Review Paper on Genetic Algorithm and Simulated Annealing for solving Travelling Salesman Problem

Shivani Vats[1], Nikita Bakshi[2]
[1,2]*Department of Computer Science Engineering, UIET Lalru, SAS Nagar, India*

Abstract- **The Travelling Salesman problem is one of the very important problems in Computer Science and Operations Research. It is used to find the minimum cost of doing a work while covering the entire area or scope of the work in concern. In this paper we will review the past work done in solving the travelling salesman problem using two different techniques- genetic algorithm, simulated Annealing algorithm.**

Index Terms- Travelling Salesman Problem, Genetic Algorithm, Simulated Annealing, SAGA.

INTRODUCTION

Genetic Algorithm (GA) and Simulated Annealing (SA) have been used to find optimization solutions. Both GA and SA search a solution space throughout a sequence of iterative states. However, there are also significant differences between them. The GA mechanism is parallel on a set of solutions and exchanges information using the crossover operation. SA works on a single solution at a time. GA uses the same selection strategy during the run of the algorithm, while SA regulates the temperature parameter it uses to evaluate the solution. These differences lead to different search criteria. Both GA and SA have advantages and disadvantages. A disadvantage of GA is that it is time-consuming to find an optimal solution or it might trap at local minima. In case of SA only one candidate solution is used, thus it does not build up an overall view of the search space.

Also, SA is slow because of its sequential nature. In other words, SA can find good quality solutions in a neighbourhood, but most likely it will get trapped in local minima and takes longer to escape, while GA rapidly discovers the search space, but has difficulty finding the exact minima. The hybridization of SA and GA tries to combine the advantages of GA. In this work GA and SA are combined in order to improve the quality of solutions and to minimise the execution time.

1. Genetic Algorithm for TSP: - The problem is very simple. The idea of the travelling salesman problem (TSP) is to find the best tour from given number of cities or locations, visiting each city exactly once and returning to the starting city where the length of this tour is minimized. Genetic algorithms are an optimization technique based on natural evolution. They comprise the continued existence of the fittest idea into a search algorithm which give a way of searching and do not need to elaborate each possible solution in the feasible region to obtain a good result.

GA is based on a similarity with the behaviour of chromosomes within a population of persons and the genetic structure using the following foundations:

i. Individuals in an inhabitants race for resources and mates.

ii. Those individuals who are most successful in each 'competition' will produce more offspring and individuals who perform poorly will produce more offspring.

iii. When the genes from `good' individuals spread throughout the population then two good parents will sometimes produce offspring that are better than either parent.

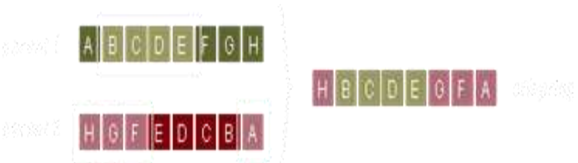
iv. In this way each successive generation will become more suited to the respective environment.

Implementation Details:-Based on Natural Selection the algorithm evolves the through three operators (described as under) after an initial population is randomly generated.

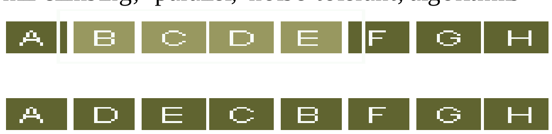
1. First is “Selection Operator” which equates to survival of the fittest;
2. Another operator is “Crossover” which represents mating between individuals;
3. Mutation which introduces random modifications is the last operator.

1. Selection Operator: - This operator gives preference to better individuals, allowing them to pass on their genes to the next generation. Fitness is the main factor to decide whether each individual is good and may be determined by a subjective judgement or by an objective function

2. Crossover Operator:-Crossover operator is key distinguished factor of GA from other optimization method. Using the selection operator two individuals are chosen from the population. A crossover site along the bit strings is arbitrarily selected. The values of the two strings are then exchanged up to this point. Now if S1=000000 and S2=111111 and the crossover point is 2 then S1'=110000 and S2'=001111.
By recombining portions of good individuals the two new offspring created from this mating are put into the next generation of the population. This process is likely to create even better individuals.



3. Mutation Operator:-Some time with some low probability, a portion of the new individuals will have some of their bits flipped. The purpose of mutation operator is to keep diversity within inhibit premature convergence and population. Mutation alone induces a random walk through the search space.
Mutation and selection (without crossover) create hill-climbing, parallel, noise-tolerant, algorithms



Algorithm

1. Randomly initialize population (t)
2. Determine fitness of population (t)
3. Repeat
   a. Select parents from population (t)
   b. Perform crossover on parents creating population (t+1)
   c. Perform mutation of population (t+1)
   d. Determine fitness of population (t+1)
4. Until best individual is good enough.

Structure of Genetic Algorithm Figure 1 depicts the very simple structure of Genetic Algorithm. GA begins with arbitrary beginning of a population using chromosomes as abstract presentations of solution candidates. Chromosomes encode feasible. The population evolves successively. A portion of population, which is evaluated as better individuals, is chosen for breeding. The selecting process is based on a fitness function, which delivers the fitness of each individual. The "breeding" or so called "reproduction" step of GA is applying natural genetic operators such as Selection, Crossover, and Mutation for each two chosen individuals.
The new child will share characteristics of its parents. The generational process will be stopped until the predefined criteria are met. The typical criteria can be the number of created generations, runtime of generation process, predefined level of fitness or the combination of all above mentioned criteria. Those criteria depend strongly on the real problem and real conditions, in which the resource like computational performance should be taken into account.

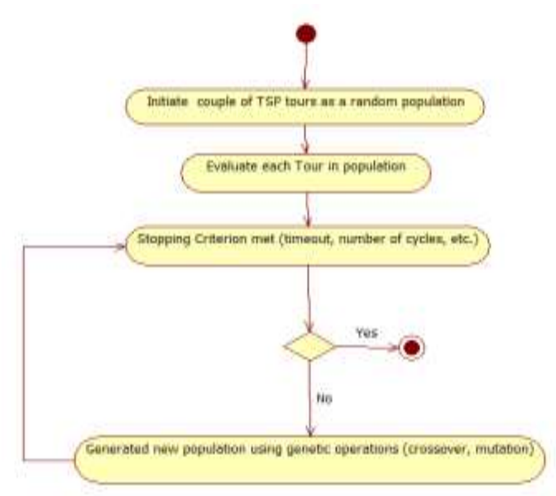


Figure 1: Simple Structure of Genetic Algorithm applied in TSP

SIMULATED ANNEALING

The purpose of physical annealing is to accomplish a low energy state of a solid. This is achieved by melting the solid in a heat bath and gradually lowering the temperature in order to allow the particles of the solid to rearrange them in a crystalline lattice structure. This structure corresponds to a minimum energy state for the solid. The initial temperature of the annealing process is the point at which all particles of the solid are arbitrarily arranged in the heat bath. At each temperature, the solid must reach what is known as thermal equilibrium before the cooling can continue. If the temperature is reduced before thermal equilibrium is achieved, a defect will be frozen into the lattice structure and the resulting crystal will not correspond to a minimum energy state. The Metropolis Monte Carlo simulation can be used to simulate the annealing method at a fixed temperature T. The Metropolis method randomly generates a sequence of states for the solid at the given temperature. A solid's state is characterized by the positions of its particles. A new state is generated by small movements of randomly chosen particles. The change in energy ΔE caused by the move is calculated and acceptance or rejection of the new state as the next state in the sequence is determined according to Metropolis acceptance condition. If ΔE< 0 the move is acceptable and if ΔE>0 the move is acceptable with probability, if >Ω The move is acceptable otherwise rejected, where Ω is random number and 0 <Ω< 1. Simulated annealing was first introduced by Metropolis et al, but it was Kirkpatrick et al, in 1983 who proposed SA as the basis of an optimization technique for combinatorial optimization problems. Simulated annealing is one of the most popular and general adaptive heuristic algorithms. Simulated annealing algorithms have been applied to solve numerous combinatorial optimization problems. The name and idea of SA comes from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat frees the atoms to move from their initial positions (initial energy). By slowly cooling the atoms the material continuously rearranges, moving toward a lower energy level. They gradually lose mobility due to the cooling, and as the temperature is reduced the atoms tend to crystallize into a solid. In the simulated annealing method, each solution in the search space is equivalent to a state of a physical system and the function f(s) to be minimized is equivalent to the internal energy of that state. The objective is to minimize the internal energy as such as possible. For successful annealing it is important to use a good annealing schedule, where the temperature reducing gradually.

SIMULATED ANNEALING ALGORITHM

Construct initial solution x0; xnow = x0
Set initial temperature T = TI
Repeat
For i = 1 to TL do
Generate randomly a neighbouring solution x′ ∈ N(xnow)
Compute change of cost ΔC = C(x′) - C(xnow)
if ΔC ≤ 0 then
xnow = x′ (accept new state)
else
Generate q = random(0,1)
if q < e-ΔC/T then xnow = x′ end if
end if
end for
set new temperature T = f(T)
until stopping criterion
return solution corresponding to the minimum cost function.

CONCLUSION

This work proposed a new approach to simulated annealing and a genetic algorithm (SAGA) for Travelling Salesman Problem in order to reap the benefits of SA and reduce the time that GA spends stuck at local minima. The proposed algorithms tend to produce better quality results in smallest amount of time.

FUTURE SCOPE

Genetic algorithms appear to find good solutions for the travelling salesman problem, however it depends very much on the way the problem is encoded and which crossover and mutation methods are used. It is observed that the technique which use heuristic information or encode the edges of the tour perform the best and give good indications for future work in this area. Genetic algorithms have not found the optimum result to the traveling salesman problem than is already known, but many of the already known best solutions have been found by some genetic algorithm method also.

Simulated annealing has great potential of finding an optimal result provided sufficient time is given for the annealing process. In fact a simple concept of annealing has significant effect on finding an optimal result. In the end we can say that simulated annealing is indeed a good motivation for other heuristic algorithms.

REFERENCES

[1] Goldberg, Addison Wesley D. E. “Genetic algorithms in Search, Optimization and Machine Learning” 1989

[2] Davis, L. (ed.). “Handbook of Genetic Algorithms: Van Nostrand Reinhold” 1991

[3] Gunnels, J., Cull, P. & Holloway, J. L. “Genetic Algorithms and Simulated Annealing for Gene Mapping”. Florida: IEEE, 1994

[4] Kylie Brant, Aurther Benjamin. Genetic Algorithm and the Travelling Salesman Problem. December 2000.

[5] Prof. Daniel Graupe&Ritesh Gandhi “Implementation of Travelling Salesman’s Problem Using Neural Network Final Project Report” (2001)

[6] Ghorbanali Mohammadi,” Using genetic algorithms to solve industrial time-cost trade-off problems”, Indian Journal of Science and Technology Vol. 4 No. 10 (Oct 2011) ISSN: 0974- 6846

[7] J. Majumdar, A.K. Bhunia "Genetic algorithm for asymmetric traveling salesman problem with imprecise travel times" 2011

[8] S. M. Chen, C. Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealingant colony system with particle swarm optimization techniques”,Expert System with application, vol 38, 2011

[9] Adewole Philip, Akinwale Adio Taofiki, Otunbanowo Kehinde, “A Genetic Algorithm for Solving Travelling Salesman Problem”. IJACSA, 2011

[10] Bert F. J. La Maire& Valeri M. Mladenov “Comparison of Neural Networks for Solving the Travelling Salesman Problem” 11th symposium on Neural Network in Electrical Engineering, Conference Paper (2012)

[11] Nitesh M. Sureja, Bharat V. Chawda “ Random Travelling Salesman Problem using SA” IJETAE 2012

[12] Naveen kumar, Karambir and Rajiv Kumar, “A Genetic Algorithm Approach To Study Travelling Salesman Problem”, Journal of Global Research in Computer Science, 2012, Vol. 3, No. (3).

[13] Z. Benkhellat and A. Belmehdi, "Genetic algorithms in speech recognition systems," in Proceedings of the International Conference on Industrial Engineering and Operations Management, 2012.

[14] Younis R. Elhaddad “Combined Simulated Annealing and GeneticAlgorithm to Solve Optimization Problems” IJCIE, Vol 6, 2012

[15] A. Sharma and A. Mehta, “Review Paper of Various Selection Methods in Genetic Algorithm”, International journal of Advanced Research in Computer Science and Software Engineering, vol. 3, No.7, 2013, pp.1476-1479.

[16] Dipta Das Chowdhary, Rahul Jain , Prof (Dr.) Anindya Jyoti Pal Review Paper on Solving the Travelling Salesman Problem Using Genetic Algorithm, Ant Colony Optimization, Artificial Neural Network IJARCCE Vol. 5, Issue 6, June 2016

[17] K. Bharathi et.al,” A Framework for the Design and Analysis of an Evolutionary Algorithm for Multi Travelling Salesman Problem”, December 2016

[18] Bao Lin1 Xiaoyan Sun and Sana Salous,” Solving Travelling Salesman Problem with an Improved Hybrid Genetic Algorithm”, Journal of Computer and Communications, 2016.

[19] Mingji Xu, Sheng Li and Jian Guo “Optimization of Multiple Traveling Salesman Problem Based on Simulated Annealing Genetic Algorithm” 2016

[20] Pooja Vaishnav, Dr. Naveen Choudhary, Kalpana Jain Traveling Salesman Problem Using Genetic Algorithm: A Survey 2017 IJSRCSEIT.

[21] A. B. Hassanat and E. Alkafaween, "On enhancing genetic algorithms using new crossovers," International Journal of Computer Applications in Technology, 2017

[22] http://en.wikipedia.org/wiki/Genetic_algorithm