Journal of Theoretical and Applied Information Technology
31st August 2017. Vol.95. No.16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SWARMED GA ALGORITHM FOR SOLVING TRAVELLING SALESMAN PROBLEM

[1]VIKAS RAMAN, [2]NASIB SINGH GILL

[1]M.Tech Student, M.D University, Department of Computer Science & Applications, Rohtak, India
[2]Professor, M.D University, Department of Computer Science & Applications, Rohtak, India
E-mail: [1]vikas.raman92@gmail.com, [2]nasibsgill@gmail.com

ABSTRACT

TSP (Travelling Salesperson Problem) is one of the leading problems that are considered as an NP-hard and has been broadly studied problem in the area of combinatorial. TSP is termed as, a salesperson who wishes to visit *m* cities, and assumed to find out the shortest tour by visiting all the cities exactly once and lastly returning to the starting city. Genetic algorithm (GA) is a heuristic algorithm used for solving the TSP. Genetic Algorithm (GA) was emphasized to give better performance in solving TSP. But GA frequently undergoes into premature convergence because of the difficulty in avoiding the loss of genomic variety in the population. To overcome this drawback, GA that uses Intelligent Swarm Optimization algorithm's characteristic is presented. The presented algorithm is referred as Swarmed Genetic Algorithm (SGA) that contains an upgrading approach for the solution. The new approach was altered by inserting the three distinguished GA's mutation operators in the proposed algorithm which are the scrambled mutation, inversion mutation and displacement inversion mutation operators. This algorithm was compared with other GAs containing various mutation operators on instances from TSPLIB. Results obtained showed that the algorithm is much more efficient as compared to the GA and outperformed in most of the TSP instances.

Keywords: *TSP, GA, Mutation, SSO.*

1. INTRODUCTION

TSP is believed to be an NP-hard problem which is comprehensively studied in the area of combinatorial optimization [1]. The first formulation of this problem was given in 1930 by K. Menger [2]. In 1934 H. Whitney presented the name "Travelling Salesperson Problem" at Princeton University [3]. Earlier in 1954, it was perceived as a mathematical problem and cutting plane method was used to solve by Dantzig [4]. In 1972 TSP was considered as an NP-hard problem because of its complexity in computation in order to find the best suitable route [5]. As a result of this complexity, a huge number of exact and heuristic algorithms have been presented so far to offer the best solutions [6]. All these reduced a bit of complexity but not at a much greater extent. The travelling salesperson problem is summarized as:

"TSP is a problem to discover the best shortest appropriate path by the salesperson to traverse *n* cities so that we can reach each and every city exactly once & finally comes to the initial position with least resources utilization as well as time."

Since the TSP problem came into existence many researchers started work on it and presented numerous algorithms to solve it. These algorithms are mainly classified as: approximate and exact algorithms [7]. The exact algorithms are said to be those algorithms that solve an optimization problem to give optimal result. The running time for the exact algorithm approaches lies in polynomial factor i.e. O(n!), it is the factorial of number of cities taken in problem, so it is much complex in case of 20 to 25 cities only. The exact algorithms are the dynamic programming, greedy algorithms and branch-and-bound. These algorithms typically require much time for computation, specifically in case of bigger problems. On the other side, the approximate algorithms are classified as heuristic and meta-heuristic algorithms. The approximate algorithms do not solve an optimization problem to give optimal result but they provide the best possible solution which is closer to the optimal

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

solution. It is a widely adopted technique in order to deal with NP – complete problems for the sake of obtaining optimized result. The most fruitful example of the approximate algorithms is the Lin-Kernighan algorithm (LK) and termed as a λ- opt algorithm. Other examples of approximate algorithms are the Ant Colony Optimization (ACO), Tabu Search (TS), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Simulated Annealing (SA), which have been used in solving TSP.

The popular among these are ant colonial optimization (ACO), particle swarm optimization (PSO) & genetic algorithm (GA). All these are influenced by naturally occurring activities that provide the idea of using it in optimization problems. The ACO is a meta-heuristic approach that is based on ants' capabilities in searching the food articles as well as their nests. They can find the shortest or in other words optimized path without having any visual aid. This can be done with the help of the hormone named 'pheromone' released by them which is used for communicating among all the ants. They follow the path chosen by other ants due to this pheromone that provide the information about the path chosen by other ants and they chooses the path having more denser pheromone secretions in order to get the best or we can say the shortest path.

And the next popular is particle swarm optimization (PSO) which is based on swarm intelligence to find and travel through the shortest path. The inspiration in designing this algorithm is taken from the methodologies used by bird flocks, humans and other communities like fishes group. It depends on their way of communicating and optimizing the path travelled by them. They exchange the information among the individuals in their group as they find the shortest path. And they does not use the strategy like mutation, crossover etc as used in genetic algorithm. The problem with these algorithms is that they usually suffers from the local optima problems and traps into sub-optimal solution.

Among these algorithms, GA has proved to be a high efficient, flexible, robust, and versatile, that has been broadly studied to solve the TSP. Furthermore, the effectiveness of the genetic algorithm relies on the used operators such as selection, encoding, crossover and mutation strategies. Moreover, GA usually suffers from local optima problem which is also termed as premature convergence [8]. The major reason behind the premature convergence is supposed to be Genetic diversity [9]. And this premature convergence arises because of the crossover strategy used in the genetic algorithm approach. In the proposed algorithm we avoid the use of the crossover operator that leads to the convergence. And instead of using crossover we implemented the swarm intelligence.

In order to make sure that the algorithm prevents the condition of local optima a sufficient genetic diversity is essential. To avoid the disadvantage, the proposed algorithm implements swarm intelligence in the GA for the sake of prevention of genetic diversity's loss which provides the improved solution.

The organization of the paper is as follows: Section 2 introduces the related work to Travelling Salesperson Problem. Section 3 defines and formulates the TSP problem. Section 4 briefly describes the research methodology i.e. genetic algorithm and simplex swarmed optimization. Section 5 illustrates the proposed method. The experimental results are presented in section 6. Lastly, Section 7 concludes our paper and future work.

2. RELATED WORK

In the past, since the TSP put its footprint into the arena of optimization problem it became a topic of more concern and many researchers from various streams starts working on it in order to give an effective approach to deal with the problem. These approaches are GA (Genetic Algorithm) [10], SA (Simulated Annealing) [11], NN (Neural Networks) [12], ACO (Ant Colony Optimization) [13], PSO (Particle Swarm Optimization) [14], EA (Evolutionary Algorithm)[15] etc. that proves to be better approaches and modified with time for making them more effective. Saloni et al. in the paper [16] put forward an improved GA to deal with a problem named TSP. In the development they worked on better genetic operators that are tournament selection for selecting parent individuals from population, 2-point crossover and interchange mutation for generating better offspring.

In paper [17] Yan et al. portrayed a new PSO algorithm that in capacitates the disadvantage of GA that is premature convergence providing suboptimal solution. The

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

new approach is a mix-up of the three approaches i.e. PSO-GA-ACO and produces a twin-stage hybrid swarm intelligent optimization algorithm that offers a much efficient solution to TSP problem. In paper [18] a hybrid Genetic Algorithm using Particle Swarm Optimization was presented in order to solve ED (Economic Dispatch) problems. Here the main emphasis is on crossover strategy to generate better offspring. The crossover operator used here is arithmetic crossover; this operator is applied randomly to produce a coefficient for the new child generated. Lopes and Machado in their paper [19] combines three strategies GA, PSO and FLS (Fast Local Search) and developed a hybrid technique for TSP. In their work, they use permutation in order to position the particles over cities. Each particle has a fitness value that ranges over the rates between the D min and tour's cost according to particle's position.

In most recent times, the SSO approach was presented based on swarm optimization for solving problems of optimization [20]. Also in GA we work with many different operators of selection, mutation and crossover with time which improves the result but not at large extent. All these operators help in developing a better population but sometimes lead to convergence that is the main concern of our work, where we deal with this situation in order to obtain the best possible result.

3. TRAVELLING SALESPERSON PROBLEM

3.1 Problem Definition

Travelling salesperson problem is one of the well recognized NP – hard problem that is said to be combinatorial optimized problem. And finding the optimal solution to this kind of problem is a challenge and is highly complex in nature. In this problem we have to discover the best shortest appropriate path by the salesperson to traverse *n* cities so that we can reach each and every city exactly once & finally comes to the initial position with least resources utilization as well as time. A figure given below shows the TSP (Travelling Salesperson Problem) problems solution.

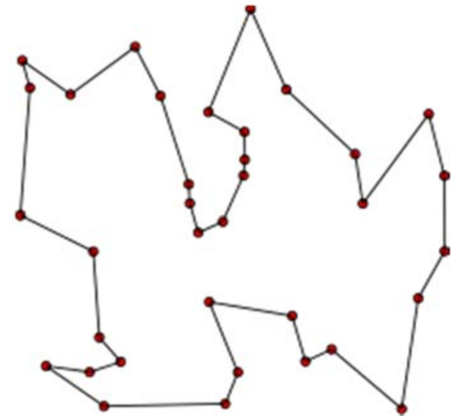

Figure 1: Representation of TSP solution

3.2 Problem Formulation

The formulation of TSP (Travelling Salesperson Problem) in the form of integer linear programming is as follows: Assign number to the cities as 1, 2, 3, …….., n.

$$P_{ij} = \begin{cases} 1 \; if\ path\ exists\ between\ city\ i\ to\ city\ j \\ 0 \quad otherwise \end{cases}$$

For i = 1, ..., n, let v_i be a model variable and take D_{ij} to be the distance from city i to city j.

Now TSP can be formulated as:

Minimize

$$\sum_{i=1}^{n} \sum_{j=1}^{n} Dij\ Pij$$

$$0 \leq P_{ij} \leq 1$$

$$v_i \in Z$$

$$\sum_{i \neq j\ i=1}^{n} Pij = 1$$

$$\sum_{j \neq i\ j=1}^{n} Pij = 1$$

Where i, j = 1, ... , n;

$v_i - v_j + nP_{ij} \leq n - 1 \qquad 2 \leq i \neq j \leq n.$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The initial equality set projects that each and every city should be visited from exactly one other city, whereas the second equality set refers that from each and every city departure to exactly one other city is possible.

The final constraints impose that there exists only one single tour that cover all cities i.e. not two or more tours exists that cover all cities.

4. RESEARCH BACKGROUND

4.1 Genetic Algorithm

GA is a meta-heuristic algorithm based on population set in order to find best offspring from the present population. It is a subset of evolutionary algorithm (EA) and came into existence in 1975.John Holland was the person who introduced it[21]. GA is a technique that uses random search inspired by the natural criteria of selection and survival of the best chromosomes. The main purpose of GA is to acquire better solutions by eliminating the worse solutions at the time of generation of population from current generation. The operators used in genetic algorithms are the most important in order to achieve the genetic diversity in the population. Usually, Genetic Algorithm has four most important operators.

The very first operator is an encoding operator that converts the problem solution into genetic information. As TSP is said to be a permutation problem, it is taken as permutation encoding operator. The next operator is selection operator used to select the best suitable parents for generation of off springs from the population on the basis of their fitness. The individuals having better fitness have larger chance to get selected. Here, we use tournament selection. The third operator is crossover operator which is used for mating process of two selected parents and for generating a child. The mutation operator is the final operator in GA, which is used for the prevention of local optima in the algorithm and escalates the genetic diversity of the population. In the paper, 3 mutation operators are cast in the presented algorithm that are scramble mutation operator (SM), Interchanging mutation operator (IM) and Displacement Inversion mutation operator (DIM). The introduction of these mutation operators is given in next paragraph. Genetic algorithm begins with the generation of a random population. From the population, every chromosome is estimated to decide the best chromosomes. These best chromosomes are now nominated for a mating process to create new chromosomes to replace the worse chromosomes. Accordingly, the best chromosomes generates the new generation. The same procedure continues for several generations until the condition is satisfied. The pseudo-code of the GA is given in Figure 1 below:

Step 1: Choose an initial random population of individuals, p.

Step 2: Evaluate the fitness of the individuals, f.

Step 3: repeat

Step 4: Select the best individual to be used by the genetic operators.

Step 5: Generate new individuals using crossover and mutation operators.

Step 6: Evaluate the fitness of the new individuals.

Step 7: Replace the worst individuals of the population by the new individuals.

Step 8: until some stop criteria is met.

Figure 2: Pseudo-code of Genetic Algorithm

4.1. Mutation Operators

Mutation operator has much significance in Genetic Algorithm. It plays an important role in diversifying the genes. In mutation a gene is instinctively changed and passes on to next generation. This results in production of new better generations and leads to evolution. There are various types of mutation operators present in the literature from which we selected the three operators which are discussed in this section. Here are the details of how these three mutation operators will be implemented.

Let us take a set of string (7 4 1 9 3 5 2 6 8 0 7) which denotes a tour with 10 cities and the salesperson starts from city 7 to city 4, 1, 9, 3, 5, 2, 6, 8, 0 and return to city 7. The mutation operator begins by selecting 2 or more cities randomly from the provided string.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.1.1. Scramble Mutation (SM)

In scramble mutation a selection of two random points takes place and shuffles the substring between these two points. Suppose here the two points are city 3 and city 6. The substring between these points is (3, 5, 2, 6) which gets shuffled and become (2, 5, 6, 3).

Before Mutation	7 4 1 9 3 5 2 6 8 0 7
After Mutation	7 4 1 9 2 5 6 3 8 0 7

4.1.2. Interchanging Mutation (IM)

In interchange mutation two points are randomly selected and the bits related to these points are exchanged. For example here the two points are city 9 and city 8 which are interchanged with each other and city 9 replace city 8 and city 8 takes the place of city 9.

Before Mutation	7 4 1 9 3 5 2 6 8 0 7
After Mutation	7 4 1 8 3 5 2 6 9 0 7

4.1.3. Displacement Inversion Mutation (DIM)

In displacement Inversion two points are selected at random and the substring between the selected points is reversed and placed at some different position along the original string. For example substring in-between city 9 and city 8 is inverted and displaced accordingly.

Before Mutation	7 4 1 9 3 5 2 6 8 0 7
After Mutation	7 4 6 2 5 3 1 9 8 0 7

4.2 Crossover Operator

Crossover operator is also of very much significance in Genetic Algorithm. It is mainly employed to obtain the better offspring quality from existing population by crossing the parents selected. In crossover many different strategies are used for crossing and generating the new individuals.

Parent 1	1 1 1 0 1 0 0 1 0
Parent 2	1 0 0 0 1 0 1 1 0

Offspring 1	1 1 1 0 1 0 1 1 0
Offspring 2	1 0 0 0 1 0 0 1 0

But along with its application of obtaining genetic diversity it also leads to a serious problem which is premature convergence that means the new derived generation is identical to the existing and no further improvement takes place in the population. The premature convergence aroused makes the solution suboptimal which is a serious drawback.

5. SIMPLEX SWARMED OPTIMIZATION

Simplex Swarm Optimization (SSO) is another type of PSO. Yeh is the person behind its development [22]. The main purpose behind its development was to improve the PSO in terms of convergence rate, accuracy and flexibility. The idea of its development was based on conventional and Discrete PSO that has the incompetence to deal with discrete problems. The main issue with PSO is the premature convergence mainly in large scale multimodal problems. The convergence rate tends to decrease with rise in iterations number. Hence, to overcome the drawbacks of PSO, Simplex Swarm Optimization was given. The main point of difference between the two algorithms is particle velocity vi. SSO has nothing to do with the particle velocity so it eliminates the particle velocity. On the other hand,in PSO this particle velocity is the main factor to decide the new positions for the problems. This particle velocity also increases the complexity in PSO i.e. time and computational. The results after computation indicated that SSO performs better in order to find the ideal solutions with accuracy. SSO's equation is as follow:

$$x_{id}^{t} = \begin{cases} x_{id}^{t-1}, & if\ rand() \in [0, C_w), \\ p_{id}^{t-1}, & if\ rand() \in [C_w, C_p), \\ g_{id}^{t-1}, & if\ rand() \in [C_p, C_g), \\ x, & if\ rand() \in [C_g, 1), \end{cases}$$

In the equation, $x_{id}{}^{t}$ show the position of i^th particle in d-dimension, whereas the C_w, C_p, and C_g are preset constants with positive values. So, $p_{id}{}^{t-1}$ and $g_{id}{}^{t-1}$ specify the present best place (pbest) and global best place (gbest) resp., and x shows the modified value of the randomly generated particle by using random function (rand ()), where the value of rand () lies between

Journal of Theoretical and Applied Information Technology
31st August 2017. Vol.95. No.16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

0 to 1. This random value generation to update pBest and gBest provides special characteristics to SSO and this special characteristic is termed as Improved Solution Approach (ISA).

Step 1. Initialize the problem and preset the values of m (Swarm size) C_w, C_p, C_g (Predetermined Constants) MaxGen (Maximum generation) MaxFit (Maximum fitness value).

Step 2. Now Generate and initialize the pbest and gbest with random position x.

Step 3. Repeat

Step 4. Evaluate the fitness value for each and Update the *pbest* and *gbest.*

Step 5. Generate random number *r*.

Step 6. Compare this '*r*' with predetermined constants C_w, C_p, C_g.

If $0 \le r < C_w$ then keep the original value

Else if $C_w \le r < C_p$ then replace value by *pbest*

Else if $C_p \le r < C_g$ then replace value by *gbest*

Else randomly generate the value to replace the original value.

Step 7. Until some stop criteria is met.

Figure 3: Pseudo-code of SSO

6. PROPOSED ALGORITHM

The algorithm proposed here is termed as Swarmed Genetic Algorithm (SGA). In this algorithm we improve genetic algorithm by embedding swarm intelligence in it. Conventional Genetic algorithm works with four main operators that are encoding, selection, crossover and mutation. In the algorithm shown in figure 3, we used three of them and eliminated the crossover operator as it produces some new strings same as the old ones which results in more time for computation. Here we are using three mutation operators scramble, interchanging and displacement inversion mutation operator and embed the Improved Solution Approach (ISA) after every mutation operator which improves the current best solution, *pbest*. And a mutation operator with ISA is said to be a stage. Accordingly we have three stages in the algorithm. After each stage the solution is compared with the existing, *gbest* and updated if found better otherwise the algorithm move onto next stage. If the existing solution, *gbest* is the best solution then the final result remains same and no updating takes place.

The working principle of proposed algorithm is that initially we assign a global fitness value to the initial population that will be compared with the fitness value of resulted population obtained after each stage in order to check whether the new value is better than the existing or not.

Firstly we encoded the population set (using permutations and combinations) for further processing. After this we select a set of population for performing the mutation operations mentioned in proposed algorithm. In the proposed approach we used tournament selection strategy for selection of individuals from a wide variety of population.

After selection criteria we implemented three different mutation operators over the selected set of population one parallel to each other in order to get more improved quality. Then evaluate their fitness value and compare it with the *gbest* if the newer value is better than existing then replace the older value with newer value otherwise perform the mutation until we get a better result.

In this way our proposed strategy works to find the best possible solutions. And this can be done by mixing genetic algorithm approach and swarmed intelligence approach together.

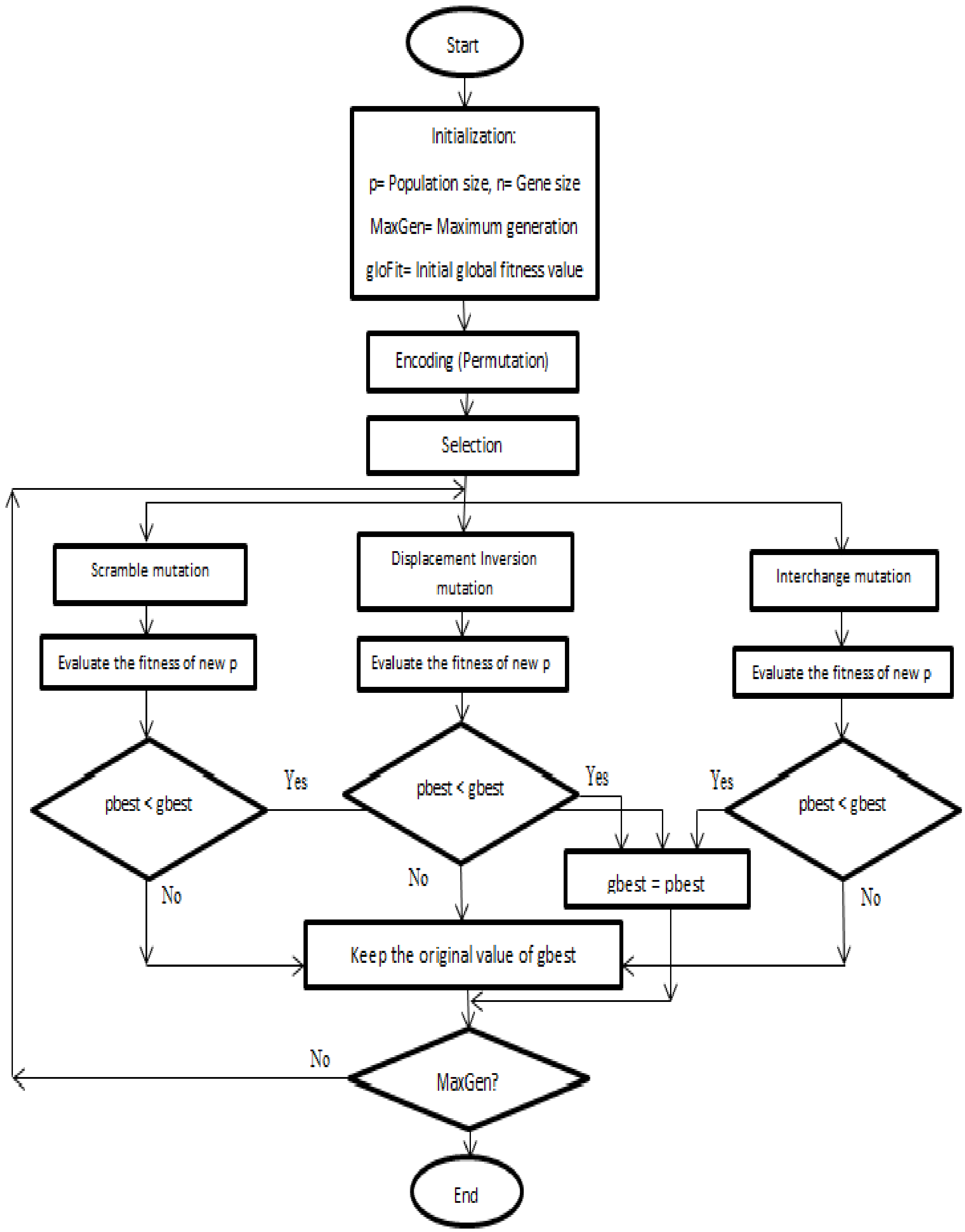

Figure 4: Swarmed Genetic Algorithm

7. RESULT AND ANALYSIS

Here the results of the proposed algorithm are presented after the computation. For this we worked on a system with core i3 2.93GHz processor and uses MATLAB for coding. For the analysis of the algorithm we considered seven TSP instances from TSPLIB and examined each for about eight to ten trails. While implementation, the population size was taken to be 100 and iterations were 20,000 in number. Chart 1 represents the results of SGA (Swarmed Genetic Algorithm) on the instances considered.

Here 3 colors represents the three cases for each problem i.e. blue for best case, red for average case and green for worst case. Also we compared the best and average results of proposed algorithm with the optimal solution present so far for the instances and other GA's. This comparison is shown in chart 2 and chart 3. Analysis shows that SGA performs better in average cases than other GA. The proposed SGA improves the computational time & also provides better result in the instances taken when works comparatively with other algorithms.

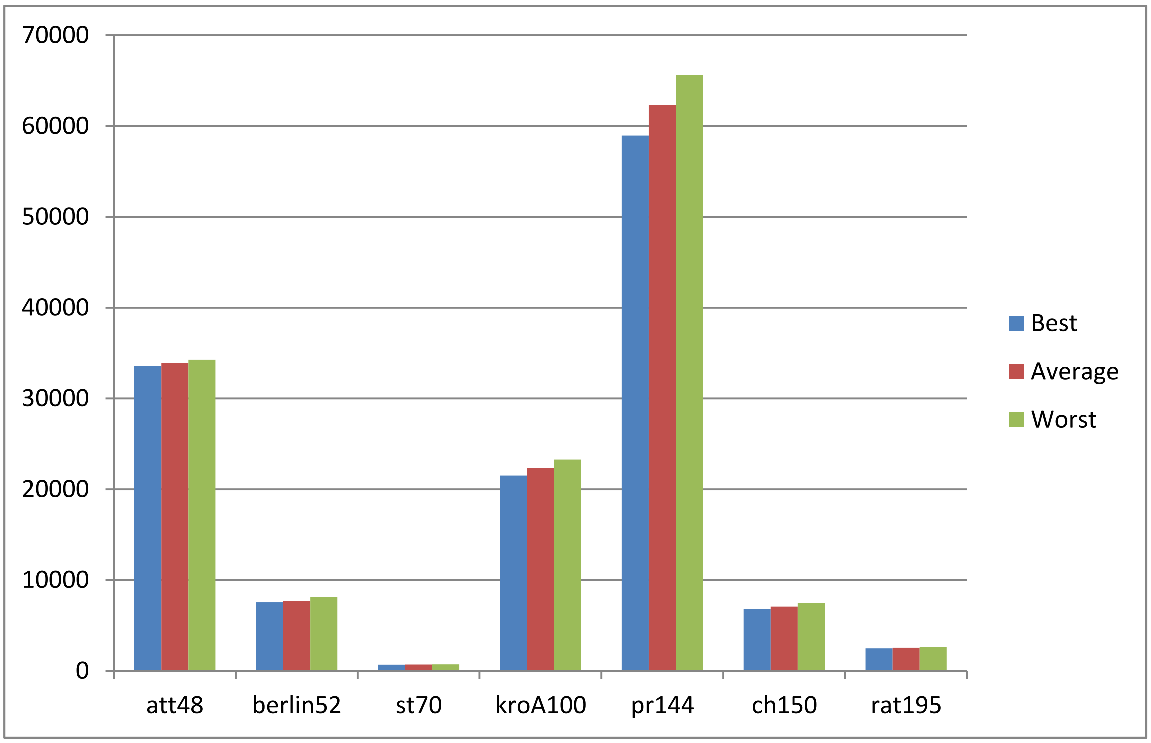

Chart 1 : Results of the SGA for TSP instance

In chart 1, we provide the results of swarmed genetic algorithm (SGA) by taking in view different TSP instances like att48, berlin52, rat145, etc. In this chart, we place all the three cases that are best, average and worst case representing the quality of the proposed algorithm. And in chart 2, a comparative study on different proposed best solutions to TSP is done and presented. Here, the swarmed genetic algorithm is represented using brown color, the genetic algorithm is represented using green color and the optimal known solution is represented using blue color. And finally in chart 3, the comparison is done on the basis of average result among the swarmed genetic algorithm, the genetic algorithm and the optimal known algorithm. Here again, the swarmed genetic algorithm is represented using brown color, the genetic algorithm is represented using green color and the optimal known solution is represented using blue color.

Journal of Theoretical and Applied Information Technology
31st August 2017. Vol.95. No.16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

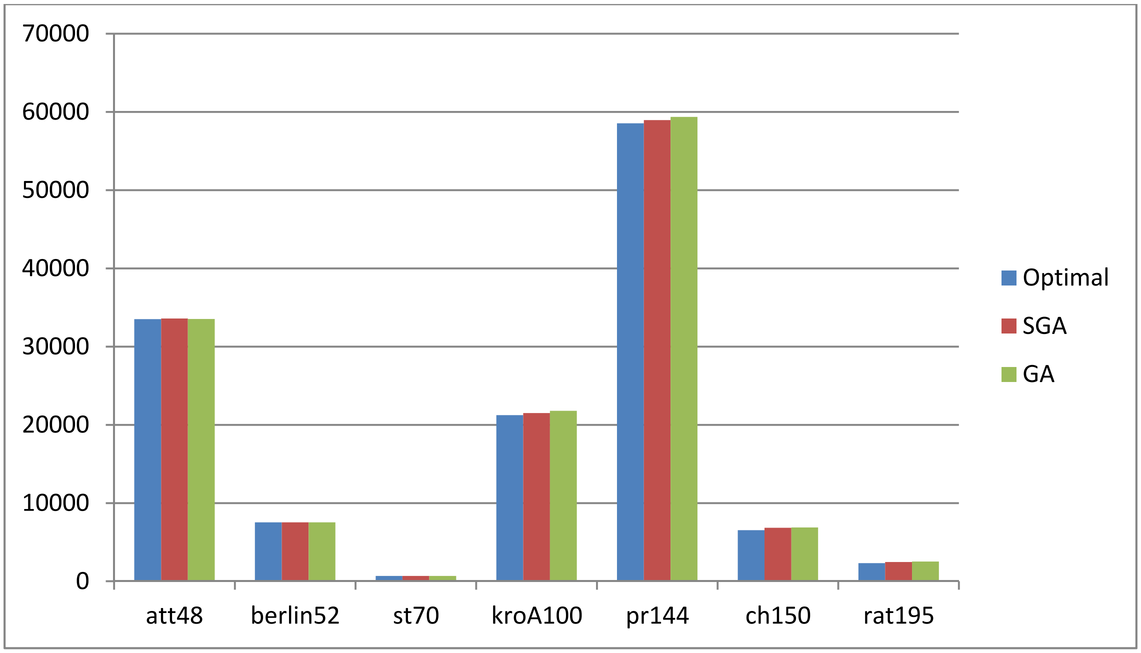

Chart 2: Comparison of best results of SGA with GA and known Optimal Solution

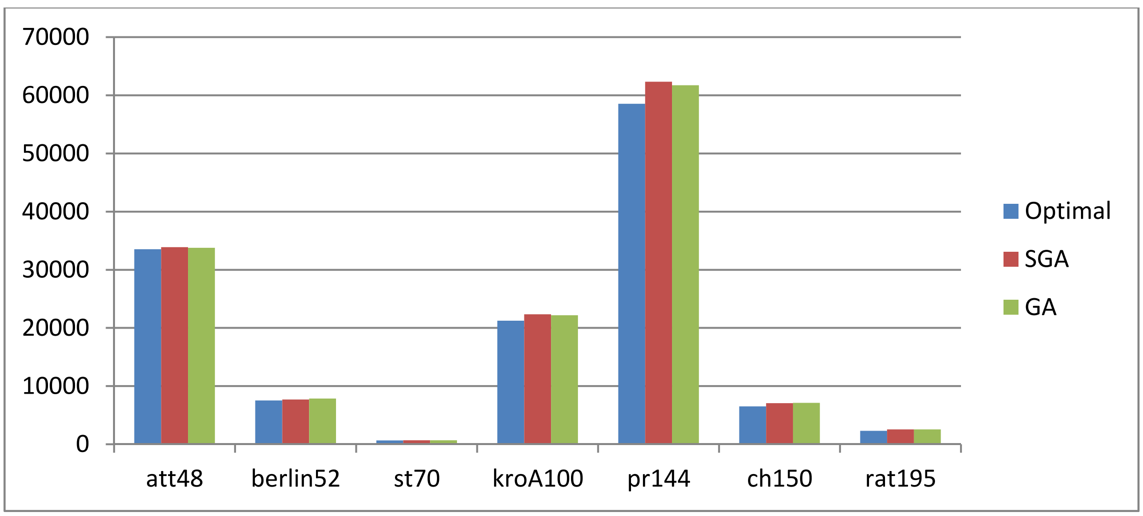

Chart 3: Comparison of average results of SGA with GA and known Optimal Solution

8. CONCLUSION AND FUTURE WORK

To solve travelling salesperson problem our proposed algorithm termed as swarmed genetic algorithm suits better. It shows better efficiency and performance than other GA's presented so far. The differences in the results after their comparison can be shown above in experimental results. This betterment can be possible by embedding simplex swarm optimization into the genetic algorithm that improves the capabilities of GA in obtaining far better solutions. By implementing this technique in GA we obtain vast population diversity that overcomes the major drawback called premature convergence from GA. This makes the algorithm more efficient and a better approach that enhances the results and reduces the complexity. However there are further improvements possible in the algorithm and it can be further improved by doing some modifications in it for reduction of its computational complexity and work for its easy implementation. Even one can develop a new approach or algorithm to deal with travelling salesperson problem and other combinatorial problems. Also by mixing present approaches and make an ideal hybrid algorithm for getting optimal results.

REFERENCES

[1] Yan, X., Zhang, C., Luo, W., Li, W., Chen, W., & Liu, H., "Solve traveling salesman problem using particleswarm optimization algorithm", International Journal of Computer Science, vol. 9, no. 2012, pp.264-271, 2012.
[2] Maredia, A., "History, Analysis, and Implementation of Traveling Salesman Problem (TSP) and Related Problems", Doctoral dissertation, University of Houston, 2010.
[3] Alexander, S., "On the history of combinatorial optimization (till 1960)", Handbooks in Operations Research and Management Science: Discrete Optimization, 2005.
[4] Dantzig, G., Fulkerson, R., & Johnson, S., "Solution of a large-scale traveling-salesman problem", Journal of the operations research society of America, vol. 2, no. 4, pp. 393-410, 1954.
[5] Karp, R. M, "Reducibility among combinatorial problems", Springer US, pp. 85-103, 1972.
[6] Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J, "The Traveling Salesman Problem: A Computational Study: A Computational Study", Princeton university press, 2011.
[7]Laporte, G., "The traveling salesman problem: An overview of exact and approximate algorithms", European Journal of Operational Research, vol. 59, no. 2, pp 231-247, 1992.
[8]Gupta, D., & Ghafir, S., "An overview of methods maintaining diversity in genetic algorithms", International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 5, pp. 56-60, 2012.
[9] Malik, S., & Wadhwa, S., "Premature Convergence In Genetic Algorithm Using Ellite Selection Scheme: Review Paper", 2014.
[10]Hua Z, Huang F, "A variable-grouping based genetic algorithm for large-scale integer programming", Inf Sci 176(19), 2006, pp.2869–2885.
[11] Lo CC, Hus CC, "Annealing framework with learning memory", IEEE Trans Syst Man Cybern Part A 28(5), 1998, pp.1–13.
[12] Masutti TA, de Castro LN, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem", Inf Sci 179(10), 2009, pp.1454–1468.
[13] Ellabib I, Calamai P, Basir O, "Exchange strategies for multiple ant colony system", Inf Sci 177(5), 2007, pp.1248– 1264.
[14] Onwubolu GC, Clerc M, "Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization", Int J Prod Res 42(3), 2004, pp.473– 491.
[15] Shen G, Zhang YQ, "A new evolutionary algorithm using shadow price guided operators", Appl Soft Comput 11(2), 2011, pp.1983–1992.
[16]Gupta S, Panwar P. "Solving Traveling salesman problem using genetic algorithm" International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE). 2013; 3(6):376–80.
[17]Xuesong Yan, Can Zhang, WenjingLuo, Wei Li, Wei Chen and Hanmin Liu "Solve Traveling Salesman Problem Using Particle Swarm
[18]Guvenc U, Duman S, Saracoglu B, Ozturk A, "A hybrid GA–PSO approach based on similarity for various types of economic dispatch problems", Electron ElectrEng Kaunas: Technologija 2(108), 2010, pp.109–114
[19]Machado, T.R. & Lopes, H.S, "A hybrid particle swarm optimization model for the traveling salesman problem", In: Natural Computing Algorithms, Ribeiro, H.; Albrecht, R.F. &Dobnikar, A. (Eds.), 2005, pp.255-258
[20]Chung, Y. Y. & Wahid, N., "A hybrid network intrusion detection system using simplified swarm optimization (SSO)", Applied Soft Computing, pp. 3014-3022, 2012.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[21]Malhotra, R., Singh, N. & Singh, Y., "Genetic algorithms: Concepts, design for optimization of process controllers", Computer and Information Science, vol. 4, no. 2, pp.39-54, 2011.

[22]Bae, C., Yeh, W. C., Wahid, N., Chung, Y. Y. & Liu, Y., "A new simplified swarm optimization (SSO) using exchange local search scheme", International Journal of Innovative Computing, Information and Control, vol. 8, no. 6, pp. 4391-4406, 2012.