A Perturbed Self-organizing Multiobjective Evolutionary Algorithm to solve Multiobjective TSP

J. Kanimozhi[1], V. Swathilakshmi[2], B. Thiyagarajan[3], R. Subramanian[4]

[1,2,3] Sri Manakula Vinayagar Engineering College, Puducherry, India.
janathakani@gmail.com, swathilakshmi60@gmail.com, thiyagarajan0484@gmail.com

[4] Pondicherry University, Puducherry, India
rsmanian.csc@pondiuni.edu.in

Abstract: Travelling Salesman Problem (TSP) is a very important NP-Hard problem getting focused more on these days. Having improvement on TSP, right now, consider the multi-objective TSP (MOTSP), broadened occurrence of the travelling salesman problem. Since TSP is NP-hard issue MOTSP is additionally an NP-hard issue. There are a lot of algorithms and methods to solve the MOTSP among which Multiobjective evolutionary algorithm based on decomposition is appropriate to solve it nowadays. This work presents a new algorithm which combines the Data Perturbation, Self-Organizing Map (SOM) and MOEA/D to solve the problem of MOTSP, named Perturbed Self-Organizing multiobjective Evolutionary Algorithm (P-SMEA). In P-SMEA Self-Organizing Map (SOM) is using extract neighborhood relationship information and with MOEA/D subproblems are generated and solved simultaneously to obtain the optimal solution. Data Perturbation is applied to avoid the local optima. So by using the P-SMEA, MOTSP can be handled efficiently. The experimental results show that P-SMEA outperforms MOEA/D and SMEA on a set of test instances.

Keywords: **Multiobjective TSP, Self-Organizing Map, Data perturbation, Decomposition based MOEA, Population, Fitness value.**

(Article history: Received: 24th April 2020 and accepted 26th June 2020)

I. Introduction

The Traveling Salesman Problem (TSP) is an upgrade issue used to find the most constrained path through the given number of urban cities. TSP expresses that given various urban areas N and the separation distance or time to go between the urban communities, the explorer needs to experience all the given urban refers to decisively, once and return to an equivalent city from where he started and more over the cost of the way is constrained. This pathway is called as the visit and the way length or travel time is the expense of the way [1] - [7]. The TSP mathematical model follows:

$$\min X = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}T_{ij}$$

Where,

$\sum_{i=1}^{N} P_{ij} = 1$ j varies from 1to N

$\sum_{j=1}^{N} P_{ij} = 1$ i varies from 1 to N

Here Tij is the time of travel between i-th urban city to j-th urban city. Here Pij = 1 represents there exist a path between the city-i to city-j, otherwise Pij = 0. X is the minimum distance of the optimal path [7].

Multi Objective Travelling Salesman Problem (MOTSP) is a multiobjective problem considers more objectives to find the optimal path. Given N cities and D distance between every pair of unmistakable urban areas to travel, the MOTSP comprises in finding a Hamiltonian pattern of the N urban areas that advances the accompanying minimization problems [2] - [9]:

$$(1) \quad \min X1 = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}D_{ij}$$

$$(2) \quad \min X2 = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}T_{ij}$$

$$(3) \quad \min X2 = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}F_{ij}$$

Where,

$\sum_{i=1}^{N} P_{ij} = 1$ j varies from 1 to N

$\sum_{j=1}^{N} P_{ij} = 1$ i varies from 1 to N

The above equation represents the scientific model of MOTSP by taking two instances for finding the optimal path. The first objective function works for the minimization of the distance traveled by the salesperson, while the second instance function takes the travelling time of the salesperson to reach the city. Here Dij is the distance between city i to j, Tij is the travelling time between city i o j, Fij is the path condition and Pij = 1 if the salesperson travels from city i to j, otherwise Pij = 0 [7].

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 II-C. Since MOTSP is a multiobjective problem it needs to be solved using MOEA's. The objectives in MOTSP are clashing with one another, for instance a path with the briefest length in one objective may likewise the most expensive in another. Subsequently, there is unquestionably not a solitary visit that can limit all of the goals at the same time [5] [8] [10]. Thus, w1, w2 and w3 means the loads that are utilized to adjust between the complete traveling distance and the travelling time with the end goal that the entirety of the destinations is 1.0 (w1+w2+w3=1.0) [3]. To make the single objective TSP as multiobjective problem, load is balanced between multiple objectives. Evolutionary algorithms are the best to deal with multiobjective problems, but some time it may lead to local optima. To avoid that situation data perturbation is used. A Data perturbation move is a technique used to escape from local optima. Instead of modifying the starting solution, DP suggests to modify input data. So by combining the data perturbation with multiobjective evolutionary algorithms, the solutions are given better than with MOEA. In the proposed algorithm data perturbation is applied to avoid the local optimal solution.

This paper presents A Perturbed Self-Organizing Multiobjective Evolutionary Algorithm (P-SMEA). The framework of P-SMEA combines Self Organizing Map (SOM), MO Decomposition and Data Perturbation. This papers sections are composed as follows: Section II reviews the meaning of MOP definition, data perturbation techniques and literature review of meta-heuristic algorithms to deal with MOTSP. After this section an introduction to SOM, MOEA and the proposed P-SMEA algorithm and its components are given in section III. Section IV explains about the experimental result analysis and finally the conclusion is given in section V.

II. PRIMARIES

We recall the basic definition of multi objective optimization first and data perturbation concept, then finally a literature review of the meta-heuristic methods to solve MOTSP.

A. *Multi Objective Optimization*

A general multiobjective optimization problem (MOPs) is defined below [9] - [11]:

$$\min F(X) = (f1(X), \ldots fm(X))$$

$$\text{where } X = (x1, \ldots xn) \in \Omega$$

Where X = (x1,… xn) is a choice variable vector, Ω = [ai, bi]n is the achievable region of the hunt space, F : Ω →Rm comprises of m target capacities fi(X), i= 1,… ,m, and Rm indicates the goal space. Let the vectors u, v ∈ Rm, u command v if and just if ui ≤ vi for each I ∈ {1, ...,m} and uj < vj for at any rate one file j ∈ {1, ...,m}1.

A possible solution x∗ ∈ X is called proficient if there doesn't exist any other achievable arrangement x ∈ X with the end goal that z(x) ≺ z(x∗). The picture z(x*) in target space of an effective solution x* is known as a non-dominated point. The effective set indicated by XE contains all the proficient solutions. The picture of the proficient set in Z (Objective space) is known as the Pareto front (or non-commanded wildest), and is signified by ZN.

B. *Data Perturbation (DP)*

The initial population size, diversity and convergence property of initial population influences more on the optimal solution [9]. "Data Perturbation" (DP) strategy, proposed by Codenotti et al. for the single-objective TSP and has been presented in MO enhancement by Lust and Teghem [9] [11]. A perturbation move is a technique used to escape from local optima. Instead of modifying the starting solution, DP suggests to modify input data [9]. The annoyance is a twofold scaffold move [14] that cuts the momentum visit at four reasonably pressed edges into four sub-visits and reconnects these in a substitute solicitation to yield another starting visit for the local pursuit.

There are two methods to do the DP which are given in [9] [11] - [14]. The primary DP technique in [9] [12] [13] is begun with the info parameters number K of cycles, three parameters that decide the perturbation scheme (the fixed scale SF, the shifting scale SV and the neighborhood bother LP) and the cost frameworks $C_{i,j}^{k}$ of the MOTSP. During a cycle k, first figure a weight set λ by following a straight plan; K consistently circulated weight sets are therefore produced. We at that point make another cost matrix Cλ. At that point, we marginally annoy each cost Cλ(i, j) of the framework Cλ to discover new possibly effective arrangements. An increment in the number K of emphasis gives a significant improvement of the markers and permits arriving at astounding outcomes, since the quantity of conceivably effective arrangements |PE| is expanding while the separation D1 is diminishing. The quantity of emphasis for the quantity of annoyance steps is equivalent to the quantity of urban areas N short 50.

The second DP method in [11] is done with a single parameter d, whereas the above one needs three parameters. Higher the d value, the larger the perturbation is. It gives an anonymous 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 percent variation. The best results are given by 5 %. So for optimal result used d=5%. This second method is used in our work since the execution time is less compared with the first method.

C. *Literature review of meta-heuristic algorithms applied to MOTSP*

This section gives the multi objective evolutionary algorithms to solve MOTSP. The algorithms used to solve MOTSP are Multiobjective genetic algorithm (MOGA), Multiobjective Ant colony optimization (MOACO), and its variants which are listed and explained in [7].

Multiobjective genetic algorithm is used to solve TSP and

MOTSP [15] – [20]. MOGA is combined with a fuzzy system [15], Ant colony optimization [18], and different crossover and mutation methods are used to solve the MOTSP. In [20] they proposed an algorithm called MOGA to work with the vehicle routing problem. GA is used to solve TSP in [21] and compared with tabu search, PSO and greedy algorithms. Among all those algorithms GA outperforming to solve TSP with a single objective. GA, PSO and ACO algorithms are explained with its advantages and suitable problems to solve [22].

In [2] Ant colony optimization is combined with decomposition based MOEA to solve MOTSP which produce a better solution to the problem than solving it with ACO. The flow shop scheduling problem is solved using MOACO in [23]. Particle swarm optimization is combined with the ACO to solve TSP in [24]. MOACO is used to solve bi-objective TSP in [25].

NSGA II is used to solve MOTSP in [26], where individuals are selected based on the rand and crowding distance. It is giving better results than MOGA. NSGA II is hybridized with MOGA in [27]. The initial population is calculated using the way used in Multiobjective Differential Evolution algorithm (MODE) and followed with NSGA II. It's giving better result that the general NSGA II. Still improvement is made in NSGA II [28] which is an improved NSGA II. They have used the arena's principle to construct non-dominance set which reduce the dominance count and order crossover operator and an inversion mutation operator also used in it. Fuel utilization minimization for vehicle steering issue is settled utilizing NSGA II in [29].

Decomposition based MOEA combined adaptive guidance algorithm (AG-MOEA/D) uses the concept of differential evolution algorithm and solves the problem (MOEA/D-DE) [3] [30]. Dynamic multi objective TSP is solved using general MOEA/D in [4]. Estimation of distribution algorithm is combined with MOEA/D and used to solve MOTSP in [5] [6] [31]. MOEA/D is combined with the ACO in [2] [32] which follow the decomposition method to decompose the problem and ACO to solve the subproblems. MOEA/D is combined with multi-objective chemical reaction based decomposition algorithm (MOCRO/D), to solve MOTSP [8]. From [7] and [33] it is proved that MOEA/D outperforms all the above mentioned algorithms to solve MOTSP.

III. A PERTURBED SELF-ORGANIZING MULTIOBJECTIVE EVOLUTIONARY ALGORITHM (P-SMEA)

There are many algorithms available to solve MOTSP in the domain of evolutionary algorithm. They are Multiobjective Genetic Algorithm, PSO, Multiobjective ACO, NSGA II, MOEA/D and its variations which are explained in the above section. Among the listed algorithms, MOEA/D outperforms to solve MOTSP [7] [33]. MOEA/D solves the given problem by decomposed into subproblems and the solutions of each subproblem is combined together to get an optimal solution. Here the number of subproblems plays a major role to get the optimal solution. But the number of subproblems needs to be decomposed, should be given by the user manual which leads to two different issues in MOEA/D [11]. One is, the new created children are of similar to the parents and another is it spoils the diversity property. Since it is lacks in learning about the neighborhood information.

The issue with the MOEA/D subproblem decomposition gives the need for SOM to learn neighborhood information as explained in [10] [11]. To avoid this problem SOM is used to learn the neighborhood information and it will be continued with MOEA/D process and the algorithm named as self-organizing MOEA (SMEA). Still to improve the problem solution data perturbation is used. It slightly modifies the input data to get a better solution than with SMEA. The motivation for doing the perturbation is to avoid the local optima [9] [11] [12] and the algorithm proposed in this paper is named as Perturbed Self-Organizing Multiobjective Evolutionary Algorithm (P-SMEA). The rest of this chapter explains about the Introduction of Self-Organizing Map (SOM) and the proposed Perturbed Self-Organizing Multiobjective Evolutionary Algorithm (P-SMEA).

A. Introduction of Self-Organizing Map (SOM) in MOEA

SOM algorithm, introduced by Kohonen, is an unsupervised learning method, which provides the topological relationship between the information utilizing the learning algorithm [33] [34]. Fig 1 shows the illustrations of SOM, where X is the input neurons, which get the input as city coordinates and map it to the output neurons which are fully connected with the input neurons. Z is the position representation and W is the weight vector of neurons [10] [33] [37] [38].

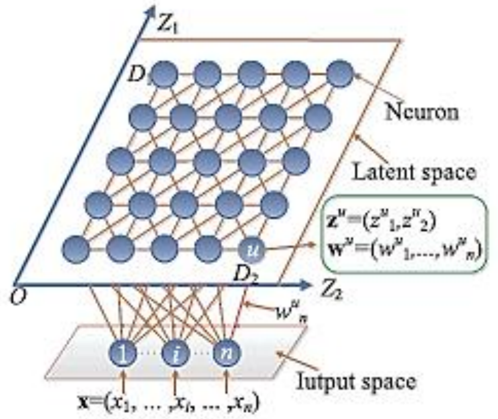

Fig 1. 2D SOM Illustration

Initially SOM is used to deal with single objective TSP using the learning algorithm [33] – [40]. The efficiency of solving TSP using SOM made the research by combining SOM and MOEA. In [41] SOM combined with MOEA and the results is better that solving it with SOM or with MOEA and in [42] water distribution problem is efficiently solved with SO-MOEA. Finally SOM combined with MOEA/D to learn neighborhood information and problems can be solved efficiently [10] [43]. In SOM the initialization method is modified to give an efficient initial population [33] which gives the need and of data perturbation in the proposed work.

B. *P-SMEA framework*

This section presents the Perturbed Self-Organizing Multiobjective Evolutionary Algorithm (P-SMEA) characteristics, flowchart and algorithm steps. Characteristics of P-SMEA follow:

- Initialization of P-SMEA starts with the data perturbation of cost matrix. The initial cost matrix is computed using data set and using the d parameter, cost matrix is modified.
- SOM training step is conducted first, then continued with population developing step and it will be conducted in the loop.
- A neighborhood relationship established by the SOM is used to generate new solutions.

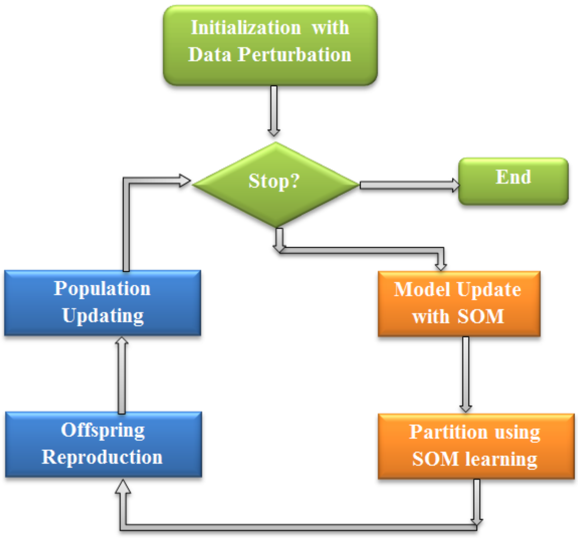

Fig 2. P-SMEA framework

Fig 2 gives the framework of P-SMEA, which starts with Data perturbation and continued with SOM process, followed by Evolutionary steps. In the figure, initialization with data perturbation is perturbation of initial cost matrix of MOTSP, Model update with SOM demonstrates the refreshing of neighborhood range, learning rate and neuron weight vector with the assistance of training data. The Partition utilizing SOM learning indicates the solution grouping dependent on the neighborhood information. Offspring reproduction is restricted within the neighboring solutions dependent on the found neighborhood data [10]. Population update will be done with the new offspring's generated.

The following are the notations used in the description of P-SMEA.

A unique weight λk = (λ1k,…. λjk)

$C^k(e) = \sum_{j=1}^{p} \lambda_j^k \ C_j(e)$ is the initial cost function

N = N1× • • • ×Nm−1: Number of neurons, where N is equal to the population size.

τ0: Initial SOM learning rate.

σ0 =(1/2) $\frac{\sqrt{\sum_{i=1}^{m-1} N_i^2}}{(m-1)}$: Initial neighborhood radius

H: Neighborhood mating pool size

N: Size of Population

T: Maximum number of generations

Algorithm 1 gives the Framework of P-SMEA and data perturbation steps in algorithm 2. The Algorithm1 (P-SMEA) starts with the Cost matrix (C^k) and continued with the data perturbation (Algorithm 2). The DP adds noise into the multi objective cost function. The data perturbation parameter $d \geq 0$, will limit the greatest variety of noise added with the cost. The v value 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 × $\sum_{j=1}^{p} \lambda_j^k \ C_j(e)$ and with experimental analysis, it is proved that d value must be equal to 5% for optimal solution.

Algorithm 1: P-SMEA Framework

Input: Multi-objective TSP, a stopping criterion, data perturbation parameter d, SOM parameters τ0, σ0

Output: set A of efficient solutions

Begin

Let C^k be the cost matrix related to MOTSP

Data perturbation (C^k, d)

Randomly initialize the population P = {x1,…… xn}

Set initial training set S = P and neuron weight vector {w1,…….. wn} and the uk be the index of the kth nearest neuron to neuron u.

For i=1:T

For each xs ∈ S, s varies from 1to |S|

Update the SOM training parameters:

σ = σ0 × $\left(1 - \frac{(t-1)N+s}{TN}\right)$

τ = τ0 × $\left(1 - \frac{(t-1)N+s}{TN}\right)$

Find the closest neuron to xs

u'= arg $arg \min_{1 \leq u \leq N} \|x^s - w^u\|_2$

Locate and update the neighboring neurons

U = $\left\{u \middle| \ 1 \leq u \leq N \wedge \left\|z^u - z^{u'}\right\|_2 < \sigma\right\}$

wu = wu + τ . exp$\left(-\left\|z^u - z^{u'}\right\|_2\right)$ $(x - w^u)$

end

Generate a new solution y

Do crossover and mutation within the neighborhood mating pool (H)

Archive the best individual

Update the population P
End
Return the Population P
End

Framework continues with the perturbed cost matrix and the initial population (P) is given random from p1….pn followed by SOM parameters are initialized. The variables σ0, τ0 and T represents the initial neighborhood radios, and initial learning rate and maximum number of iterations respectively. The SOM learning process starts with initializing neurons and assigning each with weight vectors. The neurons closer to the selected input pattern will be identified which is a winning neuron. The weight vector of winning neuron neighbors is updated and which gives the neighborhood relationship for further process. The neighborhood relationship produced by SOM is used for crossover and mutation operators by EA. The crossover has taken place between the neighboring solutions and the best individuals will be updated in the population and the process continues till the termination condition met.

Algorithm 2: Data perturbation (C^k, d)

Input: Cost matrix C^k and data perturbation parameter d
Output: Perturbed cost matrix
For each e ∈ E do

v ← U (1-d, 1+d)

$C^k(e) \leftarrow v \times \sum_{j=1}^{p} \lambda_j^k \; C_j(e)$
End
Return perturbed cost matrix

As in [44] MOEA can be gathered into elitist and non-elitist computations. Elitist MOEAs have a component to secure great solution at every generation while non-elitist MOEAs don't have such framework. In our work elitism based MOEA is employed so that, optimal solutions are archived at each iteration for better result. Section IV continues with the metrics used to measure the algorithm performance.

IV. RESULT ANALYSIS

By combining the SOM, data perturbation and MOEA/D, a new algorithm called Perturbed Self-organizing Multiobjective Evolutionary Algorithm based on Decomposition (P-SMEA) is implemented in the last section. In this section MOTSP is solved by using the P-SMEA and the results are shown below. In order to assess the performances of the P-SMEA algorithm, the instances of TSP are taken from TSPLIB which are instance eil51, st70, kroA100, kroC100, lin105 and tsp225.

The algorithm is implemented in Matlab and the result is analyzed using fitness function, convergence and error rate to check for the performance of the algorithm on the single objective problem [45]. For multiobjective TSP, the metric used is Inverted Generational Distance (IGD) as in [10] [19]. Therefore, metrics used to evaluate the performance are 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 [7],

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

The fitness value is one of the noteworthy evaluation criteria which give the unmistakable result of optimal solutions. Every algorithm was run on each instance 30 times and in this way the best among the 30 runs are taken for investigation and approval purposes.

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

Let A* be a lot of reliably disseminated Pareto Optimal focuses on the Pareto front (PF). Let A be an estimate to the PF. The IGD metric is portrayed as follows,

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

Where d (v, A) is a minimum distance between v and any point in A, and |A*| is the cardinality of A*. The IGD metric can quantify both convergence and diversity. Lower the IGD esteem, better the solution is. To have a low IGD regard, A unquestionable requirement be close to the PF and can't miss any piece of the entire PF.

These above assessment criteria structure a strong base for demonstrating the presentation of the proposed Perturbed P-SMEA in solving single objective TSP and MOTSP.

The table I show the computational results of the algorithms on single objective TSP based on fitness, convergence, average convergence and error rate. The fitness esteem is one of the noteworthy assessment criteria which give the substantial result of the optimal solution.

TABLE I. COMPUTATIONAL RESULTS OF SMEA, P-SMEA (5%, 10%, AND 20%) ON TSP

S. No	TSP Instance	Technique	Optimum value	Fitness		Convergence rate (%)	Error rate (%)	Average Convergence (%)
				Best	Average			
1	eil51	SMEA	426	439.45	457.89	96.8	3.1	92.5
		P-SMEA (5%)		436.23	444.16	97.6	3.1	92.5
		P-SMEA (10%)		439.83	454.62	96.2	3.1	92.5
		P-SMEA (20%)		439.26	456.26	92.9	3.1	92.5
2	st70	SMEA	675	701.27	740.09	96.10	3.89	90.35
		P-SMEA (5%)		690.85	726.93	97.65	2.3	92.30
		P-SMEA (10%)		701.19	701.19	96.11	3.8	91.97
		P-SMEA (20%)		710.33	710.32	94.76	4.5	90.96
3	kroA100	SMEA	21282	21783.6	22451.56	97.6	2.3	92.5
		P-SMEA (5%)		21330.07	21330.8	99.7	2.3	95.2
		P-SMEA (10%)		21745.43	22745.76	97.8	2.3	93.8
		P-SMEA (20%)		21733.03	21733	97.7	2.3	93.3
4	kroC100	SMEA	20749	21314.06	22524.78	97.2	2.7	91.44
		P-SMEA (5%)		21000.54	22038.53	98.8	2.7	93.76
		P-SMEA (10%)		21314.99	22105.3	97.6	2.7	92.46
		P-SMEA (20%)		21141.49	22185.49	97.1	2.7	92.3
5	Lin105	SMEA	14379	14707.69	15176.18	94.45	2.28	91.45
		P-SMEA (5%)		14614.3	14998.33	98.36	2.28	95.03
		P-SMEA (10%)		14782.5	15057.2	97.28	2.28	92.45
		P-SMEA (20%)		14719.3	15215.2	97.63	2.28	92.83
6	Tsp225	SMEA	3919	4401.67	4401.2	87.67	4.9	90.7
		P-SMEA (5%)		4209.32	4209.3	92.57	4.9	96.03
		P-SMEA (10%)		4408.78	4508.3	87.50	4.9	92.45
		P-SMEA (20%)		4407.54	4627.3	87.53	4.9	92.83

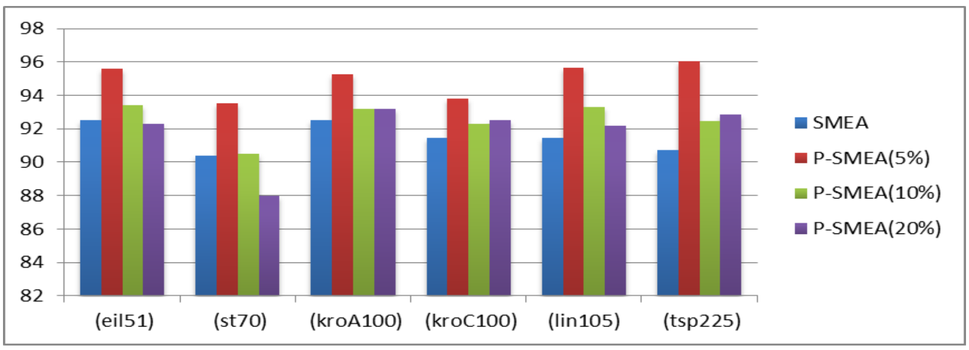

Fig 3. Performance evaluation based on Average Convergence

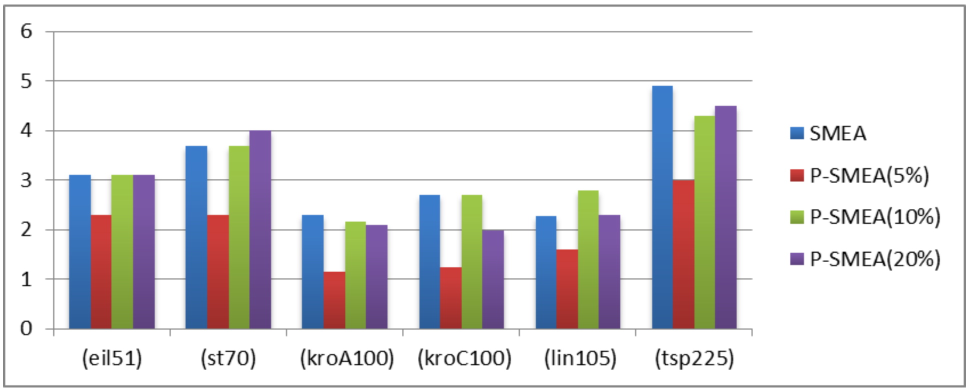

Fig 4. Performance evaluation based on Error rate

TABLE II. RESULTS (MEAN, SD) OF SMEA, P-SMEA (5%, 10%, 20%) OVER 30 RUNS.

S. No	TSP Instance	SMEA	P-SMEA (5%)	P-SMEA (10%)	P-SMEA (20%)
1	eli51	811.866±13.43	**660.0271±6.7113**	760.54±10.865	928.0114±16.105
2	St70	1496.252±50.33	**1262.455±16.1404**	1603.16±26.4331	1700±26.6565
3	kroA100	16711 ±6402	**11292 ±390.0847**	13953.54±874.665	18183.5±1681.417
4	kroC100	61150.12±556.31	**45096.44±472.84**	55689.54±647.82	65457.04±727.59
5	lin105	9226.342±221.22	**6157.44±148.67**	8432.91±192.93	9326.92±233.10

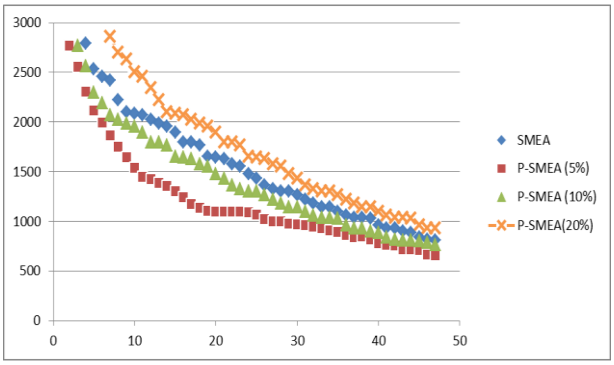

Fig 5(a)

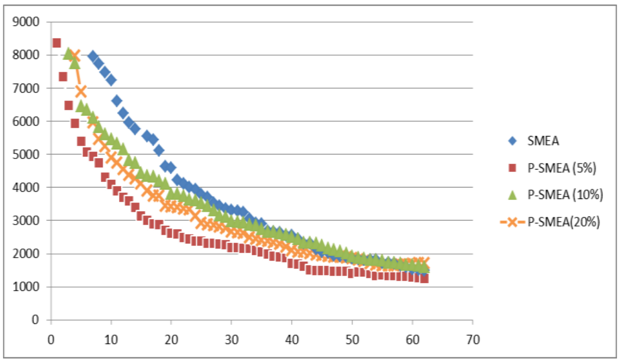

Fig 5(b)

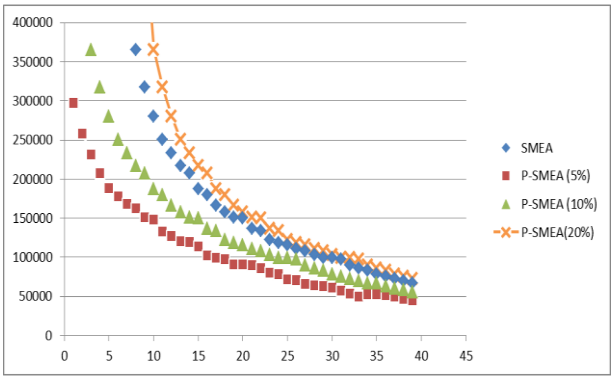

Fig 5(c)

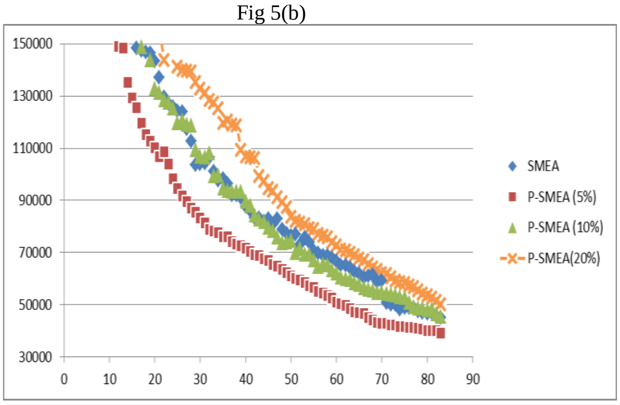

Fig 5(d)

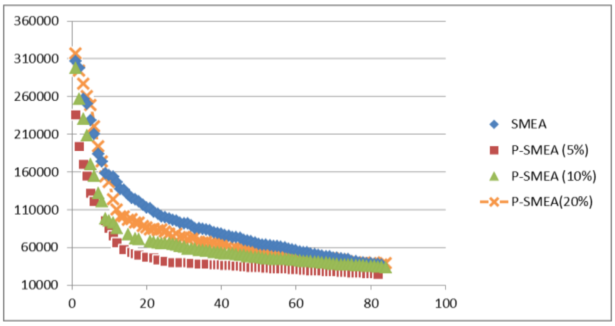

Fig 5(e)

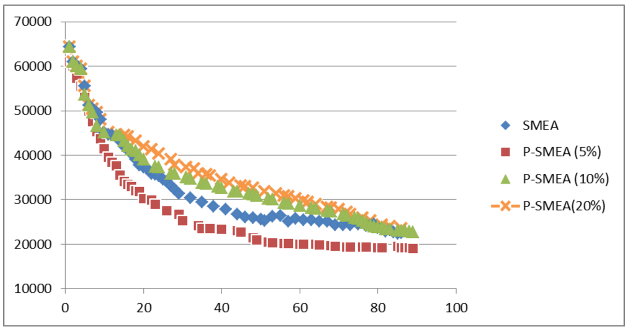

Fig 5(f)

Fig 5. Mean IGD metric versus numbers of function assessments for the four algorithms over 30 independent runs

Every algorithm was run on each instance 30 times and henceforth the best among the 30 runs are taken for investigation and validation purposes. The convergence rate demonstrates the nature of the optimal solution created from the populace. The evaluation of the proposed P-SMEA algorithm as far as the error rate is significant for the investigation. The best error rate demonstrates how far the best individual convergence rate goes amiss from the optimal fitness value while the most noticeably terrible error rate shows the distinction between the convergence pace of most noticeably terrible individual from the populace and the optimal solution [7]. With all the TSP instances, the proposed algorithm (P-SMEA (5%)), solves the problem with optimal fitness which is shown in table I. Fig 3 shows the convergence rate of the P-SMEA which is more contrasted with all the others; hence the quality of the population generation is better than the others. An error rate of the proposed algorithm in solving the problem is less and so the algorithm is not deviating from solving the problem as shown in the fig 4.

Table II shows the IGD mean and standard deviation of the different algorithms on eli51, st70, kroA100, kroC100, lin105 and tsp225 respectively. Fig 5(a) – 5(f) shows the Mean IGD values versus the number of function assessments of the four algorithms (SMEA, P-SMEA (5%), P-SMEA (10%), P-SMEA (20%)) over 30 runs. From the IGD values P-SMEA (20%) performing the worst for some instances and for some other instance it is giving better than SMEA. Hence the performance of P-SMEA (20%) cannot be predicted for any instances and it is not stable. P-SMEA (10%) performing similar to SMEA almost in all the instances and in all the instances mean IGD of P-SMEA (5%) is very less which implies that the algorithm solves the problem efficiently than the others on multiobjective problem.

V. Conclusion

There are many methods available to solve MOTSP in the field of evolutionary algorithm. Among which MOEA/D is performing better. In this proposed framework, a new efficient algorithm to solve Multiobjective TSP named Perturbed Self Organizing Multiobjective Evolutionary Algorithm (P-SMEA) is introduced by combining the data perturbation, SOM and decomposition based MOEA. From the result analysis by utilizing data perturbation with SMEA operators, multiobjective TSP is solved better than the available algorithms. Perturbed SMEA is implemented with d parameter varies from 5 to 20% and compared with general SMEA. The optimal solution is given when the d parameter is kept 5%. The strength of the Perturbed SMEA has been evidently proved with respect to the fitness value for single objective TSP problem and IGD for the multiobjective TSP problem.

Acknowledgment

I cannot express enough thanks to my guide Dr. R. Subramanian for his great guidance, support and encouragement throughout my research work to complete the algorithm development successfully. I thank a lot for the learning opportunities provided by him.

References

[1] Khanra, A., Maiti, M. K., & Maiti, M. (2015). Profit maximization of TSP through a hybrid algorithm. Computers and Industrial Engineering, 88, 229–236. https://doi.org/10.1016/j.cie.2015.06.018

[2] Souza, M. Z. De, Trinidad, A., & Pozo, R. (2014). A GPU Implementation of MOEA / D-ACO for the Multiobjective Traveling Salesman Problem, 6–11. https://doi.org/10.1109/BRACIS.2014.65

[3] Chong, J. K. (2016). An Opposition-Based Self-adaptive Differential Evolution with Decomposition for Solving the Multiobjective Multiple Salesman Problem, 4096–4103.

[4] Li, W. (2012). A Parallel Procedure for Dynamic Multi-objective TSP. https://doi.org/10.1109/ISPA.2012.10

[5] Gao, F., Zhou, A., & Zhang, G. (2012). An Estimation of Distribution Algorithm based on Decomposition for the Multiobjective TSP, (Icnc), 817–821.

[6] Shim, V. A., Tan, K. C., & Tan, K. K. (2012). A Hybrid Estimation of Distribution Algorithm for Solving the Multi-objective Multiple Traveling Salesman Problem. 2012 Ieee Congress on Evolutionary Computation, 42(5), 682–691. https://doi.org/10.1109/TSMCC.2012.2188285.

[7] Kanimozhi, J., Subramanian, R., (2017). A Survey on Multi-Objective Travelling Salesman Problem. Institute of Integrative Omics and Applied Biotechnology (IIOAB), 2, 223-233.

[8] Bouzoubia, S., & Chikhi, S. (2015). Multi-objective chemical reaction optimization based decomposition for multi-objective traveling salesman problem, (1).

[9] T. Lust and J. Teghem, (2009). Two-phase Pareto local search for the biobjective traveling salesman problem. Springer Science, pp. 475–510, 2010.

[10] 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.

[11] Cornu, M., Cazenave, T., & Vanderpooten, D. (2016). 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

[12] T. Lust and A. Jaszkiewicz, “Speed-up techniques for solving large-scale biobjective TSP,” Comput. Oper. Res., vol. 37, no. 3, pp. 521–533, 2010.

[13] T. Lust and A. FNRS, “New metaheuristics for solving {MOCO} problems : application to the knapsack problem , the traveling salesman problem and {IMRT} optimization,” 2010.

[14] 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.

[15] Changdar, C., Mahapatra, G. S., & Kumar Pal, R. (2014). An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness. Swarm and Evolutionary Computation, 15, 27–37. https://doi.org/10.1016/j.swevo.2013.11.001

[16] Lopes, C. R. (2015). Using Genetic Algorithms to minimize the distance and balance the routes for the multiple Traveling Salesman Problem, 3171–3178.

[17] Xu, M., Li, S., & Guo, J. (2017). Optimization of Multiple Traveling Salesman Problem Based on Simulated Annealing Genetic Algorithm, 25.

[18] Maity, S., Roy, A., & Maiti, M. (2017). Journal of Industrial Information Integration An intelligent hybrid algorithm for 4-dimensional TSP. Journal of Industrial Information Integration, 5, 39–50. https://doi.org/10.1016/j.jii.2017.02.001

[19] Maity, S., Roy, A., & Maiti, M. (2016). An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem. Expert Systems with Applications, 46, 196–223. https://doi.org/10.1016/j.eswa.2015.10.019

[20] Pierre, D. M., & Zakaria, N. (2017). Stochastic partially optimized cyclic shift crossover for multi-objective genetic algorithms for the vehicle routing problem with time-windows. Applied Soft Computing Journal, 52, 863–876. https://doi.org/10.1016/j.asoc.2016.09.039

[21] Antosiewicz, M., Koloch, G., & Kaminski, B. (2013). Choice of best possible metaheuristic algorithm for the travelling salesman problem with limited computational time : quality, uncertainty and speed. Journal of Theoretical and Applied Computer Science, 7(1), 46–55.

[22] Expressway, Y., Noida, G., Nagar, G., & Pradesh, U. (2014). A Survey : Swarm Intelligence vs . Genetic Algorithm, 3(5), 231–235.

[23] Yagmahan, B., & Yenisey, M. M. (2010). A multi-objective ant colony system algorithm for flow shop scheduling problem. Expert Systems with Applications, 37(2), 1361–1368. https://doi.org/10.1016/j.eswa.2009.06.105

[24] Khanra, A., Maiti, M. K., & Maiti, M. (2015). Profit maximization of TSP through a hybrid algorithm. Computers and Industrial Engineering, 88, 229–236. https://doi.org/10.1016/j.cie.2015.06.018

[25] Garcia-Martinez, C., Cordon, O., & Herrera, F. (2007). A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. European Journal of Operational Research, 180(1), 116–148. https://doi.org/10.1016/j.ejor.2006.03.041

[26] Ruben Ivan Bolanos, Mauricio Granada Echeverry and John Willmer Escobar, A multiobjective non-dominated sorting genetic algorithm (NSGA-II) for the Multiple Traveling Salesman Problem, T. S. (2015). Decision Science Letters, 4, 559–568. https://doi.org/10.5267/j.dsl.2015.5.003

[27] Psychas, I.-D., Delimpasi, E., & Marinakis, Y. (2015). Hybrid evolutionary algorithms for the Multiobjective Traveling Salesman Problem. Expert Systems with Applications, 42(22), 8956–8970. https://doi.org/10.1016/j.eswa.2015.07.051

[28] Luo, Y., Liu, M., Hao, Z., & Liu, D. (2014). An Improved NSGA-II Algorithm for Multi-objective Traveling Salesman Problem, 12(6), 4413–4418. https://doi.org/10.11591/telkomnika.v12i6.5476

[29] Psychas, I.-D., Marinaki, M., Marinakis, Y., & Migdalas, A. (2016). Non-dominated sorting differential evolution algorithm for the minimization of route based fuel consumption multiobjective vehicle routing problems. Energy Systems. https://doi.org/10.1007/s12667-016-0209-5

[30] Zhang, N., & Fan, Z. (2016). A Multi-phase Adaptively Guided Multiobjective Evolutionary Algorithm Based on Decomposition for Travelling Salesman Problem, 3284–3291.

[31] Zhou, A., Gao, F., & Zhang, G. (2013). A decomposition based estimation of distribution algorithm for multiobjective traveling salesman problems. Computers and Mathematics with Applications, 66(10), 1857–1868. https://doi.org/10.1016/j.camwa.2013.05.031

[32] Ke, L., Zhang, Q., & Battiti, R. (2013). MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and AntColony. IEEE Transactions on Cybernetics, 43(6), 1845–1859. https://doi.org/10.1109/TSMCB.2012.2231860

[33] W. D. Zhang, Y. P. Bai, and H. P. Hu, "The incorporation of an efficient initialization method and parameter adaptation using self-organizing maps to solve the TSP," Appl. Math. Comput., vol. 172, no. 1, pp. 603–623, 2006.

[34] T. A. S. Masutti and L. N. de Castro, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem," Inf. Sci. (Ny)., vol. 179, no. 10, pp. 1454–1468, 2009.

[35] P. H. Siqueira, M. T. A. Steiner, and S. Scheer, "A new approach to solve the traveling salesman problem," Neurocomputing, vol. 70, no. 4–6, pp. 1013–1021, 2007.

[36] H. Schabauer, E. Schikuta, and T. Weishäupl, "Solving very large traveling salesman problems by SOM parallelization on cluster architectures," Parallel Distrib. Comput. Appl. Technol. PDCAT Proc., vol. 2005, pp. 954–958, 2005.

[37] Y. Bai, W. Zhang, and Z. Jin, "A new self-organizing maps strategy for solving the traveling salesman problem," Chaos, Solitons and Fractals, vol. 28, no. 4, pp. 1082–1089, 2006.

[38] D. Markovi, V. Tomi, and S. Stojkovi, "SOLVING TRAVELLING SALESMAN PROBLEM BY USE OF KOHONEN," pp. 21–24.

[39] R. Sushkov (2015), Self-Organizing Structures for the Travelling Salesman Problem in a Polygonal Domain, CZECH TECHNICAL UNIVERSITY IN PRAGUE.

[40] X. Xu, Z. Jia, J. Ma, and J. Wang (2008) .A Self-organizing Map Algorithm for the Traveling Salesman Problem. 2008 IEEE Fourth International Conference on Natural Computation pp. 431–435.

[41] Dirk Buche, Michele Milano, Petros Koumoutsakos (n.a) Self –Organizing Maps for Multi-Objective Optimization. Institute of Computational Science.

[42] K Norouzi, G R Rakhshandeheroo (2011). A Self Organizing Map based hybrid Multi-Objective Optimization of water Distribution networks. IJST Transcations of Civil and Environmental Engineering, 35.pp 105-119.

[43] Hu Zhang, XiujieZhang, Xiao-ZhiGao, ShenminSong (2016). Self-organizing multiobjective optimization based on decomposition with neighborhood ensemble. Neurocomputing, 173.1868–1884.

[44] Vachhani, V. L., & Prajapati, H. B. (2015). Survey of Multi Objective Evolutionary Algorithms, IEEE Conference on Circuit, Power and Computing Technologies.

[45] Hassani Hicham, Benkachcha, Benhra (2015). Multi-Objective Optimization Using Genetic Algorithms In Motsp (Co2 Emissions), International Journal Of Artificial Intelligence & Applications (Ijaia) Vol. 6, No. 5.

AUTHOR PROFILE

MS. J. Kanimozhi pursued B.Tech (IT) at Pondicherry Engineering College and M.Tech (CSE) at Pondicherry University, Puducherry, India. Currently working as an Assistant Professor in Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College, Puducherry, India. Her area of interest is Computer network Security, Evolutionary Algorithms and Android Application. She has 2 years of teaching experience.

Ms V. Swathilakshmi pursued B.Tech CSE and M.Tech (CSE) from Pondichery University and Currently working as an Assistant Professor in Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College, Puducherry, India. Her area of interest is DBMS, Android Application. She has 2 years of teaching experience.

Mr. B. Thiyagarajan pursued B.Tech and M.Tech (CSE) at Pondicherry University, Puduchery, India. India and working as an Assistant Professor in Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College, Puducherry, India. His area of Interest is Digital systems, Computer Networks and data structures. He has 8 years of teaching experience.

Dr. R. Subramanian is working as a Professor & Head, Dept. of Computer Science, School of Engineering & Technology, Pondicherry University, Puducherry, India. He has completed his Ph.D. from IIT, Delhi, India in 1989 and M.Sc. from IIT, Delhi, India in the year of 1984 and B.Sc. from Madurai Kamaraj University in 1982. He is having more than 20 years of total teaching experience. His specialization is in Parallel and Distributed Algorithms, Robotics, Evolutionary Algorithms.