

Non-dominated Sorting Tournament Genetic Algorithm for Multi-Objective Travelling Salesman Problem

Paweł B. Myszkowski

Wrocław University of Science and Technology
Faculty of Computer Science and Management
ul. Ignacego Łukasiewicza 5, 50-371 Wrocław, Poland
pawel.myszkowski@pwr.edu.pl

Maciej Laszczyk

Wrocław University of Science and Technology
Faculty of Computer Science and Management
ul. Ignacego Łukasiewicza 5, 50-371 Wrocław, Poland
maciej.laszczyk@pwr.edu.pl

Kamil Dziadek

Wrocław University of Science and Technology
Faculty of Computer Science and Management
ul. Ignacego Łukasiewicza 5, 50-371 Wrocław, Poland
220901@student.pwr.edu.pl

Abstract—A Travelling Salesman Problem (TSP) is an NP-hard combinatorial problem that is very important for many real-world applications. In this paper, it is shown, that proposed approach solves multi-objective TSP (mTSP) more effectively than other investigated methods, i.e. Non-dominated Sorting Genetic Algorithm II (NSGA-II). The proposed methods use rank and crowding distance (well-known from NSGA-II), combining those mechanisms in a novel, unique way: competing and co-evolving in the evolution process. The proposed modifications are investigated and verified by the benchmark mTSP instances, and results are compared to other methods.

I. INTRODUCTION

A TRAVELLING Salesman Problem (TSP) is an NP-hard combinatorial optimization problem. The goal is to find a Hamiltonian cycle, that minimizes the sum of edge weights in a complete weighted graph [1]. Importance of TSP is accentuated by the fact, that it is a part of NP-complete class of problems [2].

A multi-objective Travelling Salesman Problem (mTSP) is an extension of TSP, where more than one objective is considered. It can be cost or time of the travel, the length of the route, etc. All objectives are optimized simultaneously [3]. An mTSP with two objectives is considered in this paper.

In multi-objective optimization, there is no prioritization of the objectives. Hence, to compare different solutions a dominance relation is used. A solution dominates another, if it has the value of at least one objective better, and value of no objectives worse than that solution. The goal of multi-objective optimization is to find all non-dominated solutions, a Pareto Front (PF). In practice, it is often not known whether found solutions comprise a true PF. Hence, the result of each method is called a PF approximation.

A Non-dominated Sorting Genetic Algorithm II (NSGA-II) [4], a classical multi-objective approach, uses two distinct mechanisms in its selection. The first one is the *rank*

comparison, which is based on the dominance relation, and aims to improve the convergence of the results. The second one is the *crowding distance*, which aims to increase the diversity of the results. However, a recent Non-dominated Sorting Tournament Genetic Algorithm (NTGA) [5] does not utilize the crowding distance at all. The authors show increased effectiveness of NTGA. This paper verifies the effectiveness of both *rank* and *crowding distance*. Two methods are presented that combine those mechanisms in a novel, unique way. One that uses them sequentially and forces competition between them. The other that utilizes two populations, where the mechanisms cooperate.

A set of experiments is designed to verify the quality of PF approximations generated by all methods. The Multi-Objective Evolutionary Algorithm integrating NSGA-II, SPEA2, and MOEA/D (MOEA/NSM) [12] is currently the best-known method for mTSP. Hence, it is used to compare the results. The results are evaluated by measuring convergence and diversity of the PF approximation and efficiency of the method. The set of Quality Measures (QMs) proposed in [6] is used. Moreover, visualizations of selected results are provided and a thorough theoretical analysis is presented.

The rest of the article is structured as follows. Section II contains the overview of existing work related to mTSP and multi-objective optimization. Section III provides a formal definition of the problem. All of the proposed approaches are described in section IV. Experiments and their results are presented in section V. The paper is concluded and additional remarks are given in section VI.

II. RELATED WORK

A TSP is one of the most commonly researched problems. Many modifications to its original definition have been proposed. A TSP with asymmetric distances between the cities

[7], with multiple travelling salesmen [8], with stochastic travel times [9] or a vehicle routing problem [10].

Due to the NP-hard nature of mTSP, researchers often tackle it with metaheuristics. Genetic Algorithms are commonly used (e.g. [11], [12]). Authors of [14] and [15] have used an ant colony optimization methods. In [16] and [17] different memetic algorithms have been researched in the context of mTSP. It is also not uncommon to apply local search based methods [18], [19].

Researches often approach multi-objective optimization with genetic algorithms. They have proven to generate very high-quality PF approximations [20]. NSGA-II [4] is one of the most commonly used methods. It uses only a single population, where parents are forced to compete with children. It utilizes the rank and crowding distance of the individuals in the selection process and to truncate the population after every generation.

NTGA [5] is an extension of classical NSGA-II. The authors have shown its efficiency for a bi-objective scheduling problem – Multi-Skill Resource Constrained Project Scheduling Problem (MS-RCPSP). Modified selection in NTGA no longer utilizes the crowding distance. Instead, a clone elimination method is employed to maintain the diversity of the population. Additionally, the size of the tournament has been adjusted. Moreover, children are created in a new population and no longer have to compete with the parent population. NTGA uses also an archive that contains current approximation of PF. Such approach is a base of considerations in the given paper.

Recently created Multi-Objective Evolutionary Algorithm integrating NSGA-II, SPEA2, and MEA/D (MOEA/NSM) [12] has been successfully applied to mTSP. It uses the crowding distance, decomposition and Pareto strength. The solution space is explored using subpopulation tables, where each subpopulation contains the best results for a given aggregation of criteria. Every individual undergoes a crossover, mutation, and 2-opt optimization. At the end of each generation, subpopulations of SPEA2 and NSGA-II are updated. Rank and crowding distance mechanisms are considered. The authors show that MOEA/NSM outperforms all other methods and is the *state-of-the-art* population-based algorithm for mTSP. Hence, MOEA/NSM is used in this paper for comparison.

Initial multi-objective methods focused mostly on the convergence. However, recent research has shifted the focus onto the diversity [13]. This article tries to find the balance between the two. Two methods are proposed. One that switches the focus between convergence and diversity. The other that emphasizes both in parallel populations.

III. PROBLEM

A TSP comprises of a set of m cities. In the problem a salesman must visit every city exactly once and return to the place where the travel started (initial city). The goal is to minimize the cost of travel of that route, given the cost of travel from city i to city j , is defined as c_{ij} and is part of the problem definition. TSP is equivalent to finding the minimum Hamiltonian cycle in a non-directed, weighted graph

[22], where nodes represent the cities, and weights represent the travel costs. Total cost of travel is calculated as the sum of edge weights and should be minimized. A symmetric TSP is considered in this paper, where $c_{ij} = c_{ji}$ for all cities $i, j \in \{0, 1, \dots, m-1\}$

In Multi-objective Travelling Salesman Problem (mTSP) multiple aspects of the route are evaluated [12]. It could be cost, time, length or risk of travel. The problem with n criteria and m cities is represented by n weighted graphs. For each $k \in 1, \dots, n$, graph G_k is a weighted graph, that represent k -th criterion. The edge weight between cities i and j in graph G_k is represented by $c_{ij}^{(k)}$. In this paper two criteria are considered.

IV. EVOLUTIONARY METHODS

This section contains the description of all the investigated methods. First, definitions of important terms related to the work are given. Next, parts that are common for each method are described. Then, reference methods are presented. Finally, two novel modifications are described.

A. Definitions of Terms

1) *Dominance Relation*: The comparison of multi-objective solutions is done with the dominance relation. Let z, z' be two points in the multi-objective solution space. z dominates z' when both Eq.1 and Eq.2 are satisfied:

$$\forall_{k=1}^n f_k(z) \leq f_k(z') \quad (1)$$

$$\exists_{k=1}^n f_k(z) < f_k(z') \quad (2)$$

Where n is the number of criteria, f_k is the objective function of k -th criterion.

2) *Pareto Front*: A set of all non-dominated solutions is called a Pareto Front (PF). Since the set of globally non-dominated points is not known, all methods create an approximation of PF.

B. Representation

The representation of an individual in genetic algorithm defines how a genome represents the solution in a given problem. It also determines the use of genetic operators. All methods in this paper use the same representation.

An individual, for the problem with m cities, is represented by the permutation vector $z = (m_1, m_2, \dots, m_m)$. Each gene is the number of the next city on the route. In TSP the full route must end on the same city that it started. Hence, in the calculation of the objective functions the cost of travel between m_m and m_1 must also be considered.

C. Initial Population

The first step of a genetic algorithm is the generation of an initial population. A random initialization is used. Every individual is initialized with a random permutation of all m cities. An additional mechanism enforces the uniqueness of all generated genotypes.

D. Genetic Operators

This chapter contains the description of both crossover, and mutation, which allow for exploitation, and exploration of the solution space.

1) *Crossover*: Crossover operator is responsible for the exploitation of space [23]. In the process two parent individuals are used to create two children individuals. In all methods, crossover is performed with a given probability (P_x parameter). In case of no crossover, parent genomes are copied over to the children individuals.

An Order Crossover (OX) has been selected [24]. It tends to retain the relative order of the genes and has been proven to work well for the ordering problems [25]. First, a part of the route is copied from the first parent, and then the rest of the route is reconstructed based on the genomes of the second parent. The part to copy is selected by randomly choosing two cut-points of the chromosome. The part between those two points is selected and copied to the child individual (in the same place of the genome). The remaining genes are filled, starting with the second cut-point, from the second parent. The order is maintained and already existing cities are skipped.

For example, given two parent individuals p_1 and p_2 :

$$\begin{aligned} p_1 &= (3\ 2\ 1\ | 8\ 4\ 6\ 7\ | 5\ 9), \\ p_2 &= (2\ 3\ 6\ | 5\ 8\ 4\ 1\ | 9\ 7). \end{aligned}$$

The first child c_1 is:

$$c_1 = (3\ 5\ 1\ | 8\ 4\ 6\ 7\ | 9\ 2).$$

The second child c_2 is generated by swapping the roles of two parents in the crossover process:

$$c_2 = (2\ 6\ 7\ | 5\ 8\ 4\ 1\ | 9\ 3).$$

2) *Mutation*: Mutation introduces a random perturbation in the genome and is responsible for exploration of the solution space [1]. It introduces one or more small changes within the genome with given probability P_m . The parameter is a probability of a mutation of a single individual.

An *Inversion* mutation has been selected. It performs an inversion of a randomly selected sequence of genes. The sequence is selected by randomly choosing two cut-points within the genome. All genes between those points are inverted.

For example, let's consider a parent p with the following genome and selected cut-points:

$$p = (3\ 2\ 1\ | 8\ 4\ 6\ 7\ | 5\ 9).$$

Mutation would result in the following genome c :

$$c = (3\ 2\ 1\ | 7\ 6\ 4\ 8\ | 5\ 9).$$

E. Selection

Selection operator is used to provide parent individuals for the genetic operators. It pressures the evolutionary process towards the desired results. In the case of multi-objective optimization it is important to find the PF approximation close to the true PF, but also to promote the diversity of the population. However, selection must also allow for the weaker

individuals in order to avoid local optima. In multi-objective optimization the selection is based on the rank and crowding distance [4].

The *rank* is calculated based on the dominance relation. First, all non-dominated individuals within the population gain rank 1. Then those individuals are exempt from further calculations. Rank 2 is assigned to non-dominated individuals from the remaining individuals. The process is iteratively repeated, until every individual has a rank assigned. Higher rank means that the individual is closer to the true PF.

The *crowding distance* is calculated based on the distance to other individuals. It is a volume of the largest cube that contains only that individual. A larger value means that there are fewer individuals in that part of the space.

Researched methods use a *tournament* selection. First, given number of individuals is randomly drawn from the population. They are compared according to given selection operators. The best individual, according to the operators, is returned. NSGA-II originally uses a tournament selection with two individuals, while NTGA allows for higher values of the tournament size. Moreover, in selection method NTGA uses an *archive* that contains all non-dominated individuals found in a given evolution process.

F. Evolutionary Process

The same evolutionary process is used in all methods. It is used to generate a new population P_{next} , from the current population $P_{current}$. It also includes a clone prevention method and archive usage introduced by NTGA. The process is described in pseudocode 1.

Algorithm 1 Pseudocode of the evolutionary process

```

1:  $P_{next} \leftarrow \emptyset$ 
2: while  $|P_{next}| < populationSize$  do
3:    $parents \leftarrow select(P_{current} \cup archive, operators)$ 
4:    $children \leftarrow mutate(crossover(parents))$ 
5:   while  $P_{next}$  contains  $children$  do
6:      $children \leftarrow mutate(children)$ 
7:   end while
8:    $evaluate(children)$ 
9:    $P_{next} \leftarrow P_{next} \cup children$ 
10: end while
11: return  $P_{next}$ 

```

In line 1 of Pseudocode 1, the next population is initialized to an empty collection. The loop between lines 2 and 10 performs the evolution until the next population reaches the desirable size. In line 3, parents are selected from the current population. NSGA-II does not use archive, but NTGA does (see line 3) in the selection method. The comparison is performed based on selected operators. Then, the children are created by performing crossover and mutation on the parents in line 4. Lines 5 to 7 describe the clone prevention mechanism. As long as the children already exist in the next population, they are mutated. Eventually, the children are evaluated in

line 8, and added to the next population in line 9. The next population is returned in line 11.

G. Switch Non-Dominated Tournament Genetic Algorithm

The proposed method switches selections (*crowding distance* and *rank*) to obtain an evenly distributed PF approximation with high spread. Switch Non-dominated Tournament Genetic Algorithm (sNTGA) is based on a recent NTGA [5]. It switches between two competing selection operators - primary $S_{primaryOperator}$ and a temporary one $S_{switchOperator}$. The former is used to obtain a diverse PF approximation, while the latter to also guarantee the convergence of the approximation. Both operators work in turns, and they promote different solutions, which allows for a better exploration of the solution space. Additionally, switch of the operators makes it easier to escape local optima. In consequence, improved solutions can be achieved. The time-frame, in which the operators work is defined by the number of births and the following parameters.

- S_{delay} - number of births, after which the temporary operators is switched on. At the very beginning of the evolution, the switch is not necessary, as the population is still diverse, and not yet converged.
- S_{each} - parameter determining a single cycle of the operators
- $S_{duration}$ - number of births after which $S_{switchOperator}$ should be switch off and $S_{primaryOperator}$ should be switched back on. It should always be lesser than S_{each} .

Pseudocode 2 describes the sNTGA.

Algorithm 2 Pseudocode of sNTGA

```

1:  $archive \leftarrow \emptyset$ 
2:  $P_{current} \leftarrow generateInitialPopulation()$ 
3:  $evaluate(P_{current})$ 
4: while  $stoppingCriteria()$  do
5:    $nonDominatedSorting(P_{current})$ 
6:    $crowdingDistanceAssignment(P_{current})$ 
7:    $updateArchive(P_{current})$ 
8:    $operator \leftarrow selectionOperator()$ 
9:    $P_{current} \leftarrow evolve(P_{current} \cup archive, operator)$ 
10: end while
11:  $updateArchive(P_{current})$ 
12: return  $archive$ 

```

An empty archive is initialized in line 1. In line 2, the current population is randomly initialized. It is then evaluated in line 3. The loop between lines 4 and 10 runs until the stopping criteria is reached. In the article it has been set to the given number of births. In line 5, the non-dominated sorting is performed and the crowding distance is assigned in line 6. The archive is updated in line 7, by adding all non-dominated individuals, and removing those that became dominated. In line 8 a selection operator is determined (described in pseudocode 3). Then the current population is evolved (described in pseudocode 1), based on the current individuals, the archive, and selection operators. Finally, the archive is again updated

in line 11 and it is returned in line 12, where it contains the PF approximation.

Algorithm 3 Pseudocode of $selectionOperator$ in sNTGA

```

1:  $switchBirthsCount \leftarrow currentBirthsCount - S_{delay}$ 
2: if  $switchBirthsCount < 0$  then
3:    $operator \leftarrow S_{primaryOperator}$ 
4: else if  $switchBirthsCount \bmod S_{each} < S_{duration}$  then
5:    $operator \leftarrow S_{switchOperator}$ 
6: else
7:    $operator \leftarrow S_{primaryOperator}$ 
8: end if
9: return  $operator$ 

```

To determine the current selection operators for sNTGA, the moment, at which operator should change is calculated in line 1. A $switchBirthsCount$ variable is used. It is calculated as the current number of births minus the delay parameter. If that number is smaller than 0 (check in line 2) then the primary operator is selected in line 3. Otherwise, a check is performed to verify, which operator should be used, in line 4. If not enough births have happened, then the temporary operator is used in line 5, otherwise primary operator is used in line 7. The selected operator is returned in line 9.

Researched sNTGA uses the *crowding distance* as the primary operator and *rank* as the temporary operator.

H. Co-Evolutionary Non-Dominated Tournament Genetic Algorithm

Both *rank* and *crowding distance* operator can be successfully used in multi-objective optimization methods. However, since both operators work on the same population, one operator might diminish the effect of the other operator. Hence, a Co-evolutionary Non-Dominated Tournament Genetic Algorithm (cNTGA) is proposed. The main motivation in cNTGA is to enforce cooperation of selection methods by operating on two separate populations connected in the evaluation process. Thus co-evolution mechanism is applied.

A cNTGA utilizes two populations with different selection operators. The exchange of information is possible due to the use of the same archive. The archive is used in the evaluation and selection process of both populations. Additionally, at the end of each generation, individuals of both populations are added to the archive. The method requires only one additional parameter K_{RANK} - it defines the percentage of the initial population size, that should be assigned to the population that uses *rank* operator. cNTGA is described in pseudocode 4.

Algorithm 4 Pseudocode of cNTGA

```

1: archive ← ∅
2:  $P_{RANK}, P_{CD} \leftarrow generateInitialPopulation()$ 
3:  $evaluate(P_{RANK})$ 
4:  $evaluate(P_{CD})$ 
5: while  $stoppingCriteria()$  do
6:    $nonDominatedSorting(P_{RANK})$ 
7:    $nonDominatedSorting(P_{CD})$ 
8:    $crowdingDistanceAssignment(P_{CD})$ 
9:    $updateArchive(P_{RANK} \cup P_{CD})$ 
10:   $P_{RANK} \leftarrow evolve(P_{RANK} \cup archive, \geq_{RANK})$ 
11:   $P_{CD} \leftarrow evolve(P_{CD} \cup archive, \geq_{CD})$ 
12: end while
13:  $updateArchive(P_{RANK} \cup P_{CD})$ 
14: return archive

```

An empty archive is initialized in line 1. Then two populations P_{RANK} and P_{CD} are randomly initialized in line 2. Then, they are evaluated in lines 3 and 4 respectively. The loop between lines 5 and 12 runs until the stopping criteria is reached. In the paper, the number of births is used. In lines 6 and 7, the populations are sorted according to the appropriate operator. The crowding distances are assigned for population P_{CD} in line 8. Then, the archive is updated in line 9, with the individuals from both populations. The populations are evolved (pseudocode 1) in line 10 and 11 using \geq_{RANK} and \geq_{CD} operators. The archive is once again updated in line 13. Finally, the archive containing the PF approximation is returned in line 14.

In cNTGA, co-evolution significantly reduces the number of parameters (comparing to sNTGA) and results in an easier investigation process. Finally, the method decides which selection operator is more “useful” in the given problem, not the researcher in the tuning process.

I. Reference Methods

The researched methods are compared to three selected, reference methods. First, a NSGA-II [4] has been selected, as it is the most common method in the literature. NTGA [5] is selected, because approaches described in this paper are based on it. Finally, MOEA/NSM [12] is used as the current best *state-of-the-art* method.

V. EXPERIMENTS AND RESULTS

This section presents experimental procedure to verify effectiveness of the proposed methods (sNTGA and cNTGA) by empirically comparing results to other reference methods. Thus, used mTSP instances are presented, Quality Measure for multi-objective optimization and methods setup are given. Finally, results and selected visualizations are presented.

A. Data Instances

Data instances used in the research are commonly used in literature, e.g. 9 instances *euclid**** from TSPLIB (e.g. [21]) and {kroAB100, kroAB200} generated from DIMACS code. Instances differ in number of cities, which affects the complexity and size of the solution landscape.

B. Quality Measures

To evaluate the results, the QMs proposed in [6] are used. This section contains their description along with the description of reference points required for their calculation.

1) *Euclidean Distance*: Euclidean distance (*ED*) is the average distance between every point on the PF approximation and a so called Perfect Point. Where the Perfect Point comprises of the best values of all objectives. ED can be formally defined by equation 3.

$$ED(PF) = \frac{\sum_{i=1}^{|PF|} d_i}{|PF|} \quad (3)$$

Where PF is the Pareto Front, d_i is the distance from the i 'th point to the Perfect Point.

ED measures the convergence of the PF approximation and should be minimized.

2) *Hypervolume*: Hypervolume (*HV*) is the volume of hypercube defined by the PF approximation and the Nadir Point. Where the Nadir Point comprises of the worst values of all objectives. HV can be formally defined by equation 4.

$$HV(PF) = \Lambda\left(\bigcup_{s \in PF} \{s' | s \prec s' \prec s^{nadir}\}\right) \quad (4)$$

Where PF is an approximation of PF. s is the point of approximated PF. s^{nadir} is a *Nadir Point*. Λ is a Lebesgue measure, which generalizes the a volume. \prec is a domination relation.

Hypervolume is a measure of spread, but is also influenced by the convergence of the PF approximation. It should be maximized.

3) *Pareto Front Size*: Pareto Front Size (*PFS*) is the number of points on the PF approximation. It is the measure of diversity and should be maximized.

4) *Spacing*: Spacing (*S*) is the average distance between the consecutive points on the PF approximation. S can be formally defined by equation 5.

$$S(PF) = \sqrt{\frac{1}{|PF|} \sum_{i=1}^{|PF|} (d_i - \bar{d})^2} \quad (5)$$

Where PF is the approximation of the PF. d_i is the distance from the i – th point the next consecutive point.

S is the measure of uniformity and should be minimized.

5) *Ratio of Non-dominated Individuals*: Ratio of Non-dominated Individuals (*RNI*) is the value of *PFS* divided by the number of births. It measures efficiency of the method and should be maximized.

6) *Purity*: Purity is used for a direct comparison of two PF approximations. It is the number of points that remain non-dominated, when combined with the PF approximation from another method. Purity should be maximized.

C. Parameters

Each method is tuned experimentally (like cNTGA, sNTGA) or optimal configuration has been used based on publications (e.g. MOEA/NSM [21] or NTGA [5]). Each method is limited by number of *births* (a number of all visited points). All configurations are presented in Tab. I.

For all investigated methods this value is given according to number of cities as follows:

- 100 cities – 10 mln births,
- 200 cities – 13 mln births,
- 300 cities – 16 mln births,
- 500 cities – 22 mln births,

Such *births* limitations are connected to the size of landscape (number of cities) and limit of computational time required by experimental procedure.

All investigated methods (NTGA, sNTGA, cNTGA and NSGA-II) have been implemented in Java using standard libraries. An exception is MOEA/NSM, where authors of [21] code¹ has been used.

D. Experiments

Averaged results from 50 runs of 5 methods for 11 instances are presented in Table II. The comparison shows that in each case the best values of *RNI* have been achieved by cNTGA or sNTGA. Proposed methods also outperform others in *PFS* context – they give 2–3 times larger number of points in approx. PF.

NSGA-II created interesting results – in 6/11 cases the best *ED* and *S* values have been achieved – it is connected to the fact, that this method gives very narrow PF approx. that is located in the “centre” (see Fig.1 or Fig.2). Other cases (5/11) are occupied by NTGA, what confirms that *ED* prefers methods that focus on PF “centre”.

In opposite cases (4/11) better *S* values were obtained by cNTGA with much large *PFS* value. Almost every time (9/11 cases) the best value of *HV* was achieved by MOEA/NSM.

All results have been averaged and presented in Table III. Two proposed methods (sNTGA and cNTGA) compete successfully with MOEA/NSM. In case of *ED*, all mentioned methods give results that are almost the same. The Wilcoxon signed-rank test showed that results of these three methods are not statistically different.

Results presented in Table III show that MOEA/NSM gives better approx. PF in *HV* context. The difference between sNTGA and MOEA/NSM results is statistically significant – Wilcoxon signed-rank test ($W_{0.05} = 66 > W_c = 13$) confirmed that.

For other QM’s, two proposed methods – cNTGA and sNTGA – give results (*ED*, *RNI* and *S*) that are not statistically different. The exception is *HV*, where sNTGA outperforms cNTGA and it is statistically significant ($W_{0.05} = 65 > W_c = 13$). However, both methods outperform MOEA/NSM. E.g. comparing results for cNTGA and MOEA/NSM Wilcoxon signed-rank test gives: *PFS* ($W_{0.05}$

$= 66 > W_c = 13$), *RNI* ($W_{0.05} = 64 > W_c = 13$) and *S* ($W_{0.05} = 66 > W_c = 13$).

E. Visualizations and approx. PF analysis

To visualize approximations of PF, selected graphs have been prepared. They present the “averaged” PF – modified version of empirical attainment function (EAF) [26]. Each graph contains data of PF from 50 independent runs of selected method.

For smaller instances (i.e. euclidAB100) the results of 5 investigated methods are very similar. Hence the visualization has been omitted. Only two methods – NSGA-II and NTGA – give worse results: approx. PF is dominated by others and have larger standard deviation.

Graph presented on Fig.1 shows results for more complex instance, euclidAB300. Methods cNTGA, sNTGA and MOEA/NSM compete successfully. NTGA gives the worst solution. Quite interesting are results of NSGA-II – approx. PF is too short but focused in central region of PF.

The more difficult instance (see Fig.2, euclidAB500) suggests that MOEA/NSM is effective in the central area of PF – QM’s *S* and *HV* confirm that. However, in other areas, sNTGA and cNTGA can compete and give very good results – a large number of points. Moreover, PF approximation created by NSGA-II is very short but of very high quality and focused in the central area.

To get a more detailed image of results **Purity** measure have been (see Tab. IV) used to compare gained approx. PF, where methods are compared in pairs. Results show that it is quite difficult to select the method with the best results. Fig.??- Fig.1 showed that MOEA/NSM gives better results in “centre” region, cNTGA wins in other regions. Using Purity measure, it gives domination of MOEA/NSM in 53.9% but cNTGA dominates in 54.3%. Detailed analysis of the data shows that cNTGA “wins” in instances with 500 cities and fails in 300 cities – it can be a suggestion that *births* limits cNTGA to much and such aspect should be investigated more carefully in further research. It is worth mentioning that cNTGA and sNTGA give significantly larger PF approx. (see *PFS* values in Tab.II or Tab.III) what can disturb a little interpretation of Purity values.

Another conclusion gained from Tab.IV is that cNTGA outperforms results of sNTGA in each instance – approx. PF “dominates” in 62%. Moreover, sNTGA does not compete with MOEA/NSM so successfully, and is “dominated” in 59.8%.

F. Summary

cNTGA and sNTGA methods are effective and can compete with MOEA/NSM results (e.g. better *PFS* and *S* values). However, there is a strong need to focus the methods in central area of PF. Moreover, proposed methods have better efficiency than MOEA/NSM – the *RNI* confirmed that. The last but not least, is the conceptual aspect – cNTGA and sNTGA are methods that have only few parameters, are easy to understand and tune.

¹MOEA/NSM code: <https://github.com/MOEA-NSM/moea-nsm>

TABLE I
CONFIGURATIONS FOR INVESTIGATED METHODS

| | NSGAI | MOEA/NSM | NTGA | cNTGA | sNTGA |
|--------------------------------|-----------|---|-----------------|------------------|---|
| mutation (P_m) | Swap, 10% | Swap, 10% | Inversion, 0.5% | Inversion, 0.25% | Inversion, 0.5% |
| crossover (P_x) | OX, 80% | OX, 80% | OX, 40% | OX, 70% | OX, 60% |
| tournament size (T_{size}) | 2 | 2 | 25 | 25 | 25 |
| pop_size (K) | 500 | 100 | 500 | 500 | 500 |
| additional parameters | | P_{obj}^1, P_{obj}^1 20 P_{pon} 40 M_t 100 N_t, S_t 50 P_{2OPT} 10% | | K_{RANK} 60% | S_{delay} 4 000 000 S_{each} 1 000 000 $S_{duration}$ 750 000 |

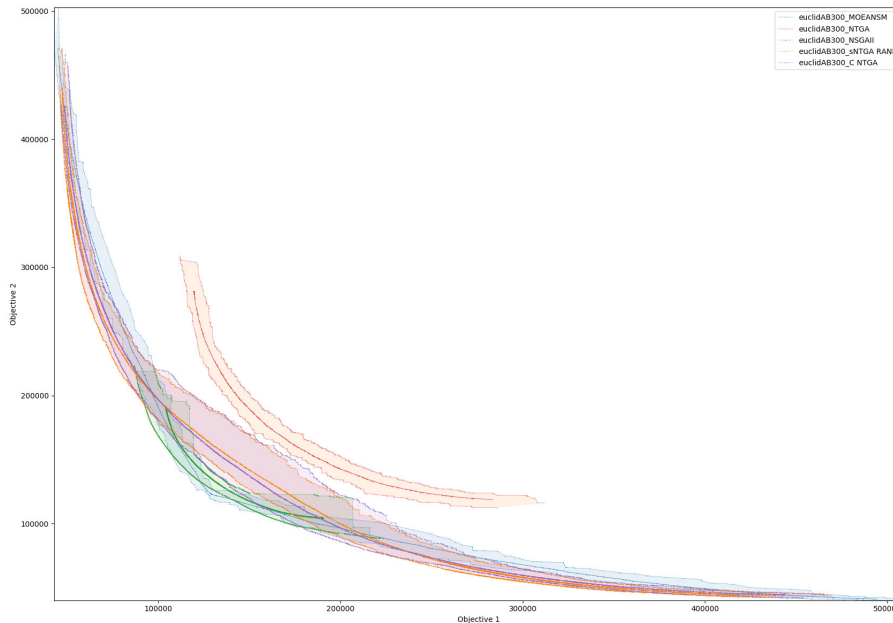


Fig. 1. Comparison of average approx. Pareto Fronts for data instance euclidAB300

VI. CONCLUSIONS AND FUTURE WORK

This article introduces two new methods based on recent NTGA. The first one is sNTGA, which uses the rank and crowding distance operators sequentially. The second one - cNTGA utilizes two subpopulations, where one of them uses rank, and the other one uses crowding distance in their respective selection methods. Both subpopulations cooperate by performing the selection on a shared archive of non-dominated individuals.

The methods are compared to the *state-of-the-art* population-based MOEA/NSM. Proposed methods require fewer parameters and it is argued that they are less complex. Additionally, the results are evaluated with a set of QMs on the selected instances of TSP.

Another promising trend in literature is specialization and hybridization of metaheuristics. Proposed methods could be

extended by some local search techniques like 2-opt or/and genetic operators that are considered to be more effective in TSP, like Edge Crossover Operator [27].

It is shown that MOEA/NSM achieves better results in the 'centre' of the PF approximation. A different selection method that would emphasize the convergence in that area could be beneficial. Also, research on the different multi-objective problems in the context of sNTGA and cNTGA is an interesting direction for future work.

REFERENCES

- [1] Abdoun, Otman, Jaafar Abouchabaka, and Chakir Tajani. "Analyzing the performance of mutation operators to solve the travelling salesman problem." arXiv preprint arXiv:1203.3099 (2012).
- [2] Hoffman, Karla L., Manfred Padberg, and Giovanni Rinaldi. "Traveling salesman problem." Encyclopedia of operations research and management science (2013): 1573-1578.

TABLE II
AVERAGED VALUES OF QMS FOR SELECTED DATA INSTANCES AND ALL INVESTIGATED METHODS

| Instance | Method | ED | | HV [10^8] | | PFS | | S | | RNI | |
|-------------|----------|----------------|----------------|----------------|----------------|----------------|---------------|----------------|----------------|-----------------|-----------------|
| | | Avg | Std | Avg | Std | Avg | Std | Avg | Std | Avg | Std |
| kroAB100 | NSGA-II | 0.19726 | 0.00504 | 0.86063 | 0.00426 | 499.96 | 0.20 | 0.00062 | 0.00026 | 0.000050 | 0.000000 |
| | NTGA | 0.18122 | 0.00705 | 0.84713 | 0.00662 | 928.16 | 294.22 | 0.00154 | 0.00060 | 0.000093 | 0.000029 |
| | sNTGA | 0.24831 | 0.00999 | 0.88968 | 0.00091 | 2816.50 | 481.15 | 0.00041 | 0.00005 | 0.000282 | 0.000048 |
| | cNTGA | 0.24838 | 0.00880 | 0.88959 | 0.00103 | 3062.04 | 451.13 | 0.00037 | 0.00004 | 0.000306 | 0.000045 |
| | MOEA/NSM | 0.23644 | 0.00229 | 0.88974 | 0.00068 | 134.46 | 3.80 | 0.00295 | 0.00020 | 0.000013 | 0.000000 |
| kroAB200 | NSGA-II | 0.15208 | 0.00209 | 0.86446 | 0.00318 | 499.92 | 0.56 | 0.00031 | 0.00014 | 0.000038 | 0.000000 |
| | NTGA | 0.14295 | 0.00363 | 0.85400 | 0.00418 | 1178.50 | 319.39 | 0.00147 | 0.00064 | 0.000091 | 0.000025 |
| | sNTGA | 0.24601 | 0.01243 | 0.91769 | 0.00078 | 2895.34 | 587.05 | 0.00031 | 0.00004 | 0.000223 | 0.000045 |
| | cNTGA | 0.23118 | 0.00960 | 0.91756 | 0.00072 | 2937.00 | 512.92 | 0.00026 | 0.00004 | 0.000226 | 0.000039 |
| | MOEA/NSM | 0.21721 | 0.00416 | 0.91778 | 0.00064 | 118.92 | 9.13 | 0.00305 | 0.00020 | 0.000009 | 0.000001 |
| euclidAB100 | NSGA-II | 0.22915 | 0.00593 | 0.83400 | 0.00560 | 500.00 | 0.00 | 0.00061 | 0.00027 | 0.000050 | 0.000000 |
| | NTGA | 0.21413 | 0.00879 | 0.82423 | 0.00551 | 765.86 | 187.23 | 0.00143 | 0.00050 | 0.000077 | 0.000019 |
| | sNTGA | 0.27353 | 0.00808 | 0.86446 | 0.00083 | 2104.08 | 376.61 | 0.00043 | 0.00005 | 0.000210 | 0.000038 |
| | cNTGA | 0.27143 | 0.00706 | 0.86375 | 0.00101 | 2238.10 | 361.25 | 0.00041 | 0.00005 | 0.000224 | 0.000036 |
| | MOEA/NSM | 0.25528 | 0.00254 | 0.86419 | 0.00058 | 122.04 | 6.35 | 0.00264 | 0.00018 | 0.000012 | 0.000001 |
| euclidAB300 | NSGA-II | 0.15484 | 0.00210 | 0.84077 | 0.00335 | 499.92 | 0.34 | 0.00031 | 0.00021 | 0.000031 | 0.000000 |
| | NTGA | 0.20411 | 0.00287 | 0.81625 | 0.00215 | 282.52 | 20.00 | 0.00119 | 0.00039 | 0.000018 | 0.000001 |
| | sNTGA | 0.24592 | 0.00805 | 0.91417 | 0.00128 | 2044.48 | 344.46 | 0.00038 | 0.00006 | 0.000128 | 0.000022 |
| | cNTGA | 0.24155 | 0.00640 | 0.91322 | 0.00108 | 1891.90 | 281.66 | 0.00038 | 0.00008 | 0.000118 | 0.000018 |
| | MOEA/NSM | 0.25118 | 0.00426 | 0.91606 | 0.00065 | 70.86 | 4.21 | 0.00345 | 0.00035 | 0.000004 | 0.000000 |
| euclidAB500 | NSGA-II | 0.13579 | 0.00335 | 0.84135 | 0.00425 | 493.62 | 28.90 | 0.00026 | 0.00024 | 0.000022 | 0.000001 |
| | NTGA | 0.20428 | 0.00272 | 0.79587 | 0.00221 | 311.54 | 20.54 | 0.00086 | 0.00026 | 0.000014 | 0.000001 |
| | sNTGA | 0.23147 | 0.01084 | 0.91348 | 0.00370 | 1236.66 | 104.87 | 0.00060 | 0.00013 | 0.000056 | 0.000005 |
| | cNTGA | 0.24321 | 0.00796 | 0.91286 | 0.00244 | 1158.04 | 119.49 | 0.00062 | 0.00015 | 0.000053 | 0.000005 |
| | MOEA/NSM | 0.25053 | 0.00432 | 0.92940 | 0.00052 | 71.00 | 3.63 | 0.00385 | 0.00030 | 0.000003 | 0.000000 |
| euclidCD100 | NSGA-II | 0.22927 | 0.00579 | 0.83732 | 0.00461 | 499.84 | 1.12 | 0.00058 | 0.00021 | 0.000050 | 0.000000 |
| | NTGA | 0.22511 | 0.01555 | 0.82967 | 0.00487 | 813.28 | 262.10 | 0.00142 | 0.00045 | 0.000081 | 0.000026 |
| | sNTGA | 0.27988 | 0.00796 | 0.86868 | 0.00082 | 2165.74 | 372.63 | 0.00045 | 0.00005 | 0.000217 | 0.000037 |
| | cNTGA | 0.27551 | 0.00883 | 0.86785 | 0.00103 | 2308.40 | 347.60 | 0.00042 | 0.00006 | 0.000231 | 0.000035 |
| | MOEA/NSM | 0.25717 | 0.00210 | 0.86806 | 0.00089 | 124.74 | 4.49 | 0.00275 | 0.00016 | 0.000012 | 0.000000 |
| euclidCD300 | NSGA-II | 0.15006 | 0.00167 | 0.84324 | 0.00413 | 499.86 | 0.75 | 0.00030 | 0.00028 | 0.000031 | 0.000000 |
| | NTGA | 0.19801 | 0.00301 | 0.82061 | 0.00233 | 280.48 | 19.78 | 0.00102 | 0.00022 | 0.000018 | 0.000001 |
| | sNTGA | 0.23836 | 0.00792 | 0.91524 | 0.00101 | 2004.44 | 402.02 | 0.00037 | 0.00006 | 0.000125 | 0.000025 |
| | cNTGA | 0.23630 | 0.00694 | 0.91429 | 0.00094 | 1820.68 | 276.83 | 0.00039 | 0.00007 | 0.000114 | 0.000017 |
| | MOEA/NSM | 0.24571 | 0.00345 | 0.91729 | 0.00054 | 71.90 | 6.03 | 0.00347 | 0.00036 | 0.000004 | 0.000000 |
| euclidCD500 | NSGA-II | 0.13560 | 0.00294 | 0.84205 | 0.00339 | 498.18 | 7.65 | 0.00023 | 0.00028 | 0.000023 | 0.000000 |
| | NTGA | 0.20330 | 0.00282 | 0.79741 | 0.00236 | 319.60 | 24.18 | 0.00078 | 0.00025 | 0.000015 | 0.000001 |
| | sNTGA | 0.23244 | 0.00873 | 0.91342 | 0.00319 | 1252.72 | 117.90 | 0.00059 | 0.00012 | 0.000057 | 0.000005 |
| | cNTGA | 0.24222 | 0.00694 | 0.91302 | 0.00246 | 1164.88 | 105.37 | 0.00057 | 0.00007 | 0.000053 | 0.000005 |
| | MOEA/NSM | 0.24629 | 0.00383 | 0.92956 | 0.00031 | 71.40 | 1.85 | 0.00405 | 0.00041 | 0.000003 | 0.000000 |
| euclidEF100 | NSGA-II | 0.21616 | 0.00468 | 0.84541 | 0.00425 | 499.94 | 0.31 | 0.00051 | 0.00012 | 0.000050 | 0.000000 |
| | NTGA | 0.20595 | 0.01472 | 0.83559 | 0.00686 | 907.94 | 276.57 | 0.00118 | 0.00033 | 0.000091 | 0.000028 |
| | sNTGA | 0.25322 | 0.00778 | 0.87290 | 0.00084 | 2252.02 | 356.98 | 0.00043 | 0.00005 | 0.000225 | 0.000036 |
| | cNTGA | 0.24937 | 0.00871 | 0.87260 | 0.00100 | 2380.86 | 393.59 | 0.00042 | 0.00005 | 0.000238 | 0.000039 |
| | MOEA/NSM | 0.24236 | 0.00249 | 0.87311 | 0.00065 | 124.38 | 5.19 | 0.00258 | 0.00018 | 0.000012 | 0.000001 |
| euclidEF300 | NSGA-II | 0.15452 | 0.00225 | 0.83933 | 0.00347 | 499.94 | 0.24 | 0.00030 | 0.00018 | 0.000031 | 0.000000 |
| | NTGA | 0.20415 | 0.00259 | 0.81662 | 0.00218 | 282.42 | 20.09 | 0.00110 | 0.00025 | 0.000018 | 0.000001 |
| | sNTGA | 0.24362 | 0.00937 | 0.91296 | 0.00091 | 2111.34 | 462.54 | 0.00037 | 0.00006 | 0.000132 | 0.000029 |
| | cNTGA | 0.23730 | 0.00765 | 0.91214 | 0.00080 | 1831.44 | 280.77 | 0.00038 | 0.00006 | 0.000114 | 0.000018 |
| | MOEA/NSM | 0.24726 | 0.00395 | 0.91467 | 0.00059 | 70.00 | 4.14 | 0.00327 | 0.00032 | 0.000004 | 0.000000 |
| euclidEF500 | NSGA-II | 0.13630 | 0.00316 | 0.84093 | 0.00374 | 494.20 | 21.96 | 0.00024 | 0.00025 | 0.000022 | 0.000001 |
| | NTGA | 0.20351 | 0.00259 | 0.79635 | 0.00238 | 318.98 | 20.26 | 0.00083 | 0.00029 | 0.000014 | 0.000001 |
| | sNTGA | 0.23069 | 0.00885 | 0.91223 | 0.00321 | 1207.88 | 112.44 | 0.00056 | 0.00012 | 0.000055 | 0.000005 |
| | cNTGA | 0.24066 | 0.00608 | 0.91137 | 0.00223 | 1114.08 | 100.15 | 0.00057 | 0.00007 | 0.000051 | 0.000005 |
| | MOEA/NSM | 0.24131 | 0.00376 | 0.92867 | 0.00050 | 70.20 | 2.89 | 0.00368 | 0.00040 | 0.000003 | 0.000000 |

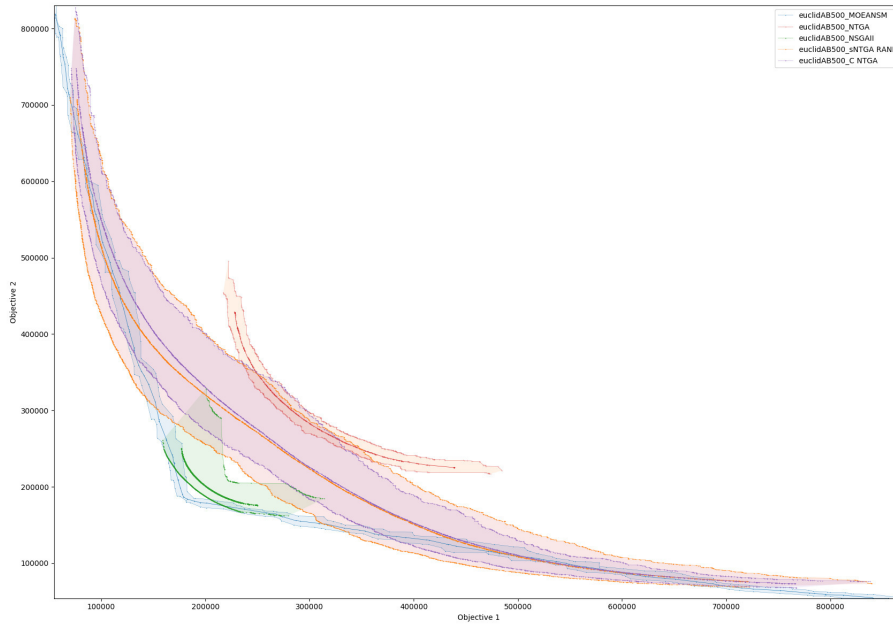


Fig. 2. Comparison of average approx. Pareto Fronts for data instance euclidAB500

TABLE III
AVERAGED RESULTS COMPARISON FOR ALL INVESTIGATED METHODS

| | min ED | max HV [10^8] | max PFS | min S | max RNI |
|----------|----------------|----------------------|----------------|----------------|----------------|
| NSGA-II | 0.17191 | 0.84450 | 498.67 | 0.00039 | 0.00004 |
| NTGA | 0.19879 | 0.82125 | 580.84 | 0.00117 | 0.00005 |
| sNTGA | 0.24759 | 0.89954 | 2008.29 | 0.00045 | 0.00016 |
| cNTGA | 0.24701 | 0.89893 | 1991.58 | 0.00044 | 0.00016 |
| MOEA/NSM | 0.24461 | 0.90441 | 95.45 | 0.00325 | 0.00001 |

- [3] Cai, Xinye, et al. "An adaptive memetic framework for multi-objective combinatorial optimization problems: studies on software next release and travelling salesman problems." *Soft Comp.* 21.9 (2017): 2215-2236.
- [4] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE transactions on evolutionary computation* 6.2 (2002): 182-197.
- [5] Laszczyk, Maciej, and Paweł B. Myszowski. "Improved selection in evolutionary multi-objective optimization of Multi-Skill Resource-Constrained project scheduling problem." *Information Sciences* 481 (2019): 412-431.
- [6] Laszczyk, Maciej, and Paweł B. Myszowski. "Survey of quality measures for multi-objective optimization. Construction of complementary set of multi-objective quality measures." *Swarm and Evolutionary Computation* 48 (2019): 109-133.
- [7] Miller, Donald L., and Joseph F. Pekny. "Exact solution of large asymmetric traveling salesman problems." *Science* 251.4995 (1991): 754-761.
- [8] Bolanos, R., M. Echeverry, and J. Escobar. "A multiobjective non-dominated sorting genetic algorithm (NSGA-II) for the Multiple Traveling Salesman Problem." *Decision Science Letters* 4.4 (2015): 559-568.
- [9] Elgesem, Aurora Smith, et al. "A traveling salesman problem with pickups and deliveries and stochastic travel times: An application from chemical shipping." *European Journal of Operational Research* 269.3

- (2018): 844-859.
- [10] Braekers, Kris, Katrien Ramaekers, and Inneke Van Nieuwenhuysse. "The vehicle routing problem: State of the art classification and review." *Computers & Industrial Engineering* 99 (2016): 300-313.
- [11] Maity, Samir, Arindam Roy, and Manoranjan Maiti. "An imprecise multi-objective genetic algorithm for uncertain constrained multi-objective solid travelling salesman problem." *Expert Systems With Applications* 46 (2016): 196-223.
- [12] Moraes, Deyvid Heric, et al. "A novel multi-objective evolutionary algorithm based on subpopulations for the bi-objective traveling salesman problem." *Soft Computing* (2018): 1-12.
- [13] Seada, Haitham, Mohamed Abouhawwash, and Kalyanmoy Deb. "Multiphase Balance of Diversity and Convergence in Multiobjective Optimization." *IEEE Transactions on Evolutionary Computation* 23.3 (2018): 503-513.
- [14] Ariyasingha, I. D. I. D., and T. G. I. Fernando. "Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem." *Swarm and Evolutionary Comp.* 23 (2015): 11-26.
- [15] Ke, Liangjun, Qingfu Zhang, and Roberto Battiti. "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant-colony." *IEEE transactions on cybernetics* 43.6 (2013): 1845-1859.
- [16] Ke, Liangjun, Qingfu Zhang, and Roberto Battiti. "Hybridization of decomposition and local search for multiobjective optimization." *IEEE transactions on cybernetics* 44.10 (2014): 1808-1820.
- [17] Chen, Xinye, et al. "Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems." *IEEE Access* 6 (2018): 21745-21757.
- [18] Cornu, Marek, Tristan Cazenave, and Daniel Vanderpooten. "Perturbed decomposition algorithm applied to the multi-objective traveling salesman problem." *Computers & Operations Research* 79 (2017): 314-330.
- [19] Lust, Thibaut, and Jacques Teghem. "The multiobjective traveling salesman problem: a survey and a new approach." *Advances in Multi-Objective Nature Inspired Computing*. Springer, Berlin, Heidelberg, 2010. 119-141.
- [20] Qamar, Nosheen, Nadeem Akhtar, and Irfan Younas. "Comparative Analysis of Evolutionary Algorithms for Multi-Objective Travelling

TABLE IV
AVERAGED VALUES OF PURITY FOR SELECTED INVESTIGATED METHODS

| Instance | sNTGA vs cNTGA | | cNTGA vs sNTGA | | sNTGA vs MOEA/NSM | | MOEA/NSM vs sNTGA | | cNTGA vs MOEA/NSM | | MOEA/NSM vs cNTGA | |
|-------------|----------------------|-------|----------------------|-------|-------------------------|-------|-------------------------|-------|-------------------------|-------|-------------------------|-------|
| | Avg | Std | Avg | Std | Avg | Std | Avg | Std | Avg | Std | Avg | Std |
| | euclidAB100 | 0.397 | 0.187 | 0.652 | 0.187 | 0.450 | 0.173 | 0.687 | 0.166 | 0.571 | 0.141 | 0.595 |
| euclidAB300 | 0.275 | 0.181 | 0.656 | 0.210 | 0.216 | 0.076 | 0.728 | 0.066 | 0.294 | 0.102 | 0.714 | 0.082 |
| euclidAB500 | 0.391 | 0.363 | 0.607 | 0.356 | 0.716 | 0.216 | 0.369 | 0.210 | 0.854 | 0.140 | 0.240 | 0.138 |
| euclidCD100 | 0.384 | 0.204 | 0.642 | 0.202 | 0.431 | 0.156 | 0.694 | 0.141 | 0.534 | 0.187 | 0.616 | 0.163 |
| euclidCD300 | 0.243 | 0.152 | 0.695 | 0.180 | 0.201 | 0.068 | 0.748 | 0.064 | 0.305 | 0.094 | 0.714 | 0.075 |
| euclidCD500 | 0.462 | 0.327 | 0.540 | 0.322 | 0.750 | 0.187 | 0.336 | 0.172 | 0.863 | 0.124 | 0.233 | 0.130 |
| euclidEF100 | 0.420 | 0.216 | 0.623 | 0.195 | 0.485 | 0.183 | 0.647 | 0.147 | 0.563 | 0.187 | 0.584 | 0.161 |
| euclidEF300 | 0.239 | 0.196 | 0.684 | 0.231 | 0.187 | 0.052 | 0.751 | 0.059 | 0.293 | 0.083 | 0.720 | 0.076 |
| euclidEF500 | 0.393 | 0.298 | 0.605 | 0.293 | 0.788 | 0.188 | 0.293 | 0.190 | 0.922 | 0.085 | 0.154 | 0.093 |
| kroAB100 | 0.448 | 0.210 | 0.597 | 0.210 | 0.453 | 0.177 | 0.669 | 0.146 | 0.527 | 0.194 | 0.594 | 0.185 |
| kroAB200 | 0.447 | 0.251 | 0.513 | 0.251 | 0.230 | 0.123 | 0.655 | 0.138 | 0.251 | 0.116 | 0.771 | 0.122 |
| Average | 0.372 | 0.235 | 0.620 | 0.240 | 0.446 | 0.146 | 0.598 | 0.136 | 0.543 | 0.132 | 0.539 | 0.125 |

Salesman Problem." INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS 9.2 (2018): 371-379.

- [21] Moraes, Deyvid Heric, et al. "A novel multi-objective evolutionary algorithm based on subpopulations for the bi-objective traveling salesman problem." *Soft Computing* (2018): 1-12.
- [22] Hassin, Refael, and Shlomi Rubinstein. "Better approximations for max TSP." *Information Processing Letters* 75.4 (2000): 181-186.
- [23] Abdoun, Otman, and Jaafar Abouchabaka. "A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem." *arXiv preprint arXiv:1203.3097* (2012).
- [24] Varun Kumar, S. G., and R. Panneerselvam. "A study of crossover operators for genetic algorithms to solve VRP and its variants and new sinusoidal motion crossover operator." *Int. J. Comput. Intell. Res* 13.7 (2017): 1717-1733.
- [25] Stehling, Thiago Muniz, and Sergio Ricardo de Souza. "A Comparison of Crossover Operators Applied to the Vehicle Routing Problem with Time Window." *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2017.
- [26] Lopez-Ibanez M., Paquete L., and Stutzle T. "Exploratory Analysis of Stochastic Local Search Algorithms in Biobjective Optimization", *Experimental Methods for the Analysis of Opt. Alg.* (2010): 209-222.
- [27] Whitley, L. Darrell, Timothy Starkweather, and D'Ann Fuquay. "Scheduling problems and traveling salesmen: The genetic edge recombination operator." *ICGA*. Vol. 89. 1989.