CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Journal Paper

An Analytical Hierarchy Process-Based Approach to Solve the Multi-Objective Multiple Traveling Salesman Problem

Sahar Trigui

Omar Cheikhrouhou

Anis Koubâa*

Anis Zarrad

Habib Youssef

*CISTER Research Centre

CISTER-TR-180901

2018/09

An Analytical Hierarchy Process-Based Approach to Solve the Multi-Objective Multiple Traveling Salesman Problem

Sahar Trigui, Omar Cheikhrouhou, Anis Koubâa*, Anis Zarrad, Habib Youssef

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: aska@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

Intelligent Service Robotics
https://doi.org/10.1007/s11370-018-0259-8

ORIGINAL RESEARCH PAPER

An analytical hierarchy process-based approach to solve the multi-objective multiple traveling salesman problem

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

Received: 15 September 2017 / Accepted: 22 August 2018

Abstract
We consider the problem of assigning a team of autonomous robots to target locations in the context of a disaster management scenario while optimizing several objectives. This problem can be cast as a multiple traveling salesman problem, where several robots must visit designated locations. This paper provides an analytical hierarchy process (AHP)-based approach to this problem, while minimizing three objectives: the total traveled distance, the maximum tour, and the deviation rate. The AHP-based approach involves three phases. In the first phase, we use the AHP process to define a specific weight for each objective. The second phase consists in allocating the available targets, wherein we define and use three approaches: market-based, robot and task mean allocation-based, and balanced-based. Finally, the third phase involves the improvement in the solutions generated in the second phase. To validate the efficiency of the AHP-based approach, we used MATLAB to conduct an extensive comparative simulation study with other algorithms reported in the literature. The performance comparison of the three approaches shows a gap between the market-based approach and the other two approaches of up to 30%. Further, the results show that the AHP-based approach provides a better balance between the objectives, as compared to other state-of-the-art approaches. In particular, we observed an improvement in the total traveled distance when using the AHP-based approach in comparison with the distance traveled when using a clustering-based approach.

Keywords Assignment · MTSP · Multiple depots · Multi-objective problem · AHP

1 Introduction

Motivation The multiple traveling salesman problem (MTSP) [2] is a generalized form of the well-known traveling salesman problem (TSP) [14,31], where two or more salesmen, sharing the same workspace, are involved in visiting a set of cities. The TSP consists of finding the best route for the salesmen while going through all cities, with the condition of visiting each city only once, and then returning to the depot position. In the case of the MTSP, however, the objective is to find a set of routes with the shortest cost for all salesmen to visit all cities. As for the TSP, each city must be explored by only one salesman. It has been proved that both the TSP and MTSP are NP-hard. There exist several variations of the MTSP, including single or multiple depots. In addition, the routes can either have closed or open paths. A closed path starts from and ends at the home city, whereas in the case

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

Omar Cheikhrouhou
o.cheikhrouhou@tu.edu.sa

Anis Koubaa
aska@isep.ipp.pt

Anis Zarrad
a.zarrad@bham.ac.uk

Habib Youssef
Habib.youssef@fsm.rnu.tn

1 ENSI, University of Manouba, Manouba, Tunisia

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

3 Taif University, Taif, Saudi Arabia

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

5 Prince Sultan University, Riyadh, Saudi Arabia

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

7 School of Computer Science, University of Birmingham, Edgbaston, Birmingham, UK

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

Published online: 08 September 2018

of an open path, the salesmen do not need to return to the home city [43]. In the MTSP, the cities are identical. In other words, each city is accessible for any salesman [25]. Bektas [2] provided a comprehensive survey of the MTSP and its applications. Exact and heuristic solution procedures have been proposed for solving this problem. Genetic algorithms (GA) have been widely used to solve the MTSP [4,5,36], and ant colony optimization (ACO) algorithms have been proposed in [13,26,46]. Venkatesh and Singh [41] presented two approaches for solving the MTSP based on the use of two new swarm intelligence metaheuristic techniques: the artificial bee colony (ABC) [18] algorithm and the invasive weed optimization (IWO) algorithm [28]. Compared to the TSP, the MTSP is more appropriate for modeling real-world applications such as logistics transportation, job planning, and vehicle scheduling.

Disaster management is one of the most challenging applications for multi-robot systems. The problem is how to assign the robots to specific areas affected by disaster events, such as fires, earthquakes, or water floods. In such applications, there is a crucial need to optimize several metrics, also known as objectives, that can be conflicting in nature. In its abstract form, the problem can be mapped to a multi-objective MTSP, where a set of agents must visit a set of locations while considering a set of objectives.

Recently, several evolutionary algorithms [19,30,33,34] have been used to solve the multi-objective MTSP including GA, ACO, artificial neural networks, and particle swarm optimization. These algorithms usually work on a set of solutions that provide a trade-off between the objectives. Such solutions are called Pareto optimal solutions [29]. Although in many cases metaheuristic-based algorithms (including evolutionary algorithms) can be very effective in solving hard optimization problems, in some cases they have extensive computation overheads, and their convergence is challenging, especially when applied to large problem instances.

This work investigates the multi-objective multiple-depot multiple traveling salesman problem in a disaster management application while optimizing three objectives: the sum of the costs of all robots, the maximum cost among all robots, and the workload among the robots. We propose the use of the analytical hierarchy process (AHP) [32] to systematically determine the optimized weights for the different objectives. The benefit of using AHP is that it allows effective assignment of weights to objective functions. Instead of assigning weights while relying on heuristic knowledge of the problem domain, AHP relies on the rigor of statistical analysis.

Contributions This paper proposes the following contributions:

- We provide a comprehensive literature review of the relevant works that solve the MTSP.
- We propose a solution that uses the AHP to systematically determine a precise weight for each objective.
- We assess three different approaches to solve the problem.
- We compare our solution approach against multi-objective algorithms.

2 Related work

2.1 Single-objective algorithms

GA-based heuristics GA [17] are among the most widely used algorithms to solve hard combinatorial optimization problems such as the MTSP [20,25,37,38].

To solve the MTSP, Yuan et. al [35] used a two-part chromosome representation and proposed a new crossover operator. The new solution was compared to other crossover methods, including the ordered crossover operator (ORX), the cycle crossover operator (CYX), and the partially matched crossover operator (PMX). The authors used two different objective functions: the total distance traveled by all salesmen and the longest route among all the salesmen. The experiments of Yuan et al. showed that the new crossover method enables the genetic algorithm to produce better solution quality.

Grouping genetic algorithms (GGA) [12] are a variation of the GA. In [4], a GGA was developed for the MTSP. The concept is to group the cities into routes and then order the cities on each route. For the comparison, the authors considered two objectives: the maximum tour length and the total tour length. The simulation study showed that the GGA produces better solutions when the objective is to minimize the maximum tour length as the GGA is designed for grouping problems. When the objective is to minimize the total tour length, however, the performance of the GGA degrades.

Another GGA called the steady-state grouping genetic algorithm (GGA-SS) was proposed in [36]. The chromosome representation and genetic operators used in that paper are different from those used in [4]. Similar to several other works in the literature, the aim was to minimize the total distance traveled by all the salesmen, as well as the distance traveled by any salesmen. The authors used the test problems proposed in [4,5], and the results showed that the GGA-SS is able to produce better solutions for the adopted objectives than those found in [4,5].

ACO-based solutions The ACO algorithm is an iterative search technique inspired by the behavior of real ant colonies. In the literature, the ACO algorithm has been used to solve several robotic problems, such as path planning [6,7] and coordination [24]. In [26], an ACO algorithm was used to solve the MTSP with the objective of minimizing both the total maximum tour length of all the salesmen and the maximum tour length of each salesman. The simulation results of the algorithm showed that the ACO algorithm outperforms the GA-based algorithms proposed in [4,5,36].

Additionally, a modified ACO algorithm (NMACO) was proposed in [46]. To improve the quality of the solution, the authors suggested modifications including the transition rule, the candidate list, the global pheromone updating rules, and several local search techniques. The aim of the algorithm is to minimize the distance traveled by the salesmen. The solution was tested on standard benchmarks available from the literature, and performance results showed that the NMACO algorithm gives better solutions than the existing solutions for the MTSP.

Other heuristic solutions In [41], the authors studied the MTSP and proposed two metaheuristic approaches. The first approach is based on the ABC algorithm [18], and the second approach is based on IWO algorithm [28]. Both approaches were evaluated using benchmark instances available in the literature. The results were comparable to state-of-the-art solutions in terms of total distance traveled by all the salespersons, and the maximum distance traveled by anyone salesperson demonstrated that the ABC and IWO algorithms give better results for both objectives.

Market-based approaches Other solution approaches known as market-based approaches have been used to solve the MTSP. In [9], the authors proposed a new approach called *move and improve*. The approach includes four steps: initial target allocation, tour construction, elimination of conflicting targets, and solution improvement. To measure the performance of the algorithm, two metrics were used: the total cost and maximum cost. The simulation results demonstrated the superiority of the move and improve algorithm in comparison with a centralized GA.

The solution proposed in [23] consists of four steps: market auction, agent-to-agent trade, agent switch, and agent relinquish. Three performance criteria were considered: the quality of the solution, the number of iterations required to obtain a solution, and the execution time. It was shown that the approach generates better solutions than other suboptimal solutions.

As several real-world applications must optimize multiple objectives, we address the multi-objective problem and propose an approach that provides trade-off solutions while trying to optimize several objectives simultaneously, including the total traveled distance and the maximum tour length.

2.2 Multi-objective algorithms

In the literature, few researchers have considered the MTSP as a multi-objective optimization problem.

In [3], the authors proposed a multi-objective, non-dominated sorting genetic algorithm (NSGA-II) to solve the MTSP. The objectives to be optimized were: the total traveled distance and the working times of the salesmen. The authors sought to find a set of non-dominated solutions that, when compared, were better for certain objectives, while others were better for other objectives. To evaluate the performance of the proposed approach, two different test instances (three salesmen with 29 nodes and three salesmen with 75 customers) were considered. The results showed the effectiveness of the NSGA-II in minimizing both objectives.

In [44], the authors used the ACO algorithm to solve a task assignment problem for multiple unmanned underwater vehicles. The authors aimed to optimize two objectives: the total distance necessary to visit all targets and the total turning angle while considering the constraint of balancing the number of targets visited by each vehicle. The solution approach consisted of two phases. The first phase is the task number assignment phase, which consists of specifying the number of targets for each vehicle. The second phase solves the MTSP using an ant colony for each objective. Performance evaluation showed that the algorithm generates good solutions.

The market-based approach has been widely used to solve a number of problems, including the MTSP. In [11], the reported approach consisted of using a clustering technique with an auction process. The objectives are to minimize the distance traveled by all the robots and to balance the workload equally between the robots. The first step is meant to decompose N tasks into n groups, in such a way that the distance inside each cluster is minimized. The cost for each robot to visit n clusters is then computed. Finally, in the auction step, each cluster is allocated to the robot that provides the lowest bid. We noticed that the complexity of the algorithm is relatively high because all possible combinations of the assignment of clusters to robots are considered. This means that the approach can only be used with a very small number of clusters. To evaluate the performance of the algorithm, the authors used the benchmark VRP data set “A-n32-K5.vrp.” The total cost used for the assignment is equal to the sum of the cost of visiting the tasks in the cluster and the idle cost (i.e., sum of the difference in cost of travel between any two robots). Two scenarios were considered: one with two clusters and the other with three clusters. We noticed that the scenarios that were used are not sufficient to prove the effectiveness of the algorithm.

Recently, another clustering market-based approach (CM-MTSP) [40] was presented to solve the multi-objective MTSP. The algorithm consists of grouping the targets into clusters and then allocating each cluster to the best robot. In that work, the authors assumed that the number of clusters is equal to the number of robots. The comparison results showed that the CM-MTSP provides a good balance between conflicting objectives and reduces the execution time as compared to a greedy market-based solution.

We focused on comparing our approach with the clustering approaches proposed in [11,40] in terms of total traveled distance and maximum tour length.

In [15], an auction algorithm using a clustering technique has been proposed with the objective of minimizing both the maximum traveled distance of each robot and the sum of the distance traveled by all robots. The algorithm proceeds as follows. Initially, it is assumed that all robots have a list of allocated tasks. In the case where a robot reaches the position of a task, it sends a specific signal to the other robots to be able to make an auction. After all auction offers are completed, the robot re-plans its path and moves to the next task. If a robot receives a message to make an auction offer, it forms a new set of clusters of its assigned tasks and then an auction process starts for the newly formed clusters, with the exception of the cluster that contains its currently initialized task. When a robot receives an auction message for a cluster, it bids for that cluster. Finally, the robot with the best bid wins the cluster. The performance evaluation has shown the percentage of improvement in the initial assignment in comparison with the final assignment.

To solve the multi-objective TSP, Lust and Teghem [27] presented a new method called two-phase Pareto local search. The solution consists of two phases. The first one concerns the resolution of each single-objective problem separately using the Lin–Kernighan heuristic. In the second phase, a Pareto local search method is adopted. A 2-opt neighborhood with candidate lists is applied to improve the solutions generated in the first phase. It is important to note that a high number of weighted single-objective problems must be solved before applying the Pareto local search, which may cause efficient degradation. Additionally, the integration of the 2-opt process may achieve poor effectiveness with low efficiency when the number of feasible objective vectors is small, whereas it obtains the desired effectiveness with low efficiency when the number of feasible objective vectors is large.

In [19], a MOEA/D-ACO algorithm was proposed to solve the multi-objective MTSP. The proposed algorithm is a combination of ACO algorithm with the multi-objective evolutionary algorithm based on decomposition (MOEA/D). The problem is decomposed into a number of mono-objective subproblems. Each ant is assigned to solve one of the mono-objective subproblems. The ants are split into groups, and each one has multiple neighboring ants. Each group has a pheromone matrix, and each single ant has a heuristic information matrix. Each ant is responsible for finding the best solution for its assigned subproblem. For that purpose, the ant uses its heuristic information matrix, the pheromone matrix of its group, and its current solution. The main issues related to this approach are the uncertainty of the time convergence and the implementation complexity.

In [30], a detailed comparison between two multi-objective evolutionary algorithms, MOEA/D and NSGA-II, was presented. Also, the authors studied the effect of local search on the performance of MOEA/D. The test problem used was the multi-objective TSP. Compared to MOEA/D, NSGA-II has no bias in searching any particular part of the Pareto front. All non-dominated solutions in the current population have an equal chance of being selected for reproduction. However, this might not be efficient when sampling offspring solutions, for the following reasons. First, the non-dominated solutions might have very different structures in the decision space. Therefore, the possibility of generating high-quality offspring solutions by recombining these solutions is low. Second, designing recombination operators is often problem dependent. In MOEA/D, weight vectors and aggregate functions play a very important role in solving various kinds of problems. Overall MOEA/D has been shown to be much better algorithmic improvement than NSGA-II.

A new fuzzy logic-based approach (FL-MTSP) was proposed to solve the multi-objective MTSP [39]. This approach consists of the combination of two objectives, the maximum traveled distance and the total traveled distance, to reduce the problem to a single-objective optimization problem. A comparative study of the FL-MTSP approach proved its effectiveness in comparison with an existing MTSP solver based on the GA [22] and with a NSGA-II for MTSP. In the simulation, we compare the FL-MTSP approach with our solution.

Most of the existing proposed approaches have been criticized mainly for their computational complexity, their necessity for prior system knowledge to define a weight for each objective, and their lack of specifying sharing parameters.

In this work, we followed a three-phase mechanism based on the AHP to define weights systematically for each objective, depending on the application characteristics.

3 Problem formulation

We address the multi-objective multiple-depot MTSP. We consider a set of m robots $\{R_1, \ldots, R_m\}$, initially located at different depots $\{T_1, \ldots, T_m\}$, which must visit a set of n target locations $\{T_m + 1, \ldots, T_m + n\}$ and return to their depots after mission completion. The objective is to find an effective assignment of robots to the set of target locations, such that each target is visited only once by *exactly* one robot. We define tour$_{R_i}$ as the tour of robot R_i starting from and ending at its depot T_i and going through the list of its allocated targets $\{T_{i_1}, \ldots, T_{i_{ni}}\}$ in that order. The tour cost of robot R_i may be any of several things, including Euclidean distance, time, and consumed energy. In the context of the multi-objective optimization problem, the goal is to generate solutions that provide a good trade-off between the objectives.

The objective functions can be classified into three categories. The first category includes objective functions that minimize the sum of the costs of all robots, such as minimizing the total distance traveled and minimizing the total consumed energy. This category of objective functions is defined as:

$$\text{minimize} \quad \sum_{k=1}^{m} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} x_{ijk} C(T_i, T_j) \tag{1}$$

subject to :

$$\sum_{k=1}^{m} \sum_{i=1}^{n+m} x_{ijk} = 1; \quad \forall j = 1, \ldots, n+m \tag{2}$$

$$\sum_{k=1}^{m} \sum_{j=1}^{n+m} x_{ijk} = 1; \quad \forall i = 1, \ldots, n+m \tag{3}$$

$$\sum_{i=1}^{n+m} x_{kik} = 1; \quad \forall k = 1, \ldots, m \tag{4}$$

$$\sum_{i=1}^{n+m} x_{ikk} = 1; \quad \forall k = 1, \ldots, m \tag{5}$$

$$x_{ijk} \in \{0, 1\}; \quad \forall i, j = 1, \ldots, n+m \quad \text{and} \quad k = 1, \ldots, m \tag{6}$$

Equations (2) and (3) ensure that each node is visited only once by a single robot, whereas Eqs. (4) and (5) ensure that each robot starts from each corresponding depot and returns back to it. Finally, constraint (6) ensures that the decision variables are binary, where $x_{ijk} = 1$ if robot R_k is assigned to target T_i and 0 otherwise.

The second category includes objective functions that minimize the maximum cost among all robots, so as to minimize the maximum tour, and the mission time, which corresponds to the maximum time. This category of objective functions can be modeled as:

$$\text{minimize} \quad \max_{k \in 1 \ldots m} \left(\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} x_{ijk} C(T_i, T_j) \right) \tag{7}$$

subject to the same constraints defined in Eq. (2)–(6).

The third category of objective functions is related to balancing the workload among the robots, such as balancing the length of tours, the mission times, and the number of allocated targets. This category of objective functions can be expressed as follows:

$$\begin{aligned}
&\text{minimize} \quad \sum_{k=1}^{m} \left| C_k - C_{\text{avg}} \right| \\
&C_k = \left(\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} x_{ijk} C(T_i, T_j) \right), \quad k \in [1, m] \\
&C_{\text{avg}} = \frac{\sum_{k=1}^{m} \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} x_{ijk} C(T_i, T_j)}{m}
\end{aligned} \tag{8}$$

C_k represents the tour cost for robot k. As previously mentioned, the cost can refer to time, traveled distance, or energy. C_{avg} represents the average tour cost.

In the system model, it is assumed that every robot has a global knowledge of the targets positions. In addition, each robot has the capability to estimate the cost to reach a position. This cost can be the Euclidean distance, time, or energy.

4 Multi-objective multiple-depot MTSP solution

The idea of our proposal is based on the use of a weight-based approach with the aim of assigning appropriate weights to the objectives using the AHP approach [32]. We define the *global cost* as the sum of the weighted costs of the different objective functions under consideration. The cost function is computed using Eq. (9):

$$\begin{aligned}
&g(x|W) = \sum_{i=1}^{p} w_i f_i(x) \\
&\quad \text{subject to :} \quad x \in \Omega
\end{aligned} \tag{9}$$

where $W = (w_1, \ldots, w_p)$, $0 < w_i < 1 \quad \forall i = 1, \ldots, p$, w_i is the weight of the objective function $f_i()$, $\sum_{i=1}^{p} w_i = 1$, and Ω is the decision space.

The proposed solution approach comprises three main steps: (1) determination of the relative weights for the individual objective functions using the AHP approach, (2) determination of optimum tours for the robots using selected solution approach, market-based, RTMA-based, or balanced-based approach, and (3) an improvement phase (see Fig. 1).

We define a comparison matrix that represents the priority of each objective function relative to the other objectives. This matrix is used as an input for the AHP process, which generates a weighted vector. This vector is then used to compute the global cost (Eq. 9). After applying all the approaches, the best solution will be selected (Algorithm 1).

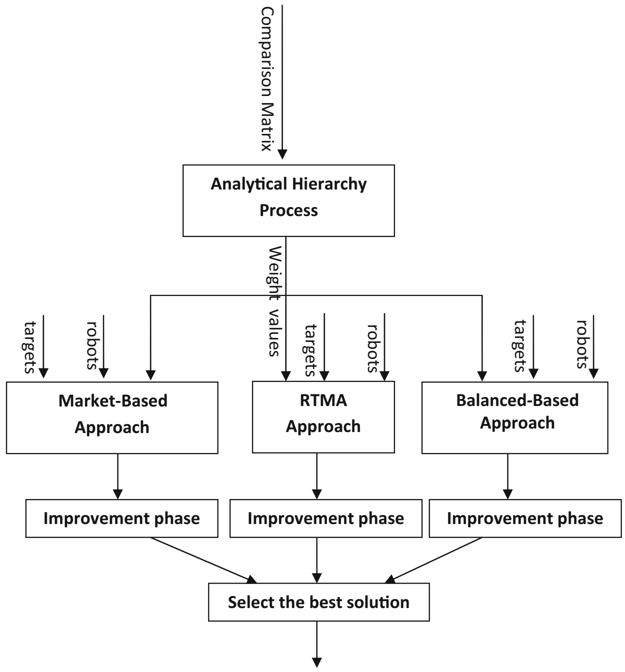

Fig. 1 Flowchart of the proposed solution

Algorithm 1 Proposed Solution General Algorithm

Input: Comparison matrix, Targets, Robots
Output: Best tours assignment

```
Begin
    Generate weight vector using AHP
    Market-Based approach
    RTMA approach
    Balanced-based approach
    Select the best solution
End
```

4.1 The analytical hierarchy process

The AHP is a structured technique developed in 1970s to solve complex decision-making problems [32]. The AHP helps decision makers quantify the elements of a decision problem. The decision problem is, firstly, decomposed into a hierarchy of subproblems. Then, the elements of the hierarchy will be evaluated via pairwise comparisons to construct a comparison matrix. The input data can be either an actual measurement (e.g., price, weight) or a subjective opinion (e.g., preference, satisfaction, feeling).

Regarding a disaster management application, we consider three objective functions: the total traveled distance (TTD), the maximum tour (MT), and the deviation rate of tour lengths (DR). Mission time is the most important metric in applications like fire disasters. However, it is proportional to the MT. In addition, the minimization of both the TTD and the DR leads to minimize the energy consumed by the team and to balance the workloads among all the team members.

We then consider the following comparison matrix:

$$A_{i,j} = \begin{array}{c} \\ \mathrm{TTD} \\ \mathrm{MT} \\ \mathrm{DR} \end{array} \begin{pmatrix} \mathrm{TTD} & \mathrm{MT} & \mathrm{DR} \\ 1 & 1/2 & 1/3 \\ 2 & 1 & 1/2 \\ 3 & 2 & 1 \end{pmatrix} \tag{10}$$

Regarding the comparison matrix, the MT is twice as important as TTD, and the DR is three times as important as TTD and twice as important as the MT. Note that the values of the comparison matrix describe the user preferences and are generally related to the applications use case. For example, in case of a disaster management application, the most important criterion is the mission time which is proportional to the MT.

Suppose that $a_{i,j}$ is the element of ith row and jth column of the comparison matrix, then $a_{i,j} = \frac{1}{a_{j,i}}, \forall i, j$ and $a_{i,i} = 1$. The eigenvector W is computed according to: $A_{i,j} W = \lambda W$, where λ is the eigenvalue.

In this work, we use the eig() MATLAB function to compute the eigenvector, which gives $W = \{0.2565, 0.4660, 0.8468\}$. The three numbers in the eigenvector are proportional to the relative weights of the three criteria. Because the relative weights must sum up to 1, we normalized the eigenvector W by dividing each number in it by the sum of all numbers. The corresponding weight vector is $W = \{0.1634, 0.2970, 0.5396\}$.

4.2 Market-based approach

The market-based approach consists of an auction process for targets. All robots compete to win the best target. More precisely, each robot selects the best target (i.e., the target that has the minimum *local cost*) and sends a bid for that target to a central machine. The *local cost* is defined as the weighted sum of the objective function costs for that robot. For example, to bid for target $T1$, the robot $R1$ computes its tour cost, including $T1$, using the Lin–Kernighan heuristic TSP solver [16]. The bid contains the selected target and the corresponding costs for each objective function. Note that each robot bids independently from the others. More precisely, the target to bid for can differ from one robot to another. For example, $R1$ bids for $T1$, while $R2$ bids for $T2$ simultaneously.

Upon receiving bids from the different robots, the central machine computes the *global cost* for each corresponding bid and then assigns the best target to its corresponding robot. The best target refers to the target with the minimum global cost. Unlike local cost, the global cost considers all tour costs, such as the sum of all tour lengths, the maximum tour length, and the tours length deviation rate. Robots continue the process of bidding until all targets are assigned (Algorithm 2).

Algorithm 2 Market-based approach

Input:
Robots R_i $(1 < i < m)$, Targets T_j $(1 < j < n)$, weights, $Available\_Targets\_List$ = non allocated targets
while $Available\_Targets\_List \neq \emptyset$ **do**
  for each robot R_i **do**
    if $tour_{R_i}$ is changed **then**
      Compute $cand\_tour_{R_i}$ cost including each available target
      Select the target with the lowest *local cost* to $cand\_tour_{R_i}$
      Bid on the selected target
    else
      if R_i bids for last allocated target **then**
        Remove the last allocated target from $cand\_tour_{R_i}$
        Choose another best target
      else
        Bid for the last chosen target
      end if
    end if
  end for
  for each robot R_i **do**
    Compute the *global cost*
  end for
  Select the robot R_j with the lowest *global cost*
  Add the allocated target to $tour_{R_j}$
  Remove the allocated target from the $Available\_Targets\_List$
end while
Output:
$tour_{R_i}, TTD, MT, DR$

Figure 2a illustrates the market-based approach. Consider a scenario with two robots and six targets, and two objective functions TTD and MT with the assumption that the priority of TTD is twice that of MT. Assume the weight vector as W={0.66, 0.33}. Table 1 shows the stepwise execution of the market-based approach for that example. First, $R1$ selects $T1$ and $R2$ selects $T5$. As the global cost when assigning $T1$ to $R1$ is less than the global cost when assigning $T5$ to $R2$, the assignment will be made to $R1$. The assignment process continues until all targets are allocated (Fig. 2a).

4.3 RTMA-based approach

The idea of this approach is inspired by the robot and task mean allocation algorithm (RTMA) method proposed in [42]. We extend the original RTMA-based approach to solve a multi-objective optimization problem.

The idea of the RTMA-based approach is to make the robot choose the target that leads to the best cost for the group, instead of choosing the one that results in minimum cost for the robot itself. In other words, each robot selects the target that appears to give an optimized RTMA cost, instead of choosing the target that would give an optimized local cost. The RTMA cost is computed as the difference between the cost of the robot visiting a target and the mean cost for this robot to visit all the targets. Formally, for a given robot, the RTMA cost to move from target T_i to target T_j is:

$$\mathrm{Cost}_{\mathrm{RTMA}}(T_i, T_j) = C(T_i, T_j) - \frac{\sum_{t=1}^{n} C(T_i, T_t)}{n} \tag{11}$$

where $C(T_i, T_j)$ is the (normal) cost to move the robot from T_i to T_j and n is the number of targets.

To better illustrate the RTMA cost, consider as an example the Euclidean distance between two targets as the value of cost. The RTMA cost is:

$$\mathrm{Cost}_{\mathrm{RTMA}}(T_i, T_j) = D(T_i, T_j) - \frac{\sum_{t=1}^{n} D(T_i, T_t)}{n} \tag{12}$$

where $D(T_i, T_j)$ is the Euclidean distance between T_i and T_j.

We compute the RTMA cost for each robot to travel from its depot to each target. Each target is then assigned to the robot having the lowest global cost.

To illustrate the effectiveness of the RTMA-based approach, let us consider the example shown in Fig. 2. We define

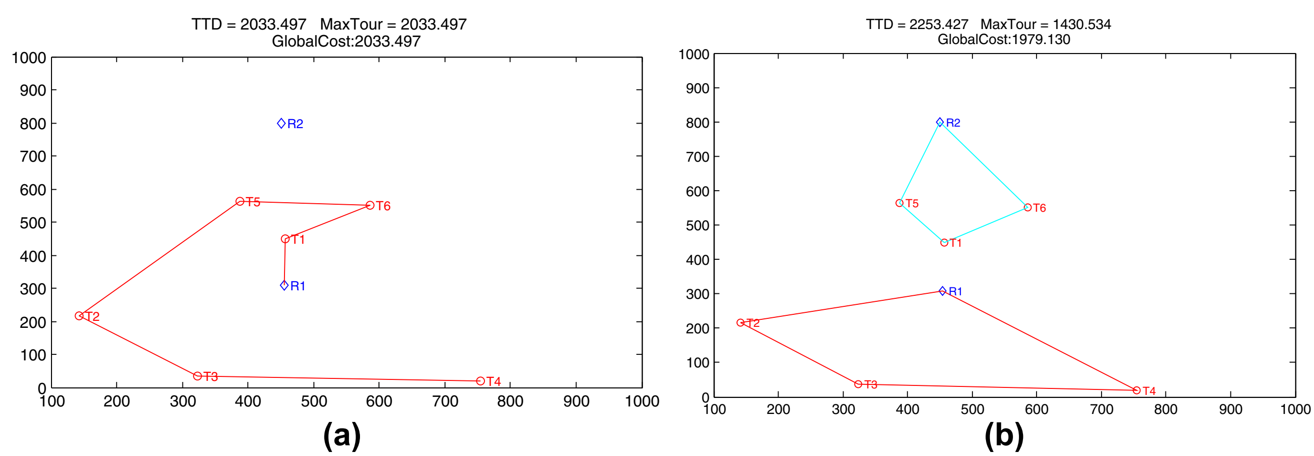

Fig. 2 Example of specific scenario (two robots and six tasks), solved using either **a** the market-based approach or **b** the robot and task mean allocation algorithm (RTMA)-based approach

Table 1 Step-by-step execution of the market-based approach for the scenario illustrated in Fig. 2 (two robots and six tasks); *TTD* total travelled distance, *MT* maximum tour

Step	Robots	Bidding	Server side				Winner
		Target	Tour	TTD	MT	Global cost	
1	$R1$	$T1$	280	280	280	280	$R1$
	$R2$	$T5$	486	486	486	486	
2	$R1$	$T5$	537	537	537	537	$R1$
	$R2$	$T5$	486	766	486	673	
3	$R1$	$T6$	748	748	748	748	$R1$
	$R2$	$T6$	562	1099	562	920	
4	$R1$	$T2$	1254	1254	1254	1254	$R1$
	$R2$	$T2$	1316	2065	1316	1815	

Table 2 Robot and task mean allocation algorithm (RTMA) assignment for the scenario illustrated in Fig. 2 (two robots and six tasks)

Targets	RTMA cost		Winner
	$R1$	$R2$	
$T1$	– 147.1609	– 173.9680	$R2$
$T2$	38.0983	134.7985	$R1$
$T3$	14.5316	249.0928	$R1$
$T4$	129.5453	313.1256	$R1$
$T5$	– 24.0174	– 280.4695	$R2$
$T6$	– 10.9970	– 242.5793	$R2$

the cost between two targets or between a robot and a target as the Euclidean distance. In this example, it is shown that the market-based approach does not give the best solution and that the RTMA-based approach outperforms the market-based one in this particular scenario. Indeed, when following the market-based approach, the global cost is 2033; however, with the RTMA-based approach, the global cost is 1979, with a weight vector of $W = \{0.66, 0.33\}$. Table 2 reports the distances between the targets and the robots.

4.4 Balanced-based approach

The idea of the balanced-based approach is to fairly balance the number of assigned targets among all robots. More precisely, if we have m robots and n targets, each robot will be assigned approximately $\frac{n}{m}$ targets.

The balanced-based approach process is similar to the market-based one, with the addition of the assumption that *a robot exits the bidding process when it is assigned a sufficient number of targets*. The sufficient number of targets is no more than $\frac{n}{m}$. This allows us to fairly balance the workload between the robots in terms of the number of targets allocated to each robot and also in terms of their tour length (Algorithm 3).

Algorithm 3 Balanced-based approach

Input:
Robots R_i $(1 < i < m)$, Targets T_j $(1 < j < n)$, weights, $Available\_Targets\_List$ = non allocated targets, $Available\_Robots\_List$ = available robots, $targets\_portion$

while $Available\_Targets\_List \neq \emptyset$ and $Available\_Robots\_List \neq \emptyset$ **do**
  for each robot R_i **do**
    if $tour_{R_i}$ is changed **then**
      Compute $cand\_tour_{R_i}$ cost including each available target
      Select the target with the lowest *local cost* to $cand\_tour_{R_i}$
      Bid on the selected target
    else
      if R_i bids for last allocated target **then**
        Remove the last allocated target from $cand\_tour_{R_i}$
        Choose another best target
      else
        Bid for the last chosen target
      end if
    end if
  end for
  for each robot R_i **do**
    Compute the *global cost*
  end for
  Select the robot R_j with the lowest *global cost*
  Add the allocated target to $tour_{R_j}$
  Remove the allocated target from the $Available\_Targets\_List$
  if number of allocated targets to $R_j = targets\_portion$ **then**
    Remove robot R_j from $Available\_Robots\_List$
  end if
end while

Output:
$tour_{R_i}, TTD, MT, DR$

Figure 3 shows an illustrative example where the balanced-based approach outperforms the market-based one, giving a better solution. In this case, the use of the market-based approach resulted in the assignment of all targets to one robot, here $R1$, However, the balanced-based approach ensures that the targets will be assigned uniformly between robots $R1$ and $R2$. This will lead to a reduction in the global cost in comparison with the global cost found when applying the market-based approach. More precisely, with the market-based approach, the tour lengths of $R1$ and $R2$ are 3362.5 and

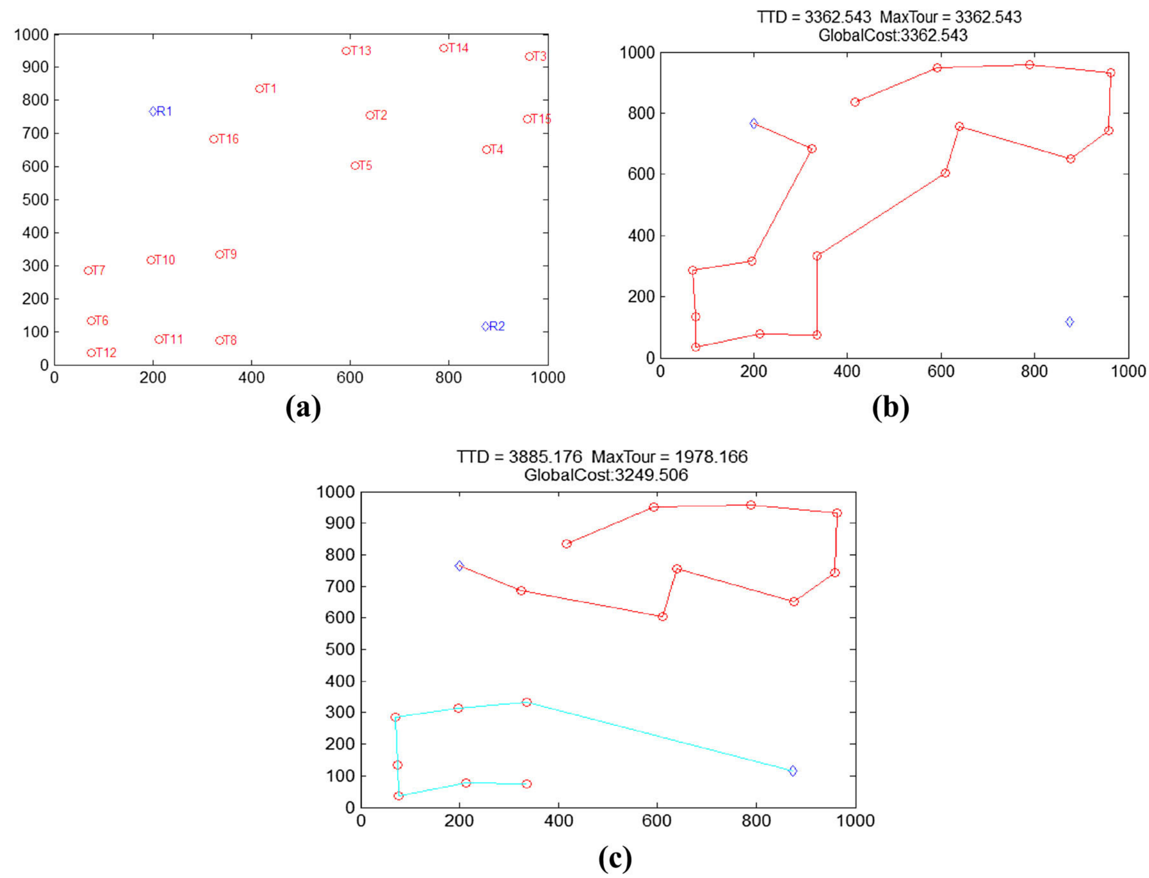

Fig. 3 Example of specific scenario (two robots and 16 tasks); **a** initial configuration of tasks and robots; scenario solved following one of two approaches: **b** market-based approach and **c** balanced-based approach

0, respectively, whereas with the balanced-based approach, they are 1978.16 and 1271.33, respectively.

4.5 Improvement phase

The aim of this phase is to improve the solutions generated after applying the approaches described above. The improvement consists in minimizing the global cost. Each robot selects its worst target, i.e., the target that introduces the largest cost, and then all robots bid on this target. For example, if a robot $R1$ is able to visit robot $R2$'s worst target with a lower global cost, the target will be added to $R1$'s tour and deleted from $R2$'s tour.

5 Simulation study

In this section, we present the performance evaluation of the proposed strategy to solve the multi-objective MTSP. We conduct our simulation using MATLAB. We evaluated the total traveled distance, the maximum tour, and the deviation rate objectives. The global cost is computed as follows:

$$\text{Global cost} = w_1 \sum_{k=1}^{m} C_k + w_2 \max_{k \in 1 \ldots m} (C_k) + w_3 \sum_{k=1}^{m} \left| C_k - C_{\text{avg}} \right| \tag{13}$$

where C_k, C_{avg} are computed as indicated in Eq. 8 and $C(T_i, T_j)$ is the Euclidean distance between the two targets T_i and T_j. We used the LKH-TSP solver [16] to compute the tour cost of a robot. The LKH-TSP solver has shown its ability to produce optimal solutions to most problem instances. Also, the LKH-TSP is efficient for large-scale problems[1,45]. For all simulations, the weight vector used is $W = \{0.1634, 0.2970, 0.5396\}$.

5.1 Comparison between the market-based, RTMA-based, and balanced-based approaches

We adopted the test problems where the number of target locations is equal to 3 × number of robots and the number of robots varies in the interval [3 5 10 15]. Robots and targets positions are placed in a 1000 ∗ 1000 space. For each con-

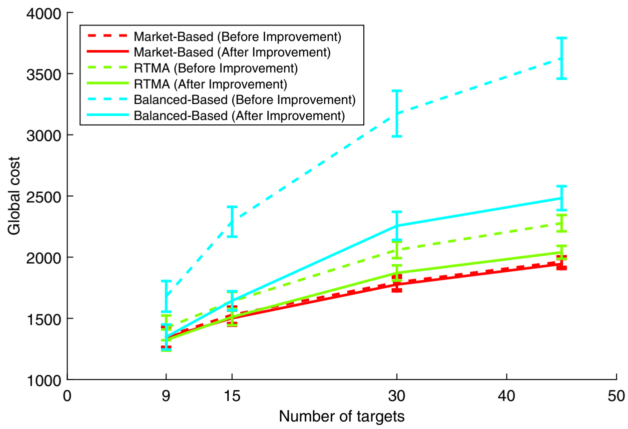

Fig. 4 Comparison between the proposed approaches and impact of the improvement phase

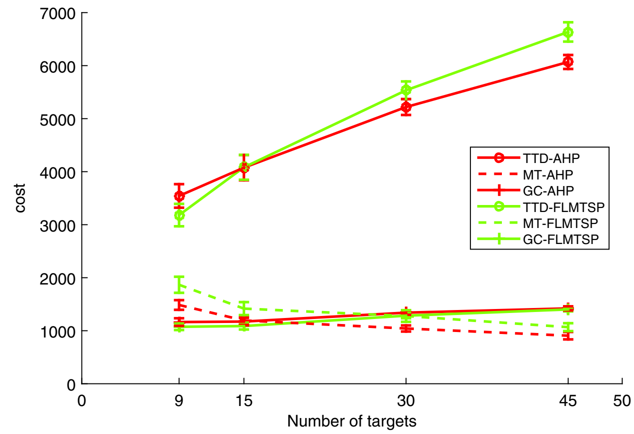

Fig. 5 Comparison between AHP-based solution and FL-MTSP [39]

figuration of a number of robots and number of targets, we generate randomly 30 scenarios and then we plot the mean of the obtained results from these scenarios.

Figure 4 presents a comparison between the market-based, RTMA-based and balanced-based approaches used for the solution and the impact of the improvement phase. The market-based approach decreases the global cost as compared to the balanced-based and RTMA-based approaches, especially in large scenarios. For example, in the case of 15 robots and 45 targets, the gap between the market-based approach and the other two approaches is in the range of [5%, 30%].

In addition, we observe that the improvement phase significantly minimizes the global cost, especially for the balanced-based approach where the reduction is around 19% in the case of nine targets and 30% in the case of 45 targets. For the market-based approach, the enhancement of the improvement phase did not exceed 11%. This demonstrates the feasibility of this approach to solve the MTSP.

5.2 Comparison with multi-objective solutions

5.2.1 Comparison with the FL-MTSP solution

Overview of the FL-MTSP The FL-MTSP approach [39] uses the fuzzy logic algebra to combine two objectives: the total traveled distance by all the robots and the maximum traveled distance by any robot. The solution consists of two phases: the assignment phase and the tour construction phase. In the first phase, the inputs of the fuzzy logic system were computed. Then, the output of the fuzzy logic system is used to assign the targets to the robots. Each target will be assigned to the robot with the minimum output value. After allocating all targets, an improvement process starts. If the number of targets won by a robot is larger than the ratio of the number of targets to the number of robots, the farthest target is selected and added to the nearest robot. For the tour construction phase, a GA TSP solver [21] was used. A detailed description of the FL-MTSP is provided in [39].

Simulation Setting To compare the AHP-based solution with the FL-MTSP approach, we adopted the same test problem described in Sect. 5.1.

Results Figure 5 shows the comparison of the AHP-based approach and the FL-MTSP approach [39]. We mention that the FL-MTSP solution considers only two objectives: the total traveled distance and the maximum distance. For the comparison, the global cost of the FL-MTSP is computed as the weighted sum of these objectives. It is clear from the figure that the gap between the two approaches in terms of global cost is very small. This is due to the fact that the FL-MTSP and AHP-based approaches consider the optimization of multiple objectives when solving the MTSP. Also, the TTD and the MT of our solution were decreased in comparison with the FL-MTSP solution. For example, in the case of 45 targets, the TTD is reduced by around 9%. This means that our solution provides a better trade-off in satisfying the application objectives. Also, the quality of the solution becomes better with the increase in the number of objectives.

5.2.2 Comparison with the clustering market-based solution (CM-MTSP)

Overview of the CM-MTSP The CM-MTSP solution [40] is a hybrid approach that combines a clustering technique with an auction process with the goal of minimizing the total traveled distance and the maximum traveled distance and the mission time. The algorithm includes three steps: a clustering step, an auction-based step, and an improvement step. The clustering step consists in grouping the targets into n clusters. For that purpose, the K-means technique was used [8]. Note that the

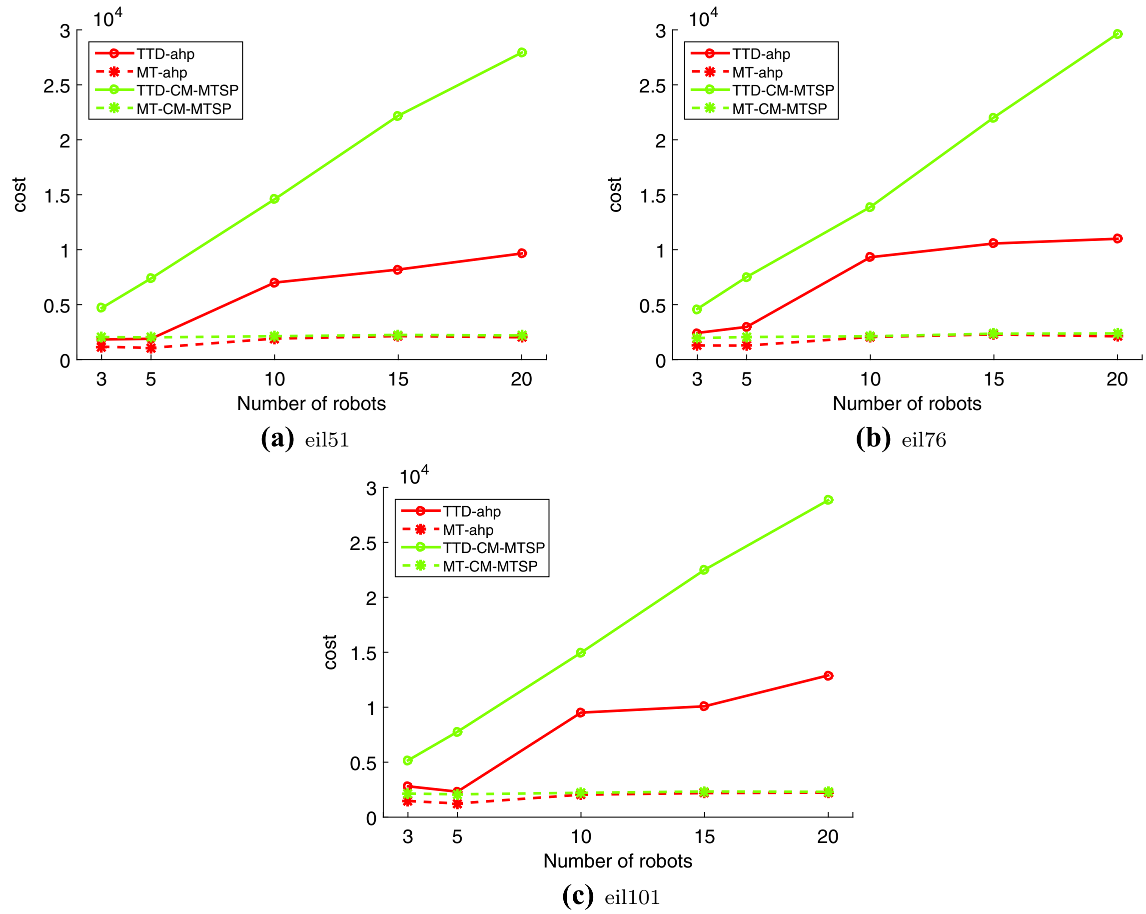

Fig. 6 Comparison results of the AHP-based solution and CM-MTSP solution in terms of TTD and MT

number of clusters is equal to the number of robots and the number of targets in each cluster is equal as much as possible.

In the auction-based step, the robots compete to take the cluster with the minimum cost. The clusters are announced one by one, and each robot bids for each cluster separately. The bid cost is defined as the time necessary to reach all the target locations into the cluster starting from and ending at the initial position. In the case where a robot has already won a cluster, it has the possibility to bid for another cluster with a lower cost. A server unit is responsible to decide which robot to assign to which cluster. It evaluates all the bids received from the robots and selects the winner.

After allocating all clusters, the server evaluates the whole assignment and tries to optimize the solution. The improvement is achieved by exchanging clusters between robots in the case where the mission time and the maximum traveled distance will be minimized.

Simulation Setting To compare the performance of both solutions AHP and the CM-MTSP, we selected six instances (eil51, eil76, eil101, berlin52, rat99, and kroA200) from TSPLIB (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/) with a different number of cities. In our performance evaluation study, these cities are considered as the targets (i.e., 51, 76, 101, 52, 99, and 200 targets). The number of robots is randomly generated and varies depending on the scenario [3, 5, 10, 15, 20], which gives a total of 30 instances. The obtained results have been averaged over ten independent runs. In each run, we randomly generated the depots' positions of the robots. We mention that each robot has its own depot.

Results In Fig. 6, we show the obtained results for the *eil51*, *eil76*, and *eil101* instances in terms of TTD and MT. It is clearly shown that the gap between the AHP-based solution and the CM-MTSP solution in terms of TTD is large especially when the number of robots is large. For example, for instance *eil51*, the gap between the two solutions increases from around 60% in the case of three robots to around 65% in the case of 20 robots. This is due to the fact that in the CM-MTSP solution, all the robots are involved and assigned to clusters, whereas in the AHP-based solution, not all robots are assigned to targets. This will significantly reduce the TTD especially in the cases of a large number of robots. For the

Table 3 The total traveled distance obtained by the two algorithms on the benchmark problems

No	Instances	n	m	AHP		CM-MTSP	
				$\text{avg}_{n,m}$	$\text{best}_{n,m}$	$\text{avg}_{n,m}$	$\text{best}_{n,m}$
1	berlin52	52	3	10,305.9	9380.2	**9397.26**	**8770.8**
2		52	5	**10,940.4**	**9777.7**	11,384.1	10,221
3		52	10	**11,574.5**	**9407.4**	15,083	12,830
4		52	15	**11,117.2**	**9872.7**	19,242.8	15,429
5		52	20	**10,292.9**	**8920.7**	20,910.3	16,611
6	rat99	99	3	**4466.71**	**1518.6**	5111.38	3877.2
7		99	5	**5430.91**	**1250.3**	7354.08	5414.1
8		99	10	**11,020.6**	**7343.6**	13,533.3	10,664
9		99	15	**11,260.9**	**7467.9**	19,770.8	15,856
10		99	20	**11,941**	**8928**	27,318.7	19,147
11	kroA200	200	3	39,371.6	37,773	**36,769.5**	**36,029**
12		200	5	46,432.8	43,463	**43,686.9**	**39,137**
13		200	10	**56,051.1**	52,485	58,479.5	**52,015**
14		200	15	**63,950.6**	**58,057**	76,035.8	68,472
15		200	20	**65,906.5**	**60,584**	91,479.9	85,151

Table 4 The maximum traveled distance obtained by the two algorithms on the benchmark problems

No	Instances	n	m	AHP		CM-MTSP	
			$\text{avg}_{n,m}$	$\text{best}_{n,m}$	$\text{avg}_{n,m}$	$\text{best}_{n,m}$	
1	berlin52	52	3	**3565.48**	**3152.3**	3920.48	3346.2
2		52	5	2957.42	**2614.3**	**2926.52**	2622.6
3		52	10	3344.65	2078	**2319.31**	**1871.9**
4		52	15	2629.73	1954.8	**2091.23**	**1735**
5		52	20	2176.5	1722.6	**1833.73**	**1588.7**
6	rat99	99	3	**1740.72**	**1348**	2105.1	1619.5
7		99	5	**1640.6**	**1199**	1929.89	1360
8		99	10	**1989.53**	**1539.9**	2005.1	1640
9		99	15	2148.78	1855.9	**2044.79**	**1779.4**
10		99	20	2171.34	1702.1	**2118.46**	**1687.8**
11	kroA200	200	3	13,461.2	12,920	**12,673.8**	**12,486**
12		200	5	**11,164.5**	**10,071**	11,183	10,413
13		200	10	10,082	9364.3	**8705.3**	**7779.7**
14		200	15	9328.12	8818.9	**7578.28**	**6553.7**
15		200	20	8814.18	8428.4	**7228.01**	**6553.7**

MT, the gap between the two solutions is very small. This is because both solutions considered the MT as an objective to be optimized.

In addition, we noticed that, for the AHP-based solution, the gap between the TTD and the MT is large. The solutions with a good value for the MT have a higher TTD. This improves the conflicting nature of the objectives, such that it is not possible to obtain good values for both objectives at the same time, without degrading the value of at least one of the objectives. Moreover, the increase in the TTD with the increase in the number of robots in comparison with the small increase in the MT indicates the balance of workload among the robots. From Fig. 6c, we noticed that our solution decreases the TTD as compared to the CM-MTSP approach for around 45% in the case of three robots and around 55% in the case of 20 robots. This means that the AHP-based solution improves the TTD as compared to the CM-MTSP solution for around 10% when increasing the number of robots. The result of the ten runs of *berlin52*, *rat99*, and *kroA200* is summarized in Tables 3 and 4. We used the $\text{avg}_{n,m}$ to represent the mean result of the runs and the $\text{best}_{n,m}$ to represent the shortest result among the ten runs of each scenario. According to Table 3, we can see that the AHP-based solution has the largest number of the $\text{avg}_{n,m}$ and $\text{best}_{n,m}$. More precisely,

Table 5 Wilcoxon ranking test for the average total traveled distance of both solutions AHP and CM-MTSP

No.	AHP	CM-MTSP	d	Rank	P value
1	10,305.9	9397.26	908.64	3	
2	10,940.4	11,384.1	– 443.7	– 1	
3	11,574.5	15,083	– 3508.5	– 9	
4	11,117.2	19,242.8	– 8125.6	– 10	
5	10,292.9	20,910.3	– 10, 617.4	– 12	
6	4466.71	5111.38	– 644.67	– 2	
7	5430.91	7354.08	– 1923.17	– 4	
8	11,020.6	13,533.3	– 2512.7	– 6	
9	11,260.9	19,770.8	– 8509.9	– 11	
10	11,941	27,318.7	– 15, 377.7	– 14	
11	39,371.6	36,769.5	2602.1	7	
12	46,432.8	43,686.9	2745.9	8	
13	56,051.1	58,479.5	– 2428.4	– 5	
14	63,950.6	76,035.8	– 12, 085.2	– 13	
15	65,906.5	91,479.9	– 25, 573.4	– 15	
					0.0151

Table 6 Wilcoxon ranking test for the average maximum traveled distance of both solutions AHP and CM-MTSP

No.	AHP	CM-MTSP	d	Rank	P value
1	3565.48	3920.48	– 355	8	
2	2957.42	2926.52	30.9	3	
3	3344.65	2319.31	1025.34	12	
4	2629.73	2091.23	538.5	10	
5	2176.5	1833.73	342.77	7	
6	1740.72	2105.1	– 364.38	9	
7	1640.6	1929.89	– 289.29	6	
8	1989.53	2005.1	– 15.57	1	
9	2148.78	2044.79	103.99	5	
10	2171.34	2118.46	52.88	4	
11	13,461.2	12,673.8	787.4	11	
12	11,164.5	11,183	– 18.5	2	
13	10,082	8705.3	1376.7	13	
14	9328.12	7578.28	1749.84	15	
15	8814.18	7228.01	1586.17	14	
					0.0554

there are 12 and 11 benchmark problems where the AHP-based solution generates the shortest $\text{avg}_{n,m}$ and $\text{best}_{n,m}$, respectively. This is consistent with the decision made for the *eil51*, *eil76* and *eil101* instances. According to Table 4, there are 5 and 6 benchmark problems where the AHP-based solution generates the shortest $\text{avg}_{n,m}$ and $\text{best}_{n,m}$, respectively. This is due to the fact that the CM-MTSP algorithm generates m tours and each salesman is assigned to a tour. This result is also consistent with the results obtained for the *eil51*, *eil76* and *eil101* instances. Furthermore, we have employed the Wilcoxon ranking test [10] to prove that there is a significant difference between the average total distance obtained by the AHP-based solution and the CM-MTSP solution. After computing the difference d between each pair, we rank all of them regardless of their sign.

These results are listed in Table 5. As we can see, P value (i.e., the probability of observing a test statistic) is small than the threshold value fixed to 0.05. We deduce that the difference between the solutions is significant. Table 6 shows the Wilcoxon ranking test for the maximum traveled distance and demonstrates that the maximum distance obtained by the AHP and the CM-MTSP is close (i.e., P value $= 0.0554 > 0.05$).

5.2.3 Comparison with balanced multi-robot task allocation (BMRTA) [11]

Overview of the BMRTA The solution is based on the use of the clustering method with an auction process. The objectives are to minimize the distance traveled by all robots and equally balance the workload between robots. The solution is very simple. First, the tasks are decomposed into clusters using K-means algorithm. Second, the possible number of combinations of robots to win clusters is obtained. Then, the total cost for each robot to perform any cluster is computed using the following equation:

$$\text{Total cost (TC)} = \text{Travel cost}\left(\sum C_{i,j}\right) + \text{Idle cost (IC)} \tag{14}$$

The travel cost $C_{i,j}$ is the cost needed to reach location j from location i. The idle cost (IC) is the summation of the difference of travel cost between every two robots. The assignment of clusters to robots is based on the least total cost.

Simulation Setting We have tested our AHP-based approach using the benchmark data set A-n32-K5.vrp (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/). The number of targets is set to 32. In this simulation, we used two robots with the following positions: (20, 20) and (80, 20). In [11], two analyses were performed. The first one comprises two robots and two clusters. It will be referred to as BMRTA-2C. The second analysis involves two robots and three clusters. It will be referred to as BMRTA-3C. For the AHP-based approach, we compute the total cost as it is mentioned in Eq. (14).

Results Table 7 reports the results obtained of the BMRTA-2C, BMRTA-3C, and AHP-based approaches in terms of travel cost, idle cost, and total cost. Data for BMRTA-2C and

Table 7 Comparison results of the BMRTA-2C, BMRTA-3C, and AHP-based approaches

	Travel cost	Idle cost	Total cost
BMRTA-2C	492.595	53.885	546.48
BMRTA-3C	504.696	59.04	563.736
AHP-based approach	527.55	10	537.55

BMRTA-3C are taken from [11]. Table 7 shows the superiority of the AHP-based approach over the BMRTA approach in terms of total cost. This is due to the fact that our approach allocates targets one by one to the robot with the lowest cost, in contrast to the BMRTA solution, that groups the targets into clusters and then allocates each cluster to the best robot. We noticed from the table that the travel cost of the BMRTA is decreased in comparison with the AHP-based approach, but the gap is very small (4% for BMRTA-3C and 7% for BMRTA-2C) in comparison with the gap of the idle cost. The smaller value obtained for the idle cost proves that our approach balances the workload between the robots much better than the BMRTA solution.

5.3 Lessons learned

We have learned several lessons from the simulation study. First, we observe that centralized approaches are not tractable for a large-scale system where the number of robots and targets is very large. In such case, distributed approaches are more appropriate. We also learned that clustering-based approaches are not efficient, especially in the case of large-scale system where the targets and robots are far from one another, as this will lead to a decrease in system performance.

6 Computational complexity

In this section, we compute the computational complexity of the AHP-based approach. The complexity of each of the market-based, balanced-based, and RTMA-based algorithms is given as follows:

- The complexity of bidding for each target by the m robots is $O(m)$.
- The complexity of computing the global cost for each robot is $O(m)$.
- The complexity of allocating a target is $O(2 * m) = O(m)$.
- Thus, the complexity of allocating all the n targets is $O(n * m)$.

The complexity of the improvement phase is $O(n^2)$.

7 Conclusion

To solve the multi-objective MTSP, we proposed an AHP-based solution where each objective has a specific weight. The aim of our work is to find a solution that simultaneously optimizes three objectives: the total traveled distance, the maximum tour length, and the deviation rate. We designed three different approaches: market-based, RTMA-based, and balanced-based. From the simulation study, we observed that in most cases the market-based approach generates the best solution. Furthermore, our comparison of the AHP-based solution with existing multi-objective approaches shows that our solution outperforms the FL-MTSP and the CM-MTSP solutions, and provides a good trade-off between the objectives.

Acknowledgements The authors would like to thank the Center of Excellence and the Robotics and Internet-of-Things Unit at Prince Sultan University, Saudi Arabia, for their support of this work. The work is also supported by Gaitech Robotics.

References

1. Alexis K, Darivianakis G, Burri M, Siegwart R (2016) Aerial robotic contact-based inspection: planning and control. Auton Robots 40(4):631–655. https://doi.org/10.1007/s10514-015-9485-5
2. Bektas T (2006) The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34(3):209–219
3. Bolaños R, Echeverry M, Escobar J (2015) A multiobjective non-dominated sorting genetic algorithm (NSGA-II) for the multiple traveling salesman problem. Dec Sci Lett 4(4):559–568
4. Brown EC, Ragsdale CT, Carter AE (2007) A grouping genetic algorithm for the multiple traveling salesperson problem. Int J Inf Technol Dec Mak 6(02):333–347
5. Carter AE, Ragsdale CT (2006) A new approach to solving the multiple traveling salesperson problem using genetic algorithms. Eur J Oper Res 175(1):246–257. https://doi.org/10.1016/j.ejor.2005.04.027
6. Chaari I, Koubaa A, Bennaceur H, Trigui S, Al-Shalfan K (2012) Smartpath: a hybrid ACO-GA algorithm for robot path planning. In: 2012 IEEE congress on evolutionary computation, pp 1–8. https://doi.org/10.1109/CEC.2012.6256142
7. Chaari I, Koubaa A, Trigui S, Bennaceur H, Ammar A, Al-Shalfan K (2014) Smartpath: an efficient hybrid aco-ga algorithm for solving the global path planning problem of mobile robots. Int J Adv Robot Syst 11(7):94. https://doi.org/10.5772/58543
8. Chan ZS, Collins L, Kasabov N (2006) An efficient greedy k-means algorithm for global gene trajectory clustering. Expert Syst Appl 30(1):137–141. https://doi.org/10.1016/j.eswa.2005.09.049
9. Cheikhrouhou O, Koubaa A, Bennaceur H (2014) Move and improve: a distributed multi-robot coordination approach for multiple depots multiple travelling salesmen problem. In: 2014 IEEE international conference on autonomous robot systems and competitions (ICARSC), pp 28–35. https://doi.org/10.1109/ICARSC.2014.6849758
10. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

11. Elango M, Nachiappan S, Tiwari MK (2011) Balancing task allocation in multi-robot systems using k-means clustering and auction based mechanisms. Expert Syst Appl 38(6):6486–6491. https://doi.org/10.1016/j.eswa.2010.11.097
12. Falkenauer E (1992) The grouping genetic algorithms widening the scope of the GAs, jorbel. Belg J Oper Res Stat Comput Sci 33:79–102
13. Ghafurian S, Javadian N (2011) An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. Appl Soft Comput 11(1):1256–1262. https://doi.org/10.1016/j.asoc.2010.03.002
14. Gutin G, Punnen AP (2006) The traveling salesman problem and its variations, vol 12. Springer, New York
15. Heap B, Pagnucco M (2012) Repeated sequential auctions with dynamic task clusters. In: Proceedings of the twenty-sixth AAAI conference on artificial intelligence, AAAI'12. AAAI Press, pp 1997–2002. http://dl.acm.org/citation.cfm?id=2900929.2901010
16. Helsgaun K (2012) LKH. http://www.akira.ruc.dk/~keld/research/LKH/. Accessed 2012
17. Holland JH (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, Ann Arbor
18. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department
19. Ke L, Zhang Q, Battiti R (2013) MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and antcolony. IEEE Trans Cybern 43(6):1845–1859. https://doi.org/10.1109/TSMCB.2012.2231860
20. Király A, Abonyi J (2011) Optimization of multiple traveling salesmen problem by a novel representation based genetic algorithm. Springer, Berlin, pp 241–269. https://doi.org/10.1007/978-3-642-21705-0_9
21. Kirk J (2011) Traveling-salesman-problem-genetic-algorithm. http://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm. Accessed 11 July 2011
22. 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. Accessed 15 June 2011
23. Kivelevitch E, Cohen K, Kumar M (2013) A market-based solution to the multiple traveling salesmen problem. J Intell Robot Syst 72(1):21–40. https://doi.org/10.1007/s10846-012-9805-3
24. 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, pp 19–57
25. Li J, Sun Q, Zhou M, Dai X (2013) A new multiple traveling salesman problem and its genetic algorithm-based solution. In: 2013 IEEE international conference on systems, man, and cybernetics, pp 627–632. https://doi.org/10.1109/SMC.2013.112
26. Liu W, Li S, Zhao F, Zheng A (2009) An ant colony optimization algorithm for the multiple traveling salesmen problem. In: 2009 4th IEEE conference on industrial electronics and applications, pp 1533–1537. https://doi.org/10.1109/ICIEA.2009.5138451
27. Lust T, Teghem J (2010) The multiobjective traveling salesman problem: a survey and a new approach. Springer, Berlin, pp 119–141. https://doi.org/10.1007/978-3-642-11218-8_6
28. Mehrabian A, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inform 1(4):355–366. https://doi.org/10.1016/j.ecoinf.2006.07.003
29. Miettinen K (1999) Nonlinear multiobjective optimization. Springer, New York
30. Peng W, Zhang Q, Li H (2009) Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem. Springer, Berlin, pp 309–324. https://doi.org/10.1007/978-3-540-88051-6_14
31. Punnen AP (2007) The traveling salesman problem: applications, formulations and variations. Springer, Boston, pp 1–28. https://doi.org/10.1007/0-306-48213-4_1
32. Saaty R (1987) The analytic hierarchy process: what it is and how it is used. Math Model 9(3–5):161–176. https://doi.org/10.1016/0270-0255(87)90473-8
33. Shim VA, Tan KC, Cheong CY (2012) A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem. IEEE Trans Syst Man Cybern Part C (Appl Rev) 42(5):682–691. https://doi.org/10.1109/TSMCC.2012.2188285
34. Shim VA, Tan KC, Tan KK (2012) A hybrid estimation of distribution algorithm for solving the multi-objective multiple traveling salesman problem. In: 2012 IEEE congress on evolutionary computation, pp 1–8. https://doi.org/10.1109/CEC.2012.6256438
35. Shuai Y, Bradley S, Shoudong H, Dikai L (2013) A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. Eur J Oper Res 228:72–82
36. Singh A, Baghel AS (2009) A new grouping genetic algorithm approach to the multiple traveling salesperson problem. Soft Comput 13(1):95–101. https://doi.org/10.1007/s00500-008-0312-1
37. Singh H, Kaur R (2013) Resolving multiple traveling salesman problem using genetic algorithms. Int J Comput Sci Eng 3(2):209–212
38. Singh S, Lodhi EA (2014) Comparison study of multiple traveling salesmen problem using genetic algorithm. Int J Comput Sci Netw Secur (IJCSNS) 14(7):107–110
39. Trigui S, Cheikhrouhou O, Koubaa A, Baroudi U, Youssef H (2016) FL-MTSP: a fuzzy logic approach to solve the multi-objective multiple traveling salesman problem for multi-robot systems. Soft Comput. https://doi.org/10.1007/s00500-016-2279-7
40. Trigui S, Koubaa A, Cheikhrouhou O, Qureshi B, Youssef H (2016) A clustering market-based approach for multi-robot emergency response applications. In: 2016 International conference on autonomous robot systems and competitions (ICARSC), pp 137–143. https://doi.org/10.1109/ICARSC.2016.14
41. Venkatesh P, Singh A (2015) Two metaheuristic approaches for the multiple traveling salesperson problem. Appl Soft Comput 26:74–89. https://doi.org/10.1016/j.asoc.2014.09.029
42. Viguria A, Howard AM (2009) An integrated approach for achieving multirobot task formations. IEEE/ASME Trans Mechatron 14(2):176–186. https://doi.org/10.1109/TMECH.2009.2014056
43. Wang X, Liu D, Hou M (2013) A novel method for multiple depot and open paths, multiple traveling salesmen problem. In: 2013 IEEE 11th international symposium on applied machine intelligence and informatics (SAMI), pp 187–192. https://doi.org/10.1109/SAMI.2013.6480972
44. Xu Z, Li Y, Feng X (2008) Constrained multi-objective task assignment for UUVs using multiple ant colonies system. In: 2008 ISECS international colloquium on computing, communication, control, and management, vol 1, pp 462–466. https://doi.org/10.1109/CCCM.2008.318
45. Yong W (2015) Hybrid max–min ant system with four vertices and three lines inequality for traveling salesman problem. Soft Comput 19(3):585–596. https://doi.org/10.1007/s00500-014-1279-8
46. Yousefikhoshbakht M, Didehvar F, Rahmati F (2013) Modification of the ant colony optimization for solving the multiple traveling salesman problem. Rom Acad Sect Inf Sci Technol 16(1):65–80