and decode chromosomes to get new population .

Step 5

Repeat steps (3,2,and 4) with increasing the length of interons every 20 generation (for example)until reaching the maximum number of generations(G). The sequential steps of this algorithm are shown in Fig.3. This algorithm has been tested on single and multiobjective optimization problems in order to investigate its performance and the influence of proposed modifications [8]

4. The distribution Problem

4.1 Problem definition

For N number of existing objects and M number of groups where each object is characterized by k variables of k-dimensional vector. The input data are of this problem are represented by the coordinates $x_{1,}\ x_2,\ldots,x_\mathrm{k}$ That characterize the objects. It is possible to define any number of groups. The fitness function represents the sum of the squares of distances between the objects and the centroids. The coordinates of centroids c_{j1}, $c_{j2},\ldots,$ c_{jk} (j=1,……M) are changed. The calculation assigns the objects to their centroids. The whole process is repeated until the termination condition is reached. The process of optimization ensures that the defined coordinates $x_{i1,}\ x_{2i},\ldots,$ x_{ik} (i=1,2,…=N) of objects and the assigned coordinates $c_{j1},c_{j2},\ldots,c_{jk}$ of groups have the minimum distances. The fitness function is expressed by the formula:

$$F_{\min} = \sum_{i=1}^{N} \min_{j=(1,2,..M)} \left(\sqrt{\sum_{l=1}^{k} (x_{il} - c_{jl})^2} \right) \qquad (4)$$

4.2 Case study

In this case study yhe coordinates of towns are given in table1 and the coordinates of centeroids or distribution centers are searched. The centroids have minimum distances to allocated places. This task is solved using the GA based DNA algorithm.

Table.1 Coordinates of towns

Number	Town	X	Y
1	London	9	118
2	Paris	20	66
3	Bruxelles	48	98
4	Amesterdam	55	121
5	Luxemburg	64	73
6	Bern	75	27
7	Vaduz	100	27
8	Berlin	149	118
9	Praha	159	76
10	Ljubjana	159	6
11	Zagreb	177	0
12	Wien	181	42
13	Bratislava	191	42
14	Budapest	213	31
15	Warszawa	230	117

5. Results and analysis

5.1 Results

After solving this problem in the test case using the GA based DNA algorithm, the results are shown in table 2 and Fig.4

Table.2: Results of GA Bases DNA algorithm

```
Number of groups: 4

Population size: 1000

Assign= 1  1  1  1  1  2  2  3  3   4  4  4  4  4 3

Objective value= 410.7

Coordinates
X             Y
47.987        97.58
86.365        25.748
169.47        99.004
184.68        35.609
```

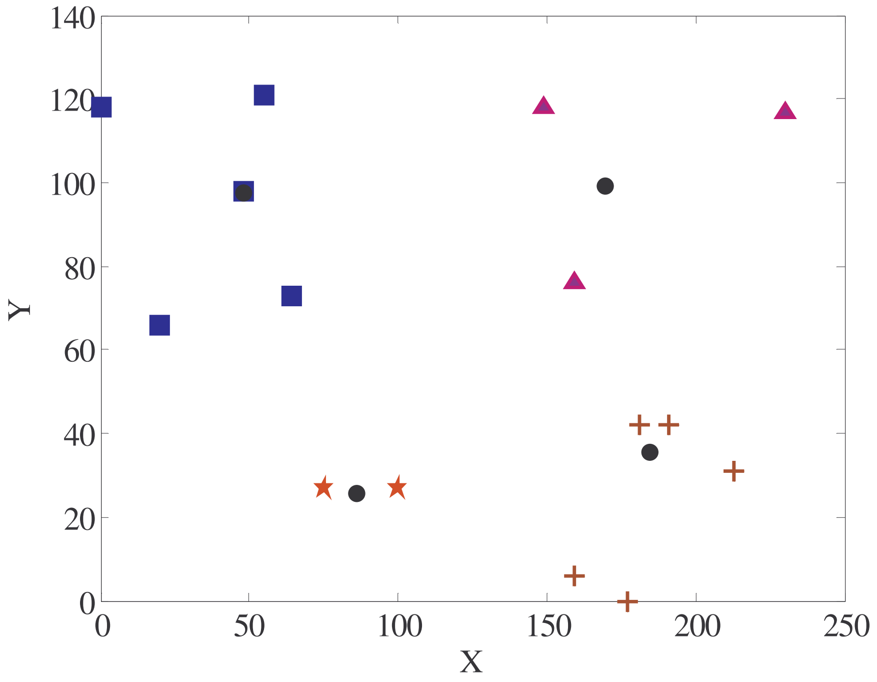

Fig.4: Groups and their centers

5.2 Analysis of results

1. When the same problem was solved using the genetic algorithm the objective value was (415.4) and when it was solved using the neural network algorithm the objective value was (448.4) . Both results of genetic algorithm and neural network ere obtained using number of generations as 1000 four groups and a population size of 1000 [9] . However in the GA Based DNA algorithm the

objective value is (410.7) using the same number of groups and population size but after 500 generations.

2. These results show the effectiveness of adding the DNA features to the standard genetic algorithm. The protection of selected solutions gives them the chance to evolve with protection to their parts. Even those chromosomes with selected weak fitness may be improved after having this chance.
3. The adaptive mutation reduces the disruption coming from the random mutation and save the algorithm from falling into local minimum.
4.

6. Conclusion

The distribution problem is one of the most important of economy and business branches. As an application the search for the best location of bank or market. This problem can be solved using the artificial intelligence tools as genetic algorithm and neural network and The GA based DNA algorithm. The GA based DNA algorithm is a modified genetic algorithm uses some DNA features as the interons, hot and cold spots which wee added to the standard genetic algorithm to improve its performance. A case study for this problem was solved using The GA based DNA algorithm. The results were compared to those yield from the genetic and neural network algorithms. This comparison clears the effectiveness of these modifications and the better performance of this proposed algorithm.