Soft Comput
DOI 10.1007/s00500-016-2279-7

METHODOLOGIES AND APPLICATION

FL-MTSP: a fuzzy logic approach to solve the multi-objective multiple traveling salesman problem for multi-robot systems

Sahar Trigui[1,2] · **Omar Cheikhrouhou**[3,4] · **Anis Koubaa**[5,6] · **Uthman Baroudi**[7] · **Habib Youssef**[8]

Abstract This paper considers the problem of assigning target locations to be visited by mobile robots. We formulate the problem as a multiple-depot multiple traveling salesman problem (MD-MTSP), an NP-Hard problem instance of the MTSP. In contrast to most previous works, we seek to optimize multiple performance criteria, namely the maximum traveled distance and the total traveled distance, simultaneously. To address this problem, we propose, FL-MTSP, a new fuzzy logic approach that combines both metrics into a single fuzzy metric, reducing the problem to a single-objective optimization problem. Extensive simulations show that the proposed fuzzy logic approach outperforms an existing centralized Genetic Algorithm (MDMTSP_GA) in terms of providing a good trade-off of the two performance metrics of interest. In addition, the execution time of FL-MTSP was shown to be always faster than that of the MDMTSP_GA approach, with a ratio of 89 %.

Keywords MD-MTSP · Fuzzy logic · Optimization problem · Multi-objective

Communicated by V. Loia.

✉ Sahar Trigui
sahar.trigui@coins-lab.org

Omar Cheikhrouhou
o.cheikhrouhou@tu.edu.sa

Anis Koubaa
aska@isep.ipp.pt

Uthman Baroudi
ubaroudi@kfupm.edu.sa

Habib Youssef
Habib.youssef@fsm.rnu.tn

1 University of Manouba, Manouba, Tunisia

2 Cooperative Intelligent Networked Systems (COINS) Research Group, Riyadh, Saudi Arabia

3 Taif University, Taif, Kingdom of Saudi Arabia

4 Computer and Embedded Systems Laboratory, University of Sfax, Sfax, Tunisia

5 Prince Sultan University, Riyadh, Saudi Arabia

6 CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal

7 Wireless Sensors and Robotics Laboratory, Computer Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

8 PRINCE Research Unit, University of Sousse, Sousse, Tunisia

1 Introduction

In complex robotics applications, such as Trigui et al. (2012), Koubâa et al. (2012), Khamis et al. (2011), Pippin et al. (2013), and Fazli et al. (2010), robots typically need to collaborate together in order to complete their mission efficiently. In fact, cooperative robots systems represent a recommended alternative to single-robot systems for a vast array of applications, considering the collaborative effect between robots that leads to accomplishing their missions more efficiently. The multi-robot task allocation problem (MRTA) deals with assigning tasks to robots to perform collaborative missions. The MRTA problem can be formulated as follows: given n robots and m tasks, the objective consists of ensuring an efficient assignment of the tasks under consideration in order to minimize the overall system cost. In the literature, several works have proposed different solutions to the MRTA problem and applied it in different contexts. A formal analysis and taxonomy of multi-robot task allocation problems in several fields is given in Trigui et al. (2014). The paper defines three main categories: (1) single-task robots (ST) versus multi-task robots (MT), (2) single-robot tasks (SR) versus multi-robot

Published online: 09 August 2016

tasks (MR), and (3) instantaneous assignment (IA) versus time-extended assignment (TA). According to this taxonomy, our proposed approach belongs to the category of (ST, SR, TA).

In the literature, existing approaches can be divided into two categories, namely: (1) the centralized approach: it assumes the knowledge of global information by a central agent (e.g., control station), which is able to calculate a near-optimal solution to the allocation problem, (2) the distributed approach: decisions (or local solutions) are based on local information for each agent performing the task (e.g., robot).

In this paper, we consider a disaster situation where n robots are to be assigned $m > n$ target locations. The objective of this application is to use the robots to collect sensors data and take live images from locations impacted by the disaster, to help rescuers take appropriate actions in real time. In its abstract form, the problem can be considered as an instance of MTSP problem. In particular, we model the problem as multiple-depot multiple traveling salesman problem (MD-MTSP) where each robot, initially at a certain depot location, needs to make a tour over certain location of interest, then returns to its initial locations, while minimizing the mission cost. We seek to find a near-optimal assignment of the target locations to robots. Most of the previous related works (Cheikhrouhou et al. 2014; Wang et al. 2013; Sariel et al. 2007; Singh and Baghel 2009; Brown et al. 2007; Carter and Ragsdale 2006; Liu et al. 2009; Yousefikhoshbakht et al. 2013) considered a mission cost equal to a particular single metric, most typically the total traveled distance (*TTD*) or the maximum tour length (MT). In this paper, we consider the mission cost as a combination of the *TTD* and the MT metrics. Several approaches have been proposed to deal with multi-objective combinatorial optimization problems. A first approach is to classify the objectives on the basis of their importance (Nikolić 2007), then, to optimize with respect to the most important objective, and in case of a tie, with respect to the second objective, and so on. A second approach is to optimize with respect to the most important objective and to transform the rest of the objectives into constraints (Mavrotas 2009; Bérubé et al. 2009). A third approach is to combine all objectives in a single cost function such as a weighted sum (Marler and Arora 2010; Shim et al. 2012b). A fourth approach uses the notion of pareto optimality and seeks to find a pareto-optimal front of solutions, and let the decision maker makes a choice (Deb et al. 2002; Bolaños et al. 2015). In this work, we resort to fuzzy logic to express individual objectives in linguistic terms, and then use fuzzy algebra to combine them into a crisp value that represents the degree of membership of the particular solution into the fuzzy subset of good solutions (with short *TTD* and short MT). We propose a new centralized, yet fast, approach that takes as input the set of robots and their initial depots, and a set of target locations, and produces optimized tour assignment for each robot over a certain number of allocated target locations. The contributions of the paper can be summarized as follows:

– The first contribution lies in the design of a centralized, yet fast, algorithm (FL-MTSP) that uses the fuzzy logic to combine both the *TTD* and the MT metrics. and assigns target locations to robots.
– The second contribution is a thorough performance evaluation of the proposed FL-MTSP algorithm and its comparison with single-objective function optimization algorithms.
– The third contribution is the comparison of the FL-MTSP proposed algorithm with a centralized genetic algorithm solution Kivelevitch (2011) and with NSGA-II Deb et al. (2002).

The rest of this paper is organized as follows. Sect. 2 presents a comprehensive literature review on relevant works on multi-objective MTSP problem and existing approaches to solve it. Sect. 3 describes the problem formulation, system model and assumptions of the MD-MTSP problem considering multiple objectives. Sect. 4 presents the FL-MTSP algorithm, while Sect. 5 provides performance results. Finally, Sect. 6 concludes the paper.

2 Related works

Multi-Robot Task Allocation problem is a challenging research axis in the field of robotics. In the literature, several researchers have considered the MRTA problem as an instance of the multiple traveling salesmen problem due to the strong analogy between them.

In the literature, researchers proposed different solutions to solve the MTSP. In Sariel et al. (2007), the authors presented a distributed algorithm to solve the MTSP. Initially, each robot selects the nearest target. Next, each robot makes an auction for its task and task assignment is performed using the contract net protocol. Simulation results proved the efficiency of the algorithm in terms of scalability, total path length and communication message overhead. In Kivelevitch et al. (2013), the authors proposed a market-based algorithm that consists of four steps: market auction, agent-to-agent trade, agent switch and agent relinquish step. In the first step, each robot takes the best task. In the agent-to-agent trade step, each robot checks its ability to perform any task of the other robots. In the agent switch step, the solutions that are not in the local minima are explored. After a number of iterations with no improvement, the algorithm stops. In Cheikhrouhou et al. (2014), the authors proposed a market-based solution called move and improve to solve the MD-MTSP. The solution consists of four steps: initial target allocation, tour construction, negotiation of conflicting targets and solution improvement. From the simulation study, it was shown that the move and improve algorithm gives good results compared with the results generated by a centralized approach.

Several research works proposed a Grouping Genetic algorithm (GGA) (Carter and Ragsdale 2006; Brown et al. 2007) to solve the MTSP problem. A GGA algorithm is based on dividing the cities into m groups where m is the number of salesmen. In Singh and Baghel (2009), the authors solved the single depot MTSP problem using the Genetic Algorithm. They proposed a steady-state grouping genetic algorithm (GGA-SS) that uses different chromosome representation and genetic operators. Also, they have used steady-state population replacement model. The objectives are: (1) minimizing the total distance traveled by all the salespersons, and (2) minimizing the maximum distance traveled by any salesperson. Simulation results have shown that the GGA-SS finds the solution of least average cost compared with the solutions proposed in Carter and Ragsdale (2006) and Brown et al. (2007). The authors assume every salesperson must visit at least one city in addition to the home city. This restriction can cause an increase of the total traveled distance.

In Liu et al. (2009), the authors solved the MTSP problem using the ant colony optimization (ACO) algorithm. The objectives are to minimize the maximum tour length of all the salesmen and minimize the maximum tour length of each salesman. A comparison between the ACO-based algorithm with the ones proposed in Carter and Ragsdale (2006), Brown et al. (2007) and Singh and Baghel (2009) was performed. Computational results demonstrated that the ACO-based solution outperforms the GA-based solutions.

In Yousefikhoshbakht et al. (2013), a new modified ACO algorithm (NMACO) was proposed to solve the MTSP problem. Modifications include the transition rule, the candidate list, the global pheromone updating rules and several local search techniques. These modifications improve the quality of solution of the ACO. The objective is to minimize the distance traveled by the salesmen. Computational experiments have shown that in general, the NMACO produces better results compared to the existing solution methods for MTSP.

In contrast to several previous works, our objective is to optimize multiple performance criteria namely the total traveled distance, and the maximum tour, which turns our problem as a multi-objective optimization problem (MOO). Generally, the MOO problem is expressed in mathematical terms as follow:

$$\begin{aligned} min F(x) &= [f_1(x), f_2(x), \ldots, f_k(x)] \\ &s.t. x \in S \\ x &= (x_1, x_2, \ldots, x_n)^T \end{aligned} \tag{1}$$

where $f_1(x), f_2(x), \ldots, f_k(x)$ are the k objective functions, $(x_1, x_2, \ldots, x_n)$ are the n optimization parameters, and $S \in R^k$ is the decision variables. Note that the objectives are usually in conflict with each other. Thus, it is impossible to simultaneously improve all objective functions.

Few research works were conducted to solve the MTSP problem as a MOO problem. In Shim et al. (2012a), the authors proposed a solution based on the combination of the estimation of distribution algorithm (EDA) with a gradient search to solve the multi-objective MTSP. They considered an objective function equal to the weighted sum of the total traveling costs of all salesmen and the highest traveling cost of any single salesman. The algorithm includes four steps: chromosome representation, decomposition, modeling and local search. The comparison of the proposed algorithm with several state-of-the-art algorithms demonstrates its effectiveness. Also, from simulation study, it was shown that the hybridization of EDA with a local search algorithm improves the quality of the solution. Nevertheless, the authors did not present results related to the execution time of their algorithm. In Ke et al. (2013), the authors proposed a multi-objective ant colony algorithm using the MOEA/D framework Zhang and Li (2007) (MOEA/D-ACO). The basic idea of this algorithm is to decompose the multi-objective problem into single-objective problems. The ants are decomposed into groups. The number of groups is the same as the number of objectives. Each ant is responsible for finding a solution to the sub problem of its group. The ants of the same group share the same pheromone matrix but each ant has its own heuristic information matrix. Experimental results proved the efficiency of the MOEA/D-ACO solution.

Unlike the solutions proposed in Ke et al. (2013) and Shim et al. (2012a) that consider a decomposition framework of the multi-objective problem, we propose a new solution based on the combination of the objectives. Also, the authors did not specify how they decompose the multi-objective problem into single-objective problems. For the performance evaluation, the authors did not present results related to the objectives (total traveled cost and the highest traveled cost by any salesman).

In Xu et al. (2008), the authors proposed an ACO-based algorithm for solving the task assignment problem for multiple Unmanned Underwater Vehicles. Two objectives were considered: minimizing the total distance of visiting all targets and minimizing the total turning angle. The problem was defined as a multi-objective MTSP taking into account the constraints of balancing the number of targets visited by each vehicle. The solution has two phases. The task number assignment phase consists in defining the number of targets for each vehicle while the second phase solves the MTSP problem using an ant colony for each objective. Performance evaluation shows that the algorithm generates good solution.

In Bolaños et al. (2015), the authors proposed a non-dominated sorting genetic algorithm (NSGA-II) to treat the multi-objective MTSP. They addressed two objectives: minimizing of the total traveled distance and the balance of the working times of the traveling salesmen. The evaluation was performed on two instances ((3 salesmen, 29 nodes) and (3 salesmen, 75 customers)) and the results shows the effectiveness of the NSGA-II in minimizing both objectives. However, the solutions proposed in Bolaños et al. (2015) and Xu et al. (2008) are efficient but only in small scale scenarios. The authors did not prove the efficiency of their solution for large scale scenarios.

In contrast to several research works that decompose the multi-objective problem into single-objective problems, the main contribution of this work is to combine multiple objectives using a fuzzy logic system in order to solve the MTSP problem. In this way, our multi optimization problem becomes a single optimization problem. Also, we are interested to study the MTSP where robots are initially located into different positions in contrast to several works that considers that all salesmen start from the same initial position.

3 Problem formulation

The multiple traveling salesmen problem has received significant attention in a large number of robotic applications. In this work, we address the multi-objective MTSP. We consider a set of robots responsible for following a set of target locations in the case of disaster management. The objective is to find the effective assignment of robots to a set of locations so as to optimize the overall cost and maximize the performance. For a given robot, the task is to visit a set of particular points of interests. The MTSP is an NP-hard problem and can be solved using heuristic approaches. In other words, an optimal task sequence for the robots must be planned firstly to ensure that it takes least cost for the robot to move. In order to solve the problem, the task allocation algorithm has to answer the following two questions:

1. To which target locations a robot should be assigned?
2. If a robot was assigned to more than one target locations, how does it determine which one to visit before the others?

We assume the offline version of the problem where the number of locations to be monitored is known from the beginning.

In this work, our objective is to optimize two main performance criteria namely the total tour length and the maximum tour length.

1. The total traveled distance of the robots on the target: We define *TTD* as the sum of all tours length performed by all the robots. The tour length of each robot is calculated using existing TSP solver once the target locations are allocated. The total tour length is calculated by summing up the traveled distance of all edges included in a tour. We define $tour_{r_i}$ (Eq. 3) as the tour of the robot i starting from and ending at the same position. The *TTD* is given according to Eq. 2.

$$TTD = \sum_{i=1}^{n} tour_{r_i} \tag{2}$$

where

$$\begin{aligned} tour_{r_i} = {} & distance(r_i, t_{i_1}) \\ & + \sum_{j=1}^{k_i-1} distance(t_{i_j}, t_{i_{j+1}}) \\ & + distance(t_{i_{k_i}}, r_i) \end{aligned} \tag{3}$$

where k_i is the number of target locations assigned to robot i. $distance(t_{i_j}, t_{i_{j+1}})$ represents the distance between target location j and target location $j+1$ for robot i. t_{i_1} and $t_{i_{k_i}}$ represent the first and the last target locations respectively for robot i. $distance(r_i, t_{i_1})$ represents the distance to travel from the depot of r_i to the first target t_{i_1} and $distance(t_{i_{k_i}}, r_i)$ represents the distance to return back from the last target $t_{i_{k_i}}$ to the initial depot of r_i.
2. The maximum tour MT measured in terms of distance: it is the maximum distance traveled by any robot after the scheduled mission is completed. The maximum tour length among all the tours of the robots is expressed as follows:

$$\begin{aligned} MT = {} & max(tour_{r_i}) \\ & 1 \leq i \leq n \\ s.t. \ \ & tour_{r_i} \neq tour_{r_j} \\ & 1 \leq j \leq n, \ \ i \neq j \end{aligned} \tag{4}$$

In the context of multi-objective optimization, the objective function of the problem is obtained by combining the objectives to be minimized (*TTD* and MT). For this purpose, we used a fuzzy logic system as described in Sect. 4.1.

4 Proposed solution

4.1 Fuzzy logic rules design

In our work, we propose the use of fuzzy logic (Zadeh 1965, 1975) to combine conflicting objectives. We consider each of the Objectives mentioned in Sect. 3 namely

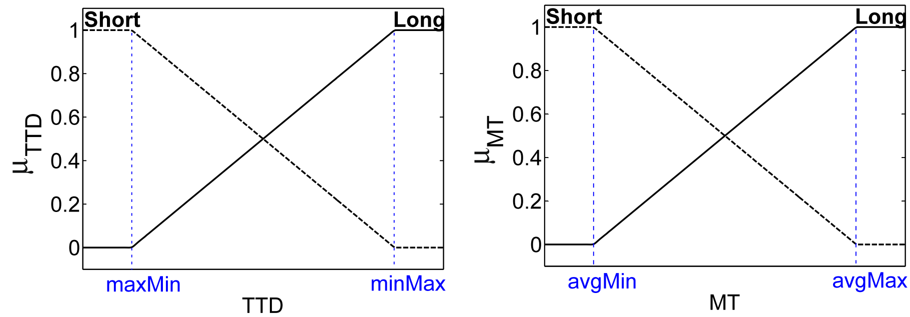

Fig. 1 Membership function of the *TTD* and MT objectives

MT and *TTD*, which are the inputs of the system. The proposed solution attempts to simultaneously minimize the total traveled distance and the maximum traveled distance. We assume that the mission time is proportional to the MT as the velocities of all robots are the same. In addition, we assume that the required time to collect sensor data is the same at all target locations.

We rely on the expressive power of fuzzy logic to state the desired objectives to optimize. Recall that we seek to distribute a number of targets on a number of robots while minimizing the sum of all robots tour lengths as well as the maximum among all robot tour lengths. Hence, we seek solutions with Short-TTD and Short-MT. In fuzzy logic, Short-TTD and Short-MT represent fuzzy linguistic values for the fuzzy variables *TTD* and MT. To each linguistic value corresponds a fuzzy subset with an associated membership function denoted typically by the Greek letter μ. For example, in our case μ_{TTD} gives for each solution with a given *TTD* (the base value) the degree of membership of that solution in the fuzzy subset of solutions with short *TTD*. The desire to seek solutions with simultaneous short *TTD* and short MT can be described by the following fuzzy rule:

IF Solution is Short-TTD **AND** solution is Short-MT **THEN** Good-Solution.

In fuzzy logic, Good-Solution is a linguistic value for the fuzzy variable solution. According to the above fuzzy rule, the membership function (μ_{GS}) of Good-Solution is expressed as follows:

$$\mu_{GS}(s) = min(\mu_{Short-TTD}(s), \mu_{Short-MT}(s))$$

The above expression assumed the min-max logic of Zadeh, where the fuzzy **AND** is interpreted as min and the fuzzy **OR** as a max.

The membership function of each objective is determined by two thresholds. We define the shortest tour of a robot as the tour length obtained by visiting all target location using a greedy algorithm that selects the closest next target to the current position of the robot. We define the longest tour of a robot as the tour length obtained by visiting all target locations using a greedy algorithm that selects the farthest next target to the current position of the robot. We define four variables:

- $maxMin$ is defined as the longest tour among all the shortest tours of all robots going through all target locations considering our definition above.
- $minMax$ is defined as the shortest tour among all the longest tours of all robots going through all target locations considering our definition above.
- $avgMin$: as the average of the shortest tour length of all robots going through all target locations.
- $avgMax$: as the average of the longest tour length of all robots going through all target locations.

The $maxMin$ and $minMax$ are used as thresholds of the *TTD* fuzzy variable, because they represent reasonable upper and lower bounds on the total tour. On the other hand, $avgMin$ and $avgMax$ are used as thresholds of the MT fuzzy variable because they represent reasonable bounds of the maximum tour length (Fig. 1). For example, for μ_{TTD}, for a value of *TTD* below $maxMin$, the total tour cost is considered short compared to the value of *TTD* above $minMax$. The same reasoning holds for μ_{MT}. The thresholds values can be obtained using a simple greedy algorithm or using an existing TSP solver based on genetic algorithm Kirk (2011). In our solution, we have used a greedy algorithm to compute $maxMin$, $minMax$, $avgMin$ and $avgMax$.

In the inference phase, we define the fuzzy rules that allow the combination of our objectives which represent the inputs of our system. In this paper, we use the Mamdani Fuzzy model (Mamdani and Assilian 1975) to represent the "if-then" rules (Table 1). After the establishment of the rules, we move to the defuzzification step which computes the output value. In our work, we use the simple and the most commonly used method in the literature, which is the centroid defuzzification method (Takagi and Sugeno 1985). The output value is calculated by the following equation:

$$CrispOutput = \frac{\sum_{i=1}^{N} W_i * \mu_A(W_i)}{\sum_{i=1}^{N} \mu_A(W_i)} \quad (5)$$

Table 1 Fuzzy rules base

TTD	MT	Assignment
Short	Short	Good
Long	Long	Bad
Short	Long	Bad
Long	Short	Bad

Algorithm 1. The FL-MTSP Algorithm

Inputs:
nbTargets: number of targets
nbRobots: number of robots
matrixDistances: matrix of distance between robots and targets
Assignment phase
tour construction phase
Outputs:
tour: tour of each robot
tourCost: tour cost of each robot
MT: max tour cost among all tours cost

where N is the number of rules (in our case N = 4), W_i is the input value and $\mu_A(W_i)$ is the membership function of rule i.

4.2 Algorithm design

In general, the goal of solving the MTSP is to find an optimal order to pass through all locations in order to minimize the total traveled distance. For the case where several objectives must be optimized, the goal is to find trade-off solutions while optimizing multiple performance criteria. The inputs of our solution are the number of robots, the number of target locations, the cost matrix between targets and the cost matrix between each robot and all target locations. The cost we used is the Euclidean distance. Our FL-MTSP algorithm consists of two main phases: the assignment phase (Algorithm 2) and the tour construction phase (Algorithm 3).

4.2.1 Assignment phase (Algorithm 2)

First, we select each target location from t_1 to t_m (m is the number of targets) and we calculate the cost of the tour for each robot if this target is assigned to it, using a greedy algorithm. The reason of not using a TSP solver instead of a greedy algorithm to estimate the tour length, is that the TSP solver typically takes very long time when the system becomes large, so it is limited in terms of scalability, but still can be used for small size instances. Then, we determine the *TTD* cost (Eq. 2) and the MT cost (Eq. 4), which will be used as input to the fuzzy logic system to determine the membership function that represent the combination of both objectives. Then, the target location is assigned to the robot that produces the minimum value of the membership

Algorithm 2. The assignment phase

Inputs: nbTargets, nbRobots, matrixDistances
For each target t_i **do**
  For each robot r_i **do**
    Calculate the tourCost for the robot r_i when t_i is assigned to r_i using a greedy algorithm
    Compute the *TTD* when t_i is assigned to r_i
    Apply the fuzzy logic system for *TTD* and MT
  End
  Select the best output obtained by the fuzzy logic system
  If multiple robots has the same best output **do**
    Select the nearest robot that leads to obtaining the minimum total tour cost.
  End
  Add target t_i to the tour list of the best robot
End
For each robot r_i **do**
  While the length of the allocated targets list of $r_i > (m/n)$
    Select the farthest target t_i of robot r_i from its allocated targets list
    Find the nearest robot r_j from t_i
    If $r_j \neq r_i$
      Add t_i to the allocated targets list of r_j
      Remove t_i to the allocated targets list of r_i
    End
  End
Outputs: tour of each robot

Algorithm 3. Tour construction phase

Inputs: nbRobots,
tour: tour of each robot after the assignment step
For each robot r_i **do**
  Apply the TSP_GA solver
End
output: tour, tourCost

function. If multiple robots have the same fuzzy output, then we assign the target to the closest robot to the target location that leads to obtain the minimum total tour cost. This process is repeated until all targets are assigned to the robots. If the length of the allocated targets list of a robot is higher than the ratio of m by $n(m/n)$, we select the farthest target from its assigned targets and add it to the nearest robot. A sample illustration of the FL-MTSP solution is given in Fig. 2. It is shown that the robot $R5$ leaves targets $T5$, $T7$ and $T8$ and the robot $R2$ takes them.

4.2.2 Tour construction phase (Algorithm 3)

Once the process of allocating targets to robots is completed, we use a TSP solver to determine the optimal tour for each robot based on the target locations assigned to it in the previous phase. For this, we used an existing genetic algorithm TSP solver (Kirk 2011) with a population size equal to 100 and a number of iteration equal to 10,000.

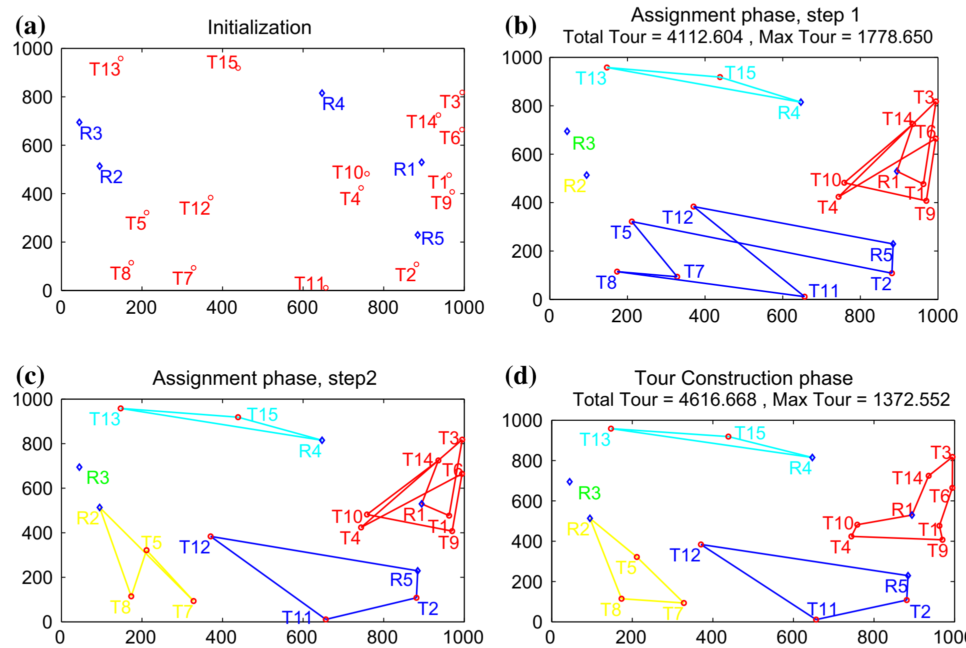

Fig. 2 Simulation example with 5 robots and 15 target locations. **a** The initial position of the robots and the targets to be allocated. The *blue squares* represent the robots and the *red circles* represent the target locations. **b** The tour of each robot after applying the fuzzy logic approach. **c** The final assignment after redistributing the targets and **d** presents the final tour of each robot after applying the TSP_GA solver (color figure online)

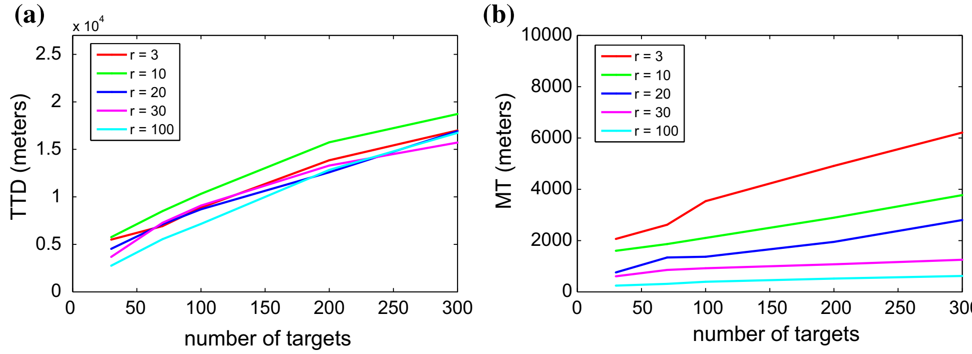

Fig. 3 Impact of the number of targets on the total traveled distance and max tour cost (number of robots is fixed). **a** Total traveled distance, **b** max tour cost

5 Performance evaluation

We have built our own custom simulation using MATLAB under windows OS to implement the proposed approach. All simulations are run on a PC with an Intel Core i7 CPU @ 2.40 GHz and 6 GB of RAM. We evaluate the performance of the FL-MTSP algorithm with two objective functions in scenarios without obstacles where the tour cost is calculated as the Euclidean distance. We adopted the same test problems used in Cheikhrouhou et al. (2014). The number of robots n varies in [3, 10, 20, 30, 100] whereas the number of target locations m varies in the interval [30, 70, 100, 200, 300]. An $(m * m)$ cost matrix is randomly generated and contains the distances between targets. Targets positions are placed in the range of [0, 1000]. In addition, an $(n * m)$ cost matrix is randomly generated and contains the distances between each robot and all targets. Moreover, robots are randomly placed in the range of [0, 1000]. The GA is used to find the least distance for the robot to travel from a fixed starting point and end positions while visiting the other places exactly once. For each scenario, we performed 30 different runs for the algorithm to ensure 95 % confidence interval. For each run, we recorded the tour cost for each robot, the *TTD*, the MT cost and the execution time. The execution time of the algorithm is the average of the 30 execution times. We have explored the performance the proposed approach under varying number of robots and targets.

Impact of the number of target locations: Figure 3 shows the *TTD* and the MT cost as a function of the number of targets, for a fixed number of robots. The *TTD* is presented in Fig. 3a and the MT cost is presented in Fig. 3b. We can observe that, in most cases, the *TTD* and the MT cost increase with the increase of the number of targets for a fixed number of robots. The assignment becomes more difficult when more target locations are involved since the algorithm needs to determine the route for each robot while maintaining the minimum *TTD* and MT cost. We conclude that the increase of the number of targets affect the system performance. It is interesting to

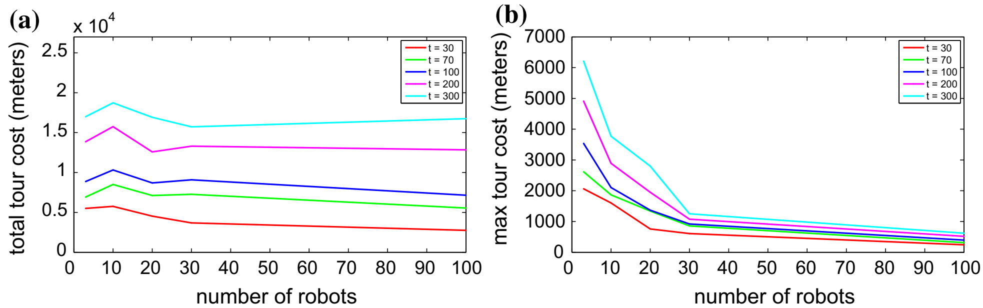

Fig. 4 Impact of the number of robots on the total traveled distance and max tour cost (number of targets is fixed). **a** Total traveled distance, **b** max tour cost

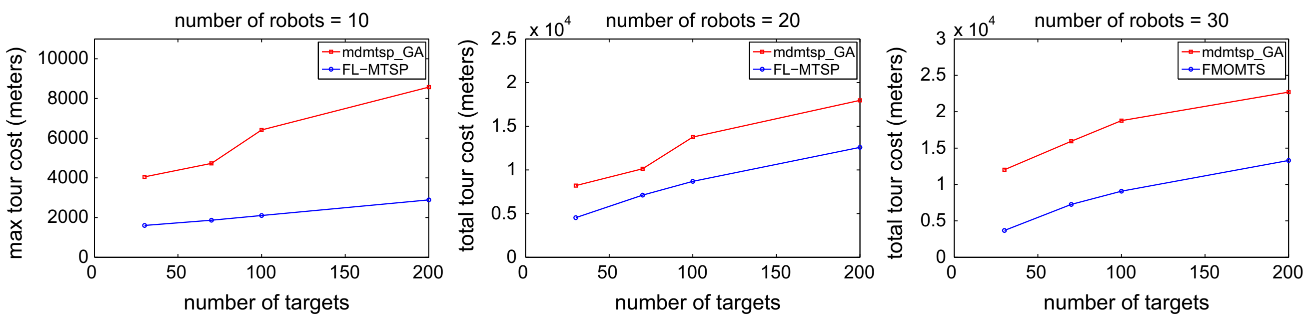

Fig. 5 Comparison between FL-MTSP and the MDMTSP_GA in terms of total traveled distance

notice that increasing the number of robots does not change much the total traveled distance, while there is a huge drop in the maximum tour cost.

Impact of the number of robots: To study the impact of the number of robots, we performed simulations where we fixed the number of targets while varying the number of robots. The results are shown in Fig. 4. We observe that, in most cases, the total traveled distance (Fig. 4a) slightly decreases when the number of robots increases. Moreover, the max tour cost exponentially decreases while increasing the number of robots especially when the number of robots is <30 (Fig. 4b). This means that the target locations are shared between multiple robots in a manner to decrease both the total traveled distance and the maximum tour cost. This result shows the benefit of the use of multiple robots to solve the TSP problem. However, as indicated earlier, the overall traveled distance cost is not affected much by increasing the number of robots for a given number of tasks.

Comparison with MDMTSP_GA: In order to evaluate the performance of our solution, we use the MDMTSP_GA Kivelevitch (2011) which is a centralized approach based on Genetic Algorithm. The MDMTSP_GA solution was simulated in MATLAB. We used a population size equal to 240 and a number of iteration equal to 10,000. These parameters are sufficient to generate good solutions. From Fig. 5, we observe that our algorithm outperforms the MDMTSP_GA in terms of *TTD*. We can noticed that for a large number of robots and targets, the gap between our solution and the MDMTSP_GA solution increases. For example, when the number of target is 200 and the number of robot is 30, the obtained traveled distance cost using our approach is dropped by 70 %. Moreover, for the max tour cost, in FL-MTSP, it changes slowly compared to MDMTSP-GA (Fig. 6). The above two findings demonstrates the effectiveness of our approach in minimizing the overall cost in addition the max tour cost.

As it is difficult to balance multiple objectives simultaneously, the above results prove that the fuzzy logic system is a good process that allows to combine several conflicting objectives and convert a multiple-objective system into a single-objective system.

Also, we prove that the combination of a fuzzy logic system with a heuristic approach leads to optimize the system performance in terms of total traveled distance and max tour cost.

In terms of execution time, it is clear that the gap between our solution and the MDMTSP-GA is very large. The reason of this gap is the fact that the fuzzy logic system from one iteration allows to make decision with no need to repeat the process. This result improves the impact of the use of the fuzzy logic concept that helps to find a solution faster that using only a Heuristic approach (Fig. 7).

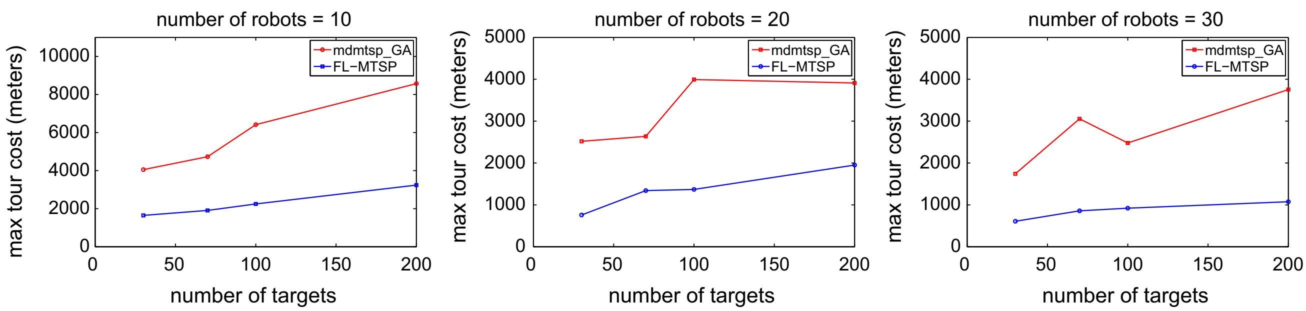

Fig. 6 Comparison between FL-MTSP and the MDMTSP_GA in terms of max tour cost

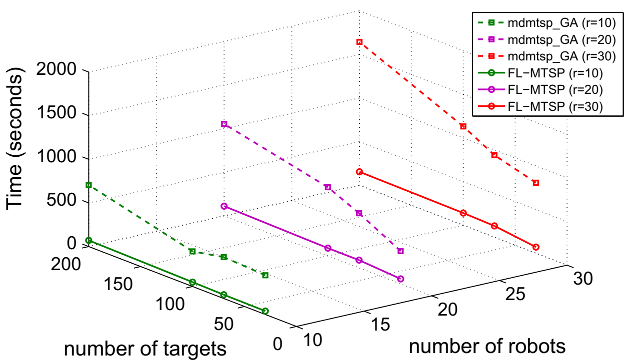

Fig. 7 Time comparison between FL-MTSP and the MDMTSP_GA

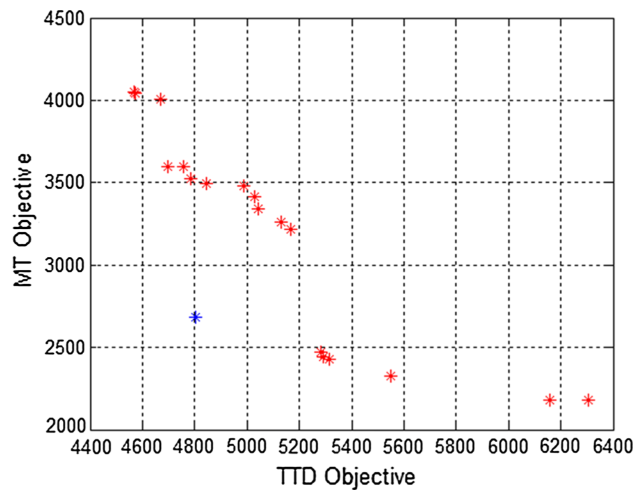

Fig. 8 Solutions example obtained for FL-MTSP (*blue star*) and NSGA-II (*red stars*) (color figure online)

Comparison with NSGA-II: We also compared FL-MSTP with NSGA-II Deb et al. (2002). NSGA-II is an implementation of multi-objective GA for the MSTP problem. NSGA-II adopted the notion of pareto optimality to tackle the problem of multi-objective minimization. Suppose we wish to minimize two objectives O_1 and O_2, and let S_i and S_j be two individual solutions. Let O_{1i}, O_{2i}, O_{1j} and O_{2j} be the values of each of the objectives for both solutions. S_i is said to dominate S_j if and only if $((O_{1i} < O_{1j}) \wedge (O_{2i} \leq O_{2j}))$ **OR** $((O_{2i} < O_{2j}) \wedge (O_{1i} \leq O_{1j}))$. The set of non-dominated solutions is called the pareto-optimal set. Approaches that adopt the notion of pareto optimality maintain a set of pareto-optimal solutions from which the decision maker must choose.

NSGA-II description: As any GA, the first step of the NSGA-II is the population initialization step that must be adequate to the problem formulation. Then, the population is sorted based on the non-domination concept. Each individual is given a rank value. Moreover, for each individual, a crowding distance is calculated. The crowding-distance value is calculated as the sum of individual distance values corresponding to each objective (Deb et al. 2002). After sorting the population based on the crowded distance and the rank, the best individuals are selected. Next, the crossover and mutation operators are applied to the selected population to generate a child population. The parent population and the child population are combined, sorted based on non-domination and N individuals are selected based on their crowding distance and their rank. Note that N is the population size. A detailed description of the NSGA-II algorithm is provided in (Seshadri 2006).

In our implementation, we used the Partial Mapped Crossover (PMX) operator Bolaños et al. (2015) and the strategy of swapping nodes (belonging to different tours) for mutation. The population size in NSGA-II is set to be 100. The algorithm stops after 50 generations (maximum number of iterations). The crossover probability is equal to 0.7 while the mutation probability is equal to 0.9. We mention that the selection of default parameter values is guided by the simulation results. The simulation scenario for the MD-MTSP consists of 3 robots and 30 targets. Figure 8 shows an example of the obtained solutions for both algorithms (FL-MTSP and NSGA-II). The x-axis represents the *TTD* objective while the y-axis represents the MT objective. The red stars are the solutions generated by the NSGA-II and the blue star is the solution generated by FL-MTSP.

A set of non-dominated solutions was generated by NSGA-II and the selection of the best solution depends on the application needs. From the simulation example used for

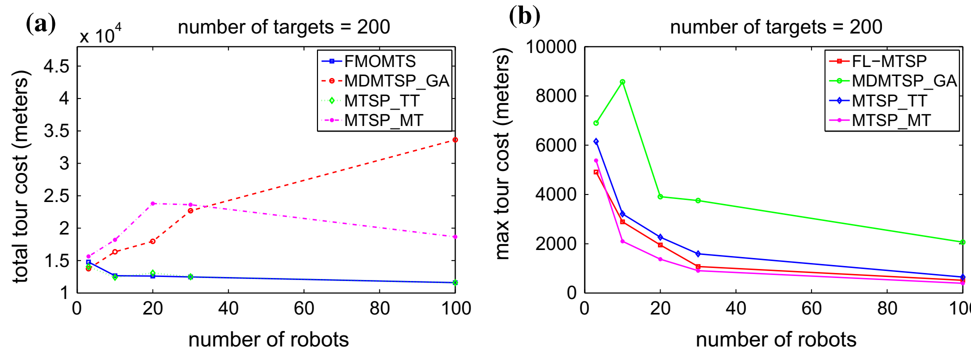

Fig. 9 Comparison between FL-MTSP, MDMTSP_GA, MTSP_TT and MTSP_MT, **a** total traveled distance, **b** max tour cost

the comparison, we noted that the gap between the solutions of NSGA-II and of FL-MTSP is in the range of [4–24 %] for the *TTD* objective. Also, for the MT objective, the gap varies in [23–34 %]. Using the NSGA-II, the solutions are ranged based on their rank of domination. So, most of the solutions will be discarded as they will be assigned lower ranks. This will lead to the loss of promising solutions. Therefore, FL-MTSP provides an acceptable solution in terms of *TTD* and MT as compared to the NSGA-II. This means that the solution obtained by FL-MTSP is better than at least one of the solutions obtained by NSGA-II at least for one of the objectives.

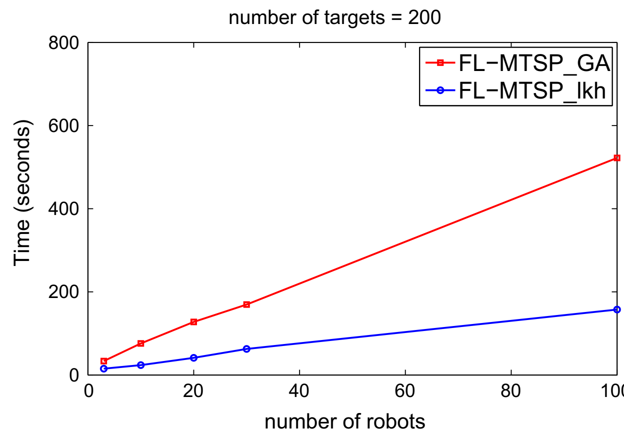

Fig. 10 Time comparison between FL-MTSP using TSP_GA solver and FL-MTSP using TSP_LKH solver

Comparison between FL-MTSP, MDMTSP_GA, MTSP_TT and MTSP_MT algorithms: In this part, we define two new algorithms namely MTSP_TT, which use the total traveled distance as a metric to assign target locations for the MTSP and the MTSP_MT, which use the max tour cost as a metric to assign target locations for the MTSP. For the MTSP_TT algorithm, we use a greedy algorithm to compute the tour cost for each robot if this target is assigned to it. Then the target will be assigned to the robot that leads to obtain the minimum *TTD*. The process is repeated until all targets are assigned to the robots. For the MTSP_MT algorithm, we compute the tour cost and select the max tour cost for each robot if this target is assigned to it, using a greedy algorithm. The target will be assigned to the robot with the minimum max tour cost. Also, like the MTSP_TT algorithm, the process is repeated until all targets are assigned to the robots. If the length of the allocated targets list of a robot is higher than m/n, we select the farthest target from its assigned targets and add it to the nearest robot. Then, we apply an existing TSP solver for the tour construction step (Sect. 4.2.2). To demonstrate that the proposed fuzzy logic approach provides a good trade-off between MT cost and *TTD* cost, we performed simulations where we compare our FL-MTSP algorithm MDMTSP_GA Kivelevitch (2011), MTSP_TT algorithm and MTSP_MT algorithm. Figure 9a, b shows the total traveled distance and the max tour cost respectively for the four algorithms. From Fig. 9a, we depict that the FL-MTSP algorithm gives better results than the MDMTSP_GA and the MTSP_MT algorithm in terms of total traveled distance. Indeed, when using the max tour cost as a metric, the algorithm optimizes the performance of each robot without considering the benefits of the whole system. This result improves the increase of the total traveled distance for the MTSP_MT algorithm. From Fig. 9b, we depict that the FL-MTSP algorithm gives better results than the MDMTSP_GA and the MTSP_TT algorithm in terms of max tour cost. The use of *TTD* as an optimization criteria leads to increase the max tour cost. We deduced that our FL-MTSP solution proposed to solve the MD-MTSP provides a trade-off between total traveled distance and max tour cost.

Impact of the TSP solver on the execution time: To study the impact of the TSP solver, we performed simulations where we used two well known TSP solvers: TSP_GA solver and TSP_LKH solver (Helsgaun 2012). The number of robots varies in the interval [3, 10, 20, 30, 100] and the number of target locations was fixed to 200. Figure 10 shows the results obtained. It is clearly shown that the execution time of the FL-MTSP algorithm using the TSP-GA solver is more time consuming than the FL-MTSP algorithm using the TSP-LKH solver. The gap between the FL-MTSP algorithm using the TSP-LKH solver and the FL-MTSP algorithm using the TSP-GA solver increases while increasing the number of robots. Hence, the choice of a good TSP solver helps to improve the execution time of the algorithm while providing a good solution.

6 Conclusion

Multiple-depot multiple traveling salesman problem is an interesting research area applied in several robotic applications where salesmen share the same workspace. To solve the MD-MTSP, the paper proposed a centralized solution based on the use of the fuzzy logic algebra to combine two objectives: the objective of minimizing the total traveled distance by all the salesmen and the objective of minimizing the maximum traveled distance by any robot. The approach consists of two phases: The assignment phase where the targets allocation is based on the output of the fuzzy logic system, and the tour construction phase, where we used an existing genetic algorithm to build a sub-optimal tour for each robot. Our approach is compared against an existing MD-MTSP solver based on the genetic algorithm. Our approach outperformed the GA approach on both the objectives and also in terms of execution time. We compared our solution against two single-objective algorithms. We proved that our multi-objective algorithm provides a trade-off between total traveled distance and the maximum traveled distance.

Acknowledgments The authors would like to acknowledge the support provided by the National Plan for Science, Technology and Innovation (MAARIFAH)—King Abdulaziz City for Science and Technology through the Science and Technology Unit at King Fahd University of Petroleum and Minerals (KFUPM), the Kingdom of Saudi Arabia, award Project No. 11-LE2147-4.

Compliance with ethical standards

Conflict of interest The authors have no competing financial interest to disclose.

References

Bérubé JF, Gendreau M, Potvin JY (2009) An exact ε-constraint method for bi-objective combinatorial optimization problems: application to the traveling salesman problem with profits. Eur J Oper Res 194:39–50

Bolaños R, Echeverry M, Escobar J (2015) A multiobjective non-dominated sorting genetic algorithm (NSGA-II) for the multiple traveling salesman problem. Decis Sci Lett 4:559–568

Brown EC, Ragsdale CT, Carter AE (2007) A grouping genetic algorithm for the multiple traveling salesperson problem. Int J Inf Technol Decis Mak 6:333347

Carter AE, Ragsdale CT (2006) A new approach to solving the multiple traveling salesperson problem using genetic algorithms. Eur J Oper Res 175:245–257

Cheikhrouhou O, Koubâa A, Bennaceur H (2014) Move and improve: a distributed multi-robot coordination approach for multiple depots multiple travelling salesmen problem. In: IEEE international conference on autonomous robot systems and competitions (ICARSC) 2014, IEEE, p 28–35

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197

Fazli P, Davoodi A, Pasquier P, Mackworth AK (2010) Complete and robust cooperative robot area coverage with limited range. In: 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS), p 5577–5582

Helsgaun K (2012) LKH. http://www.akira.ruc.dk/~keld/research/LKH/

Ke L, Zhang Q, Battiti R (2013) MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and antcolony. IEEE Trans Cybern 43:1845–1859

Khamis AM, Elmogy AM, Karray FO (2011) Complex task allocation in mobile surveillance systems. J Intell Robot Syst 64:33–55

Kirk J (2011) Traveling-salesman-problem-genetic-algorithm. http://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm

Kivelevitch E (2011) MDMTSPV_GA—multiple depot multiple traveling salesmen problem solved by genetic algorithm. http://www.mathworks.com/matlabcentral/fileexchange/31814-mdmtspv-ga-multiple-depot-multiple-traveling-salesmen-problem-solved-by-genetic-algorithm

Kivelevitch E, Cohen K, Kumar M (2013) A market-based solution to the multiple traveling salesmen problem. J Intell Robot Syst 72:21–40

Koubâa A, Trigui S, Châari I (2012) Indoor surveillance application using wireless robots and sensor networks: coordination and path planning. In: Mobile Ad Hoc Robots and Wireless Robotic Systems: Design and Implementation: Design and Implementation, pp 19–57

Liu W, Li S, Zhao F, Zheng A (2009) An ant colony optimization algorithm for the multiple traveling salesmen problem. In: 4th IEEE conference on industrial electronics and applications, 2009. ICIEA 2009, p 1533–1537

Mamdani EH, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man Mach Stud 7:1–13

Marler RT, Arora JS (2010) The weighted sum method for multi-objective optimization: new insights. Struct Multidiscip Optim 41:853–862

Mavrotas G (2009) Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. Appl Math Comput 213:455–465

Nikolić I (2007) Total time minimizing transportation problem. Yugosl J Oper Res 17(1):125–133

Pippin C, Christensen H, Weiss L (2013) Performance based task assignment in multi-robot patrolling. In: Proceedings of the 28th annual ACM symposium on applied computing, p 70–76

Sariel S, Erdogan N, Balch T (2007) An integrated approach to solving the real-world multiple traveling robot problem. In: 5th international conference on electrical and electronics engineering

Seshadri A (2006) A fast elitist multiobjective genetic algorithm: NSGA-II. MATLAB Central 182

Shim VA, Tan KC, Tan KK (2012b) A hybrid estimation of distribution algorithm for solving the multi-objective multiple traveling salesman problem. In: 2012 IEEE congress on evolutionary computation (CEC), p 1–8

Shim VA, Tan KC, Cheong CY (2012a) A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem. IEEE Trans Syst Man Cybern C Appl Rev 42:682–691

Singh A, Baghel AS (2009) A new grouping genetic algorithm approach to the multiple traveling salesperson problem. Soft Comput 13:95–101

Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern 1:116–132

Trigui S, Koubâa A, Cheikhrouhou O, Youssef H, Bennaceur H, Sriti MF, Javed Y (2014) A distributed market-based algorithm for the multi-robot assignment problem. Proced Comput Sci 32:1108–1114

Trigui S, Koubâa A, Ben Jamâa M, Châari I, Al-Shalfan K (2012) Coordination in a multi-robot surveillance application using wireless sensor networks. In: 16th IEEE mediterranean electrotechnical conference (MELECON), IEEE, p 989–992

Wang X, Liu D, Hou M (2013) A novel method for multiple depot and open paths, Multiple Traveling Salesmen Problem. In: 11th IEEE international symposium on applied machine intelligence and informatics (SAMI), IEEE, p 187–192

Xu Z, Li Y, Feng X (2008) Constrained multi-objective task assignment for UUVs using multiple ant colonies system. In: ISECS international colloquium on computing, communication, control, and management, 2008. CCCM'08., vol 1, p 462–466

Yousefikhoshbakht M, Didehvar F, Rahmati F (2013) Modification of the ant colony optimization for solving the multiple traveling salesman problem. Roman Acad Sect Inf Sci Technol 16:65–80

Zadeh LA (1965) Fuzzy sets. Inf Control 8:338–353

Zadeh LA (1975) The concept of a linguistic variable and its application to approximate reasoningI. Inf Sci 8:199–249

Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11:712–731