A Gossip Based Heuristic Algorithm for Heterogeneous Multi-Vehicle Routing Problems

Daniele Rosa * **Mauro Franceschelli** * **Carla Seatzu** *
Francesco Bullo **

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

Abstract: In this paper we propose a novel algorithm based on gossip to solve the Heterogeneous Multi-Vehicle Routing (HMVR) problem. A set of tasks is arbitrarily placed in a plane and a set of robots has to cooperate to minimize the maximum time required to visit and execute all tasks. Each task and each robot has different cost/speed. The proposed algorithm exploits only pairwise asynchronous communications between robots and attempts to balance the routing and execution cost of the tasks assignment of each robot through an heuristic. The proposed heuristic exploits polynomial time approximation algorithms to solve the Travelling Salesman Problem (TSP). Some characterization of the convergence properties of the algorithm are given together with extensive simulations to corroborate the results.

1. INTRODUCTION

The Multi-Vehicle Routing (MVR) problem has been addressed with several approaches that differ in the control architecture, the vehicle model and the number of limitations considered such as finite load capacity, service time windows, variable number of vehicles, placements of depots and several more, for instance in Toth and Vigo [2002], Laporte [1992b], Bektas [2006], Pisinger and Ropke [2007], Bullo et al. [2011].

Solutions to the MVR problem are often based on approaches that deal with variations of the Traveling Salesman Problem (TSP). Interesting surveys can be found in Lawler et al. [1985], Gutin and Punnen [2002], Laporte [1992a]. Extensions that deal with the multi-TSP problem can be found in Carlsson et al. [2009].

In this paper we consider a heterogeneous problem in which the total number of robots n is known a priori, each robot has different speed and task execution speed and the set of tasks $\mathcal{K}$ is arbitrarily distributed in a plane, each task with different size/cost. Furthermore we are interested in the development of a distributed algorithm that implements decentralized decision making, thus we consider the framework of *gossip* algorithms, such as described in Boyd et al. [2005], that consists in a set of local interaction rules that pair of the vehicles asynchronously execute to update their tasks assignment and thus reduce their maximum execution time.

In Franceschelli et al. [2011] the HMVR problem was addressed with an algorithm based on gossip in which, at each iteration, a pair of vehicles solves a mixed integer linear programming (MILP) problem that minimizes the maximum local execution time by optimizing the cost of both the vehicle route and the tasks assignment.

In this paper, the results in Franceschelli et al. [2011] are extended by proposing a different local interaction rule that provides similar performance and that does not require the solution of a MILP problem at each iteration, thus greatly reducing the computational complexity and the algorithm scalability with respect to the number of tasks.

It is shown that the performance of the proposed algorithm scales as $O(k/n)$. Furthermore we provide simulations showing that the achieved performance is within the upper and lower bounds to the optimal performance charcaterized in Franceschelli et al. [2011].

Summarizing, the paper is structured as follows.

- In Section 2 we formalize HMVR problem and introduce the adopted notation.
- In Section 3 the main contribution of this paper is presented, namely an algorithm based on gossip that improves the results in Franceschelli et al. [2011], and characterize some of its properties.
- In Section 4 extensive simulations are provided to characterize the performance of the proposed algorithm.
- In Section 5 concluding remarks are given.

⋆ 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.)

2. PROBLEM STATEMENT

Let us consider a set $\mathcal{N}$ of n mobile robots and a set $\mathcal{K}$ of k tasks scattered in a region $\mathcal{R}$. Each robot R_r is characterized by a speed v_r and a tasks execution speed w_r. Moreover each robot R_r starts from a depot D_r, the set of n depots is called $\mathcal{D}$. Each task i has cost c_i. The k tasks should be assigned to the n robots to be executed: robots are supposed to first coordinate themselves to improve their tasks assignment and then start to serve the tasks autonomously. The Heterogeneous Multi-Vehicle Routing (HMVR) problem, can be stated as follows: find an optimal assignment of the k tasks to the n robots in order to minimize the service time of all the tasks, which is a minimization of the following quantity

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

where

- $\mathcal{K}_r$ is the set of tasks assigned to 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.
- d_{max} is the maximum length of the shortest path between any two points in the region $\mathcal{R}$.

The optimization problem in eq. (1) can be solved using Mixed Integer Linear Programming (MILP). In Franceschelli et al. [2011] we have proposed the *MILP Gossip Algorithm*, it consists in a distributed strategy to solve the HMVR problem exploiting gossip based on asynchronous pairwise optimizations and the iterative solution of MILP problems to update the local tasks assignments of the robots. The MILP Gossip Algorithm leads to a suboptimal solution of the HMVR problem, with some advantages:

- the local interactions are much less complex than the centralized solution;
- the distributed algorithm is scalable with respect to the number of robots and allows parallelization.

In general, simulations show that the performance of the MILP gossip algorithm is within the bounds of the optimal centralized solution J^*:

$$J^* \leq C_{up} + D_{up} \quad (2)$$

where

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

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

The optimal centralized solution is also lower bounded as follows:

$$J^* \geq C_{lo} - D_{lo}, \quad (5)$$

where

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

Furthermore, in most of the cases, the solution obtained with gossip is very close to the centralized solution.

The main limit of the MILP gossip algorithm is that, if the number of tasks assigned to the robots involved in the local update is large, the computation of the optimal solution of the MILP problem requires high computational effort, and, in many real systems, agents do not have enough resources to solve such problems.

In this paper, we address the HMRV problem by proposing an alternative local update rule with respect to the MILP gossip algorithm which leads to similar performance with much less computational complexity.

3. A GOSSIP BASED HEURISTIC FOR THE HMVR PROBLEM

In this section we present the Decentralized Heuristic Algorithm, then we discuss its convergence properties and computational complexity in comparison with Franceschelli et al. [2011].

The robots update their states following Algorithm 1, while the task exchange rule is described in Algorithm 2. The basic idea is as follows. When two robots are selected at step 3.a of Algorithm 1, the two agents start to balance their execution time by the iterative execution of Algorithm 2. At each execution of Algorithm 2 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 TSP ($ATSP$), thus, unlike in the MILP gossip algorithm, this approach can be solved with polynomial time algorithms. There exist a vast literature on polynomial time algorithms to compute an $ATSP$ such that

$$ATSP \leq \alpha TSP$$

where TSP denotes the value of the optimal TSP and α represents the worst case ratio. In Nilsson [2003] 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 $\mathcal{O}(m^2 \log_2(m))$, where m denotes the number of nodes to be visited. To the best of our knowledge, the polynomial time heuristic based on MST which provides the best value of α is the Christofides algorithm described in Christofides [1976], which is characterized by a worst case ratio of $\alpha = 1.5$ and a running time of $\mathcal{O}(m^4)$.

Algorithm 1 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 2 repeatedly on R_p and R_r until no more tasks exchange is 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 the ATSP Algorithm.

Algorithm 2 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

$$J_{new} = \frac{ATSP(K_p \cup \{i\})}{v_p} + \frac{\sum_{j\in(K_p\cup\{i\})} c_j}{w_p}.$$

- **If** $J_{new} < J_r$
  (a) $\mathcal{K}_{ex} = \mathcal{K}_{ex} \cup \{i\}$.
  (b) $F = 1$.

End While.

- **STOP**: $\mathcal{K}_p(t+1) = \mathcal{K}_p(t) \cup \mathcal{K}_{ex}$ and $\mathcal{K}_r(t+1) = \mathcal{K}_r(t) \setminus \mathcal{K}_{ex}$.

3.1 Finite time and almost sure convergence

In this section we characterize the convergence properties of Algorithm 1. We prove that, there exist some conditions which allow Algorithm 1 to converge to an equilibrium state in finite time. Our proofs are based on the concepts of *deterministic persistence* and *stochastic persistence*. Furthermore we show that the analysis presented in Franceschelli et al. [2011] can be easily adapted to Algorithm 1.

The interactions among agents are modeled by an undirected graph $G = \{V, E\}$ where nodes represent robots, and the edge (r, q) belongs to E if the interaction among robots R_r and R_q is possible. We assume that all robots may interact with all the other robots, thus $E = V \times V$. We denote $e(t)$ as the edge selected randomly 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., it is

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

Definition *3.1.* (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 an interval.

Definition *3.2.* (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. ■

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.

For the following analysis we need to introduce the concept of *network state*. Given the set $\mathcal{K}_r$ 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. Now, 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*.

The following theorems describe some conditions on the gossip communication scheme which allows the robot to converge to a stable tasks assignment in a finite time. Proofs are omitted here because they follow the same lines of Theorems 4.6 and 4.7 respectively in Franceschelli et al. [2011].

Theorem *3.3.* 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 tasks 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 *3.4.* 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 tasks assignment, there exists a network state $\tilde{\mathcal{K}}^*_{heur}$ and almost surely a finite time $T > 0$ such that $\tilde{\mathcal{K}}(t) = \tilde{\mathcal{K}}^*_{heuristic}$ for all $t \geq T$, i.e., the network state converges almost surely in finite time to $\tilde{\mathcal{K}}^*_{heur}$.

3.2 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 computational complexity of the single task exchange rule described in Algorithm 2. The following proposition fix the running time of Algorithm 2.

Proposition *3.5.* Assume to compute the $ATSP$ using, at step 2 of Algorithm 2, the Christofides algorithm described in Christofides [1976]. The worst case running time of Algorithm 2 is $\mathcal{O}(k^5)$.

Proof: The maximum number of nodes assigned to a robot is k, thus at each iteration of the while loop of Algorithm 2 the running time of the Christofides algorithm is at maximum $\mathcal{O}(k^4)$. 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 2 is $k \cdot \mathcal{O}(k^4) = \mathcal{O}(k^5)$. □.

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

Proposition *3.6.* 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 1. 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 2. Let R_r and R_q be the couple of robots selected by Algorithm 1 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 2 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 3.5 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 tasks assignment. We conjecture that the expected number of iterations of Algorithm 1 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 2 leads to a different tasks assignment only if the maximum execution time among the involved robot can be decreased, otherwise the tasks assignment does not change. In the proposed framework if at time t the execution of Algorithm 2 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 3.6 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 1 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 which are presented in Section 4.

3.3 Some characterizations of the heuristic solution

In this section we focus on some properties of J^*_{heur}, i.e., the solution of Algorithm 1 at the equilibrium, when no better balancing among robots may be obtained. As the MILP gossip algorithm, Algorithm 1 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 1.

***Theorem** 3.7.* 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 1. It holds

$$|J^*_{r,heur} - J^*_{q,heur}| \leq K_{rq} = 2\frac{d^{rq}_{max}}{v^{rq}_{min}} + \frac{c^{rq}_{max}}{w^{rq}_{min}} \tag{7}$$

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 1 at time t with execution time respectively $J_r(t)$ and $J_q(t)$ after t iterations. By step 2 of Algorithm 1 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 tasks assignments is finite and at each iteration of Algorithm 2 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 3.7 and the fact that each robot interacts with any other sufficiently often, a significant result follows.

***Corollary** 3.8.* 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 1. It holds

$$|J^*_{r,heur} - J^*_{q,heur}| \leq K \tag{8}$$

where

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

■

Finally the following result can be proved using the same arguments as in the proof of Theorem 4.11 in Franceschelli et al. [2011].

Theorem 3.9. Let J^*_{heur} be the value of the objective function (1) resulting from the execution of Algorithm 1. It is

$$J^*_{gossip} \leq \frac{k}{n}\frac{2d_{max}}{v_{min}} + \frac{1}{n}\frac{\sum_{i\in\mathcal{K}} c_i}{w_{min}} + K, \quad (9)$$

where $K = 2\frac{d_{max}}{v_{min}} + \frac{c_{max}}{w_{min}}$. ■

4. NUMERICAL SIMULATIONS

In this section we present some numerical results which show a comparison between the performance of the proposed heuristic and the performance of the MILP gossip algorithm. We first analyse the value of J^*_{heur} for different values of k and n, comparing it with the lower and upper bounds, given in eq. (2) and eq. (5), of the centralized optimal solution and with the value of J^*_{gossip} obtained with the MILP gossip Algorithm.

Then we compare the convergence time of the two decentralized solutions either in terms of number of iterations required or in terms of absolute time.

In all the experiments robots and tasks are randomly scattered in a square whose edge is equal to 5 units. Costs of tasks are integer values uniformly randomly generated in the interval $[1, 5]$. Speeds v_i and w_i are real values uniformly randomly generated in $[1, 2]$. In both decentralized algorithms the edge selection is performed in a uniformly random way. The MILP problems are solved using the well known MATLAB optimization tool *glpk*, while the results related with Algorithm 1 are obtained using a MATLAB tool ad hoc developed to test our framework. The value of the $ATSP$ is computed using an approximated algorithm with worst case ratio $\alpha = 1.5$.

As a first result we propose an example which shows that, in general, Algorithm 1 leads to a sub-optimal solution of the HMVRP problem.

Example 4.1. Let us consider a system with $n = 2$ robots and $k = 4$ tasks. Robots are initially positioned at the same depot in the XY plane as summarized in Table 1. This table also summarizes the position and costs of tasks and the initial tasks assignment. Moreover, for each robot R_r it is $v_r = w_r = 1$. Table 2 presents the results of the load balancing carried out using both the centralized and the heuristic approach. As it can be seen, the optimal solution J^* of the centralized approach presented in Franceschelli et al. [2011] is better than the one obtained with Algorithm 1. In particular, it is $J^* = 23$ and $J^*_{heur} = 24$. ■

	X	Y	Init. Assig.	c_i
Robot 1	0	0	-	-
Robot 2	0	0	-	-
Task 1	1	0	Robot 1	10
Task 2	1	0	Robot 1	10
Task 3	1	0	Robot 2	11
Task 4	1	0	Robot 2	11

Table 1. Example 4.1: initial tasks assignment.

As a second result, 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 1;
- the value of J^*_{gossip} obtained by the execution of the MILP gossip algorithm;
- the upper and lower bound of the centralized approach given respectively by (2) and (5).

	Centralized $\mathcal{K}_r$	J^*_r	Heuristic $\mathcal{K}_r$	$J^*_{r,heur}$
Robot 1	$\{1, 2, 6\}$	23	$\{1, 3, 5, 6\}$	22
Robot 2	$\{3, 4\}$	23	$\{2, 4\}$	24

Table 2. Example 4.1: simulation results.

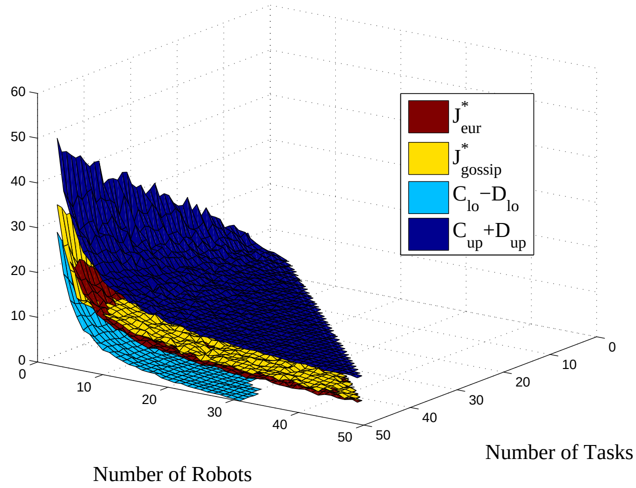

Fig. 1. J^*_{heur}, J^*_{gossip} and the upper bound (2) and the lower bound (5) of the centralized solution.

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. Simulation shows that the maximum service time obtained with the two gossip approaches lies always between the upper and the lower bound of the centralized approach. Moreover, the performance of the two approaches are similar.

In Fig. 2, Fig. 3 and Fig. 4 the execution times of Algorithm 1 are compared with the execution times of the MILP gossip algorithm. In particular, Fig. 2 and Fig. 3 show the execution time respectively of the MILP gossip algorithm and Algorithm 1 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 3.2: the execution time in terms of number of iterations are of the same order in Algorithm 1 and in the MILP gossip algorithm.

5. CONCLUSIONS

In this paper we have presented a novel heuristic based on gossip to solve the HMVR problem. The proposed heuristic leads to a sub-optimal solution but, differently than the previously presented solutions, is based on a polynomial time local interaction rule and is characterized by a fast execution time. We have compared the proposed heuristic

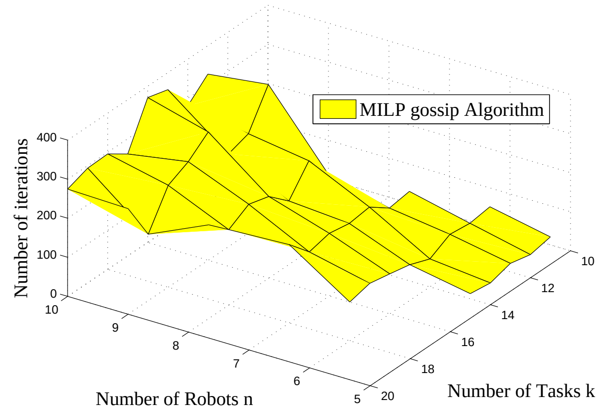

Fig. 2. Number of iterations required to reach an equilibrium state with MILP gossip algorithm.

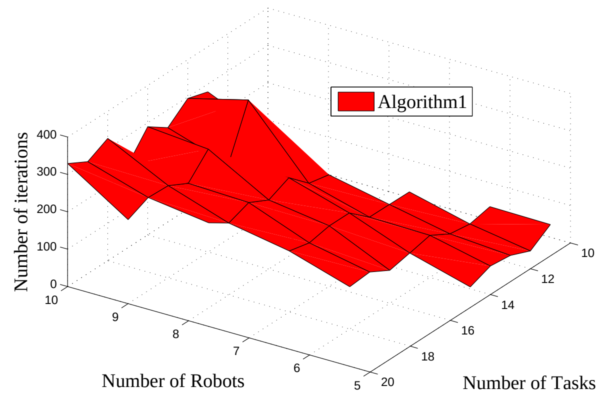

Fig. 3. Number of iterations required to reach an equilibrium state Algorithm 1.

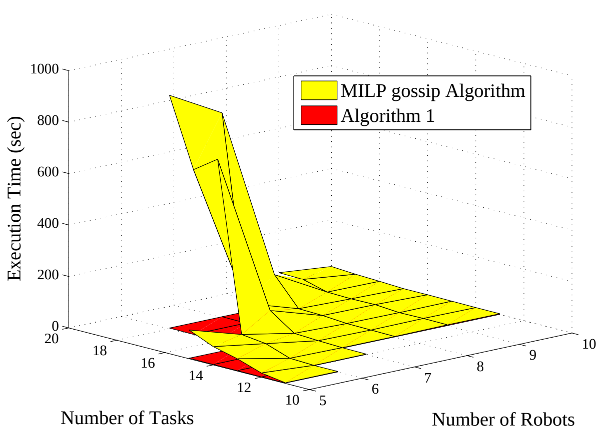

Fig. 4. Execution time of MILP gossip algorithm and Algorithm 1.

with the MILP gossip algorithm. The comparison of the performance of the two strategies has been made through exhaustive simulations which show similar performance and confirm that the proposed heuristic has a lower computational complexity than the MILP gossip algorithm.

REFERENCES

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

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2005). Gossip algorithms: design, analysis and applications. In *Proc. IEEE Infocom 2005*. Miami, USA.

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

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

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

Franceschelli, C., Seatzu, C., Rosa, D., and Bullo, F. (2011). A gossip algorithm for heterogeneous multi-vehicle routing problems. *Conference on Analysis and Design of Hybrid Systems (ADHS 12)*.

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

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

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

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

Nilsson, C. (2003). Heuristics for the traveling salesman problem. *Department of Computer Science, Linkoping University*.

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

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