Gossip Algorithms for Heterogeneous Multi-Vehicle Routing Problems

Mauro Franceschelli[a], Daniele Rosa[a], Carla Seatzu[a], Francesco Bullo[b]

[a]*Dep. of Electrical and Electronic Engineering, Univ. of Cagliari, Italy (e-mail: {mauro.franceschelli, daniele.rosa,seatzu@diee.unica.it})*
[b]*Dep. of Mechanical Engineering, Univ. of California at Santa Barbara, California, USA (e-mail: bullo@engineering.ucsb.edu)*

Abstract

In this paper we address a class of heterogeneous multi-vehicle task assignment and routing problem. We propose two distributed algorithms based on gossip communication: the first algorithm is based on a local exact optimization and the second is based on a local approximate greedy heuristic. We consider the case where a set of heterogeneous tasks arbitrarily distributed in a plane has to be serviced by a set of mobile robots, each with a given movement speed and task execution speed. Our goal is to minimize the maximum execution time of robots.

1. Introduction

The traveling salesman problem (TSP) is a well known topic of research and can be stated as follows: find the Hamiltonian cycle of minimum weight to visit all the nodes in a given graph. Instructive surveys can be found in [15, 17, 21]. This problem has received great attention for both its theoretical implications and its several practical applications. The Vehicle Routing Problem (VRP) is a generalization of the TSP and was firstly introduced in [10]: given a fleet of n vehicles and a set of locations to be visited, the vehicle routing problem consists of finding n tours to visit all locations in minimum time.

Several extensions of the TSP and the VRP have been proposed to better suit practical applications by introducing several additional constraints and objectives such as a variable number of vehicles, a finite load capacity, a cost associated to each node which represents the demand of the costumer, service time windows and several more. Numerous extensions are well summarized

[☆]This work has been partially supported by the European Community's Seventh Framework Programme under project HYCON2 (Grant Agreement n. FP7-ICT-2009-5/N.257462.) and in part by ARO grant W911NF-11-1-0092. Daniele Rosa gratefully acknowledges Sardinia Regional Government for the financial support of his PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2007-2013 - Axis IV Human Resources, Objective l.3, Line of Activity l.3.1.)

in [3, 24, 26]. Finally, several extensions explore a dynamic setting in which multiple vehicles serve a dynamic number of tasks as discussed in [5].

Multi-vehicle routing problems have a combinatorial nature, as all the possible tours must be explored to find the optimal configuration. Exact algorithmic formulations are based, for example, on Integer Linear Programming (ILP) as described in [18, 26]. General ILP solvers are characterized by an exponential computational complexity, thus in the last decades many approximate algorithms have been proposed which are characterized by a lower computational complexity. Examples of heuristics and approximate algorithms are presented in [9, 12, 14, 20, 22, 24].

In this paper we are interested in an instance of the VRP, called the Heterogeneous Multi Vehicle Routing Problem (HMVRP), with the following properties: the number n of vehicles is given a priori, a set $\mathcal{K}$ is given containing k tasks arbitrarily distributed in a plane, to each task is assigned a servicing cost, each vehicle is characterized by a movement speed and a task execution speed.

It has been shown in [7] that when comparing the length of the optimal tour of one vehicle that visits all tasks locations with the multiple vehicle case, the maximum length of the tours for the multiple vehicle case is proportional to the tour length of the single vehicle case and proportionally inverse to the number of vehicles. Both upper and lower bounds with such scaling were given.

In this paper we extend the result in [7] by considering execution times instead of tour lengths to account for vehicles of different speeds, tasks with arbitrary execution costs and vehicles with different task execution speeds. We provide upper and lower bounds to the optimal solution as function of the single vehicle optimal tour length to put in evidence how the performance is affected by the number of vehicles.

We propose two distributed and asynchronous algorithms for the HMVRP: the first one is based on the iterative optimization of the local task assignment between pairs of vehicles [13], the second one is based on local task exchange of assigned tasks, one by one, between couples of vehicles [25]. For both algorithms we provide deterministic bounds to their performance. The proposed approaches to the HMVRP are distributed algorithms easy to implement in a networked system and have favorable computational complexity with respect to the ratio k/n between the number of tasks and vehicles instead of k as in the centralized approach.

Note that the considered problem can also be seen as a particular instance of a min/max VRP problem whose main feature is the heterogeneity of the speed and the tasks execution speed of the vehicles. Related works on the min/max VRP problem include [1, 19, 2].

Summarizing, the following are the main contributions of this paper.

- We formalize the centralized problem in terms of a mixed integer linear programming (MILP) problem and extend the bounds in [7] for the multi TSP to the HMVRP.

- We propose a first distributed algorithm, based on gossip communication

and on the solution of local MILP, to solve the HMVRP and characterize some of its properties.

- We propose a second distributed algorithm to solve HMVRP, based on gossip communication and on local task exchanges, characterized by a low computational complexity.
- We provide simulations that show that the proposed algorithms attain a constant factor approximation of the optimal solution with respect to the number of vehicles. A detailed comparison among the performances of the two proposed decentralized solutions is also presented.

2. Problem statement

Consider a set $\mathcal{N}$ of n mobile robots scattered in a connected region $\mathcal{R}$ in a plane. Let $\mathcal{K}$ be a set of k tasks scattered in region $\mathcal{R}$, that should be assigned to robots to be executed.

Robots move at different speeds and have different execution speeds of tasks. Tasks have different costs. In particular, the following notation is used:

- v_r is the speed of robot R_r,
- w_r is the task execution speed of robot R_r,
- v_{min} (v_{max}) is the minimum (maximum) speed of robots,
- w_{min} (w_{max}) is the minimum (maximum) task execution speed of robots,
- c_i is the cost of the i-th task,
- c_{min} (c_{max}) is the minimum (maximum) cost of tasks.

Moreover, d_{max} is the maximum length of the shortest path between any two points in the region $\mathcal{R}$.

Robots are supposed to first coordinate themselves to decide upon their task assignment and then start to serve the tasks autonomously.

To use a notation that is standard in the literature, we assume that robots are initially positioned in depots and should go back to them after the execution of tasks. The set of depots is called $\mathcal{D}$ and the generic r-th depot is D_r.

Now, if $\mathcal{K}_r$ denotes the set of tasks assigned to robot R_r, our goal is to minimize the objective function:

$$J = \max_{r \in \mathcal{N}} J_r = \left(\frac{TSP(\mathcal{K}_r \cup \{D_r\})}{v_r} + \frac{\sum_{i \in \mathcal{K}_r} c_i}{w_r} \right) \tag{1}$$

where TSP ($\mathcal{K}_r \cup \{D_r\}$) is the minimum TSP tour length of robot R_r that, initially positioned in D_r, visits all tasks in $\mathcal{K}_r$ and go back to D_r.

In simple words we want to minimize the maximum execution time of the n robots that have to visit and execute all tasks assigned to them, guaranteeing that each task is executed by exactly one robot.

The above problem can be seen as a generalization of the classical multi-TSP problem. First, because we are also assuming that tasks should not only be visited by the robots, but should be processed by them. Secondly, because the optimization is carried out over an heterogeneous network due to the heterogeneity of the agents and the tasks. Similar problems have been recently addressed in the literature, see e.g. [7], but to the best of our knowledge, never under the assumption of heterogeneous agents and tasks.

Let us conclude this section with the introduction of some notation that will be used in the remaining of the paper. Let $\mathcal{K}_r$ be the set of tasks assigned to robot R_r. We denote as $\tilde{\mathcal{K}}_r$ the ordered set with the same elements of $\mathcal{K}_r$, but whose ordering specifies the order in which tasks in $\mathcal{K}_r$ are visited by robot R_r. Therefore, sets $\tilde{\mathcal{K}}_r$ are the unknown variables of the optimization problem (1).

Finally, let $\tilde{\mathcal{K}} = \{\tilde{\mathcal{K}}_1, \ldots, \tilde{\mathcal{K}}_n\}$ be an ordered set of n ordered sets, that summarizes the generic solution of the considered tasks allocation problem. The set $\tilde{\mathcal{K}}$ is called *network state*.

3. Optimal centralized solution

In this section we first discuss a centralized strategy that leads to an optimal solution of the above task assignment problem. Such an approach is based on mixed linear integer programming (MILP). Then we provide a characterization of the optimal solution in terms of an upper and a lower bound on the optimal value of the objective function. This will be useful when evaluating the effectiveness of the decentralized approach proposed in the next section.

To represent all possible directed tours of n robots, let us define a complete directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where:

- $\mathcal{V} = \mathcal{N} \cup \mathcal{K}$ is the set of $n + k$ nodes;
- $\mathcal{E} = (\mathcal{N} \cup \mathcal{K}) \times (\mathcal{N} \cup \mathcal{K})$ is the set of $(n + k)^2$ edges representing directed paths from the depots in which robots are initially placed to tasks, and viz, and from tasks to tasks[1].

Moreover, we define the following binary variables that completely identify a task allocation and the order in which tasks are executed by robots. In simple words they completely identify a network state $\tilde{\mathcal{K}}$. Since we want to minimize the total execution times of robots, we always assume that distances among tasks, and among tasks and depots, are covered through straight lines.

- We assign n binary variables x_{ir} to each node $i \in \mathcal{V}$; here $r \in \mathcal{N}$: if $i \in \mathcal{N}$, $x_{ir} = 1$ means that robot R_r starts its tour from node i, while if $i \in \mathcal{K}$, $x_{ir} = 1$ means that task i is executed by robot R_r.

[1] In the sets $\mathcal{V}$ and $\mathcal{E}$ the generic r-th depot is identified via the r-th element in $\mathcal{N}$. This has been done for clearity of presentation as it will appear in the following.

- We assign n binary variables y_{ijr} to each edge $(i,j) \in \mathcal{E}$; here $r \in \mathcal{N}$: $y_{ijr} = 1$ means that robot R_r goes directly from node i to node j in its path.

Moreover, we introduce the following cost coefficients.

- We assign n costs $c_{ir} = c_i/w_r$ to each node $i \in \mathcal{K}$; here $r \in \mathcal{N}$: c_{ir} represents the execution time of task i by robot R_r with an execution speed of w_r.
- We assign n costs $d_{ijr} = l_{ij}/v_r$ to each edge $(i,j) \in \mathcal{E}$; here $r \in \mathcal{N}$: d_{ijr} represents the time spent by robot R_r to pass the length l_{ij} of edge (i,j) with speed v_r.

Proposition 3.1. *Let us consider the allocation problem formalized in Section 2. An optimal solution can be computed solving the following MILP problem:*

$$
\begin{cases}
J = \min \lambda & & \\
s.t. & & \\
\sum_{i \in \mathcal{K}} x_{ir} c_{ir} + \sum_{(i,j) \in \mathcal{E}} d_{ijr} y_{ijr} < \lambda, & \forall r \in \mathcal{N} & (a) \\
x_{rr} = 1, & \forall r \in \mathcal{N} & (b) \\
\sum_{r \in \mathcal{N}} x_{ir} = 1, & \forall i \in \mathcal{K} & (c) \\
\sum_{j \in \mathcal{V}} y_{jir} = \sum_{j \in \mathcal{V}} y_{ijr} = x_{ir}, & \forall i \in \mathcal{V},\ \forall r \in \mathcal{N} & (d) \\
\sum_{i \notin \mathcal{S}} \sum_{j \in \mathcal{S}} y_{ijr} \geq x_{qr} & \forall \mathcal{S} \subseteq \mathcal{K}, & \\
 & \forall q \in \mathcal{S},\ \forall r \in \mathcal{N} & (e) \\
\lambda \in \mathbb{R} & & (f) \\
x_{ir} \in \{0,1\} & \forall i \in \mathcal{V},\ \forall r \in \mathcal{N} & (g) \\
y_{ijr} \in \{0,1\} & \forall (i,j) \in \mathcal{E},\ \forall r \in \mathcal{N}. & (h)
\end{cases}
$$

Proof: The proof is carried out via a detailed explanation of all the constraints and the objective function.

— *Constraints (a)* and *objective function:* The left hand side term of (a) is equal to the total execution time of robot R_r. Thus, given the objective function, constraints (a) aim to minimize the maximum execution time of robots.

— *Constraints (b):* These constraints force each robot to move from its initial position (depot).

— *Constraints (c):* Each task i must be executed by exactly one robot.

— *Constraints (d):* If robot R_r executes task i, it must arrive at node i in some way and at the end of the execution has to leave it. The same holds if node i models a depot, i.e., $i \in \mathcal{N}$.

— *Constraints (e):* Each robot R_r has to make a single connected tour visiting all its tasks, so we have to exclude all the disjoint paths. In words constraint (e) relative to robot R_r, imposes that if robot R_r executes a task

$i \in \mathcal{S} \subseteq \mathcal{K}$, there must be an edge passed by R_r to enter in $\mathcal{S}$. These constraints are named *Subtour Elimination Constraints* (SEC) and are typical of vehicle routing problems and TSP models [3]. □

The number of unknowns in the MILP (3.1) is equal to

$$N = n(n+k)^2 + n(n+k) + 1 = \mathcal{O}(n^3 + nk^2 + n^2k).$$

The total number of constraints is $\mathcal{O}(n^2k + nk2^k)$. Indeed we have n constraints of type (a), n constraints of type (b), k constraints of type (c), $(n+k)n$ constraints of type (d), and $n\sum_{i=1}^{k} i\frac{k!}{(k-i)!i!} \leq nk2^k$ constraints of type (e).

The following two theorems provide a characterization of the optimal value of the performance index J^*.

Theorem 3.2. *The optimal solution J^* of the objective function* (1) *is upper bounded by*

$$J^* \leq C_{up} + D_{up} \tag{2}$$

where

$$C_{up} = \frac{1}{n}\left(\frac{TSP(\mathcal{K})}{v_{min}} + \frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}\right), \tag{3}$$

$$D_{up} = 2\frac{d_{max}}{v_{min}} + \frac{c_{max}}{w_{min}}. \tag{4}$$

Proof: The proof is based on an heuristic that can be summarized in the following main steps.

- Generate an optimal tour that visits all tasks. Obviously, if an agent with speed v_{min} and execution speed w_{min} follows the tour and executes all tasks, its service time is equal to

$$\hat{J} = \left(\frac{TSP(\mathcal{K})}{v_{min}} + \frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}\right).$$

- Divide the tour in n consecutive sub-tours using the following rule. Take a robot (e.g. R_1) at random and make it follow the route of the optimal single vehicle tour at the previous item, starting from the position of an arbitrary task. Stop it as soon as its service time $\hat{J}_1$ satisfies the condition $\hat{J}_1 \geq \hat{J}/n$. Now, since the largest cost of tasks is equal to c_{max}, the smallest execution speed of robots is w_{min}, and the time taken to travel between tasks is continuous, it is

$$\hat{J}_1 \leq \frac{\hat{J}}{n} + \frac{c_{max}}{w_{min}}.$$

  Select at random a new robot (e.g. R_2) and put it at the end of the route of R_1 and repeat the same strategy, until all robots are considered. If there aren't enough tasks for the robots, simply consider null the service time for the remaining robots.

- Now, if d_{max} is the maximum length of the shortest path between any two points in the region $\mathcal{R}$, the execution time J_r of each robot R_r is such that $J_r \leq \hat{J}_r + 2d_{max}/v_{min}$. Indeed the total service time of each robot corresponds to the time it takes to complete its sub-tour along the route of the optimal single vehicle TSP, plus the time to go from its depot to its first task and go back to the depot. Therefore, it is

$$J_r \leq \frac{\hat{J}}{n} + \frac{c_{max}}{w_{min}} + 2\frac{d_{max}}{v_{min}}, \qquad \forall r \in \mathcal{N}.$$

Since the optimal solution J^* of the objective function (1) can only be smaller or equal than the solution resulting from the above heuristic, for sure it is

$$J^* \leq \max_{r \in \mathcal{N}} J_r \leq \frac{\hat{J}}{n} + \frac{c_{max}}{w_{min}} + 2\frac{d_{max}}{v_{min}} = C_{up} + D_{up}$$

thus proving the correctness of the upper bound. □

Theorem 3.3. *The optimal solution J^* of the objective function* (1) *is lower bounded by*

$$J^* \geq C_{lo} - D_{lo} \tag{5}$$

where

$$C_{lo} = \frac{1}{n}\left(\frac{TSP(\mathcal{K})}{v_{max}} + \frac{\sum_{i \in \mathcal{K}} c_i}{w_{max}}\right), \tag{6}$$

$$D_{lo} = \frac{d_{max}}{v_{min}}. \tag{7}$$

Proof: Let $S_{opt} = \sum_{r \in \mathcal{N}} J_r^*$ be the sum of all the service times corresponding to an optimal task assignment. Since, by definition $J^* = \max_{r \in \mathcal{N}} J_r^*$, obviously it is

$$J^* \geq \frac{S_{opt}}{n}. \tag{8}$$

Now, let S_{opt}^p be the sum of the contributions to J_r^*, with $r \in \mathcal{N}$, relative to the only time spent moving from one task to another one, or from/toward the depots, without including the time spent to execute tasks.

Obviously, it is

$$S_{opt} \geq S_{opt}^p + \frac{\sum_{i \in \mathcal{K}} c_i}{w_{max}}. \tag{9}$$

Moreover, trivially generalizing the result in [7] to the case of heterogeneous robots, we have that

$$S_{opt}^p + \frac{TSP(\mathcal{D})}{v_{min}} \geq \frac{TSP(\mathcal{D} \cup \mathcal{K})}{v_{max}} \geq \frac{TSP(\mathcal{K})}{v_{max}} \tag{10}$$

or equivalently

$$S_{opt}^p \geq \frac{TSP(\mathcal{K})}{v_{max}} - \frac{TSP(\mathcal{D})}{v_{min}}. \tag{11}$$

By equations (9) and (11) it follows that

$$\begin{aligned} S_{opt} &\geq \frac{TSP(\mathcal{K})}{v_{max}} - \frac{TSP(\mathcal{D})}{v_{min}} + \frac{\sum_{i\in\mathcal{K}} c_i}{w_{max}} \\ &\geq \frac{TSP(\mathcal{K})}{v_{max}} - n\frac{d_{max}}{v_{min}} + \frac{\sum_{i\in\mathcal{K}} c_i}{w_{max}}. \end{aligned} \tag{12}$$

Finally, by equations (8) and (12), it is

$$\begin{aligned} J^* \geq \frac{S_{opt}}{n} &= \frac{1}{n}\left(\frac{TSP(\mathcal{K})}{v_{max}} + \frac{\sum_{i\in\mathcal{K}} c_i}{w_{max}}\right) - \frac{d_{max}}{v_{min}} \\ &= C_{lo} - D_{lo} \end{aligned} \tag{13}$$

thus proving the statement. □

4. Decentralized solution based on optimal local task assignment

In this section we first propose a decentralized approach to solve the task allocation problem in Section 2 that is based on gossip. Then, a comparison among the computational complexity of the proposed algorithm and the centralized algorithm is provided. Convergence properties of the gossip algorithm are discussed. Finally, some characterizations of the solution obtained via the decentralized approach are proposed.

4.1. MILP Gossip algorithm

The idea of the proposed decentralized algorithm is that robots locally balance their loads according to a gossip interaction rule, i.e., via pairwise communications, under the following main assumption:

(A1) All robots may interact with all the other robots.

Starting from an initial task assignment, e.g., assuming that robots have the same number of tasks, a couple of robots is selected at random. Selected robots optimally balance their load; a new couple of robots is selected and so on, until no better balancing among robots can be obtained. This can be summarized in Algorithm 1. The variable T_{max} denotes a maximum number of steps to be executed that is assumed to be large enough so that no further improvement of the objective function can be obtained.

4.2. Computational complexity of the local optimization

Let us now discuss the advantages in terms of computational complexity coming from local optimizations using Algorithm 1 with respect to a centralized optimization.

To this aim, let us first present some preliminary results. In particular, the following proposition ensures that when the number of iterations of Algorithm 1 increases, the optimal value of the objective function can never increase. Obviously this does not imply that an optimal solution is obtained.

Algorithm 1 MILP Gossip algorithm

1. Tasks are initially assigned to robots so that each robot has either k/n or $k/n + 1$ tasks.
2. Let $t = 0$.
3. While $t \leq T_{max}$
   (a) Choose at random two robots r and q. Let them solve the MILP (3.1) where $\mathcal{N} = \{r, q\}$ and $\mathcal{K} = \mathcal{K}_r \cup \mathcal{K}_q$.
   (b) If the new task assignment leads to a smaller total execution time, then update the assignments of robots r and q accordingly, else leave them unchanged.
   (c) Let $t = t + 1$ and go back to Step 3.
4. All robots process their own set of tasks following the order specified by the optimal local solution.

Proposition 4.1. *Let $J_{gossip}(t)$ be the maximum execution time of robots computed after t iterations of Algorithm 1. For any $t \geq 0$, it is $J_{gossip}(t+1) \leq J_{gossip}(t)$.*

Proof: Let R_r and R_q be the two robots selected at time $t + 1$. By Algorithm 1 this means that only the tasks allocation of such robots may change, while the load of all the other robots keeps unaltered. Now, since at Step 3.a of Algorithm 1 tasks are assigned to robots R_r and R_q so as to minimize the maximum execution time among them, this implies that the maximum execution time among R_r and R_q either decreases or it keeps unaltered at time $t+1$. Moreover, the maximum execution time among all robots may decrease at time $t + 1$ if and only if either R_r or R_q, or both, are the robots to which it corresponds the maximum execution time among all robots at time t. Indeed with no loss of generality, we may assume that R_r is the "critical" robot at time t, i.e., the robot to which it corresponds the maximum execution time among all robots at time t. Three different cases may occur at time $t + 1$, after the new tasks allocation. First, R_r may still be the robot with the maximum execution time, but in such a case for sure, its execution time cannot be larger than that at time t. Secondly, robot R_q may be at time $t+1$ the robot with the maximum execution time but for sure its execution time cannot be larger than that of robot R_r at time t. Finally, at time $t+1$, neither to R_r nor to R_q it corresponds the maximum execution time among robots. This implies that a third robot, e.g., R_p, has become the critical one at time $t + 1$. In any case for sure its execution time is smaller than that of robot R_r at time t, since by assumption robot R_r was the critical robot at time t. □

Let us now provide an upper bound on the value of the maximum execution time of robots resulting from Algorithm 1 at a generic iteration t. To this aim, we first recall some deterministic upper bounds to the maximum length of the shortest path (SP) between a set $\mathcal{K}$ of k locations in a unit square area, that

are due to [11] and [16], respectively:

$$SP(\mathcal{K}) \leq \sqrt{2}\sqrt{k} + 7/4, \tag{14}$$

and

$$SP(\mathcal{K}) \leq 0.984\sqrt{2}\sqrt{k} + 11. \tag{15}$$

To the best of our knowledge the above two upper bounds are the best actually proposed in the literature. Moreover, we cannot a priori say which of the above bounds is the most strict one. Indeed the bound in [16] has a smaller multiplicative factor with respect to [11], but has a larger additive constant. In the following, we focus on upper bound (14), but obviously similar results can be repeated considering (15).

Proposition 4.2. *Let $J_{gossip}(t)$ be the maximum execution time of robots computed after t iterations of Algorithm 1, then $\forall\ t \geq 0$ it is*

$$J_{gossip}(t) \leq \left(\sqrt{2}\sqrt{\frac{k}{n} + 2} + \frac{7}{4} + \sqrt{2}\right)\frac{d_{max}}{v_{min}} + \left(\frac{k}{n} + 1\right)\frac{c_{max}}{w_{min}}.$$

Proof: By Algorithm 1 at time $t = 0$ the maximum number of tasks that can be assigned to a robot is equal to $k/n+1$. Moreover, since each robot starts its path from its depot and has to come back to it, then by equation (14), for any $r \in \mathcal{N}$ it is

$$TSP(\mathcal{K}_r(0) \cup \{D_r\}) \leq \left(\sqrt{2}\sqrt{\frac{k}{n} + 2} + \frac{7}{4} + \sqrt{2}\right) d_{max}. \tag{16}$$

Note that the additional term $\sqrt{2}$ between parenthesis comes from the fact that to form a Euclidean TSP tour from a path in a unit square it is sufficient to connect the start and end point to form a cycle, thus increasing the size of the path of at most $\sqrt{2}$ in the unit square. Moreover, d_{max} comes from the fact that in our problem statement depots and robots are not distributed in a square of unitary edge, but in a region $\mathcal{R}$ that is contained in a square of edge d_{max} being by definition d_{max} the maximum length of the shortest path between any two points in $\mathcal{R}$.

Finally, since by assumption $\sum_{i \in \mathcal{K}_r(0)} c_i \leq \left(\frac{k}{n} + 1\right) c_{max}$, it follows that

$$J_{gossip}(0) \leq \left(\sqrt{2}\sqrt{\frac{k}{n} + 2} + \frac{7}{4} + \sqrt{2}\right)\frac{d_{max}}{v_{min}} + \left(\frac{k}{n} + 1\right)\frac{c_{max}}{w_{min}}$$

that proves the statement being by Proposition 4.1 $J_{gossip}(t) \leq J_{gossip}(0)$ for all $t \geq 0$. □

Let us now provide a proposition that characterizes the maximum number of tasks that are assigned to robots at a generic iteration t of Algorithm 1.

Proposition 4.3. *Let $K_{max}(t) = \max_{r\in\mathcal{N}} |\mathcal{K}_r(t)|$ be the maximum number of tasks that are assigned to robots at a generic iteration t of Algorithm 1. For any $t \geq 0$ it is:*

$$K_{max}(t) \leq \frac{w_{max}}{c_{min}} \left[\left(\sqrt{2}\sqrt{\frac{k}{n}+2} + \frac{7}{4} + \sqrt{2} \right) \frac{d_{max}}{v_{min}} + \left(\frac{k}{n} + 1 \right) \frac{c_{max}}{w_{min}} \right]. \tag{17}$$

Proof: By Proposition 4.2, for all $t \geq 0$, it holds

$$J_{gossip}(t) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+2} + \frac{7}{4} + \sqrt{2} \right) \frac{d_{max}}{v_{min}} + \left(\frac{k}{n} + 1 \right) \frac{c_{max}}{w_{min}}. \tag{18}$$

Now, it is

$$J_{gossip}(t) \geq \frac{K_{max}(t) c_{min}}{w_{max}} \tag{19}$$

since the execution time of $K_{max}(t)$ tasks is greater or equal than that we have if such tasks are at a null distance from the robot that has to process them, all tasks have a cost equal to c_{min} and the robot who process them has an execution speed equal to w_{max}. By equations (18) and (19) the statement of the proposition follows. □

An important remark needs to be done. The above proposition provides an upper bound on the maximum number of tasks that can be assigned to a robot at any iteration. For particular values of the parameters it may happen that the upper bound given by Proposition 4.3 is not significant because it is larger than k. However, this only occurs for very particular cases, while for most of the significant and general situations where the number of tasks is sufficiently large, robots and tasks are sufficiently distributed in $\mathcal{R}$ and their costs and speeds are in reasonable ratio, Proposition 4.3 enables us to conclude that

$$K_{max}(t) = \mathcal{O}(k/n).$$

Now, since local optimization considers two robots at a time, the number of tasks that are involved in a local optimization is surely smaller or equal than $2K_{max}(t)$. This means that the number of unknowns of the MILP that should be solved at the generic iteration t of Algorithm 1 is

$$N_{gossip} = \mathcal{O}(k^2/n^2)$$

rather than $N = \mathcal{O}(n^3 + nk^2 + n^2k)$ as in the centralized case. Moreover, the number of constraints is $\mathcal{O}(k2^{k/n}/n)$ rather than $\mathcal{O}(n^2k + nk2^k)$ as in the centralized case.

4.3. Finite time and almost sure convergence

We now introduce two definitions to formalize two important properties of gossip communication schemes, namely *deterministic persistence* and *stochastic persistence*. Similar definitions have been recently proposed in [4]. As usual in this framework, we assume that the possible interactions among agents are modeled by an undirected graph $G = \{V, E\}$ where agents correspond to vertices, and an edge exists if and only if the interaction among the agents corresponding to the incidence nodes is possible. Obviously, assumption **(A1)** implies that in our case it is $E = V \times V$. At each iteration t of the gossip algorithm a different edge is selected. In the following we denote as $e(t)$ the edge selected at time t, while the set of edges selected in the time interval $[t_1, t_2]$ is denoted as $\bar{e}(t_1, t_2)$, i.e., we have

$$\bar{e}(t_1, t_2) = \bigcup_{t=t_1}^{t_2} e(t).$$

Definition 4.4 (Deterministic persistence).
A gossip communication scheme is said to be deterministically persistent *if* $\forall t \geq 0$ *there exists a finite* $T > 0$ *such that*

$$\forall e' \in E, \qquad Pr(e' \in \bar{e}(t, t+T)) = 1$$

or equivalently, $\bar{e}(t, t+T) = E$. ■

Deterministic persistence implies that, if we consider a finite but sufficiently large time interval, then for sure all arcs are selected at least once during such interval.

Definition 4.5 (Stochastic persistence).
A gossip communication scheme is said to be stochastically persistent *if* $\forall t \geq 0$ *there exists a finite* $T > 0$ *and a probability* $p \in (0, 1)$ *such that*

$$\forall e' \in E, \qquad Pr(e' \in \bar{e}(t, t+T)) \geq p$$

where $Pr(\cdot)$ *denotes a probability.* ■

In simple words, stochastic persistence implies that, if we consider a finite but sufficiently large time interval, then each edge has a probability greater or equal than a finite value p of being selected during such an interval.

Theorem 4.6. *Let* $\tilde{\mathcal{K}}(t)$ *be the network state resulting at time* t *from the execution of Algorithm 1. If the gossip communication scheme satisfies the* deterministic persistence *property then, for every initial task assignment, there exists a network state* $\tilde{\mathcal{K}}^*_{gossip}$ *and a finite time* $T > 0$ *such that* $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{gossip}$, *for all* $t \geq T$.

Proof: Let us present some preliminary comments.

— First, $\tilde{\mathcal{K}}^*_{gossip}$ is an invariant network state for the state evolution following Algorithm 1. This follows from Step 3.b of Algorithm 1.

— Secondly, if at a given time the network state is updated then the previous network state is no more visited during the algorithm evolution. This also follows from Step 3.b of Algorithm 1 and the monotonicity property expressed by Proposition 4.1.

— Thirdly, the number $N_{n,k}$ of admissible network states is finite since both the number of robots and the number of tasks are finite.

Now, with no loss of generality we assume that at the initial time $t = 0$ it is $\tilde{\mathcal{K}}_r \neq \tilde{\mathcal{K}}^*_{gossip,r}$ for all $r = 1, \ldots, n$, i.e., no robot is in its final assignment. If the communication scheme among agents is deterministically persistent, since the graph modeling the possible interactions among robots is fully connected and the number $N_{n,k}$ of admissible network states is finite, then for sure after some finite time T_0 the robot with the maximum cost in the final assignment reaches its final assignment. Let R_r be such a robot. By Step 3.b of Algorithm 1 this implies that the assignment of R_r is no more changed during the algorithm evolution, i.e., $\tilde{\mathcal{K}}_r(t) = \tilde{\mathcal{K}}^*_{gossip,r}$ for all $t \geq T_0$.

Analogously, after some further finite time T_1 the final assignment is reached by the robot with the second largest cost, and so on, until all robots have reached their final assignment. Since all T_i's are finite, this proves that the final network state $\tilde{\mathcal{K}}^*_{gossip}$ is reached in a finite time $T = \sum_{i=1}^{n} T_i$. □

Theorem 4.7. *Let $\tilde{\mathcal{K}}(t)$ be the network state resulting at time t from the execution of Algorithm 1. If the gossip communication scheme satisfies the* stochastic persistence *property, then, for every initial task assignment, there exists a network state $\tilde{\mathcal{K}}^*_{gossip}$ and almost surely a finite time $T > 0$ such that $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{gossip}$ for all $t \geq T$, i.e., the network state converges almost surely in finite time to $\tilde{\mathcal{K}}^*_{gossip}$.*

Proof: We prove this theorem following the same arguments an in [6]. The proof is based on verifying the following three facts:

(i) $\tilde{\mathcal{K}}^*_{gossip}$ is an invariant network state for the state evolution following Algorithm 1;

(ii) $\tilde{\mathcal{K}}(t)$ is a Markov process on a finite number of states;

(iii) starting from any initial network state $\tilde{\mathcal{K}}(0)$, there is a positive probability for the network state to reach $\tilde{\mathcal{K}}^*_{gossip}$ in a finite number of steps.

Let us now check the above three properties in order.

— (i) As already discussed in Theorem 4.6, this follows from Step 3.b of Algorithm 1.

— (ii) As already discussed in the proof of Theorem 4.6, the number of admissible network states $N_{n,k}$ is finite, being finite both the number of robots and the number of tasks. Markovianity immediately follows from the fact that subsequent random selection of edges are independent.

— (iii) This issue can be proved using similar arguments as in Theorem 4.6 with the only difference that now the communication scheme is stochastically persistent, rather than deterministically persistent. This implies that for any

initial network state $\tilde{\mathcal{K}}(0)$ there is a finite probability that after some finite time T_0 the robot with the maximum cost in the final assignment reaches its final assignment, that is no more changed during the algorithm evolution. The same holds for the robot with the second largest execution cost in the final assignment, and so, until the invariant network state $\tilde{\mathcal{K}}^*_{gossip}$ is reached. Since the number of possible states is finite, item (iii) holds. □

4.4. Performance characterization of the MILP algorithm

Algorithm 1 does not guarantee the convergence to an optimal solution. However, some results can be given to characterize its solution at the equilibrium, i.e., after a number of iterations that is sufficiently large so that no better balancing among robots may be obtained. In particular, the following theorem provides a characterization of the maximum distance among the processing times of robots that have locally balanced their loads.

Theorem 4.8. *Let $J^*_{gossip,r}$ and $J^*_{gossip,q}$, respectively, be the total execution times of two generic robots R_r and R_q resulting from the application of Algorithm 1. It holds*

$$|J^*_{gossip,r} - J^*_{gossip,q}| \leq K_{rq} = 2\frac{d^{rq}_{max}}{v^{rq}_{min}} + \frac{c^{rq}_{max}}{w^{rq}_{min}} \tag{20}$$

where d^{rq}_{max} is the maximum distance among tasks in $\mathcal{K}_r$ and tasks in $\mathcal{K}_q$, $v^{rq}_{min} = \min\{v_r, v_q\}$, and $w^{rq}_{min} = \min\{w_r, w_q\}$.

Proof: We prove the statement by contradiction, i.e., we assume that

$$|J^*_{gossip,r} - J^*_{gossip,q}| > K_{rq}.$$

With no loss of generality, we assume that it is $J^*_{gossip,r} > J^*_{gossip,q}$. Now, let z be the task in $\mathcal{K}_r$ whose distance with respect to tasks in $\mathcal{K}_q$ is minimum. Remove z from $\mathcal{K}_r$ and put it in $\mathcal{K}_q$. Let $\tilde{J}_r$ and $\tilde{J}_q$ be the resulting execution times of robots r and q, respectively. Obviously, we have

$$\tilde{J}_q \leq J^*_{gossip,q} + \frac{c_z}{w_q} + 2\frac{d^{rq}_{max}}{v_q} = J^*_{gossip,q} + K_{rq} \tag{21}$$

where the inequality follows from the fact that the optimal TSP of robot q is surely smaller than the path obtained by simply adding twice the path from the closest task in $\mathcal{K}_q$ to z. Now, by the contradictory assumption, we have

$$J^*_{gossip,r} > J^*_{gossip,q} + K_{rq} \tag{22}$$

thus (21) can be rewritten as

$$\tilde{J}_q < J^*_{gossip,r}. \tag{23}$$

As a consequence

$$\max\{\tilde{J}_q, \tilde{J}_r\} < \max\{J^*_{gossip,q}, J^*_{gossip,r}\}. \tag{24}$$

However, this contradicts the assumption that $J^*_{gossip,r}$ and $J^*_{gossip,q}$ are the time executions corresponding to an optimal task assignment, thus proving the statement. □

Corollary 4.9. *Let $J^*_{gossip,r}$ and $J^*_{gossip,q}$, respectively, be the total execution times of two generic robots R_r and R_q resulting from the application of Algorithm 1. It holds*

$$|J^*_{gossip,r} - J^*_{gossip,q}| \leq D_{up} \tag{25}$$

where D_{up} is defined as in equation (4).

Let us now provide a theorem that gives an upper bound on the maximum execution time resulting from the application of Algorithm 1. First, we introduce the following Lemma necessary to the proof of Theorem 4.11.

Lemma 4.10. *Let $S_{gossip}(t)$ be the sum of all J_i's at iteration t of Algorithm 1. Then*

$$\forall t > 0, \quad S_{gossip}(t) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+1} + \frac{7}{4} + \sqrt{2}\right)\frac{n d_{max}}{v_{min}} + \frac{\sum_{j\in\mathcal{K}} c_j}{w_{min}}. \tag{26}$$

Proof: By definition $S_{gossip}(t) = \sum_{i=1}^{n} J_i(t)$. Since

$$J_i(t) = \frac{TSP\left(\mathcal{K}_i(t) \cup \{D_i\}\right)}{v_i} + \frac{\sum_{j\in\mathcal{K}_i} c_j}{w_i},$$

it is

$$S_{gossip}(t) = \sum_{i=1}^{n} \frac{TSP\left(\mathcal{K}_i(t) \cup \{D_i\}\right)}{v_i} + \sum_{i=1}^{n} \frac{\sum_{j\in\mathcal{K}_i} c_j}{w_i}.$$

By considering the worst case scenario in which each agent has speed $v_i = v_{min}$ and task execution speed $w_i = w_{min}$ for $i = 1, \ldots, n$, we have the following straightforward upper bound

$$S_{gossip}(t) \leq \sum_{i=1}^{n} \frac{TSP\left(\mathcal{K}_i(t) \cup \{D_i\}\right)}{v_{min}} + \frac{\sum_{j\in\mathcal{K}} c_j}{w_{min}}. \tag{27}$$

To each robot $k_i(t) = |\mathcal{K}_i(t)|$ tasks are assigned at any given time. By exploiting the result by Few [11] and [16] given in eq. (14) and eq. (15), and taking into account that such results refer to a unit square area, the maximum tour length that each robot has to drive to visit all its assigned tasks is

$$TSP\left(\mathcal{K}_i(t) \cup \{D_i\}\right) \leq \left(\alpha\sqrt{k_i + 1} + \beta\right) d_{\max} \tag{28}$$

where $\alpha, \beta \in \mathbb{R}$ are appropriate constants that depend on the considered bound. Thus, we may now write

$$S_{gossip}(t) \leq \frac{\alpha d_{\max}}{v_{min}} \sum_{i=1}^{n} \left(\sqrt{k_i(t) + 1}\right) + \frac{n\beta d_{\max}}{v_{min}} + \frac{\sum_{j\in\mathcal{K}} c_j}{w_{min}}. \tag{29}$$

The only term in eq. (29) that is affected by the task assignment to the robots is $\sum_{i=1}^{n}\left(\sqrt{k_i(t)+1}\right)$. We now find the task assignment that maximizes the bound in eq. (29) by solving the following optimization problem:

$$\begin{cases} max \sum_{i=1}^{n}\left(\sqrt{k_i+1}\right) \\ s.t. \\ \sum_{i=1}^{n} k_i = k \\ k_i \geq 0 \qquad i=1,\ldots,n \\ k_i \in \mathbb{N} \qquad i=1,\ldots,n \end{cases} \tag{30}$$

Any solution to Problem (30) found by relaxing the constraint to have integer variables is an upper bound to the solution of the given problem. We therefore solve Problem (30) by relaxing the integer constraint using Lagrange multipliers:

$$f(k_1,\ldots,k_n,\lambda) = \sum_{i=1}^{n}\left(\sqrt{k_i+1}\right) + \lambda\left(\sum_{i=1}^{n} k_i - k\right) \tag{31}$$

By setting partial derivatives of the objective function (31) to zero we get

$$\begin{aligned} \frac{\partial f(k_1,\ldots,k_n,\lambda)}{\partial k_i} &= \frac{1}{2\sqrt{k_i+1}} + \lambda = 0 \qquad i=1,\ldots,n \\ \frac{\partial f(k_1,\ldots,k_n,\lambda)}{\partial \lambda} &= \left(\sum_{i=1}^{n} k_i - k\right) = 0 \end{aligned} \tag{32}$$

Thus, for any $i,j \in \mathcal{N}$, it is

$$\frac{1}{2\sqrt{k_i+1}} = \frac{1}{2\sqrt{k_j+1}},$$

i.e., the maximum of function (31) is found for $k_i = \frac{k}{n}$ for all $i \in \mathcal{N}$. Therefore, an upper bound to the solution of Problem (30) is

$$\sum_{i=1}^{n}\left(\sqrt{k_i+1}\right) \leq \sum_{i=1}^{n}\left(\sqrt{\frac{k}{n}+1}\right) = n\sqrt{\frac{k}{n}+1}.$$

Finally, by substituting the solution of (31) into (29)

$$S_{gossip}(t) \leq \alpha n\left(\sqrt{\frac{k}{n}+1}+\beta\right)\frac{d_{max}}{v_{min}} + \frac{\sum_{j\in\mathcal{K}} c_j}{w_{min}}. \tag{33}$$

If we consider the results by Few (14) we get

$$S_{gossip}(t) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+1}+\frac{7}{4}+\sqrt{2}\right)\frac{nd_{max}}{v_{min}} + \frac{\sum_{j\in\mathcal{K}} c_j}{w_{min}}. \tag{34}$$

□

We are now ready to state one of the main results of this paper.

Theorem 4.11. *The maximum execution time J^*_{gossip} resulting from the application of Algorithm 1 satisfies*

$$J^*_{gossip} \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+1}+\frac{7}{4}+\sqrt{2}\right)\frac{d_{max}}{v_{min}}+\frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}+D_{up}. \tag{35}$$

Proof: Let $S_{gossip}(t)$ be the sum of all J_i's at iteration t of Algorithm 1. By Lemma 4.10 we have

$$S_{gossip}(t) \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+1}+\frac{7}{4}+\sqrt{2}\right)\frac{nd_{max}}{v_{min}}+\frac{\sum_{j\in\mathcal{K}} c_j}{w_{min}} \tag{36}$$

Let $J^*_{gossip,min}$ be the smallest execution time between the vehicles after the execution of Algorithm 1. Corollary 4.9 implies $J^*_{gossip,min} \geq J^*_{gossip} - D_{up}$. Moreover, $\forall t \geq 0$ it obviously is

$$J^*_{gossip,min}(t) \leq \frac{1}{n} S_{gossip}(t) \tag{37}$$

thus

$$\begin{aligned} J^*_{gossip} &\leq J^*_{gossip,min} + D_{up} \leq \frac{1}{n} S_{gossip}(t) + D_{up} \\ &\leq \left(\sqrt{2}\sqrt{\frac{k}{n}+1}+\frac{7}{4}+\sqrt{2}\right)\frac{d_{max}}{v_{min}}+\frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}+D_{up}. \end{aligned} \tag{38}$$

proving the statement. □

4.5. Asymptotic behavior

We now study what is the performance to expect from the proposed algorithm in the limit cases in which the ratio between tasks and robots goes to infinity. In particular we obtain the following result.

Proposition 4.12. *Let J^*_{gossip} be the maximum execution time resulting from the application of Algorithm 1 and let J^* be the optimal solution to the HMVR problem. Then*

$$\lim_{\frac{k}{n}\to\infty} \frac{J^*_{gossip}}{J^*} \leq \frac{c_{max}}{c_{min}}\frac{w_{max}}{w_{min}}. \tag{39}$$

Proof: By taking the ratio between the upper bound to J^*_{gossip} given in Theorem 4.11 and the lower bound of the optimal solutions to the HMVR problem J^* given in eq.(5) we get

$$\lim_{\frac{k}{n}\to\infty} \frac{J^*_{gossip}}{J^*} \leq \frac{\left(\sqrt{2}\sqrt{\frac{k}{n}+1}+\frac{7}{4}+\sqrt{2}\right)\frac{d_{max}}{v_{min}}+\frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}+D_{up}}{\frac{1}{n}\left(\frac{TSP(\mathcal{K})}{v_{max}}+\frac{\sum_{i\in\mathcal{K}} c_i}{w_{max}}\right)-D_{lo}}. \tag{40}$$

The term $\frac{1}{n}\frac{TSP(\mathcal{K})}{v_{max}}$, being at the denominator, can be lower bounded by zero. The term $\frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}}$ at the numerator can be upper bounded by $\frac{k}{n}\frac{c_{max}}{w_{min}}$ while the equivalent term $\frac{\sum_{i\in\mathcal{K}} c_i}{w_{max}}$ at the denominator can be lower bounded by $\frac{k}{n}\frac{c_{min}}{w_{max}}$. Therefore, we get

$$\lim_{\frac{k}{n}\to\infty} \frac{J^*_{gossip}}{J^*} \leq \frac{\left(\sqrt{2}\sqrt{\frac{k}{n}+1}+\frac{7}{4}+\sqrt{2}\right)\frac{d_{max}}{v_{min}}+\frac{k}{n}\frac{c_{max}}{w_{min}}+D_{up}}{\frac{1}{n}\frac{TSP(\mathcal{K})}{v_{max}}+\frac{k}{n}\frac{c_{min}}{w_{max}}-D_{lo}}. \tag{41}$$

The term $\frac{k}{n}$ dominates both on the constants and on the term $\sqrt{\frac{k}{n}}$, thus we get

$$\lim_{\frac{k}{n}\to\infty} \frac{J^*_{gossip}}{J^*} \leq \frac{c_{max}}{c_{min}}\frac{w_{max}}{w_{min}}. \tag{42}$$

proving the statement. □

5. An heuristic gossip algorithm

In this section we present a new algorithm, called the Decentralized Heuristic Algorithm, and discuss its convergence properties and computational complexity in comparison with the algorithm in the previous section.

The robots update their states following Algorithm 2, while the task exchange rule is described in Algorithm 3. The basic idea is as follows. When two robots are selected at step 3.*a* of Algorithm 2, the two agents start to balance their execution time by the iterative execution of Algorithm 3. At each execution of Algorithm 3 only two scenarios are possible:

- the sets of assigned tasks of the two robots do not change;
- one task is given by the robot with the higher execution time to the other robot.

Note that the determination of the possible exchanges is made by the computation of the Approximated Euclidean TSP ($ATSP$), thus, unlike in the MILP gossip algorithm, this approach involves polynomial time algorithms. There exist a vast literature on polynomial time algorithms to compute approximations to the Euclidean TSP such that

$$ATSP \leq \alpha TSP$$

where TSP denotes the value of the optimal TSP and α represents the worst case ratio. In [23] some heuristics for the TSP problem are summarized. Many heuristics are based on the computation of the Minimum Spanning Tree (MST) among the nodes and guarantee a worst case ratio of $\alpha = 2$ with a running

time of $\mathcal{O}(m^2)$, where m denotes the number of nodes to be visited. Another polynomial time heuristic based on MST which provides a value of $\alpha = 1.5$ is the Christofides algorithm described in [8], which is characterized by a running time of $\mathcal{O}(m^3)$.

We observe that the **STOP** of Algorithm 3 ensures that after the execution of Algorithm 3 it holds

$$\max\{J_r(t+1), J_q(t+1)\} \leq \max\{J_r(t), J_q(t)\}$$

whatever is the choice of the algorithm to compute the value of the $ATSP$.

Algorithm 2 Decentralized Heuristic Algorithm

1. Tasks are initially arbitrarily assigned to robots.
2. Let $t = 0$.
3. While $t \leq T_{max}$
   (a) Select two robot R_p and R_r at random.
   (b) Apply Algorithm 3 repeatedly on R_p and R_r until no more task exchanges are possible.
   (c) Let $t = t + 1$ and go back to Step 3.
4. All robots process their own set of tasks following the order specified by the local solution of an ATSP Algorithm.

As a final remark we note that conditions can be given on the gossip communication scheme which allow the robot to converge to stable task assignment in a finite time. In particular, the following two theorems can be given, whose proofs are omitted here because they follow the same lines of Theorems 4.6 and 4.7, respectively.

Theorem 5.1. *Let $\tilde{\mathcal{K}}(t)$ be the network state resulting at time t from the execution of Algorithm 2. If the gossip communication scheme satisfies the* deterministic persistence *property then, for every initial task assignment, there exists a network state $\tilde{\mathcal{K}}^*_{heur}$ and a finite time $T > 0$ such that $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{heur}$, for all $t \geq T$.*

Theorem 5.2. *Let $\tilde{\mathcal{K}}(t)$ be the network state resulting at time t from the execution of Algorithm 2. If the gossip communication scheme satisfies the* stochastic persistence *property, then, for every initial task assignment, there exists a network state $\tilde{\mathcal{K}}^*_{heur}$ and almost surely a finite time $T > 0$ such that $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{heur}$ for all $t \geq T$, i.e., the network state converges almost surely in finite time to $\tilde{\mathcal{K}}^*_{heur}$.*

5.1. Computational complexity of the local optimization

In this section we discuss about the advantages of the proposed heuristic in terms of computational complexity with respect to the MILP gossip algorithm.

Let us begin with the analysis of the computational complexity of the single task exchange rule in Algorithm 3. The following proposition characterizes the running time of Algorithm 3.

Algorithm 3 Local Balancing between robots R_r and R_q

- **INPUT:** $\mathcal{K}_r(t)$ and $\mathcal{K}_q(t)$.
- **OUTPUT:** $\mathcal{K}_r(t+1)$ and $\mathcal{K}_q(t+1)$.
- **ASSUMPTION:** We assume, with no loss of generality, that $J_r(t) > J_q(t)$.
- **STEPS:**

1. Let $\mathcal{K}_{ex} = \emptyset$, $\mathcal{K}_v = \mathcal{K}_r$ and $F = 0$.
2. **While** $F = 0$ **and** $\mathcal{K}_v \neq \emptyset$
   - Select $i \in \mathcal{K}_v$ randomly.
   - Let $\mathcal{K}_v = \mathcal{K}_v \setminus \{i\}$.
   - Compute
   $$J_{new} = \frac{ATSP(\mathcal{K}_q \cup \{i\})}{v_q} + \frac{\sum_{j \in (K_q \cup \{i\})} c_j}{w_q}.$$
   - **If** $J_{new} < J_r(t)$
     (a) $\mathcal{K}_{ex} = \mathcal{K}_{ex} \cup \{i\}$.
     (b) $F = 1$.

   End While.

- **STOP**:

- $\mathcal{K}_q(t+1) = \mathcal{K}_q(t) \cup \mathcal{K}_{ex}$ and $\mathcal{K}_r(t+1) = \mathcal{K}_r(t) \setminus \mathcal{K}_{ex}$.
- $$J_q(t+1) = \frac{ATSP(\mathcal{K}_q(t+1))}{v_q} + \frac{\sum_{j \in (\mathcal{K}_q(t+1))} c_j}{w_q},$$
$$J_r(t+1) = \min \left\{ J_r(t) - \frac{\sum_{i \in \mathcal{K}_{ex}} c_i}{w_r}, \frac{ATSP(\mathcal{K}_r(t+1))}{v_r} + \frac{\sum_{j \in (\mathcal{K}_r(t+1))} c_j}{w_r} \right\}.$$

Proposition 5.3. *Assume to compute the ATSP using, at step* 2 *of Algorithm 3, an algorithm with a running time of* $\mathcal{O}(k^p)$*. The worst case running time of Algorithm 3 is* $\mathcal{O}(k^{p+1})$*.*

Proof: The maximum number of nodes assigned to a robot is k, thus at each iteration of the while loop of Algorithm 3 the running time of the algorithm to compute the ATSP is at maximum $\mathcal{O}(k^p)$. The while loop can be repeated at maximum k times, as there may be at maximum k tasks exchange. Thus the total running time of Algorithm 3 is $k \cdot \mathcal{O}(k^p) = \mathcal{O}(k^{p+1})$. □.

An important property of the proposed heuristic is presented in the following proposition.

Proposition 5.4. *Let* $J_{heur}(t) = \max_{i \in \mathcal{N}} J_i(t)$ *be the maximum execution time of robots at time* t *resulting from the execution of Algorithm 2. The following holds*

$$\forall t \in \mathbb{N}, \quad J_{heur}(t+1) \leq J_{heur}(t).$$

Proof: The proof directly follows from the update rules of Algorithm 3. Let R_r and R_q be the couple of robots selected by Algorithm 2 at time t with execution time respectively $J_r(t)$ and $J_q(t)$. Let R_{max} be the robot with the maximum execution time at time $t \geq 0$, so it is $J_{max}(t) = J_{heur}(t)$. Now, by Algorithm 3 is holds $\max\{J_r(t+1), J_q(t+1)\} \leq \max\{J_r(t), J_q(t)\}$, and only two cases may occur

- if $R_r, R_q \neq R_{max}$, $J_{heur}(t+1) = J_{heur}(t)$, i.e., the maximum execution time does not change;
- if either $R_r = R_{max}$ or $R_q = R_{max}$, $J_{heur}(t+1) \leq J_{heur}(t)$, i.e., the maximum execution time may be reduced.

□

A similar property was discussed for the MILP gossip algorithm as well: at each iteration of the local optimization rule the maximum execution time can not increase. Note that in the MILP gossip algorithm each local optimization requires to solve a MILP problem, which is an exponential time algorithm. Proposition 5.3 shows that the proposed heuristic is based on a local balance with a considerably smaller computational complexity than the MILP gossip algorithm.

We conclude this section with some considerations about the total number of local interactions required to reach a final task assignment. We conjecture that the expected number of iterations of Algorithm 2 required to converge are of the same order as the number of iterations required in the MILP gossip algorithm. Our conjecture is based on the following observations. The execution of Algorithm 3 leads to a different task assignment only if the maximum execution time among the involved robots can be decreased, otherwise the task assignment does not change. In the proposed framework if at time t the execution of

Algorithm 3 leads to a decrement of the maximum execution time, the network state $\tilde{\mathcal{K}}(t)$ changes to a new one $\tilde{\mathcal{K}}(t+1)$. It follows from Proposition 5.4 that $\tilde{\mathcal{K}}(t)$ is no more visited during the algorithm evolution. This property holds for the MILP gossip algorithm as well. Starting from an initial network state $\tilde{\mathcal{K}}(0)$, in both decentralized solutions all the possible network states may be visited before to reach the equilibrium state. For that reason we can reasonably conjecture that the MILP gossip algorithm and Algorithm 2 have computational complexity of the same order in terms of total number of iterations. Our conjecture is supported also by the results of some simulations presented in the following.

5.2. Characterizations of the heuristic solution

In this section we focus on some properties of J^*_{heur}, i.e., the solution of Algorithm 2 at the equilibrium, when no better balancing among robots may be obtained. As the MILP gossip algorithm, Algorithm 2 does not guarantee the convergence to an optimal solution. Firstly we present a theorem that characterizes the maximum distance among the execution times of two robots that have locally balanced their loads. Then we provide an upper bound on the maximum execution time resulting from the application of Algorithm 2.

Theorem 5.5. *Let $J^*_{r,heur}$ and $J^*_{q,heur}$, respectively, be the total execution times of two generic robots R_r and R_q resulting from the application of Step 2 of Algorithm 2. It holds*

$$|J^*_{r,heur} - J^*_{q,heur}| \leq K_{rq} = 2\frac{d^{rq}_{max}}{v^{rq}_{min}} + \frac{c^{rq}_{max}}{w^{rq}_{min}} \tag{43}$$

where d^{rq}_{max} is the maximum distance among tasks in $\mathcal{K}_r$ and tasks in $\mathcal{K}_q$, $v^{rq}_{min} = \min\{v_r, v_q\}$, and $w^{rq}_{min} = \min\{w_r, w_q\}$.

Proof: Let R_r and R_q be a couple of robots selected in Algorithm 2 at time t with execution time respectively $J_r(t)$ and $J_q(t)$ after t iterations. By step 2 of Algorithm 2 robots R_r and R_q exchange tasks one by one until no more exchanges are possible. Assume, without lack of generality, that at time t it holds $J_r(t) > J_q(t)$. Now, let us assume to exchange one task from R_r to R_q. Surely the execution time of R_r decreases, thus $J_r(t+1) \leq J_r(t)$. On the contrary, the execution time of robot R_q increases but the resulting value is such that:

$$J_q(t+1) \leq J_q(t) + \frac{c^{rq}_{max}}{w_q} + 2\frac{d^{rq}_{max}}{v_q}.$$

Thus, by exchanging one task a reduction of the maximum execution time is guaranteed if

$$J_q(t) + \frac{c^{rq}_{max}}{w_q} + 2\frac{d^{rq}_{max}}{v_q} \leq J_r(t).$$

In other words, if

$$J_r(t) - J_q(t) \geq \frac{c_{max}}{w_q} + 2\frac{d^{rq}_{max}}{v_q}$$

then there exists at east task that can be exchanged such that

$$\max\{J_q(t+1), J_r(t+1)\} < \max\{J_q(t), J_r(t)\}.$$

Since the number of possible task assignments is finite and at each iteration of Algorithm 3 the local maximum may be decreased due to a task exchange, some of these configurations are never visited again. Thus we have that in finite time

$$|J^*_{r,heur} - J^*_{q,heur}| \leq K_{rq} = 2\frac{d^{rq}_{max}}{v^{rq}_{min}} + \frac{c^{rq}_{max}}{w^{rq}_{min}}$$

□

By Theorem 5.5 and the fact that each robot interacts with any other sufficiently often, a significant result follows.

Corollary 5.6. *Let $J^*_{r,heur}$ and $J^*_{q,heur}$, respectively, be the total execution times of two generic robots R_r and R_q resulting from the application of Algorithm 2. It holds*

$$|J^*_{r,heur} - J^*_{q,heur}| \leq D_{up} \tag{44}$$

where

$$D_{up} = 2\frac{d_{max}}{v_{min}} + \frac{c_{max}}{w_{min}}.$$

□

Finally, the following result can be proved using the same arguments as in the proof of Theorem 4.11.

Theorem 5.7. *Let J^*_{heur} be the value of the objective function* (1) *resulting from the execution of Algorithm 2. It is*

$$J^*_{gossip} \leq \left(\sqrt{2}\sqrt{\frac{k}{n}+1} + \frac{7}{4} + \sqrt{2}\right)\frac{d_{max}}{v_{min}} + \frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}} + D_{up}. \tag{45}$$

where $D_{up} = 2\dfrac{d_{max}}{v_{min}} + \dfrac{c_{max}}{w_{min}}$.

Proof: Follows the same steps of Theorem 4.11. □

6. Numerical simulations

In this section we present some numerical results comparing the performance of the proposed heuristic and the performance of the MILP gossip algorithm. We first analyze the value of J^*_{heur} and J^*_{gossip} for different values of k and n, comparing them with the lower and upper bounds, given in eq. (2) and eq. (5), respectively. We then compare the convergence time of the two decentralized solutions either in terms of number of iterations required or in terms of absolute time.

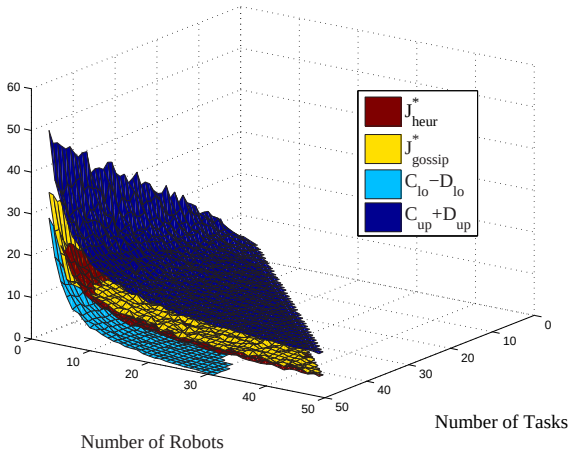

Figure 1: J^*_{heur}, J^*_{gossip} and the upper bound (2) and the lower bound (5) of the centralized solution.

In all the experiments robots and tasks are randomly scattered in a square box of side 5. Costs of tasks are integer values randomly generated with uniform distribution in the interval $[1, 5]$. Speeds v_i and w_i are real values randomly generated with uniform distribution in $[1, 2]$. In both decentralized algorithms the edge selection is performed in a uniformly random way. The MILP problems are solved using the MATLAB optimization tool *glpk*, while the results related with Algorithm 2 are obtained using our own MATLAB script. The value of the $ATSP$ is computed by calculating a minimum spanning tree and adding shortcuts in the induced cycle, thus approximating the optimal TSP length by a factor of $\alpha = 2$.

In Fig.1 are reported the results of the comparison between the following values:

- the value of J^*_{heur}, obtained by the execution of Algorithm 2;
- the value of J^*_{gossip} obtained by the execution of Algorithm 1;
- the upper and lower bound of the centralized approach given respectively by (2) and (5).

For each couple (n, k) of n robots and k tasks, J^*_{heur}, J^*_{gossip} and the two bounds are the mean values of 10 experiments. Simulations show that the maximum service time obtained with the two approaches lies always between the upper and the lower bound of the centralized approach. Moreover, the performance of the two approaches are similar. It can be observed that Algorithm 1 leads to better results than Algorithm 2 when the ratio $\frac{k}{n}$ is high.

In Fig. 2, Fig. 3 and Fig. 4 the execution times of Algorithm 2 are compared with the execution times of Algorithm 1. In particular, Fig. 2 and Fig. 3 show

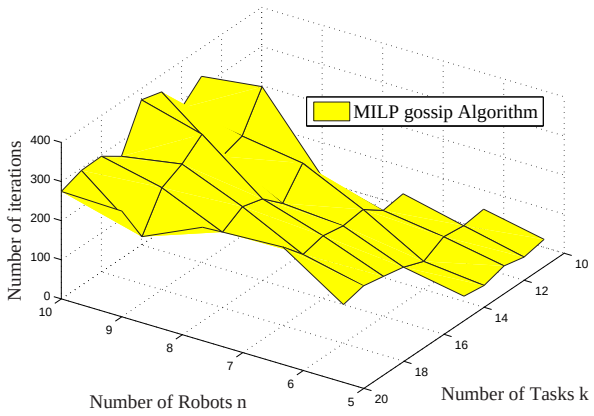

Figure 2: Number of iterations required to reach an equilibrium state with MILP gossip algorithm.

the execution time respectively of the MILP gossip algorithm and Algorithm 2 in terms of number of iterations, while in Fig. 4 the comparison is made in terms of time in seconds spent by MATLAB to execute the Algorithms. The two figures confirm that the proposed framework has a computational complexity considerably lower than the MILP gossip algorithm.

The results in Fig. 2 and Fig. 3 confirm also the conjecture that we have discussed in the final part of Section 5.1: the execution time in terms of number of iterations are of the same order in Algorithm 2 and in the MILP gossip algorithm.

Finally we focus on the execution time of Algorithm 2 in seconds and in terms of number of cycles. Figure 5 shows the number of iterations while Figure 6 shows the execution time in seconds for Algorithm 2 for different values of k in a system with $n = 10$ robots.

Figure 5 shows that the expected number of iterations of Algorithm 2 grows linearly with the number of tasks if the number of robots is kept constant. On the other hand, in Figure 6 is shown that the actual computational time is of the order of $\mathcal{O}(n^3)$ seconds. This is due to the fact that the complexity of the task exchange according to the heuristic grows linearly with the number of tasks for each iteration of Algorithm 2 thus accounting for at least a quadratic grow of computational time, the remaining difference can be accounted by the software implementation and execution in Matlab.

7. Conclusions and future work

In this paper we proposed upper and lower bounds for the cost of the optimal solution to the HMVRP which considers vehicles with different movement

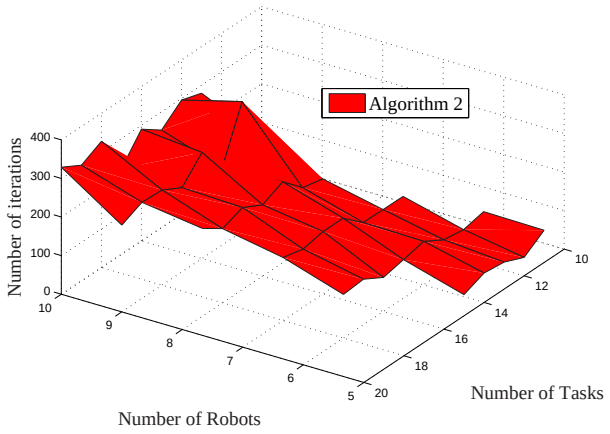

Figure 3: Number of iterations required to reach an equilibrium state with Algorithm 2.

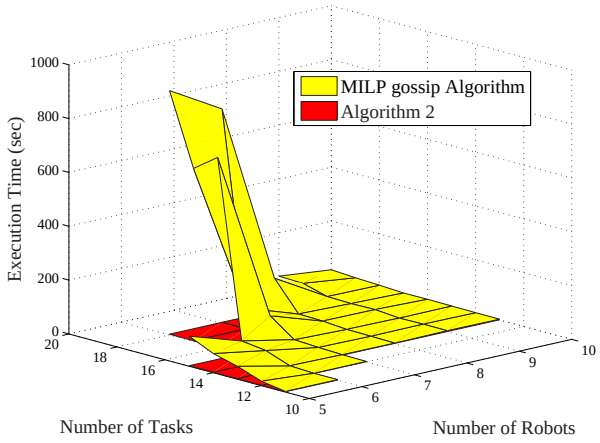

Figure 4: Execution time of MILP gossip algorithm and Algorithm 2.

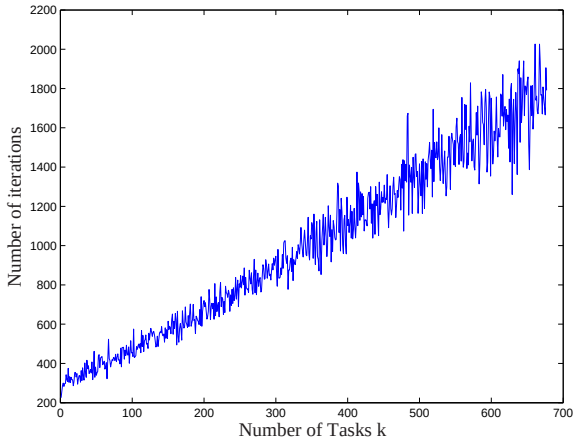

Figure 5: Number of iterations of Algorithm 2 for $n = 10$ and different values of k

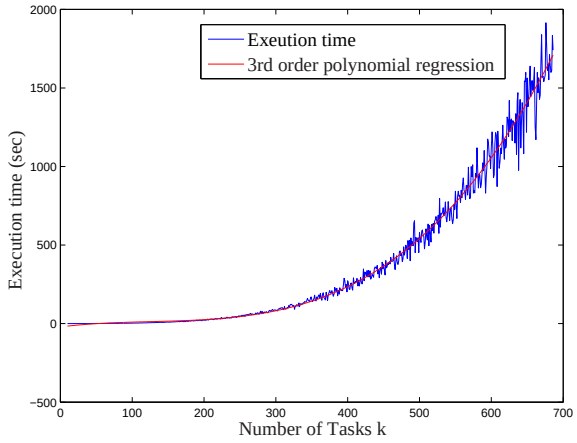

Figure 6: Execution time of Algorithm 2 for $n = 10$ and different values of k

and task execution speed and tasks with different servicing costs. We extended to our framework the bounds for the multi-vehicle routing problem in [7]. Furthermore, we proposed two algorithms based on gossip to solve the HMVRP in a distributed fashion exploiting only pairwise task exchanges between vehicles. The first algorithm is based on local, asynchronous and pairwise optimizations to improve the local task assignment. The second one is an heuristic with linear complexity with respect to the number of tasks. The computational complexity of the first method scales with exponential complexity with respect to the ratio between the number of tasks and vehicles, improving with respect to a centralized optimization that scales exponentially with the number of tasks. The proposed algorithms have been characterized in terms of finite-time almost sure convergence and in terms of minimum guaranteed performance.

We validated through simulations that the proposed algorithms compute a solution that scales with the number of robots within a constant factor of approximation with respect to the optimal centralized solution.

As future work we plan to extend the framework to a dynamic case in which robots start to move and serve tasks while the decentralized optimization is being executed and new tasks appear in the region.

References

[1] D. Applegate, W. Cook, S. Dash, and A. Rohe. *Solution of a min-max vehicle routing problem*. Forschungsinstitut für Diskrete Mathematik, Rheinische Friedrich-Wilhelms-Universität, 2001.

[2] E.M. Arkin, R. Hassin, and A. Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59(1):1–18, 2006.

[3] T. Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219, 2006.

[4] F. Bullo, R. Carli, and P. Frasca. Gossip coverage control for robotic networks: Dynamical systems on the space of partitions. *SIAM Journal on Control and Optimization*, July 2011. to appear.

[5] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, 99(9):1482–1504, 2011.

[6] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44 (3):671–684, 2008.

[7] J. Carlsson, D. Ge, A. Subramaniam, A. Wu, and Y. Ye. Solving min-max multi-depot vehicle routing problem. *Lectures on global optimization*, 55:31–46, 2009.

[8] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, DTIC Document, 1976.

[9] J.F. Cordeau, M. Gendreau, G. Laporte, J.Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, pages 512–522, 2002.

[10] G.B. Dantzig and J.H. Ramser. The truck dispatching problem. *Management science*, pages 80–91, 1959.

[11] L. Few. The shortest path and the shortest road through n points. *Mathematika*, 2(2):141–144, 1955.

[12] M.L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.

[13] M. Franceschelli, D. Rosa, C. Seatzu, and F. Bullo. A gossip algorithm for heteroeneous multi-vehicle routing problems. In *4th IFAC Conf. on Analysis and Design of Hybrid Systems*, Eindhoven, Netherlands, June 2012.

[14] M. Gendreau, A. Hertz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management science*, pages 1276–1290, 1994.

[15] G. Gutin and A. P. Punnen. *The Traveling Saleman Problem and Its Variations*, volume 12 of *Series in Combinatorial Optimization, Springer*. Kluwer Academic Publishers, 2002.

[16] H. J. Karloff. How long can a Euclidean traveling salesman tour be? *SIAM Journal on Discrete Mathematics*, 2(1):91–99, 1989.

[17] G. Laporte. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992.

[18] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[19] G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

[20] G. Laporte, M. Gendreau, J.Y. Potvin, and F. Semet. Classical and modern heuristics for the vehicle routing problem. *International transactions in operational research*, 7(4-5):285–300, 2000.

[21] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy-Kan, and D. B. Shmoys. *The Traveling Saleman Problem: A Guided Tour of Combinatorial Optimization*. Wiley Series in Discrete Mathematics and Optimization. John Wiley and Sons, 1985.

[22] R. Nallusamy, K. Duraiswamy, R. Dhanalaksmi, and P. Parthiban. Optimization of multiple vehicle routing problems using approximation algorithms. *Arxiv preprint arXiv:1001.4197*, 2010.

[23] C. Nilsson. Heuristics for the traveling salesman problem. *Department of Computer Science, Linkoping University*, 2003.

[24] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

[25] D. Rosa, M. Franceschelli, C. Seatzu, and F. Bullo. A gossip based heuristic algorithm for heteroeneous multi-vehicle routing problems. In *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Santa Barbara, California, USA, September 2012.

[26] P. Toth and D. Vigo. *The Vehicle Routing Problem*, volume 9 of *Monographs on Discrete Mathematics and Applications*. SIAM, 2002.