

A NEW METHOD FOR SELF-ADAPTATION OF GENETIC ALGORITHMS OPERATORS

Nada Thanoon Ahmed, Yasmin Makki Mohialden and Dina Riadh Abdulrazzaq

Computer Science Department, Collage of Science, MustansiriyahUniversity, Iraq

ABSTRACT

Genetic Algorithms (GAs) plays a vital role in finding the optimal correct solution of different problems. Some factors and operators it depends on are: population size, selection, mutation and crossover. Many researchers were engaged in the process of determining these operators. In this paper, we will introduce a new mechanism for self-adaptions of Steady State Genetic Algorithms (SSGAs) operators. Applying the suggested method to the problem of the Traveling Salesman Problem (TSP) proved to be more successful in comparison to other traditional methods.

Keywords: Genetic Algorithm, self-adaptions Crossover, self-adaptions Mutation, Genetic Algorithm, Traveling Salesman Problem, optimization.

Cite this Article: Nada Thanoon Ahmed ,Yasmin Makki Mohialden and Dina Riadh Abdulrazzaq, A New Method for Self-Adaptation of Genetic Algorithms Operators, International Journal of Civil Engineering and Technology, 9(11), 2018, pp. 1279–1285.

<http://www.iaeme.com/IJCIET/issues.asp?JType=IJCIET&VType=9&IType=11>

1. INTRODUCTION

Darwin's theory of biological evaluation inspired the invention of genetic algorithm (GA). It can be used as an optimization algorithm to solve both minimization and maximization problem. Usage of GA depends on some genetic operators like; selection, crossover and mutation. It is specified as a well-established, flexible, of easy programming method [1].

GA strength is demonstrated by its capability to locate the global optimum in a multimodal surrounding. Unfortunately. The results it delivers bear a definite measure of untrustworthiness regardless of how efficient and strong a genetic algorithm may be. Genetic Algorithm can, therefore, locate the global optimum only with some probability of successes. A substantial attention has been paid to increase this probability. In that regard, two major approaches can be recognized:

The first approach is designing a GA for a class of specified problems. This approach includes creating data structures and a genetic operator's characteristic program. The approach is nevertheless problem-specific and needs a lot of modeling for each purpose.

The second approach acts on the algorithm directly by trying to increase the efficiency of change in its internal structure. This method does not, generally, improve the performance of the algorithm as the first one, but it is not problem-dependent, and it does not limit the applicability of the algorithm at all.

This paper suggests a new method for self-adaptation of Steady State Genetic Algorithms (SSGAs) operators and applied on a well-known problem the traveling salesman problem (TSP) [2, 3, and 4]. The adaptive method presented here is an example of the second approach mentioned earlier.

The remaining part of the paper is organized in four sections: in the next section we describe briefly the genetic algorithm and TSP. A brief literature survey follows in section 3. The test results are presented in section 4 and the conclusions and the future perspectives of this work are discussed in section 5.

2. BACKGROUNDS

In the following sections a brief description of GA and TSP were introduced.

2.1. Genetic Algorithms.

GA is inspired natural selection, the process that drives biological evolution and is used as a method for solving optimization problems. GA modifies a population of individual solutions repeatedly. The genetic algorithm selects randomly, at each step, individuals from the current population are selected to be parents and the GA uses them to produce the children for the next generation. GA operators are: Selection, Crossover, Mutation and Replacement. Figure 1 showing GA cycle [5].

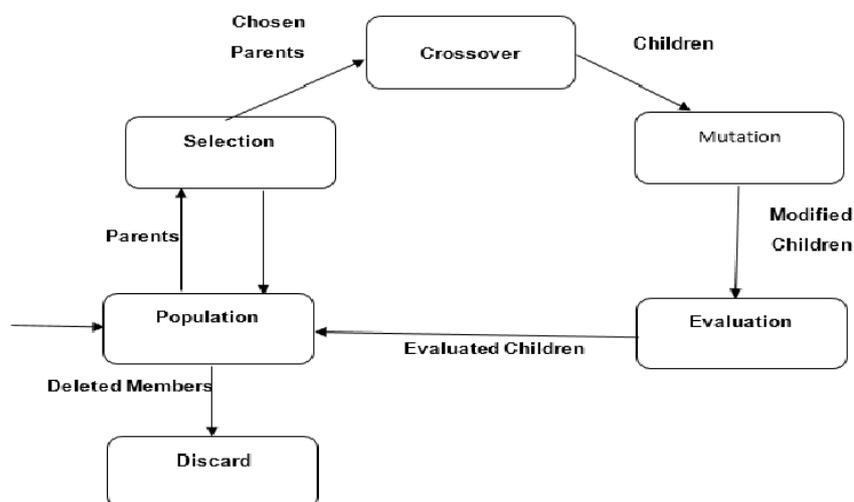


Figure 1 GA cycle

2.2. Travel Salesman Problem (TSP):

TSP is defined as an assignment to discover the shortest Hamiltonian cycle or path in complete graph of L nodes. The aim is to find out a visit of a specified number of cities (visiting each city precisely once) where the length path of this visit is minimized [5].

3. LITERATURE SURVEY

To update the distribution index , A self-adaptive procedure is proposed by Umbarkar, A.J. and Sheth, P.D. [1], to be used in the simulated binary crossover propose, this crossover is also good for multi-objective optimization problem.

Pellerin, E., Pigeon, L. and Delisle, S. [6], propose an approach to encode the control of a GAs parameters within every individual's chromosome. The parameters values are totally reliant on on the evolution mechanism and on the problem context, the results in this paper show that a GA can learn and assess the quality of self-set parameters according to their role in solving problem. These results indicate a promising approach to the development of GAs with self-adaptive parameter settings. The user does not require to pre-adjust parameters at the outset according to that approach, Tree Adjoining Grammar Guided Genetic Programming (TAG3P) is a new algorithm for self-adapting crossover and mutation rates in the specific genetic programming is proposed by Nam L., Hao H., and Canh V. [7]

Rajan K. [8] developed a new adaptive technique and a new crossover operator. It is based on the conjugative gradient technique. That technique is applied in adaptive GA and applications of adaptive GA for two types of real life problems. The algorithm is even useful for finding the solution of Travelling Sales Man Problem, one can see that the present algorithm can produce, within a short execution time the best solutions for TSP. The efficacy of the algorithm is shown with 15 cities and 40 cities TSP.

Deb, K. and Beyer, H.G [9] have developed a self-adaptive GA. This paper shows the self-adaptive feature of real-parameter genetic algorithms (GAs) while using the simulated binary crossover (SBX) operator. The self-adaptive performance of real parameter GAs is demonstrated on several test problems commonly-used in the ES literature.

4. THE PROPOSED SELF-ADAPTATION ALGORITHMS

In this section, the suggested algorithm is illustrated in Algorithm1;

Algorithm 1.

Input: the size of the population, the maximum number of genetic cycles , the total number of cities and a matrix named Cost that contains the distance between two cities.

Output: the optimization solution of TSP using self-adaptation algorithms (the best solution is the least distance).

Setting the parameters:the required parameters are set to identify the problem to be resolved and include: the size of the population, the maximum number of genetic cycles , the total number of cities and a matrix named **Cost** that contains the distance between two cities.

Generating a random population:the population is designed to consist of a group of individuals represented by chromosomes. Each chromosome in turn contains two parts

- 1) The first part presents the problem to be solved in this paper and includes all the cities that must be visited by the traveling salesman. length is the number of all cities to be visited.

The second part represents a gene for one of the genetic operators (selection, crossover, mutation, replacement), their initial values is calculated randomly during implantation.

This will be explained further in section 4.1

After the initialization of the GA, the evaluation of suitability for every individual in the population starts as the following: the first part of the chromosome in this problem are

evaluated according to the distance taken by the traveling sales man (the best solution is the least distance).

Fitness = Distance

Where Distance = $\sum_{i=0}^n Cost[gen[i], gen[i + 1]]$

Where Cost is the matrix that contains the distance between two cities.

i is the index of genes of chromosome

Gen [i] is the city number in the gene of chromosome.

N is the length of first part of chromosome.

Selection: select randomly the first individual from the population, the second individual is recognized using the gen selected from the second part of the chromosome of the first individual.

Crossover: Crossing of two individuals by using the parameters of the selected individual with best fitness to form new offspring.

Mutation of new children: This is achieved by using the parameter in the individual with the best fitness.

Evaluation: Estimate the fitness of each children.

Replacement: Substitute two children instead of two individuals in the population using the parameter in the child with the best fitness.

Stopping point: The implementation of the replication of the genetic algorithm depends on one of the following conditions:

- Access to the maximum number of generations previously identified.
- When the whole population becomes a solution

Table (1) shows the types of operators used in the proposed self –adaption method In this experiment, 3 selection operators, 4 crossover operators, 2 mutation operators and 4 replacement operators are chosen .

The mutation probabilities and crossover are randomly set in the algorithm. The random numbers values are in the range [0.0, 1.0]. The number of cities is variable. the genetic operators to steady state genetic algorithm are applied to test the performance of our approach.

Table 1 genetic operators in steady state genetic algorithm

Selection operator (Os)	Binary Tournament selection Triple Tournament Selection Boltzmann Tournament Selection
Crossover operator (Oc)	One-point crossovers Two points crossovers Uniform crossovers Partial Match Crossover (PMX)
Mutation operator (Om)	One mutation at random position Two mutations at random position
Replacement operator (Or)	Binary Tournament Replacement Triple Tournament replacement Weak Parent Replacement Random Replacement

4.1. Generate random population Details

Suppose that the total number of cities visited by the travelling salesman = 20. Thus, the number of genes in the chromosome equals 20 genes. Each gene is the city number and represents an integer number between 0 and 19, in addition to four genetic operators, hence the length of the chromosome equals 24 genes. Figure 2 represents the chromosome shape of the example above:

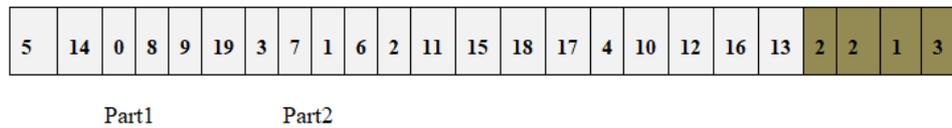


Figure 2 The proposed coding method for the chromosome

The first part of the chromosome represents the path of the traveling salesman. The second part of the chromosome represents genetic factors:

The first gene represents the type of selection. Here in this example, the value of 2 is the Triple Tournament Selection method as shown in Table 1.

The second gene represents the type of crossover. Here in this example, the value of 2 is the two point crossover method as shown in Table 1.

The third gene represents the type of mutation. Here in this example, the value of 1 is the one mutation at a random position method as shown in Table 1.

The fourth gene represents the type of replacement. Here in this example, the value of 3 is the Weak Parent Replacement method as shown in Table 1.

4.2. Experimental Results

From experiments with the TSP and self-adaptive GA some preliminary results were gained to learn the best set of parameters for this specific problem. The choice between various genetic operators increases the performance of the GA, in comparison to the steady state GA. Table (2) shows a comparison between 10 runs of the traditional steady state GA and the suggested self-adaptation steady state GAs. The results show that GA with self-adaptation is better than the TSSGA when trying to find the best solution, knowing that the number of cities equal 20. Please note that the distance is the solution to the traveling salesman problem

Figure (3) shows a comparison the average of number of generations between self-adaptive of steady state genetic algorithm and the TSSGAs for more than one example of the problems of the traveling salesman and each problem in several different cities. The results shows that self-adaptive better than the TSSGAs in finding the appropriate result.

Table 2 Comparison between Traditional steady state genetic algorithm and the suggested self-adaptation genetic algorithm

Traditional steady state genetic algorithm.(TSSGA)										
(size of the population = 120)										
Distance	41	40	38	36	40	42	40	41	39	39
NO. of generation	432	459	402	392	463	501	460	376	425	384
self-adaptive parameters with steady state Genetic algorithms										
(size of the population = 120)										
Distance	38	36	39	36	36	36	36	36	40	36
NO. of generation	360	304	342	273	282	320	305	294	337	355

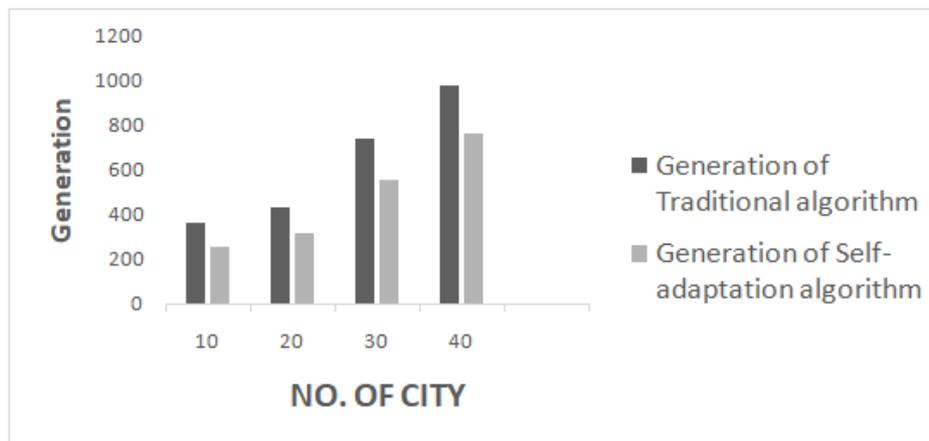


Figure 3 comparison the average of number of generation between self-adaptive of steady state genetic algorithm and TSSGA.

5. CONCLUSION

The suggested method can be characterized by the following factors:

- Reducing the number of inputs for the GA, thus providing ease and clarity of implementation for the user who doesn't need to enter the GA operators because it is randomly generated. Thus, people with little experience in the genetic algorithm can easily use genetic algorithms
- Reducing number of the generations of GA.
- In the future it is possible to employ the proposed algorithm to solve more complex problems
- The self-adaptive genetic algorithm, which was designed to enable us to use all types of selection, crossover, replacement and mutation within the problem to be solved. The results also show that the self-made choice between several genetic operators increases the performance of the GA.

REFERENCES

- [1] Umbarkar, A.J. and Sheth, P.D., 2015. CROSSOVER OPERATORS IN GENETIC ALGORITHMS: A REVIEW. *ICTACT journal on soft computing*, 6(1).
- [2] Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y. and Gao, Z., 2017. A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing*, 256, pp.56-62.
- [3] Abdoun, O., Tajani, C., Abouchabaka, J. and El Khatir, H., 2016. Improved genetic algorithm to solve asymmetric traveling salesman problem. *Int. J. Open Problems Compt. Math*, 9(4), pp.42-55.
- [4] Kalsi, S., Kaur, H. and Chang, V., 2018. DNA Cryptography and Deep Learning using Genetic Algorithm with NW algorithm for Key Generation. *Journal of medical systems*, 42(1), p.17.
- [5] Deng, Y., Liu, Y. and Zhou, D., 2015. An improved genetic algorithm with initial population strategy for symmetric TSP. *Mathematical Problems in Engineering*, 2015.
- [6] Pellerin, E., Pigeon, L. and Delisle, S., 2004, April. Self-adaptive parameters in genetic algorithms. In *Data Mining and Knowledge Discovery: Theory, Tools, and Technology VI* (Vol. 5433, pp. 53-65). International Society for Optics and Photonics.

- [7] Nam L., Hao H., and Canh V. , 2015 . Self-adaptive crossover and mutation parameters in tree adjoining grammar guided genetic programming . Tạp chí Khoa học và Kỹ thuật - Học viện KTQS - Số 167 (4-2015)
- [8] Rajan, K., 2014. Adaptive techniques in genetic algorithm and its applications. Thesis submitted to. mahatma Gandhi university, Kottayam. for the degree of. Doctor of philosophy in mathematics, 24-Jul-2014
- [9] Deb, K. and Beyer, H.G., 1999, July. Self-adaptation in real-parameter genetic algorithms with simulated binary crossover. In Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1 (pp. 172-179). Morgan Kaufmann Publishers Inc.
- [10] Prof. P. Ramanjaneyulu, Prof. Dr. V.C. Veera Reddy. An Enhanced Genetic Algorithm Approach for Optimal Placement of Facts Devices To Enhance Atc, *International Journal Of Electrical Engineering & Technology*, 5(4), (2014), pp. 96-103
- [11] Prof. Sharadindu Roy, Prof. Samar sen Sarma., An Approach to Solve Graph Partitioning Problem Using Genetic Algorithm, *International Journal of Computer Engineering & Technology*, 4(6), (2013), pp. 204-211
- [12] Manish Sharma and Mr. Rahul Patel, An Improving The Effectiveness of Text to Matrix Generator By Using Genetic Algorithm *International Journal of Computer Engineering & Technology*, 5(1), (2014), pp. 162-167
- [13] Geetkiran Kaur and Parvinder Kaur, A New Approach to Context Aware Saliency Detection Using Genetic Algorithm, *International Journal of Computer Engineering & Technology*, 4(5), (2013), pp. 67-75
- [14] Harshdeep Sharma and Gianetan Singh Sekhon, A Review on Load Balancing in Cloud Using Enhanced Genetic Algorithm. *International Journal of Computer Engineering & Technology*, 8(2), 2017, pp. 23–29
- [15] Şengül CAN and Mustafa GERŞİL, A Literature Review on the Use of Genetic Algorithms In Data Mining. *International Journal of Computer Engineering Technology*, 8(6), 2017, pp. 67–76
- [16] Mhd. Furqan, Herman Mawengkang, Opim Salim Sitompul, A.P.U. Siahaan, Muhammad Donni Lesmana Siahaan, Wanayumini, Nazaruddin Nasution, Sriani, A Review of Prim and Genetic Algorithms in Finding and Determining Routes on Connected Weighted Graphs. *International Journal of Civil Engineering and Technology*, 9(9), 2018, pp. 1755-1765