210	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
220	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
230	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
240	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
250	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ

ايجاد خوارزمية جينية مع الترميز الثنائي لمشاكل الامثلية الاستبدالية

د. براء علي عطية*

المستخلص

تعتبر مشكلة التسلسلية واحدة من أهم مشاكل الامثلية الاستبدالية والتي استقطبت اهتمام الباحثين في مجال الخوارزميات الجينية. ان الطريقة المعتادة لتمثيل المشاكل التسلسلية في الخوارزميات الجينية هي استعمال الترميز الاستبدالي لتمثيل الكروموسوم. توصلنا في هذا البحث الى طريقة جديدة لتمثيل الكروموسومات لشاكل الامثلية الاستبدالية باستخدام خوارزمية جينية ذات ترميز ثنائي. اجريت تجارب الخوارزمية الجينية المقترحة على مشكلة البائع المتجول في عشرة مدن. اوضحت نتائج التجارب على فعالية الخوارزمية الجينية المقترحة في حل مثل هكذا مشاكل.

* جامعة بغداد – كلية العلوم– قسم علوم الحاسبات

Table 2 Two different runs

Gen. number	*Run1*		Run2	
	Minimum distance	Path	Minimum distance	Path
1	3.546500	CBJIDGHFEA	3.182000	EFHJGICBAD
10	2.816100	BCADEFHGIJ	2.974300	JHIBCADEFG
20	2.751800	BCADEGFHIJ	2.974300	JHIBCADEFG
30	2.690800	BCADEFGHIJ	2.974300	JHIBCADEFG
40	2.690800	BCADEFGHIJ	2.974300	JHIBCADEFG
50	2.690800	BCADEFGHIJ	2.974300	JHIBCADEFG
60	2.690800	BCADEFGHIJ	2.751800	BCADEGFHIJ
70	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
80	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
90	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
100	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
110	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
120	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
130	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
140	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
150	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
160	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
170	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
180	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
190	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ
200	2.690800	BCADEFGHIJ	2.690800	BCADEFGHIJ

Table 1: 10-city problem with distance measure

	A	B	C	D	E	F	G	H	I	J
A	.0000	.3361	.3141	.3601	.5111	.5176	.2982	.4564	.3289	.2842
B	.3361	.0000	.1107	.6149	.8407	.8083	.5815	.6418	.4378	.3934
C	.3141	.1107	.0000	.5349	.7919	.8207	.5941	.6908	.4982	.4501
D	.3601	.6149	.5349	.0000	.3397	.6528	.5171	.7375	.6710	.6323
E	.5111	.8407	.7919	.3397	.0000	.4579	.4529	.6686	.7042	.6857
F	.5176	.8083	.8207	.6528	.4579	.0000	.2274	.2937	.4494	.4654
G	.2982	.5815	.5941	.5171	.4529	.2274	.0000	.2277	.2690	.2674
H	.4564	.6418	.6908	.7375	.6686	.2937	.2277	.0000	.2100	.2492
I	.3289	.4378	.4982	.6710	.7042	.4494	.2690	.2100	.0000	.0498
J	.2842	.3934	.4501	.6323	.6857	.4654	.2674	.2492	.0498	.0000

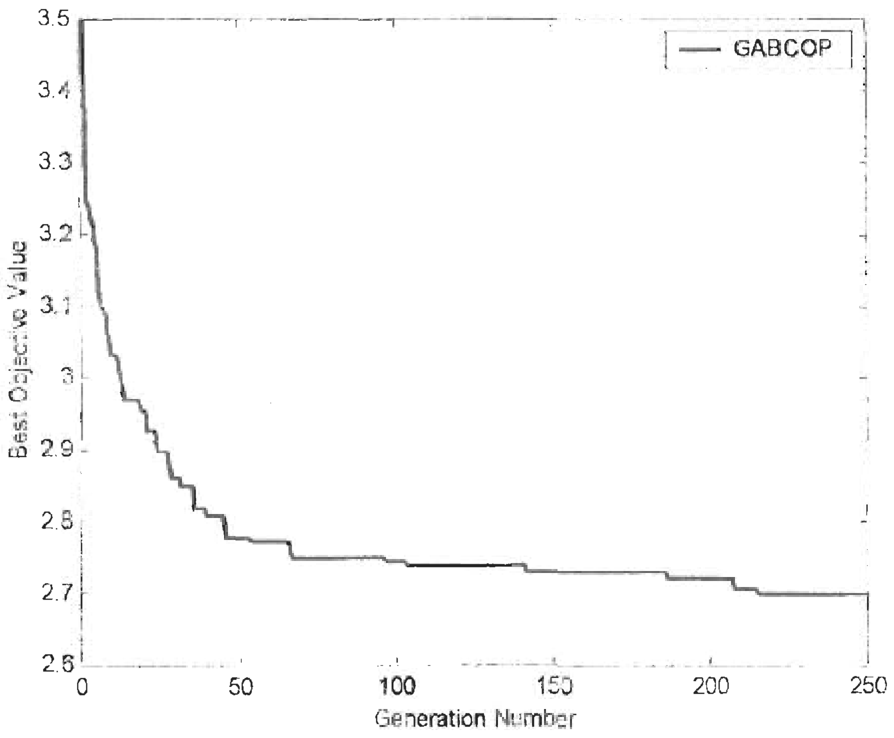

Figure 2: Best objective function averaged over 15 different successful runs.

```
Stringconvert_code(int    path_code)
/* Input: path_code is the corresponding integer value of the path
  Output: path is the actual path string */
{
 z=1;
 i=0;
 first_city = trunc (path_code / (n-z)!); /* find the first city in the
                                            code  sequence */
 rem = remainder (path_code / (n-z)!); /* calculate the remainder of
                                            the path  code */
 z++;
 i++;
 path[i] = first_city;
 While (rem != 0) && (z < n)
  {
next_city = trunc (rem / (n-z)!);
rem = remainder (rem / (n-z)!);
path[i]= next_city; /* assemble next city in the actual path string */
z++;
i++;
  }
 return (path);
}
```

Figure 1: C-like function for path decoding

[3] Goldberg, D. E. Genetic Algorithms In Search, Optimization, And Machine Learning, Addison Wesley, 1989.

[4] Evonet Flying Circus, Chapter 3: Applications Of Genetic Algorithms, Available At Http://Www.Wi.Leidenuniv.Nl/~Gusz/Flying_Circus.

[5] Fausett, L., Fundamentals Of Neural Networks,Prentice Hall International,Inc., 1994.

[6] Wilson, G. V. And Pawely, G. S., "On The Stability Of The Traveling Salesman Problem Algorithm Of Hopfield And Tank", Biological Cybernetics, 58, Pp. 63-70, 1988.

with fitness 2.690800 at respectively generation number 30 and 70.

In this paper, we investigate a genetic algorithm-GABCOP that uses bit coding for individual representation instead of the traditional ordering representation. The reason from using binary encoding rather than permutation encoding can be returned to the following. Binary encoding is most commonly used due to their generality to more than one problem.

Bit representation used less storage space than ordering representation. While in ordering representation each city has to take one character (i.e., one byte) for coding.

Number of bits required to represent one actual path (one complete solution) can be calculated by $trunc(\log_2(n!))$.

Traditional perturbation operators like one-point crossover, two-point crossover, and bit inverting mutation can easily be applied. However, many methods can be used to improve the GA performance with bit representation.

Finally, the results in this paper are preliminary, and more detailed investigation is needed.

References

[1] Goldberg, D. E., "Genetic And Evolutionary Algorithms Come Of Age", Communication Of The ACM, Vol. 37, No. 3, 113-119, March, 1994.

[2] Asveren, T. And Molitor, P., "New Crossover Methods For Sequencing Problems", Parallel Problem Solving From Nature-PPSN III, Eds. H. Voigt, W. Ebeling, I. Rechenberg And H. Schwefel, Berlin, Springer-Verlage, September, Pp.336-345, 1996.

In this paper we illustrate the function of GABCOP in terms of their ability to find solutions for the 10-city problem, which has been used for comparison by several authors [5][6].

The distances between the cities are given in symmetric distance matrix shown in table 1.

The following setting of the tested GA were used for experimental test runs:

Goal	Minimum path
Selection	Tournament with size two
Recombination	Two-point crossover with rate 0.6
Mutation	Flip mutation with rate 0.2
Population size	50
Termination criterion	250 generations
Run	20 runs with different random generator seeds

5. Results and Conclusions

Figure 2 summary the result for TSP. The horizontal axis denotes the generation number. The vertical axis denotes the average values of best objective function. The results have averaged on the successful runs. Fifteen runs out of twenty runs reach the optimum path (i.e. minimum distance).

Let take as examples two different runs and illustrates their results as depicted in table 2. the table gives indication about generation number, fitness value of best individuals obtained, and the actual path sequence displayed after every ten generations. Both runs reach optimal solutions demonstrated by path "BCADEFGHIJ"

1. A GABCOP GA coding uses binary encoding instead of permutation encoding.
2. GABCOP GA uses bit-level for GA perturbation operators (i.e., recombination, and mutation).

In GABCOP, the chromosome is represented as a collection of bits. The number of bits is determined according to the following:

where **n** is the number of cities in the problem. Thus each chromosome represents one path code from the possible paths.

This method generates illegal (or lethal) chromosomes, and we can avoid this situation by using penalty value or apply mod(n!) for each code.

Figure 1 shows the C-like *function that illustrates the* conversion of path code to the actual path. The input to the function is the genetic individual with binary coding which is first decoded into its equivalent integer value that represents the number of the path or in other words the order of the path according to all possible paths in the search space. It then iteratively extract from this integer value code city sequences starting from left city in the path sequence (highest digits of the path code) toward right city in the path sequence (lowest digits of the path code).

Since the chromosomes are represented as binary bits; the usual sequencing crossover and mutation operators can no longer be used. So the classical crossover and mutation for bit representation are used as perturbation operators for GABCOP algorithm.

4. Experiments Setting

The difficulty in the TSP using a GA is encoding the solutions. The encoding cannot simply be the list of cities in the order they are traveled. As shown below, the crossover operation will not work. The crossover point is the 3rd character. Every character in parent 1 before the crossover point is copied into the same position in offspring 1. Then, every character after the crossover point in parent 2 is put into offspring 1. The opposite is done for offspring 2.

Parent 1 A B C D E

Parent 2 C E B A D

Offspring 1 A B C A D

Offspring 2 C E B D E

As we can see, the city A is used twice and city E is missing in offspring 1. A more complicated form of encoding (or a more complicated crossover) must be used. This form of encoding should ensure that city adjacencies in the list are preserved from the parents to the offspring. This method should also realize that the tours A B C D E and C B A E D are the same. Partially matched crossover and cycle crossover are examples of crossover which permit the exchange of important ordering similarities between pairs of parents to form offspring [3].

3. The Proposed Genetic Algorithm

The proposed GA (coined as Genetic Algorithm with Bit coding for Combinatorial Optimization Problem GABCOP) is different from a GA that used permutation encoding in a number of ways:

$$number\ \ of\ \ bits = trunc(\log_2(n))$$

Even for this problems for classical crossover and mutation corrections must be made to leave the chromosome consistent (i.e., have real sequence in it).

It has often been the case that progress on the TSP has led to progress on many combinatorial optimization problems and more general optimization problems. In this way, the TSP is a playground for the study of combinatorial optimization problems. Though the present work concentrates on the TSP.

In this paper genetic algorithm that uses bit representation as a coding for solving combinatorial optimization problems is suggested. Section 2 gives an overview of traveling salesman problem. While section 3 presents the proposed genetic algorithm. Then section 4 presents the experiments and results of testing the proposed algorithm on traveling salesman problem. Finally section 5 draws conclusions.

2. Traveling Salesman Problem

The most prominent member of the rich set of combinatorial optimization problems is undoubtly the traveling salesman problem (TSP). TSP is defined as follows: Given n cities, compute a minimal cost tour on which the salesman visits each city once. Obviously, TSP is a sequencing problem as defined above. This problem is a classical example of NP-complexity: the work-area to be explored grows exponentially according with the number of cities, and so does complexity of every known algorithm to solve this problem. It is not only a good example of a combinatorial optimization problem, but also an approach to real problems like Touring Airports, VLSI-design or X-ray crystallography [4].

1. Introduction

Genetic algorithms (GAs) are stochastic search algorithms based on evolutionary and biological processes that enable organisms to adapt more to their environment over many generations. They are being successfully applied to problem in business, engineering and science [1]. GAs work on a set of possible solutions, which is called the population. The most basic operations used by GAs are selection, crossover, and mutation. The selection operator identifies the individuals of the current population which will serve as parents for the next generation. Crossover randomly chooses pairs of individuals to combine properties of them by creating offspring. Mutation is usually considered as a secondary operator, which makes small changes on single individuals to restore diversity of the population.

A very prominent and interesting class of combinatorial optimization problems is the class of sequencing problems. A sequencing problem is defined as follows: given a finite set I of items and a cost function c which assigns costs to permutations of I, compute minimum cost permutation of I. Problems as, e.g., the traveling salesman problem (TSP), the problem of computing good variable orders of Binary Decision Diagrams (BDD) which has to be solved for circuit verification and logical synthesis, scheduling problems, or serration in archeology are representatives of this class [2].

The usual method of applying genetic algorithms to sequencing problems is to encode every chromosome as a string of numbers (characters), which represents number in a sequence. This encoding is called permutation encoding, which is only useful for sequencing problems [3].

An Investigation Of Genetic Algorithm With Binary Coding For The Combinatorial Optimization Problems

Dr.Bara'a Ali Attea*

Abstract

One of the most promising class of combinatorial optimization problems is the sequencing problem that has received great attention of genetic algorithm researchers. The usual way of handling sequencing problem in genetic algorithms is through using permutation encoding of chromosomes. Hence, a special kind of perturbation operators has to be applied. In this paper, we investigate a new genetic algorithm however with bit coding representation of chromosomes for the sequencing problems- coined as Genetic Algorithm with Bit coding for Combinatorial Optimization Problems (GABCOP). The computational performance of GABCOP is illustrated as a test performed on a 10-city TSP. The results indicate that the algorithm is effective for optimizing such combinatorial problems.

*University of Baghdad - College of Science - Department of Computer Science