Noname manuscript No.
(will be inserted by the editor)

Event-MILP-Based Task Allocation for Heterogeneous Robotic Sensor Network for Thermosolar Plants

J. G. Martin · R.A. García · E.F. Camacho

Received: March 30, 2021/ Accepted: Pending

Abstract In this paper, an event-Mixed Integer Linear Programming (MILP)-based algorithm is proposed to solve the task allocation problem in a Robotic Sensor Network (RSN). A fleet of two types of vehicles is considered, giving, as a result, a heterogeneous configuration of the network, since each type of vehicle has a nominal velocity and a set of allowed paths to go. The algorithm can be applied to the distributed estimation of the solar irradiance on a parabolic trough thermosolar power plant which can be used to increase the global efficiency of the plant. A simulation environment has been built to test the proposed algorithm, taking into account the behavior of the vehicles and the structure of the solar plant. The algorithm has been compared with traditional methods such as the Optimal Assignment Problem (OAP) using a set of indexes that have been defined to this purpose.

Keywords Thermosolar plant · Multi-Robot Task Allocation · Multi-Robot Systems · Robotic Sensor Network · UAV · UGV.

1 Introduction

Significant advances have been made in the field of robotics in the last few years. On the one hand, robotic arms have experienced great progress like [23], where an autonomous robotic arm for surgical operations is developed and demonstrated; and [17], where a human skin inspired automatic ultrasound system is used for three-dimensional imaging. Another field where robotics is also taking great strides is that of humanoid robots with studies such as [34], where inverse kinematics is used to balance the full lower body of a three-dimensional NAO humanoid robot, converting the trajectories already known in reference positions for its joints, or [18], where a model-free, robust, adaptative control of humanoid robot with flexible joints is proposed. Last but not least, the emergence of more accessible vehicles in terms of price and size, together with the lower cost of sensors and electronic devices have multiplied Unmanned Aerial Vehicle (UAV) and Unmanned Ground Vehicle (UGV) applications. For instance, in the field of surveillance [32], as in [15], where chaotic maps are used for motion planning of UAV with surveillance purposes. Another field in which unmanned vehicles are having a great impact is agriculture [26], with works such as [30], where a fleet of heterogeneous ground and aerial robots is developed for weed and pest control. Autonomous vehicles are also acquiring great relevance in mapping applications [33], where there are outstanding works such as [37], in which a vision-based localization and mapping algorithm is developed for an unmanned aerial vehicle (UAV) that can operate in a riverine environment. Other application fields for unmanned vehicles are inspection [2]; logistics [9] and filming [38], among others.

J. G. Martin
Department of Systems Engineering and Automation, University of Seville, Seville, Spains
E-mail: jgarmar@us.es

R.A. García
Department of Systems Engineering and Automation, University of Seville, Seville, Spains
E-mail: ramongr@us.es

E.F. Camacho
Department of Systems Engineering and Automation, University of Seville, Seville, Spains
E-mail: efcamacho@us.es

Solar power plants are usually built over wide extensions of land located in places with high solar irradiance, which is the main parameter that determines the plant energy production.

In particular, Concentrated Solar Plants with Parabolic Trough Collector (CPS PTC), which are the most common ones nowadays, use mirrors to focus the solar energy towards some piping through which a Heat Transfer Fluid (HTF) flows. This HTF is often transported to a power generation plant where electric power is produced using the Ranking cycle. The control of this type of plant is normally carried out by controlling the flow of HTF that circulates through the plant and considering the Direct Normal Irradiance (DNI) a key parameter, since, in general terms, the more DNI, the greater the flow will have to be so that the HTC temperature does not exceed a certain upper bound and, in the same way, when the DNI is low, the flow will also have to be low so that the temperature mainstains in a certain range without falling below a lower bound.

However, DNI is usually not constant in the whole plant area, as there are factors, such as clouds, which can make the irradiance vary from one point of the solar plant to another. This becomes a problem for the control of the plant, since, when a localized drop of irradiance occurs, the system counteracts it decreasing the flow and the temperature rises too much in the rest of the plant. This has the drawback of having to defocus mirrors in order to keep the temperature in those areas under the permissible limits, thus wasting energy.

Some works deal with the problem of defocusing mirrors, e.g., in [27] and in [29] control strategies such as Model Predictive Control (MPC) [8] are proposed and in [28], controlling the flows in the different sectors of the plant by using valves is proposed. For these approaches, a spatial irradiance map throughout the plant could be very useful. More about the control of thermosolar power plants can be consulted in [7].

Network sensors [1] have been widely used for distributed data acquisition. Yet, the high cost of DNI sensors, pyrheliometers, makes the idea of setting up a network of static sensors not worth its cost, so other alternatives must be explored.

An alternative proposed in this work is the use of a Robotic Sensor Network (RSN) to collect and infer solar irradiance. To do so, UAVs and UGVs are going to be considered as sensor carriers with different mobility possibilities. The proposed RSN has a Heterogeneous Multi-Robot Task Allocation (MRTA) problem associated.

According to the taxonomy proposed in [13], this MRTA problem can be classified as:

- A problem with Single Task vehicles (ST), since each vehicle can only take a measurement in just one place at a time.
- A problem with Single-vehicle tasks (SR), since only one vehicle is needed to take a measurement at one point.
- Instead of considering it an instantaneous allocation (IA) problem, we are going to consider it as a time-extended allocation (TA) problem, as, even though we do not have complete information on how tasks are going to appear, some information of tasks is available.

MRTA has been broadly studied during the last two decades as it can be seen in [20], [36], [25], [24] and [21]. It is a fundamental part of Multi-Robot Systems (MRS) and is usually subdivided into two different problems: Multi-Robot Task Planning (MRTP) and Multi-Robot Task Decomposition (MRTD). In this case, tasks are simple, so it can be fully addressed as a Task Planning problem. The most commonly used approaches for these kinds of problems are Market-based and Optimization-based approaches.

On the one hand, Market-based approaches [5] are more robust and scalable than optimization-based ones. However, they involve explicit communication between vehicles and, in some cases, the communication cost can be excessive. Moreover, since there is no agent that has all the information of the problem available, the final allocation achieved is not guaranteed to be an optimal one.

On the other hand, optimization-based approaches [16] are highly centralized, and, as they are usually NP-Hard problems, they can pose a high computational cost, especially when an exact solution is desired as in the Optimal Assignment Problem (OAP) [12]. Furthermore, these approaches have other problems like not being robust against failures in communications, not being able to adapt themselves to dynamic environments and a high vulnerability to attacks [36]. In order to avoid high computational costs, some meta-heuristic algorithms have been used to perform the optimization, as in [19] where an approach using genetic algorithm (GA) is developed, or in [10] where approaches using Distributed Constraint Optimization Problems (DCOPs) and other graphical models are explored.

Focusing on the market-based approach, the most extended ones are the denominated auction protocols. In general, any protocol where agents indicate their interest in resources by bidding is considered an auction protocol. This kind of protocol has been widely used in human history to set aside resources. Moreover, auction algorithms have already been used to solve the MRTP problem [14], since even though MRS are cooperative systems, vehicles are not programmed to be selfish or to lie for their own benefit. Also, in [22], an auction algorithm that takes into account the availability of different types of resources and the cost of communications is developed; and in [3] a distributed algorithm which makes use of auctions and consensus is developed.

Notice that the auction algorithm may have some of the problems found in centralized approaches, like sensitivity to failures in communication (there must be an auctioneer for the bidders). However, the sensitivity towards these problems is much lower than in optimization approaches. For this reason, in well-known auction-based MRTA algorithms such as MURDOCH [14], there is a great emphasis on monitoring the progress of the allocated task and in task re-allocating in case of expiring time limitations. This way, any failure derived from a failure in communication can be detected and solved.

In this paper a fleet of heterogeneous vehicles in which vehicles may have different mobility characteristics (speed, charge and discharge rates, etc) is taken into account to solve the problem of solar irradiance data acquisition in a thermosolar power plant.

The main contribution of this paper is to present an event Mixed Integer Linear Programming (MILP)-based algorithm for the Multi-Robot task allocation problem. The proposed algorithm deals with heterogeneous vehicles, taking into consideration the tasks that have already been allocated to vehicles in previous auctions and the possibility of a task not being allocated by an auctioneer if the costs of doing so are too high. To test this algorithm, a simulation environment, which takes into account the behavior of the vehicles, their simplified dynamics and the power plant layout, has been built. Eventually, the results obtained from testing the algorithm using different weights have been compared using a set of indexes that have been defined to contrast the performance of any MRTA algorithm.

The rest of the paper is organized as follows. In Section 2 the problem under consideration and the objectives pursued are presented. Besides, the power solar plant model and the allowed paths for each type of vehicle are introduced. The task management algorithm is introduced in Section 3. The simulation environment including the vehicle models and their behaviour are shown in Section 4. Results from simulations are analyzed in Section 5 and finally, some conclusions are drawn in Section 6.

2 Problem Formulation

In this section the MRTA problem considered is formulated, assuming that there is a heterogeneous fleet of vehicles and a structured known environment, in which each type of vehicle can move around through different paths.

2.1 Thermosolar plant layout

To address this problem a generic thermosolar power plant of 30 MW and 63 ha is taken into consideration and discretization in several measurement spots is made. Figure 1 depicts the discretized plant.

More in detail, the elements that appear in Figure 1 are:

- A restricted rectangular area in the middle between subsectors 1 and 2 which represents the power plant main building.
- A pair of horizontal large obstacles between subsectors 1 and 3 and between subsectors 2 and 3 which represents the central cold and hot manifolds.
- A set of vertical rectangular obstacles which represents the collectors.
- A set of four grey squares which represents charge stations for the vehicles.

Another element shown in Figure 1 is the mesh of 434 red circles which represents the different measurement spots where a vehicle can be allocated to take a measurement. Notice that irradiance can only be measured in the spots belonging to this discrete grid.

UGVs and UAVs have different sets of paths allowed since they have different mobility characteristics. In particular, the well-known structure of a parabolic trough thermosolar power plant has free space under the collectors and obstacles in the ground level due to the manifolds.

A sample of the grid has been selected as it is shown in Figure 2, where a grid of 4×5 points with the allowed paths among them is depicted; these graphs are closely related to the previous figure. Specifically, Figure 2a shows the case of UGV mobility. It is only restricted by ground-level obstacles such as manifolds. From the UGV perspective, the disposition of these obstacles leads to the division of the field into three sectors separated by two bottlenecks, as it can be seen in Figure 1. This gives the UGV great freedom to move inside the different sectors, but it allows only one way to move from one sector to another.

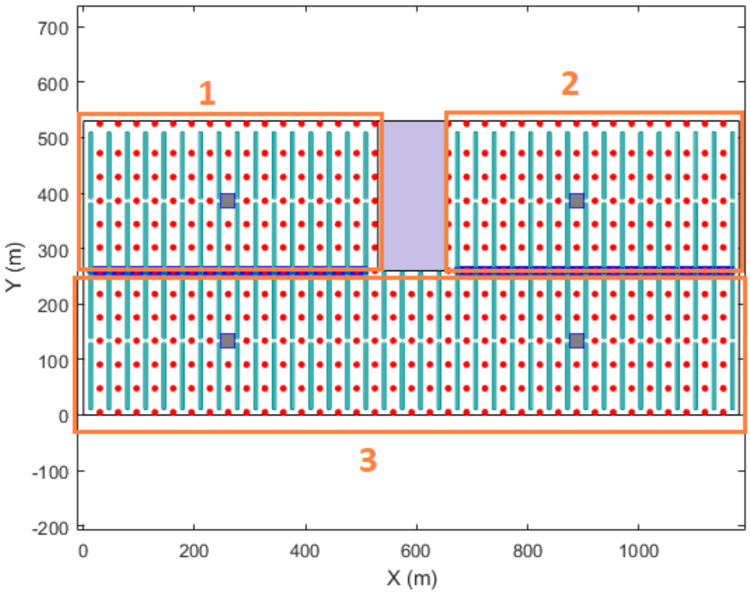
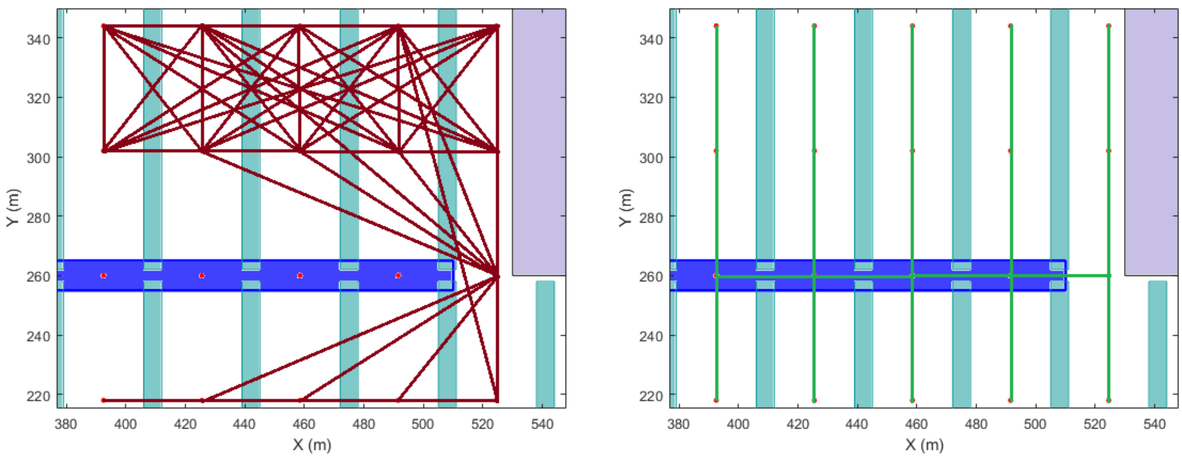


Fig. 1: Loops scheme. Distribution of the 434 nodes in the solar plant and in subsectors.



(a) Allowed paths for UGVs. Vehicles can move under the collectors, but they cannot do it under the manifolds.

(b) Allowed paths for UAVs. Vehicles cannot fly over the collectors but they can fly over the manifolds.

Fig. 2: Section of the grid of 5×4 measurement spots showing the bottleneck in the allowed path for the UGVs and the prohibition of flying over the collectors for the UAVs.

Meanwhile, Figure 2b illustrates the mobility of the UAVs, where the only restriction that is taken into account is that they are not allowed to fly above the collectors. This limitation has been introduced in order to preserve the solar plant integrity in case of failure of the UAVs. This constraint only allows the UAVs to move between corridors in certain rows of the plant.

2.2 Problem statement

Given a surface to be covered and a number of unmanned vehicles N_r, the problem is to fulfill the tasks required within the operation time of the plant. The surface can be discretized into p measurement spots, $V = \{1, 2, \ldots, p\}$ and the tasks are equivalent to sending an unmanned vehicle to a measurement spot.

The different paths that each type of vehicle can follow can be modeled by graphs. The use of graphs to model the possible paths of the vehicles has been widely used in the literature [11].

In this work two graphs, one for each kind of unmanned vehicle considered, are defined:

- $\mathcal{G}_{ugv} = \{V, E_{ugv}\}$
- $\mathcal{G}_{uav} = \{V, E_{uav}\}$

These graphs are defined by their vertexes, which are common in this case since they are the measurement spots, V; and by the edges, which are the distances of the allowed paths among those points, which are not common since UGVs and UAVs cannot move along the same allowed paths. The set of paths available for each type of vehicle can be denoted as $E_{ugv} = \{e_1^{ugv}, e_2^{ugv}, \ldots, e_{lg}^{ugv}\}$ and $E_{uav} = \{e_1^{uav}, e_2^{uav}, \ldots, e_{la}^{uav}\}$ respectively.

Notice that these graphs are important to compute the distances between the vehicles and the measurement spots using Dijkstra algorithm, which is well known in the vehicle path planning field [35].

Assumption 1 *There is an upper layer that decides in which spot or spots measures must be taken, and sends them to the MRTA layer.*

In this way, every time there are new tasks, the N_m new tasks must be allocated to the unmanned vehicles.

The unmanned vehicles are divided into two types, unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs), being $N_r = N_{ugv} + N_{uav}$.

The goal of the fleet of vehicles is to fulfill the missions or tasks in the most efficient way possible, i.e., with the least possible cost attending to certain defined metrics. These costs are usually referred to as “Utilities”, which must be scalar values so that they can be easily compared. In the present work, the utilities are chosen in order to reduce the dispersion of the vehicles around the map, since the more packed the missions that a certain vehicle performs, the less distance this vehicle travels, and therefore the less energy it uses.

3 Task management

In this section the considerations that have been taken into account are presented along with the protocol and the utilities that are used to calculate the cost of performing or not performing missions.

3.1 Proposed approach

Since the aim of this work is not to tackle the problem of losing communications, the following assumption has been taken:

Assumption 2 *All task allocations are received correctly by the bidders and all bids reach the auctioneer correctly.*

Besides, in order to implement the task planning auction algorithm, some different considerations have been taken into account:

- There is always an auctioneer, which can be any of the nodes of the system (one of the vehicles or the central system in which the MRTA layer runs)
- In an auction, there is the possibility of not allocating a task if the cost of allocating it is too high.
- New tasks can appear one by one, or in groups, so there could be a group of tasks available for the next auctions. Notice that, N_m can be lower or bigger than N_r depending on the number of new tasks and on the amount of unallocated tasks from previous batches.
- Only one mission per vehicle can be allocated in a single auction, since in any case, a vehicle is only going to be able to fulfill one mission at a time. In this way, grouping the missions into batches is favored.
- Auctions are going to be done every time that a vehicle queue of tasks empties or whenever a new mission shows up. If none of these events occur in a time limit an auction will be performed anyway. This consideration is needed since the price of not allocating missions will increase as time passes and not having it could leave missions unallocated for a long time.
- All the available tasks are offered in batches and the auctioneer chooses the allocation of the complete batch of tasks in order to minimize the global cost instead of allocating the missions one by one according to its particular cost.

– Vehicles which are currently completing a previously allocated task are also taken into account during auctions and, if chosen for a new mission, this mission is queued up in the vehicle.
– In the calculation of costs, the vehicles take into consideration not only the actual cost of the new possible missions, but also the remaining missions in their queues and the positions in where they will be when finishing them.
– The vehicles will reorder their queued up tasks in order to maximize the efficiency of performing them.

3.2 Proposed protocol

The proposed protocol is described in the following algorithm.

Algorithm 1: Proposed protocol

```
Initialization;
while simulation active do
    Simulate environment;
    if An event triggers an auction then
        All unallocated tasks and new tasks are joined into a batch of tasks;
        The auctioneer asks the vehicles their prices to perform each task;
        Each vehicle computes their cost of performing each task and send it back to the auctioneer;
        The auctioneer computes the cost of not allocating each tasks;
        After receiving all the bids the auctioneer decides the allocation using MILP;
        The auctioneer sends their newly allocated tasks to the vehicles;
        Each vehicle checks if there is a place in its queue in which its total remaining tour can be
          shortened;
        All unallocated tasks are stored for future auctions
    end
end
```

Remark 1 Events that can trigger an auction are either the appearance of a new task, the fact that a vehicle has no more missions stacked or, that the time since the last auction has exceeded a defined time limit.

Remark 2 The auctioneer has a global perspective since it takes into account the cost that each vehicle has sent for each task and the cost of not allocating missions (which depends on global variables that vehicles do not know like the time each task has been unallocated) and it calculates the allocation that minimizes the global cost using MILP.

Remark 3 Since the main aim is to group tasks in the best way and tasks appear randomly and sequentially to the eyes of the vehicles, the fact of reordering them, or placing them in the best possible place of the queue will always favor the objetive.

3.3 Cost function

The problem that the auctioneer must solve to maximize its revenue every time that an event triggers an auction is described by expressions (1a-1e).

$$\min_{V,U} J = \sum_{i=1}^{N_r}\sum_{j=1}^{N_m} \lambda_{ij}u_{ij} + \sum_{j=1}^{N_m} c_j v_j \tag{1a}$$

$$s.t \sum_{j=1}^{N_m} u_{ij} \leq 1 \; \forall i \tag{1b}$$

$$\sum_{i=1}^{N_r} u_{ij} + v_j = 1 \; \forall j \tag{1c}$$

$$u_{ij} = 0, 1 \; \forall i, \; \forall j \tag{1d}$$

$$v_j = 0, 1 \; \forall j \tag{1e}$$

where $U = [u_{11}, u_{12}, ..., u_{1N_m}, ..., u_{N_r1}, u_{N_rN_m}]$ with variables u_{ij} representing if mission j is allocated to vehicle i, $V = [v_1, v_2, ..., v_{N_m}]$ with variables v_j representing if mission j remains non-allocated, λ_{ij} and c_j are the cost of allocating a mission j to a vehicle i and of leaving mission j non-allocated respectively. Furthermore, constraint (1b) indicates that a vehicle i can choose only one mission in one step and constraint (1c) indicates that a mission is allocated to a vehicle or it is free.

Remark 4 Notice that if the possibility of not allocating a task is not contemplated, i.e. $c_j \to \infty$, the formulation on (1a-1e) gets similar to solving an OAP problem each time there is an auction, since v_j are slack variables for (1c).

3.4 Metrics

As this is a heterogeneous problem, the weight parameters of the cost function (1a) may not be the same for all the vehicles, or even the same parameters may have different weights in the cost estimation of an UGV and the cost estimation of an UAV. This difference is justified because an UAV can cover a longer distance than an UGV in the same time. However, it is more critical to have a low battery level in an aerial vehicle than in a ground vehicle. As it has been mentioned before, the possibility of not fulfilling a mission is considered with an associated cost. The main metric to calculate this cost will be the time that the mission has been unallocated.

The cost associated with allocating mission j to a vehicle i, λ_{ij}, in (1a), is computed as follows:

$$\lambda_{ij} = Q_1 \cdot L_{ij} + Q_2 \cdot L_{jch} \tag{2}$$

where the first term depends on the length of the path between the vehicle and the objective point of the mission, i.e., the distance that the vehicle must travel to finalize its queued missions plus the distance from the final point of its queue to the objective point, and the second term depends on the distance from the objective point to the nearest charge station. In this way, getting farther from a charge station is also penalized. Notice that both L_{ij} and L_{jch} depend on the type of vehicle, since it depends on the graphs presented in Section 2 (the distance between two nodes is not the same for the different vehicles since different vehicles *see* different obstacles).

The cost associated to not allocating mission j, c_j in (1a), is computed as follows:

$$c_j = Q_3 + Q_4 \cdot t_{na} \tag{3}$$

where the first term is the minimum cost of not allocating mission j and the second term penalizes the time that the mission has been unallocated. In this way, not allocating a mission which is too far will become less attractive as time passes and these far tasks cannot remain unallocated eternally.

These weights are not necessarily constant. They can be functions of different parameters such as, for example, weight $Q_2 = Q_2(SOC)$, since a vehicle can penalize more the fact of getting farther from a charge station when it has a low battery level.

4 Simulation environment

To check the performance of the proposed task-allocation algorithm, a simulation environment for the vehicles behavior and battery levels will be implemented.

The behavior of vehicles is detailed below:

1. If stopped it will check if there are any missions in its queue.
2. If there are no missions it will remain stopped but if there are any it will load the next mission as “current mission”. Once a mission has been loaded the vehicle will turn it into a set of reference points.
3. It will change its state to “march state”, so it will start moving.
4. It will continue in “march state” until it reaches the objective, i.e., the distance to the objective is less than a certain tolerance level.
5. Once it reaches the objective it will check if the current mission is a “charge mission” or a “measurement mission”.
6. If it is a charge mission, it will change its state to “charging state” and if it is a measurement mission it will change its state to “measuring state”.
7. If it is in “charging state”, it will continue in that state until it is fully charged and once it is fully charged it will change its state to “stopped”.
8. If it is in “measuring state”, it will continue in that state until it has taken the measurement and it will change its state to “stopped”.
9. If the vehicle completes a mission, it will check its battery level, and if it is lower than 40% in the case of an UAV, or 20% in the case of an UGV, it will insert a charge mission to the nearest free charge station as the next mission. Notice that even though this security battery level can seem high, the vehicles may have yet a mission to finish before going to a charge station. When a vehicle takes on a charge mission to a station as “next mission”, this charge station is not considered free anymore until this vehicle finishes charging. In case all charge stations are being used, this vehicle will remain “Stopped” until a charge station is free again.
10. In the case there are no more missions for UAVs they land and in the case they are currently on land when a mission is loaded, the first thing that they must do is to reach their reference altitude.

The simulation environment can be observed in a more schematized way in the flowchart in Figure 3

The models of the vehicles that are being used are simplified models in order to obtain a lighter and faster simulation. It is supposed that each vehicle has a perfect low level controller so that they can be considered as points that move on the space. A mean velocity on plane XY and for the height, Z, is assumed. A simplified model is also taken into account for the batteries. It is considered that vehicles lose battery and recharge it in a lineal way with a charge and discharge rate respectively. The concrete value of these parameters depend on the type of vehicle, UGV or UAV.

The parameters considered for UGVs and UAVs in the simulations are depicted in Table 1 and Table 2 respectively. These parameters are based on the characteristic of commercial products, like DJI Matrice 200 drones [6] and Summit-XL ground vehicle [31]. Notice that the speed selected for the UAVs is high since they are flying far over the plant and taking into account the considerations presented in Section 2.

Table 1: Value of the UGV parameters based on commercial vehicle for outdoors purpose

Parameter	Value	Parameter	Value
V_{mean}	1.5 m/s	Discharge rate	0.005
V_{Zmean}	0 m/s	Charge rate	0.0025

Table 2: Value of the UAV parameters based on commercial vehicles.

Parameter	Value	Parameter	Value
V_{mean}	10 m/s	Discharge rate	0.1
V_{Zmean}	3 m/s	Charge rate	0.03

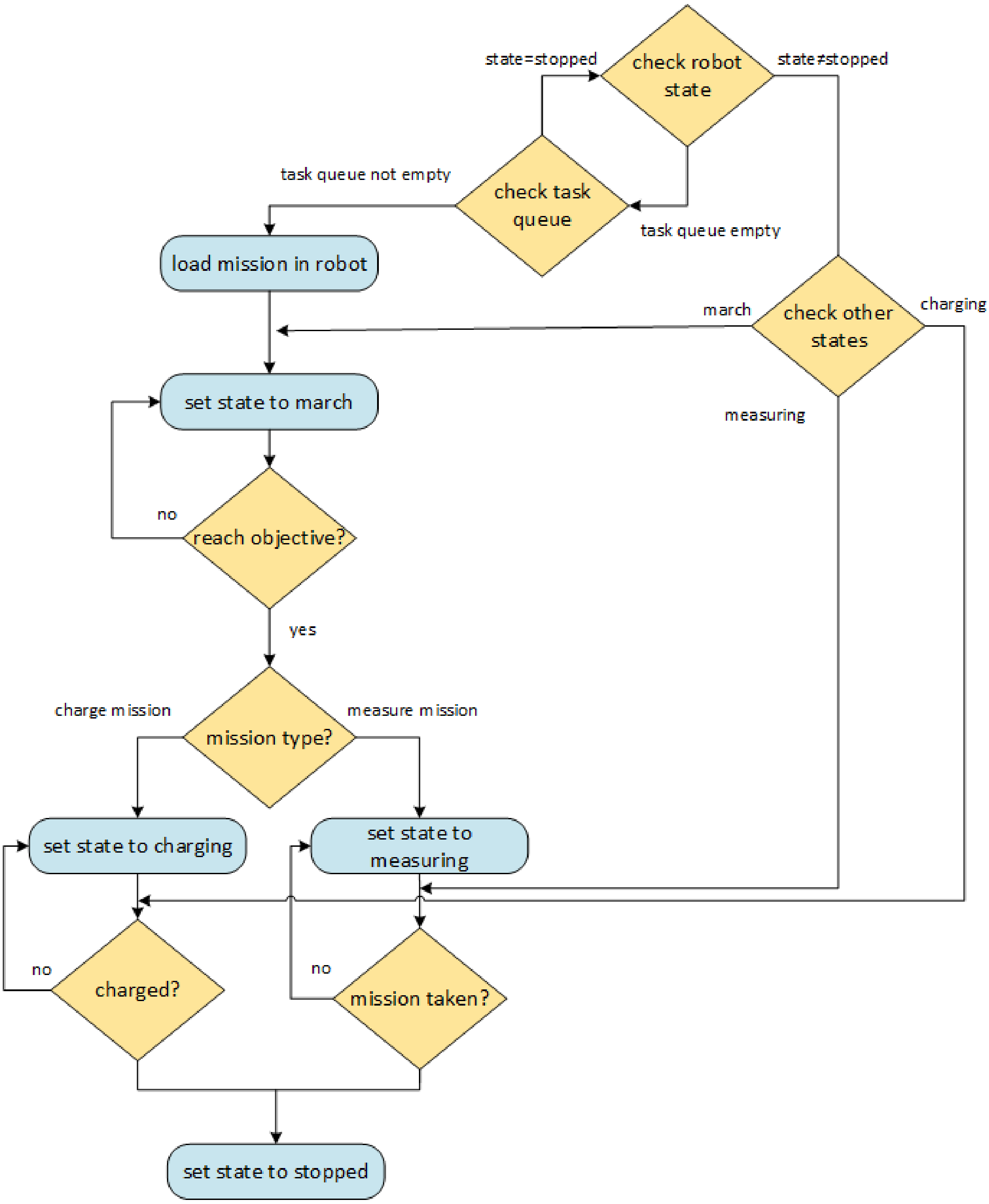


Fig. 3: Simulation environment scheme

where V_{mean} is the speed of the vehicles on a plane XY, V_{Zmean} is the speed of vertical movements, charge and discharge rates are the state of charge (SOC) charged or discharged per second.

5 Results

The simulation environment depicted in Section 4 has been programmed in Matlab® in a 3.2 GHz Intel Core® $i7$, 16 GB RAM PC. In order to solve the cost function (1) each time an allocation is needed, *intlinprog* function from the Optimization Toolbox has been used.

The proposed algorithm has been tested considering 2 UGVs and 3 UAVs and a duration of 12 hours that corresponds to an average working day on a thermosolar plant. During these 12 hours a total of 420 measurement missions appear. These measurement missions have been generated randomly with the same seed so that they are the same for the compared algorithms. The time required to perform these simulations has been 4 hours and 5 minutes for the OAP based approach and 4 hours and 15 minutes for the proposed approach.

The weights for the proposed algorithm have been tuned fixing the values for the weights related to the cost of allocating a vehicle i to a mission j, Q_1 and Q_2, and varying the weights related to not allocating mission j, i.e., Q_3 and Q_4, equations (2) and (3) respectively. The proposed algorithm has been compared with the case in which not allocating a mission is not allowed unless there are not enough vehicles to allocate all the missions, i.e., $Q_3 = \infty$ as stated in Section 3. All the values for the weights are in Table 3.

Table 3: Weights. Values could differ for UGV and UAV.

	Proposed algorithm	OAP algorithm
Q_1	UGV: 1 / UAV: 0.05	
Q_2	UGV: $1.1 - 0.1 \cdot SOC_j$ / UAV: $1.1 - 0.1 \cdot SOC_j$	
Q_3	0.05	∞
Q_4	0.15	-

To analyze the performance, several indexes usable for any MRTA algorithm have been defined. They consider different points of view such as distance, energy, dispersion, etc, and their values are represented in Tables 4 to 11

- The distance traveled by the vehicles, since distance is one of the parameters used to perform the allocations.
  1. I_1 is the total distance traveled by all the vehicles.
  2. I_2 is the sum of the remaining distances that each vehicle still needs to travel to complete all the missions stacked in its queue.
  3. I_3 is related to the total distance that the vehicles travel, i.e., the sum of the remaining distances that each vehicle still needs to travel to complete all the missions stacked in its queue and the distance that they have already traveled.
  4. I_4 is the percentage of distance traveled by UGVs.
  5. I_5 is the percentage of distance traveled by UAVs.

Table 4: Indexes relate to distance. The best results of each index is highlighted in bold. In case neither of them is highlighted in bold it means that any of them can be considered better than the other one.

	Proposed algorithm	OAP algorithm
I_1	250204.02	**245949.86**
I_2	**1305**	5109
I_3	251509.02	**251058.86**
I_4	17.45	18.20
I_5	82.55	81.80

- The dispersion of the vehicles around the map which can be calculated by:
  $disp = \frac{1}{N_r \cdot N_k} \sum_{i=1}^{N_r} \sum_{k=1}^{N_k} \|(\mu_x^i, \mu_y^i) - (x_k^i, y_k^i)\|$,
  where N_k is the number of iterations of the simulation, μ_x^i and μ_y^i are the mean values in x and in y among the complete simulation. These are the most interesting indexes since the main aim of leaving missions unallocated is having the possibility of allocating them in later batches, performing better allocations by doing this.
  1. I_6 is related to the dispersion of all the vehicles.
  2. I_7 is related to the dispersion of the UGVs.
  3. I_8 is related to the dispersion of the UAVs.

- The energy, which is related to the traveled distance.
  1. I_9 is the mean of the energy of the vehicles. Note that the less this index value is, the more energy the vehicles will have used.

Table 5: Indexes relate to dispersion. The best results of each index is highlighted in bold. In case neither of them is highlighted in bold it means that any of them can be considered better than the other one.

	Proposed algorithm	OAP algorithm
I_6	**406.44**	442.05
I_7	311.58	**296.95**
I_8	**469.68**	538.78

Table 6: Indexes relate to energy. The best results of each index is highlighted in bold. In case neither of them is highlighted in bold it means that any of them can be considered better than the other one.

	Proposed algorithm	OAP algorithm
I_9	**59.13**	58.25
I_{10}	33.50	**49.58**

2. I_{10} is related to the total energy that vehicles have at the end of the simulation.

– The stacked missions during the simulation.
  1. I_{11} is the maximum number of stacked missions during the simulation.
  2. I_{12} is the mean number of stacked missions during the simulation.

Table 7: Indexes relate to stacked missions. The best results of each index is highlighted in bold. In case neither of them is highlighted in bold it means that any of them can be considered better than the other one.

	Proposed algorithm	OAP algorithm
I_{11}	25	25
I_{12}	**3.83**	4.21

– The number of missions that have been performed during the simulation.
  1. I_{13} is the number of missions of any type completed by all the vehicles.
  2. I_{14} is the number of charge missions completed by all the vehicles.
  3. I_{15} is the number of charge missions completed by UGVs.
  4. I_{16} is the number of charge missions completed by UAVs.
  5. I_{17} is the number of measurement missions completed by all the vehicles.
  6. I_{18} is the number of measurement missions completed by UGVs.
  7. I_{19} is the number of measurement missions completed by UAVs.

Table 8: Indexes relate to the number of missions performed. The best results of each index is highlighted in bold. In case neither of them is highlighted in bold it means that any of them can be considered better than the other one.

	Proposed algorithm	OAP algorithm
I_{13}	**441**	432
I_{14}	25	**24**
I_{15}	0	0
I_{16}	25	**24**
I_{17}	**416**	408
I_{18}	78	**84**
I_{19}	**338**	324

– The number of non allocated missions.
  1. I_{20} is the mean number of unallocated missions during the simulation.

Table 9: Indexes relate to the non allocated missions. The best results of each index is highlighted in bold. In case neither of them is highlighted in bold it means that any of them can be considered better than the other one.

	Proposed algorithm	OAP algorithm
I_{20}	1.1179583	0.0009726
I_{21}	2	0

2. I_{21} is the number of unallocated missions at the end of the simulation.

- The time that was employed to serve the missions.
  1. I_{22} is the mean time employed to serve the missions, taking into account both types of missions.
  2. I_{23} is the mean time employed to serve the charge missions.
  3. I_{24} is the mean time employed to serve the measurement missions.

Table 10: Indexes relate to time to serve missions. The best results of each index is highlighted in bold. In case neither of them is highlighted in bold it means that any of them can be considered better than the other one.

	Proposed algorithm	OAP algorithm
I_{22}	**250.51**	252.92
I_{23}	**2396.59**	2419.56
I_{24}	**121.54**	125.47

- The time employed to solve the allocation problem, which is one of the main parameters while evaluating the general performance of the algorithm.
  1. I_{25} is the mean time employed to solve the allocation algorithm.
  2. I_{26} is the maximum time employed to solve the allocation algorithm.
  3. I_{27} is the total time employed to solve the allocation algorithm during the simulation.

Table 11: Indexes relate to the solving allocation times. The best results of each index is highlighted in bold. In case neither of them is highlighted in bold it means that any of them can be considered better than the other one.

	Proposed algorithm	OAP algorithm
I_{25}	0.01	0.01
I_{26}	0.22	**0.21**
I_{27}	6.29	**3.92**

Remark 5 It is important to emphasize that although the missions are the same for all cases, the allocation may affect the order in which they are carried out.

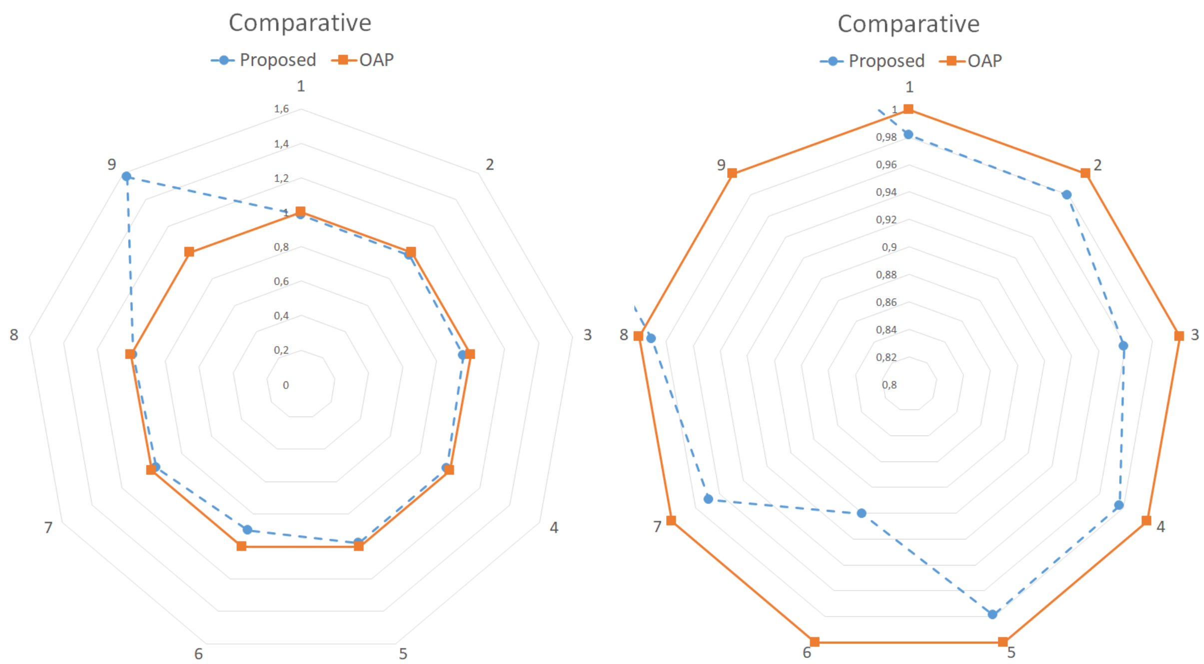
Remark 6 Even though the best values for each considered index have been marked in bold in Tables 4 to 11, since the indexes are coupled, it is important to analyze the values of the indexes using a global perspective. For example, it could be misleading to compare the indexes related to the traveled distance without taking into account the number of missions performed and other variables.

From the results for the different indexes, and taking as a reference the OAP algorithm, the following conclusions can be obtained:

- Generally, using the proposed approach made the dispersion of the vehicles decrease. This was in part as a result of a better grouping of the missions, which was the main aim of the proposed algorithm. However, in some cases, due to a bad tuning of the weights that affect c_j, the time that the algorithm let some missions unallocated can have affected badly the mean time in which measurement missions were carried out, or even the number of missions performed.

Table 12: Comparison of the results.

	Proposed algorithm	OAP algorithm	Proposed / OAP
1: Total distance / total missions (I_3/I_{13})	570.32	581.15	0.9813
2: Stacked missions / total missions (I_{11}/I_{13})	0.05669	0.05787	0.9796
3: Mean energy consumed / total missions $((100 - I_9)/I_{13})$	0.09268	0.09664	0.9944
4: Mean energy consumed / total distance $(((100 - I_9)/I_3))$	0.0001625	0.0001663	0.9771
5: Measurement missions / charge missions (I_{17}/I_{14})	16.64	17	0.9788
6: Total dispersion / total missions (I_6/I_{13})	0.9216	1.023	0.9007
7: Mean time to serve measurement missions (I_{24})	121.54	125.47	0.9687
8: Mean time to serve charge missions (I_{23})	2396.59	2419.56	0.9905
9: Total time allocating missions / total missions (I_{27}/I_{13})	$0,0142$	$0,0091$	$1,5718$



(a) General view

(b) In detail

Fig. 4: Analysis of the results of Table 12. The represented indexes are the relativized values with respect to the OAP.

- In general, the lower the c_j are, the more the indexes related to time employed to solve the allocation increase. This is logical since these times depend mostly on the number of variables, their associated costs and the number of auctions carried out.
- It has to be taken into account that the fact that missions are reordered in the queue of each vehicle every time a new mission is allocated to them improves the performance of those cases in which the algorithm favors allocating missions (c_j are high from the beginning and increase fast) including the OAP.
- When the costs of not allocating missions are lower, the mean number of unallocated missions, I_{20}, increases. Besides, the maximum number of stacked missions, I_{11}, is lower as well.
- By tuning the weights Q_3 and Q_4 properly, not only has the dispersion of the vehicles been minimized but the mean time on serving charge and measurement missions has also been reduced 0.95% and 3.13% respectively. It can also be seen that with this tuning, more measurement missions than in the OAP have been fulfilled and that the ratio of distance traveled per measurement mission is bigger as well.
- Every time that the pile of stacked missions increases and that UGVs fulfill missions, it is as a consequence of having the UAVs in "charge mode". This can be seen in Figure 5 and in Figure 6, since UGVs are significantly slower than UAVs and since the algorithm favors allocating missions to UAVs.
- Depending on what goals are being pursued, the weights can be tuned to prioritize different objectives such as using less energy, using a certain type of vehicle better or fulfilling the missions faster.

In summary, although the algorithm proposed requires more computation time to carry out an allocation since it deals with a larger amount of alternatives (tasks can remain unallocated and unallocated tasks add up to general tasks for future allocations), it can be concluded that it is achieving its goal of grouping the tasks by zones, thus achieving a better performance in the course of the full simulation.

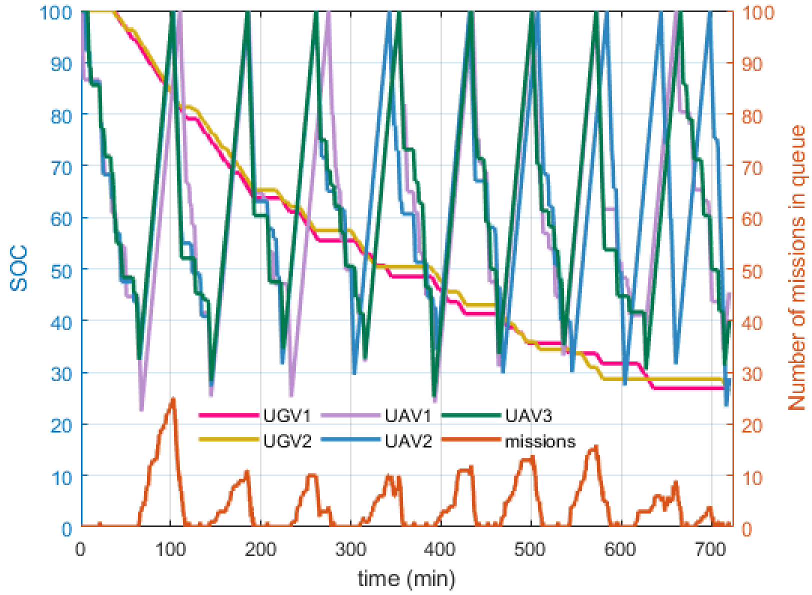


Fig. 5: SOC and stacked missions of the vehicles in the proposed algorithm

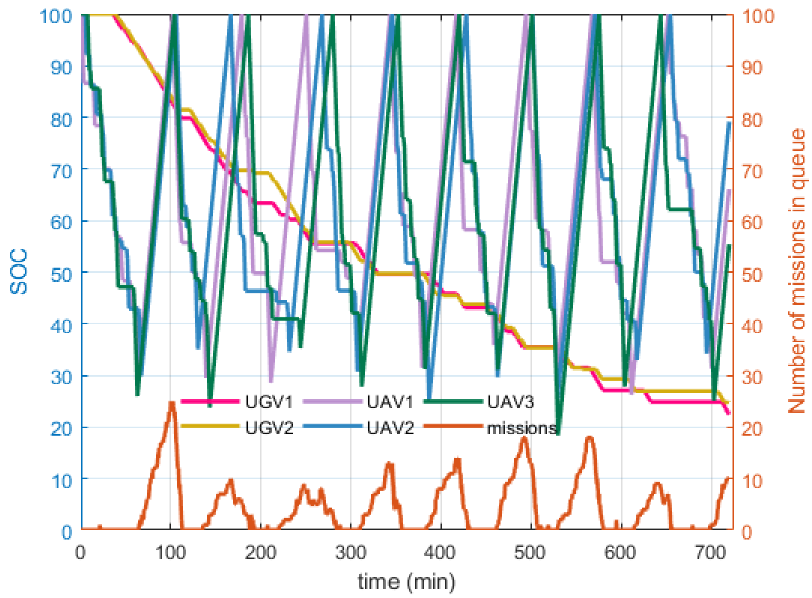


Fig. 6: SOC and stacked missions of the vehicles in the OAP algorithm

6 Conclusions

In this paper, an algorithm based on MILP has been proposed for the heterogeneous vehicle task allocation problem in the known structured environment of a thermosolar plant.

The proposed algorithm allocates missions in an asynchronous way, considering the option of not allocating missions with an associated cost. The cost of not allocating a mission varies with the time that those specific missions have been unallocated. The utilities that have been used by each vehicle to measure its cost to perform a certain mission depend on the distance that it has to travel, the distance that it will be from a charge station once the mission is completed and the energy level that it has at the moment of receiving the allocation.

This algorithm has been tested in a simulator, considering a simulation time of twelve hours and using several indexes that have been proposed to compare any MRTA allocation problem. It has been found out that, as long as the weights are well-tuned, having the possibility of not allocating missions can involve benefits in the grouping of missions so that they are performed in a more efficient way. However, a badly chosen cost on not allocating missions can be detrimental as missions may not be allocated even in a case where performing them could be better than not doing so.

Although the proposed algorithm means an increase in the allocation solving time with respect to the OAP, it may improve the efficiency in terms of energy consumption and distance traveled and the mean time for serving the missions.

Future improvements for the proposed algorithm could include: implementing CBAA algorithm proposed in [3], instead of MILP to perform the allocation; contemplating the possibility of dropping missions when new tasks are introduced in the system, or when vehicles need to charge their batteries, as it is proposed in [4]; treating charge missions as part of the batch in the auctions, with their associated cost depending on the SOC level of the vehicles; taking into account the possibility of reallocating tasks which have been already stacked into a vehicle queue, with an associated cost of unstacking the mission; considering different priorities for missions so that some of them cannot be put off; and testing the proposed algorithm in the obtaining of a spatially distributed DNI estimation.

Acknowledgements This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 789051)".

DECLARATIONS

Ethical Approval

Not applicable

Conflict of interest/Competing interest

The authors declare that they have no conflict of interest.

Consent to Participate

J.G. Martin, R.A. Garcia and E.F. Camacho consent to be part of the developed work in this paper.

Consent to Publish

J.G. Martin, R.A. Garcia and E.F. Camacho consent the journal to publish this work

Author Contributions

- **J.G. Martin**: Idea, code, writing of the paper, review and submission.
- **R.A. Garcia**: Idea, code, writing of the paper, review.
- **E.F. Camacho**: Idea, writing of the paper, review.

Funding

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 789051)".

Availability of data and material

Data and material has been described in this paper so that results can be reproduced.

Code availability

Code is available by request to the first author.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer networks **38**(4), 393–422 (2002)
2. Brusell, A., Andrikopoulos, G., Nikolakopoulos, G.: A survey on pneumatic wall-climbing robots for inspection. In: 2016 24th Mediterranean Conference on Control and Automation (MED), pp. 220–225 (2016)
3. Choi, H., Brunet, L., How, J.P.: Consensus-based decentralized auctions for robust task allocation. Transactions on Robotics **25**(4), 912–926 (2009)
4. Das, G.P., McGinnity, T.M., Coleman, S.A., Behera, L.: A distributed task allocation algorithm for a multi-robot system in healthcare facilities. Journal of Intelligent & Robotic Systems **80**(1), 33–58 (2015)
5. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multi-robot coordination: A survey and analysis. Proceedings of the IEEE **94**(7), 1257–1270 (2006)
6. DJI: Dji matrice v2. URL `https://www.dji.com/es/matrice-200-series-v2`
7. F. Camacho, E., Berenguel, M., R. Rubio, F.: Advanced control of solar plants. Springer (1997)
8. F. Camacho, E., Bordons-Alba, C.: Model Predictive Control in the Process Industry. Springer (1995)
9. Farinelli, A., Zanotto, E., Pagello, E., et al.: Advanced approaches for multi-robot coordination in logistic scenarios. Robotics and Autonomous Systems **90**, 34–44 (2017)
10. Farinelli, A., Zanotto, E., Pagello, E., et al.: Advanced approaches for multi-robot coordination in logistic scenarios. Robotics and Autonomous Systems **90**, 34–44 (2017)
11. Fentanes, J.P., Lacerda, B., Krajník, T., Hawes, N., Hanheide, M.: Now or later? predicting and maximising success of navigation actions from long-term experience. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 1112–1117 (2015)
12. Gale, D.: The Theory of Linear Economic Models. University of Chicago press (1989)
13. Gerkey, B., Mataric, M.J.: Are (explicit) multi-robot coordination and multi-agent coordination really so different. In: Proceedings of the AAAI spring symposium on bridging the multi-agent and multi-robotic research gap, pp. 1–3 (2004)
14. Gerkey, B.P., Mataric, M.J.: Sold!: Auction methods for multirobot coordination. IEEE transactions on robotics and automation **18**(5), 758–768 (2002)
15. Gohari, P.S., Mohammadi, H., Taghvaei, S.: Using chaotic maps for 3d boundary surveillance by quadrotor robot. Applied Soft Computing **76**, 68 – 77 (2019)
16. Horst, R., Pardalos, P.M., Van Thoai, N.: Introduction to global optimization. Springer Science & Business Media (2000)
17. Huang, Q., Lan, J., Li, X.: Robotic arm based automatic ultrasound scanning for three-dimensional imaging. IEEE Transactions on Industrial Informatics **15**(2), 1173–1182 (2018)
18. Jin, M., Lee, J., Tsagarakis, N.G.: Model-free robust adaptive control of humanoid robots with flexible joints. IEEE Transactions on Industrial Electronics **64**(2), 1706–1715 (2016)
19. Jose, K., Pratihar, D.K.: Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. Robotics and Autonomous Systems **80**, 34–42 (2016)
20. Khamis, A., Hussein, A., Elmogy, A.: Multi-robot task allocation: A review of the state-of-the-art. In: Cooperative Robots and Sensor Networks 2015, pp. 31–51. Springer (2015)
21. Koubâa, A., Khelil, A.: Cooperative Robots and Sensor Networks 2014. Springer (2014)
22. Lee, D.H.: Resource-based task allocation for multi-robot systems. Robotics and Autonomous Systems **103**, 151–161 (2018)
23. Muradore, R., Fiorini, P., Akgun, G., Barkana, D.E., Bonfe, M., Boriero, F., Caprara, A., De Rossi, G., Dodi, R., Elle, O.J., et al.: Development of a cognitive robotic system for simple surgical tasks. International Journal of Advanced Robotic Systems **12**(4), 37 (2015)
24. P. Gerkey, B., Mataric, M.: A formal framework for the study of task allocation in multi-robot systems. International Journal of Robotic Research - IJRR (2003)
25. P. Gerkey, B., Mataric, M.J.: Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. In: ICRA, vol. 3, pp. 3862–3868 (2003)
26. Roldán, J.J., del Cerro, J., Garzón-Ramos, D., Garcia-Aunon, P., Garzón, M., de León, J., Barrientos, A.: Robots in agriculture: State of art and practical experiences. Service Robots (2018)
27. Sánchez, A., Gallego, A., Escaño, J., F. Camacho, E.: Event-based mpc for defocusing and power production of a parabolic trough plant under power limitation. Solar Energy **174**, 570–581 (2018)
28. Sánchez, A., Gallego, A., Escano, J., F. Camacho, E.: Temperature homogenization of a solar trough field for performance improvement. Solar Energy **165**, 1–9 (2018)

29. Sánchez, A., Gallego, A., Escaño, J., F. Camacho, E.: Adaptive incremental state space mpc for collector defocusing of a parabolic trough plant. Solar Energy **184**, 105–114 (2019)
30. Gonzalez-de Santos, P., Ribeiro, A., Fernandez-Quintanilla, C., Lopez-Granados, F., Brandstoetter, M., Tomic, S., Pedrazzi, S., Peruzzi, A., Pajares, G., Kaplanis, G., et al.: Fleets of robots for environmentally-safe pest control in agriculture. Precision Agriculture **18**(4), 574–614 (2017)
31. S.L.L, R.A.: Summit-xl. URL `https://www.robotnik.es/robots-moviles/summit-xl/`
32. Theodoridis, T., Hu, H.: Toward intelligent security robots: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **42**(6), 1219–1230 (2012)
33. Thrun, S., et al.: Robotic mapping: A survey. Exploring artificial intelligence in the new millennium **1**(1-35), 1 (2002)
34. Tutsoy, O., Erol Barkana, D., Colak, S.: Learning to balance an nao robot using reinforcement learning with symbolic inverse kinematic. Transactions of the Institute of Measurement and Control **39**(11), 1735–1748 (2017)
35. Wang, H., Yu, Y., Yuan, Q.: Application of dijkstra algorithm in robot path-planning. In: Second international conference on mechanic automation and control engineering, pp. 1067–1069 (2011)
36. Yan, Z., Jouandeau, N., Cherif, A.A.: A survey and analysis of multi-robot coordination. International Journal of Advanced Robotic Systems **10**(12), 399 (2013)
37. Yang, J., Dani, A., Chung, S.J., Hutchinson, S.: Vision-based localization and robot-centric mapping in riverine environments. Journal of Field Robotics **34**(3), 429–450 (2017)
38. Zema, N.R., Natalizio, E., Yanmaz, E.: An unmanned aerial vehicle network for sport event filming with communication constraints (2017)