Comparative Analysis of Evolutionary Algorithms for Multi-Objective Travelling Salesman Problem

Nosheen Qamar[1]
Department of Computer Science and Information Technology, University of Lahore,
Lahore, Pakistan

Nadeem Akhtar[2], Irfan Younas[3]
Department of Computer Science
National University of Computer and Emerging Sciences,
Lahore, Pakistan

***Abstract*—The Evolutionary Computation has grown much in last few years. Inspired by biological evolution, this field is used to solve NP-hard optimization problems to come up with best solution. TSP is most popular and complex problem used to evaluate different algorithms. In this paper, we have conducted a comparative analysis between NSGA-II, NSGA-III, SPEA-2, MOEA/D and VEGA to find out which algorithm best suited for MOTSP problems. The results reveal that the MOEA/D performed better than other three algorithms in terms of more hypervolume, lower value of generational distance (GD), inverse generational distance (IGD) and adaptive epsilon. On the other hand, MOEA-D took more time than rest of the algorithms.**

***Keywords*—Evolutionary computation; algorithms; NSGA-II; NSGA-III; MOEA-D; comparative analysis**

I. Introduction

The optimization problems with a single objective are relatively easy to solve but in case of more than one objectives the optimization become harder and these kinds of problems are very common in the existing world. It is difficult to come up with unique solution for problems having more than one objective. The two or more objectives optimization problems are called Multi-Objective Optimization Problems (MOP). Most of the MOP are of NP-hard nature and require complex optimization algorithms to solve them. Evolutionary Algorithms (inspired by biological evolutionary theory) is a relatively new field which came into existence from the last few years and has widely been discussed in the last decade [1].

The aim of Traveling Salesman Problem (TSP) is to come across the possible trip with least length for salesman who had to complete his cycle of visiting all the cities with a constraint of visiting each city exactly one time. The nature of Traveling Salesman Problem (TSP) is NP-hard [1]. When there is not just one objective i.e. the minimum distance, but also time, cost and risk etc., then it will become a Multi Objective Traveling Salesman Problem (MOTSP).

In the case of Multi Objective Traveling Salesman Problem (MOTSP), it cannot be solved using deterministic methods, especially when there are large numbers of cities to visit. Heuristic Methods are based on approximations of Pareto Solutions (PS) and Pareto Front (PF) of multi objective traveling salesman problem (MOTSP). The Evolutionary Algorithms (EA) are most promising from other heuristic methods due to their ability to give approximate solutions in a single go. In most of the cases, the target of Multi Objective Evolutionary Algorithms (MOEA) is to come up with approximate PS/PF that would be as close and as diverse as possible to actual PS/PF. The convergence (close to actual/real PF) and diverse (fully spread on the PF) are two important challenges to take care while finding the PF [2], [3].

Two most famous multi objective optimization approaches are Vector Evaluated Genetic Algorithm (VEGA) and Multi Objective Genetic Algorithm (MOGA). The VEGA converts multiple objective functions into one composite function by assigning weights to given functions. But challenging part of this approach is careful assigning of weights to each solution function. This is a difficult task for the assigner to assign some weight to any objective function without deep knowledge of that specific domain [4]. The Second approach Multi Objective Genetic Algorithm (MOGA) aims to find a set of pareto optimal solutions (PS) and then choose a subset of solutions from PS which will then be called pareto optimal front (PF). As going forward from one solution to another, it needs some sacrifices to one objective while optimizing the other. The non-dominated sorting genetic algorithm (NSGA) based on MOGA was proposed in [5]. Later on, the NSGA-II [6] was proposed by avoiding the problems associated with NSGA to deal with Multi Objective Optimization Problems. To deal with more than three objectives problems, (Many-Objective) the NSGA-II did not prove to be very effective hence a new solution was proposed called NSGA-III [7] which was an extension of NSGA-II algorithm.

The Multi Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [8] is a recently developed algorithm inspired by evolutionary algorithms suggesting optimization of multi objectives by decomposing them. The MOEA/D performs better than Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Multi Objective Genetic Local Search (MOGLS). To solve different complex Multi Objective Problems (MOPs), different extensions of Multi Objective Evolutionary Algorithm based on Decomposition (MOEA/D) have been practiced. Multiple initially developed MOEA/D and its multiple extensions are already being applied on MOTSP problem. A new extension named Multi Objective Evolutionary Algorithm derived from Decomposition with Ant Colony Optimization (MOEA/D-ACO) [9] which was proposed based on the idea that each ant will be responsible for one sub problem. The MOEA/D-ACO was compared with BicriterionAnt [10] algorithm by applying it on dual objectives traveling Salesman Problem (b-TSP) and improvement has been clearly observed.

The popularity of Traveling Salesman Problem, its NP-hard nature and it is well known and widely used problem has motivated us to use this problem to test our comparative analysis. In this study, we have applied NSGA-II, NSGA-III SPEA2, MOEA/D and VEGA. This study is a comparative analysis of the above mentioned five algorithms to find out that which algorithm proves to be the best for MOTSP problem.

This paper is structured as follows, Section II discusses the Literature Review, and Section III highlights the comparative analysis of evaluation that which algorithm works best for MOTSP. Finally, Section IV discusses the conclusion along with future work.

II. Literature Review

The Traveling Salesman Problem (TSP) is a combinational optimization problem [11] with an aim of finding shortest tour visiting all cities (from a given set) exactly at once. This could be the most popular NP-hard optimization problem and lots of studies could be made to get an optimized solution for this problem. There are different variants [12], [24] of Traveling Salesman Problem proposed including multiple-traveling salesman problem [13], [14], Multi-objective two-depot traveling salesman problem [15], probabilistic traveling salesman problem [16], Multi-objective Multiple Traveling Salesman Problem [17], Multi-objective Physical traveling [18], [19] and Multi-objective generalized Travelling Salesman Problem [20], etc.

The classic Travelling Salesman Problem (TSP) includes a number of variants, the Multi-Objective Traveling Salesman Problem (MOTSP) is the one which has been explored by a large number of researchers where multiple objectives i.e. time, cost, distance, etc. need to be optimized [21]. Due to its NP-hard nature, it is very difficult to get the optimal solution in the reasonable time. That is because multiple approximation techniques were proposed in three major categories i.e. classical heuristics, population based meta-heuristics and meta-heuristics based on single solution. The chapters of Johnson and McGeoch [22], [23] from the book of Gutin and Punnen [14] discuss the symmetric heuristics and asymmetric heuristics versions of Travelling Salesman Problem respectively. The [25] discussed the survey of local search (meta-heuristics for TSP), while the [26] describes genetic algorithms (GA) and [27] covers mimetic algorithms used for TSP.

In 1999 Preux and Talbi [28] describe the search algorithm's behavior with intent that the structure of the search space may improve the performance of the algorithm. In their study they reviewed the knowledge related to search spaces of combinational optimization problems and discussed the hybridization in detail. They also presented different techniques of hybridization based on their knowledge, on search space structure and the performance of an algorithm.

Borges and Hansen in 2000 [29] discussed the Multi-Objective TSP. The authors discussed the "global convexity" in Multi-Objective Combinational Optimization Problems generally and Multi-Objective TSP specifically. The paper focused on local optima landscape by using classical two-opt neighbors (without breaking the tour it will replace two edges with single possible solution, and two edges would get removed) with help of famous scalar functions i.e. Tchebycheff or weighted sum of multiple objectives.

The [30] in 2004 discussed the solution for TSP based on hybrid evolutionary algorithm, authors proposed an algorithm with strategy of distance preserving crossover (DPX) integrating memory as ant pheromone during the city selection process aiming to compliment the successful results of genetic algorithm (GA). The probability of distance and previous success for city selection along with combination of genetic algorithm (GA) and DPXwould be considered as additional information and would help in finding optimized quality solutions for TSP with reduced computational complexity.

The Pareto Converging Genetic Algorithm (PCGA) was proposed by Kumar and Singh [31] in 2007, doing hybridization of Pareto Rank Genetic Algorithm with Local Search. The evaluation criterion for each solution was its rank and total numbers of dominating individuals. The two individuals were selected based on raffle wheel and the distance preserving crossover (DPX) operation was performed to generate offspring. The produced offspring were again merged with population based on its rank. After doing mutation operation the converging criterion was defined depending on "rank-histograms" and within the population the rank of individuals is one plus the total number of individuals dominating it aiming to assign all non-dominated individuals to one. The union of new population with older one was ranked. As close as possible the pareto will be converged to rank histogram equal to a single value which is not equal to zero entry of 1/2for rank equal to 1 correspond to that no solution is better than the previous (older) population originated in evolving the new population.

Changdar et al. [32] in the year 2014 considered two objectives, cost as first and time as second to solve the multi-objective Static TSP in their suggested multi-objective genetic algorithm (MOGA). The nature of proposed algorithm was not clear. In the same year, Li [33] managed to propose an algorithm for multi-objective dynamic TSP with two and three objectives with a parallel search system. Moreover, Florios and Mavrotas's [34] proposed solution for Multi Objective Travelling Salesman Problem (MOTSP) and Set Covering Problem (SCP) which was based on Pareto front for dual objectives functions with help of AUGMECON2 method. Another contribution by Bouzoubia et al. [35] in the same year, made a difference by using couple of variations derived by Multi Objective Chemical Reaction Optimization (MOCRO) to get good solutions for multi-objective TSP by the use of non-dominated sorting technique which was already used in NSGA-II algorithm. He [36] also contributed to solve multi-objective TSP using membrane algorithms. Labadie et al. [37] were also one of those who put their part to get optimise solution for multi-objective TSP using two objectives with profits (BOMTSPP) in same year.

Bolano et al. [38] in 2015 proposed NSGA-II algorithm to solve multi-objective TSP using a NSGA II algorithm. Wang et al. [39] suggested hybrid NSGA-II algorithm to achieve optimal good solution for multi-objective TSP initially and then he proposed a new hybrid algorithm [40] which combined an uncertain approach with Artificial Bee Colony (ABC) algorithm. Ariyasingha and Fernando [41] conducted a review of Colony Optimization Algorithms (COA) for MOTSP for bi-objective and tetra-objective functions.

The 2016 researches on multi objective travelling salesman problem contain a research by Cornu et al. [42] proposed a novel multi objective decomposition algorithm called perturbed Decomposition Algorithm (PDA). The newly proposed PDA algorithm suggests combination of decomposition methods, data perturbation and local search. Authors claimed that PDA performs better than existing algorithms available on multi objective travelling salesman problem (MOTSP).

Author in [43] suggest a new solution for Multi Objective Travelling Salesman Problem (MOTSP) with imprecise Multi Objective Genetic Algorithm (iMOGA) with fuzzy age selection. The proposed algorithm also used adaptive crossover and mutation which depends on generation. The fuzzy age was replaced by fuzzy extended age.

III. Algorithms Selected for Experiment

A. NSGA-II

The NSGA-II is a faster and better algorithm than the MOEA algorithms in terms of close coverage and correct pareto optimal front. The NSGA-II works as the initial population has been defined with some set of solutions, then λ solutions are generated with help of stochastic variation operators. The λ generated solutions evaluated and then ranked on pareto from as best solutions on first non-dominated front and so on. The main reason behind the selection of this algorithm was its low complexity, good coverage and better diversity.

B. NSGA-III

The many objective optimization problems are very challenging to optimize and are difficult to handle. The NSGA-III is the algorithm used to handle many objective problems. The reason behind selection of this algorithm was that during our experiments, we had up to 5 objectives and in that scenario this was an effective algorithm to measure results.

C. SPEA-2

The SPEA-2 is an improved version of SPEA algorithm and it starts its working with initial population and an empty archive. Then the fitness values of solutions are evaluated and then the solutions with best fitness are added to the archive (with a specific number) with non-dominated solutions and if there is still space, the good dominated solutions can also be added. After fulfilling the termination criteria, binary tournament is performed and the next generation is created after recombination and mutation operation and this process repeat with some specific set of generations. The reason behind selection of this algorithm is its focus on dominance count and rank and that good coverage can be achieved with the help of this algorithm.

D. MOEA/D

The main idea behind the MOEA/D algorithm is the decomposition of multi-objective optimization problem into a number of small scalar optimization problems and then optimizes those scalar problems simultaneously. Every sub problem was optimized with help of its multiple neighbors providing information. The motivation behind using this algorithm was its lower computational complexity due to breaking a larger and complex problem into multiple scalar problems and then their optimization based on their neighbors.

E. VEGA

The VEGA (Vector Evaluated Genetic Algorithm) is pioneer algorithm to find non-dominated solution for multi objective optimization problems. It is an extension of single objective genetic algorithm to optimize the multi objective problems. We used this algorithm due to its efficiency and higher speed.

IV. Experiments and Results

As mentioned above different experiments were conducted on TSP problem using five different (NSGA-II, NSGA-III, SEPA-2, MOEA/D and VEGA) algorithms and this section discusses the experimental setup and has the results of those experiments.

A. Experiment Setup

In the multi-objective (K-objective) Traveling Salesman Problem, K objective functions need to be defined. These objectives can be cost of the tour, travel time or any other factor which need to be optimized. Table I demonstrates the experimental setup. The Cent OS, 8cores platform with Java8 and MOEA framework were used. The population size was decided as 50 and 100 with 50, 100, 1000 and 10000 generations. The experiment was repeated for 10,100 and 1000 iterations. The results were compared based on Hypervolume, Generational-Distance (GD), Inverted-Generational-Distance (IGD), Additive-ε and Time taken to conduct the experiment. Let's assume all contributing factors are on different graphs with same number of vertices but have different values for edges. In order to simulate multiple objectives for the TSP, different TSPLIB problem situations which have the same number of nodes were used. Each situation was considered to be a single objective which requires to be minimized. Multiple experiments were conducted for 5 objectives (5 cyclic tours for 5 libraries) of TSP problem by using TSPLIB standard dataset library [44]. System was implemented in Java language and the use MOEA framework [45] for the conditions of experiments.

B. Results and Analysis

Results have been compared by using four indicators as Hypervolume, Generational-Distance (GD), Inverted-Generational-Distance (IGD) and Adaptive-ε.

TABLE I. EXPERIMENTAL SETUP

Platform	Cent OS, 8 cores 8 GB Memory (6 GB user memory)
Framework	Java 8 MOEA framework
TSPLIB Libraries	kroA100, kroB100, kroC100, kroD100, kroE100
Algorithms	NSGA-II, NSGA-III, SPEA-2, MOEA/D, VEGA
Population size	50, 100
# Generations	50, 100, 1000, 10000
# Iterations	10, 100, 1000
Result Indicators	Hypervolume, Generational-Distance (GD), Inverted-Generational-Distance (IGD), Additive-ε and Time

1) Hypervolume: In Table II and Fig. 1 below, all the experimental results have been shown for the Hypervolume indicator. The results in Table II and Fig. 1 clearly show that the MOEA/D performed well for the population size 50 and 100, generations 10000 and the iterations 10,100 and 1000. The highest gain hypervolume produced by MOEA/D is between 0.172291 to 0.206567. Based on the given data we can say that the MOEA/D has performed better than the other algorithms for the TSP problem.

TABLE II. HYPERVOLUME COMPARISON FOR ALL ITERATIONS

Population-Size	Generations	Iterations	NSGA-II	NSGA-III	SPEA-2	MOEA/D	VEGA
50	50	10	0.002788	0.002561	0.002653	0.003207	0.003035
50	50	100	0.002701	0.002741	0.002682	0.002830	0.002638
50	50	1000	0.002712	0.002691	0.002744	0.002680	0.002730
50	100	10	0.003655	0.004549	0.004026	0.003426	0.002398
50	100	100	0.003721	0.003631	0.004269	0.003277	0.002720
50	100	1000	0.003792	0.003777	0.004420	0.003154	0.002697
50	1000	10	0.005910	0.007092	0.007180	0.012533	0.002285
50	1000	100	0.006470	0.007180	0.007078	0.010248	0.002643
50	1000	1000	0.006407	0.007465	0.007502	0.011357	0.002739
50	10000	10	0.008954	0.040115	0.013966	**0.206567**	0.016630
50	10000	100	0.009051	0.022306	0.013521	**0.172291**	0.015493
50	10000	1000	0.009039	0.027165	0.013355	**0.184121**	0.017165
100	50	10	0.003925	0.003964	0.003511	0.004261	0.003702
100	50	100	0.003828	0.003838	0.003736	0.004014	0.003931
100	50	1000	0.00379	0.003747	0.003797	0.003777	0.003827
100	100	10	0.003868	0.004339	0.003717	0.003901	0.004148
100	100	100	0.003757	0.003687	0.003882	0.003732	0.003876
100	100	1000	0.003749	0.003780	0.003763	0.003830	0.003752
100	1000	10	0.008978	0.008367	0.008371	0.009061	0.003326
100	1000	100	0.008202	0.008122	0.008506	0.008382	0.003980
100	1000	1000	0.008323	0.008314	0.008699	0.008442	0.003806
100	10000	10	0.011123	0.012378	0.013859	**0.179984**	0.004620
100	10000	100	0.011630	0.012774	0.015254	**0.136949**	0.003826
100	10000	1000	0.011542	0.012093	0.014958	**0.136986**	0.004074

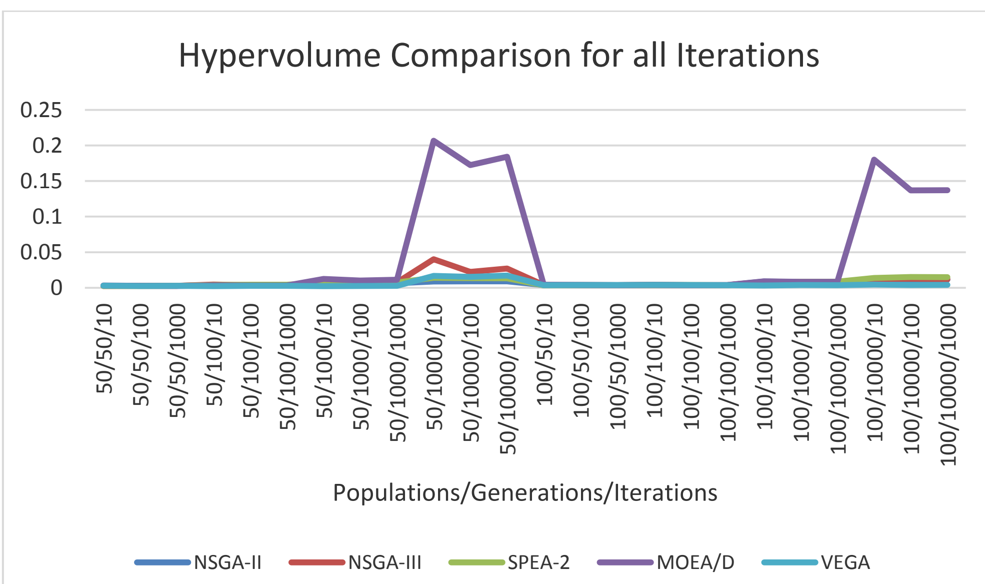

Fig. 1. Hypervolume comparison for all iterations.

2) Generational Distance (GD):

Table III contains the comparison data and based on that, Table II demonstrates that MOEA/D performed better than the rest of the three algorithms in terms of generational distance. The NSGA-III was less better with higher value of generational distance. The figure shows that while comparing base of GD, the results shows the MOEA/D performed least for the population size 50 and 100, the generations 10000 and the iterations 10,100 and 1000. NSGA-II, SPEA2 and VEGA performed almost equal for TSP problem with multiple objectives. Fig. 2 is a graphical representation of Table III.

TABLE III. GENERATIONAL DISTANCE (GD) COMPARISON FOR ALL ITERATIONS

Population-Size	**Generations**	**Iterations**	**NSGA-II**	**NSGA-III**	**SPEA-2**	**MOEA/D**	**VEGA**
50	50	10	0.139000	0.146823	0.140682	0.148243	0.163895
50	50	100	0.140185	0.142021	0.140469	0.143294	0.185806
50	50	1000	0.140536	0.140822	0.141191	0.140571	0.237233
50	100	10	0.115427	0.111052	0.102978	0.159342	0.170148
50	100	100	0.110141	0.108757	0.102570	0.148458	0.191301
50	100	1000	0.109818	0.109527	0.102568	0.147099	0.240964
50	1000	10	0.097494	0.094884	0.095641	0.096621	0.157229
50	1000	100	0.096075	0.094440	0.095249	0.101171	0.199204
50	1000	1000	0.096173	0.094236	0.095051	0.100675	0.234012
50	10000	10	0.090736	0.064280	0.086743	0.024569	0.150873
50	10000	100	0.090904	0.074443	0.088067	0.029978	0.169089
50	10000	1000	0.091203	0.070661	0.087542	0.027356	0.168105
100	50	10	0.115540	0.114182	0.107821	0.115765	0.117951
100	50	100	0.109221	0.108482	0.110741	0.110618	0.146849
100	50	1000	0.110426	0.109321	0.109581	0.109903	0.157706
100	100	10	0.108834	0.110043	0.107679	0.104872	0.121678
100	100	100	0.109281	0.108790	0.110245	0.110162	0.146884
100	100	1000	0.109025	0.109573	0.109145	0.109929	0.159126
100	1000	10	0.067435	0.067517	0.067733	0.089163	0.133054
100	1000	100	0.067604	0.067645	0.067243	0.088394	0.137151
100	1000	1000	0.067518	0.067437	0.067328	0.08828	0.279085
100	10000	10	0.064252	0.062686	0.062755	0.019408	0.127604
100	10000	100	0.063733	0.063687	0.062082	0.026588	0.165126
100	10000	1000	0.064086	0.063720	0.062128	0.027225	0.156897

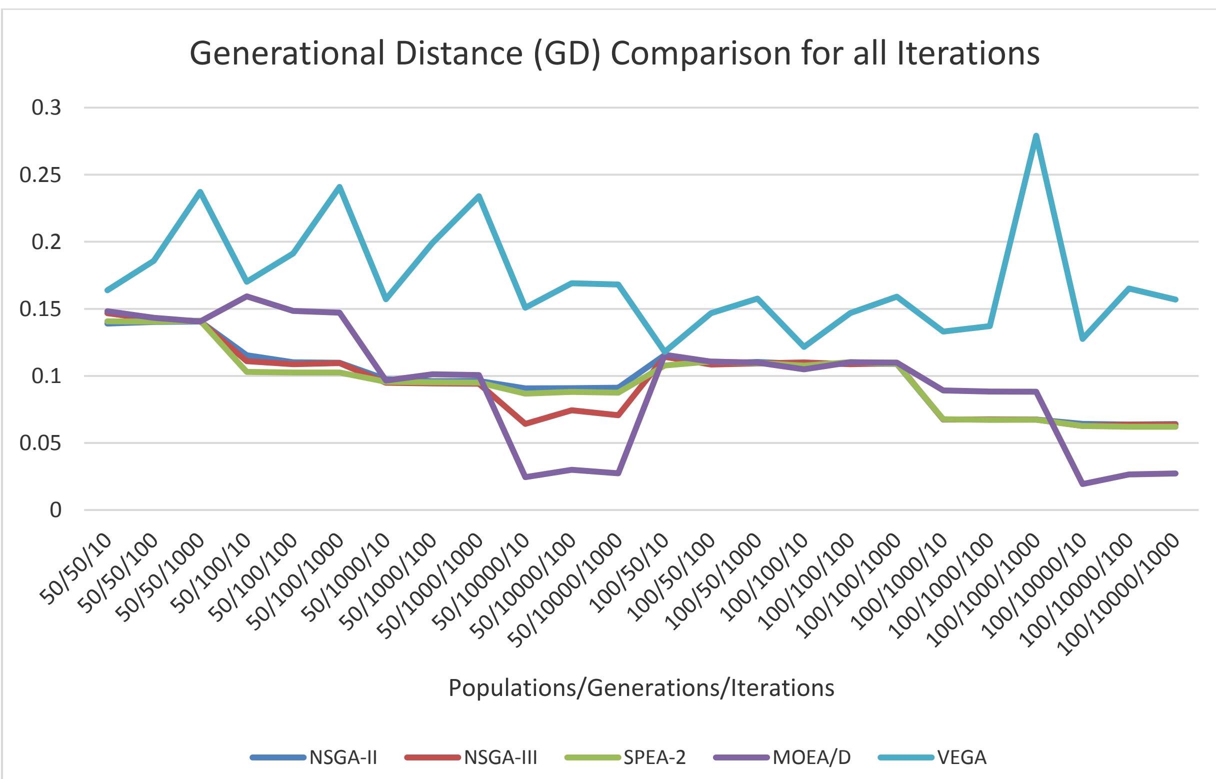

Fig. 2. Generational Distance (GD) comparison for all iterations.

3) Inverted-Generational Distance (IGD):

Table IV and so Fig. 3 (constructed from the data available in Table IV) represents that the MOEA/D performed better from rest of the four algorithms, specifically at the noticeable point of 50 and 100 population size, 10000 generations and 10,100, 1000 iterations. The rest of the three algorithms were with almost equal results.

TABLE IV. INVERTED-GENERATIONAL DISTANCE (IGD) COMPARISON FOR ALL ITERATIONS

Population-Size	Generations	Iterations	NSGA-II	NSGA-III	SPEA-2	MOEA/D	VEGA
50	50	10	0.876195	0.883478	0.875846	0.851304	0.841886
50	50	100	0.870580	0.871930	0.876809	0.867707	0.871765
50	50	1000	0.871787	0.872499	0.869325	0.873542	0.871335
50	100	10	0.840716	0.817917	0.838676	0.83858	0.890045
50	100	100	0.841584	0.850365	0.830244	0.849765	0.873776
50	100	1000	0.843890	0.844259	0.828859	0.854852	0.872269
50	1000	10	0.787684	0.776834	0.768491	0.698727	0.896334
50	1000	100	0.781947	0.774108	0.771899	0.730504	0.878822
50	1000	1000	0.782636	0.769031	0.766264	0.712183	0.873151
50	10000	10	0.730746	0.556786	0.690203	0.261163	0.651132
50	10000	100	0.740443	0.620768	0.685077	0.294764	0.656258
50	10000	1000	0.736205	0.592826	0.684586	0.283587	0.643210
100	50	10	0.838765	0.836466	0.849648	0.820317	0.845035
100	50	100	0.841628	0.844725	0.843842	0.838111	0.829266
100	50	1000	0.843522	0.845131	0.843501	0.845035	0.841485
100	100	10	0.847687	0.834806	0.853649	0.833991	0.837925
100	100	100	0.845193	0.845431	0.841668	0.840722	0.835674
100	100	1000	0.845553	0.845276	0.844949	0.842207	0.844789
100	1000	10	0.759297	0.770600	0.770172	0.745906	0.857111
100	1000	100	0.773435	0.771152	0.768211	0.755973	0.836586
100	1000	1000	0.767053	0.766924	0.762013	0.757285	0.844768
100	10000	10	0.734211	0.718311	0.702650	0.258231	0.810780
100	10000	100	0.721530	0.713585	0.685823	0.316330	0.841607
100	10000	1000	0.722457	0.718744	0.690895	0.314585	0.834649

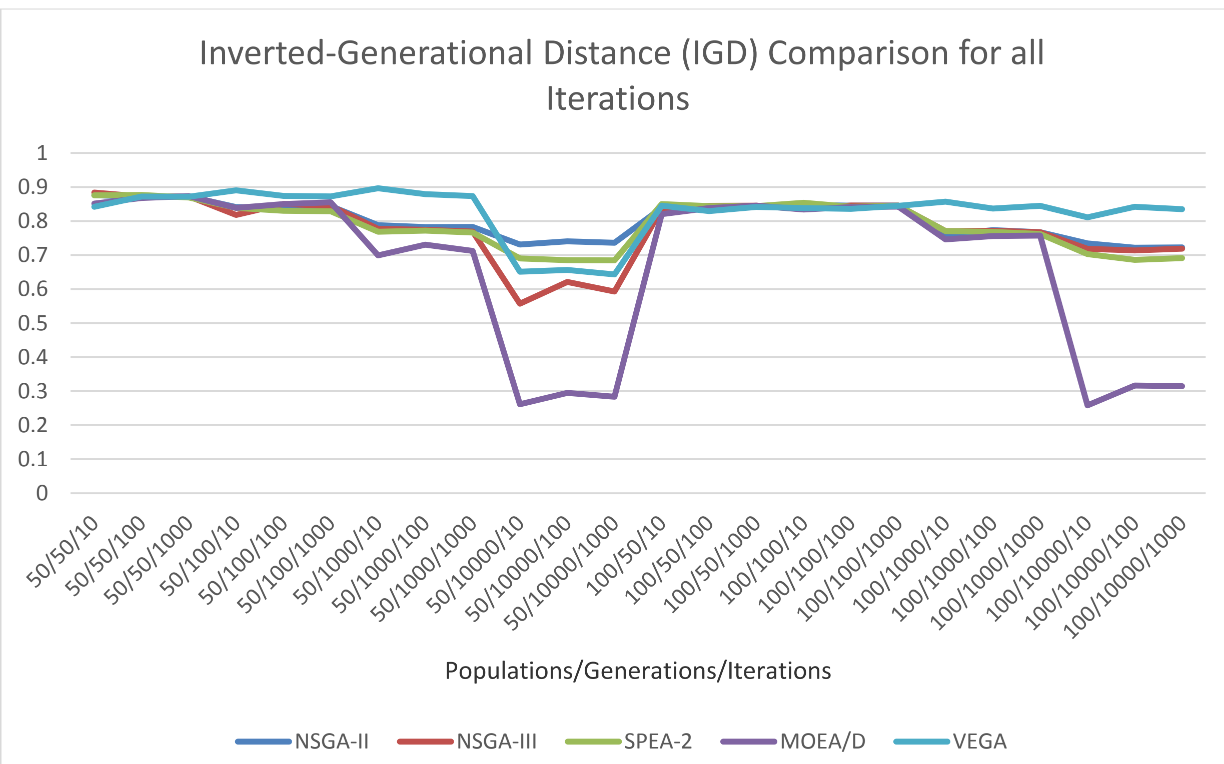

Fig. 3. Inverted-Generational Distance (IGD) comparison for all iterations.

4) Adaptive-ε:

The data in Table V was used to make the graph represented in Fig. 4 which illustrates that the MOEA/D is better when analysing the results based on Adaptive epsilon. As we know that lower epsilon value is better to achieve and when comparing with other four algorithms the MOEA/D is comaratively better.

TABLE V. ADAPTIVE EPSILON COMPARISON FOR ALL ITERATIONS

Population-Size	Generations	Iterations	NSGA-II	NSGA-III	SPEA-2	MOEA/D	VEGA
50	50	10	0.648598	0.652171	0.646459	0.654711	0.639249
50	50	100	0.653534	0.651258	0.653652	0.653722	0.649986
50	50	1000	0.651104	0.651957	0.652818	0.652084	0.652297
50	100	10	0.621849	0.614253	0.603638	0.645525	0.667193
50	100	100	0.623724	0.625773	0.609900	0.645131	0.654024
50	100	1000	0.622009	0.622433	0.608512	0.650904	0.652784
50	1000	10	0.589174	0.574628	0.568762	0.555046	0.653651
50	1000	100	0.581862	0.571760	0.569654	0.560187	0.652413
50	1000	1000	0.581643	0.571665	0.566568	0.557785	0.652627
50	10000	10	0.566406	0.503615	0.542692	0.308017	0.575066
50	10000	100	0.577152	0.529691	0.540753	0.327255	0.584646
50	10000	1000	0.575764	0.514632	0.539941	0.322224	0.574605
100	50	10	0.619609	0.619759	0.625357	0.617392	0.619747
100	50	100	0.625448	0.618305	0.621055	0.624260	0.623730
100	50	1000	0.621742	0.622209	0.621561	0.622711	0.621343
100	100	10	0.618887	0.624665	0.627339	0.619170	0.622793
100	100	100	0.621013	0.622225	0.621658	0.624664	0.617485
100	100	1000	0.621829	0.622342	0.621681	0.622295	0.622017
100	1000	10	0.550720	0.551214	0.550506	0.561869	0.631512
100	1000	100	0.553415	0.554696	0.550566	0.570871	0.622054
100	1000	1000	0.552924	0.553912	0.549637	0.568403	0.622550
100	10000	10	0.547026	0.539875	0.517950	0.314718	0.611036
100	10000	100	0.540999	0.532733	0.514461	0.355844	0.619680
100	10000	1000	0.543220	0.535499	0.513611	0.358324	0.619954

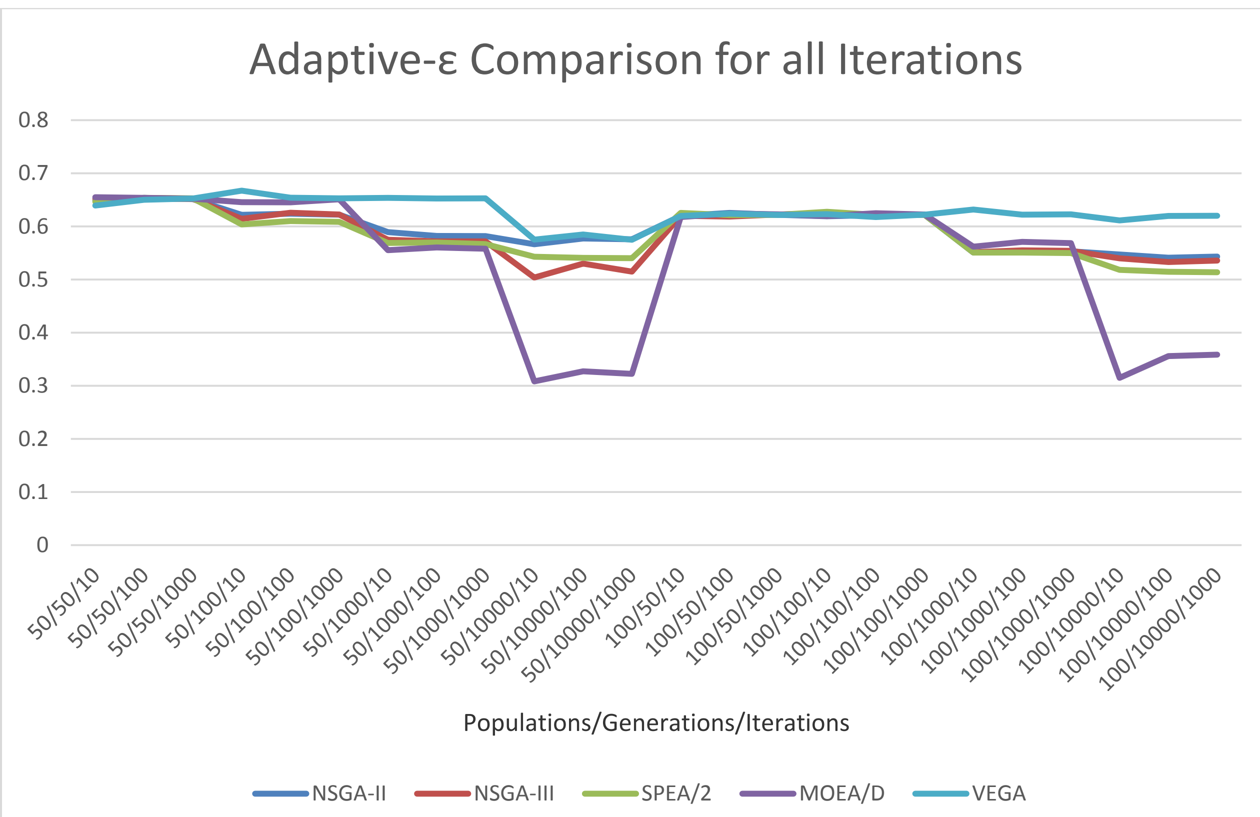

Fig. 4. Adaptive Epsilon comparison for all iteration.

5) Time Comparison:

Fig. 5 give us representation of the time comparison of our applied algorithms. Fig. 6 demonstrates the comparison of time for all the values including different populations, generations and iterations. The results clearly show that MOEA-D took more time than all other algorithms. The SPEA and NSGA-III took almost equal time and NSGA-II and VEGA took lowest time for the combination of 50 population size, 10000 generations and 1000 iterations. Almost same is the case for 100 population, with same number of generations and iterations.

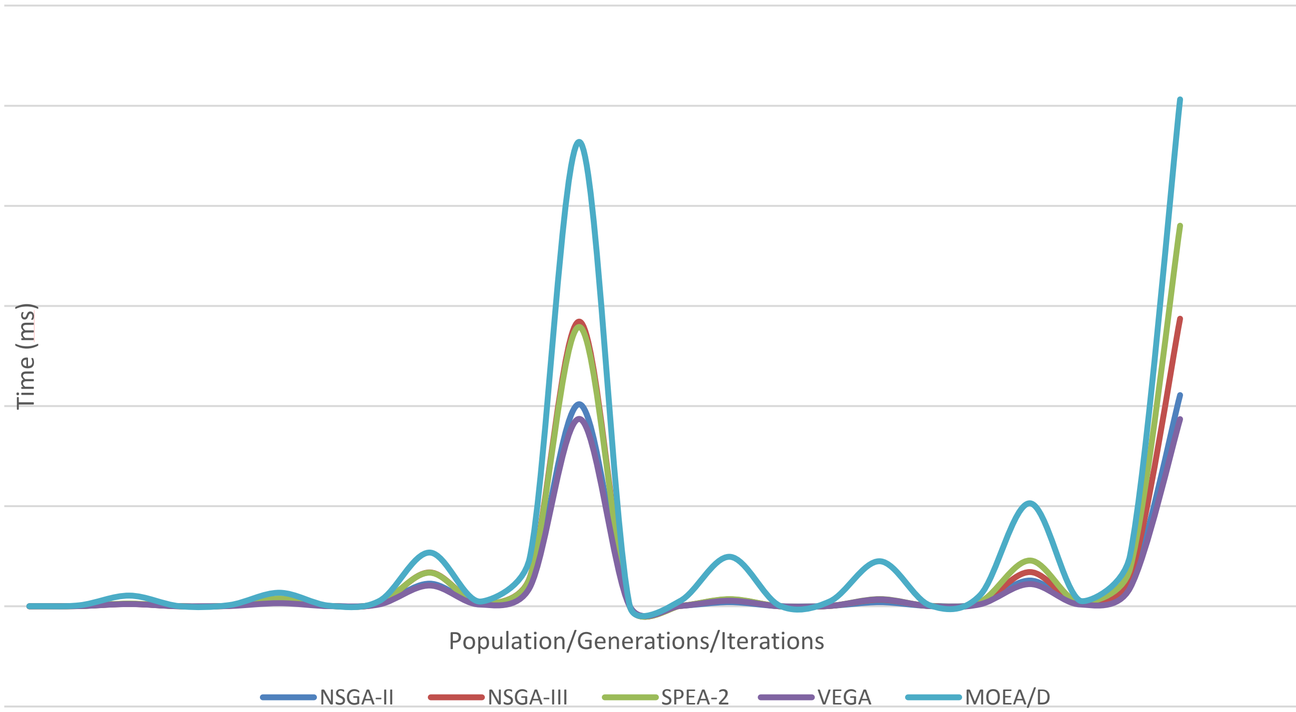

Fig. 5. Time comparison of all the algorithms.

V. Conclusions and Future Work

The TSP is a widely evaluated single objective problem. This problem can be expanded by converting it into multi objective or many objectives by considering different objectives like cost, time, speed etc. In our study we have applied five popular evolutionary algorithms (NSGA-II, NSGA-III, SPEA2, MOEA/D and VEGA) to solve the TSP problem and came up with the results that MOEA/D performed better on different (hypervolume, generational distance, inverted generational distance and adaptive epsilon) indicators more specifically with the population size 50 and 100 with 10000 no. of generations and 10, 100 &1000 iterations. The results further show that although MOEA-D performed better than other algorithms but it took more time in comparison to the rest of algorithms. With respect to time the NSGA-II and VEGA took the lowest time.

The future work can cover the implementation of the above algorithms on some other well-known problem like the problem of knapsack or different combination of algorithms can be used (other than ours) to find out what algorithm works best for kanpsack problem. It would also be interesting to know what would be the results when using different number of populations (greater than 100 as our maximum population size was 100) and different number of iterations and generations.

References

[1] R. Baraglia, J. Hidalgo, and R. Perego, “A hybrid heuristic for the traveling salesman problem,” IEEE Transactions on Evolutionary Computation, vol. 5, no. 6, pp. 613 –622, 2001.

[2] K. C. Tan, E. F. Khor, and T. H. Lee, Multiobjective evolutionary algorithms and applications. Springer, 2005.

[3] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont,Evolutionary algorithms for solving multi-objective problem. Kluwer, 2002.

[4] J.D. Schaffer, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," In Proc. 1st International Conference on Genetic Algorithms, pp.93-100, 1985.

[5] N. Srinivas and K. Deb, “Multiobjective function optimization usingnondominated sorting genetic algorithms,” *Evol. Comput.*, vol. 2, no.3, pp. 221–248, Fall 1995.

[6] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, Vol. 6, APRIL 2002.

[7] K. Deb, H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," IEEE Transactions of Evolutionary Computation, Vol. 18, Aug 2014.

[8] Q. Zhang and H. Li, “Moea/d: A multiobjective evolutionary algorithm based on decomposition.” IEEE Trans. Evolutionary Computation, vol. 11, no. 6, pp. 712–731, 2007.

[9] L. Ke, Q. Zhang, and R. Battiti, “Moea/d-aco: A multiobjective evolutionary algorithm using decomposition and ant colony.” IEEE Trans.Cybern., vol. 43, no. 6, pp. 1845–1859, 2013.

[10] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms." in EMO, ser. Lec. Notes in Comp. Science, E. Zitzler, K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, Eds., vol.1993. Springer, 2001, pp. 359–372.

[11] G. Gutin, ,A. Punnen, "The traveling salesman problem and its variations,"Kluwer Academic Publishers, Dordrecht, 2002.

[12] G. Reinelt, "TSPLIB A traveling salesman problem library," ORSA J.Comput., 1991.

[13] H. Zhou, M.Song, "An improvement of partheno-genetic algorithm to solve multiple travelling salesmen problem" in Proc. 15th International Conference on Computer and Information Science (ICIS), IEEE/ACIS, 2016. DOI. 10.1109/ICIS.2016.7550780

[14] J. Li, M. Zhou, Q. Sun, X. Dai, X. Yu, "Colored Traveling Salesman Problem" in IEEE Transactions on Cybernetics, vol. 45, pp. 2390 – 2401, Nov. 2015. DOI. 10.1109/TCYB.2014.2371918

[15] H. Zhong, N. Zhuang, Z. Wu, W.Cai, "Multi-objective two-depot traveling salesman problem with uncertain fleet size for hazardous materials" in Proc. 8th International Conference on Supply Chain Management and Information Systems (SCMIS), 2011.

[16] A. Henchiri, M. Bellalouna, W. Khasnaji, "Probabilistic traveling salesman problem: a survey" in Proc. Federated Conference on Computer Science and Information Systems, pp. 55-60, 2014. DOI. 10.15439/2014F381

[17] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A Hybrid Estimation of Distribution Algorithm with Decomposition for Solving the Multiobjective Multiple Traveling Salesman Problem" IEEE Transactions on systems, man and cybernetics, vol. 42, pp. 682-691, Oct. 2012. DOI. 10.1109/TSMCC.2012.2188285

[18] D. Perez, E. Powley, D.l Whitehouse, S. Samothrakis, S. Lucas, P.I. Cowling, "The 2013 Multi-Objective Physical Travelling Salesman Problem Competition," in Proc. IEEE Congress on Evolutionary Computation, 2014,pp. 2314-2321.

[19] E.J. Powley, D. Whitehouse, P.I. Cowling, "Monte Carlo Tree Search with macro-actions and heuristic route planning for the Multiobjective Physical Travelling Salesman Problem," in Proc. IEEE Conference on Computational Intelligence in Games (CIG), 2013. DOI. 10.1109/CIG.2013.6633658

[20] J.V.Pinxten, M. Geilen, T. Basten, U. Waqas, L. Somers, "Online Heuristic for the Multi-objective Generalized Traveling Salesman Problem" Design, Automation & Test in Europe Conference & Exhibition, 2016.

[21] F. Samanlioglu, W.G. Ferrell,M.E. Kurz, "A memetic random-key genetic algorithm for a symmetric multi-objective traveling salesman problem,"Computers and Industrial Engineering, vol. 55, pp. 439-449, 2008.

[22] D.S. Johnson, L.A. McGeoch, "Experimental analysis of heuristics for the ATSP," in The Traveling Salesman Problem and its Variations, G. Gutin, A. Punnen, eds. Kluwer, Dordrecht, 2002.

[23] D.S. Johnson, L.A. McGeoch, "Experimental analysis of heuristics for the STSP," in The Traveling Salesman Problem and its Variations, G. Gutin, A. Punnen, Kluwer, Dordrecht, 2002.

[24] G. Gutin, A. Punnen, "The Traveling Salesman Problem and its Variations," Kluwer, Dordrecht. 2002.

[25] D.S. Johnson, L.A. McGeoch, "The Traveling Salesman Problem: A Case Study in Local Optimization," in Local Search in Combinatorial Optimization, E.H.L. Aarts, J.K. Lenstra, Eds. John Wiley and Sons Ltd., Chichester, 1997, pp. 215–310.

[26] P. Larranaga, C.M.H. Kuijpers, R.H. Murga, I. Inza, S. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," Artificial Intelligence Review, vol. 13, pp. 129–170, 1999.

[27] P. Merz, B. Freisleben, " Memetic algorithms for the traveling salesman problem. Complex Systems," vol. 13, pp. 297–345, 2001.

[28] P. Preux, E.G. Talbi, "Towards hybrid evolutionary algorithms," International Transactions in Operational Research, Wiley, vol. 6, Nov. 1999, pp. 557-570

[29] P.C. Borges, M.P. Hansen, "A study of global convexity for a multiple objective travelling salesman problem," in Essays and surveys in metaheuristics, C.C. Ribeiro, P. Hansen, Eds. Kluwer, Dordrecht, pp. 129–150, 2000.

[30] C.M. White, G.G. Yen, "A hybrid evolutionary algorithm for traveling salesman problem" in Proc. Congress on Evolutionary Computation, IEEE, 2004. DOI. 10.1109/CEC.2004.1331070

[31] R. Kumar,P.K. Singh, " Pareto evolutionary algorithm hybridized with local search for biobjective TSP" in Hybrid Evolutionary Algorithms, C. Grosan,A. Abraham, H. Ishibuchi, Eds. Springer, Heidelberg, 2007.Ff

[32] C. Changdar,G.S. Mahapatra, R.K. Pal, " An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness," Swarm and Evolutionary Computation, vol. 15, pp. 27-37,2014.

[33] W. Li, "A parallel search system for dynamic multi-objective traveling salesman problem," Journal of Mathematics and System Science, vol. 4, pp. 295-314, 2014.

[34] K. Florios, G. Mavrotas, "Generation of the exact pareto set in multi-objective traveling salesman and set covering problems," Applied Mathematics and Computation, vol. 237, pp. 1-19, 2014.

[35] S. Bouzoubia, A. Layeb, S. Chikhi, "A multi-objective chemical reaction optimization algorithm for multi-objective travelling salesman problem," International Journal of Innovative Computing and Applications, vol. 6, pp. 87-101, 2014.

[36] J. He, "Solving the multiobjective multiple traveling salesmen problem using membrane algorithm, Bio-Inspired Computing, Theories and Applications," Communications in Computer and Information Science, vol. 472, pp. 171-175, 2014.

[37] N. Labadie, J. Melechovsky, C. Prins, "A parallel search system for dynamic multi-objective traveling salesman problem, Applications of Multi-Criteria and Game Theory Approaches," Springer-Verlag, London, 2014.

[38] R.I. Bolanos, M.G. Echeverry, J.W. Escobar, " A multiobjective nondominated sorting genetic algorithm (NSGA-II) for the multiple traveling salesman problem," Decision Science Letters, vol. 4, pp. 559-568, 2015.

[39] P. Wang, C. Sanin, E. Szczerbicki, "Evolutionary algorithm and decisional DNA for multiple traveling salesman problem," Neurocomputing, vol. 150, pp. 50-57, 2015.

[40] Z. Wang, J. Guo, M. Zheng, Y., Wang, "Uncertain multiobjective traveling salesman problem," European Journal of Operational Research, vol. 241, pp. 478-489, 2015.

[41] I.D. Ariyasingha, T.G.I. Fernando, "Performance analysis of the multiobjective ant colony optimization algorithms for the traveling salesman problem," Swarm and Evolutionary Computation, in Press, 2015.

[42] M. Cornu, T. Cazenave, D. Vanderpooten, "Perturbed Decomposition Algorithm applied to the multi-objective Traveling Salesman Problem," Computers & Operations Research elsevier, Apr 2016.

[43] S. Maity, A. Roy, M. Maitic, "An imprecise Multi-Objective Genetic Algorithm for uncertain Constrained Multi-Objective Solid Travelling Salesman Problem," Expert Systems with Applications elsevier, vol. 46, pp. 196-223, Mar 2016.

[44] S http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.

[45] http://moeaframework.org/.