2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Consensus-based Robust Decentralized Task Assignment for Heterogeneous Robot Networks

Donato Di Paola, David Naso, and Biagio Turchiano

Abstract— **This paper considers the problem of decentralized task assignment in a network of heterogeneous robots. We introduce a new algorithm named heterogeneous robots consensus-based allocation (HRCA), which can be viewed as a possible extension of the recently proposed consensus-based bundle algorithm (CBBA) for homogeneous robot networks. The HRCA is based on a two stage decentralized procedure. In the first stage, similarly to CBBA, an initial assignment based on market-based decision strategies and local communication is determined, disregarding possible constraints on the maximum number of tasks assignable to each robot. Constraint violations are handled in the second stage, in which an iterative procedure is used by the robots to redistribute the tasks exceeding their individual capacity with minimal losses in terms of score function. Numerical simulations are used to evaluate the performance of the HRCA in a set of randomly generated scenarios, which include some examples of homogeneous networks to allow a comparison with CBBA.**

I. Introduction

In the last few years robotic networks (RNs) have drawn the attention of a large part of the robotics research community. A RN is commonly defined as a collection of robots which cooperate in order to achieve a global objective. The motivation behind the research on this topic is that a multi-robot approach offers several advantages over single robot approaches, such as parallel execution of tasks, robustness by adding redundancy and elimination of the single point of failure that is present in single robot systems [1]. A common assumption in robotics networks is that all the agents or vehicles are identical, i.e., in other words, that the network is homogeneous. However, recent trends in multi-agent systems highlight the advantages of networks composed by agents with different capabilities, which are generally referred to as heterogeneous robotics networks [2]. In principle, this type of networks offers several advantages, which include cost reduction, higher versatility, scalability and flexibility.

A great deal of research in RNs has focused on the area of software architectures, task planning algorithms, and control approaches [3]. In particular, a great effort has been done by the robotics research community to develop new solutions to these problems complying with the recent trends of decentralizing as much as possible information processing and decision activities among the various agents in the network. In this vast research area, a particular attention has been paid to the decentralization of the task planning and control. Several different directions have been investigated. In [4] a novel robust approach for decentralized task assignment for fleets of robots has been proposed. In the context of tasks and mission control, the decentralization of the sequencing has recently been investigated in [5], [6]. References [7], [8] address the integration of geometrical information about targets in the environment and local motion control algorithms to the distributed task assignment, while [9] investigates the problem of maintain certain connectivity properties while defining distributed motion control algorithms. Moreover, in the context of the coverage control, novel distributed solutions are proposed in [10], [11].

An alternative approach to the decentralized multi-assignment task assignment problem that is particularly relevant to our research is the consensus-based bundle algorithm (CBBA) developed in [4]. The CBBA is a decentralized task assignment method that greedily generates a vector of tasks using an auction-based approach and then resolves conflicts on the assignment by means of consensus algorithms and nearest-neighbor communications. Several evolutions of the CBBA have been recently proposed. Reference [12] explores a variant capable to address cooperation constraints. The authors propose a framework to embed the cooperation preferences in the scoring structure along with a decentralized method to eliminate invalid assignments. In [13] the CBBA is extended in order to address complex missions for a team of heterogeneous agents (in terms of scheduling capabilities) in a dynamic environment, proposing appropriate handling of time windows of validity for tasks and fuel costs of the vehicles.

Moving in this specific research line, this paper proposes a new variant, named heterogeneous robots consensus-based allocation (HRCA), to deal with decentralized task assignment in heterogeneous robot networks. This algorithm, inspired by CBBA, is based on a two stage decentralized task assignment strategy for heterogeneous RNs. In the first stage a consensus-based auction phase allocates all the tasks with score-based criteria and disregarding possible constraints on the maximum number of tasks assignable to each robot; in the second stage, an iterative procedure redistributes the tasks exceeding robot's capacity using a strategy based on the least penalty in terms of score function.

II. Problem Statement

Consider a network of heterogeneous robots, i.e. robots that cannot be considered identical due to their sensory or actuating hardware or functionalities. Suppose that each

D. Di Paola is with the Institute of Intelligent Systems for Automation (ISSIA), National Research Council (CNR), 70126 Bari, Italy dipaola@ba.issia.cnr.it

D. Naso and B. Turchiano are with the Department of Electrical and Electronic Engineering (DEE), Polytechnic of Bari, 70126 Bari, Italy naso@poliba.it, turchiano@poliba.it

978-1-4577-0079-8/11/$26.00 ©2011 AACC

robot is characterized by a set of skills (i.e., basic operations that it is able to execute), and that a list of tasks, each requiring one or multiple skills, has to be assigned to and performed by the available robots. This is a standard assignment problem often shortly referred to as *task assignment for heterogeneous robots*. This paper focuses on the case in which the assignment has to be determined with a distributed algorithm, as the result of local interactions between the robots without the intervention of any supervisory entity.

More formally, given N_r robots, with the index set $\mathcal{I} = \{1, \ldots, N_r\}$, N_t tasks, with the index set $\mathcal{J} = \{1, \ldots, N_t\}$, let $\mathbf{\Delta} \in \mathbb{B}^{N_r \times N_t}$ a nonzero matrix, called *Matrix of Possible Assignments* (MPA), that represents the ability for each robot to perform one or more tasks, i.e. if $\Delta_{ij} = 1$ task j can be performed by robot i and 0 otherwise, the goal of the task assignment for heterogeneous robots is to find a matching, a set of robot-task pairs (i, j), that fulfills capability constraints defined by $\mathbf{\Delta}$, $\forall i \in \mathcal{I}$ and $\forall j \in \mathcal{J}$, which maximizes some overall reward (or minimizes some global cost). The case in which only one task can be assigned to each robot is called *single-assignment* problem, while the more general case in which each robot can handle a sequence of maximum L_t task is referred to as *multi-assignment* problem. Since each robot is capable of executing at most one task at a time, in the case of multi-assignment, tasks will be executed in an ordered sequence. An assignment is said to be *free of conflicts* if each task is assigned to no more than one robot. Moreover, the assignment is said to be *complete* if all assignable tasks $\alpha \in \mathbb{N}$ have been assigned. Even finding the set of assignable tasks in a heterogeneous robot networks is nontrivial, as it depends on the actual variety and number of skills of the robots in the network. A way to address this problem is described in the following.

Similarly to the case of homogeneous networks already considered in related literature, we formulate the problem as an Integer Linear Program (ILP). More specifically, given N_r robots, each capable to handle a sequence of no more than L_t tasks, and given a set of N_t tasks that have to be executed by the robots, find the decision variables $x_{ij} \in \mathbb{B}$, where $\mathbb{B} = \{0, 1\}$, to obtain

$$Z = \max \sum_{i=1}^{N_r} \sum_{j=1}^{N_t} c_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{j=1}^{N_t} x_{ij} \le L_t \qquad \forall i \in \mathcal{I} \tag{2}$$

$$\sum_{i=1}^{N_r} x_{ij} \le 1 \qquad \forall j \in \mathcal{J} \tag{3}$$

$$\tilde{x}_{ij} \le \Delta_{ij} \qquad \forall (i, j) \in \mathcal{I} \times \mathcal{J} \tag{4}$$

where $x_{ij} = 1$ if task j is assigned to robot i and 0 otherwise.

It is assumed that the *score value* $c_{ij} \ge 0$ is a generic, nonnegative function of the assignment, quantifying its effectiveness or desirability. The determination of the global objective function Z is subject to a set of constraints, namely, the maximum number of task assignable to a robot is L_t ; a task can be assigned to no more than one robot; a task can be assigned only to robots that can execute it. These constraints are expressed by equations (2)-(4), respectively.

A key aspect in task assignment for heterogeneous robots is to guarantee that the number of tasks remaining unassigned at the end of the assignment stage is always minimized. To evaluate this aspect, however, it is necessary to determine the maximum number of assignable tasks α for the considered task and robot sets. This value can be determined as the solution of the following ILP problem.

***Problem** 2.1 (Maximum Number of Allocable Tasks):* Given N_r robots that have to execute N_t tasks, considering that each robot is allowed to accept and perform a sequence of no more than L_t tasks, and given the assignment constraints expressed by matrix $\mathbf{\Delta}$, find the decision variables $\tilde{x}_{ij}$ to obtain

$$\alpha = \max \sum_{i=1}^{N_r} \sum_{j=1}^{N_t} \tilde{x}_{ij} \tag{5}$$

subject to

$$\sum_{j=1}^{N_t} \tilde{x}_{ij} \le L_t \qquad \forall i \in \mathcal{I} \tag{6}$$

$$\sum_{i=1}^{N_r} \tilde{x}_{ij} \le 1 \qquad \forall j \in \mathcal{J} \tag{7}$$

$$\tilde{x}_{ij} \le \Delta_{ij} \qquad \forall (i, j) \in \mathcal{I} \times \mathcal{J} \tag{8}$$

The constraints (6)-(8) have a similar role of those just described in (2)-(4).

To solve this ILP problem in a straightforward way, these conditions can be placed into a classical matrix formulation as

$$\mathbf{A}\tilde{\mathbf{x}} \le \mathbf{d} \tag{9}$$

where $\mathbf{A}$ and $\mathbf{d}$ are a block matrix and a block vector, respectively, and the vector $\tilde{\mathbf{x}} \in \mathbb{B}^{N_r N_t \times 1}$ represents the decision variables. In order to reflect the constraints these objects are structured as follows

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_1} & \mathbf{A_2} & \mathbf{A_3} \end{bmatrix}^T \qquad \mathbf{d} = \begin{bmatrix} \mathbf{d_1} & \mathbf{d_2} & \mathbf{d_3} \end{bmatrix}^T$$

where the matrix $\mathbf{A_1} \in \mathbb{B}^{N_r \times N_r N_t}$, which represents the constraint in (6), is structured as a diagonal block matrix with the element on the diagonal is $\mathbf{1}^{1 \times N_t}$; $\mathbf{d_1} \in \mathbb{N}^{L_t \times 1}$ is a vector with all elements equal to L_t. The matrix $\mathbf{A_2} \in \mathbb{B}^{N_t \times N_r N_t}$ represents the constraint in (7) and is structured as

$$\mathbf{A_2} = \begin{bmatrix} \mathbf{I}_1 & \mathbf{I}_2 & \cdots & \mathbf{I}_{N_t} \end{bmatrix}$$

where $\mathbf{I}_i$, $\forall i \in \mathcal{I}$, is the identity matrix $\mathbf{I}^{N_t \times N_t}$; $\mathbf{d_2}$ is the vector $\mathbf{1}^{N_t \times 1}$. The matrix $\mathbf{A_3}$ is the identity matrix $\mathbf{I}^{N_r N_t \times N_r N_t}$, and the vector $\mathbf{d_3} \in \mathbb{B}^{N_r N_t \times 1}$ is structured as

$$\mathbf{d_3} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_{N_r} \end{bmatrix}^T$$

where $\mathbf{a}_i$ is the i-th row of MPA matrix.

Following this structure, the problem in (9) can be solved using well-known ILP algorithms such as the Branch and Bound technique. The sum of all elements of the solution vector $\overline{\mathbf{x}}$ is the number of all assignable tasks α, which can be used to determine how much a solution for the problem (1)-(4) is satisfactory in terms of amount of assigned or unassigned tasks.

In this work we develop a decentralized strategy, based on local communications among robots in the network, so as to provide a solution of the task assignment problem for heterogeneous RNs (1)-(4). In the following, we will use the value α as a performance metric for the proposed approach, i.e., we will grade the quality of an assignment not only in terms of global score, but also considering the number of unassigned tasks with respect to the theoretical maximum α.

III. Heterogeneous Robots Consensus-based Allocation

The HRCA is a two stage iterative algorithm inspired by the Consensus-Based Bundle Algorithm (CBBA) [4]. CBBA is a distributed strategy that makes use of an auction protocol and a consensus algorithm. The latter is used to converge to a good approximated solution for the multi-assignment task assignment problem over a network of homogeneous robots. More precisely, CBBA consists of iterations between two phases: a market-based phase where each robot in the network greedily generates an ordered vector of tasks, called a bundle, and a consensus phase where conflicting assignments are resolved through nearest-neighbor communication.

Using an enhanced allocation strategy, the HRCA exploits core features of the CBBA to provide a robust distributed task assignment for heterogeneous RNs. The HRCA consists of iterations between two nested stages. In the first one, that is the consensus-based phase, each robot fills its own bundle with all tasks that it can execute basing on the owned skills. When the convergence is reached, i.e. when all conflicts on the assignment are solved, the second stage is performed. In this stage bundles of overloaded robots, i.e. robots with a number of assigned tasks exceeding their limit L_t, are reduced through a task elimination phase based on the *least penalty* in terms of the bundle global score. This process finds the task that causes the least penalty evaluating the score produced by removing each task from the bundle and the associated second bid value obtained in the first stage.

After this check the whole algorithm is run again until final convergence, i.e. until a conflict free assignment that meets bundles capacity constraints is reached.

A. Stage 1: Consensus-based Bundle Construction

The first stage of HRCA consists of the filling of the bundle and the conflict resolution process. Similar to CBBA, this stage is divided into two phases.

1) Phase 1 (Bundle Filling): In this phase each robot creates its bundle $\mathbf{b}_i \in (\mathcal{J} \cup \{\emptyset\})^{N_t}$, adding freely all tasks it is able to perform, where $\emptyset$ indicates an empty task. In other words, at this stage the limit L_t on bundle size is disregarded. Each bundle of each robot has an associated path $\mathbf{p}_i \in (\mathcal{J} \cup \{\emptyset\})^{N_t}$. Tasks in the bundle are ordered based on which ones were added first in time, while in the path tasks are ordered based on their location in the assignment. Note that even if the size of $\mathbf{b}_i$ and $\mathbf{p}_i$ is equal to the number of tasks in the network N_t, in order to obtain a complete assignment the cardinality of bundle and path cannot be greater than the maximum assignment size L_t, and this condition will be checked in the second stage of the algorithm. Thus, in the following we will define *overloaded* each robot for which $|\mathbf{b}_i| > L_t$, where $|\cdot|$ denotes the cardinality of the vector.

The objective of this phase it to build the bundle and the path in order to obtain the maximum local reward for the robot i. Let $S_i^{\mathbf{p}_i}$ be the *score function* defined as the total reward value for robot i performing the tasks along the path $\mathbf{p}_i$. We assume two important constraints: the score function has to satisfy the *Diminishing Marginal Gain* (DMG) [4] condition, which states that the value of a task does not increase as other elements are added to the bundle before it, and the score of each task has to be increased by a constant factor ξ. The latter is needed for the comparison between different types of scores, as we will discuss in the Stage 2 .

A task is added in the bundle considering its *marginal score*. If a task j is added to the bundle $\mathbf{b}_i$, it incurs the marginal score improvement

$$c_{ij} = \begin{cases} \max_{n \leq |\mathbf{p}_i|} S_i^{\mathbf{p}_i \oplus_n \{j\}} - S_i^{\mathbf{p}_i}, & \forall j \in \mathcal{J} \backslash \mathbf{b}_i \\ 0, & \text{if } a_{ij} = 0 \end{cases} \tag{10}$$

where $\oplus_n$ denotes the operation that inserts the second vector right after the n-th element of the first vector. Note that the bundle $\mathbf{b}_i$ contains only tasks the robot i can execute: if the robot i cannot perform a task due to the $\mathbf{a}_i$ vector which represents the i-th row of the matrix $\mathbf{\Delta}$, this task is not included in the bundle. Also, note that if the task is already included in the bundle, then it does not provide any additional improvement in score.

In order to properly update information about the assignment in the path, three other vectors are defined: a winning bid vector $\mathbf{y}_i \in \mathbb{R}_+^{N_t}$, which stores the highest bids for each task; a winning robot vector $\mathbf{z}_i \in \mathcal{I}^{N_t}$, which stores the current bidders for values in $\mathbf{y}_i$; a first bid vector $\mathbf{u}_i \in \mathbb{R}_+^{N_t}$, which stores bids for all tasks in the bundle at the end of the first instance of this phase. Note that values in $\mathbf{u}_i$ represent the best bids on the best path of all executable tasks, and this value does not decrease during the Stage 1 of the algorithm. The bundle and path are recursively updated as

$$\mathbf{b}_i = \mathbf{b}_i \oplus_{|\mathbf{b}_i|} \{J_i\}, \qquad \mathbf{p}_i = \mathbf{p}_i \oplus_{n_{i,J_i}} \{J_i\} \tag{11}$$

with $J_i = \mathrm{argmax}_j (c_{ij} \times \mathbb{I}(c_{ij} > y_{ij}))$, where $\mathbb{I}(\cdot)$ is one when the argument is true and zero otherwise, and $n_{i,J_i} = \mathrm{argmax}_n S_i^{\mathbf{p}_i \oplus_n \{J_i\}}$. Moreover the other vectors are updates as follows: $y_{i,J_i} = c_{i,Ji}$, $z_{i,J_i} = i$, and $u_{i,J_i} = y_{i,J_i}$.

At the end of this phase each robot maintains information about all tasks it can bid over. The mechanism to obtain a conflict free assignment among the robots is described in the following phase.

2) Phase 2 (Conflict Resolution): In this phase tasks are released from the bundle $\mathbf{b}_i$, for each robot, if it receives a higher value from its neighbors for that task. To perform the release of the exceeding tasks, some fundamental information has to be shared among the robots. More precisely, six vectors are communicated for consensus. Beside the previously described vectors $\mathbf{y}_i$, $\mathbf{z}_i$, and $\mathbf{u}_i$, we introduce the second winning bid vector $\mathbf{v}_i \in \mathbb{R}_+^{N_t}$, which contains the second bid value, for each task; the second winning robot vector $\mathbf{w}_i \in \mathcal{I}^{N_t}$, which stores the current bidders for values contained in $\mathbf{v}_i$; and the time stamp vector $\mathbf{t}_i \in \mathbb{R}^{N_r}$, which contains the iteration number of the last information update from each of the other robots.

After the communication process, two set of vectors are updated. Vectors $\mathbf{y}_i$ and $\mathbf{z}_i$ are updated in order to decide which tasks have to be released, vectors $\mathbf{v}_i$ and $\mathbf{w}_i$ are updated to provide the correct information about the second higher bid and the associated bidder for the task elimination phase.

There are three possible actions, determined with the same rules of CBBA [4], that a robot i can take on task j:

1) *update:* $y_{ij} = y_{kj}, \quad z_{ij} = z_{kj}$
2) *reset:* $y_{ij} = 0, \quad z_{ij} = \emptyset$
3) *leave:* $y_{ij} = y_{ij}, \quad z_{ij} = z_{ij}$

In order to update vectors $\mathbf{v}_i$ and $\mathbf{w}_i$, if a bid exists for task j (i.e. $y_{ij} \neq 0$), four temporary variables are built as follows. Let $v^\star_{ij} = \max_k v_{kj}$ be the current second bid value $\forall j \in \mathcal{J}$, where k is a neighbor of robot i. Let $w^\star_{ij} = \mathrm{argmax}_k\, v_{kj}$ be the current second bidder $\forall j \in \mathcal{J}$. Let $v^{\star\star}_{ij} = 2\mathrm{max}_k\, u_{kj}$ be the global higher second bid value $\forall j \in \mathcal{J}$, based on each vector $\mathbf{u}_i$, where the operation $2\mathrm{max}$ return the second maximum value in a vector. If no second maximum value exists, i.e. all values in a vector are equal, the result of this function is zero. Let $w^{\star\star}_{ij} = 2\mathrm{argmax}_k\, u_{kj}$ be the global second bidder $\forall j \in \mathcal{J}$ associated to $v^{\star\star}_{ij}$. Before the update of $\mathbf{v}_i$ and $\mathbf{w}_i$ a preliminary check is performed: if $w^\star_{ij} = z_{ij}$ then $v^\star_{ij} = 0,\ w^\star_{ij} = 0$, and if $w^{\star\star}_{ij} = z_{ij}$ then $v^{\star\star}_{ij} = 0,\ w^{\star\star}_{ij} = 0$.

After the aforementioned definition of these temporary variables, vectors $\mathbf{v}_i$ and $\mathbf{w}_i$ are obtained using rules shown in the Table I for each $j \in \mathcal{J}$. Note that all information used for the update of vector $\mathbf{v}_i$ and $\mathbf{w}_i$ came from the current or the last value of the vector $\mathbf{u}_i$. Since this vector is related to the vector $\mathbf{y}_i$, that is updated at the end of this phase, all values $v^\star_{ij}$, $w^\star_{ij}$, $v^{\star\star}_{ij}$, and $w^{\star\star}_{ij}$ are synchronized accordingly.

If a bid is changed as an outcome of communication, each agent checks if any of the updated or reset task were in their bundle, and if so, these tasks, along with all of the tasks that we added to the bundle after them, are released as follows: $y_{i,b_{in}} = 0$, $z_{i,b_{in}} = \emptyset$, $v_{i,b_{in}} = 0$, and $w_{i,b_{in}} = \emptyset$, where b_{in} denotes the n-th entry of the bundle $\mathbf{b}_i$ that has to be removed.

At the end of this phase the consensus is checked on $\mathbf{z}_i$, $\mathbf{y}_i$, $\mathbf{v}_i$ and $\mathbf{w}_i$ vectors. If the consensus is not reached the algorithm returns to the first phase in order to add new tasks.

TABLE I

RULE TABLE FOR THE UPDATE OF VECTORS v_{ij} AND w_{ij}

$v^\star_{ij} = 0$	$v^{\star\star}_{ij} = 0$	$v_{ij} = 0,\ w_{ij} = 0$
	$v^{\star\star}_{ij} > 0$	$v_{ij} = v^{\star\star}_{ij},\ w_{ij} = w^{\star\star}_{ij}$
$v^\star_{ij} > 0$	$v^{\star\star}_{ij} = 0$	$v_{ij} = v^\star_{ij},\ w_{ij} = w^\star_{ij}$
	> 0	$v^\star_{ij} > v^{\star\star}_{ij} \longrightarrow v_{ij} = v^\star_{ij},\ w_{ij} = w^\star_{ij}$
		$v^\star_{ij} \leq v^{\star\star}_{ij} \longrightarrow v_{ij} = v^{\star\star}_{ij},\ w_{ij} = w^{\star\star}_{ij}$

B. Stage 2: Bundle Check

After the convergence of the Stage 1, a conflict-free assignment is produced. However, since limitations in the maximum bundle length have been neglected until now, a check for overloaded robots has to be performed, and if necessary the assignment has to be revised to redistribute the tasks exceeding the maximum allowed capacity. This operation is performed in the following phase.

1) Phase 3 (Bundle Resize): This phase starts for each robot having a number of tasks greater than the maximum size of the bundle L_t. Thus, if all robots are not overloaded, i.e. after the first stage results $|\mathbf{b}_i| \leq L_t$, this phase does not take place and the HRCA ends, confirming the solution achieved as the final assignment of Stage 1.

In this phase an iterative task elimination procedure is performed. The rationale behind this process is that, for the agent i, tasks that produce the *least penalty*, in terms of score, in $\mathbf{b}_i$ have to be removed until the size of the bundle is equal to L_t. The estimation process of the "penalty" produced by a task is based on both local and previously shared data. Local information is based on the marginal score that is produced by removing task j from the path $\mathbf{p}_i$ and the shared information is based on the second bid value, that is the maximum bid in the network on the task j after the current maximum bid y_{ij}. Thus the *least penalty* causing task is:

$$J_i^- = \arg\min_{n \leq |\mathbf{p}_i|} S_i^{\mathbf{p}_i} - S_i^{\mathbf{p}_i \ominus_n} - \gamma_j \tag{12}$$

where n is the position of task j in the path and γ_j is a parameter equal to v_{ij} if the second bidder is not an overloaded robot and 0 otherwise, that is $\gamma_j = \mathbb{I}(|\mathbf{b}_{w_{ij}}| \leq L_t) \times v_{ij} \quad \forall j \in \mathcal{J}$. Note that the v_{ij} value derives from a score determined as in the equation (10), i.e. it is the marginal score containing the basic score ξ. Using this assumption on the score scheme, if there is no second bidder for tasks j (this means that the robot i is the only executor for j in the network), the penalty produced removing the task j is much higher than the penalty caused removing a task $j^\star$ that has more than one bidder ($v_{ij^\star} \neq 0$).

Once the task associated to the least penalty J_i^- has been identified, the i-th robot removes it from the bundle and the path and sets to zero the value of a_{i,J_i^-}. This implies that the task J_i cannot be assigned to robot i anymore. Thus, as in the Phase 2 of the Stage 1, tasks in the bundle after the task

TABLE II

COMPARISON BETWEEN HRCA AND CBBA: PERCENTAGE IMPROVEMENT OF FINAL SCORE.

Number of tasks N_t	Best value	Mean value	Worst value
10	0.95%	0.07%	-1.09%
20	2.05%	0.72%	-0.40%
30	5.66%	1.74%	-1.88%
40	1.03%	0.78%	0.63%
50	0.63%	-0.98%	-2.66%

J_i^- are also released and the value of y_{i,J_i^-}, z_{i,J_i^-}, v_{i,J_i^-}, and w_{i,J_i^-} are reset.

At the end of this phase $\mathbf{y}_i$, $\mathbf{z}_i$, $\mathbf{v}_i$, and $\mathbf{w}_i$ variables, are shared among robots in order to update information due to the task elimination phase performed by all the overloaded robots. Then, the algorithm starts again from Stage 1, until the assignment is completed and there are no more overloaded robots in the network.

IV. NUMERICAL RESULTS

This section summarizes the results of a preliminary investigation of HRCA based on numerical simulations. To assess the overall effectiveness of the approach, this section initially considers a comparison with CBBA for the case of homogeneous RNs, and then analyzes the results in the case of randomly generated heterogeneous RNs.

A. Simulation Setup

In the simulations we consider a RN composed by N_r robots randomly placed in the environment $\mathcal{E} \triangleq [0, L] \times [0, L] \subset \mathbb{R}^2$, where $L = 2000$, that is $\mathcal{E}$ is a square field with side length L. Tasks are considered as target in the environment placed on a grid-like pattern. Each robot estimates the target position using a measuring model subject to a Gaussian zero-mean white noise with variance $\sigma_0^2 = 0.1L$. The score function, implemented on each robot, is the *Time-Discounted Reward*

$$S_i^{\mathbf{p}_i} = \sum_{j=1}^{N_t} \lambda_j^{\tau_i^j(\mathbf{p}_i)} + \xi \tag{13}$$

where $\lambda = 0.95s^{-1}$, and every agent moves at a speed of 40 m/s as in the original CBBA. Moreover, for each task a basic score value $\xi = 10$ is added in order to perform the score comparison shown in the Phase 3. The value of ξ has to be selected greater than the greatest difference between two generic marginal scores in the proposed scenario.

In order to compare the global score of the final assignment of HRCA $Z^\star_{\text{HRCA}}$, with the global score of other algorithms, the contribution of the basic score ξ has to be removed. Thus, the HRCA global score is calculated using the following expression:

$$Z^\star_{\text{HRCA}} = \sum_{j=1}^{N_t} \overline{\mathbf{y}}_j - (\xi\, N_a), \tag{14}$$

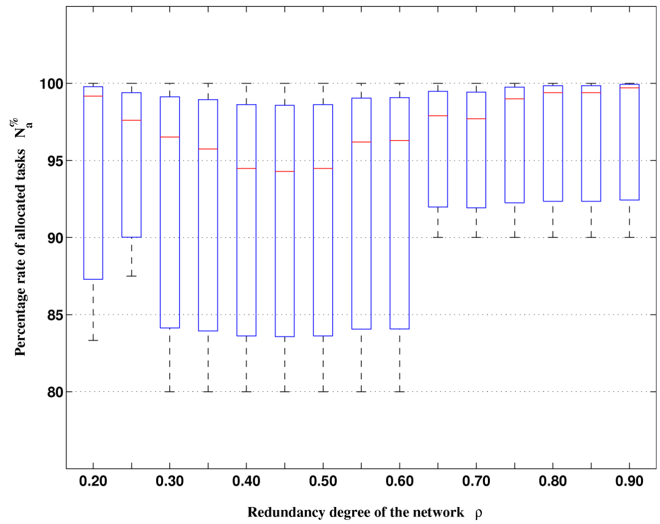

Fig. 1. Evaluation of the HRCA task assignment rate $N_a^{\%}$: variation of the number of tasks allocated by HRCA N_a, respect to α, as a function of the network redundancy degree ρ.

where $\overline{\mathbf{y}}_j$ is the common value of the score of task j after the consensus convergence and N_a is the number of task finally assigned by the HRCA algorithm.

B. Comparison with CBBA for Homogeneous Networks

The proposed HRCA algorithm inherits some interesting aspects of CBBA. In oder to evaluate these properties, the HRCA is compared with CBBA in the case of homogeneous RNs. In this case each robot has the vector $\mathbf{a}_i$ equal to $\mathbf{1}$.

To analyze the improvement of HRCA with respects to CBBA networks with $N_r = 5$ robots and $N_t = \{10, \cdots, 50\}$ are considered. The bundle limit varies as a function of the number of tasks and the robot $L_t = \lceil N_t/N_r \rceil$. The Table II summarizes the results, in term of percentage improvement of the final score, for a Monte Carlo simulation, with up to 50 trials for each task set, of the analyzed scenario. It can be noticed that the final score of HRCA is always comparable with the CBBA one. This means that HRCA maintains the good approximated solution of CBBA also in the case of homogeneous robots.

C. Performance Evaluation of HRCA

In order to evaluate the performance of the proposed HRCA algorithm over different heterogeneous networks, we define the *redundancy degree* of the network ρ. This parameter, which represents a measure of the heterogeneity of the network, is defined as:

$$\rho = \frac{1}{N_r N_t} \sum_{i=1}^{N_r} \sum_{j=1}^{N_t} a_{ij} \tag{15}$$

It is straightforward to note that ρ is related to the characteristics of the MPA matrix. Thus, in the case a network of homogeneous robots results $\rho = 1$.

As a global measure of performance, we consider the *percentage rate of allocated tasks* $N_a^{\%}$. This measure is the percentage ratio between the number of task finally assigned

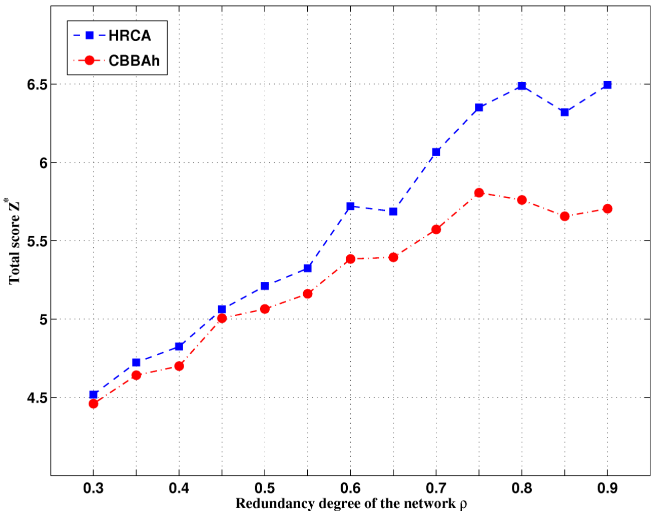

Fig. 2. Comparison between HRCA and CBBA$_h$: variation of total scores $Z^\star_{\text{HRCA}}$ and $Z^\star_{\text{CBBA}_h}$ respect to the network redundancy degree ρ.

by the HRCA N_a and the maximum number of allocable tasks α. We have studied a network with $N_r = 5$, $N_t = 10$, and $L_t = 2$, varying the redundancy degree ρ. The results of this simulation campaign are summarized in Fig. 1. We have considered the variation of the scenario from a network with very specialized robots ($\rho = 0.15$) to a network with versatile robots ($\rho = 0.90$), analyzing each value of ρ with a step of 0.05. For each simulation up to 100 random tasks and robots configurations have been considered. In Fig. 1, for each simulation, the mean value (red line), the 25 and 75 percentile (blue box), and the minimum and maximum values (black dashed lines) are shown. Is straightforward to note that the highest value of $N_a^{\%}$, about the 98%, is obtained for both the cases of very specialized (small number of alternative assignments) and very versatile (large number of alternatives) robots. In the intermediate cases ($\rho = \{0.30, \ldots, 0.60\}$) HRCA allocates anyway a good amount of tasks, with mean values of $N_a^{\%}$ of about 95% and worst case values of about 80%.

For a validation on the effectiveness of the strategy proposed in HRCA, the CBBA$_h$ (a straightforward adaptation of the CBBA to the case of heterogeneous networks) is considered. We implement two simple modifications of the CBBA algorithm. First, during the bundle construction phase, as in HRCA, a task j is added to the bundle $\mathbf{b}_i$ if the robot i can execute it, i.e. if $a_{ij} = 1$. Second, the consensus check is performed without taking into account the number of assigned tasks (for cases in which not all tasks can be assigned). Also in this case, we have performed different simulations of heterogeneous RN scenarios. A scenario with $N_r = 5$, $N_t = 20$, and $L_t = 4$ is considered. For this comparison ρ varies from 0.30 to 0.90. In Fig. 2 total scores $Z^\star_{\text{HRCA}}$ and $Z^\star_{\text{CBBA}_h}$ of the assignment for each scenario are shown. The scores of HRCA and CBBA$_h$ are compared for the identical experiment, without any variation on the initial positions of robots and tasks locations. It can be noticed that the final score of HRCA is always better than the CBBA$_h$ one; this means that the proposed task assignment strategy for heterogeneous networks is advantageous with respect to straightforward extensions of the CBBA. Moreover, the benefits of the HRCA algorithm become more significant as the heterogeneity of the robot increases, due to the specific operations performed during Stage 2.

V. Conclusion

In this paper a novel decentralized task assignment algorithm for heterogeneous robotic networks has been presented. The method is based on a rigorous formulation of the task assignment problem taking into account the different capabilities of robots in the network. The proposed method extends the robust decentralization structure of the consensus-based bundle algorithm in order to provide a conflict free multi-assignment solution in presence of constrained task to robot association. Current works are focused on the development of mechanism to guarantee that the maximum amount of task is always assigned to the available robots, and on the theoretical analysis of convergence and complexity of the proposed algorithms.

References

[1] C. V. Jones and M. J. Mataric, *Behavior-Based Coordination in Multi-Robot Systems. Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Applications*, S. S. Ge and F. L. Lewis, Eds. Marcel Dekker, Inc., 2005.

[2] L. E. Parker, “Multiple mobile robot systems,” in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, ch. 40, pp. 921 – 941.

[3] T. Arai, E. Pagello, and L. E. Parker, “Guest editorial: Advances in multi-robot systems,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655 – 661, 2002.

[4] H.-L. Choi, L. Brunet, and J. P. How, “Consensus-based decentralized auctions for robust task allocation,” *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912 – 926, 2009.

[5] N. Palomeras, P. Ridao, C. Silvestre, and A. El-Fakdi, “Multiple vehicles mission coordination using petri nets,” in *Proceedings of IEEE International Conference on Robotics and Automation*, 2010, pp. 3531 – 3536.

[6] D. Di Paola, A. Gasparri, D. Naso, G. Ulivi, and F. L. Lewis, “Decentralized task sequencing and multiple missions control for heterogeneous robotic network,” in *Proceedings of IEEE International Conference on Robotics and Automation*, 2011.

[7] S. L. Smith and F. Bullo, “Monotonic target assignment for robotic networks,” *IEEE Transactions on Automatic Control*, vol. 54, no. 9, pp. 2042 – 2057, 2009.

[8] M. Zavlanos and G. J. Pappas, “Dyamic assignment in distributed motion planning with local coordination,” *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 232 – 242, 2008.

[9] ——, “Distributed connectivity control of mobile networks,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1416 – 1428, 2008.

[10] P. Frasca, R. Carli, and F. Bullo, “Multiagent coverage algorithms with gossip communication: Control systems on the space of partitions,” in *Proceedings of American Control Conference*, St. Louis, MO, USA, Jun. 2009, pp. 2228–2235.

[11] J. Cortes, “Coverage optimization and spatial load balancing by robotic sensor networks,” *Automatic Control, IEEE Transactions on*, vol. 55, no. 3, pp. 749 –754, 2010.

[12] H.-L. Choi, A. Whitten, and J. P. How, “Decentralized task allocation for heterogeneous teams with cooperation constraints,” in *Proceedings of American Control Conference*, 2010.

[13] S. Ponda, J. Redding, H.-L. Choi, J. P. How, M. Vavrina, and J. Vian, “Decentralized planning for complex missions with dynamic communication constraints,” in *Proceedings of American Control Conference*, 2010.