

The Effect of Data Perturbation on MOTSP solved using Multiobjective Genetic Algorithm

J Kanimozhi¹, R Subramanian²

M.Tech, Dept. of CSE, Pondicherry University, Puducherry, India¹

Head of the Dept., Dept. of CSE, Pondicherry University, Puducherry, India²

ABSTRACT:Traveling Salesman Problem (TSP) is a challenging problem in combinatorial optimization. The important of this problem is due to the fact that it is used in many fields such as transportation, logistics, semiconductor industry, problem of routing etc. In this paper Multiobjective Travelling Salesman Problem (MOTSP), the extended instance of TSP is solved using Multiobjective Genetic Algorithm (MOGA) combined with data perturbation (DP) and the algorithm named as Perturbed MOGA (P-MOGA). DP is a technique used to avoid local optima and to increase the diversity property of the problem. Efficiency of the algorithm is calculated in terms of fitness, IGD and from the result analysis, it's proved that Perturbed MOGA outperforming MOGA to solve MOTSP.

KEYWORDS:Multiobjective TSP, Genetic algorithm, Data perturbation, Cost function.

I. INTRODUCTION

The Travelling Salesman Problem (TSP) is a NP-hard problem [1] - [4]. Travelling salesman problem states that given a number of cities N and the distance or time to travel between the cities, the traveler has to travel through all the given cities exactly once and return to the same city from where he started and also the cost of the path is minimized [1] - [7]. This path is called as the tour and the path length or travel time is the cost of the path. The TSP mathematical model follows [3] [8] [9]:

$$\min Z = \sum_{i=1}^N \sum_{j=1}^N X_{ij} C_{ij}$$

Subject to

$$\sum_{i=1}^N X_{ij} = 1 \quad j=1, 2, 3, \dots, N$$

$$\sum_{j=1}^N X_{ij} = 1 \quad i=1, 2, 3, \dots, N$$

Let C_{ij} be the cost for traveling from i-th city to j-th city. Where $X_{ij} = 1$ if the salesman travels from city-i to city-j, otherwise $X_{ij} = 0$. Z is the minimum distance of the optimal path.

Multi Objective Travelling Salesman Problem (MOTSP) is a multiobjective problem considers more objectives to find the optimal path. Given N cities and p cost C_{ij}^k ($k = 1, \dots, p$) between each pair of distinct cities to travel, the MOTSP consists in finding a Hamiltonian cycle of the N cities that optimizes the following minimization Problems [9] - [12]:

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

$$\min Z_2 = \sum_{i=1}^N \sum_{j=1}^N X_{ij} T_{ij}$$

$$\min Z = \sum_{i=1}^N \sum_{j=1}^N X_{ij} C_{ij}$$

Subject to

$$\sum_{i=1}^N X_{ij} = 1 \quad j=1, 2, 3, \dots, N$$

$$\sum_{j=1}^N X_{ij} = 1 \quad i=1, 2, 3, \dots, N$$

The above equation gives the mathematical model of MOTSP by considering two objectives for optimization. The first objective function considers the minimization of the distance traveled by the salesman, while the second objective function considers the travelling time of the salesman. Let C_{ij} be the cost for traveling from i -th city to j -th city. T_{ij} is the travelling time from i -th city to j -th city and $X_{ij} = 1$ if the salesman travels from city- i to city- j , otherwise $X_{ij} = 0$.

Problem with multiple objectives cannot be solved as like single objective problems. It can be solved efficiently with Multiobjective Evolutionary Algorithms. The algorithms used to solve MOTSP are listed and reviewed in section 2.3. Since MOTSP is a multiobjective problem it needs to be solved using MOEA's. The objectives in MOTSP are conflicting with each other, for example a tour with the shortest length in one objective might also be the most expensive in another one. Therefore, there is not a single tour that can minimize all the objectives simultaneously [12] - [15]. And so, w_1 and w_2 denotes the weights that are used to balance between the total traveling distance and the traveling time such that $w_1 + w_2 = 1.0$. In this study, w_1 and w_2 are both set to 0.5 [16].

This paper presents A Perturbed MOGA Algorithm to solve MOTSP. The framework of Perturbed MOGA combines Multiobjective optimization and Data Perturbation. The paper is organized as follows: Section 2 gives the literature review of Genetic algorithm to solve MOTSP and data perturbation techniques to avoid local optima. Section 3 describes perturbed MOGA algorithm and its components. Section 4 gives the Experimental result and section 5 concludes the work.

II. RELATED WORK

The following section 2.1 gives the literature review of the GA to solve MOTSP and 2.2 gives the data perturbation concept and two methods of doing the data perturbation.

2.1 Literature review of MOGA to solve MOTSP

From the literature survey of MOTSP, Meta-heuristic algorithms are better to solve multiobjective optimization problems. Multi Objective Genetic Algorithm (MOGA) is a meta-heuristic algorithm which solves the MOTSP efficiently. Multi Objective Genetic Algorithm (MOGA) was given by Fonseca et al [12]. It is an extension of single objective optimization algorithm. In MOTSP, the rank to an individual is assigned based on the number of solutions in the population by which it is dominated. GA has two main parts, an evolution function and a fitness function. In the case of the MOTSP, the parameters produced by the evolution function might be the order of the nodes through which the path will go. The fitness function in that same case would return the total length of the path found. The GA would then compare fitness values for each input string and assign priority to the ones that returns lower path lengths and other objectives. Based on the objective values the best solutions will be produced. The framework of the MOGA was explained in [11] [12]. The steps in MOGA are, Initial Population, Fitness Evaluation, GA operators (Crossover and Mutation), Update population with new solutions, termination [17].

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

Initially GA is used to solve TSP [1] – [8], which gives better result than solving it with ACO or with exact algorithms. Later the need of multiobjective optimization evolved GA to MOGA. In [9] MOTSP is solved using MOGA where the rank is calculated using Pareto function and continued with tournament selection for crossover. In [18] multiobjective GA and mono objective GA combined together to find the fitness of the population to solve Multiple TSP. Fuzzy system is combined with GA to solve MOTSP and which is giving better result than the previous algorithms [19], and in [20] [21] [22] [23] vehicle routing problem is solved using MOGA. Simulated annealing and GA are combined together to solve multiple TSP in [24]. Still there is problem with MOGA in solving MOTSP. MOGA give local optimal solutions in a minimal time [25]. So to avoid the local optima and to get global optima data perturbation is used. The need of DP, method of doing DP and its literature survey are given in the next section 2.2.

2.2 Data Perturbation (DP)

In Evolutionary Algorithms, the initial population size, diversity and convergence property of initial population influences more on the optimal solution [9]. “Data Perturbation” (DP) technique, originally proposed by Codenotti et al. for the single-objective TSP and has been introduced in MO optimization by Lust and Teghem [26] [27]. A perturbation move is a technique used to escape from local optimum which is an issue found in MOGA. Instead of modifying the starting solution DP suggests to modify input data [9]. The perturbation is a double bridge move [14] that cuts the current tour at four appropriately chosen edges into four sub-tours and reconnects these in a different order to yield a new starting tour for the local search.

There are two methods to do the DP which are given in [26] - [30]. The first DP method in [26] [28] [29] is started with the input parameters number K of iterations, three parameters that determine the perturbation scheme (the fixed scale SF , the varying scale SV and the local perturbation LP) and the cost matrices $C_{i,j}^k$ of the MOTSP. During an iteration k , first compute a weight set λ by following a linear scheme; K uniformly distributed weight sets are thus generated. We then create a new cost matrix C^λ . Then, we slightly perturb each cost $C^\lambda(i, j)$ of the matrix C^λ to find new potentially efficient solutions. Increase in the number K of iterations gives an important improvement of the indicators and allows reaching excellent results, since the number of potentially efficient solutions $|PE|$ is increasing while the distance $D1$ is decreasing. The number of iterations for the number of perturbation steps is equal to the number of cities N minus 50.

The second DP method in [27] is done with a single parameter d , whereas the above one needs three parameters. Higher the d value, the larger the perturbation is. It adds a random noise to the cost function and so the search direction is given in all the way. The value of d is set from 3 to 20% variations. The better results are given with 5%. So for optimal result use $d=5\%$.

III. PERTURBED MULTI-OBJECTIVE GENETIC ALGORITHM

From the survey of MOGA and Data perturbation techniques, it is efficient to use data perturbation with MOGA to solve MOTSP. In existing from [26] – [30] Data perturbation technique is used to solve bi-objective TSP and MOTSP. In [27], the second method of DP is used in all other the first technique is used. From the literature survey is proved that DP is giving better solution than the algorithm without DP, by increasing the diversity property of the problem. The second DP method with single parameter d is used in our work since the execution time is less compared with the first method. It is appropriate to use DP, when the number of cities is more. This section gives the framework of Perturbed MOGA, characteristics and flow chart. Characteristics of Perturbed MOGA follow:

1. Initialization of Perturbed MOGA starts with the data perturbation of cost matrix. The initial cost matrix is computed using data set and using d parameter cost matrix is modified.
2. Apply MOGA operator to generate new solutions.
3. Update the optimal solutions in population and continue the same process till termination condition met.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

The following are the notations used in the description of Perturbed MOGA.

- A unique weight $\lambda^k = (\lambda_1^k, \dots, \lambda_j^k)$
- $C^k(e) = \sum_{j=1}^p \lambda_j^k C_j(e)$ is the initial cost function
- $P = \{x^1, \dots, x^n\}$
- N: Population Size
- T: Maximum number of generations

The Perturbed MOGA algorithm is given below in the Algorithm 1.

Algorithm 1: Perturbed MOGA

Input: a MOTSP problem, a stopping criterion, data perturbation parameter d, Initial Population P

Output: set A of potentially efficient solution

Begin

```

Let  $C^k$  be the cost matrix related to MOTSP
Data perturbation ( $C^k, d$ )
  Foreach  $e \in E$  do
     $v \leftarrow U(1-d, 1+d)$ 
     $C^k(e) \leftarrow v \times \sum_{j=1}^p \lambda_j^k C_j(e)$ 
  end

Randomly initialize the population  $P = \{x^1, \dots, x^n\}$ 
For  $i=1:T$ 
  GA Operator ( $C^k(e), P$ )
  Generate a new solution  $y$  with crossover and mutation
  Update the population P
end
Return the Best Individual

```

End

The framework starts with the Cost matrix (C^k) and continued with the data perturbation. The DP adds noise into the multi objective cost function. The data perturbation parameter $d \geq 0$, which controls the maximum variation of the noise added to the cost. The value of v is a real number calculated using d value varies with uniform distribution of $(1-d)$ to $(1+d)$. So the perturbed cost matrix is $C^k(e) \leftarrow v \times \sum_{j=1}^p \lambda_j^k C_j(e)$. With $C^k(e)$ the GA operator continues to generate new solutions. The best individuals are added in Population (P) and the process end when the termination condition met. With experimental analysis d value must be equal to 5% to produce an optimal solution which is show in the result analysis section.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

IV. EXPERIMENTAL RESULTS

The significance of the proposed Perturbed MOGA is demonstrated using the Multiobjective travelling salesman problem (MOTSP). The MOTSP is one of the most important optimization problems for the researcher has been chosen as the testbed. Naturally, the MOTSP belong the class of NP-hard problem and act as tested for new optimization algorithm. Hence, the instances ei51, st70, kroA100, kroC100, lin105 and tsp225 are chosen as the test dataset which is obtained from the standard library of TSPLIB.

The algorithm is implemented in Matlab and the result analysis is done using fitness function to check the performance of the algorithm for single objective function. For multiobjective TSP the metric used to measure the performance is Inverted Generational Distance (IGD). Therefore, metrics used to evaluate the performance of the proposed algorithm is given below:

Fitness Function

$$Fit = \min \left\{ \left(\sum_{j=1}^P dist(C_i, C_{i+1}) \right) + dist(C_p, C_1) \right\}$$

Whereas,

- P refers to the number of cities in the individual,
- $dist(C_i, C_{i+1})$ Refers the distance between two cities C_i and C_{i+1} ,
- $dist(C_p, C_1)$ Refers to the distance between last city and first city during return after the tour.

The fitness value is one of the important assessment criteria which give the tangible result of the optimal solution. Each algorithm was run on each instance 25 times and hence the best among the 25 runs are considered for analysis and validation purposes.

Inverted generational distance (IGD) [12] [13]

Let A^* be a set of uniformly distributed Pareto Optimal points in the Pareto front (PF). Let A be an approximation to the PF. The IGD metric is defined as follows,

$$IGD(A^*, A) = \frac{\sum d(v, A)}{|A^*|}$$

Where $d(v, A)$ is a minimum distance between v any point in A , and $|A^*|$ is the cardinality of A^* . The IGD metric can measure both convergence and diversity. Lower the IGD value better the solution is. To have low IGD value, A must be close to the PF and cannot miss any part of the whole PF.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

Table 1: Computational results of MOGA, P-MOGA (5%, 10% and 20%) in term of fitness

S. No	TSP Instance	Technique	Optimum value	Fitness	
				Best	Average
1	eil51	MOGA	426	439.45	457.89
		P-MOGA (5%)		436.23	444.16
		P-MOGA (10%)		439.83	454.62
		P-MOGA (20%)		439.26	456.26
2	kroA100	MOGA	21282	21783.6	22451.56
		P-MOGA (5%)		21330.07	21330.8
		P-MOGA (10%)		21745.43	22745.76
		P-MOGA (20%)		21733.03	21733
5	Tsp225	MOGA	3919	4401.67	4401.2
		P-MOGA (5%)		4209.32	4209.3
		P-MOGA (10%)		4408.78	4508.3
		P-MOGA (20%)		4407.54	4627.3

To judge the effectiveness of the proposed algorithm, we applied the algorithm to three standard TSP problems taken from TSPLIB. In table I the computational results of MOGA, P-MOGA (5%, 10% and 20%) algorithms on three different datasets are calculated with respect to fitness in terms of distance to solve the TSP. From table 1 it is shown that proposed algorithm outperforms MOGA based on fitness on the above TSP instances for single objective problem. Hence, it is working well for TSP and it is tested for MOTSP with two objectives. The performance measure is shown in IGD metric below for MOTSP in table 2.

Table 2: Results(Mean,SD) of MOGA, P-MOGA (5%, 10% and 20%) over 30runs

S. No	TSP Instance	MOGA	P-MOGA (5%)	P-MOGA (10%)	P-MOGA (20%)
1	eli51	461.8±25.8	447.7±11.7	463.9±27.9	470.6±34.6
2	kroA100	22884.5 ±1001.5	22177.3 ±294.3	22757.9±874.9	23395.9±1512.9
3	kroC100	22081.3±744.3	21656.2±319.2	21990.9±653.9	22231.8±894.8
4	lin105	15754±983	14899.7±128.7	15232.7±461.7	15813.2±1042.2
5	tsp225	4556.9±83.9	4515.2±42.2	4524.1±51.1	4626.9±153.9

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

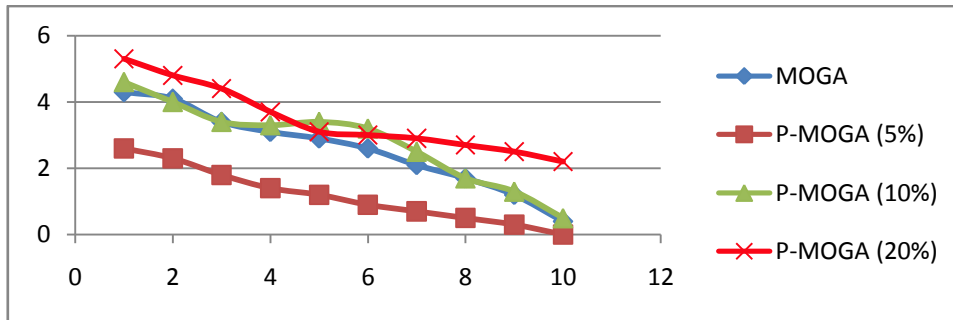


Fig 1(a)

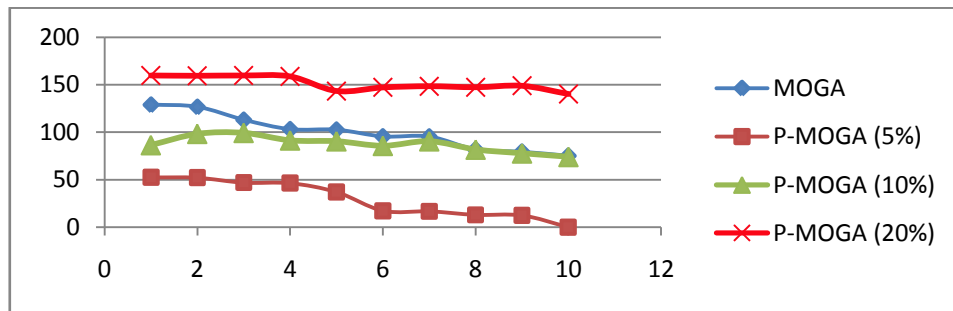


Fig 1(b)

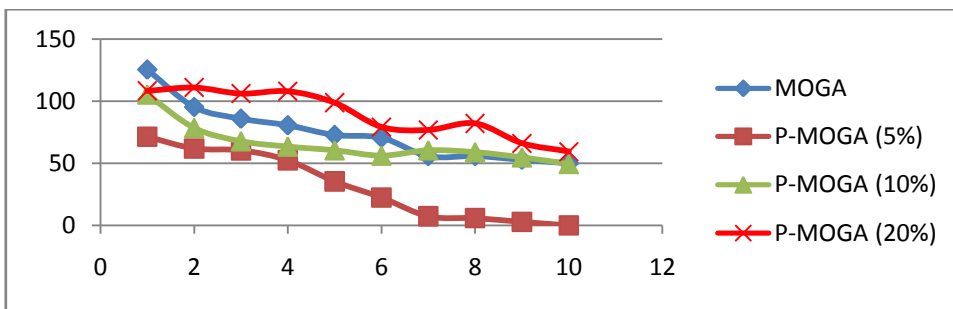


Fig 1(c)

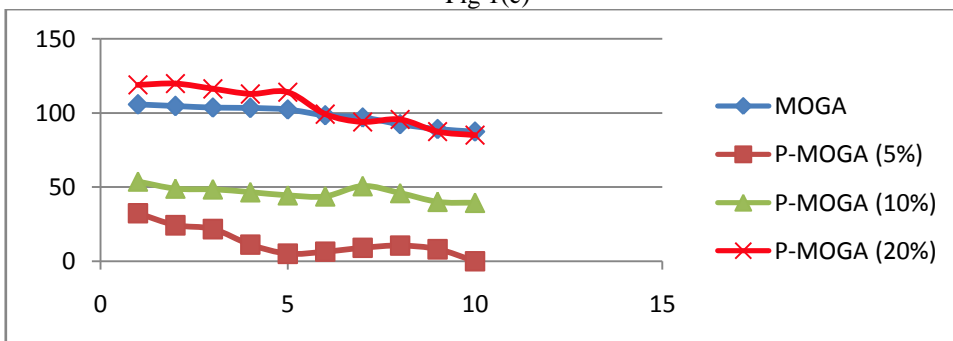


Fig 1(d)

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

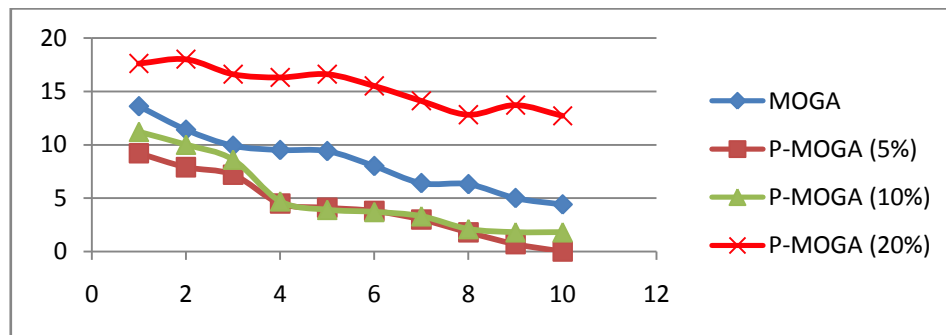


Fig 1(e)

Fig 1(a) – 1(e) Mean and SD of the algorithms over 30 runs on the MOTSP instances in terms of IGD

Table 2 shows the calculated mean and standard deviation of the MOGA and P-MOGA (5%, 10% and 20%) algorithms over 30 independent runs on the MOTSP instances in terms of IGD value and the fig 1(a) – 1(e) shows the IGD values of the instances in the form of graphical representation. The instances used for MOTSP are eli51, kroA100, kroC100, lin105 and tsp225 respectively for IGD calculation. In all the instances IGD value of P-MOGA is very less which implies that the algorithm solves the problem efficiently than the others and it produces good individuals at each generation on multiobjective problem. From the table 2, P-MOGA (20%) performing worse than the MOGA in all instance. In instance eli51, MOGA and P-MOGA (10%, 20%) are all performing like a same and in instance tsp225 P-MOGA (5%) performing the best and P-MOGA (10%) performing nearer to 5% perturbation and from this result analysis it is observed that, when the number of cities is more then, it is efficient to solve MOTSP with data perturbation.

V. CONCLUSION

In the present work, we introduced a new efficient component in MOGA algorithm: the data perturbation. We showed that by using data perturbation combined with MOGA operators work well to solve the multiobjective TSP. Perturbed MOGA is implemented with d parameter varies from 5 to 20% and compared with general MOGA. The optimal solution is given when the d parameter is kept 5% and the perturbation above or below 5% is not giving an optimal or a standard solution for any of the instances. The robustness of the Perturbed Multiobjective Genetic Algorithm has been clearly illustrated with respect to fitness value for single objective problem and IGD for multiobjective problem hence Perturbation solves the MOTSP problem optima.

REFERENCES

- [1] Kylie Bryant, A. Benjamin, "Genetic Algorithms and the Traveling Salesman Problem", Department of mathematic Harvey Mudd college, December, 2000.
- [2] X. Meng, J. Li, S. Member, X. Dai, and J. Dou, "Variable Neighborhood Search for a Colored Traveling Salesman Problem," IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, vol. 19, no. 4, pp. 1018–1026, 2018.
- [3] Philip, "A Genetic Algorithm for Solving Travelling Salesman Problem," International Journal of Advanced Computer Science and Applications, vol. 2, no. 1, pp. 26–29, 2011.
- [4] A. Kai, "A simple algorithm for solving travelling salesman problem", 2012 Second IEEE Conference on Instrumentation & Measurement, Computer, Communication and Control, pp. 1–5, 2012.
- [5] S. G. E. Brucal and E. P. Dadios, "Comparative Analysis of Solving Traveling Salesman Problem using Artificial Intelligence Algorithms," IEEE, 2017.
- [6] R. Takahashi, "Quantitative Evaluation of Iterative Extended Changing Crossover Operators to Solve the Traveling Salesman Problem", 10th International Conference on Natural Computation, pp. 235–244, 2014.
- [7] A. Al-dallal, "Using Genetic Algorithm with Combinational Crossover to Solve Travelling Salesman Problem." Ahlia University, (n.a).
- [8] L. Settat, "A new approach based a genetic algorithm for the Selective Travelling Salesman Problem", IEEE, no. 2, pp. 785–788, 2016.
- [9] S. Lianshuan, "An Improved Pareto Genetic Algorithm for Multi-Objective TSP", Fifth International Conference on Natural Computation, pp. 585–588, 2009, pp. 585–588, 2009.
- [10] K. Bakhayt, "Using Proposed Hybrid Algorithm for Solving the Multi Objective Traveling Salesman Problem," Global Journal of Pure and Applied Mathematics, vol. 12, no. 1, pp. 711–722, 2016.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 4, April 2018

- [11] W. Dong, K. Sheng, G. F. Hertono, B. D. Handari, B. Ahmed, and S. S. Chouhan, "A hybrid genetic algorithm for solving bi-objective traveling salesman problems", IOP Conf. Series: Journal of Physics: Conf. Series 887 (2017).
- [12] S. Maity, A. Roy, and M. Maiti, "An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem," in Expert Systems with Applications, 2016, vol. 46, pp. 196–223.
- [13] H. Zhang, A. Zhou, S. Song, Q. Zhang, and S. Member. (2016). A Self-Organizing Multiobjective Evolutionary Algorithm. IEEE Transactions on Evolutionary Computation, vol. 20, no. 5, pp. 792–806, 2016.
- [14] V. L. Vachhani and H. B. Prajapati, "Survey of Multi Objective Evolutionary Algorithms", 2015 International Conference on Circuit, Power and Computing Technologies, 2015.
- [15] Y. Hou, N. Q. Wu, M. C. Zhou, and Z. W. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, vol. 47, no. 3, pp. 517–530.
- [16] Chong, J. K. (2016). An Opposition-Based Self-adaptive Differential Evolution with Decomposition for Solving the Multiobjective Multiple Salesman Problems, 4096–4103.
- [17] A. Hussain, "Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator", Computational Intelligence and Neuroscience, Volume 2017 (2017).
- [18] C. R. Lopes, "Using Genetic Algorithms to minimize the distance and balance the routes for the multiple Traveling Salesman Problem", IEEE, pp. 3171–3178, 2015.
- [19] C. Changdar, G. S. Mahapatra, and R. Kumar Pal, "An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness," in Swarm and Evolutionary Computation, 2014, vol. 15, pp. 27–37.
- [20] R. J. Kuo, "A Fuzzy Multi-Objective Vehicle Routing Problem for Perishable Products Using Gradient Evolution Algorithm", IEEE conference on Industrial Engineering and applications, 2017.
- [21] K. Katrin, "Using genetic algorithms to solve large-scale airline network planning problems" Transportation Research Procedia, vol. 10, no. July, pp. 900–909, 2015.
- [22] D. M. Pierre and N. Zakaria, "Stochastic partially optimized cyclic shift crossover for multi-objective genetic algorithms for the vehicle routing problem with time-windows," in Applied Soft Computing Journal, 2017, vol. 52, pp. 863–876.
- [23] X. Sun, "Study on A Dynamic Traffic Route Choice Model with Travel Time Reliability Constrains of Intelligent Transportation Systems", IEEE, 2014.
- [24] M. Xu, S. Li, and J. Guo, "Optimization of Multiple Traveling Salesman Problem Based on Simulated Annealing Genetic Algorithm", MATEC Web of Conferences, vol. 25, 2017.
- [25] Y. Expressway, G. Noida, G. Nagar, and U. Pradesh, "A Survey : Swarm Intelligence vs . Genetic Algorithm", International Journal of Science and Research, vol. 3, no. 5, pp. 231–235, 2014.
- [26] T. Lust and J. Teghem, "Two-phase Pareto local search for the biobjective traveling salesman problem", Springer Science, pp. 475–510, 2010, 2009.
- [27] M. Cornu, T. Cazenave , D. Vanderpooten, "Perturbed Decomposition Algorithm applied to the multi-objective Traveling Salesman Problem. Computers & Operations Research, 79, 1–17. <https://doi.org/10.1016/j.cor.2016.04.025>, 2016.
- [28] T. Lust and A. Jaskiewicz, "Speed-up techniques for solving large-scale biobjective TSP," Comput. Oper. Res., vol. 37, no. 3, pp. 521–533, 2010.
- [29] T. Lust and A. FNRS, "New metaheuristics for solving {MOCO} problems : application to the knapsack problem, the traveling salesman problem and {IMRT} optimization," 2010.
- [30] L. Paquete and T. Stutzle, "A two-phase local search for the biobjective traveling salesman problem," Evol. Multi-Criterion Optim. Proc., vol. 2632, pp. 479–493, 2003.