Balancing the Subtours for Multiple TSP Approached with ACS: Clustering-Based Approaches Vs. MinMax Formulation

Chapter · January 2018

DOI: 10.1007/978-3-319-69710-9_15

CITATIONS
4

READS
108

3 authors:

Raluca Necula
Universitatea Alexandru Ioan Cuza
9 PUBLICATIONS **54** CITATIONS
SEE PROFILE

Madalina Raschip
Universitatea Alexandru Ioan Cuza
24 PUBLICATIONS **83** CITATIONS
SEE PROFILE

Mihaela Breaban
Universitatea Alexandru Ioan Cuza
33 PUBLICATIONS **193** CITATIONS
SEE PROFILE

Balancing the Subtours for Multiple TSP Approached with ACS: Clustering-based Approaches vs. MinMax Formulation

Raluca Necula[1], Madalina Raschip[2], and Mihaela Breaban[1]

[1] Alexandru Ioan Cuza University, Iasi, Romania
{raluca.necula,pmihaela}@info.uaic.ro
[2] University of Neuchatel, Switzerland
madalina.raschip@unine.ch

Abstract. The algorithms belonging to the Ant Colony Optimization metaheuristic can be naturally applied to shortest path related problems. Although not so intensively studied as TSP, the multiple traveling salesman problem (multiple-TSP) is a straightforward extension of TSP of practical importance, requiring more than one salesman to be used for covering the whole set of cities. This work tackles the MinMax formulation of multiple-TSP, which aims at obtaining balanced subtours for the salesmen. An Ant Colony System that follows the MinMax formulation is proposed. Two other algorithms combining K-Means and Fuzzy C-Means clustering with Ant Colony Systems are described. The experimental analysis investigates the performance of the proposed algorithms from a bi-objective perspective, counting for the total length/cost of a solution and its balancing degree measured as the amplitude of its subtours.

Keywords: multiple-TSP, ant colony optimization, fuzzy clustering, hybridization

1 Introduction

The single-depot multiple-TSP problem can be stated as follows. There are m salesmen who must visit a number of cities, n. Each city must be visited exactly once by a salesman and each salesman must start and end its subtour at the same depot. The objective is to minimize the total distance traveled by the salesmen.

Derived from the well-known TSP problem, multiple-TSP is more difficult to solve, since in addition it requires to determine the optimal allocation of cities to the subtours of salesmen such that each city is visited only once by one salesman and the total cost of the traveled subtours is minimized. Multiple-TSP gained less interest in the literature than TSP, although it is more suited for real-world problems related to scheduling and routing. In [1], a comprehensive review of applications for multiple-TSP is presented, ranging from school bus routing problem to mission planning, which arises in the case of autonomous mobile robots.

Since multiple-TSP is a NP-hard problem, which is difficult to solve, exact methods cannot be successfully applied for solving large size instances of multiple-TSP in a reasonable time. Numerous approaches were developed in literature based on deterministic or probabilistic heuristics [1]. Encouraged by their initial success when applied to the standard TSP, metaheuristics like ant colony optimization were proposed in the literature for solving multiple-TSP. Such a first method is introduced in [2], where the authors propose an Ant System to solve the multiple-TSP with ability constraints, in which the number of cities visited by each salesman is limited.

The criterion of balancing the workloads amongst salesmen, desired in real-world scenarios, but less addressed in the literature, is considered in [3]. A tabu search heuristic as well as two exact algorithms are the approaches proposed for solving this variant of multiple-TSP. Besides these methods, other approaches like adaptive neural networks [4] or ant colony [5] were proposed in literature. In [5] an ant colony system algorithm is developed, where a team of ants construct in parallel solution to the problem. The member of a team that will make a move is chosen according to the route lengths, most of the time being the one with the minimum route length. The main idea is to move always the member of a team with the minimum route length to guarantee approximately even route lengths.

There are also papers that considers both objectives, minimizing the total distance traveled by all the salesmen, respectively minimizing the maximum distance traveled by a salesman, and supply results for both of them. A genetic algorithm is proposed in [6], where the authors present a new chromosome representation that works with the classical genetic operators for the TSP. The representation dramatically reduces the number of redundant solutions. In [7] two new metaheuristics based on artificial bee colony and invasive weed optimization are proposed, whilst in [8] a market-based solution is employed, where the problem is of assigning mobile agents to tasks. In [9] the two objectives are treated independently. The approach is based on a hybrid Max-Min Ant System algorithm, which is combined with a local improvement procedure.

A good balance of workloads among salesmen is also obtained by using clustering algorithms. In [10] a clustering algorithm which minimizes the variation of distances traveled within each cluster is proposed. A variety of evolutionary computation algorithms and paradigms for the euclidean multiple TSP are compared in [11]. The first level of optimization namely, the optimal subdivision of cities into groups is considered. The neighborhood attractor schema which is a variation of k-means influences the chromosome representation.

In this study we consider the variant of multiple-TSP, denoted as MinMax multiple-TSP, in which we aim to minimize the longest subtour of a salesman, such that to achieve fairness among salesmen. This formulation leads to obtaining balanced subtours, in which the workload among salesmen is evenly distributed. This paper is organized as follows. In Section 2 an integer linear programming formulation for the MinMax multiple-TSP is presented, that can be used for solving the problem with exact methods. Section 3 summarizes the Ant Colony System, as the underlying algorithm for the proposed methods that

will be described in Section 4. Section 5 presents the conducted experiments and the obtained results, whilst the final remarks will be concluded in Section 6.

2 The single-depot multiple traveling salesman problem

Several integer linear programming formulations for the multiple-TSP were proposed in the literature. A review of them is presented in [1]. We will refer only to the formulation for the MinMax variant of the multiple-TSP.

The multiple-TSP can be defined over a directed graph $G = (V, A)$, with V being the set of nodes and A the set of arcs, and the graph has associated a cost matrix $C = (c_{ij})$ for each arc $(i, j) \in A$. The problem is to find the optimal assignment of cities to the salesmen, such that each salesman starts and ends its subtour from the depot and the longest subtour of a salesman is minimized. Initially, all salesmen are located at the depot, considered to be the first city.

The integer linear programming formulation for MinMax multiple-TSP with n cities and m salesmen, used when solving the model with dedicated software is the following:

$$\min \quad T \tag{1}$$

$$\text{s.t.} \quad \sum_{j=2}^{n} x_{1jk} = 1,\ k = 1, .., m \tag{2}$$

$$\sum_{j=2}^{n} x_{j1k} = 1,\ k = 1, .., m \tag{3}$$

$$\sum_{i=1}^{n} \sum_{k=1}^{m} x_{ijk} = 1,\ j = 2, .., n,\ i \neq j \tag{4}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{m} x_{ijk} = 1,\ i = 2, .., n,\ i \neq j \tag{5}$$

$$\sum_{i=1}^{n} x_{ijk} = \sum_{i=1}^{n} x_{jik},\ j = 2, .., n,\ k = 1, .., m,\ i \neq j \tag{6}$$

$$u_i - u_j + (n - m) \cdot \sum_{k=1}^{m} x_{ijk} \leq n - m - 1,\ 2 \leq i \neq j \leq n \tag{7}$$

$$\sum_{(i,j) \in A} c_{ij} x_{ijk} \leq T,\ k = 1, .., m \tag{8}$$

$$x_{ijk} \in \{0, 1\},\ \forall (i, j) \in A,\ k = 1, .., m. \tag{9}$$

where x_{ijk} is a binary variable that is equal to 1 if the arc (i, j) is used in the optimal solution by the salesman k and 0 otherwise. u_i denotes the number of nodes visited on that salesman's path from the origin to node i, for any salesman, i.e. the position of node i in a subtour.

Constraints (2) and (3) ensure that each salesman leaves, and respectively returns to the depot city. Constraint (4) denotes that in each city, except the depot, we enter only once, whilst constraint (5) denotes that from each city, except the depot, we exit only once. Constraint (6) indicates that in order to form subtours, the salesman that enters the city j, must exit from it, for each j. Constraint (7) corresponds to the subtour elimination constraint, whilst constraint (8) reflects that variable T to be higher than any subtour. Thus T represents the longest subtour, that should be minimized according to the objective function from (1).

3 The Ant Colony System

The Ant Colony Optimization metaheuristic belongs to the class of swarm intelligence methods and draws its inspiration from the way real ants succeed in finding the paths between their nest and food sources. This is achieved by ants laying down pheromone trails throughout their path, used as a form of indirect communication. The variation in the amount of pheromone deposited in the paths is an indicator of the quality of the route selected to reach the food. Although initially designed to solve shortest-path problems in graphs, the application of ant colony algorithms is not restricted to this class of problems; they have been applied successfully to other various combinatorial optimization problems as well. The proposed algorithms described in the next section are based on the Ant Colony System (ACS) [12], initially designed for solving the TSP.

In ACS, several ants are placed in the nodes of the graph representing the TSP instance. At each iteration, each ant builds a tour in the graph; on each edge it traverses, the ant lays a constant quantity of pheromone. At the end of each iteration, after all ants built their tours, the best solution recorded so far, called global solution, is updated (if necessary). Then, the edges on the path given by the global solution receive an extra quantity of pheromone which is inverse proportional with the cost of the global solution. Three important phases are thus involved in this optimization process: node selection at route construction, local pheromone update performed each time an ant traverses an edge and global pheromone update that highlights the route with the minimum cost identified so far.

4 Algorithms investigated for multiple-TSP

4.1 Problem decomposition with k-Means followed by ACS for TSP (kM-ACS)

A first algorithm that resorts to clustering for solving the multiple-TSP is $kM-ACS$ and it was introduced in a previous work of the authors [13]. The main idea behind this approach is to first divide the initial problem into several groups of cities by using K-Means clustering algorithm, then solve each formed subgroup as an individual TSP problem with the Ant Colony System. In this way the

original multiple-TSP instance is transformed into several smaller subproblems, that represent disjoint subset of cities to be visited in a certain order by the m salesmen. The two coordinates in the cartesian plan that each city has, serves as two numerical attributes to be used when performing the cluster analysis. The K-Means algorithm was chosen due to the fact that it is one of the most known clustering algorithms and it generates groups of equal volumes. This characteristic is desirable since we aim to obtain subtours of approximately equal cost. The final solution to the initial multiple-TSP instance is obtained by aggregating the solutions found during the m independently runs of the ACS, which will represent subtours to be assigned to the m salesmen.

4.2 Fuzzy C-Means combined with ACS with global-solution pheromone update (fuzzy g-ACS)

Another clustering based algorithm uses a global-solution pheromone update and it incorporates the fuzzy C-Means algorithm. Like the previous algorithm, it employs the similar idea of using clustering for the grouping of cities.

Fuzzy C-Means clustering algorithm does not compute disjoint partitions of data points - as in the case of k-means algorithm - but rather a degree of flexibility is introduced, by allowing an observation to belong to more than one group with a certain probability. The output of the fuzzy C-Means is a matrix of membership levels (i.e. probabilities), giving for each data point the probability of belonging to every cluster. In the case of our studied problem, the number of clusters is equal to the number of salesmen from the multiple-TSP instance. Given the cost matrix encoding the distances between each pair of cities from the multiple-TSP instance, each city can be seen as an item in the data set, having two numerical attributes corresponding to its two coordinates.

Since the depot city need to be included in the subtour of each salesman, the Fuzzy C-Means algorithm need to be adapted so as to take into account the position of the depot city. To this end, a change was done when computing the position of the centroid for each partition, such that all the centroids will be biased towards the location of the depot city.

The corresponding nearest neighbour subtours, needed in order to compute the value for the initial pheromone level, are constructed by using the crisp partitions obtained after running Fuzzy C-Means. To get these crisp partitions each element is assigned to the cluster for which it has the highest membership level, so that each element will belong to only one partition. Therefore, each subtour is constructed within the cities belonging to the same partition and by always choosing next the closest unvisited city. For each ant, initially all the salesmen are positioned in the depot city. Then for each possible partition corresponding to a salesman, a probability is computed according to the crisp clusters obtained at a precedent step. For the k salesman, this probability is calculated as the division between the number of cities assigned to the k crisp cluster and the total number of cities from the multiple-TSP instance. This probability will be used when deciding which salesman should perform the next move, such that each salesman will be chosen in a probabilistic way by some type

of roulette wheel selection. The rationale behind this selection mechanism is to reflect for each salesman the number of expected cities that it should visit. In this way, a salesman with a greater number of assigned cities will have a higher probability of being chosen. Subsequently, when the selected salesman decides which node to traverse next based on its current position, to the standard ACS transition rule it is added the membership level obtained after running Fuzzy C-Means algorithm. More exactly, in the subtour construction process, the next node to be visited according to the pseudo-random-proportional rule is given by the following equation:

$$s = \begin{cases} \arg\max_{s \in C} \tau(r,s) \cdot \eta^{\beta}(r,s) \cdot mem^{\beta}(s,k), & \text{if } rand(0,1) < q_0 \\ S, & \text{otherwise} \end{cases}$$

where q_0 is a parameter and $s \in C$ denotes that node s is from the candidate set C, that consists of nodes not visited yet. $mem(s,k)$ expresses the membership level of the city s to the k cluster corresponding to the subtour of the k salesman which is under construction, and S is a random variable with the probability distribution given by equation:

$$p(r,s) = \frac{\tau(r,s) \cdot \eta^{\beta}(r,s) \cdot mem^{\beta}(s,k)}{\sum_{u \in C} \tau(r,u) \cdot \eta^{\beta}(r,u) \cdot mem^{\beta}(u,k)} \tag{10}$$

The pheromone level on the traversed connections is updated regardless of the salesman which visited the edge. The construction process of the subtours ends when the candidate set of nodes is empty, meaning there are no left unvisited cities to be chosen in the route selection step. The global best ant, for which the global pheromone update will be applied, is chosen as the one with the minimum value for the total cost of its m subtours. Then the edges (r,s) belonging to the subtours of the globally best ant receives an additional amount of pheromone like in the standard ACS:

$$\tau(r,s) = (1-\rho) \cdot \tau(r,s) + \rho \cdot \Delta\tau(r,s) \tag{11}$$

where

$$\Delta\tau(r,s) = \begin{cases} (L_{gb})^{-1}, & \text{if } (r,s) \in \text{subtours of globally best ant} \\ 0, & \text{otherwise} \end{cases}$$

with ρ being the pheromone evaporation rate and L_{gb} being the total cost of the global solution. In the experimental section this algorithm is denoted as $fuzzy\ g-ACS$.

4.3 Fuzzy C-Means combined with MinMax ACS with global-solution pheromone update (fuzzy g-MinMaxACS)

This ACS based algorithm, denoted as $fuzzy\ g-MinMaxACS$, is similar with the prior described one, using as well Fuzzy C-Means clustering, but differs in

the criterion used for deciding the best ant. More exactly, the ant with the smallest cost for its longest subtour, which corresponds to the MinMax criterion, is considered as the global best ant when performing the global pheromone update. Therefore, during the global pheromone update, all the edges visited by the global best ant will receive the same deposit of pheromone, equal to the length of the longest subtour from the global best solution.

4.4 MinMax ACS with global-solution pheromone update (g-MinMaxACS)

Another investigated algorithm relying on ACS, uses a global-solution pheromone update, along with considering the MinMax criterion for achieving balanced subtours. Initially, for a problem instance with m salesmen, m salesmen are placed at the depot city. Then at each iteration, several ants construct solutions for the multiple-TSP instance, by employing a team of m ants. In this way, the solution built by an ant will be composed of m individual subtours, one corresponding to each salesman.

Since a city should be included only in the subtour of one salesman, several salesmen compete towards building its subtour in an iteration of the algorithm. To this end, at each step, the next salesman that will visit a city is chosen randomly from the set of m available salesmen. After being chosen, the salesman is replaced in the set of possible salesmen so that next time it can get another chance of being selected (i.e. selection with replacement). The selected salesman chooses the city to visit in its subtour according to the pseudo-random-proportional rule from the standard ACS algorithm. The next city is selected among the nodes from the candidate set, comprising of nodes not visited yet in any of the m subtours under construction. This process continues until there are no remaining unvisited nodes, meaning that a complete candidate solution consisting of m subtours was constructed. During the local pheromone update, the pheromone level on the visited edges is updated regardless of which salesman traverses it.

For computing the value of the initial pheromone level, the equivalent nearest neighbour tour for multiple-TSP is constructed by randomly choosing the salesman that will perform the next move. Then the city to be visited by the salesman is selected by a greedy choice of the closest unvisited node, relative to the current node. After all the ants finish to construct a complete solution, a global pheromone update takes place, reinforcing with additional pheromone the connections that belong to the global best ant. The amount of deposited pheromone is given by the cost of the longest subtour of the global best solution. Among the ants from the current iteration, the best iteration ant is chosen as being the one with the smallest cost for its longest subtour. Based on the best iteration ant, the global best so far ant is updated if necessary.

This version of ACS algorithm will be referred in the experimental section as $g - MinMaxACS$.

5 Experiments

5.1 Problem instances

Since there are no publicly available benchmarks for multiple-TSP, as it is TSPLIB [3] for TSP, we had to construct our own set of multiple-TSP instances. The experimental analysis in this paper is conducted on 8 problem instances which were created based on 2 TSPLIB instances: for each TSPLIB instance we specified 4 different values for the number of salesmen. The 8 multiple-TSP instances, were solved with exact methods by using CPLEX [4]. The description of the multiple-TSP instances [5] and the obtained results are publicly available as multiple-TSPLIB.[6].

As mentioned in Section 2, where we introduced the linear programming formulation for the MinMax variant of multiple-TSP, the objective is to minimize the cost of the longest subtour. This is different from the formulation of the standard multiple-TSP, which aims to minimize the total cost of the traveled subtours. The formulation we considered for the multiple-TSP is necessary such as to obtain balanced subtours as much as possible. In fact, when no such constraint is imposed, the solutions obtained by CPLEX are highly unbalanced: most of the cities are visited by a salesman, while the rest of the salesmen visit only a small fraction of cities. This corresponds to an uneven distribution of work among the salesmen, which is undesirable in real-world scenarios. At the same time, the MinMax version of multiple-TSP is more difficult to solve by CPLEX than the standard multiple-TSP. For the eil51-m5 problem instance, in case of the standard multiple-TSP formulation, CPLEX found an optimal solution in at most 1 minute, while in case of the MinMax multiple-TSP formulation the reported solution was obtained with CPLEX after 120 hours. In comparison, one run of any algorithm presented in this paper required on average 10 seconds in a single-threaded implementation. All the runs were performed on a PC with 8 GB RAM, processor Intel Core i5-4590S 3.00 GHz using 4 processing units (multi-threaded implementation of CPLEX). However our purpose is not to compare our investigated algorithms with the CPLEX solver, since for the execution of the ACS algorithms we set as limit a maximum number of iterations, whilst in the case of the CPLEX solver we set a high time limit as stopping criterion. In the experimental section of our paper we report the solutions obtained by CPLEX, just to have an overview of the quality of solutions found by our proposed approaches.

5.2 Parameters settings

For the parameters of the ACS algorithm we used the recommended values for the standard ACS [12], more specifically we used $q_0 = 0.9$, $\alpha = 0.1$, $\rho = 0.1$,

[3] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
[4] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
[5] www.infoiasi.ro/~mtsplib
[6] www.infoiasi.ro/~mtsplib/MinMaxMTSP

$\beta = 2.0$ in all the algorithms presented in Section 4. For the $kM - ACS$, the number of used ants was 10, for each of the m TSP subproblems being solved, leading to a total of $10 \cdot m$ ants employed for solving the initial multiple-TSP instance. For all the other algorithms we used the same number of $10 \cdot m$ ants as follows: 10 groups of ants of size m build independently in a single run of the algorithm complete solutions to the multiple-TSP.

As a stopping criterion for the ACS based algorithms we set a maximum number of iterations. More exactly, each algorithm was run for 1400 iterations in case of eil51 and 1800 iterations for eil76 problem instances. For the conducted experiments we performed 50 runs for each investigated algorithm.

K-Means is an iterative method, which aims to minimize the within-cluster variance and the obtained clusters are highly dependent on the initialization step. Thus, we performed 30 runs of K-Means on the same multiple-TSP instance to split the n cities into m groups and selected the solution which provided the lowest sum of the within-cluster variances. However, in the case of fuzzy clustering algorithms, we didn't impose a certain grouping of cities to be used in all the runs, but instead different membership probabilities were used in the conducted runs, such that to exploit the random nature of Fuzzy C-Means.

5.3 Results

For assessing the performance of the investigated algorithms, we report their obtained values for the total cost of the traveled subtours and for the cost of the longest subtour. Though we report the obtained results for both objectives, the main focus is on the MinMax criterion. We specify the values for the classic objective (i.e. total cost) only for a better comparison. As a measure for the balancing degree of the subtours, we used amplitude, computed as the difference between the cost of the longest subtour and the cost of the shortest subtour of the solution.

Comparing fuzzy g-ACS and fuzzy g-MinMaxACS Although both $fuzzy\ g-ACS$ and $fuzzy\ g-MinMaxACS$ algorithms resort to clustering, they exhibit different behaviour according to the multiple-TSP objectives: total cost and the balancing of subtours. For Figure 1, we post-processed the 50 solutions obtained by the two fuzzy clustering based algorithms and extracted only the Pareto non-dominated solutions according to the two objectives. For the considered multiple-TSP instances we illustrate the values for the two objectives: the values on the horizontal axis corresponds to the total cost of the traveled subtours, whilst the values on the vertical axis corresponds to the amplitude of the traveled subtours. It can be noticed that for both eil51 and eil76 instances, $fuzzy\ g-ACS$ obtains lower values for the total cost, but that are highly unbalanced, whereas, on the other hand, $fuzzy\ g-MinMaxACS$ obtains good balanced solutions but of higher total costs. This observation actually indicate the fact that these two objectives, of minimizing the total cost of the traveled subtours and of minimizing the cost of the longest subtour are conflicting ones: improving

one objective, degrades the obtained values for the other objective. This leads to multiple-TSP being an inherently bi-criteria problem. As a matter of fact, our future studies will be focused on tackling the multiple-TSP from a bi-criteria perspective. Since $fuzzy\ g-ACS$ obtained worse results regarding the MinMax criterion, in the following we will only refer to $fuzzy\ g-MinMaxACS$ as a fuzzy clustering based algorithm.

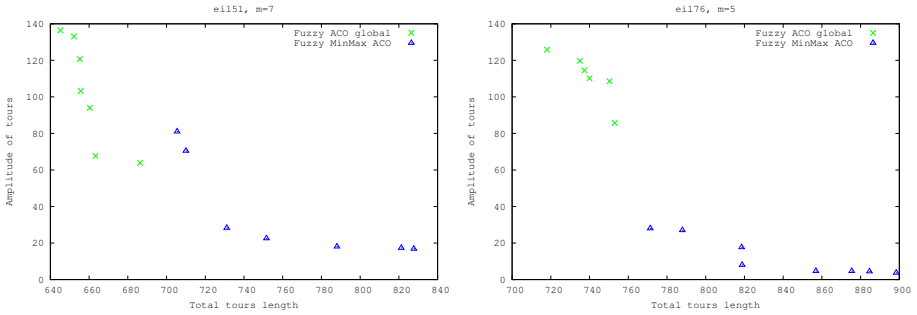

Fig. 1: A comparison between the fuzzy ACS global and fuzzy MinMax ACS algorithms for eil51-m7 and eil76-m5 instances

From our experiments it resulted that in most of the multiple-TSP instances, $fuzzy\ g-ACS$ obtains better values for the total cost of the traveled subtours, whilst $fuzzy\ g-MinMaxACS$ succeeds in attaining good values for the amplitude of the subtours, achieving balanced subtours. This fact can be explained by their different mechanism of selecting the best ant and enforcing the global best so far solution. While in $fuzzy\ g-ACS$ the best ant is selected to be the one with the minimum total cost of the traveled subtours, in $fuzzy\ g-MinMaxACS$ the global best solution and the additional pheromone deposit take into account the cost of the longest subtour.

Algorithms comparison Table 1 reports the optimal values for the longest subtour of minimal cost (MinMax) obtained by CPLEX, along with the values obtained by our investigated algorithms. For the ACS based algorithms the MinMax value represents the minimum value obtained for the longest subtour from the found solutions. When CPLEX was stopped before reaching an optimal solution, the upper (corresponding to the best known solution) and the lower bound on the cost of the longest subtour are given. The best value from a row is highlighted with boldface. From the analysis of these values, among all the ACS algorithms, $g-MinMaxACS$ achieves the best values for the longest subtour, corresponding to the MinMax criterion. In case of eil76-m5 multiple-TSP instance it even gets a better value for MinMax than the best known solution found by CPLEX.

Figures 2 to 5 depicts for eil51 and eil76 multiple-TSP instances the distribution for the total costs and for the cost of the longest subtour, and confidence intervals for the mean of the total costs and for the mean of the longest subtour, for the solutions obtained by each algorithm. On each plot there are four groups corresponding to the different number of m (2,3,5 and 7). It can be noticed that with the increase in the number of salesmen, the total cost increases, whilst the length of longest subtour decreases, reflecting the division of work among more salesmen. For each of these four groups there are indicated the obtained values for each of the three considered algorithms ($kM-ACS$, $g-MinMaxACS$ and $fuzzy\ g-MinMaxACS$) together with the best known solution, obtained with CPLEX, that serves as lower bound for the ACS-based algorithms.

Analyzing the boxplots, but also also from the size of the confidence intervals, $kM-ACS$ is the most stable algorithm. However, it does not achieve such good values as $g-MinMaxACS$ and $fuzzy\ g-MinMaxACS$ for the length of the longest subtour. Figure 2 that illustrates the boxplots for the total cost, indicates the low values obtained for the standard deviation in case of K-Means, such that the obtained solutions are not so diverse. Figure 5 shows that between the two clustering algorithms, $fuzzy\ g-MinMaxACS$ obtains better values than $kM-ACS$ regarding the cost of the longest subtour. Although both these algorithms attempts to create groups of cities of equal volumes, it seems that the flexible nature of Fuzzy C-Means helps in that sense. In contrast to this, $kM-ACS$ seems to loss diversity when a higher number of clusters is considered, proven by the overall lower values obtained for the standard deviation of the total cost.

A comparison of the investigated algorithms on eil51-m5 and eil76-m7 instances is illustrated in Figure 6, that plots, like in Figure 1, the Pareto non-dominated solutions according to the two objectives. It can be noticed the good values obtained by $g-MinMaxACS$ and $fuzzy\ g-MinMaxACS$ for the amplitude, taken as a measure for the balancing degree of the subtours. In contrast to these two algorithms, there are fewer solutions corresponding to $kM-ACS$, which are highly unbalanced. On most of the multiple-TSP instances, $g-MinMaxACS$ achieves better performance both in terms of total cost and MinMax criterion when compared to $fuzzy\ g-MinMaxACS$.

6 Conclusions

Ant colony algorithms can be easily adapted and applied to shortest path related problems as multiple-TSP. Unlike exact methods, ACO based approaches can obtain acceptable solutions, both in matter of time and performance, which makes them more suited for real-world applications. In this paper, we tackled the MinMax variant of multiple-TSP, which aims to minimize the cost of the longest subtour. To this end, we proposed four ACS based algorithms, three of them employing clustering methods for the division of cities into equal groups, and the other one following a MinMax approach. Between the two versions of fuzzy clustering algorithms, the one that takes into account the MinMax cri-

Table 1: The performance of the ACS variants for the eil51 and eil76 instances using MinMax as measure

m	CPLEX optimum	kM-ACS	g-MinMaxACS	fuzzy g-MinMaxACS
		eil51		
2	222.73	254.42	**226.54**	231.05
3	[150.70, 159.57]	176.16	**164.64**	167.06
5	[96.91, 123.96]	169.03	**127.86**	133.12
7	[72.42, 112.46]	129.80	**116.05**	118.26
		eil76		
2	280.85	301.74	**288.96**	290.37
3	[186.34, 197.34]	266.92	**214.92**	221.05
5	[116.02, 173.18]	176.31	**160.49**	167.24
7	[88.35, 139.62]	158.03	**145.28**	150.37

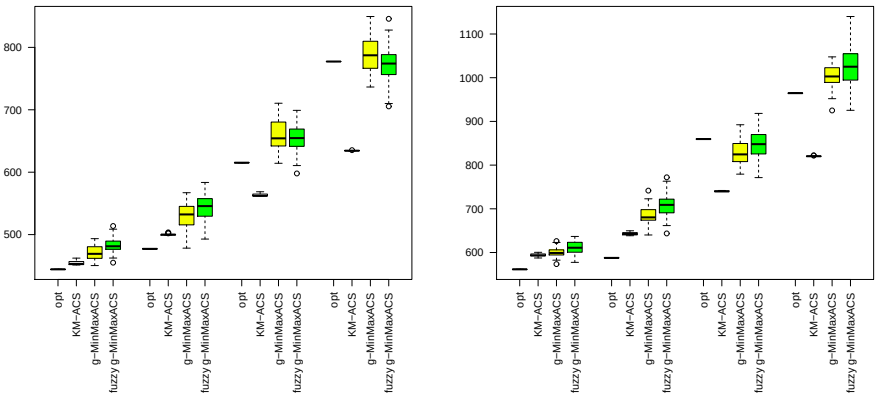

Fig. 2: Results obtained in 50 runs for the total cost: a) eil51, b) eil76; the groups correspond to different settings for m: 2, 3, 5, 7

terion achieves better values for the cost of the longest subtour. The approach involving K-Means is the most stable algorithm, but due to its decomposition scheme, it imposes rigid boundaries on the candidate solution space and the obtained solutions are not so balanced. Among all the investigated algorithms, $g-MinMaxACS$, which follows a MinMax approach, achieves the best performance, succeeding in finding balanced subtours. As future work, our study will be directed towards applying multi-objective ACO algorithms for the bi-criteria multiple-TSP and evaluate their relative performance.

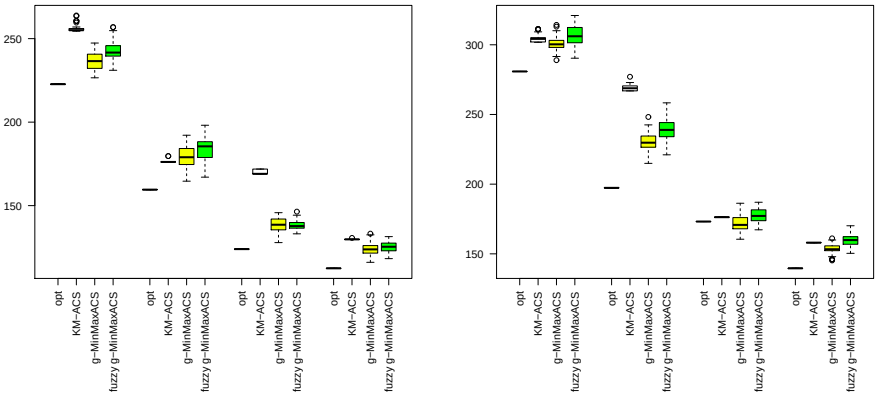

Fig. 3: Results obtained in 50 runs for the longest subtour: a) eil51, b) eil76; the groups correspond to different settings for m: 2, 3, 5, 7

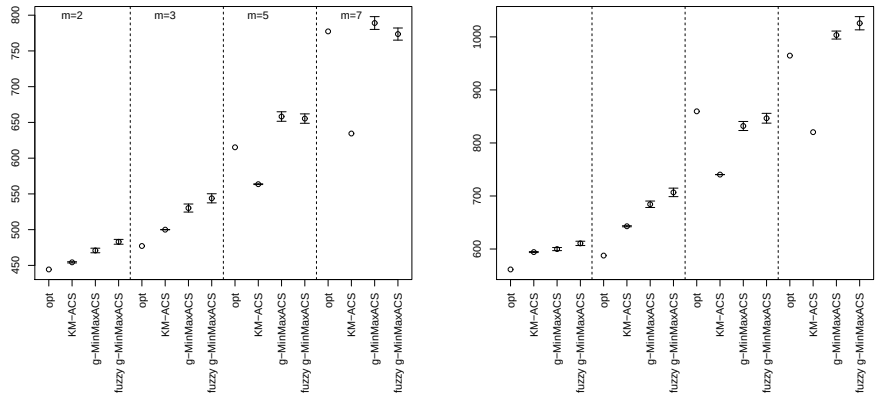

Fig. 4: Results obtained in 50 runs for the a) eil51, b) eil76: confidence intervals for the means computed for total costs; the groups correspond to different settings for m: 2, 3, 5, 7

References

1. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega, 34, (3), 209-219 (2006)
2. Junjie, P., Dingwei, W.: An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem. ICICIC 2006, vol. 1, 210-213 (2006)
3. França, P.M., Gendreau, M., Laporte, G., Müller, F.M.: The m-traveling salesman problem with minmax objective. Transportation Science, 29(3), 267-275 (1995)

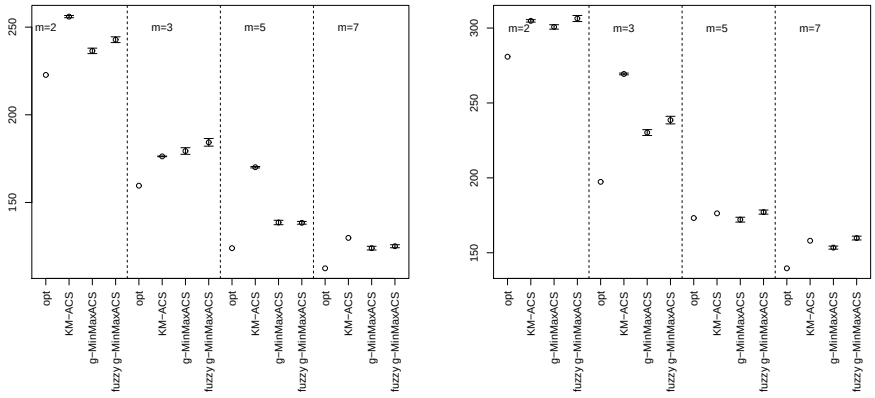

Fig. 5: Results obtained in 50 runs for the a) eil51, b) eil76: confidence intervals for the means computed for the longest subtour; the groups correspond to different settings for m: 2, 3, 5, 7

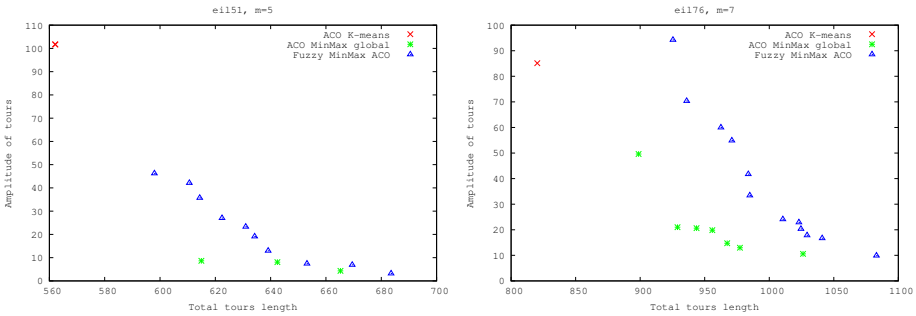

Fig. 6: A comparison between the investigated algorithms for eil51-m5 and eil76-m7 instances

4. Somhom, S., Modares, A., Enkawa, T.: Competition-based neural network for the multiple travelling salesmen problem with minmax objective. Computers & Operations Research, 26(4), 395-407 (1999)
5. Vallivaara, I.: A team ant colony optimization algorithm for the multiple travelling salesmen problem with minmax objective. MIC '08, 387-392 (2008)
6. Carter, A.E., Ragsdale, C.T.: A new approach to solving the multiple traveling salesperson problem using genetic algorithms. EJOR, vol. 175(1), 246-257 (2006)
7. Venkatesh, P., Singh, A.: Two metaheuristic approaches for the multiple traveling salesperson problem. Applied Soft Computing, 26, 74-89 (2015)
8. Kivelevitch, E., Cohen, K., Kumar, M.: A Market-based Solution to the Multiple Traveling Salesmen Problem. JIRS journal, vol. 72(1), 21-40 (2013)
9. Liu, W., Li, S., Zhao, F., Zheng, A.: An ant colony optimization algorithm for the Multiple Traveling Salesmen Problem. ICIEA 2009, 1533-1537 (2009)

10. N. Chandran, N., Narendran, T.T., Ganesh, K.: A clustering approach to solve the multiple travelling salesmen problem, International Journal of Industrial and Systems Engineering, 1(3), 372-387 (2006)
11. Sofge, D., Schultz, A., De Jong, K.: Evolutionary computational approaches to solving the multiple traveling salesman problem using a neighborhood attractor schema, Applications of Evolutionary Computing, 153-162 (2002)
12. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, vol. 1(1), 53-66 (1997)
13. Necula, R., Breaban, M., Raschip, M., Performance evaluation of ant colony systems for the single-depot multiple traveling salesman problem. HAIS 2015, vol. 9121, 257-268 (2015)