Optimal Task Allocation in Heterogeneous Multi-Robot Systems Using a Mixed Centralized/Decentralized Strategy

Siddharth Mayya[1], Gennaro Notomista[2] and Magnus Egerstedt[1]

***Abstract*— We present a mixed centralized/decentralized strategy to allocate tasks to a team of robots with heterogeneous capabilities. In the decentralized part of the algorithm, where the robots only have access to local information, individual robots choose between tasks based on the energy consumed in performing each task. This is done by encoding task as constraints in an energy minimization problem being solved by the robots at each point in time and using slack variables in the constraints to achieve relative prioritization among the tasks to be executed. Furthermore, global specifications on the required allocation of tasks among the robots are enforced by placing relative constraints among the slack variables themselves. In this paper, we demonstrate how a central computer can enforce global specifications on the task allocation by performing a centralized computation and intermittently disseminating information to all the robots. Real robot experiments demonstrate the efficacy of the proposed task allocation framework.**

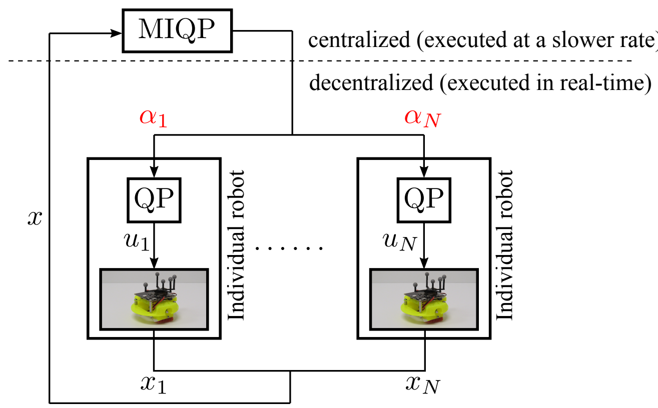


Fig. 1. A mixed centralized/decentralized architecture for task allocation in a team of robots with heterogeneous capabilities.

I. INTRODUCTION

Multi-robot task allocation is an active area of research with various solutions tailored for specific application scenarios [1], [2], [3]. The problem becomes especially involved when individual robots differ in their capabilities to perform different tasks [4] and when robots are expected to operate in unknown and dynamic environments for long periods of time, e.g., [5].

In [6], we presented a task allocation framework which exhibits the following salient features: (i) constraint-based task execution which explicitly accounts for the energy required by the robots in executing the tasks (ii) task allocation that accounts for the heterogeneous capabilities of the robots (iii) a mechanism to incorporate global specifications in the task allocation as well the energy considerations of the robots.

We allow individual robots execute all tasks at the same time but with different levels of effectiveness—which is encoded via slack variables in a constrained optimization problem. In this formulation, the execution of M tasks by a given robot i is encoded in the following manner:

$$\min_{u_i} \|u_i\|^2 \quad \text{s.t.} \quad c_{task_m}(x_i,u_i) \geq 0, \forall m \in \{1,\dots,M\},$$

where u_i is the control effort expended by robot i, x_i is its state, and c_{task_m} denotes a constraint function which ensures the execution of task m. This *constraint-based* formulation is better suited for long-term autonomy applications than cost-based formulations [5]. The feasibility of this task execution framework is ensured by the introduction of slack variables corresponding to each constraint:

$$\begin{aligned}\min_{u_i,\delta_i}\ & \|u_i\|^2 + \|\delta_i\|^2\\ \text{s.t.}\ & c_{task_m}(x_i,u_i) \geq -\delta_{i,m}, \forall m \in \{1,\dots,M\},\end{aligned}$$

where $\delta_i = [\delta_{i,1}, \delta_{i,2}, \dots, \delta_{i,M}]^T$ is the set of slack variables corresponding to robot i's execution of the M tasks. Then, for robot i, a task prioritization can be introduced by adding constraints on the slack variables pertaining to each task:

$$\begin{aligned}\min_{u_i,\delta_i}\ & \|u_i\|^2 + \|\delta_i\|^2\\ \text{s.t.}\ & c_{task_i}(x_i,u_i) \geq -\delta_i\\ & K_i\delta_i \geq 0, \quad \forall i \in \{1,\dots,M\}.\end{aligned} \tag{1}$$

In [6], we formulated a mixed-integer quadratic program (MIQP) to compute the task priorities K_i, slack variables δ_i, and control inputs u_i for the individual robots. In this formulation, the computation of the task priorities requires knowledge of the current task allocation of the swarm, thus not allowing for a decentralized implementation of the framework. Furthermore, the computational burden of solving an MIQP prevents it from being solved at each point in time, due to which the problem was relaxed to a quadratic program (QP) in [6].

In this paper, we propose a mixed centralized/decentralized [7] task allocation framework that addresses both of the above mentioned problems. We envision a central computer—which has access to the states

This work was sponsored by ARL DCIST CRA W911NF-17-2-0181.

[1]S. Mayya and M. Egerstedt are with the School of Electrical and Computer Engineering, Institute of Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, 30308 {siddharth.mayya,magnus}@gatech.edu

[2]G. Notomista is with the School of Mechanical Engineering, Institute of Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, 30308 g.notomista@gatech.edu

of the robots—that computes an *unrelaxed* solution to the MIQP. These solutions are computed at regular intervals of time and the resulting task priorities K_i are transmitted to the individual robots. The robots utilize the most recently received K_i matrices to solve the QP given by (1) at a real-time frequency. Thus, the computational burden of solving an unrelaxed MIQP is transferred to the cloud, while the robots optimize for their own energy consumption in a decentralized manner. Real robot experiments demonstrate the efficacy of the developed task allocation architecture.

II. TECHNICAL BACKGROUND

In this section, we present some technical results which will be used in this paper. Let $h : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function, and define the *safe set* S as its zero-superlevel set:

$$S = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}. \tag{2}$$

Let $\partial S = \{x \in \mathbb{R}^n \mid h(x) = 0\}$ and $S^\circ = \{x \in \mathbb{R}^n \mid h(x) > 0\}$ denote the boundary and the interior of S, respectively. The function h is called a *(zeroing) control barrier function (ZCBF)* if the following condition is satisfied:

$$\sup_{u \in U} \{L_f h(x) + L_g h(x) u + \gamma(h(x))\} \geq 0 \quad \forall x \in \mathbb{R}^n, \tag{3}$$

where γ is a Lipschitz continuous extended class $\mathcal{K}$ function [8], and $L_f h(x)$ and $L_g h(x)$ denote the Lie derivatives of h in the directions of f and g, respectively. The following theorem summarizes two important properties of ZCBFs.

Theorem 1. *Consider a dynamical system in control affine form $\dot{x} = f(x) + g(x)u$, where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the state and the input, respectively, f and g are locally Lipschitz. Let $S \subset \mathbb{R}^n$ be a set defined by a continuously differentiable function h as in* (2)*. Then, any Lipschitz continuous controller u such that* (3) *holds renders the set S forward invariant and asymptotically stable, i. e.,:*

$$x(0) \in S \Rightarrow x(t) \in S \ \forall t \geq 0$$
$$x(0) \notin S \Rightarrow x(t) \to \in S \text{ as } t \to \infty,$$

where $x(0)$ denotes the state x at time $t = 0$.

Proof. See [8] and [9]. □

As our primary objective is the allocation of tasks among different robots, we abstract the motion of the robots using single integrator dynamics, assuming that we can manipulate their velocities directly. We are interested in synthesizing a control signal $u(t)$ that allows the minimization of the cost $J(x(t))$. This can be achieved by solving, at each point in time, the minimization problem

$$\min_u J(x), \tag{4}$$

where x and u are coupled through the single integrator dynamics $\dot{x} = u$.

As explained in [5], the constraint-driven control strategy has advantages in terms of robustness against unpredictable and changing environmental conditions—properties which are useful when considering long-duration autonomy. In [9], we show that solving (4) in order to synthesize $u(t)$ is equivalent to solving the following constraint-based optimization problem, in the sense that they both achieve the goal of minimizing the cost J:

$$\begin{aligned} \min_{u,\delta} \quad & \|u\|^2 + |\delta|^2 \\ \text{s.t.} \quad & \frac{\partial h}{\partial x} u \geq -\gamma(h(x)) - \delta \end{aligned} \tag{5}$$

where $\delta \in \mathbb{R}$ is the slack variable signifying the extent to which the task constraint can be violated, γ is an extended class $\mathcal{K}$ function, and $h(x) = -J(x)$ is a *(zeroing) control barrier function*. The zero-superlevel set of h is $S = \{x \mid h(x) \geq 0\} = \{x \mid J(x) \leq 0\} = \{x \mid J(x) = 0\}$, where the last equality holds because the cost $J(x)$ is a non-negative function. In the particular case in which J is strictly convex and $J(0) = 0$, we have that $\frac{\partial J}{\partial x}(x) \neq 0 \, \forall x \neq 0$. Then, Theorem 1 directly implies that $x \to \in S$, i. e., $J(x(t)) \to 0$, as $t \to \infty$. For a proof of the general case, we refer to [9].

In the next section, we introduce the heterogeneous task allocation framework developed in [6], and motivate the need for a centralized/decentralized architecture.

III. PREVIOUS WORK

Consider a team of N robots executing M different tasks $T_1, \ldots, T_M$. Each task T_m is encoded as the minimization of a cost function $J_m, m \in \mathcal{M} = \{1, \ldots, M\}$. We assume that the costs J_m have the following structure:

$$J_m(x) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} J_{i,m}(\|x_i - x_j\|),$$

i.e. they can be broken down into the sum of pairwise costs among each robot i and its neighboring robots $j \in \mathcal{N}_i$. In [10], the authors illustrate the various multi-robot tasks represented by such a cost, and demonstrate the decentralized nature of the control law obtained by performing a gradient-descent on this cost. Furthermore, in [9], we showed that in this case, the constraint-based task execution can be rendered decentralized provided that certain conditions on the extended class $\mathcal{K}$ function γ are satisfied. We also assume single integrator robot dynamics $\dot{x}_i = u_i, \ i \in \mathcal{N} = \{1, \ldots, N\}$.

As described by the formulation in (5), the execution of M tasks by each robot can be encoded using the following quadratic program:

$$\begin{aligned} \min_{u_i,\delta_i} \quad & \|u_i\|^2 + \|\delta_i\|^2 \\ \text{s.t.} \quad & \frac{\partial h_{i,m}}{\partial x_i} u_i \geq -\gamma(h_{i,m}(x)) - \delta_{i,m}, \ \forall m \in \{1, \ldots, M\} \\ & \|\delta_i\|_\infty \leq \delta_{max} \end{aligned}$$

where $h_{i,m} = -J_{i,m}$ for each $i \in \mathcal{N}$ and each $m \in \{1, \ldots, M\}$. Now, let $\pi^* = [\pi_1^*, \ldots, \pi_M^*]^T$ denote the desired task allocation for the entire team where π_m^* denotes the desired fraction of robots that need to perform task T_m with the highest priority. Also, let $\alpha_i = [\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,M}]^T$

denote the vector that indicates the priorities of the tasks for robot i:

$$\alpha_{i,m} = \begin{cases} 1, & \text{if task } T_m \text{ has the highest priority for robot } i \\ 0, & \text{otherwise.} \end{cases}$$

Since $\delta_{i,m}$ denotes the relative effectiveness with which robot i executes task T_m, this implies that

$$\alpha_{i,m} = 1 \quad \Rightarrow \quad \delta_{i,m} \leq \frac{1}{\kappa}\delta_{i,n} \quad \forall n \in \mathcal{M},\ n \neq m,$$

where $\kappa > 1$ allows us to encode how the task priorities impact the relative effectiveness with which robots perform different tasks.

The heterogeneity in the capabilities of the robots is encoded using the *specialization* matrix $S_i = diag([s_{i,1}, \ldots, s_{i,M}])$ where $s_{i,m}$ denotes the relative suitability of robot i towards executing task T_m. This allows us to formulate the following Mixed Integer Quadratic Program (MIQP) to obtain the task priorities α, slack variables δ, and control input u:

$$\min_{u,\delta,\alpha} \ C\|\pi^* - \pi_h(\alpha)\|^2 + \sum_{i=1}^{N} \left(\|u_i\|^2 + \|\delta_i\|^2_{S_i} \right) \tag{6a}$$

$$\text{s.t.} \ \frac{\partial h_{i,m}}{\partial x_i} u_i \geq -\gamma(h_{i,m}(x)) - \delta_{i,m} \tag{6b}$$

$$\delta_{i,n} \geq \kappa\big(\delta_{i,m} - \delta_{max}(1 - \alpha_{i,m})\big), \quad n \neq m \tag{6c}$$

$$\mathbf{1}^T \alpha_i = 1 \tag{6d}$$

$$\|\delta_i\|_\infty \leq \delta_{max} \tag{6e}$$

$$\alpha \in \{0,1\}^{NM} \tag{6f}$$

$$\forall i \in \mathcal{N},\ \forall n, m \in \mathcal{M},$$

where $\pi_h(\alpha)$ is the current task prioritization of the robot team, and C is a constant which allows a trade-off between achieving the global task allocation specifications and minimizing energy consumption of the robots.

IV. A MIXED CENTRALIZED/DECENTRALIZED APPROACH

The optimization problem (6a)-(6f) introduced in the previous section is a MIQP since the entries of α, which determine the task priorities for each robot, can be either 0 or 1. As discussed before, there are two main issues related to solving this MIQP: (i) in most cases, it is a NP-hard problem and, as such, it is practically impossible to solve it in an online fashion [11]; (ii) in order to solve it, information about all robots is required in the evaluation of $\pi_h(\alpha)$. For these reasons, in this section, we introduce a mixed centralized/decentralized approach which allows each robot to solve a QP instead of a MIQP using only locally available information.

The presented approach is schematically represented in Fig. 1. We assume there exists a central agent that is able to communicate with all the robots and, in particular, gather all the information required to solve the MIQP (6a)-(6f), namely the positions x of all the robots. Once these values have been received, solving the MIQP provides the central agent with the optimal task allocation, α. The values of the entries of α_i are then transmitted from the central agent to robot i. Each robot, with the knowledge of the received α_i, can solve the following optimization problem:

$$\min_{u_i,\delta_i} \ \|u_i\|^2 + \|\delta_i\|^2_{S_i} \tag{7a}$$

$$\text{s.t.} \ \frac{\partial h_{i,m}}{\partial x_i} u_i \geq -\gamma(h_{i,m}(x)) - \delta_{i,m} \tag{7b}$$

$$\delta_{i,n} \geq \kappa\big(\delta_{i,m} - \delta_{max}(1 - \alpha_{i,m})\big), \quad n \neq m \tag{7c}$$

$$\|\delta_i\|_\infty \leq \delta_{max} \tag{7d}$$

$$\forall n, m \in \mathcal{M},$$

where the constraint (7c) encodes the relative priority between each pair of tasks. The problem (7a)-(7d) is a QP and, therefore, it can be efficiently solved by each robot even under real-time constraints [11].

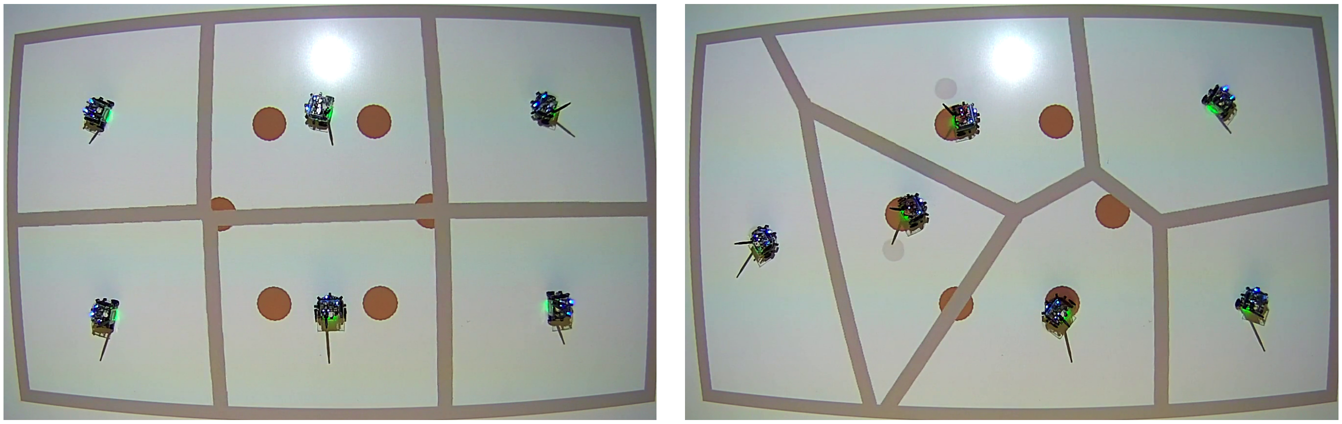
It is worthwhile pointing out that the presented task allocation strategy is *dynamic* and it does not need to be synchronous with the calculation of the robots' inputs u_i. In fact, as will be shown in the next section through experiments with real robots, while each robot evaluates its control input through (7a)-(7d), the central agent allocates the tasks to the robot by solving (6a)-(6f). This way, if the global task specification π^* changes over time, the robots will receive an updated value of α which will affect the task priorities through the constraint (7c).

V. EXPERIMENTS

The presented centralized/decentralized task allocation framework has been implemented on a team of 6 differential-drive robots operating on the Robotarium, a remotely accessible swarm robotics testbed [12]. The experimental setup consists in the robots moving in a 2.5m$\times$1.5m rectangular domain and performing two tasks: environmental surveillance and formation control. The former is realized by implementing the coverage control algorithm proposed in [13], whereas the latter task consists in driving to specified locations in the domain. Figure 2 presents snapshots of the experiment.

The robots are heterogeneous in their ability to perform the two different tasks, with three different specialization matrices: $S_i = diag([0.75, 0.25])$ for $i = 1$, $S_i = diag([0.25, 0.75])$ for $i = 2$, $S_i = diag([1, 0])$ for $i \in \{3, 4\}$, and $S_i = diag([0, 1])$ for $i \in \{5, 6\}$. The global specification on the task allocation π^* is changed from $\pi^* = [1, 0]^T$ to $\pi^* = [0.5, 0.5]^T$ halfway through the experiment.

The robots are initialized at random locations in the domain. When $\pi^* = [1, 0]^T$, the robots are asked to surveil the environment. The centralized unit assigns, to all the robots, an $\alpha_i = [1, 0]^T$ which places highest priority on performing the coverage task. As a result, the robots execute the gradient descent algorithm presented in [13] and attain a centroidal Voronoi tessellation (CVT). Figure 2a illustrates the robots attaining this configuration. At a certain point in time, the π^* changes to $\pi^* = [0.5, 0.5]^T$. The robots receive updated values of α_i from the centralized computer



(a) (b)

Fig. 2. Experimental deployment of the proposed mixed centralized/decentralized task allocation strategy on a team of 6 differential-drive robots operating on the Robotarium [12]. With the initial task allocation specifications set to $\pi^* = [1, 0]^T$, all the robots perform coverage control with a higher priority than formation control. Consequently, they achieve a centroidal Voronoi tessellation (CVT). At a later time, the task allocation specifications are changed to $\pi^* = [0.5, 0.5]^T$. As a result, the fraction of robots executing coverage control and formation control with highest priority are equal. The centralized computer assigns task priorities to the robots by solving the MIQP in (6) taking into account the heterogeneity in the capabilities of the robot.

and accordingly change their task prioritizations. Figure 2b illustrates half of the robots achieving the formation while the rest continuing to perform coverage. This takes into account the specialization matrices of the robots as well as the energy required to execute each task.

REFERENCES

[1] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[2] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," in *Cooperative Robots and Sensor Networks 2015*. Springer, 2015, pp. 31–51.

[3] 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.

[4] T. Balch and L. E. Parker, *Robot teams: from diversity to polymorphism*. AK Peters/CRC Press, 2002.

[5] M. Egerstedt, J. N. Pauli, G. Notomista, and S. Hutchinson, "Robot ecology: Constraint-based control design for long duration autonomy," *Annual Reviews in Control*, 2018.

[6] G. Notomista, S. Mayya, S. Hutchinson, and M. Egerstedt, "An optimal task allocation strategy for heterogeneous multi-robot systems," *arXiv preprint arXiv:1903.08641*, 2019.

[7] M. T. Hale, A. Nedić, and M. Egerstedt, "Cloud-based centralized/decentralized multi-agent optimization with communication delays," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 700–705.

[8] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[9] G. Notomista and M. Egerstedt, "Constraint-driven coordinated control of multi-robot systems," *arXiv preprint arXiv:1811.02465*, 2018.

[10] J. Cortés and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1699–1706.

[13] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.