A Subgradient Based Algorithm for Distributed Task Assignment for Heterogeneous Mobile Robots

Alessandro Settimi and Lucia Pallottino

Abstract— **In this paper the problem of assigning tasks to a set of mobile and heterogeneous robots based on their ability and their costs to accomplish a task is considered. Moreover, the dynamics of the robot and a private cost function to be optimized together with the assignment are also taken into account. To deal with a possibly high number of tasks and robots a distributed approach based on the subgradient method is used. A local dynamic optimal control and task assignment problem based on the information exchanged through a communication network are solved. With the proposed dynamic approach different types of kinematic vehicles with different motion constraints can be taken into account.**

I. INTRODUCTION

In an industrial environment, autonomous vehicles and more in general all robots can accomplish a large variety of tasks depending on their peculiarities. For example, an autonomous vehicle can carry materials across a warehouse, put materials on conveyor belts while a robotic arm may be able to accomplish tasks such as welding, screwing, assembling, etc.. In a scenario with different task typologies, it's natural to try to assign a task to the most appropriate robot.

At this point, after defining an optimality criterion, an optimal task for a robot can be found. If every robot has a specific optimal task, the problem resolve itself. But, if two or more robots have some conflicts on the task to execute this must be resolved to continue normal workflow. In such cases an assignment that will be in some sense optimal for the multi-robot system as a whole must be found.

Several algorithms that solve this problem with a centralized and a distributed approaches have been developed in the literature. The need of a distributed architecture occurs whenever there is a high number of robots and whenever we are interested in exploiting the benefits of the distributed systems (absence of a central unit, robustness to faults, scalability, heterogeneity, changes in the number of robots, etc.). Algorithms, in this case, must cope with issues that typically occur in the distributed systems such as the communication architecture, difficulty in proving the convergence of algorithms, etc..

The Distributed Task Assignment Problem is a well-known problem in Decision and Control Theory. Several works have been made over the years, involving many different approaches.

For example, the works supervised by *J. How* use various methods to achieve the assignment: consensus and auction approach, for which the *CBAA* and the *CBBA* algorithms are proposed ([1]), implicit coordination ([2]) and other approaches (e.g. [3], [4]). Alami et al. have proposed the *M+* algorithm and the *Contract-Net protocol* ([5], [6]). Other relevant examples of approaches are the distributed simplex ([7]), attraction fields ([8]), auction approach ([9], [10], [11], [12]), swarm approach ([13], [14]). Examples of the application to multi-robot systems can be found in [15], [16], [17]. In general, due to the distributed fashion of the problem, most of the algorithms do not ensure that an optimal solution is achieved.

The problem we want to solve in this paper is to obtain the optimal assignment for heterogenous robots based on the robots ability in accomplish tasks. Moreover, we also want to take into account the ability/cost of the robot to physically reach the task to accomplish. Indeed, in an industrial environment with various tasks and robots, it's reasonable to assume that a task is going to be performed by a robot (able to accomplish it) that is sufficiently close to it. This means that the concept of *neighborhood* between tasks and robots, and between robots and other robots as well, becomes important and must be taken into account. Moreover, each robot may have also a private cost function (other than the assignment cost) that it wants to optimize. For example a low battery robots may prefer to minimize energy consumption while a robot handling urgent material may prefer to minimize the time to accomplish a task. To solve this problem, a *Complete Assignment Problem* ([9]) is considered together with the vehicle dynamics and the private costs in a distributed approach. The proposed approach is based on the *dual decomposition method* of the *subgradient method* and, in those terms, it is similar to the work proposed in [18] and [9].

K. Hirayama et al. ([18]) present an algorithm for GAP (*Generalized Assignment Problem*) based on the dual decomposition method and the subgradient method. The problem addressed in [18] allows each robot to have more than one assigned task. In this case, the robots need to exchange additional information, with respect to the typical information needed by the subgradient method, to detect that convergence of the algorithm has been obtained. Finally, they don't provide the theoretical foundation of the dual variable update law that is different from the one we will determine.

The other main work related to this paper has been

Authors are with the Research Center "E. Piaggio", Dipartimento di Ingegneria dell'Informazione, University of Pisa, Italy `l.pallottino(at)centropiaggio.unipi.it`, `ale.settimi(at)gmail.com`

The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreements n.257462 HYCON2 Network of Excellence and n.2577649 PLANET.

proposed by *D. Bertsekas* in [9] where the auction algorithm has been introduced for the first time. A comparison with the subgradient method is made, but since the subgradient method requires the existence of a unique optimal solution, he proceed with a different approach in which, at each step, a dual variable component is updated instead of the entire vector.

As first contribution we provide the model of the considered problem as an optimization problem in which the optimization variables are the assignment values and the robot control inputs. In particular, the optimization takes into account both the assignment costs and the robots' private cost functions. To distribute the overall computation to single robots, the problem is thus decoupled into subproblems using the subgradient method. Moreover, we provide a *convergence condition* that can be checked, locally by each robot, without the need of sending further information other than the locally computed vector (a subgradient vector). Finally, the proposed algorithm is described and tested in simulation in different scenarios.

A. Problem Formulation

For reader convenience, and to introduce the notation used in the paper, in this section we briefly report the classical task assignment problem [19]. Consider an environment with n robots R_i and n tasks T_i, and the problem of optimally assigning a task to each robot where each task has an associated cost for each robot. We consider a *cost matrix* C and an *assignment matrix* A:

$$C \in \mathbb{R}^{n \times n} \quad A \in \{0;1\}^{n \times n} \tag{1}$$

where the element $C_{i,j}$ is the cost of task T_j for robot R_i while with $a_{i,j} = 1$ ($a_{i,j} = 0$) we denote that robot R_i is (is not) assigned to task T_j. The maximal complete assignment problem can be written as follows:

$$\begin{cases} \min\limits_{A} \sum\limits_{i,j} c_{ij} a_{ij} \\ \sum\limits_{j=1}^{n} a_{ij} = 1 \quad \forall i = 1, \dots, n \\ \sum\limits_{i=1}^{n} a_{ij} = 1 \quad \forall j = 1, \dots, n \\ A \in \{0;1\}^{n \times n}, \end{cases} \tag{2}$$

where $\sum\limits_{i,j} c_{ij} a_{ij}$ is the so called *loss function* that we want to minimize.

The n constraints $\sum\limits_{j=1}^{n} a_{ij} = 1$, $\forall i = 1, \dots, n$ can be written in matrix form as $A\underline{\mathbf{1}} = \underline{\mathbf{1}}$ where $\underline{\mathbf{1}} \in \mathbb{R}^n$ is the vector with element 1 in any component. Those constraints impose that no more than one task can be assigned to a single robot. Similarly, the n constraints $\sum\limits_{i=1}^{n} a_{ij} = 1$, $\forall j = 1, \dots, n$ can be written in matrix form as $\underline{\mathbf{1}}^T A = \underline{\mathbf{1}}^T$. Those constraints impose that each task is assigned to at most one robot. Concluding, the constraints of the stated task assignment problem imply that the assignment matrix A is *doubly stochastic*.

Consider the n^2 dimensional column vector $c^T = \left[c_1^T, \dots, c_n^T\right]$ where each component c_i of c is a row of matrix C, i.e. $c_i^T = [C_{i1}, \dots, C_{in}]$. Similarly, consider the n^2 dimensional column vector $x^T = \left[x_1^T, \dots, x_n^T\right]$ where each component x_i of x is a row of matrix A, i.e. $x_i^T = [A_{i1}, \dots, A_{in}]$. The task assignment problem (2) can be written in the vector form:

$$\begin{cases} \min\limits_{x} c^T x \\ \underline{\mathbf{1}}^T x_i = 1 \quad \forall i = 1, \dots, n \\ \sum\limits_{i=1}^{n} e_j^T x_i = 1 \quad \forall j = 1, \dots, n \\ x \in \{0;1\}^{n^2} \end{cases} \tag{3}$$

where the vector e_i represents the i-th vector of the canonical basis of $\mathbb{R}^n$.

The model falls in the *Binary Integer Programming* problems (BIP), a particular case of Integer Linear Programming (ILP).

II. The Optimal Control problem for task assignment

In the considered problem, tasks are assigned to mobile robots and thus the assignment costs will depend on the time needed by the robots to physically reach the task other than the ability of the robot to accomplish it. For this purpose we introduce a term $d \in \mathbb{R}^{n^2}$ to take into account, in the loss function, the costs to reach the goal. Thus the considered problem can be modeled as:

$$\begin{cases} \min\limits_{x} (c + d)^T x \\ \underline{\mathbf{1}}^T x_i = 1 \quad \forall i = 1, \dots, n \\ \sum\limits_{k=1}^{n} e_j^T x_k = 1 \quad \forall j = 1, \dots, n \\ x \in \{0;1\}^{n^2}. \end{cases} \tag{4}$$

The introduction of vector d gives us more flexibility in the characterization of the loss function that depends on the assignment robot-task. Hence, other than including motion costs, the presence of d allows us to include in the loss function also terms that depend only on tasks (e.g., priorities, deadlines, activation/deactivation) or only on robots (e.g. energy status). With this approach, we model the battery charge necessity as a typical industrial task. For example, $d \in \mathbb{R}^{n^2}$ may represent the distance vector between the robots and the tasks that is updated over time. Depending on the robot kinematics and motion constraints the time or path length to reach a task may depend also on the orientation. For example, in case of non-holonomic vehicles such as Dubins vehicles (with minimum turning radius), d may represent the minimum Dubins paths lengths for the agents toward the tasks (an example of allocating tasks for Dubins vehicles can be found in [20]), see Fig. 1. As another example, in case of differential drive vehicles, the time to turn on the spot may also be considered in the term d, see Fig. 2.

Hence, the introduction of term d leads to a more realistic assignment in real multi-robot mobile systems. Fig. 3 shows

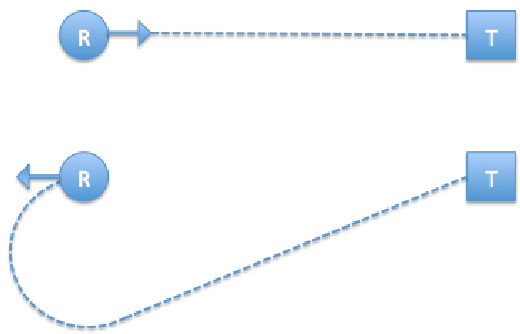

Fig. 1: Path traveled by a Dubins vehicle R toward a task T.

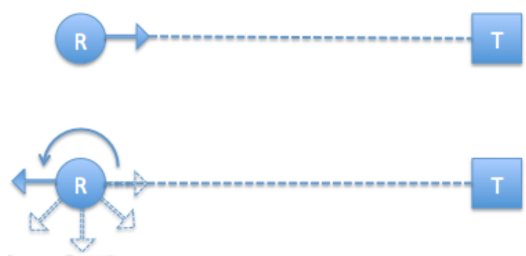

Fig. 2: Path traveled by a differential drive vehicle R toward a task T.

that, though the Dubins robot R is closer (for the Euclidean distance) to T_1 than to T_2, the best assignment is task T_2 that can be reached in less time or with a shorter path.

Using the technique shown in [21], we want to include the dynamic D_i of the robot in the assignment problem. Furthermore, we want to consider the possibility that each robot is interested in optimizing a private cost such as the energy consumption, (maximizing) the distance from other robots for safety purposes, (minimizing) the distance to points of interests to be monitored etc. For this purpose, let z_i be the robot state vector and u_i the control input. We denote with $\mathbf{z}_i = (z_i, u_i)$ and we consider a local cost function $J_i(\mathbf{z}_i)$ that depends only on robot i. Considering the vector $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$ and $\boldsymbol{D} = (D_1, \ldots, D_n)$, the overall

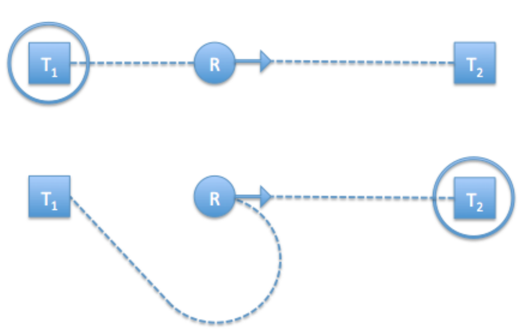

Fig. 3: Euclidean and Dubins Distances

model for the considered problem is thus

$$\begin{cases} \min_{x,\mathbf{z}} J(\mathbf{z}) + (c+d)^T x \\ \mathbf{z} \in \boldsymbol{D} \\ \underline{\mathbf{1}}^T x_i = 1 \quad \forall i = 1, \ldots, n \\ \sum_{k=1}^{n} e_j^T x_k = 1 \quad \forall j = 1, \ldots, n \qquad (*) \\ x \in \{0;1\}^{n^2} \end{cases} \tag{5}$$

Notice that, in (5), heterogeneous robots can be taken into accounts with different dynamics, different speed constraints etc.

III. A Subgradient Dual Method-based Algorithm

The constraint $(*)$ in (5) is the so-called *coupling constraint* of the problem, because it involves all the vectors x_k. Thus, a *dual decomposition method* ([22]) is applied to decouple the constraint and distribute the problem resolution. For the sake of clarity, we briefly report the theory of the subgradient dual methods (see e.g. [22]). A problem in the form:

$$\begin{cases} \min_{x_1, x_2} f_1(x_1) + f_2(x_2) \\ s.t.\ x_1 \in C_1,\ x_2 \in C_2 \\ \qquad h_1(x_1) + h_2(x_2) = 0, \end{cases} \tag{6}$$

has associated dual function

$$\begin{aligned} g(\lambda) &= \inf_{x_1 \in C_1, x_2 \in C_2} L(x_1, x_2, \lambda) \\ &= \inf_{x_1 \in C_1, x_2 \in C_2} (f_1(x_1) + f_2(x_2) + \lambda^T (h_1(x_1) + h_2(x_2))) \\ &= \inf_{x_1 \in C_1} (f_1(x_1) + \lambda^T h_1(x_1)) + \\ &\quad \inf_{x_2 \in C_2} (f_2(x_2) + \lambda^T h_2(x_2)). \end{aligned}$$

The dual problem can be distributed by fixing dual variable λ and defining two subproblems:

$$\begin{cases} \min_{x_1} f_1(x_1) + \lambda^T h_1(x_1) \\ s.t.\ x_1 \in C_1 \end{cases} \qquad \begin{cases} \min_{x_2} f_2(x_2) + \lambda^T h_2(x_2) \\ s.t.\ x_2 \in C_2 \end{cases}$$

with optimal values $g_1(\lambda)$, $g_2(\lambda)$. A first value of λ is assigned to the subproblems and it is updated as follows:

- compute a subgradient $h_i(\bar{x}_i) \in \partial_\lambda(-g_i)$, where $\bar{x}_i$ is the optimal solution of subproblem i and $\partial_\lambda(-g_i)$ is the subdifferential of $-g_i$[1] with respect to λ;
- compute the sum of all the subgradients $\sigma = h_1(\bar{x}_1) + h_2(\bar{x}_2) \in \partial_\lambda(-g)$
- the master algorithm updates λ with $\lambda + \alpha\sigma$ where $\alpha > 0$ is an update step parameter. The obtained value is used by the subproblems and the iteration continues until convergence.

In case of convex functions f_i and h_i, by choosing an appropriate update step, convergence can be ensured.

Regarding problem (5), the dual decomposition method is applied considering constraint (*) as the one to be included

[1]Notice that the dual problem of (6) consists in maximizing $g(\lambda)$ and hence minimizing $-g(\lambda)$.

in the Lagrangian function. Hence, let $\mu_T \in \mathbb{R}^n$ be the Lagrange multiplier vector associated with the coupling constraints (*) in (5), i.e.:

$$\mu_{T_j} \to \sum_{k=1}^{n} e_j^T x_k = 1 \qquad \forall j = 1, \ldots, n$$

Notice that, the other constraints in (5), including the dynamic equation constraint, depend on the robot's internal variables only (state, control and the robot's assignment to tasks) and hence they can be directly decoupled.

The dual problem associated to (5) is:

$$\begin{cases} \max\limits_{\mu_T} g(\mu_T) = \max\limits_{\mu_T} \inf\limits_{x,\ \mathbf{z} \in D} L(\mathbf{z}, x, \mu_T) \\ s.t.\ \underline{\mathbf{1}}^T x_i = 1 \quad \forall i = 1, \ldots, n \\ \qquad x \in [0;1]^{n^2} \end{cases} \tag{7}$$

where

$$\begin{aligned} &L(\mathbf{z}, x, \mu_T) = J(\mathbf{z}) + \tilde{L}(x, \mu_T), \\ &\tilde{L}(x, \mu_T) = (c+d)^T x + \sum_{j=1}^{n} \mu_{T_j} \left(\sum_{k=1}^{n} e_j^T x_k - 1 \right). \end{aligned}$$

As a consequence:

$$g(\mu_T) = \inf_{\mathbf{z} \in D} J(\mathbf{z}) + \inf_{x} \tilde{L}(x, \mu_T).$$

Since $J(\mathbf{z})$ does not depend on μ_T, the goal is now to minimize the function $-\tilde{g}(\mu_T) = -\inf\limits_{x} \tilde{L}(x, \mu_T)$ with respect to μ_T with constraints in (7). Furthermore, recall that $J(\mathbf{z})$ can be straightforwardly decoupled.

Notice that in (7), we have relaxed the boolean constraint $x \in \{0;1\}^{n^2}$ to apply the dual decomposition method. We made this choice to have a convex domain to use the subgradient method for convex functions that is known to converge to the optimal solutions under some conditions. On the other hand, this choice does not represent a problem since $\tilde{L}$ is affine in x and it attains its minimum on the domain's border, i.e. the optimal solution is in $x \in \{0;1\}^{n^2}$.

We can rewrite the Lagrangian $\tilde{L}$ as follows:

$$\begin{aligned} \tilde{L}(x, \mu_T) = &\left(c_1 + d_1 + \sum_{j=1}^{n} \mu_{T_j} e_j \right)^T x_1 + \\ &+ \cdots + \left(c_n + d_n + \sum_{j=1}^{n} \mu_{T_j} e_j \right)^T x_n - \sum_{j=1}^{n} \mu_{T_j} \\ &= \underbrace{(c_1 + d_1 + \mu_T)^T x_1 - \mu_{T_1}}_{\tilde{L}_1} + \\ &+ \cdots + \underbrace{(c_n + d_n + \mu_T)^T x_n - \mu_{T_n}}_{\tilde{L}_n} \\ &= \underbrace{(c_1 + d_1 + \mu_T)^T x_1 - \mu_T^T e_1}_{\tilde{L}_1} + \\ &+ \cdots + \underbrace{(c_n + d_n + \mu_T)^T x_n - \mu_T^T e_n}_{\tilde{L}_n}. \end{aligned} \tag{8}$$

In our case, the dual problem is subdivided into n subproblems of the form

$$\begin{cases} \inf\limits_{z_i} J_i(z_i) + \max\limits_{\mu_T} \tilde{g}_i(\mu_T) \\ s.t. \underline{\mathbf{1}}^T x_i = 1 \\ \qquad x_i \in [0;1]^n \end{cases} \tag{9}$$

where $\tilde{g}_i(\mu_T) = \inf\limits_{x_i} (c_i + d_i + \mu_T)^T x_i - \mu_T^T e_i$ with one of the possible subgradients

$$\sigma_i = x_i - e_i \in \partial_{\mu_T}(-\tilde{g}_i),$$

that must be send to other robots to compute the total subgradient σ:

$$\sigma = \sum_{i=1}^{n} \sigma_i. \tag{10}$$

After computing the total subgradient σ each robot use it to update the vector μ_T as described in the following algorithm.

A. The Algorithm

The Lagrangian function to be minimized by robot i is

$$L_i = J_i(\mathbf{z}_i) + [d_i + c_i + \mu_T]^T x_i - \mu_T^T e_i. \tag{11}$$

The proposed algorithm is

Algorithm 1

set $\mu_T = \underline{\mathbf{0}}$
repeat
  $x_i^* = \arg \min\limits_{x_i, \mathbf{z}_i \in D_i} L_i(z_i,\ x_i\, \mu_T)$
  s.t. $\underline{\mathbf{1}}^T x_i = 1,\ x_i \in [0;1]^n$
  send $\sigma_i = x_i^* - e_i$ to other robots
  receive the subgradients $\sigma_j = x_j^* - e_j$ from other robots
  compute total subgradient $\sigma = \sum\limits_{i=1}^{n} \sigma_i$
  update the Lagrange multiplier vector μ_T with:
  $\mu_T := \mu_T + \alpha_p \sigma$
until μ_T converge

where $\alpha_p > 0$ is the update parameter at step p to be chosen.

During the execution of the algorithm the total subgradient σ changes its value until it converges to the vector $\underline{\mathbf{0}}$. When this occurs the founded solution is feasible. Indeed, a vector $\sigma = \sum\limits_{i=1}^{n} \sigma_i = \sum\limits_{i=1}^{n} [x_i - e_i] = \underline{\mathbf{0}}$, from the definition of vectors x_i, is possible only if there is only one value *1* in each row and column of A and hence if the solution matrix A is doubly stochastic. This leads to have $\sum\limits_{i=1}^{n} x_i = \underline{\mathbf{1}}$ and due to the fact that $\sum\limits_{i=1}^{n} e_i = \underline{\mathbf{1}}$ from definition, we have $\sigma = \underline{\mathbf{0}}$. Concluding, the convergence to σ to $\underline{\mathbf{0}}$ implyies that a feasible and hence optimal solution has been found.

In order to have a distributed assignment, robot must communicate their subgradient to all the other robots through a communication network that may be not complete (i.e. we can not assume that any robot is always able to communicate

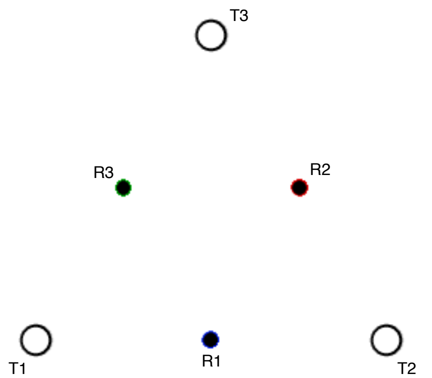

Fig. 4: Static example in symmetric configuration

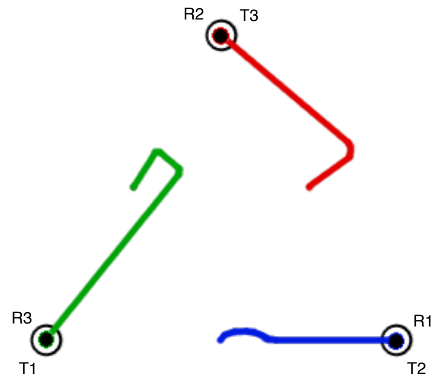

Fig. 5: Dynamic Example

with all others since communication is often limited in range). Several classical approaches can be used to tackle this problem. For example a piggybacking message can be used to communicate the subgradient while executing a classical distributed spanning three construction, see e.g.[23] under the assumption of a strongly connected network.

Notice that also a classical consensus approach can be used by robots to compute σ. Indeed, if the communication network is connected the consensus is known to converge toward the centroid of the initial values of the subgradient. Hence, the consensus will converge toward $\frac{1}{n}\sum_{i=1}^{n}\sigma_i = \frac{1}{n}\sigma$ and each agent can compute the desired value for the dual variable update by simply multiplying the obtained value by n.

IV. Simulations

Several simulations have been conducted to test the validity of the described approach. In the first set of simulations both static (constant d) or dynamic cases are considered. Assignment costs c and tasks positions and robots initial configurations have been randomly generated. A constant update value $\alpha_p = 0.1$ has been used. In this set of simulations homogeneous unicycle robots are considered while Euclidean distances and time spent by the robots to turn on the spot are taken into account. A complete communication graph has been used since we are interested only in the dual subgradient method. Results have been reported in I.

From simulations in static case there exists initial configurations that cause a particular symmetry in the problem (in particular in the costs c and d) leading to the proposed algorithm to switch between two or more possible assignment of equal cost, see Fig. 4. As expected, the simulation results in a continuous oscillation of the assignment. Notice that in the dynamic case this does not occur since motion breaks the initial symmetry. For example, the same symmetric scenario of Fig. 4 is tested in the dynamic case and the system evolution is reported in Fig. 5. The motion of the robots leads to an unbalanced cost vector and the optimal solution is found.

An example of assignment for 10 robots is shown in Fig. 6. For these simulations we have supposed that the task it's

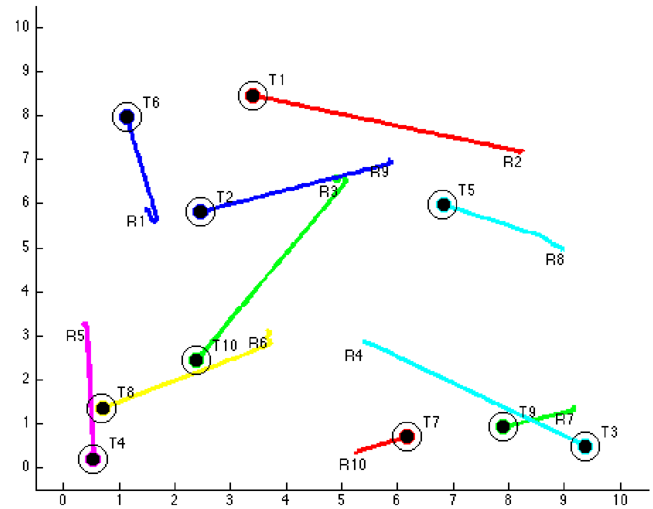

Fig. 6: Example with $n = 10$ robots that have to reach a point in the motion space

simply to achieve a certain point in the space. During the execution, robots go to the associated task which may change in time. At the end every robot has reached a different task and the obtained assignment is the optimal one. Notice that some robots change their direction when the assigned task changes during the algorithm execution.

Since robots send one message at each step of the algorithm, the total amount of messages is equal to the number of robots multiplied for the number of steps for convergence. We recall that the only information required for the proposed algorithm to converge toward the optimal solution is the robot computed subgradient. Hence a total of n^2 bits of payload are sent through the network at each step.

V. Conclusions and Future Works

In this paper, a novel algorithm for optimal control and distributed task assignment based on the subgradient dual method is presented for mobile robots. The proposed method allows to take into account robots with different characteristics and dynamics, more in general to consider

n	average convergence steps	total sent messages	total sent Kbytes
3	51	153	0,17
5	76	380	1,16
10	98	980	11,96

TABLE I: Simulation results

heterogeneous robots. In particular, eventually the optimal assignment depends also on the path that robot must follow to reach the task and not only its ability to accomplish it. Moreover, symmetric configuration that lead to assignment of equal optimal cost in the static case are solved when the problem is solved during robot motion. This makes the proposed approach more realistic and useful in mobile systems.

A formal robustness analysis of the proposed approach is part of our future work. However, for small errors, for example on the robots and tasks position, the algorithm still work correctly.

Several extensions of the proposed approach are part of on going research such as the case in which the number of tasks and robots is different and the private cost function depends on the assignment values.

REFERENCES

[1] L. Johnson, H.-L. Choi, S. Ponda, and J. How, "Allowing non-submodular score functions in distributed task allocation," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, Dec., pp. 4702–4708.

[2] M. Alighanbari and J. How, "Decentralized task assignment for unmanned aerial vehicles," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, Dec., pp. 5668–5673.

[3] M. Alighanbari and J. P. How, "Robust decentralized task assignment for cooperative uavs," in *AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado*, 2006, pp. 21–24.

[4] S. Ponda, L. Johnson, and J. How, "Distributed chance-constrained task allocation for autonomous multi-agent teams," in *American Control Conference (ACC), 2012*, June, pp. 4528–4533.

[5] S. Botelho and R. Alami, "A multi-robot cooperative task achievement system," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 3, 2000, pp. 2716–2721 vol.3.

[6] T. Lemaire, R. Alami, and S. Lacroix, "A distributed tasks allocation scheme in multi-uav context," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, 26-May 1,, pp. 3622–3627 Vol.4.

[7] M. Bürger, G. Notarstefano, F. Allgöwer, and F. Bullo, "A distributed simplex algorithm and the multi-agent assignment problem," in *American Control Conference (ACC), 2011*, 29 2011-July 1, pp. 2639–2644.

[8] D. Weyns, N. Boucké, and T. Holvoet, "Gradient field-based task assignment in an agv transportation system," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM, 2006, pp. 842–849.

[9] D. P. Bertsekas, "A distributed algorithm for the assignment problem," in *Laboratory for Information and Decision Sciences Working Paper*, 1979, pp. 1–19.

[10] B. Moore and K. Passino, "Distributed task assignment for mobile agents," *Automatic Control, IEEE Transactions on*, vol. 52, no. 4, pp. 749–753, April.

[11] N. Michael, M. Zavlanos, V. Kumar, and G. Pappas, "Distributed multi-robot task assignment and formation control," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, May, pp. 128–133.

[12] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason, "Auction protocols for decentralized scheduling," *Games and Economic Behavior*, vol. 35, no. 1, pp. 271–303, 2001.

[13] P.-Y. Yin, S.-S. Yu, P.-P. Wang, and Y.-T. Wang, "A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems," *Computer Standards & Interfaces*, vol. 28, no. 4, pp. 441–450, 2006.

[14] D. Zhang, G. Xie, J. Yu, and L. Wang, "An adaptive task assignment method for multiple mobile robots via swarm intelligence approach," in *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, June, pp. 415–420.

[15] E. Uchibe, T. Kato, M. Asada, and K. Hosoda, "Dynamic task assignment in a multiagent/multitask environment based on module conflict resolution," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4. IEEE, 2001, pp. 3987–3992.

[16] A. Farinelli, L. Iocchi, D. Nardi, and V. Ziparo, "Task assignment with dynamic perception and constrained tasks in a multi-robot system," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, April, pp. 1523–1528.

[17] L. Chaimowicz, M. F. M. Campos, and V. Kumar, "Dynamic role assignment for cooperative robots," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 1, 2002, pp. 293–298 vol.1.

[18] K. Hanada and K. Hirayama, "Distributed lagrangian relaxation protocol for the over-constrained generalized mutual assignment problem," *Agents in Principle, Agents in Practice*, pp. 174–186, 2011.

[19] P. M. Pardalos and L. S. Pitsoulis, *Nonlinear assignment problems: algorithms and applications*. Springer, 2000, vol. 7.

[20] S. Rathinam, R. Sengupta, and S. Darbha, "A resource allocation algorithm for multivehicle systems with nonholonomic constraints," *Automation Science and Engineering, IEEE Transactions on*, vol. 4, no. 1, pp. 98–104, 2007.

[21] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: Application to formation flight," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3. IEEE, 2004, pp. 2453–2459.

[22] S. Boyd and L. Vandenberghe, "Duality," in *Convex Optimization*, 1999, pp. 105–140.

[23] N. A. Lynch, *Distributed algorithms*. Morgan Kaufmann, 1996.