Research Article

Hybrid Optimization-Based Approach for Multiple Intelligent Vehicles Requests Allocation

Ahmed Hussein , Pablo Marín-Plaza , Fernando García, and José María Armingol

Intelligent Systems Lab (LSI) Research Group, Universidad Carlos III de Madrid (UC3M), Calle Butarque 15, Leganés, 28911 Madrid, Spain

Correspondence should be addressed to Ahmed Hussein; ahussein@ing.uc3m.es

Received 17 November 2017; Accepted 29 January 2018; Published 11 March 2018

Academic Editor: Aboelmaged Noureldin

Copyright © 2018 Ahmed Hussein et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Self-driving cars are attracting significant attention during the last few years, which makes the technology advances jump fast and reach a point of having a number of automated vehicles on the roads. Therefore, the necessity of cooperative driving for these automated vehicles is exponentially increasing. One of the main issues in the cooperative driving world is the Multirobot Task Allocation (MRTA) problem. This paper addresses the MRTA problem, specifically for the problem of vehicles and requests allocation. The objective is to introduce a hybrid optimization-based approach to solve the problem of multiple intelligent vehicles requests allocation as an instance of MRTA problem, to find not only a feasible solution, but also an optimized one as per the objective function. Several test scenarios were implemented in order to evaluate the efficiency of the proposed approach. These scenarios are based on well-known benchmarks; thus a comparative study is conducted between the obtained results and the suboptimal results. The analysis of the experimental results shows that the proposed approach was successful in handling various scenarios, especially with the increasing number of vehicles and requests, which displays the proposed approach efficiency and performance.

1. Introduction

Self-driving cars could revolutionize how people get around; with the introduction of automation into roads, it will contribute to solving the issues related to the traffic accidents, congestion, and energy consumption. The driverless vehicle technologies are advancing on a great scale, specifically the multiple sensors fusion techniques, deep learning, and computational intelligence. Together, they enable these vehicles to understand the nearby surroundings and take appropriate actions to navigate on their own from one point to another. The technology does not stop at vehicle being fully automated; however multiple of them cooperate together to achieve smoother driving operation [1–3].

Researchers in the Intelligent Transportation Systems (ITS) field are investing more time on the self-driving cars research and development [4–6]. However, during the last decade, Multirobot Systems (MRS) fell under the research attention of the ITS community. This increased interest comes from the significant advantages and higher potential provided by MRS over single robot systems. MRS can be simply

understood to be a group of robots cooperating together for accomplishing a certain task or mission [7, 8]. With reference to the literature review survey, the coordination and cooperation among multiagent systems can be modeled as Multirobot Task Allocation (MRTA) problem. MRTA is a NP-hard problem, which concerns the use of the available resources in an efficient manner. Accordingly, the decision of which robot will do which task strongly affects the performance of the system [9, 10].

Accordingly, this paper introduces a hybrid optimization-based approach to solve the MRTA problem. This approach can coordinate among several intelligent vehicles and aid in the cooperation for better overall performance. The algorithm is implemented on automated golf carts and tested in off-road environments in real-life scenarios. Moreover, in order to prove the proposed solution efficiency, several experiments are carried out over well-known benchmarks. The benchmarks optimal solution is obtained after hundreds of hours of computations. The proposed approach was able to obtain a near-optimal solution in a matter of seconds. This proves the proposed approach high performance.

The remainder of the paper is organized as follows: Section 2 presents the previous work in the field of multiple automated vehicles and the allocation problem, followed by Section 3, which introduces the problem formulation, solution constraints, objective function, and the proposed solving hybrid approach. In Section 4, the used platform is described, along with the experimental setup in the designed architecture, selected scenarios, and the evaluation metrics. Results and discussion are illustrated in Section 5. Finally, the conclusion and future work are summarized in Section 6.

2. Related Work

Autonomous cars are becoming more frequent in nowadays scientific and industrial context, since Google launched their driverless car project in 2011 [4]. From then onwards, many other approaches proposed different architectures and solutions, all of them moving towards the development of the autonomous vehicle. For instance, Mercedes with the Bertha project proved the viability of autonomous vehicles in German roads based on advance sensing capabilities [11]. The V-Charge project researches in the direction of allowing automated valet parking for self-driving cars [12]. Moreover, several vehicle manufacturers have proposed different solutions in the field of autonomous vehicles which are close to markets, such as BMW and Audi [13], Mercedes-Benz [14], and Volvo [15]. Furthermore, several other proposals offer the possibility of including autonomous vehicles in public transportation systems [5, 6].

The self-driving technology is advancing rapidly, but in order to safely deploy vehicles on public roads, cooperation with other road users is mandatory. This cooperation would allow the safe interaction with other vehicles, whether with a human driver or driverless. Although some of the proposed solutions already handle the presence of other vehicles in the road [16–18], by handling static and dynamic obstacles and adapting the trajectory accordingly, the cooperative driving is not yet achieved. Cooperative driving demands the information exchange and control strategies deployed in all the vehicles involved following a two-way communication. In this sense, during recent years many works have addressed this topic, trying to provide solutions based on different configuration and solutions. The Grand Cooperative Driving Challenge 2016 was created with the purpose of boosting cooperative automated vehicles in the form of a cooperative based competition [19]. On the one hand, authors in [20] presented an auction-based cooperative control for autonomous vehicles; on the other hand, authors in [21] proposed an approach to enhance common motion planning algorithms; this proposal allows cooperation with human-driven vehicles. Additionally, a novel concept is presented, based on a centralized strategy, using maneuver templates, which are formalized collaborative maneuvers, to select cooperative driving strategies [22]. All this work proves the importance of collaboration and communication among vehicles to allow a safe and efficient deployment of autonomous vehicles really in driving scenarios.

Since cooperation is essential among multiple vehicles on the road, coordination becomes the first issue to solve. In the MRS, the question of which robot is going to execute which task is answered through task allocation problem, which is commonly known as MRTA problem. By reviewing the literature, it was found that different optimization approaches have been used in order to solve the general task allocation problem and were also used in order to solve the MRTA problem. In [23], a mixed integer linear programming optimization approach was used in order to allocate heterogeneous robots for maximizing the coverage area of the regions of interest. In [24], a simulated annealing approach was used to solve the allocation of MRS through formulating the MRTA problem as multi-Traveling Salesmen Problem (mTSP). Then in [25], a market-based approach was proposed to solve the MRTA problem for heterogeneous robots and task formulated as mTSP. Moreover, the task allocation problem was also solved using hybrid optimization approaches such as the tabu search with random search method in [26] and tabu search with noising method in [27].

3. Methodology

In this section, the MRTA problem modeling and formulation are introduced, followed by the selected objective cost function. Afterward, the proposed algorithm is described, highlighting the main contribution. MRTA problem [7] is formulated to allocate multiple robots, in this case, vehicles, to numerous tasks, in this case, requests. The procedures are summarized as follows:

- (1) Given a set of n vehicles, $V = \{V_1, V_2, \dots, V_n\}$.
- (2) Given a set of m requests, $R = \{R_1, R_2, \dots, R_m\}$.
- (3) Allocation of the requests to the vehicles occurs, $A : R \rightarrow V$.
- (4) Output set S is the best allocation of the requests to the vehicles:

$$S = \{(V_1 \ R_1) \ (V_2 \ R_2) \ \dots \ (V_k \ R_k)\} \quad (1)$$

for $1 \leq k \leq m$.

- (5) Allocation S minimizes or maximizes a certain objective function in order to get the best performance of the system.

3.1. Problem Formulation. A variant of mTSP is used to model MRTA problem. In the standard mTSP formulation, n nodes are defined with the edges distances and m salesmen are known. The salesmen are required to cover all the available nodes and return back to their starting node, such that each salesman makes a round trip. The mTSP can be formally defined on a graph $G = (V, E)$ where V is the set of n nodes and E is the set of edges. Let $c = (c_{ij})$ be the distance matrix associated with E . Assuming the more general case which is

an asymmetric mTSP, thus $c_{ij} \neq c_{ji} \forall (i, j) \in E$. The mTSP can be formulated as follows [9]:

$$x_{ij} = \begin{cases} 1 & \text{if edge } (i, j) \text{ is used in the tour} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$\text{minimize } \sum_{i=1}^n \sum_{j=1}^n c_{ij} \times x_{ij} \quad (3)$$

$$\sum_{j=2}^n x_{1j} = m \quad (4)$$

$$\sum_{j=2}^n x_{j1} = m \quad (5)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 2, \dots, n \quad (6)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 2, \dots, n \quad (7)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in E \quad (8)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |\text{SubTour}| - 1, \quad (9)$$

$$\forall \text{SubTour} \subseteq V \setminus \{1\}, \text{SubTour} \neq \emptyset,$$

where (3) represents the objective function which is the summation of the total distance traveled, (4) and (5) ensure that exactly m salesmen departed their starting node and returned back. Equations (6), (7), and (8) are the usual assignment constraints. Finally, (9) is the subtour elimination constraint.

The proposed formulation is extended and adapted to the problem, where vehicles represent the salesmen and requests represent the cities. Therefore, vehicles capabilities and requests requirements are considered and included in the mTSP implementation. The added features of the vehicles are capacity, velocity, energy, efficiency, and sensors, and the ones for the requests are timestamp, priority, and passengers.

Where the vehicle capacity is the maximum number of passengers that it can hold, the velocity is a representation of the maximum speed it can reach, the energy is a representation of the battery level, the efficiency is a representation of the aging factor, and finally the sensors are a set of on-board devices to consider the vehicles as heterogeneous robots. On the other hand, the request timestamp is the date and time of the request creation, the priority is a representation of the request urgency based on the request type, and passengers are a representation of the number of users for the request.

3.2. Solution Construction. The solution is constructed as a set that includes a list of all vehicles in the system, followed by their assigned requests. The order of the list defines the quality of the solution according to the objective function. For example, for a problem with three vehicles and five requests,

one of the possible candidate solutions is represented as follows:

$$\begin{aligned} &\text{Candidate Solution} \\ &= [V1 \ R1 \ R2 \ V2 \ R3 \ R4 \ V3 \ R5]. \end{aligned} \quad (10)$$

Any combination of this list presents another candidate solution, and since the mTSP is a permutation problem, therefore the order of this list affects the quality of the solution as per the objective function. The order implies that each vehicle is going to execute all requests succeeding it. In candidate solution (10), requests 1 and 2 are executed by vehicle 1, requests 3 and 4 are executed by vehicle 2, and finally request 5 is executed by vehicle 3.

3.3. Objective Function. Although the MRTA problem is formulated as an instance of the mTSP, the same objective function of the mTSP previously explained in (3) cannot be straightforwardly used as the objective function for the MRTA problem. Therefore, some variations had to be introduced to the objective function of the mTSP in order to be effectively used for the MRTA problem [28].

There are three main variations of the MRTA problem objective function compared with the mTSP objective function. First, it is a multiobjective function instead of a single objective function, second, the variable to be minimized is the time rather than the distance, and, third, the time of the maximum subtour is minimized rather than the total time, thus dealing with it as a MinMax problem. Then, for k subtours and r requests in each subtour, the total traveling time is calculated as follows:

$$\begin{aligned} &f(\mathbf{x}) \\ &= \arg \max_{j \in \{1, 2, \dots, k\}} \frac{\sum_{i=1}^{r-1} \text{distance}(\text{subtour}_{j_i}, \text{subtour}_{j_{i+1}})}{\text{vehicle velocity}_j} \\ &+ \arg \max_{j \in \{1, 2, \dots, k\}} \frac{\sum_{i=1}^r \text{execution time}(\text{subtour}_{j_i})}{\text{vehicle efficiency}_j}. \end{aligned} \quad (11)$$

3.4. Solution Constraints. Although any arrangements of the vehicles and requests solution set are considered as a candidate solution for the mTSP problem, this does not guarantee that this solution is feasible for the MRTA problem. Therefore, few constraints are applied to the obtained solution, to ensure its validity and feasibility, which means checking whether the vehicle capabilities and request requirements are matching.

The first constraint is related to the capacity; for example, a vehicle with a maximum capacity of 4 passengers cannot handle a request of 6 passengers in one go; thus the request is decomposed into several requests and reallocated accordingly. The next constraint is the energy; the vehicle battery level is always taken into consideration before the final allocation of the request, since if the vehicle does not have sufficient energy for a specific request, it is reallocated to another vehicle. Another constraint is the priority level of the request, which implies that some requests, maintenance request, for instance, must be executed first. Last but not least,

there is a constraint related to the mounted on-board sensors, which check if the request requires the presence of a specific sensor in the vehicle.

These applied constraints strongly affect the search space of the problem through decreasing the number of candidate solutions that can be accepted as feasible solutions. One may think that this decrease of the number of feasible solutions among the candidate solutions may make it easier for the applied algorithm to find the best solution than the case without the constraints. However, these applied constraints, in fact, make it more difficult and more time consuming to find the best solution. This is mainly because the algorithm will be visiting a large number of solutions that are candidate solutions of the mTSP but are not feasible to solve the MRTA problem.

3.5. Proposed Algorithm. The proposed solution is designed as metaheuristic optimization algorithm. It is a hybrid approach, which is based on both trajectory-based and population-based techniques. The trajectory-based one is the family of optimization techniques that use a single solution throughout the algorithm, in order to find the near-optimal solution. While the population-based one is the family of optimization techniques that iteratively transforms a set of solutions, in order to generate a new population of solutions with the aim of finding the near-optimal solution [29].

On the one hand, SA was selected as a trajectory-based approach, where the neighboring operator is randomly chosen at each iteration for diversity. The mutation operators are swapping, deletion and insertion, inversion, and scrambling. On the other hand, genetic algorithm (GA) was selected as the population-based approach, where the selected mutation operators are the same; however, additional crossover operators are selected, which are partially mapped crossover and order crossover.

The swapping operator chooses two random elements of the solution list and swaps them with each other. Deletion and insertion operator chooses a random element and deletes it from its current position and randomly inserts it in a new position. The inversion operator chooses two random positions and inverts the order of elements between these two positions. The scrambling operator picks two random positions and scrambles the elements between these two positions. Figure 1 illustrates an example of all proposed operators, where positions 1 and 6 are selected as the two random elements. At each iteration, one of the four mutators is randomly chosen in order to generate a neighboring solution of the current solution. The four methods vary in their diversification and intensification effect on the generated neighboring solution.

The crossover operator is the mimicking of the biological recombination between chromosomes, when some portion of the genetic material is swapped between chromosomes producing a new offspring chromosome. On the one hand, in the partially mapped crossover, two points are selected at random in both parents solutions and the offspring is created by exchanging the in-between chromosomes. On the other hand, in the order crossover a portion of one parent is mapped to a portion of the other parent; then from

Candidate solution

V1	R1	R2	V2	R3	R4	V3	R5
----	----	----	----	----	----	----	----

Swapping

V1	V3	R2	V2	R3	R4	R1	R5
----	----	----	----	----	----	----	----

Deletion & insertion

V1	R2	V2	R3	R4	V3	R1	R5
----	----	----	----	----	----	----	----

Inversion

V1	V3	R4	R3	V2	R2	R1	R5
----	----	----	----	----	----	----	----

Scrambling

V1	R3	R4	V3	R2	R1	V2	R5
----	----	----	----	----	----	----	----

FIGURE 1: Example for the mutation operators.

Parent solution 1

V1	R1	R2	V2	R3	R4	V3	R5
----	----	----	----	----	----	----	----

Parent solution 2

V2	R2	R5	V3	R1	R3	R4	V1
----	----	----	----	----	----	----	----

Partially mapped crossover

V3	R5	R2	V2	R3	R4	R1	V1
----	----	----	----	----	----	----	----

Order crossover

V3	R1	R2	V2	R3	R4	V1	R5
----	----	----	----	----	----	----	----

FIGURE 2: Example for the crossover operators.

the replaced portion onwards, the rest is filled up by the remaining genes, where already present genes are omitted and the order one is preserved. Figure 2 illustrates an example of proposed crossover operators, where positions 2 and 5 are selected as the two random elements. At each iteration, one of the two crossover mutators is randomly chosen in order to generate a neighboring solution of the current solution. Additionally, the random choice of the used operator gives the algorithm both the explorative and exploitative features that are useful in escaping local minimum and finding a better solution through searching in the neighbors of elite solutions, respectively.

The algorithm is used to solve the formulated model of the MRTA problem. Inputs are the list of the requests with their requirements, the list of vehicles with their capabilities, and the matrix of the distances between the points of interests of the requests locations. At each iteration of the algorithm, a solution is constructed to be evaluated; initially, it is random. The initial solution set is always filled with random elements of the requests list and the vehicles list until both lists are empty; the only constraint is that the start of the solution must be an element of the vehicles list. After the construction of the initial candidate solution or any other neighboring solution through the algorithm iterations, the feasibility of this solution must be checked, according to the solutions constraints that are previously explained in Section 3.4. As the algorithm progresses, neighboring solutions of the current solution must be generated in order to explore

the search space of the problem. Algorithm 1 presents the proposed algorithm used to solve the MRTA problem in this paper.

Here the parameters can be defined as follows:

- (i) *parentsList* is the list with parents solutions.
- (ii) *childrenList* is the list with children solutions.
- (iii) *nextGenerationList* is the list with next generation solutions.
- (iv) *iterationsNumber* is the number of iterations.
- (v) *elitismPercent* is the elitism percent.
- (vi) *populationSize* is the population size.
- (vii) *initialTemperature* is the initial temperature.
- (viii) *finalTemperature* is the final temperature.
- (ix) *currentTemperature* is the current temperature.
- (x) *iterationsPerTemperature* is the number of iterations per temperature decrement.
- (xi) α is the geometric coefficient.
- (xii) *transitionProbability* is the transition probability.
- (xiii) *currentAllocation* is the current allocation solution.
- (xiv) *currentCost* is the current solution cost.
- (xv) *neighborAllocation* is the neighbor allocation solution.
- (xvi) *neighborCost* is the neighbor solution cost.
- (xvii) *bestAllocation* is the best allocation solution.
- (xviii) *bestCost* is the best solution cost.

4. Experimental Work

In this section, the real-world platform is described, followed by the proposed ROS-based architecture for the multiple vehicle cooperation. Moreover, it presents the selected scenarios along with the chosen evaluation metrics.

4.1. Platform Description. In this work, two automated vehicles were selected as the platforms for the real-world experiments. Figure 3 shows the vehicles; they are part of the Intelligent Campus Automobile (iCab) project. They are electric golf carts, which are modified mechanically and electronically for the purpose of fully automated vehicles.

Each vehicle is equipped with multiple on-board sensors, such as GPS and compass module, stereo-camera, laser-rangefinder, 3D LiDAR, optical encoders, and ultrasonic sensors. All sensors are connected to the on-board embedded computers, which are connected to a 4G router with a constant Internet connection [30].

The on-board embedded computers use Robot Operating System (ROS) architecture to carry out numerous technologies. The architecture is divided into three layers. (1) Low-level commands are executed in the reactive layer, such as communicating with sensors and actuators. (2) Communication between the layers and decomposing the complex task

FIGURE 3: iCab 1 and iCab 2 automated vehicles platforms.

to simple ones are executed in the sequencer layer. (3) Finally, high-level commands are executed in the deliberative layer, which is mainly responsible for environment perception, navigation, planning, localization, and mapping among others [31].

4.2. Communication and Cooperation Architecture. Communication and cooperation among a team of unmanned vehicles is a crucial task. Since in case of long duration missions, or vehicle failure, it is unlikely that a single vehicle is enough to complete the operation [10], therefore, for the communication among vehicles, the *multimaster_fkie* package is used [32]. It combines the required nodes to establish and manage a multimaster network over the implemented ROS-based architectures. In order to use the aforementioned package, vehicles must operate under a common network. Thus, for secure and stable paradigm, a Virtual Private Network (VPN) is used to overlay the commonly available one, which is implemented following the ITS Cooperative ITS (C-ITS) ETSI TR 101 607 V1.1.1 (2013-05) protocol. Detailed description for using the adopted communication schemes is explained in [33].

For the cooperation, *mrtat* node is implemented as the one responsible for handling the task allocation problem. The inputs for the node are vehicle status topic, vehicle pose topic, and list of requests topic. The allocation output is published to the *task_executor* node of each vehicle. The idea of this node is to decompose the requests allocation into executable tasks, according to the vehicles capabilities. The objective is to execute the user transport requests by multiple automated vehicles, using coordination and cooperation mechanisms. Passengers use their own smartphones to create a request by selecting the number of the passengers and the pick-up and drop-off locations. The transport requests are then stored on a webserver, which is accessible by every vehicle in the system [34].

To avoid the centralized paradigm, leader token approach is proposed. Thus, at each iteration, the vehicles run a token selection algorithm, which determines the leader token holder for this loop. The algorithm decision is based on the vehicle status and current computational load. Only the leader vehicle communicates with the webserver for the updates of the requests lists and then publishes it to all other vehicles in the system. Accordingly, this mechanism ensures

Input: Requests list *requests*, Vehicles list *vehicles*, Distances matrix *distances*
Output: Best allocation *bestAllocation*

```

(1) for i ← 1 to populationSize do
(2)   parentsList ← generateValidSolution(requests, vehicles, distances)
(3) end
(4) currentAllocation ← minimumOf(parentsList)
(5) for i ← 1 to iterationsNumber do
(6)   if i ≤ 25% of iterationsNumber then
(7)     elitismPercent = 20%
(8)   else if i > 25% of iterationsNumber AND i ≤ 50% of iterationsNumber then
(9)     elitismPercent = 30%
(10)  else if i > 50% of iterationsNumber AND i ≤ 75% of iterationsNumber then
(11)    elitismPercent = 40%
(12)  else
(13)    elitismPercent = 50%
(14)  end
(15) childrenList ← crossover(least 20% of parentsList)
(16) childrenList ← mutation(top 80% of parentsList)
(17) nextGenerationList ← minimum elitismPercent of parentsList
(18) nextGenerationList ← minimum elitismPercent of childrenList
(19) for j ← 1 to (100% – elitismPercent × 2) of iterationsNumber do
(20)   currentAllocation ← generateValidSolution(requests, vehicles, distances)
(21)   currentCost ← getAllocationCost(currentAllocation)
(22)   bestCost ← currentCost
(23)   while currentTemperature < finalTemperature do
(24)     for i ← 1 to iterationsPerTemperature do
(25)       neighborAllocation ← generateNeighborSolution(currentAllocation)
(26)       neighborCost ← getAllocationCost(neighborAllocation)
(27)       if neighborCost < currentCost then
(28)         currentAllocation ← neighborAllocation
(29)         currentCost ← neighborCost
(30)         if neighborCost < bestCost then
(31)           bestAllocation ← neighborAllocation
(32)           bestCost ← neighborCost
(33)         end
(34)       else
(35)         Generate: random number randomNumber ∈ ]0, 1[
(36)          $transitionProbability = \exp\left(-\frac{neighborCost - currentCost}{currentTemperature}\right)$ 
(37)         if transitionProbability > randomNumber then
(38)           currentAllocation ← neighborAllocation
(39)           currentCost ← neighborCost
(40)         end
(41)       end
(42)     end
(43)     currentTemperature = currentTemperature *  $\alpha$ 
(44)   end
(45)   nextGenerationList ← currentAllocation
(46) end
(47) if minimumOf(parentsList) < currentAllocation then
(48)   currentAllocation ← minimumOf(parentsList)
(49) end
(50) parentsList ← nextGenerationList
(51) end
(52) bestAllocation ← currentAllocation

```

ALGORITHM 1: Proposed hybrid optimization-based algorithm.

the consistency and synchronization of the requests through a distributed paradigm.

4.3. Selected Scenarios. In order to test the proposed approach, several scenarios are selected in both simulation and real-world. The simulation scenarios are selected from well-known benchmarks of mTSP; this is in order to have the optimal cost available for comparison. Each scenario consists of a different number of cities, which are distributed over the environment in various locations.

Figure 4 shows the four selected scenarios, where the top left one is Christofides/Eilon with 51 cities, top right one is Berlin (Groetschel) with 52 locations, bottom left one is Christofides/Eilon with 76 cities, and bottom right one is Rattled Grid (Pulleyblank) with 99 cities [35, 36]. Each scenario has only one depot, represented by the red marker in the graphs.

On the other hand, the real-world scenario was designed in a way to evaluate the functionality of the proposed solution and the architecture in the platforms. The scenario involved three users using the application to create transportation requests. The three users had different starting points and different destinations; moreover the request time was close to others to evaluate how the vehicles are going to respond. Figure 5 shows the environment map with the vehicles and passengers locations, the two vehicles are represented with the golf-cart clip-art, and the passengers representation is marked in three different colors, with a circular shape of the same color for the desired destination. The experiment was video recorded and its results discussion is in the next section.

4.4. Evaluation Metrics. The proposed hybrid optimization-based solution is used to solve each scenario of the MRTA problem and the results are recorded for evaluation. In order to evaluate the quality, two evaluation metrics are introduced, which are the allocation cost and computational time.

The first evaluation metric is the allocation cost of the best allocation found. The allocation cost is calculated based on the objective function. Thus two allocations costs are computed, one is the MinMax cost, which represents the length of the longest subtour in the allocation, and the other is the total overall cost of all subtours.

The second metric is the computational time required by the algorithm to reach the best solution. The timer starts after the databases of the vehicles and requests are read and stops when the algorithm stops; then the elapsed time is reported. Since the computational time calculation is based on the machine, the computer used for all experiments has the specifications in Table 1.

5. Results

In this section, a comparative study is conducted between the proposed approach results and the reference optimal results presented in [35, 36]. The optimal results are obtained for the four well-known benchmarks, which are described in the scenarios subsection. The authors in [35] adjusted the benchmark settings, to have all vehicles located at the depot, where the vehicles are required to visit all locations in the

TABLE 1: System specifications.

Processor	4 cores, 4 threads @3.8 GHz
Memory	16 GB DDR4-2133
Storage	HDD 1 TB 7200 RPM @3 Gb/s

scenario. With the condition that each location is visited exactly once, vehicles must return to the depot afterward.

Table 2 summarizes all the results from both the optimal solutions costs, compared to the obtained solutions costs, in addition to the computational time for the obtained solutions costs. The optimal costs are obtained using IBM CPLEX; it took 96 hrs for the eil-51 dataset, 120 hrs for the berlin-52 dataset, 168 hrs for the eil-76 dataset, and 216 hrs for the rat-99 dataset. On the other hand, the reported values of the obtained costs are the mean of the 25 experiments of each scenario.

These results show that the proposed approach was able to converge to the near-optimal values in much less computational time, which can actually be considered running in real-time. For the MinMax costs, Table 3 shows the deviation errors to the optimal costs, presenting the fact that all errors are less than 10%. However, the more visible contribution is that the error is actually decreasing with the increasing number of vehicles. This proves that the proposed approach is more capable of handling multiple vehicles and obtaining more accurate solutions in all tested benchmarks.

Since there are several possibilities for the solutions permutations, when the MinMax solution is optimized, this does not guarantee the optimization for the total cost as well. However, the proposed approach objective function was designed to take this into consideration. Table 4 shows the deviation errors from the optimal total costs. The same behavior of the MinMax costs is quite obvious, where the more the vehicles introduced to the system, the more capable the proposed approach to find near-optimal allocations. Moreover, in the case of berlin-52 dataset, the proposed approach obtained allocations better than the suboptimal ones in the case of 5 and 7 vehicles.

The conducted comparative study highlighted the high performance of the proposed approach in different scenarios and its scalability in handling multiple vehicles. Therefore, the proposed approach was able to not only obtain near-optimal allocations in much less computational time, but also outperform the CPLEX approach in one of the scenarios, under the same set of constraints and conditions.

Additionally, the real-world experiment was successful in terms of allocation of the requests and the vehicles performance to pick up and drop off the passengers, along with the automated navigation from the starting points to the destinations. The two iCab platforms were connected via the V2V communication and once the first request was created on the webserver, the token-holder vehicle collected the requests and the available vehicles to start the proposed task allocation algorithm. The first request was allocated to iCab-1 and while the vehicle is navigating to the first destination point, the second request came in, at which the allocation algorithm output was to allocate it to iCab-1, since this

FIGURE 4: mTSP selected benchmark scenarios. Red marker is the depot.

TABLE 2: Benchmarks comparative results.

Benchmark	Vehicles	Optimal MinMax	Optimal total	Obtained MinMax	Obtained total	Comp. time (sec)
eil-51	2	222.73	444.33	236.49	470.97	124.86
	3	159.57	477.15	167.11	498.64	137.06
	5	123.96	615.19	129.36	629.64	158.86
	7	112.07	762.83	116.87	764.24	182.67
berlin-52	2	4110.21	8217.94	4315.83	8630.19	131.94
	3	3244.37	9591.15	3387.35	10115.18	140.97
	5	2441.39	12084.90	2509.84	11969.61	163.10
	7	2440.92	16768.79	2491.83	15628.95	194.21
eil-76	2	280.85	561.48	307.68	613.75	281.89
	3	197.34	587.65	210.97	632.07	292.83
	5	150.30	748.43	158.24	772.34	333.45
	7	139.62	964.69	143.74	970.42	386.62
rat-99	2	728.71	1456.95	801.68	1603.31	442.96
	3	587.17	1751.95	636.41	1903.04	495.36
	5	469.25	2336.22	487.71	2399.87	555.85
	7	443.91	3074.30	462.59	3135.78	645.43

FIGURE 5: Environment map with the real-world scenario vehicles and passengers locations.

TABLE 3: Deviation errors to optimal MinMax costs.

Benchmark/vehicles	eil-51	berlin-52	eil-76	rat-99
2 vehicles	6.18%	5.00%	9.55%	10.01%
3 vehicles	4.73%	4.41%	6.90%	8.39
5 vehicles	4.36%	2.80%	5.29%	3.93%
7 vehicles	4.28%	2.09%	2.95%	4.21%

TABLE 4: Deviation errors to optimal total costs.

Benchmark/vehicles	eil-51	berlin-52	eil-76	rat-99
2 vehicles	6.00%	5.02%	9.31%	10.05%
3 vehicles	4.50%	5.46%	7.56%	8.62
5 vehicles	2.35%	-0.95%	3.19%	2.72%
7 vehicles	0.18%	-6.80%	0.59%	2.00%

optimizes the overall solution. Therefore, iCab-1 continues the navigation to the first destination point and then picks up the second request to continue afterward to the second destination point. The last request was communicated while the vehicle is navigating and this time the allocation algorithm assigned it to iCab-2, which was free and able to pick up the passenger. All computations are executed on the on-board vehicles computers in real-time, and since the problem was addressing 2 vehicles and 3 requests in an environment with 12 points of interests, the computational time was approx. 1860 milliseconds for each allocation solution.

6. Conclusion and Future Work

In recent years, the MRS has received a significant consideration from various researchers in the ITS field. This is due to the fact that MRS can improve the concept of cooperative driving of automated vehicles. Moreover, since the number of automated vehicles on the public roads is increasing, the necessity of MRS is also increasing. One of the main challenges for the MRS is the task allocation, to

identify which task should be executed by which vehicle, and it is commonly known as MRTA problem. In this work, the focus on the MRTA problem was from the aspect of transportation requests, where users make requests through a mobile application to be driven from one point to another; then an automated vehicle in the surrounding takes the job and so on. The idea of task allocation can be extended to much more than that to improve the quality of cooperative driving.

The main intentions of this work are to propose a generic approach for solving MRTA problem, which not only should be responsible for providing a solution that is feasible, but also should be optimized. Since the optimized allocation enables the appropriate use of all available resources and thus increasing the overall system performance and decreasing the costs, accordingly, a hybrid metaheuristic optimization-based approach is proposed, which combines both the simulated annealing and the genetic algorithm approaches, to ensure the exploitation and exploration aspects of the search space and obtain near-optimal solutions in less computational time.

Several test scenarios were implemented in order to evaluate the efficiency of the proposed approach. These scenarios are based on well-known benchmarks; thus a comparative study is conducted between the obtained results and the suboptimal results provided in the previous work over the same benchmarks. The analysis of the experimental results shows that the proposed approach was successful in handling various scenarios, especially with the increasing number of vehicles and requests, which displays the proposed approach efficiency and scalability. Additionally, the proposed approach was tested on automated vehicles platforms in a real-world scenario, at which the vehicles performed the assigned tasks flawlessly, which proves the functionality of the approach outside the simulation environment.

While the proposed approach was successful in solving the MRTA problem and optimizing the solutions of the benchmarks in significantly less time, there are some aspects that should be considered for future improvement. The cooperation architecture is designed to handle heterogeneous tasks, which means that tasks have specific requirements that constrain specific vehicles to complete it. Moreover, heterogeneous vehicles, not only in terms of capabilities, but also in terms of type, for instance, carry out experiments with both ground and aerial automated vehicles. Last but not least, it was assumed that the distance matrix includes the shortest free paths between the points of interests for the tasks; however, since the vehicles have already on-board trajectory planning algorithms, the approach should consider incorporating this work in the distance calculation.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper. The iCab platforms, which were utilized in the real-world experiments, are property of Intelligent Systems Lab (LSI) Research Group at Universidad Carlos III de Madrid (UC3M), which is the affiliation of all authors.

Acknowledgments

This research is supported by Madrid Community Project SEGVAUTO-TRIES (S2013-MIT-2713) and by the Spanish Government CICYT Projects (TRA2015-63708-R and TRA2016-78886-C3-1-R).

Supplementary Materials

The recorded video for the real-world experiment, which was described in Section 4.3, is included. The video is played at 3x of the normal speed. (*Supplementary Materials*)

References

- [1] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii, "Vehicle Control Algorithms for Cooperative Driving with Automated Vehicles and Intervehicle Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155–160, 2002.
- [2] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.
- [3] D. L. Bock, D. Kettles, and J. Harrison, "Automated, autonomous and connected vehicle technology assessment," *Electric Vehicle Transportation Center (EVTC '16)*, 2016.
- [4] S. L. Pocster and L. M. Jankovic, "The google car: driving toward a better future," *Journal of Business Case Studies*, vol. 10, no. 1, pp. 1–7, 2014.
- [5] A. Y. S. Lam, Y.-W. Leung, and X. Chu, "Autonomous vehicle public transportation system," in *Proceedings of the 3rd International Conference on Connected Vehicles and Expo, ICCVE 2014*, pp. 571–576, Austria, November 2014.
- [6] P. Khaligh and U. Weidmann, "A conceptual framework for the interactions of autonomous public transport systems and urban planning guideline," in *Proceedings of the Swiss Transport Research Conference*, pp. 1–18, 2016.
- [7] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [8] L. E. Parker, *Multiple Mobile Robot Systems*, Springer, 2008.
- [9] M. Badreldin, A. Hussein, and A. Khamis, "A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation," *Journal of Advances in Artificial Intelligence*, vol. 2013, pp. 1–11, 2013.
- [10] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative Robots and Sensor Networks*, vol. 604, Springer International Publishing, pp. 31–51, 2015.
- [11] J. Ziegler, P. Bender, M. Schreiber et al., "Making bertha drive an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.
- [12] P. Furgale, U. Schwesinger, M. Rufli et al., "Toward automated driving in cities using close-to-market sensors: An overview of the V-Charge Project," in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, IEEE IV 2013*, pp. 809–816, Australia, June 2013.
- [13] DailyNews, "Bmw, audi push self-driving cars closer to reality," in *Daily News - Autos*, 2013.
- [14] A. Bragman, *Mercedes-benz tech brings cars closer to self driving*, USA Today, 2016.
- [15] L. Laursen, "Volvo to test self-driving cars in traffic," *IEEE Spectrum - Tech Talk*, 2013.
- [16] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [17] P. Bender, O. S. Tas, J. Ziegler, and C. Stiller, "The combinatorial aspect of motion planning: Maneuver variants in structured environments," in *Proceedings of the IEEE Intelligent Vehicles Symposium, IV 2015*, pp. 1386–1392, Republic of Korea, July 2015.
- [18] X. Qian, F. Althé, P. Bender, C. Stiller, and A. De La Fortelle, "Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective," in *Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016*, pp. 205–210, Brazil, November 2016.
- [19] C. Englund, L. Chen, J. Ploeg et al., "The Grand Cooperative Driving Challenge 2016: Boosting the introduction of cooperative automated vehicles," *IEEE Wireless Communications Magazine*, vol. 23, no. 4, pp. 146–152, 2016.
- [20] H. Rewald and O. Stursberg, "Cooperation of autonomous vehicles using a hierarchy of auction-based and model-predictive control," in *Proceedings of the 2016 IEEE Intelligent Vehicles Symposium, IV 2016*, pp. 1078–1084, Sweden, June 2016.
- [21] M. Naumann and C. Stiller, "Towards cooperative motion planning for automated vehicles in mixed traffic," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–6, 2017.
- [22] S. Manzingier, M. Leibold, and M. Althoff, "Driving strategy selection for cooperative vehicles using maneuver templates," in *Proceedings of the 28th IEEE Intelligent Vehicles Symposium, IV 2017*, pp. 647–654, USA, June 2017.
- [23] N. Atay and B. Bayazit, "Mixed-integer linear programming solution to multi-robot task allocation problem," *Tech. Rep.*, Washington University, vol. 54, St. Louis, USA, 2006.
- [24] A. R. Mosteo, *Multi-robot Task Allocation for Service Robotics: From Unlimited to Limited Communication Range*, Universidad de Zaragoza, 2010.
- [25] A. Hussein and A. Khamis, "Market-based approach to Multi-robot Task Allocation," in *Proceedings of the 2013 International Conference on Individual and Collective Behaviors in Robotics, ICBR 2013*, pp. 69–74, Tunisia, December 2013.
- [26] W. Kmiecik, M. Wojcikowski, L. Koszalka, and A. Kasprzak, "Task allocation in mesh connected processors with local search meta-heuristic algorithms," *Lecture Notes in Computer Science*, vol. 5991, Springer Berlin Heidelberg, no. 2, pp. 215–224, 2010.
- [27] W. Chen and C. Lin, "A hybrid heuristic to solve a task allocation problem," *Computers & Operations Research*, vol. 27, no. 3, pp. 287–303, 2000.
- [28] A. Maji and M. K. Jha, "Multi-objective highway alignment optimization using a genetic algorithm," *Journal of Advanced Transportation*, vol. 43, no. 4, pp. 481–504, 2009.
- [29] A. Hussein, P. Marin-Plaza, F. Garcia, and J. M. Armingol, "Optimization-based approach for cooperation and coordination of multi-autonomous vehicles," in *Proceedings of the International Conference on Computer Aided Systems Theory (Eurocast)*, pp. 286–287, 2017.

- [30] A. Hussein, P. Marin-Plaza, D. Martin, A. De La Escalera, and J. M. Armingol, "Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection," in *Proceedings of the 2016 IEEE Intelligent Vehicles Symposium, IV 2016*, pp. 104–109, Sweden, June 2016.
- [31] P. Marín-Plaza, J. Beltrán, A. Hussein et al., "Stereo vision-based local occupancy grid map for autonomous navigation in ros," in *Proceedings of the Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, vol. 3, pp. 703–708, Rome, Italy, February 2016.
- [32] S. H. Juan and F. H. Cotarelo, *Multi-master ros systems*, Institut de Robotics and Industrial Informatics [master, thesis], 2015.
- [33] A. Hussein, P. Marin-Plaza, F. Garcia, and J. M. Armingol, "Autonomous cooperative driving using v2x communications in off-road environment," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC '17)*, pp. 1–6, 2017.
- [34] A. Kokuti, A. Hussein, P. Marin-Plaza, A. de la Escalera, and F. Garcia, "V2X communications architecture for off-road autonomous vehicles," in *Proceedings of the 2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 69–74, Vienna, Austria, June 2017.
- [35] R. Necula, M. Breaban, and M. Raschip, "Tackling the Bi-criteria facet of multiple traveling salesman problem with ant colony systems," in *Proceedings of the 27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015*, pp. 873–880, Italy, November 2015.
- [36] R. Necula, M. Breaban, and M. Raschip, "Performance evaluation of ant colony systems for the single-depot multiple traveling salesman problem," in *Proceedings of the 10th International Conference on Hybrid Artificial Intelligent Systems, HAIS 2015*, pp. 257–268, Spain, June 2015.