

The Traveling Salesman Problem as a Benchmark Test for a Social-Based Genetic Algorithm

Naghah Azmi AL-Madi and Ahamad Tajudin Khader
School of Computer Sciences, University Sains Malaysia, 11800 Penang, Malaysia

Abstract: Present research, showed Traveling Salesman Problem (TSP) to compare values and results concerning the averages and best fits of both, Simple Standard Genetic Algorithm (SGA) and a new approach of Genetic Algorithms named Social-Based Genetic Algorithm (SBGA). Results from TSP test on Social-Based Genetic Algorithms (SBGA) were presented. Results were encouraging in Social-Based Genetic Algorithms (SBGA) performs better in finding best fit solutions of generations in different populations than the Simple Standard Genetic Algorithm.

Key words: Genetic Algorithms (GAs), Traveling Salesman Problem (TSP), Symmetric TSP, Social-Based Genetic Algorithm (SBGA)

INTRODUCTION

Genetic algorithms are adaptive methods, which may be used to solve search and optimization problems^[1,2]. John Holland and his students were the beginners to develop these kinds of algorithms in the early 1960s, 1970s and 1980s^[2-3,4-8]. They are considered powerful search techniques as, they solve hard complex problems in various disciplines and they rely mainly on the biological process of evolution^[3,4,7,39]. As a matter of fact, Genetic Algorithms (GAs) are routines which could manage self adoption, same as neural networks. They mimic nature in a way that the survival of the fittest is to provide new generations, of approximate solutions^[4,7]. Genetic Algorithms (GAs) work with various elements "individuals", each element is called a chromosome or genotype. A fitness score is given to each individual indicating and representing a possible solution, to a given problem^[1-8]. In fact, Genetic Algorithms (GAs) were first used in solving academic problems such as the traveling salesman problem and the 8 Queens problem^[3,4,5,8]. Years later, Genetic Algorithms (GAs) increased their applications to optimize many types of complex problems such as the complex scheduling problems, spatial layout and many other problems that are hard to efficiently solve^[6].

The Traveling Salesman Problem (TSP): The Traveling Salesman Problem (TSP) is considered to be an important combinatorial problem, because of its simplicity to define^[23-26]. It is stated as an NP-hard

optimization problem. It consists of n cities which must be visited by a salesman, starting from one of them passing through each city only once and returning to the first city. The cost is given for the journey, which is referred to as a tour. Finally, the minimum cost is required to solve this problem^[9,22,27-29].

The Traveling Salesman Problem (TSP) is determined as follows: Given N cities, known as nodes, a distance matrix where, $D = [d_{ij}]$, consists of the distance between city i and city j ^[23,27-29].

In an attempt to finding near optimal solutions for NP-hard problems; the Traveling Salesman Problem (TSP) is considered a standard benchmark problem for combinatorial methods^[28]. It provides a standard optimization test bed, to find near optimum solutions to NP-hard problems^[1,30,35,37].

It is to be known, that, the Traveling Salesman Problem (TSP) is called Symmetric TSP (Standard), if the cost between any two cities are equal in both directions, that means, the distance from city $i-j$ is the same as the distance from city $j-i$. Otherwise, the Traveling Salesman Problem (TSP) is to be known as an Asymmetric TSP, which means that the distance between city $i-j$, differs than the distance from city $j-i$ ^[23,26,30].

Two alternative approaches exist to solve the Traveling Salesman Problem (TSP). First, is to find its solution and try proving its optimality, this takes a long period of time, which might take years. The second, approach, is to find an approximate solution in a short period of time^[26].

Applying the Traveling Salesman Problem using methods from many specific areas mostly based on search heuristic methods such as local search^[33,35], simulated annealing^[31,36], tabu search^[32,37], neural networks^[31,34] and genetic algorithms^[33,37]. Apparently, there are wide applications of the TSP, such as, computer cabling, traffic route, robot control and many others^[24,26].

MATERIALS AND METHODS

The most important part in the Simple Standard Genetic Algorithm (SGA) is selecting parents. But there are no constraints in this selection so the Simple Standard Genetic Algorithm (SGA) works randomly^[10]. Due to this randomness, many researches are working to tackle this problem by designing structured population and putting some constraints to control the individual's interaction^[10].

In the last few years many types and models of GAs appeared such as the Cellular GA^[10], Island GA^[12], Patchwork GA^[13,14], Terrain-Based GA^[14] and religion-Based GA^[15].

Cellular GAs (CGA): A diffusion model of a two-dimensional grid in which each individual interacts with another by its direct neighbor^[16], by Gorges-Schleuter^[10]. The genetic algorithm is designed as a probabilistic cellular automation in this type of GAs. These individuals will be distributed on a graph which is connected together, having a neighborhood of some genetic operator to work with. This type of GAs is designed as a probabilistic cellular. A self-organizing schedule is added to reproduce an operator^[17]. The individual which can interact with its immediate neighbors can only be held in the cell.

Terrain-Based GA (TBGA): In a comparison between the Terrain-Based GA (TBGA) and the Cellular GA (CGA), the first shows a more self-tuning model in which many combination parameter values will be located in different physical locations and better performance with less parameter tuning than the second^[14]. At every generation each individual should be processed and the mating will be selected from the best of four strings, located above, below, left, right.

Patchwork model: Krink *et al.*^[12] introduced this model which consists of several ideas merged together from cellular evolutionary algorithms, island models and traditional evolutionary algorithms^[13]. Here the grid is a two dimensional grid of fields, each field can have a fixed number of individuals. The patchwork model is considered a self-organized, spatial population structure^[18]. In a GA population, in order to allow self-

adaptation, patchwork model is used as a base. It contains a grid world and some interesting agents. In modeling biological systems the patchwork model is considered as a general approach.

Island models: Island models are considered a family of more advanced models of evolutionary algorithms (EAs)^[19]. These models were developed in order to solve more complex problems which are increasing rapidly. Here the individuals are divided into sections. We call each section a subpopulation which is referred to as an island. These island models are able to solve problems in a better performance than standard models^[18,21]. There is a specific relation between islands through some exchange of some individuals between islands. This process is called migration; this is what island models are famous of and without these migrations, each island is considered as a set of separate run. Therefore migration is very important^[19,21].

Religion-Based EA Model (RBEA): This model has a religious concept introduced by Rene Thomsen *et al.*^[15]. The Religion-Based EA Model (RBEA) attracts new believers to a religion which puts more control than other models such as cellular EA and the patchwork models^[15].

Social-Based Genetic Algorithm (SBGA): The Social-Based Genetic Algorithm (SBGA) as presented in^[5] is an approach to control the randomness of the Simple Standard GA. Social constraints applied to this new approach were affective.

Figure 1 shows the model of the Social-Based Genetic Algorithm (SBGA)^[5].

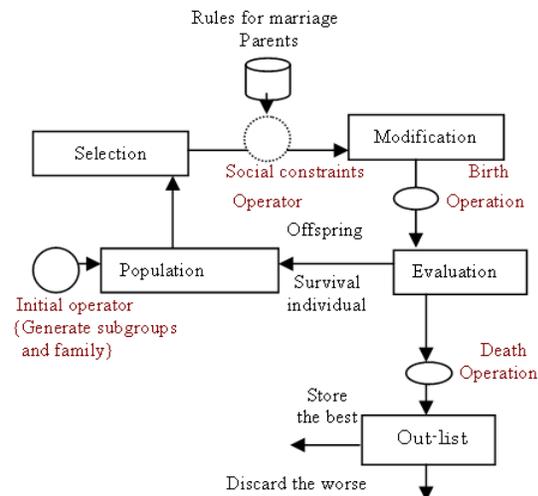


Fig. 1: The Social-Based Genetic Algorithm (SBGA) model design "The Simple Standard GA (SGA) modified by new operators"^[5]

Representing the chromosome in the SBGA:

According to the Social-Based Genetic Algorithm (SBGA)^[5] which is based on nature and social selection, an attribute is given to each individual in the population specifying its sex whether male or female. In addition, being in the same society as the population is divided into subgroups or islands is a dependable constraint for recombination. The problem of age is considered also by adding an attribute for the age. The age attribute takes three values: Youth, parent and grandparent. This chromosome representation (the presence of father and mother pointers) will keep all family relations which divides the subgroups into a Directed Acyclic Graph (DAG). All the standard operations in the GA will be changed in order to add restrictions on each operation including: Social constraints such as the male/female 'operator', this will be added in the selection part which will restrict choosing two different couples. In addition the Birth operator which is generating a new population and the Death operator which will discard the worse individuals.

In Fig. 1, the Social-Based Genetic Algorithm (SBGA) model which is a modification of the Simple Standard Genetic Algorithm (SGA) is shown. And all the standard operations in the SGA will be changed in order to add restrictions on each.

The SBGA method: Initially, the first individual is selected randomly from the population this will be the first parent. Based on the first parent's type (whether a male or a female), the second parent will be chosen such that it is the opposite type of the first parent. This process is repeated for a number of individuals creating the initial population. Next comes to the stages of selection and crossover, bringing up two new children or offspring's. Repeating this for a number of couples a second population will be generated. Again, the previous process is repeated until the maximum number of generations is reached. (The next main important thing is that the two individuals must not share the same parents).

RESULTS AND DISCUSSION

In this research we have used the Traveling salesman Problem (TSP) to test the Social-Based Genetic Algorithm Model (SBGA) in^[5]. We also used it as a test on the Simple Standard Genetic Algorithm (SGA) in order to compare between both algorithms.

A population size of 350 with seven cities and a randomly selected one-point crossover are used in a process that is both standard and simple^[38]. A random

integer (crossover point) and a crossover rate of 5% are chosen according to the maximum length of the chromosome in the model. This is the place in the chromosome at which, with probability, the crossover will occur. If the crossover does occur, then the bits up to the random integer of the two chromosomes are swapped. The mutation of a solution is a random change to a gene value^[38]. After several experiments of different mutation rates, the most suitable mutation rate is 0.04. The selection method used is the roulette wheel. The number of generations is 100. The implementation part was programmed in C# (C Sharp) language version (5.0) on a pentium 4, HP-compaq laptop.

By applying the Traveling Salesman Problem (TSP) on both the Simple Standard Genetic Algorithm (SGA) and on the Social-Based Genetic Algorithm (SBGA)^[5] we can compare the performance of both algorithms. The following comparisons below will show that the constraints put on the new Social-Based Genetic Algorithm (SBGA) gave better performance to SBGA than the Simple Standard Genetic Algorithm (SGA) which depends mainly on its randomness in finding the best fit solution. It is shown that in the Social-Based Genetic Algorithm (SBGA) the average converge toward the optimal solution better than the Simple Standard Genetic Algorithm (SGA) and the best fit values in the Social-Based Genetic algorithm (SBGA) also show better findings of best fit values in a comparison to the basic Simple Standard Genetic.

In Fig. 2 and 3, we can show the comparative results of applying the Traveling Salesman Problem (TSP) on both the Simple Standard Genetic Algorithm (SGA) and the Social-Based Genetic Algorithm (SBGA).

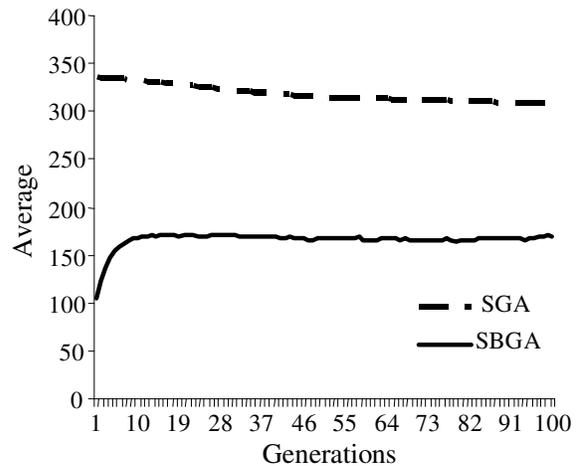


Fig. 2: The Total Average of Averages of 10 runs each between SGA and SBGA with seven cities

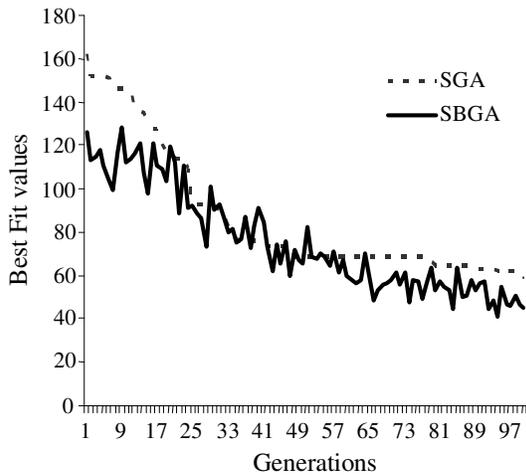


Fig. 3: The Total Average of Bestfits of 10 runs each between SGA and SBGA with seven cities

Figure 2 and 3 show that the average of the Social-Based Genetic Algorithm (SBGA) has a better performance than the Standard Genetic Algorithm (SGA) towards the minimum. In addition, they show a better finding of best fit solutions for the Social-Based Genetic Algorithm (SBGA) than the Standard Genetic Algorithm (SGA).

CONCLUSION

In this research, a test function of the Traveling Salesman Problem (TSP) is used to evaluate and compare results between the Simple Standard Genetic Algorithm (SGA) and a new approach for structured population of GA called the Social-Based Genetic Algorithm (SBGA)^[5].

It is concluded based on the analysis results that the new approach the Social-Based Genetic Algorithm (SBGA) is better in terms of best finding as shown in our given results than the Simple Standard Genetic Algorithm (SGA). The Average of the Social-Based Genetic Algorithm (SBGA) is trying to converge towards the minimum despite its restricted constraints to the best values. In addition, the findings of the best solutions of best fit values are in a better condition in the Social-Based Genetic Algorithm (SBGA) than in the Simple Standard Genetic Algorithm (SGA).

ACKNOWLEDGMENT

The researchers would like to thank the University Sains Malaysia (USM) for its support to this study.

REFERENCES

1. Beasley, D., D.R. Bull and R.R. Martin, 1993. An overview of genetic algorithms: Part 1, Research Topics. Univ. Comput., 15: 58-69. <http://www.geocities.com/francorbusetti/gabeasley1.pdf>
2. Holland, H.J., 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. 1st Edn., MIT Press, Cambridge, MA., USA., ISBN: 10: 0262581116, pp: 228.
3. Zheng, Y. and S. Kiyooka, 1999. Genetic algorithm applications. www.me.uvic.ca/~zdzong/courses/mech620/GA_App.PDF
4. Liao, Y. and C.T. Sun, 2001. An educational genetic algorithms learning tool. IEEE Trans. Educ., 44: 20. DOI: 10.1109/13.925863
5. AL-Madi, A.N. and A.T. Khader, 2007. A social-based model for genetic algorithm. Proceedings of the 3rd International Conference on Information Technology, (ICIT'07), AL-Zaytoonah University, Amman, Jordan.
6. Dalton, J., 2007. Genetic Algorithms. <http://www.edc.ncl.ac.uk/highlight/rhjanuary2007.php/>
7. Bagheri, E. and H. Deldari, 2006. Dejong function optimization by means of a parallel approach to fuzzified genetic algorithm. Proceedings of the 11th IEEE Symposium on Computers and Communications, June 26-29, IEEE Computer Society, Washington DC., USA., pp: 675-680. DOI: 10.1109/ISCC.2006.57
8. Back, T., 1996. Evolutionary Algorithms in theory and practice Evolution Strategies, Evolutionary Programming, Genetic Algorithms. 1st Edn., Oxford University Press, USA., ISBN: 0195099710, pp: 328
9. Lamom, A., T. Thepchatri and W. Rivepiboon, 2008. Heuristic algorithm in optimal discrete structural designs. Am. J. Applied Sci., 5: 943-951. http://findarticles.com/p/articles/mi_7109/is_8_5/ai_n28552361/pg_6
10. Gorges-Schleuter, M., 1989. ASPARAGOS an asynchronous parallel genetic optimization strategy. Proceedings of the 3rd International Conference on Genetic Algorithms, 1989, Morgan Kaufmann Publishers Inc., San Francisco, CA., USA., pp: 422-427. <http://portal.acm.org/citation.cfm?id=93126.94041&coll=ACM&dl=ACM>
11. Back, T., D.B. Fogel and Z. Michalewicz, 1997. Handbook on Evolutionary Computation. Ringbound Edn., IOP Publishing Ltd. and Oxford University Press, ISBN: 10: 0750303921, pp: 988.

13. Krink, T., B.H. Mayoh and Z. Michalewicz, 1999. 2A Patchwork model for evolutionary algorithms with structured and variable size populations. Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers Inc., San Francisco, CA., USA., pp: 1321-1328. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.170>
13. Krink, T. and R.K. Ursem, 2000. Parameter control using the agent based patchwork model. Proceedings of the 2nd Congress on Evolutionary Computation, July 16-19, IEEE Computer Society, Washington DC., USA., pp: 77-83. DOI: 10.1109/CEC.2000.870278
14. Gorden, V.S., R. Pirie, A. Wachter and S. Sharp, 1999. Terrain-based genetic algorithm (TBGA): Modeling parameter space as terrain. Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17, Orlando, Florida, USA., pp: 299-235. [http://bibnetwiki.org/wiki/Terrain-Based_Genetic_Algorithm_\(TBGA\):_Modeling_Parameter_Space_as_Terrain](http://bibnetwiki.org/wiki/Terrain-Based_Genetic_Algorithm_(TBGA):_Modeling_Parameter_Space_as_Terrain)
15. Thomsen, R., P. Rickers and T. Krink, 2000. A religion-based spatial model for evolutionary algorithms. Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, Sep. 18-20, Springer-Verlag, London, UK., pp: 817-826. <http://portal.acm.org/citation.cfm?id=645825.668942>
16. Whitley, D., 1993. Cellular genetic algorithms. Proceeding of the 5th International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA., USA., pp: 658. <http://portal.acm.org/citation.cfm?id=657598>
17. Alba, E., M. Giacobini and M. Tomassini, 2002. Comparing synchronous and asynchronous cellular genetic algorithms. Lecture Notes Comput. Sci., 2439: 601-610. <http://www.springerlink.com/content/bfnc36jllk6vxvvy/>
18. Ursem, R.K., 1999. Multinational evolutionary algorithms. Proceedings of the Congress of Evolutionary Computation, July 6-9, IEEE Computer Society, Washington DC., USA., pp: 1633-1640. DOI: 10.1109/CEC.1999.785470
19. Zbigniew S. and K. De Jong, 2005. The influence of migration sizes and intervals on island models. Proceedings of the Conference on Genetic and Evolutionary Computation, June 25-29, ACM Press, New York, USA., pp: 1295-1302. <http://portal.acm.org/citation.cfm?id=1068009.1068219>
20. Belal, M.A. and I.H. Khalifa, 2002. A comparative study between swarm intelligence and genetic algorithms. Egypt. Comput. Sci. J., 24. http://net.shams.edu.eg/ecs/jan_02_a2.htm
21. Babbar, M., B. Minsker and D.E. Goldberg, 2002. A multiscale island injection genetic algorithm for optimal groundwater remediation design. Proceedings of the Conference on Water Resources Planning and Management, (CWRPM'02), American Society of Civil Engineers (ASCE) Environmental and Water Resources Institute (EWRI), Roanoke, VA.
22. Wei, J. and D.T. Lee, 2004. A new approach to the travelling salesman problem using genetic algorithms with priority encoding. Proceeding of the Congress on Evolutionary Computation, June 19-23, IEEE Computer Society, Washington DC., USA., pp: 1457-1464. DOI: 10.1109/CEC.2004.1331068
23. Nguyen, H.D., I. Yoshihara, K. Yamamori and M. Yasunaga, 2007. Implementation of an effective hybrid GA for large-scale traveling salesman problems. IEEE Trans. Syst. Man Cybernet Part B., 37: 92-99. DOI: 10.1109/TSMCB.2006.880136
24. Xuan, W. and Y. Li, 2005. Solving traveling salesman problem by using a local evolutionary algorithm. Proceeding of the IEEE International Conference on Granular Computing, July 25-27, IEEE Computer Society, Washington DC., USA., pp: 318-321. DOI: 10.1109/GRC.2005.1547294
25. Lee, Z.J., 2004. A hybrid algorithm applied to traveling salesman problem. Proceedings of the IEEE International Conference on Networking, Sensing and Control, Mar. 21-23, IEEE Computer Society, Washington DC., USA., pp: 237-242. DOI: 10.1109/ICNSC.2004.1297441
26. Yang, R., 1997. Solving large traveling salesman problems with small populations. Proceeding of the Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, Sep. 2-4, IEEE Computer Society, Washington DC., USA., pp: 157-162. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=681004
27. Smith, K., 1996. An argument for abandoning the traveling salesman problem as a neural network benchmark. IEEE Trans. Neural Networks, 7: 1542-1544. DOI: 10.1109/72.548187
28. Pullan, W., 2003. Adapting the genetic algorithm to the traveling salesman problem. Proceeding of the Congress on Evolutionary Computation, Dec. 8-12, IEEE Computer Society, Washington DC., USA., pp: 1029-1035. DOI: 10.1109/CEC.2003.1299781
29. Jung, S. and B.R. Moon, 2002. Toward minimal restriction of genetic encoding and crossovers for the two-dimensional euclidean TSP. IEEE Trans. Evolut. Comput., 6: 557-565. DOI: 10.1109/TEVC.2002.804321

30. White, C.M. and G. Yen, 2004. A hybrid evolutionary algorithm for traveling salesman problem. Proceeding of the Congress on Evolutionary Computation, June 19-23, IEEE Computer Society, Washington DC., USA., pp: 1473-1478. DOI: 10.1109/CEC.2004.1331070
31. Budinich, M., 1996. A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing. *Neural Comput.*, 8: 416-424. DOI: 10.1162/neco.1996.8.2.416
32. Liu, G., Y. He, Y. Fang and Y. Oiu, 2003. A novel adaptive search strategy of intensification and diversification in tabu search. Proceedings of the IEEE International Conference on Neural Networks and Signal Processing, Dec. 14-17, IEEE Computer Society, Washington DC., USA., pp: 428-431. DOI: 10.1109/ICNNSP.2003.1279300
33. Bianchi, L., J. Knowles and J. Bowler, 2005. Local search for the probabilistic traveling salesman problem: Correction to the 2-P-opt and 1-shift algorithms. *Eur. J. Operat. Res.*, 162: 206-219. <http://cat.inist.fr/?aModele=afficheN&cpsidt=16436338>
34. Leung, S.K., D.H. Jin and B.Z. Xu, 2004. An expanding self-organizing neural network for the traveling salesman problem. *Neurocomputing*, 62: 267-292. DOI: 10.1016/j.neucom.2004.02.006
35. Reinelt, G., 1994. The travelling salesman: Computational solutions for TSP applications. *Lecture Notes Comput. Sci.*, 840. DOI: 10.1007/3-540-48661-5
36. Laarhoven, P.V. and L.E.H. Aarts, 1987. *Simulated Annealing: Theory and Applications*. 1st Edn., Springer-Verlag, London, UK., ISBN: 10: 9027725136, pp: 204.
37. Fiechter, L., 1994. A parallel tabu search algorithm for large travelling salesman problems. *Discrete Applied Math. Combinat. Operat. Res. Comput. Sci.*, 51: 243-267. DOI: 10.1016/0166-218X(92)00033-I
38. Prebys, E.K., 2007. The genetic algorithm in computer science. *MIT Undergrad. J. Math.*, 2007: 165-170. <http://www-math.mit.edu/phase2/UJM/vol1/PREBYS-F.PDF>
39. Godefroid, P. and S. Khurshid, 2002. Exploring very large state spaces using genetic algorithms. *Lecture Notes Comput. Sci.*, 2280: 266-280. <http://cat.inist.fr/?aModele=afficheN&cpsidt=14052994>