Journal of Intelligent & Robotic Systems (2023) 108:15
https://doi.org/10.1007/s10846-023-01881-8

REGULAR PAPER

A Concurrent Mission-Planning Methodology for Robotic Swarms Using Collaborative Motion-Control Strategies

Kasra Eshaghi[1] · Goldie Nejat[1] · Beno Benhabib[1]

Received: 11 October 2022 / Accepted: 30 April 2023 / Published online: 30 May 2023

Abstract

Swarm robotic systems comprising members with limited onboard localization capabilities rely on employing collaborative motion-control strategies to successfully carry out multi-task missions. Such strategies impose constraints on the trajectories of the swarm and require the swarm to be divided into worker robots that accomplish the tasks at hand, and support robots that facilitate the movement of the worker robots. The consideration of the constraints imposed by these strategies is essential for optimal mission-planning. Existing works have focused on swarms that use leader-based collaborative motion-control strategies for mission execution and are divided into worker and support robots prior to mission-planning. These works optimize the plan of the worker robots and, then, use a rule-based approach to select the plan of the support robots for movement facilitation – resulting in a sub-optimal plan for the swarm. Herein, we present a mission-planning methodology that concurrently optimizes the plan of the worker and support robots by dividing the mission-planning problem into five stages: division-of-labor, task-allocation of worker robots, worker robot path-planning, movement-concurrency, and movement-allocation. The proposed methodology concurrently searches for the optimal value of the variables of all stages. The proposed methodology is novel as it (1) incorporates the division-of-labor of the swarm into worker and support robots into the mission-planning problem, (2) plans the paths of the swarm robots to allow for concurrent facilitation of multiple independent worker robot group movements, and (3) is applicable to any collaborative swarm motion-control strategy that utilizes support robots. A unique pre-implementation estimator, for determining the possible improvement in mission execution performance that can achieved through the proposed methodology was also developed to allow the user to justify the additional computational resources required by it. The estimator uses a machine learning model and estimates this improvement based on the parameters of the mission at hand. Extensive simulated experiments showed that the proposed concurrent methodology improves the mission execution performance of the swarm by almost 40% compared to the competing sequential methodology that optimizes the plan of the worker robots first and, then, the plan of the support robots. The developed pre-implementation estimator was shown to achieve an estimation error of less than 5%.

Keywords Swarm Robotics · Task-allocation · Localization · Motion planning · Constrained optimization

JEL Classification 93C85 · 93A16 · 68T40

1 Introduction

Swarm robotic systems (SRSs) are a category of multi-robot systems (MRSs) that are distinguished by their use of a high number of simpler robots and the dependence of these on collaborating with each other to achieve mission goals [1–3]. SRSs have been proposed for search and rescue [4], surveillance [5], and manufacturing [6], among other applications. Further *operational limitations*, such as computational, locomotion and localization limitations, may be imposed on SRSs depending on the robots and application at hand.

In this paper, the use of a SRS, subject to localization limitations, is considered for multi-task missions. Each task in the mission is expected to be accomplished by a subset of the SRS at hand, commonly referred to as a *coalition* of robots [7, 8]. In this regard, the successful overall completion of a multi-task mission can be achieved through effective *mission-planning*, which can be formulated to optimize a mission execution performance index through two primary stages: (1) task-allocation and (2) trajectory planning [9].

✉ Beno Benhabib
benhabib@mie.utoronto.ca

[1] Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Rd, Toronto, ON M5S 3G8, Canada

The *task-allocation* stage, typically, focuses on the dynamic subgrouping of the swarm into coalitions for the tasks at hand. Membership in the formed coalitions is dynamic in the sense that individual robots may be part of different coalitions as the operation is ongoing. The outcome of this stage comprises *time-phased routes* for all individual swarm robots. Once a task-allocation solution is adopted, the subsequent *trajectory-planning* stage converts the time-phased routes into motion trajectories for the individual robots. These trajectories would be determined based on the swarm motion execution strategy that is in use, subject to all operational constraints.

Herein, we propose a mission-planning methodology for swarms whose members are unable to localize individually (*i.e.,* estimate their positions with respect to a global reference frame). This could be the case due to an operating environment that prevents access to external measurement devices, and due to the limited onboard sensing capabilities of the member robots (*e.g.,* short-range sensing, limited field-of-view, etc.). In such cases, the swarm would need to rely on using a collaborative motion-control strategy that allows the member robots to compensate for their localization limitations. These strategies, however, impose various constraints on the trajectories of the member robots that must be satisfied for effective motion control. This represents a *constrained mission-planning problem*, where the task-allocation stage of mission-planning is constrained to solutions that allow for determining executable swarm trajectories in the trajectory-planning stage. Conversely, the *unconstrained mission-planning problem* would simply involve swarms that operate without trajectory constraints. For the latter swarms, it is assumed that executable trajectories can be planned for any task-allocation solution considered, which frequently may not be the case.

This paper begins with a detailed discussion of related works and summarizes our novel contributions in Section 2. The swarm mission-planning problem is formally defined in Section 3, and the proposed mission-planning methodology is detailed in Section 4. Section 5 presents multiple simulated examples to illustrate the proposed mission-planning methodology, and to validate its performance. The paper is concluded in Section 6.

2 Related Works

Existing approaches to SRS mission-planning can be categorized based on the operational limitations considered in their formulation, including computational, locomotion, and localization ones. Studies that are most similar to the work proposed in this paper, typically, address locomotion [10–16] and localization [17–20] limitations of the swarm. These along with other works related to swarm mission-planning are reviewed below in detail.

2.1 Computational Limitations

The task-allocation stage of mission-planning is a NP-hard combinatoric optimization problem [21]. Existing solutions to task-allocation include decentralized market-based [22–26], probabilistic [27–33], and behavior-based [34–45] methods. Optimization-based methods that utilize various exact or heuristic algorithms have also been proposed [46, 47]. The noted approaches assume that the computational limitations of the robots only affect the task-allocation stage, and the corresponding robot trajectories can be planned after the optimal task-allocation is determined. This could also be the case when dynamic reallocation due to the changes in the mission [48–58], limited battery life [59, 60], task synchronization [61–64], and heterogeneity in task requirements [65] are addressed. However, such a sequential methodology is effective only when the executability of the found trajectories is independent of task-allocation. This does not apply when locomotion and localization limitations are considered.

2.2 Locomotion Limitations

The locomotion limitations of swarm robots would affect the executability of the trajectories that are planned for a task-allocation solution considered. In such cases, the literature advocates that if a robot trajectory is not executable within the limits of its capabilities, changes must be made to the mission plan at the task-allocation level [10–16]. This represents a constrained mission-planning problem and calls for a strategy that searches for the optimal task-allocation and swarm trajectories concurrently. For example, in [14], inter-robot and robot-obstacle collisions are considered as locomotion limitations, and a concurrent methodology is utilized to plan the mission as some task-allocations may increase the number of potential collisions that need to be avoided. In [15] and [16], a scenario where the trajectories of the robots may be blocked by debris in the environment is considered. This work also considers the collaboration of the swarm with other debris-removal *support* agents that can clear the environment.

2.3 Localization Limitations

Localization limitations also affect the executability of the planned trajectories, and a swarm with such limitations would benefit from a concurrent mission-planning methodology. Such limitations commonly burden swarms that have limited onboard sensing hardware and are operating in environments that prevent access to external measurement devices (*e.g., millirobot*-based swarms [66–73]). These swarms depend on leader-based or tether-based motion-control strategies that allow them to localize and execute their trajectories by interacting with each other. These strategies divide the swarm into *worker* robots that accomplish the tasks at hand, and *support* robots that *facilitate* the *movement* of the worker robots.

Leader-based motion control strategies [74–77] use leader robots that are equipped with enhanced localization technology, such as access to ground positioning systems (GPS) or complex onboard sensors. These leaders operate as support robots, and facilitate the movement of the remaining follower (worker) robots, that localize by fusing their measured distance to the leader, and the position of the leader [78–80]. This constrains the motion of the followers as they need to wait to be 'picked up' and accompanied to their destinations by a leader. *Tether-based* strategies utilize a secondary team of mobile sensors as support robots which form a wireless tether to maintain connectivity between the worker robots and the environment [81]. Measurements made to the environment are, then, used in conjunction with an a priori known map to localize the worker robots. Tether-based motion strategies also required the worker robots to be 'picked up' by a tether. As such, a task-allocation stage that does not consider the subsequent trajectory-planning stage could yield coalitions and corresponding time-phased routes that cannot be executed as optimally planned.

The use of collaborative motion-control strategies was addressed in [17–19], where it is assumed that task-allocation is provided, and the problem is to plan for effective coalition execution. They suggest a rule-based approach to allocating the support robots to the movements of the swarm that need to be facilitated through leader-based methods. These methods were extended to include task-allocation within the planning framework in [20]. Therein, a market-based task-allocation procedure is used to form coalitions for the tasks at hand and once completed, the abovementioned rule-based approach is applied and used to determine the trajectories of the member robots.

2.4 Challenges and Contributions

Limited past mission-planning works for swarms with localization limitations, commonly, propose sequential approaches that, first, allocate worker robots to the tasks at hand, and, then, use a rule-based strategy to allocate the support robots to the worker robot movements that need to be facilitated [17–20]. Such approaches, typically, result in sub-optimal motion plans, as they do not allow for changes to the task-allocation of the worker robots to accommodate for the constraints imposed on their trajectories. Furthermore, these methods are only applicable to leader-based motion-control strategies.

Past works that do advocate for concurrent optimization of the task-allocation and trajectory-planning stages, on the other hand, have, typically, only focused on addressing the locomotion limitations of the swarm that constrain the trajectories of the robots to being planned for avoiding inter-robot and robot-obstacle collisions [10–16]. These methods cannot be directly applied to swarms with localization limitations since the constraints imposed by a collaborative motion-control strategy are different than those imposed by collision avoidance. Thus, there, specifically, exists a need to develop a concurrent mission-planning methodology for swarms that rely on collaborative motion-control strategies due to their localization limitations. Such a methodology must also be adaptable to any collaborative swarm motion-control strategy. Herein, we, thus, present a concurrent mission-planning methodology that addresses the constraints imposed by collaborative motion-control strategies required by swarms with localization limitations.

Our proposed methodology plans robot trajectories to allow for concurrent facilitation of the worker robot movements through an adopted collaborative motion-control strategy. This contrasts past works that consider collaboration among worker and support robots, but limit the support robots to facilitating one movement/job at a time [15–20]. Our methodology is also applicable to any swarm motion-control strategy that requires collaboration with support robots, in contrast to past works that are only applicable to leader-based methods [17–20]. Finally, the proposed methodology incorporates the division-of-labor of the swarm (into worker robots and support robots), directly, into the solution of the mission-planning problem. This mechanism for optimizing the swarm's mission has not been considered in the past, where the division-of-labor for the swarm is assumed to be given [15–20].

The proposed methodology also incorporates a pre-implementation estimator to limit unnecessary use of computational resources. While a concurrent methodology is more suitable to the constrained mission-planning problem than a sequential one, it can be more computationally intensive. The pre-implementation estimator, thus, estimates the improvement that can be achieved through a concurrent methodology, and allows the user to decide if the expected improvements justify its additional computational requirements. In contrast, other works that address the constrained mission-planning problem assume that a concurrent methodology always provides significant improvements [10–16].

3 Problem Definition

This paper addresses the problem of mission-planning for robotic swarms. We, specifically, target swarms comprising robots with localization limitations, which consequently rely on collaborative motion-control strategies. As noted in the

Introduction, this represents *a constrained mission-planning problem*.

3.1 Pre-Implementation Estimator

When constraints are present, a concurrent planning methodology may tangibly improve the mission execution performance compared to the sequential planning methodology. However, occasionally, this improvement may be deemed insignificant by the user, especially since a concurrent methodology can be computationally more intensive. The first issue that needs to be addressed is, thus, the development and use of a pre-implementation estimator to estimate the potential improvement in mission execution performance that a concurrent planning methodology may achieve. Utilizing such an improvement metric, one can decide whether the computational challenges of a concurrent methodology would justify the improvement it may yield.

In this regard, let I represent the improvement in mission execution performance that may be achieved using a concurrent planning methodology versus a sequential one. In our work, the mission execution performance index is formulated as the mission-completion time (*i.e., makespan*), t_C, defined as the time it takes to complete the mission at hand in its entirety. Herein, the improvement in mission execution is formulated as the difference between the mission-completion times achievable via the sequential and concurrent planning methodologies, t_{CS} versus t_{CC}, respectively, normalized with respect to the mission-completion time of the sequential methodology:

$$I = \frac{t_{CS} - t_{CC}}{t_{CS}} \times 100\%. \tag{1}$$

The problem at hand is, thus, to develop a function, $f()$, that can estimate the improvement in mission-completion time, $\hat{I}$, as a function of the mission parameters at hand, $\boldsymbol{V}$:

$$\hat{I} = f(\boldsymbol{V}). \tag{2}$$

The parameters of the mission at hand, $\boldsymbol{V}$**,** include the number of robots in the swarm, n_R, and the parameters that define the mission's n_V tasks, $\left\{V_i\right\}_{i=1}^{n_V}$. Each task, V_i, is, in turn, defined by its Cartesian position with respect to a global reference frame, ${}^G\boldsymbol{x}_{V_i}$, the number of worker robots required to complete it, n_{WV_i}, and its working time (*i.e.,* the time that the worker robots must spend to complete it), t_{WV_i}.

3.2 Mission-Planning

Following a decision that a concurrent planning methodology could indeed yield a better solution, it would, then, need to be applied to solving the constrained swarm mission-planning problem. As noted above, mission-planning for swarms with localization limitations must consider the constraints imposed by the collaborative motion-control strategies that are used to compensate for these limitations.

Sub-problem #1 – Division-of-labor The use of collaborative motion-control strategies requires the swarm to be divided into two functionally different groups: a group of *worker robots* that accomplish the mission's tasks, and a group of *support robots* that facilitate the movement of the worker robots. Herein, we consider the use of a homogeneous swarm with n_R robots, whose members can be designated to operate in either role. Thus, as one of the goals, the mission-planning problem needs to (optimally) select the number of robots in each role, n_W worker robots and n_S support robots, respectively.

Following a division-of-labor choice, mission-planning, as formulated herein, would comprise three sub-problems:

(1) collaboration amongst the worker robots,
(2) collaboration amongst the support robots, and
(3) collaboration between these two groups.

Sub-problem #2 – Task-allocation Collaboration between the worker robots is necessary for accomplishing the mission's tasks, where each task is addressed by a *coalition* of worker robots (*i.e.,* a group of worker robots that collectively work on the task). Membership in the coalitions is dynamic in the sense that individual robots may be part of different coalitions as the mission progresses. As such, a key variable in planning the mission is determining the specific coalitions that each worker robot belongs to, denoted herein by an $[(n_V + 1) \times (n_V + 1)]$ matrix $\boldsymbol{A}$, where n_V is the number of tasks in the given mission. The elements of this *task-allocation* matrix would represent the (worker) robots arriving from different past tasks to form the specific coalition of robots needed to accomplish the task at hand. Thus, the goal of the *task-allocation* stage is to determine the collaboration amongst the worker robot group.

Sub-problem #3 – Movement-allocation To form the above-mentioned coalitions, the worker robots are required to move to the locations of their respective allocated tasks by executing a number of *movements*. These movements are to be facilitated by the support robots through the adopted collaborative motion-control strategy, which may require multiple support robots to collaborate with each other. The allocation of the support robots to these movements is denoted herein by an $\left[\left(n_M + 1\right) \times \left(n_M + 1\right)\right]$ matrix $\boldsymbol{S}$, where n_M is the number of movements. The elements of this *movement-allocation* matrix would represent the (support) robots needed to facilitate the movements of the worker robots in their attempt to form the necessary coalitions. Thus, the goal of the *movement-allocation* stage is to determine the collaboration amongst members of the support robot group.

Sub-problem #4 – Trajectory-planning An adopted motion-control strategy would also impose connectivity constraints that must be maintained between the worker robots that are executing their movements, and the support robots that facilitate these movements. Thus, to achieve collaboration between the worker and support robot groups, their respective trajectories must be planned for the task and movement-allocations, where the strings P_{TW} and P_{TS} are the trajectory of the worker and support robots, respectively. Thus, the goal of the *swarm trajectory-planning* stage is to determine the collaboration between members of the worker and support robot groups.

Thus, in this work, it is proposed to solve the above defined sub-problems by minimizing the mission-completion time:

Minimize:

$$t_C = v\left(n_W, \boldsymbol{A}, \boldsymbol{S}, P_{TW}, P_{TS}\right), \tag{3}$$

Subject to:

$$g\left(n_W, \boldsymbol{A}, \boldsymbol{S}, P_{TW}, P_{TS}\right) \leq 0, \tag{4}$$

$$h\left(n_W, \boldsymbol{A}, \boldsymbol{S}, P_{TW}, P_{TS}\right) = 0. \tag{5}$$

Above, $v()$ is a function that evaluates the mission-completion time, t_C, based on the following variables: the number of worker robots, n_W; the allocation of worker robots to the tasks, $\boldsymbol{A}$; the allocation of support robots to the movements of the worker robots, $\boldsymbol{S}$; and the trajectories of the worker and support robots, P_{TW} and P_{TS}, respectively. It must be noted that the number of support robots, n_S, is calculated based on the number of worker robots, n_W, and the number of robots in the swarm, n_R (*i.e.,* $n_W + n_S = n_R$).

The mission-completion time is the time at which the last task is completed:

$$t_C = \max\left\{t_{CV_i}\right\}_{i=1}^{n_V}, \tag{6}$$

where t_{CV_i} is the completion time of Task *i*, V_i. The objective function $v()$ in Eq. (3) is constructed based on the trajectories of the worker robots, as these determine when they arrive at their final destinations, based on which the start and end time of the tasks can be determined. However, worker robot trajectories are dependent on all other variables discussed above and as such, the construction of the objective function requires a candidate solution for all these variables. Furthermore, $g()$ and $h()$ in Eqs. (4) and (5) are general functions that describe the inequality and equality constraints on the variables. These functions correspond to the connectivity requirements of the adopted collaborative motion-control strategy, the number of worker robots required for each task, the number of support robots required for each movement, precedence constraints on the movements, etc.

4 Proposed Methodology

The proposed mission-planning methodology, summarized in Fig. 1 below, starts by obtaining an estimate of the potential improvement in mission-completion time that can be achieved by employing the proposed concurrent optimization methodology versus a sequential one, Section 4.1. This improvement is deemed satisfactory if it is above a user defined minimum threshold, in which case the proposed *concurrent* planning methodology is applied. Otherwise, a *sequential* methodology can be used.

The proposed concurrent planning methodology tries to minimize the mission-completion time of the swarm by considering the following (interdependent) five planning stages: (1) division-of-labor, (2) task-allocation, (3) determining worker robot paths, (4) determining movement-concurrency, and (5) movement-allocation. Each stage is responsible for optimizing its corresponding variable through a provided objective function that relates it to the mission-completion time, via a specific search engine.

The interdependence of the five stages noted above implies that a candidate solution considered at any stage would affect the next (on the five-stage process). For example, a possible

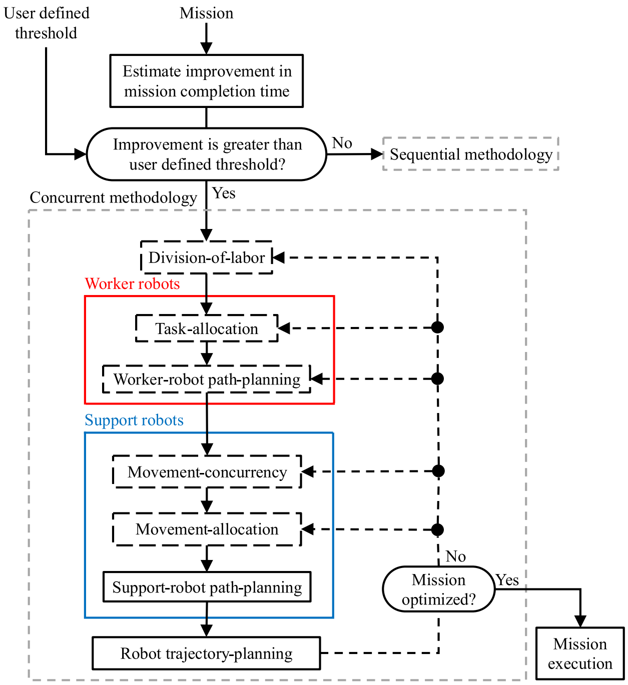


Fig. 1 Overview of the proposed concurrent mission-planning methodology

solution for task-allocation can only be generated for a candidate solution already chosen during the division-of-labor stage, which indicates the number of worker robots that are available to be allocated to the tasks at hand. Similarly, a candidate solution for worker robot paths can only be generated when a possible solution to task-allocation is available, which indicates the sequence of coalitions that each worker robot is a part of.

Once candidate solutions for all the five stages are generated, the corresponding support robot paths and swarm robot trajectories can be determined and used to calculate the mission-completion time that is used to guide the search for the optimal values of the variables in each stage. Details of the specific stages, the proposed approach to constructing and optimizing their respective objective functions, and their interdependence are detailed in Sections 4.2 to 4.5.

It must be noted that a swarm may indeed be subject to uncertainties during mission execution, as detailed in Section 4.6. These uncertainties, however, need not be considered during mission-planning [82, 83], as it is conjectured herein that the optimality of the proposed mission-planning methodology, over the competing sequential methodology is maintained even in the face of potential real-time uncertainties.

The competing sequential planning methodology, considered herein, contrasts the proposed concurrent methodology in that it optimizes the plan of the worker robots, which includes the number of worker robots selected, their allocation to the tasks at hand, and their paths, while assuming that they are not constrained to using a collaborative swarm motion-control strategy to address their localization limitations. Once a worker robot plan is determined, the plan of the support robots, which includes movement-concurrency and their allocation to the movements at hand, is optimized based on the adopted collaborative swarm motion-control strategy. The sequential methodology uses the same stages shown in Fig. 1. Unlike the proposed concurrent methodology, however, it optimizes the variables in these stages in a sequential order and does not consider their interdependence.

The optimal solution found through a sequential-planning method results in (1) a division-of-labor that selects the maximum number of robots to operate as worker robots, and the minimum number of robots to operate as support robots, (2) determining worker robot paths that move the worker robots directly to their allocated task locations, and (3) determining a support robot plan that does not facilitate multiple worker robot movements concurrently. This potentially would result in a sub-optimal plan for the swarm.

4.1 Pre-Implementation Estimator

An estimate of the improvement in mission-completion time that can be achieved by implementing the proposed concurrent-planning methodology over a sequential one would allow a user to justify the additional computational challenges of this methodology. As shown in Fig. 1, once the improvement in mission-completion time is estimated, it is compared to a user defined threshold. If the improvement is estimated to be greater than this threshold, a concurrent-planning methodology is implemented. Otherwise, a sequential-planning methodology can be deemed adequate. It must be noted that the improvement in mission-completion time is estimated while assuming the swarm is not subject to uncertainties during mission execution.

The user defined threshold would be selected based on the computational power available to the user. If the concurrent planning methodology takes negligible time due to the availability of extensive computational power, then the threshold would be set to a low value, and the concurrent planning methodology would be applied to all missions. With limited computational power, however, the concurrent planning methodology would take significantly longer than the sequential methodology. In this case, the threshold would be set to a higher value to ensure the user is rewarded with significant improvement in mission-completion time if he/she chooses to spend additional time to optimize the swarm's plan through the concurrent methodology. This approach, typically, results in the threshold being selected as a higher value in dynamic scenarios that require mission re-planning during execution, compared to offline mission-planning. This is true since the available computational power during mission execution is limited by the onboard hardware of the member robots, whereas an external computer can be used in an offline setting. It must be noted that this paper does not consider mission re-planning during execution.

Herein, a multi-layer perceptron (MLP) is proposed for obtaining an estimate of the improvement in mission-completion time that can be achieved through a concurrent methodology versus a sequential one, $\widehat{I}$. MLPs are a class of fully connected neural networks that consist of an input layer, one or more hidden layers, and an output layer [84].

Herein, the inputs to the model are selected based on an ablation study, where potential model inputs are systematically removed to evaluate their contribution to the estimation process. Inputs that are found to increase the estimation error of the model are removed. The model architecture is determined through a random-search strategy for selecting the loss function of the model, the number of hidden layers, and the number of neurons in each hidden layer.

Model Inputs The proposed model estimates the improvement in mission-completion time that can be achieved through a concurrent optimization methodology versus a sequential one, $\widehat{I}$, based on the parameters of the mission at hand, $\boldsymbol{V}$, as detailed in Eq. (2), and the mission plan found through a sequential planning methodology.

In this regard, the mission parameters used as input to the model include the number of robots in the swarm, n_R, and the following task-specific parameters:

1. The positions of the tasks:
$$\boldsymbol{x}_{x_V} = \left[{}^G x_{V_i}, {}^G y_{V_i}, \dots\right], i = 1, .., n_V, \tag{7}$$
where $\left({}^G x_{V_i}, {}^G y_{V_i}\right)$ is the position of task V_i with respect to a global reference frame, ${}^G F$. It is assumed that the starting position of the swarm (*i.e.,* the home base), V_0, is located at the origin for all missions.

2. The number of worker robots, n_{WV_i}, required per task:
$$\boldsymbol{x}_{n_W} = \left[n_{WV_i}, \dots\right], i = 1, .., n_V. \tag{8}$$

3. The working time, t_{WV_i}, of each task (*i.e.,* the duration that the worker robots must work on the task to complete it):
$$\boldsymbol{x}_{t_W} = \left[t_{WV_i}, \dots\right], i = 1, .., n_V. \tag{9}$$

The model also makes use of the number of worker robots, number of support robots, and the mission-completion time as found through the sequential planning methodology:

$$\boldsymbol{x}_S = \left[n_{WS}, n_{SS}, t_{CS}\right]. \tag{10}$$

Above, n_{WS} and n_{SS} are the optimal number of worker and support robots that are selected by the sequential planning methodology, and t_{CS} is the mission-completion time that the swarm achieves with the sequential plan. The sequential planning methodology would select the number of worker and support robots as:

$$n_{WS} = n_R - n_{Smin}, \text{ and} \tag{11}$$

$$n_{SS} = n_{Smin}, \tag{12}$$

where n_{Smin} is the minimum number of support robots required for the mission at hand. The minimum number of support robots would depend on the adopted swarm motion-control strategy.

Model architecture The proposed model, Fig. 2, uses a single fully connected layer between the input and output layers of the network. The number of robots in the swarm, n_R, the parameters of the mission at hand, Eq. (7)—Eq. (9), and the parameters of the sequential solution, Eq. (10), are concatenated into one vector, and passed through a hidden layer with 2,000 neurons. The results are, then, passed through an output layer with one neuron that estimates the improvement in mission-completion time that can be achieved through the proposed concurrent methodology to swarm mission-planning, $\hat{I}$. All neurons in the hidden and output layers use the rectified linear activation function. The model is trained using the mean absolute error as its loss function.

4.2 Division-of-Labor

The *division-of-labor* stage determines the number of robots that should operate in worker and support roles, respectively, for the mission at hand. When a homogeneous swarm is used, where the roles of the member robots are interchangeable, the division must strike an (optimal) balance between the number of worker and support robots. An increase in the number of worker robots, at the expense of support ones may at first appear as allowing for the faster parallel completion of more tasks by increasing the number of distinct coalitions. However, swarm constraints may lead to significant degradation in mission-completion time due to the lack of sufficient number of support robots required to accomplish the desired parallelism. In the case of a heterogeneous swarm, one could expect that the numbers of worker and support robots be individually constrained by their respective availability. One may note that division-of-labor does not change during mission execution – once the role of a robot is selected, it remains the same until the mission is completed. This may be due to the robots' specific onboard hardware.

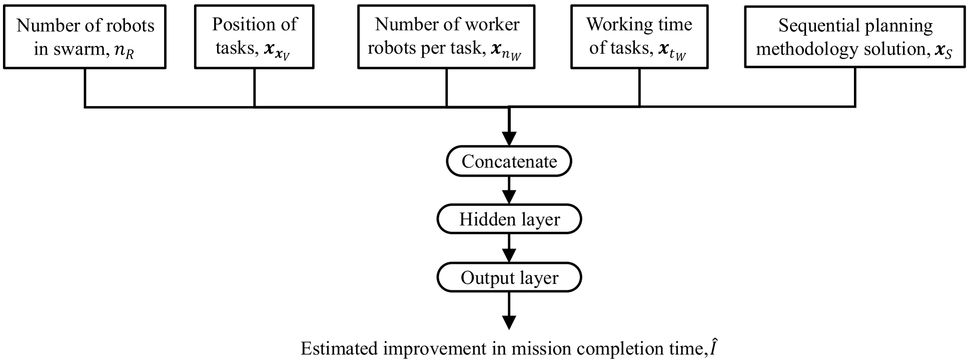

Number of robots in swarm, n_R
Position of tasks, $\boldsymbol{x}_{x_V}$
Number of worker robots per task, $\boldsymbol{x}_{n_W}$
Working time of tasks, $\boldsymbol{x}_{t_W}$
Sequential planning methodology solution, $\boldsymbol{x}_S$
Concatenate
Hidden layer
Output layer
Estimated improvement in mission completion time, $\hat{I}$

Fig. 2 Mission-completion time improvement estimation model architecture

For a homogeneous swarm of n_R robots, the search for the optimal number of worker and support robots, n_W^* and n_S^*, respectively, can be completed through a discrete search engine, where $n_W + n_S = n_R$. Herein, the selection objective function is the overall goal of minimum mission-completion time, t_C:

$$\min\ t_C = v\left(n_W\right), \tag{13}$$

subject to:

$$n_W \geq \max\left\{n_{WV_i}\right\}_{i=1}^{n_V}, \text{and} \tag{14}$$

$$n_W \leq n_R - n_{Smin}. \tag{15}$$

Above, Eq. (14) states that the overall total number of worker robots must be equal to or greater than the number of worker robots required for each task, n_{WV_i}. Equation (15), in turn, states that there must be a sufficient number of robots remaining to operate in a support role, n_{Smin}, as required by the adopted swarm motion-control strategy.

The mission-completion time in Eq. (13) can be calculated only once the four planning stages that follow division-of-labor, in Fig. 1, optimize their associated variables for the candidate division-of-labor solution passed on downward. The search for optimal division-of-labor begins by generating a candidate solution, n_W, Fig. 3. This candidate solution is, then, passed downward to the task-allocation stage, where the corresponding optimal task-allocation, $\boldsymbol{A}^*\left(n_W\right)$, as well as the optimal worker robot paths, movement-concurrency, and movement-allocation, $P_{PC}^*\left(n_W, \boldsymbol{A}^*\right)$, $\boldsymbol{Q}^*\left(n_W, \boldsymbol{A}^*, P_{PC}^*\right)$, and, $\boldsymbol{S}^*\left(n_W, \boldsymbol{A}^*, P_{PC}^*, \boldsymbol{Q}^*\right)$, respectively, are determined. These are used to determine the support robot paths and swarm trajectories, and to calculate the mission-completion time in Eq. (13) for guiding the search for the optimal division-of-labor, n_W^*. The swarm trajectories corresponding to the optimized variables of all stages, n_W^*, $\boldsymbol{A}^*\left(n_W^*\right)$, $P_{PC}^*\left(n_W^*, \boldsymbol{A}^*\right)$, $\boldsymbol{Q}^*\left(n_W^*, \boldsymbol{A}^*, P_{PC}^*\right)$, and $\boldsymbol{S}^*\left(n_W^*, \boldsymbol{A}^*, P_{PC}^*, \boldsymbol{Q}^*\right)$, are passed forward to the mission execution stage, Section 4.6. It must be noted that the details of the variables are omitted in Fig. 3 for simplicity.

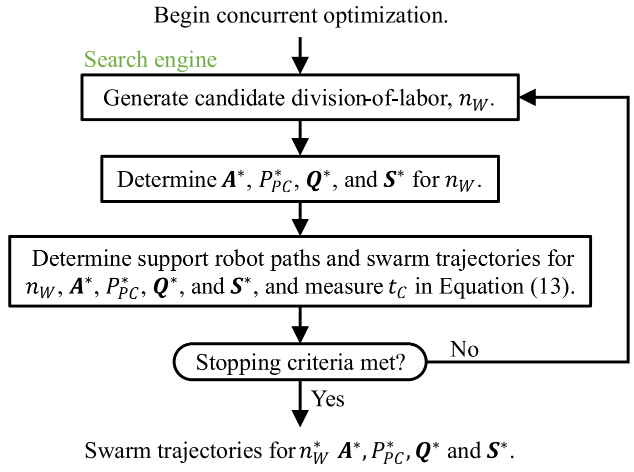

Begin concurrent optimization.
Search engine
Generate candidate division-of-labor, n_W.
Determine $\boldsymbol{A}^*$, P_{PC}^*, $\boldsymbol{Q}^*$, and $\boldsymbol{S}^*$ for n_W.
Determine support robot paths and swarm trajectories for n_W, $\boldsymbol{A}^*$, P_{PC}^*, $\boldsymbol{Q}^*$, and $\boldsymbol{S}^*$, and measure t_C in Equation (13).
Stopping criteria met?
No
Yes
Swarm trajectories for n_W^* $\boldsymbol{A}^*$, P_{PC}^*, $\boldsymbol{Q}^*$ and $\boldsymbol{S}^*$.

Fig. 3 Division-of-labor optimization process

4.3 Task-Allocation and Path-Planning for Worker Robots

A mission plan for the worker robots comprises the specification of which robots work on which tasks as well as their respective paths during the entirety of the mission. Both issues are addressed below in Sub-Sections 4.3.1 and 4.3.2, respectively.

4.3.1 Task-Allocation

The *task-allocation* stage focuses on the optimal grouping of worker robots into *coalitions* (*i.e.,* teams of robots that collectively work on a task). The number of worker robots available for this stage, n_W, is passed down by the division-of-labor stage, Section 4.1 above.

For a given mission of n_V tasks, $\left\{V_i\right\}_{i=1}^{n_V}$, the allocation of the n_W worker robots to the tasks at hand is described by an $[(n_V + 1) \times (n_V + 1)]$ matrix, $\boldsymbol{A}\left(n_W\right)$ – a possible *task-allocation solution*. The search algorithm, thus, needs to identify the optimal solution, $\boldsymbol{A}^*(n_w)$, that minimizes the mission-completion time, t_C, for the number of worker robots considered:

$$\min\ t_C = v\left(\boldsymbol{A}\left(n_W\right)\right). \tag{16}$$

The above formulation is subject to providing each coalition with the number of worker robots required for their respective task, Eq. (17), and ensuring the number of worker robots leaving a coalition (for other coalitions/tasks) is less than or equal to the number of worker robots in that coalition, Eq. (18):

$$\sum\nolimits_{i=0}^{n_V} a_{ij} = n_{WV_j}, j = 1, \ldots, n_V, \tag{17}$$

$$\sum\nolimits_{j=1}^{n_V} a_{ij} \leq n_{WV_i}, i = 0, \ldots, n_V. \tag{18}$$

Above, a_{ij} is an element of the task-allocation matrix $\boldsymbol{A}\left(n_W\right)$, and it represents the number of (worker) robots that are part of sub-coalition, c_{ij}, which breaks-off from the Coalition i, C_i, for Task i, once it is completed, and subsequently joins the Coalition j, C_j for Task j. Equation (17) indicates that the number of robots allocated to a task, comprising different sub-coalitions, must be equal to the number of robots required to complete it. Equation (18), in turn, indicates that the number of robots leaving a task, through different sub-coalitions, must be less than or equal to the number of robots in the coalition for this task.

One may note that a solution to the task-allocation sub-problem, as described above, only indicates the number of worker robots in each sub-coalition, and does not specify the robots' identities. This is true since the worker robots are homogeneous and interchangeable, and as such, the specific worker robots for each sub-coalition can be arbitrarily selected.

The search space of the task-allocation stage consists of n_V^2 discrete variables that must be selected to supply each task with the required number of worker robots. These variables represent all the elements of the task-allocation matrix, excluding those belonging to the diagonals and to the first column (associated with home base), which must be equal to zero. The number of discrete values that each element a_{ij} can take on is calculated based on the minimum number of worker robots required by either V_i or V_j, and a *base* parameter, b, that indicates the grouping of worker robots into sub-coalitions. For example, $b = 5$ indicates that the sub-coalitions must be grouped in teams of base 5 (*i.e.*, $\mathrm{mod}\left(a_{ij}, b\right) = 0$). Thus, the search space of task-allocation has $n_{sol\boldsymbol{A}}$ number of solutions:

$$n_{sol\boldsymbol{A}} = \prod_{i=0}^{n_V}\left(\prod_{j=1,j\neq i}^{n_V}\left(\frac{\min\left(n_{WV_i}, n_{WV_j}\right)}{b} + 1\right)\right). \tag{19}$$

The base parameter, b, can be used to reduce the size of the search space of task-allocation for computational efficiency. Furthermore, the search space for task-allocation can be searched through using any combinatoric optimization search engine, such as Genetic Algorithms [85].

A solution to the task-allocation sub-problem can be visualized through a directional graph, where the vertices are the n_V tasks in the swarm's mission, $\left\{V_i\right\}_{i=1}^{n_V}$, and the directed edges represent the sub-coalition of worker robots that form the coalitions of each task. For an example mission of $n_V = 3$ tasks, with $n_R = 30$ robots available, two possible task-allocation solutions are shown in Fig. 4(a) and (b), respectively. The task-allocation matrices corresponding to these candidate solutions are given in Eq. (20) and Eq. (21), respectively. The columns of these matrices represent the individual sub-coalitions that form the coalition for each task, while the rows of these matrices represent how the coalition for each task is divided to form the sub-coalitions for other tasks.

In Fig. 4(a), the coalition of worker robots for Task V_2, C_2, comprises only one sub-coalition of 15 worker robots that leave home base, V_0, to work on V_2. This is shown through a_{02} in Eq. (20). Upon completion of V_2, the Coalition C_2 is divided into two sub-coalitions: one of ten robots that move on to V_1, c_{21}, as shown through a_{21} in Eq. (20), and another of five robots that move on to V_3, c_{23}, as shown through a_{23} in Eq. (20).

$$^{1}\boldsymbol{A}\left(n_W = 20\right) = \begin{bmatrix} 0 & 0 & 15 & 5 \\ 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{20}$$

$$^{2}\boldsymbol{A}\left(n_W = 25\right) = \begin{bmatrix} 0 & 0 & 15 & 10 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \end{bmatrix}. \tag{21}$$

The mission-completion time in Eq. (16) above can be calculated only once the three planning stages that follow task-allocation, in Fig. 1, optimize their associated variables for the candidate task-allocation solution passed on downward. The search for optimal task-allocation begins by generating a candidate task-allocation solution for the division-of-labor solution considered, $\boldsymbol{A}\left(n_W\right)$, Fig. 5. This candidate task-allocation solution is, then, passed downward to the worker robot path-planning stage, where the corresponding optimal worker robot paths, $P_{PC}^{*}\left(n_W, \boldsymbol{A}\right)$, as well as the optimal movement-concurrency and movement-allocation,

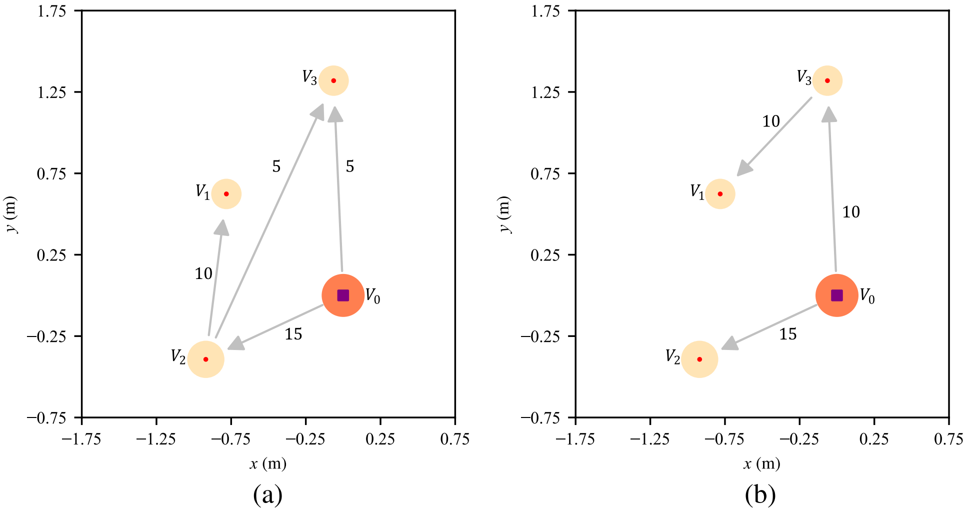


Fig. 4 Worker robot task-allocation – two possible solutions

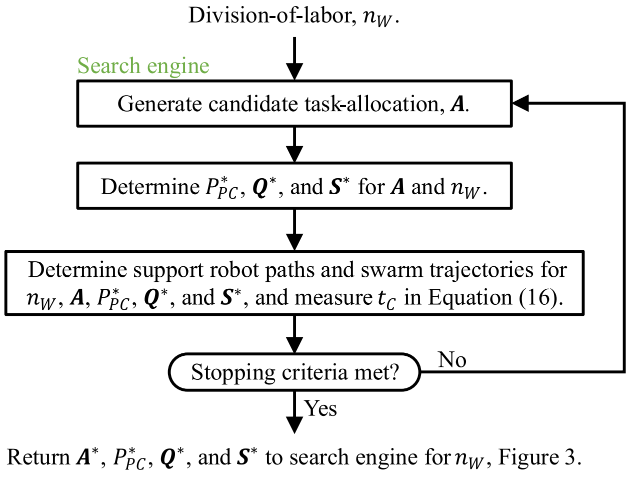

Division-of-labor, n_W.
Search engine
Generate candidate task-allocation, $\boldsymbol{A}$.
Determine P^*_{PC}, $\boldsymbol{Q}^*$, and $\boldsymbol{S}^*$ for $\boldsymbol{A}$ and n_W.
Determine support robot paths and swarm trajectories for n_W, $\boldsymbol{A}$, P^*_{PC}, $\boldsymbol{Q}^*$, and $\boldsymbol{S}^*$, and measure t_C in Equation (16).
Stopping criteria met?
No
Yes
Return $\boldsymbol{A}^*$, P^*_{PC}, $\boldsymbol{Q}^*$, and $\boldsymbol{S}^*$ to search engine for n_W, Figure 3.

Fig. 5 Task-allocation optimization process

$\boldsymbol{Q}^*\left(n_W, \boldsymbol{A}, P^*_{PC}\right)$, and $\boldsymbol{S}^*\left(n_W, \boldsymbol{A}, P^*_{PC}, \boldsymbol{Q}^*\right)$, respectively, are determined. These are used to determine the support robot paths and swarm trajectories, and to calculate the mission-completion time in Eq. (16) for guiding the search for the optimal task-allocation. It must be noted that the details of the variables are omitted in Fig. 5 for simplicity.

4.3.2 Worker Robot Path-Planning

For a task-allocation solution considered, $\boldsymbol{A}\left(n_W\right)$, the path-planning stage determines the optimal paths of the worker robots. Herein, it is assumed that worker robot members of a sub-coalition travel together along the same path. Thus, paths are planned for sub-coalitions, and the paths of individual worker robots are inferred according to the sub-coalitions they are a part of.

The goal of path-planning is to find a balance between (i) aiming for each sub-coalition to get to its next task as quickly as possible and (ii) allowing the support robots to facilitate these paths efficiently. For example, if each sub-coalition were to operate without collaborating with support robots, it would move along the shortest possible path. However, due to the need for collaboration, sub-coalition paths must be planned to also consider the support robots which may, potentially, be facilitating multiple paths concurrently for enhanced efficiency in mission execution. Herein, this goal is achieved by allowing the temporary/intermediate re-location of sub-coalitions to *task-stops* that represent tasks that they are not allocated to, enroute to their allocated task. At these intermediate task-stops they remain as *visitors*, not participating in the work carried out by other worker robots allocated to this task.

In this regard, let the point-to-point (PTP) path that each sub-coalition travels along be defined as a string of task locations that it moves to:

$$p_{Pc_{ij}} = \left\{ m_{c_{ij}k} \right\}_{k=1}^{n_{pc_{ij}}}, \tag{22}$$

where $p_{Pc_{ij}}$ is the PTP path of sub-coalition c_{ij} corresponding to allocation a_{ij} of the task-allocation solution considered, $\boldsymbol{A}\left(n_W\right)$, and $m_{c_{ij}k}$ is the k^{th} task location that the sub-coalition moves to. The PTP of the sub-coalition consists of $n_{pc_{ij}} \geq 2$ points, where the first point is the task that the sub-coalition starts from, V_i, the last point is the task that the sub-coalition is allocated to, V_j, and all points in between represent the task-stops that the sub-coalition visits.

This stage of mission-planning, thus, determines the optimal paths of all sub-coalitions (of worker robots), P^*_{PC}, for the division-of-labor and task-allocation solutions considered, to minimize the mission-completion time, t_C:

$$\min t_C = v\left(P_{PC}\left(n_W, \boldsymbol{A}\right)\right), \tag{23}$$

where $P_{PC}\left(n_W, \boldsymbol{A}\right) = \left\{ p_{Pc_{ij}} \right\}_{(\forall i,j | a_{ij} > 0)}$.

The path of a sub-coalition is a variable length string of stops that it visits, as it moves from its current to its next task. Thus, for each sub-coalition, the path-planning stage must select (i) the number of stops it makes along its path, and (ii) the ordered sequence of stops for this path. This has to be completed for all sub-coalitions of the task-allocation solution considered. The search space of path-planning, thus, includes $n_{solP_{PC}}$ number of solutions:

$$n_{solP_{PC}} = \prod_{i=1}^{n_C} \left(\sum_{j=0}^{n_{stops}} \text{Perm}\left(n_V - 1, j\right) \right). \tag{24}$$

Above, n_c is the number of sub-coalitions for the task-allocation considered and Perm is the permutation operator. The maximum number of stops that the sub-coalitions can make, n_{stops}, can be selected to reduce the size of the search space for reduce computation time. Furthermore, the search space for worker path-planning can be searched through using any combinatoric optimization search engine, such as Genetic Algorithms [85].

A solution to the path-planning of worker robots can be visualized through a graph, where the nodes represent the successive task-locations of a sub-coalition's path. Two possible example paths are shown in Fig. 6(a)-(b) for the task-allocation in Fig. 4(a), where each different color represents the path of a sub-coalition. The paths can also be visualized based on their decomposition into several *movements*, where $M_{c_{ij}k}$ represents the k^{th} movement for the path of sub-coalition c_{ij}.

In Fig. 6(a), the sub-coalition c_{21} of 10 worker robots moves directly from V_2 to V_1, as shown by the orange arrow, and movement $M_{c_{21}1}$. This path is denoted by:

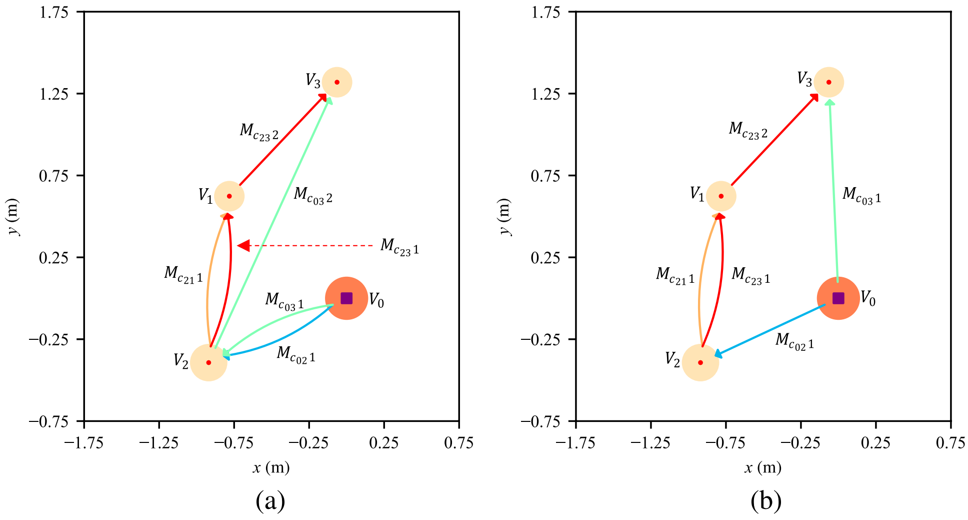


Fig. 6 Sub-coalition paths – two possibilities for the task-allocation solution in Fig. 4(a)

$$p_{Pc_{21}} = \{V_2, V_1\}. \tag{25}$$

In contrast, the sub-coalition c_{23} of five worker robots, moves along the red path (movements $M_{c_{23}}1$ and $M_{c_{23}}2$), which makes an intermediate stop at V_1. This path is denoted by:

$$p_{Pc_{23}} = \{V_2, V_1, V_3\}. \tag{26}$$

While this path may be longer than necessary, it would allow the support robots to facilitate a portion of the path of this sub-coalition concurrently with the path of sub-coalition c_{21}. This choice may be beneficial in optimizing the mission-completion time.

The mission-completion time in Eq. (23) above is calculated only once the two planning stages that follow worker robot path-planning, in Fig. 1, optimize their associated variables for the candidate worker robot paths solution passed on downward. The search for optimal worker robot paths begins by generating a candidate worker robot paths solution for the division-of-labor and task-allocation solutions considered, $P_{PC}(n_W, \boldsymbol{A})$, Fig. 7. This candidate worker robot paths solution is, then, passed downward to the movement-concurrency stage, where the corresponding optimal movement-concurrency, $\boldsymbol{Q}^*(n_W, \boldsymbol{A}, P_{PC})$, as well as the optimal movement-allocation, $\boldsymbol{S}^*(n_W, \boldsymbol{A}, P_{PC}, \boldsymbol{Q}^*)$, are determined. These are used to determine the support robot paths and swarm trajectories, and to calculate the mission-completion time in Eq. (23) for guiding the search for the optimal worker robot paths. It must be noted that the details of the variables are omitted in Fig. 7 for simplicity.

4.4 Movement-Concurrency, Movement-Allocation, and Path-Planning for Support Robots

The paths planned for the (worker robot) sub-coalitions can be divided into multiple *movements* that are facilitated by the support robots. Namely, the path of a sub-coalition, c_{ij}, may comprise multiple movements, where each movement, $M_{c_{ij}k}$, indicates the task location that the sub-coalition must be picked-up from and dropped-off to. For example, in Fig. 6(a), the path of the sub-coalition c_{21}, as shown by the orange arrow, comprises one movement, $M_{c_{21}}1$, which requires this sub-coalition to be picked-up from V_2 and dropped-off at V_1. Conversely, the path of the sub-coalition c_{23}, as shown by the red arrows, comprises two successive movements: $M_{c_{23}}1$, requiring the sub-coalition to be picked-up from V_2, and dropped-off at V_1, and $M_{c_{23}}2$, requiring this sub-coalition to be picked-up from V_1 and dropped off at V_3.

In this regard, the plan for the support robots consists of determining the movements that are facilitated concurrently, Section 4.4.1, allocating support robots to these movements, Section 4.4.2, and planning the corresponding path of the support robots, Section 4.4.3.

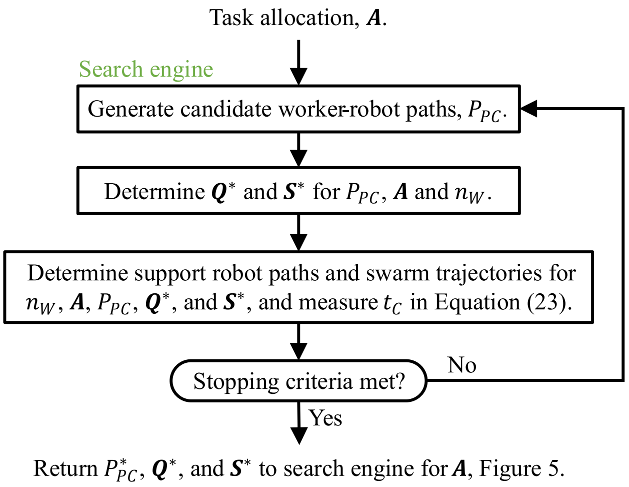

Task allocation, $\boldsymbol{A}$.
Search engine
Generate candidate worker-robot paths, P_{PC}.
Determine $\boldsymbol{Q}^*$ and $\boldsymbol{S}^*$ for P_{PC}, $\boldsymbol{A}$ and n_W.
Determine support robot paths and swarm trajectories for n_W, $\boldsymbol{A}$, P_{PC}, $\boldsymbol{Q}^*$, and $\boldsymbol{S}^*$, and measure t_C in Equation (23).
Stopping criteria met?
No
Yes
Return P_{PC}^*, $\boldsymbol{Q}^*$, and $\boldsymbol{S}^*$ to search engine for $\boldsymbol{A}$, Figure 5.

Fig. 7 Worker robot paths optimization process

4.4.1 Movement-Concurrency

As noted above, a worker robot paths solution considered, $P_{PC}(n_W, \boldsymbol{A})$, may include multiple movements that have the same pick-up and drop-off locations. For the example followed herein, Fig. 6(a), movement $M_{c_{21}1}$ of the path for sub-coalition c_{21} and movement $M_{c_{23}1}$ of the path for sub-coalition c_{23} both require their respective sub-coalition to be picked-up from V_2, and dropped off at V_1. Similarly, movements $M_{c_{03}1}$ and $M_{c_{02}1}$ also share the same pick-up and drop-off locations. Such movements can be planned to be facilitated concurrently by the support robots. Multiple sub-coalition movements that are facilitated concurrently would (theoretically) start and end at the same time and would be facilitated by the same set of support robots. This allows the efficiency of the support robots to be enhanced.

The concurrency of the n_M movements at hand can be defined by an $\left[\left(n_M+1\right)\times\left(n_M+1\right)\right]$ symmetrical matrix, $\boldsymbol{Q}\left(n_W, \boldsymbol{A}, P_{PC}\right)$, where the element $q_{ij}\epsilon\{0,1\}$ indicates whether the movements corresponding to Row i and Column j are executed concurrently (*i.e.*, 1 for yes, 0 for no). The movements at hand can be arbitrarily ordered on the Rows/Columns of this matrix. Furthermore, the diagonal elements of $\boldsymbol{Q}\left(n_W, \boldsymbol{A}, P_{PC}\right)$ are equal to one. The movement-concurrency stage, thus, decides on the optimal concurrent execution of the movements, $\boldsymbol{Q}^*\left(n_W, \boldsymbol{A}, P_{PC}\right)$, that minimizes the mission-completion time for the division-of-labor, task-allocation, and worker robot paths considered:

$$\min t_C = v\left(\boldsymbol{Q}\left(n_W, \boldsymbol{A}, P_{PC}\right)\right). \tag{27}$$

The above formulation is subject to the precedence constraints of the movements at hand, as multiple movements may have the same pick-up and drop-off location may not be executed concurrently if one corresponds to a sub-coalition that must be formed before the sub-coalition corresponding to the other.

The search space of the movement-concurrency consists of $\left[\left(n_M^2-n_M\right)/2\right]$ binary variables that must be selected to indicate the movements that are planned for concurrent facilitation. This search space has n_{solQ} number of solutions:

$$n_{solQ} = 2^{\left(n_M^2-n_M\right)/2}. \tag{28}$$

The search space can be searched through using any combinatoric optimization search engine, such as Genetic Algorithms [85].

Two example movement-concurrency candidates are shown in Eq. (29) and Eq. (30), respectively, for the worker robot path-plan in Fig. 6(a), where movement M_0 represents the starting position of the support robots (*i.e.*, the home base). The movements are ordered along the rows (left to right) and columns (top to bottom) as: M_0, $M_{c_{02}1}$, $M_{c_{03}1}$, $M_{c_{03}2}$, $M_{c_{21}1}$, $M_{c_{23}1}$, $M_{c_{23}2}$. In Eq. (29), movements $M_{c_{02}1}$ and $M_{c_{03}1}$ are planned for concurrent facilitation, as shown through $q_{12} = q_{21} = 1$. In this example, movements $M_{c_{21}1}$ and $M_{c_{23}1}$ are also planned for concurrent facilitation, as shown through $q_{45} = q_{54} = 1$. In contrast, the example in Eq. (30) does not plan for any of the movements to be executed concurrently as all non-diagonal elements are equal to zero.

$$^{1}\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{29}$$

$$^{2}\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{30}$$

The mission-completion time in Eq. (27) above is calculated only once the movement-allocation for the candidate movement-concurrency solution considered is optimized. The search for the optimal movement-concurrency begins by generating a candidate movement-concurrency solution for the division-of-labor, task-allocation and worker robot paths solutions considered, $\boldsymbol{Q}\left(n_W, \boldsymbol{A}, P_{PC}\right)$, Fig. 8. The candidate movement-concurrency solution is, then, passed downward to the movement-allocation stage, where the corresponding optimal movement-allocation, $\boldsymbol{S}^*\left(n_W, \boldsymbol{A}, P_{PC}, \boldsymbol{Q}\right)$, is determined. These are used to determine the support robot paths and swarm trajectories, and to calculate the mission-completion time in Eq. (27) for guiding the search for the optimal movement-concurrency. It must be noted that the details of the variables are omitted in Fig. 8 for simplicity.

4.4.2 Movement-Allocation

Allocation of support robots to the movements of the sub-coalitions are carried out at this stage for a movement-concurrency solution considered, $\boldsymbol{Q}\left(n_W, \boldsymbol{A}, P_{PC}\right)$. The number of support robots available for this allocation, n_S, would be defined based on the division-of-labor stage solution.

The allocation of the support robots is defined by an $\left[\left(n_M+1\right)\times\left(n_M+1\right)\right]$ matrix $\boldsymbol{S}\left(n_W, \boldsymbol{A}, P_{PC}, \boldsymbol{Q}\right)$, where the element s_{ij} represents the number of support robots which facilitate the movement corresponding to Row/Column *j*, after facilitating the movement corresponding to Row/Column *i*. This stage, thus, determines the optimal allocation of the support robots to the movements at hand, $\boldsymbol{S}^*\left(n_W, \boldsymbol{A}, P_{PC}, \boldsymbol{Q}\right)$, for

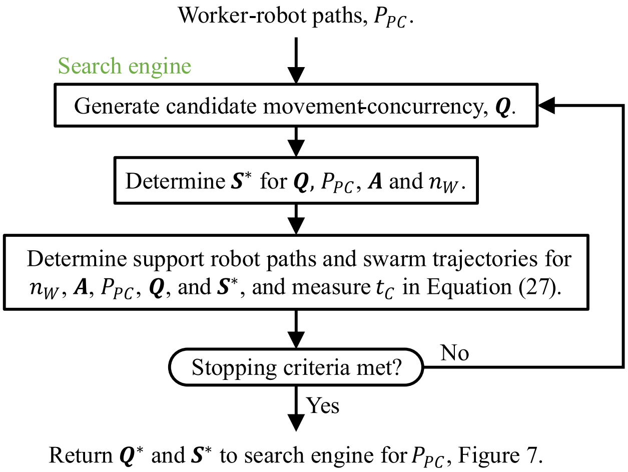

Worker-robot paths, P_{PC}.
Search engine
Generate candidate movement-concurrency, $\boldsymbol{Q}$.
Determine $\boldsymbol{S}^*$ for $\boldsymbol{Q}$, P_{PC}, $\boldsymbol{A}$ and n_W.
Determine support robot paths and swarm trajectories for n_W, $\boldsymbol{A}$, P_{PC}, $\boldsymbol{Q}$, and $\boldsymbol{S}^*$, and measure t_C in Equation (27).
Stopping criteria met?
No
Yes
Return $\boldsymbol{Q}^*$ and $\boldsymbol{S}^*$ to search engine for P_{PC}, Figure 7.

Fig. 8 Movement-concurrency optimization process

minimum mission-completion time, for the division-of-labor, task-allocation, worker robot paths, and movement-concurrency solutions considered:

$$\min t_C = v\left(\boldsymbol{S}\left(n_W, \boldsymbol{A}, P_{PC}, \boldsymbol{Q}\right)\right). \tag{31}$$

The allocation of the n_S available support robots must be completed such that each movement is allocated the required number of robots, Eq. (32), and ensure the number of support robots leaving a movement (to facilitate another movement) is less than or equal to the number of support robots allocated to that movement, Eq. (33):

$$\sum_{i=1}^{n_M}\left(q_{ij}\sum_{k=0}^{n_M} s_{ki}\right) = n_{SM_j}, j = 1, \dots, n_M, \tag{32}$$

$$\sum_{j=0}^{n_M}\left(q_{ij}\sum_{k=1}^{n_M} s_{jk}\right) \le n_{SM_i}, i = 0, \dots, n_M. \tag{33}$$

Above, n_{SM_j} is the number of support robots required to complete the movement corresponding to Row/Column j of $\boldsymbol{S}\left(n_W, \boldsymbol{A}, P_{PC}, \boldsymbol{Q}\right)$, As noted in Section 4.4.1, the movements at hand can be arbitrarily ordered on the Rows/Columns of this matrix. Equation (32) and Eq. (33) also reflect that the same support robots can be allocated to movements that are planned for concurrent facilitation. The movements of the worker-paths also have precedence constraints that must be satisfied when determining their support robot allocation. For example, for the worker path solution in Fig. 6(a), movement $M_{c_{23}1}$ must be completed before movement $M_{c_{23}2}$.

The number of support robots required for a movement would depend on the collaborative motion control strategy used. Leader-based methods would need at least one support robot for each movement, while the number of support robots required for the tether-based strategy would depend on the length of the tethers used. The minimum number of support robots required for facilitating the movements of the sub-coalitions can be calculated as the maximum number of support robots required for any of the movements:

$$n_{Smin} = \max\left\{n_{SM_i}\right\}_{i=1}^{n_M}. \tag{34}$$

The minimum number of required support robots must be used in conjunction with the constraint in Eq. (15) to select a feasible division-of-labor in the swarm.

The search space of the movement-allocation stage consists of n_M^2 discrete variables that must be selected to supply each movement with the required number of support robots. The search space has n_{solS} number of solutions:

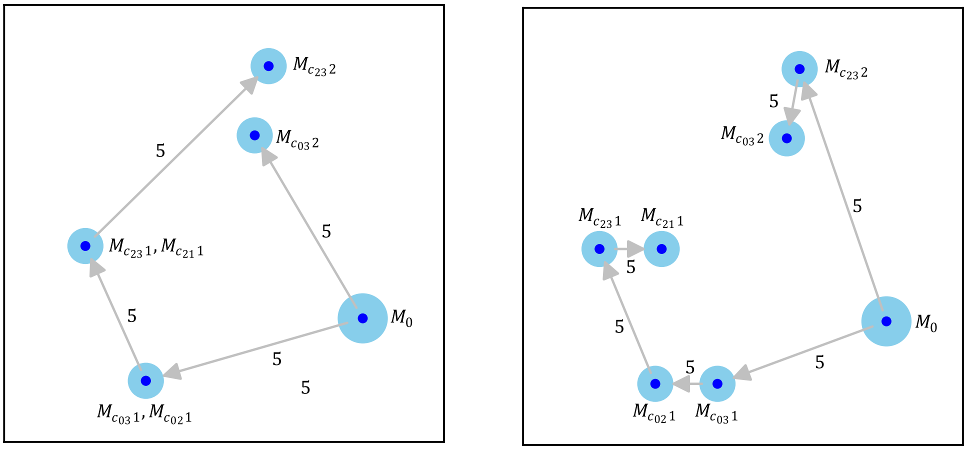
$$n_{solS} = \prod_{i=0}^{n_M}\left(\prod_{j=1,j\neq i}^{n_M}\left(\frac{\min\left(n_{SM_i}, n_{SM_j}\right)}{b}+1\right)\right), \tag{35}$$

where b is a *base* parameter that indicates the grouping of worker robots into sub-coalitions which, as described in Section 4.3.1, can be used to reduce the size of the search space of movement-allocation for computational efficiency. The size of the search space of movement-allocation is calculated through the same approach as that of task-allocation detailed in Section 4.3.1. The search space for task-allocation can be searched through using any combinatoric optimization search engine, such as Genetic Algorithms [85].

As for the (worker robot) task-allocation sub-problem, a solution to the movement-allocation sub-problem can be visualized through a directional graph, where the vertices are the movements that must be completed for the adopted sub-coalition paths, and the directed edges represent the allocation of support robots to these movements. The vertices may include multiple movements, indicating that these are planned for concurrent facilitation.

Two example movement-allocation solutions are shown in Fig. 9(a) and (b), for the worker path solution shown in Fig. 6(a). The movement-allocation in Fig. 9(a) corresponds to the movement-concurrency solution in Eq. (29), and the movement-allocation solution in Eq. (36) below, while the movement-allocation in Fig. 9(b) corresponds to the movement-concurrency solution in Eq. (30), and the movement-allocation solution in Eq. (37) below. The movements are ordered along the rows (left to right) and columns (top to bottom) in Eqs. (36) and (37), as they were in Eq. (29) and (30) in Section 4.4.1: M_0, $M_{c_{02}1}$, $M_{c_{03}1}$, $M_{c_{03}2}$, $M_{c_{21}1}$, $M_{c_{23}1}$, $M_{c_{23}2}$.

In Fig. 9(a), the movements $M_{c_{02}1}$ and $M_{c_{03}1}$ are synchronized, as shown through $q_{12} = q_{21} = 1$ of Eq. (29). The support robots allocated to these movements were previously at home base, M_0, as indicated by s_{01}, Eq. (36). Movements $M_{c_{21}1}$ and $M_{c_{23}1}$ are also planned for concurrent execution in this example, as shown through $q_{45} = q_{54} = 1$, Eq. (29). The support robots allocated to these movements are the same ones that facilitated movements $M_{c_{02}1}$ and $M_{c_{03}1}$, s_{14}, Eq. (36).



(a) (b)

Fig. 9 Swarm movement-allocation – two possibilities for the worker path solution in Fig. 6(a)

$$^{1}\boldsymbol{S} = \begin{bmatrix} 0&5&0&5&0&0&0 \\ 0&0&0&0&5&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&5 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{bmatrix}, \tag{36}$$

$$^{2}\boldsymbol{S} = \begin{bmatrix} 0&0&5&0&0&0&5 \\ 0&0&0&0&0&5&0 \\ 0&5&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&5&0&0&0 \\ 0&0&5&0&0&0&0 \end{bmatrix}. \tag{37}$$

The search for the optimal movement-allocation begins by generating a candidate movement-allocation solution for the division-of-labor, task-allocation, worker robot paths, and movement-concurrency considered, $\boldsymbol{S}(n_W, \boldsymbol{A}, P_{PC}, \boldsymbol{Q})$, Fig. 10. These are then used to determine the support robot paths and swarm trajectories, and to calculate the mission-completion time in Eq. (31) for guiding the search for the optimal movement-allocation. It must be noted that the details of the variables are omitted in Fig. 10 for simplicity.

4.4.3 Support Robot Path-Planning

The path-planning stage for support robots determines the paths that allow them to facilitate the movement of the worker robots. This stage, followed by the robot trajectory planning stage, Section 4.5, is invoked every time the mission-completion time needs to be calculated for a combination of the variables determined in the planning stages detailed above.

The paths of all the support robots are planned to facilitate the worker robot movements and to optimize the swarm-motion control strategy adopted. Such strategies determine the optimal paths of the support robots, P^*_{PS}, to minimize the localization error, and thus, the motion control performance the worker robots:

$$\min e_L = r\left(P_{PS}\left(n_W, \boldsymbol{A}, P_{PC}, \boldsymbol{Q}, \boldsymbol{S}\right)\right), \tag{38}$$

where e_L is the localization error of the worker robots along their PTP paths, $P_{PS} = \left\{p_{PS_i}\right\}_{i=1}^{n_S}$ is the PTP path of all support robots, and $r()$ is the function that describes the dependence of localization error on the PTP paths of the support robots. The PTP path of a support robot is defined as a sequence of locations that it needs to achieve, where p_{PS_i} is the PTP path of Support Robot i. Path planning for the support robots would be constrained by the limited sensing range of the member robots, and the connectivity requirements of the adopted collaborative motion-control strategy.

The objective function in Eq. (38) is constructed based on the collaborative swarm motion-control strategy adopted. For example, the tether-based motion control strategy minimizes the localization error by placing the support robots in a tether formation with minimum length. The objective

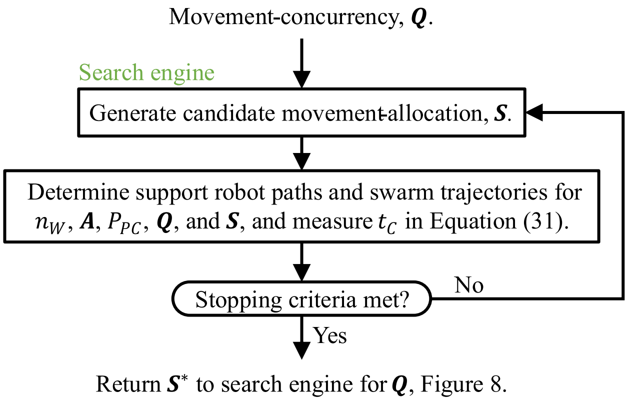

Movement-concurrency, $\boldsymbol{Q}$.
Search engine
Generate candidate movement-allocation, $\boldsymbol{S}$.
Determine support robot paths and swarm trajectories for n_W, $\boldsymbol{A}$, P_{PC}, $\boldsymbol{Q}$, and $\boldsymbol{S}$, and measure t_C in Equation (31).
Stopping criteria met?
No
Yes
Return $\boldsymbol{S}^*$ to search engine for $\boldsymbol{Q}$, Figure 8.

Fig. 10 Movement-allocation optimization process

function for this strategy would be calculated based on the lengths of the tethers that connect the swarm to its environment, as formed through the paths of the support robots. Similarly, leader-based motion-control strategies position leader-robots to enhance their connectivity with the worker robots they are supporting. For these strategies, the localization error may be calculated based on the relative positions of the support robots to the worker robot groups.

Herein, we consider the application of the tether-based swarm motion-control strategy [81]. The tether-based strategy places the support robots in a tether formation for maintaining connectivity between the worker robots and a connectivity point in the environment. The connectivity point can either be a static support robot, placed at home base, or landmarks in the environment, whose positions are a priori known and can be sensed through hardware onboard the support robots. By maintaining connectivity to these points, the worker robots can indirectly sense them through the tether of support robots and achieve accurate localization and closed-loop motion control.

Planning the paths of the support robots for the tether-based motion control strategy involves three steps: (1) tether-formation, (2) node-allocation, and (3) path-planning. The tether-formation step selects the connectivity point that the worker robots connect to and the position of the tether nodes that must be achieved by the support robots to achieve this connectivity. The node-allocation stage allocates the support to the nodes of the tethers. Finally, the paths of the support robots are determined based on the formed tethers and the robots allocated to them.

Figure 11 shows the locations of the support robots (green) on tethers connecting the coalitions of worker robots (red) to the connectivity point at home base at two different task locations.

The planned paths of the support robots, P_{PS}, are passed down to the trajectory planning stage below, Section 4.5, where all swarm robot trajectories are determined and used to calculate the mission-completion time.

4.5 Robot Trajectory-Planning

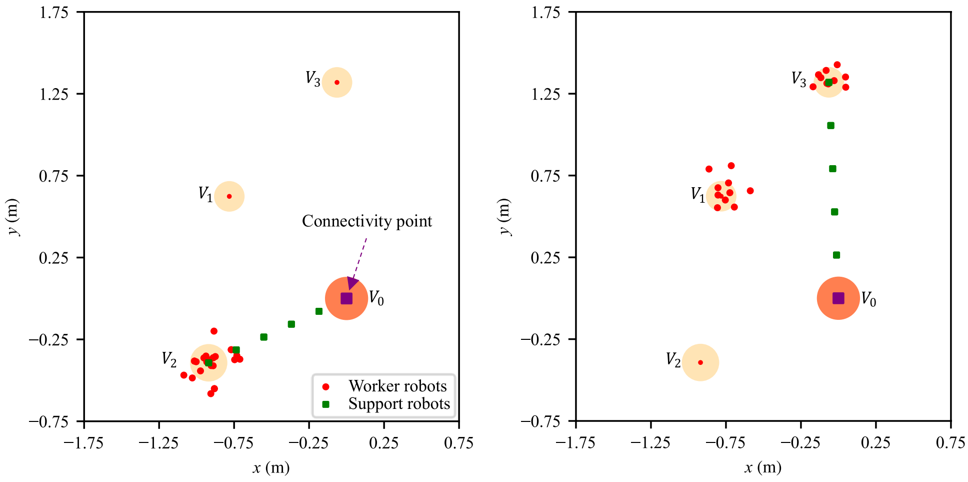
The goal of this last stage of the proposed mission-planning methodology is to calculate the mission-completion time. This metric is determined based on the trajectories of the worker robots, that are planned based on the paths of all swarm (worker and support) robots, as provided by the preceding stages, and a set of mission execution rules, discussed below.

In this regard, the PTP trajectory of Worker Robot i, p_{TW_i}, is defined as a sequence of time-phased positions that it needs to achieve. Each point on the PTP trajectory defines the position of the robot with respect to a global reference frame, ${}^{G}F$, and its arrival and departure times from these positions:

$$p_{TW_i} = \left\{ \left({}^{G}\boldsymbol{x}_{W_i 0}, t_{AW_i 0}, t_{DW_i 0} \right), \left({}^{G}\boldsymbol{x}_{W_i 1}, t_{AW_i 1}, t_{DW_i 1} \right), \ldots \right\}. \tag{39}$$

Above, ${}^{G}\boldsymbol{x}_{W_i j}$ is the global position of Worker Robot i at (PTP path) Point j on its path p_{PW_i}; $t_{AW_i j}$ is the time it must arrive at this position; and, $t_{DW_i j}$ is the time it must depart from this position toward the next. The trajectory of a Support Robot i is denoted, similarly, as p_{TS_i}. It must be noted that the PTP path of the worker robots can be inferred from the PTP path of the sub-coalitions, P_{PC}. The trajectories of the (worker and support) robots may be constrained by their battery life, which would limit the time that they can operate for mission execution.

As also noted above, the trajectory-planning stage determines the trajectories of all worker and support robots, $P_{TW} = \left\{p_{TW_i}\right\}_{i=1}^{n_W}$ and $P_{TS} = \left\{p_{TS_i}\right\}_{i=1}^{n_S}$, respectively. This consists of calculating the arrival and departure times



(a)

(b)

Fig. 11 Positions of support robots on tethers for connectivity to the worker robots at (*a*) Task V_2, and (*b*) Task V_3

of the robots at/from the positions on their path, while adhering to a set of mission execution rules. Herein it is assumed that worker robots (1) start working on their allocated tasks as soon as they arrive at their destination, (2) work on their tasks continuously, and (3) depart toward their next task location as soon as the support robots, allocated to facilitating their movement, have arrived and the task at hand has been completed. Similarly, support robots (1) start facilitating a movement as soon as they arrive at their pick-up location and the worker robots associated with this movement have completed their task, (2) facilitate the movement of the worker robots in full (*i.e.,* do not leave sub-coalitions part-way through their movement), and (3) depart from the drop-off point of their current movement to the pick-up point of their next allocated movement as soon as the current movement is completed. It must be noted that the time required by the support and worker robots to travel from one location to another is dependent on the distance travelled and subject to the circumvention of any obstacles.

The completion times of all tasks can, thus, be inferred from the trajectories of the worker robots, and used to determine the overall mission-completion time. The completion time of task V_i, t_{CV_i}, would be calculated based on the latest arrival time of all allocated worker robots to their position associated with this task, t_{AV_i}, plus the working time of this task:

$$t_{CV_i} = t_{AV_i} + t_{WV_i}, \tag{40}$$

where t_{WV_i} is the working time of task V_i, as provided by the mission parameters. The mission-completion time is, then, calculated as the latest completion time of the mission's tasks, Eq. (6).

Once calculated, the mission-completion time, t_C, is fed back upward to the division-of-labor, task-allocation, worker robot path-planning, movement-concurrency, and movement-allocation stages, to guide the search for the optimal combination of the variables considered in these stages, respectively.

4.6 Mission Execution

Trajectories of swarm robots are executed based on the rules detailed in Section 4.5 above. One may recall that these rules dictate when worker robots begin and finish their allocated tasks, and when the support robots begin and end facilitating their allocated worker robot movements.

It must also be noted that a swarm would be subject to uncertainties during its mission execution. These, for example, may include uncertainties in the working times of the tasks and robot travel times from one task location to another. The working time of a task may be different than the one accounted for during planning as various application-specific difficulties may arise during the completion of a task. For example, the task of finding a person in an environment may take longer than expected if the person's location cannot be accurately estimated [86]. The time required to travel from one task location to another may also vary from the planned travel time due to the circumvention of a priori unknown obstacles. Encountering such uncertainties may require the swarm robots to make alterations to the arrival and departure times of the points of their paths as per the mission execution rules. This may result in a difference between the mission-completion time that is expected based on the mission's plan, and the mission-completion time achieved in practice. It is conjectured herein that uncertainties encountered in run-time would not result in failure in mission execution as the mission execution methodology is event based, and neither in the optimality of the planned mission with respect to the sequential planning methodology, as will be empirically shown in the next Section.

5 Simulated Experiments

Extensive simulated experiments were conducted to evaluate the performance of the proposed mission-planning methodology. In Section 5.1, the performance of the proposed pre-implementation estimator for estimating the improvement in mission-completion time that can be achieved using the concurrent planning methodology versus a sequential one, Section 4.1, is evaluated. This is followed by a simulated example in Section 5.2 that illustrates the proposed concurrent planning methodology for a swarm of one hundred robots. Sections 5.3 and 5.4 provide comprehensive comparisons of the proposed concurrent mission-planning methodology to the sequential mission-planning methodology. Finally, Section 5.5 discusses the effects of mission parameters on the mission-completion time. The numerical simulations in Sections 5.2–5.4 were implemented in a custom simulation environment developed using Python 3.

One can note that, a simpler descriptive example of the application of the proposed concurrent planning methodology to a mission with fewer tasks than the one considered in subSection 5.2 is provided in Appendix 1. This example allows the reader to view the obtained optimal solution in graphical form.

5.1 Pre-Implementation Estimator

As noted in Fig. 1, the implementation of the proposed concurrent planning methodology could be invoked if the user notes a potential tangible improvement in mission-completion time that it can achieve over a sequential planning methodology. A MLP was proposed in Section 4.1 for estimating this improvement. In this sub-section, below, the performance of the proposed MLP is evaluated.

The proposed MLP was trained on a data set consisting of 10,000 points, where each point differed in the configuration of its mission and the number of robots in the swarm used to accomplish it. The tasks in the missions required between 15 to 35 worker robots, with individual (task) working times ranging from 10 to 15 s. The swarms, in turn, differed in their number of (homogeneous) robots, ranging from the minimum required (*i.e.,* $n_R = n_{Wmin} + n_{Smin}$) to approximately twice the minimum. For each data point, the mission plan was determined through the proposed concurrent and competing sequential mission-planning methodologies, and their respective mission-completion times, t_{CC} and t_{CS}, were calculated while assuming no uncertainties during mission execution, based on the worker robot trajectories as per the methodology in Section 4.5. The improvement in mission-completion time that can be achieved using the proposed concurrent methodology versus a sequential planning methodology, I, Eq. (1), was, then, calculated and used to train the model.

The effectiveness of the proposed MLP was evaluated through the k-fold cross validation procedure with five folds. The performance metric, for each data point, was calculated as the absolute difference between the true and estimated improvement in mission-completion time, respectively:

$$e = \left| I - \hat{I} \right|. \tag{41}$$

In summary, the proposed MLP achieved a mean estimation error of approximately 3% – on average, the estimated improvement value for using the proposed concurrent methodology (over the sequential one) was within 3% of the true value, pointing to an excellent estimator. Overall, the MLP achieved less than 5% error in 75% of the cases tested.

5.2 A Simulated Example

The implementation of the proposed swarm mission-planning methodology is illustrated herein for a mission comprising $n_V = 10$ tasks. The position of the tasks, as well as their number of required worker robots and task working times are shown in Fig. 12. The swarm at hand comprises $n_R = 100$ (homogeneous) robots, each of which can operate in either a worker or support role. Furthermore, in this example, the environment includes two points that can be used to provide connectivity between the swarm and the environment using the tether-based motion-control strategy. It is also assumed that the swarm is not subject to any uncertainties during mission execution.

The plan of all swarm robots was determined through the methodology presented in Section 4, which sought the optimal division-of-labor, task-allocation, worker robot paths, movement-concurrency, and movement-allocation. For the mission at hand, the minimum number of worker robots is $n_{Wmin} = 55$, and the minimum number of support robots is $n_{Smin} = 10$. The proposed mission-planning methodology, in turn, selected the number of worker and support robots as $n_W^* = 75$ and $n_S^* = 25$, respectively. The optimal plan of the swarm allowed it to complete the mission in $t_{CC} = 334s$. Table 1 details the start and end times of all the tasks in the swarm's mission. In contrast, the sequential planning methodology achieved a mission-completion time of $t_{CS} = 603s$. The proposed concurrent methodology, thus, achieves an improvement of $I = 45\%$. An animation of the execution of the mission, using the plan found through the proposed concurrent and competing sequential planning methodologies, is available at https://youtu.be/pVl-XXai3VE.

5.3 Comparison of the Proposed Concurrent Versus Sequential Planning Methodologies

The goal of the simulated experiments presented herein is to illustrate the superior optimality of the proposed concurrent mission-planning methodology, when compared to a typical sequential-planning methodology, where all comparative simulations are run under 'equal' conditions. In this regard, a series of 10,000 simulations were conducted, where each simulation differed in the configuration of the mission, and the number of robots in the swarm used to accomplish it. Each task required between 15 to 35 robots, with individual (task) working times ranging from 10 to 15 s. The swarms, in turn, differed in their number of (homogeneous) robots, ranging from the minimum required (*i.e.,* $n_R = n_{Wmin} + n_{Smin}$) to approximately twice the minimum. For each simulation, both the proposed concurrent- and competing sequential-planning methodologies were used to plan the missions of the swarm. The improvement in mission-completion time achieved by the proposed concurrent methodology over the sequential one, I, was then calculated through Eq. (1). In these simulations, it was assumed that the swarm is not subject to any uncertainties during mission execution.

It must be noted that the comparative evaluation was for all the 10,000 simulations considered, without removing those that might have been deemed by a user as *not worthy*. In summary, for the 10,000 simulations considered, an average improvement in mission-completion time of approximately 38% was noted when using the concurrent planning methodology versus a sequential one. The

	n_{WV_i}	t_{WV_i}
V_1	25	15
V_2	35	15
V_3	50	10
V_4	25	10
V_5	40	10
V_6	50	10
V_7	45	10
V_8	55	10
V_9	45	15
V_{10}	30	15

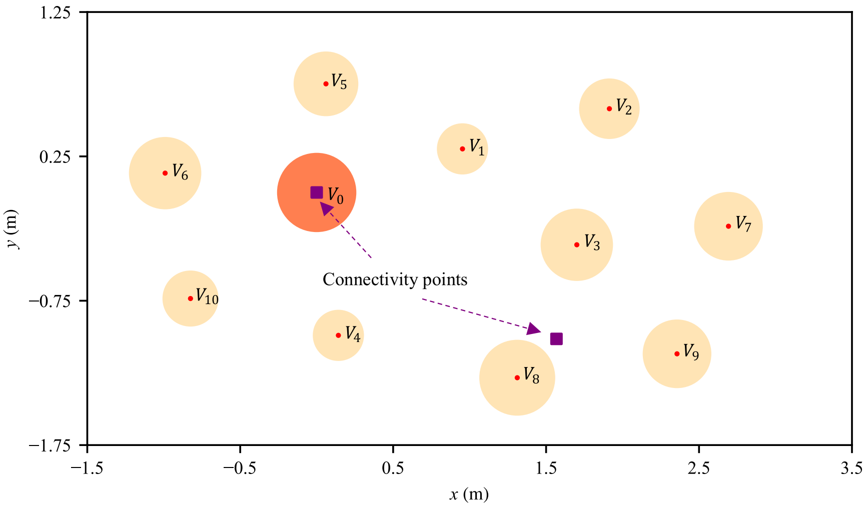


Fig. 12 Example mission

histogram of the improvements noted for these simulations is shown in Fig. 13 below.

5.4 Robustness Study

A robustness study, detailed herein, was completed for further validation of the proposed mission-planning methodology. The robustness analysis compared the proposed concurrent methodology to the competing sequential methodology for realistic scenarios where the swarm is subject to uncertainties in the working times of the tasks. In these scenarios, the working time of a task, during mission execution, varied from the working time that the swarm's mission was planned based on.

The comparative study was conducted through a series of simulated experiments, where each simulation differed in the configuration of the mission and the number of robots in the swarm used to accomplish it. The missions and swarm configurations were generated in the same manner described in Section 5.3: each task required between 15 to 35 robots, task working times ranged from 10 to 15 s, and the swarms were composed of homogeneous robots ranging from the minimum required to approximately twice the minimum. The plan of the swarm for each mission was determined through the proposed concurrent mission-planning methodology while assuming the working time of each task is known with certainty. Next, mission execution was re-simulated with random uncertainty added to the working time of each task:

$$t'_{WV_i} = t_{WV_i} + N\left(0, \sigma_W\right). \tag{42}$$

Above, t_{WV_i} is the working time of the task that the swarm's mission is planned based on, t'_{WV_i} is the true working time of the task during mission execution, and σ_W characterizes the (zero-mean) Normally-distributed uncertainty of task working time. For each simulation considered, the improvement in the mission-completion time achieved by the proposed plan compared to the plan obtained via the competing sequential methodology was calculated through Eq. (1) with uncertain task working times.

The 10,000 simulations used in Section 5.3 were repeated, per the process detailed above, for $\sigma_W = 2s$. An average improvement of approximately 38% was noted when using the plan found through the concurrent planning methodology versus a sequential one. The histogram of the improvements noted for these simulations is shown in Fig. 14 below – it is very similar to Fig. 13 above. One may note that in some rare cases (less than 1% of the 10,000 simulations), the proposed concurrent methodology fails to achieve improved mission-completion time.

This study empirically verifies our conjecture that the optimality of the proposed mission-planning methodology,

Table 1 Task start and end times for executing the mission in Fig. 12

Task	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}
Start time (s)	36	119	234	324	16	51	166	116	264	296
End time (s)	51	214	244	334	26	61	176	126	279	311

over the competing sequential methodology, is maintained even in the face of potential real-time uncertainties.

5.5 Effect of Mission Parameters on Mission-Completion Time

Mission parameters, including number of robots in the swarm, positions of tasks, number of robots required per task, and task working times affect mission-completion times. A brief discussion on their effect is provided below:

Number of swarm robots: An increase in the number of robots, n_R, would provide increased flexibility to the division-of-labor stage of the proposed mission-planning methodology. Thus, allowing the number of worker and support robots to be selected with higher optimally. This would, in turn, lead to reduced mission-completion times.

Number of worker robots: An increase in the necessary number of worker robots required per task, $\left\{n_{WV_i}\right\}_{i=1}^{n_V}$, would limit the search space of the division-of-labor stage. Thus, affecting the optimality of division of tasks and resulting in increased mission-completion times.

Task positions: Changes in the positions of the mission's tasks, $\left\{{}^{G}\boldsymbol{x}_{V_i}\right\}_{i=1}^{n_V}$, would result in changes in robot travel times from one task location to another. Thus, resulting in different mission-completion times.

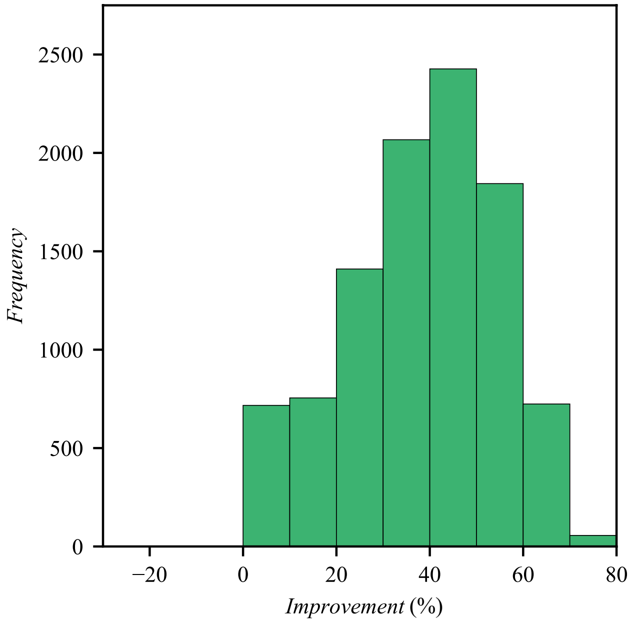


Fig. 13 Improvement in mission-completion time – concurrent versus sequential optimization

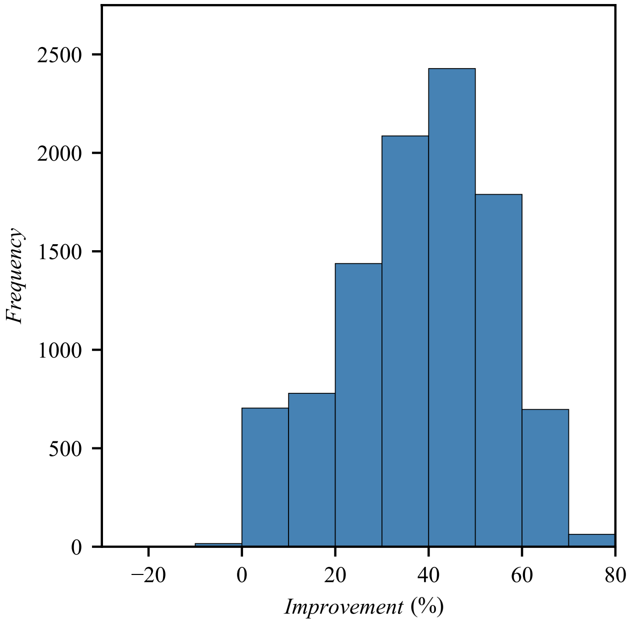


Fig. 14 Improvement in mission-completion time – concurrent versus sequential optimization for uncertain task working times

Working times: An increase in the working times of tasks, $\left\{t_{WV_i}\right\}_{i=1}^{n_V}$, would require the worker robots to spend more time for the accomplishment of each task. Thus, yielding increased mission-completion times.

6 Conclusions

This paper presents a mission-planning methodology for swarm robotic systems comprising robots with localization limitations that are constrained to using collaborative motion-control strategies. The proposed mission-planning methodology considers this constraint by dividing the swarm into two functionally separate groups: worker robots that accomplish the tasks at hand, and supporter robots that provide the necessary interactions for the worker robots to use the adopted motion-control strategy. Based on this division, the proposed methodology searches for the optimal division-of-labor of the robots to each role, task-allocation of the worker robots, worker robot paths, movement-concurrency, and movement-allocation. The search is completed through an optimization methodology that concurrently varies all the variables to optimize the mission-execution performance.

The proposed methodology is novel as it incorporates the division-of-labor into the mission-planning problem by searching for the optimal number of robots that should operate in a worker and support role. This contrasts past approaches that assume the swarm's division is given, and do not consider finding the optimal balance between the two roles in the swarm. The proposed methodology is also novel as it plans the paths of the worker robots to allow the support robots to concurrently facilitate multiple worker robot movements, achieving enhanced efficiency in mission execution. This contrasts past approaches that limit support robots to facilitating one movement/job at a time. Finally, the proposed methodology can be applied to any adopted swarm motion-control strategy that requires the worker robots to collaborate with support robots, including leader-based and tether-based ones. In contrast, past approaches that are developed for swarm with limited onboard localization capabilities are only suitable for leader-based methods. This novelty enhances the adaptability of the proposed planning methodology.

The proposed mission-planning methodology also includes a pre-implementation estimator that estimates the improvement in mission execution performance achieved through the concurrent planning methodology versus the competing sequential methodology. The estimate obtained is used to justify the additional computational requirements of the concurrent methodology. In contrast, competing approaches to mission-planning assume that a concurrent solution is always beneficial, and may spend computational resources without obtaining tangible improvements.

The details of proposed methodology were illustrated through multiple simulated experiments with swarms of up to one hundred robots. Furthermore, simulations on over 10,000 missions illustrated that the proposed concurrent methodology achieves approximately 38% improvement in mission execution performance compared to its sequential counterpart.

Future work may consider extending the proposed mission-planning methodology to different task models that require various types of worker robots [63], or a combination of sensors and actuators that must be present on the worker robots for the task to be accomplished [54]. These models require an extension of the division-of-labor in the proposed mission-planning methodology to include optimal subdivision of the worker robots based on their type and/or onboard sensing and actuation capabilities. Task models that require a specific workload, which can be achieved in a short time with many worker robots or over a longer period with few robots, may also be considered [42]. For these models, mission-planning must also optimize the number of worker robots allocated to each task. Furthermore, while the proposed pre-implementation estimator can be used to ensure that the used computational resources yield sufficient improvements in the mission-completion time, the computational complexity of the proposed mission-planning methodology may be restrictive. Thus, future work could further investigate the suitability of various optimization search engines for the problem at hand. Future work may also consist of incorporating the uncertainties that affect the swarm during mission execution in the mission planning problem.

Appendix 1: A Descriptive Example

The implementation of the proposed swarm mission-planning methodology is detailed herein for a mission comprising $n_V = 5$ tasks. The position of the tasks, as well as their number of required worker robots and task working times are shown in Fig. 15. The swarm at hand comprises $n_R = 50$ (homogeneous) robots, each of which can operate in either a worker or support role. Furthermore, the environment includes one connectivity point at home base that is used to provide connectivity for the adopted

	n_{WV_i}	t_{WV_i}
V_1	30	10
V_2	25	15
V_3	30	10
V_4	20	10
V_5	15	10

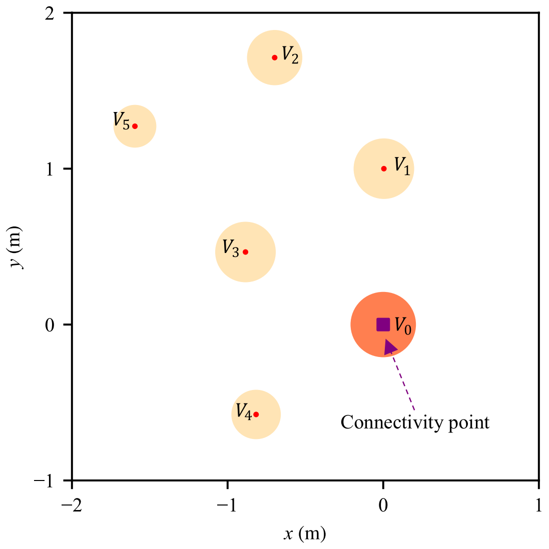


Fig. 15 A five-task swarm mission

tether-based motion control strategy. The simplicity of the example allows us to present the optimal solutions in graphical form.

Estimating the Improvement in Mission-completion time

The proposed MLP detailed in Section 4.1, and trained through the process detailed in Section 5.1, was applied to the mission at hand, and used to decide whether a concurrent planning methodology would be necessary. In this example, the user defined minimum improvement was set as 30%. Since the estimation model estimated an improvement of about 32% for this mission, the concurrent planning methodology was deemed beneficial, and applied to plan the mission of the worker and support robots.

Division-of-Labor

For the mission at hand, the minimum (total) number of necessary worker robots is $n_{Wmin} = 30$, calculated as the maximum number of robots required for any one of the tasks. The minimum (total) number of required support robots is $n_{Smin} = 10$, based on the length of the longest tether that would be used for swarm motion-control. This leaves ten (=50-30-10) swarm robots whose role can be selected to minimize the mission-completion time.

In this example, the proposed mission-planning methodology selected $n^*_W = 35$ worker robots, leaving $n^*_S = 15$ support robots for facilitating the workers' motion.

Task-Allocation and Path-Planning for Worker Robots

The mission of the worker robots is planned by determining the optimal formation of the coalitions for the tasks at hand. It also involves determining the optimal paths that the sub-coalitions take to reach their destinations.

The proposed mission-planning methodology allocated the $n^*_W = 35$ worker robots to the task at hand through:

$$\boldsymbol{A}^* = \begin{bmatrix} 0 & 0 & 0 & 30 & 0 & 5 \\ 0 & 0 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 10 \\ 0 & 20 & 0 & 0 & 0 & 0 \\ 0 & 10 & 5 & 0 & 0 & 0 \end{bmatrix}. \tag{43}$$

The optimal task-allocation is shown graphically in Fig. 16.

For the optimal task-allocation shown in Fig. 16, the optimal paths of the $n_C = 8$ sub-coalitions were determined as:

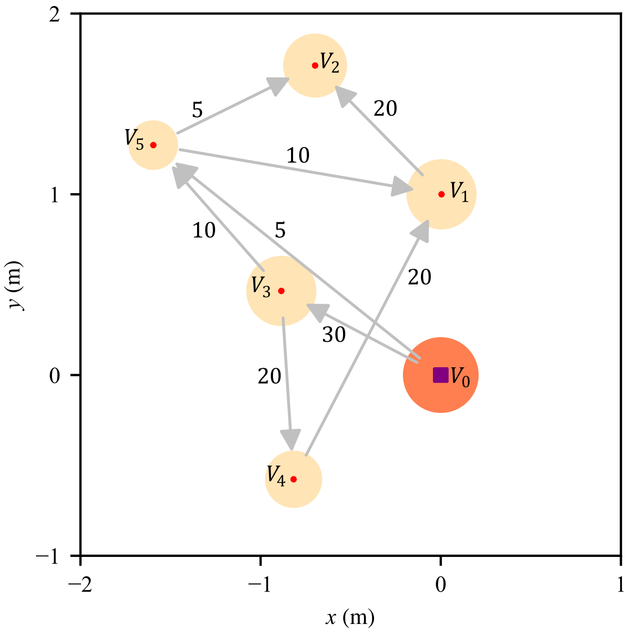


Fig. 16 Optimal task-allocation of the worker robots for the mission in Fig. 15

$$\begin{aligned} p^*_{pc_{03}} &= \{V_0, V_3\}, \\ p^*_{pc_{05}} &= \{V_0, V_3, V_5\}, \\ p^*_{pc_{12}} &= \{V_1, V_2\}, \\ p^*_{pc_{34}} &= \{V_3, V_4\}, \\ p^*_{pc_{35}} &= \{V_3, V_5\}, \\ p^*_{pc_{41}} &= \{V_4, V_1\}, \\ p^*_{pc_{51}} &= \{V_5, V_1\}, \text{ and} \\ p^*_{pc_{52}} &= \{V_5, V_1, V_2\}. \end{aligned} \tag{44}$$

These paths are shown graphically in Fig. 17, where each distinct color represents the path of a sub-coalition.

Movement-Concurrency, Movement-Allocation, and Path-Planning for Support Robots

Mission-planning for the support robots, in turn, involves optimizing the concurrent execution of the movements at hand and the allocation of the support robots to these movements. This is, then, followed by optimal path planning for the support robots based on the adopted tether-based swarm motion-control strategy.

As was shown through Fig. 17 above, there are four path segments that share the same pick-up and drop-off task locations. These four segments require the worker robots to be (1) picked-up at V_0 and dropped-off at V_3, (2) picked-up at V_3 and dropped-off at V_5, (3) picked-up

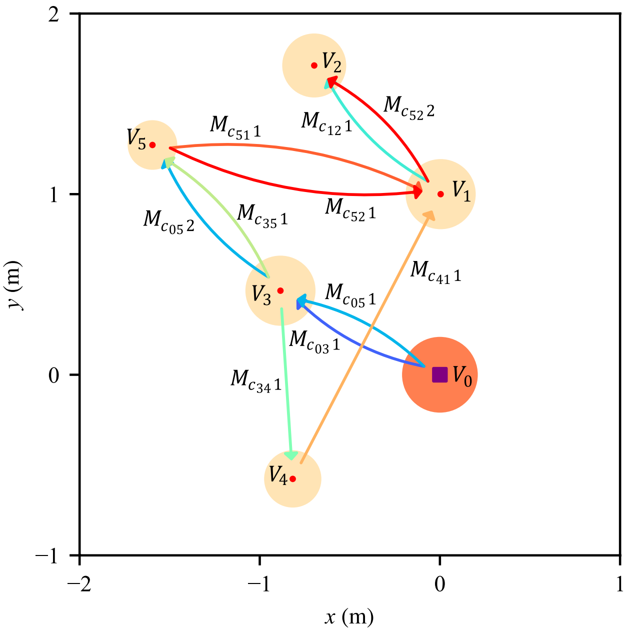


Fig. 17 Optimal sub-coalition paths for the optimal task-allocation shown in Fig. 16

at V_5 and dropped-off at V_1, and (4) picked-up at V_1 and dropped-off at V_2. The plan determined that all movements associated with these four segments should be facilitated concurrently – $M_{c_{03}}1$ and $M_{c_{05}}1$, $M_{c_{05}}2$ and $M_{c_{35}}1$, $M_{c_{52}}1$ and $M_{c_{51}}1$, and $M_{c_{52}}2$ and $M_{c_{12}}1$ were planned for concurrent execution, Eq. (45), where the movements are ordered along the rows (left to right) and columns (top to bottom) as: M_0, $M_{c_{03}}1$, $M_{c_{05}}1$, $M_{c_{05}}2$, $M_{c_{12}}1$, $M_{c_{34}}1$, $M_{c_{35}}1$, $M_{c_{41}}1$, $M_{c_{51}}1$, $M_{c_{52}}1$, $M_{c_{52}}2$.

$$\boldsymbol{Q}^* = \begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&1\\ 0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&0&0&1&1&0\\ 0&0&0&0&1&0&0&0&0&0&1 \end{bmatrix}. \tag{45}$$

For the optimal movement-concurrency detailed above, the optimal allocation of the $n_S^* = 15$ support robots to the movements at hand was determined as Eq. (46), where the movements are ordered along the rows and columns as detailed above.

$$\boldsymbol{S}^* = \begin{bmatrix} 0&0&5&0&0&0&10&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&5&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&5&0&0&0\\ 0&0&0&0&0&0&0&0&0&5&5\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&5\\ 0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}. \tag{46}$$

The optimal movement-concurrency and the optimal movement-allocation are shown graphically in Fig. 18, where the nodes with multiple movements represent movements that are planned for concurrent facilitation.

As the final step, the paths of the support robots were planned to allow for collaboration with the worker robots through the tether-based motion control strategy, where the paths of the support robots are planned to form the shortest tethers for connecting the worker robots to the connectivity point at home base. Figure 19 illustrates example tethers formed for the connectivity of the swarm to the home base.

Robot Trajectory-Planning

As the final step to mission-planning, the trajectories of all (worker and support) robots were planned based on the execution rules detailed in Section 4.5. These rules allow the

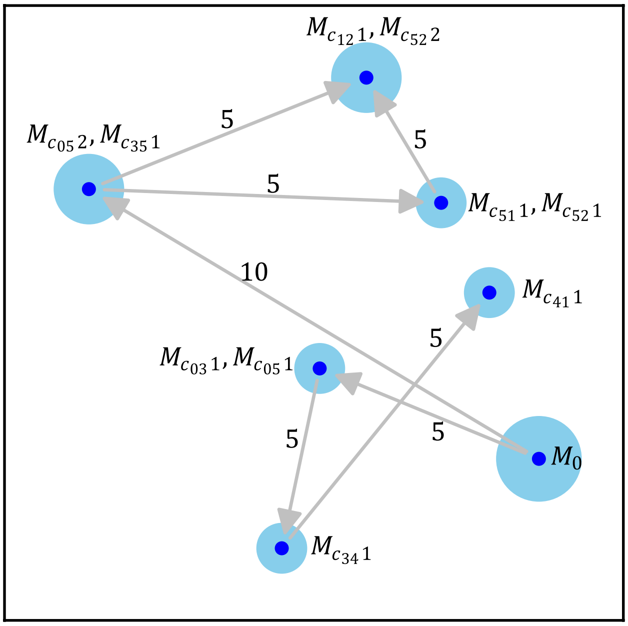

$M_{c_{12}}1, M_{c_{52}}2$
$M_{c_{05}}2, M_{c_{35}}1$
$M_{c_{51}}1, M_{c_{52}}1$
$M_{c_{41}}1$
$M_{c_{03}}1, M_{c_{05}}1$
M_0
$M_{c_{34}}1$
5
5
5
10
5
5
5

Fig. 18 Optimal movement-concurrency and allocation for the worker paths shown in Fig. 17

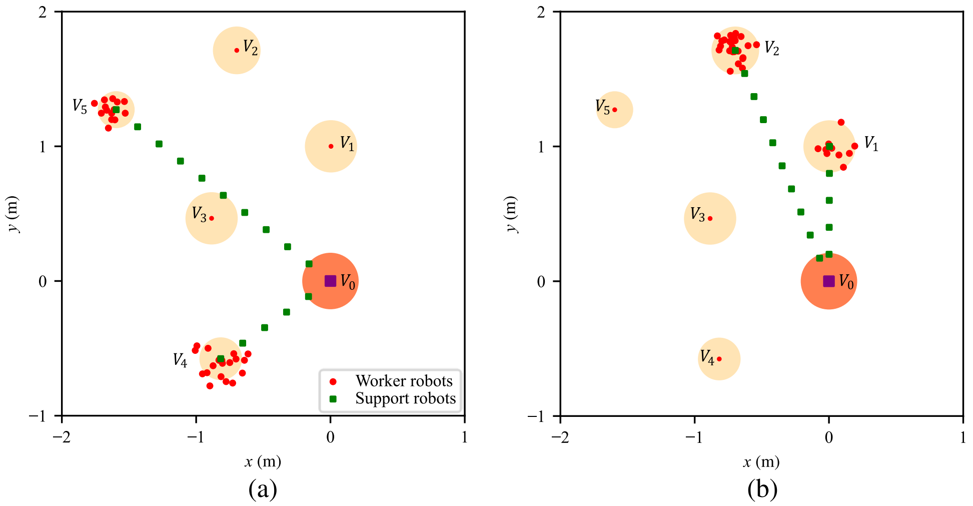


Fig. 19 Tethers for connectivity to the worker robots at (*a*) tasks V_4 and V_5, and (*b*) tasks V_1 and V_2

Table 2 Task start and end times for executing the mission in Fig. 15

Task	V_1	V_2	V_3	V_4	V_5
Start time (s)	97	127	20	51	52
End time (s)	107	142	30	61	62

swarm robots to follow their planned paths, and to synchronize their motion for effective collaboration.

For the optimal division-of-labor and the optimal mission plan of the worker and support robots, the planned trajectories achieved the task completion times shown in Table 2 below. In this example, the optimal mission-completion time, as found through the proposed concurrent methodology, was determined as $t_{CC} = 142s$.

In contrast, the application of the sequential methodology resulted in a mission-completion time of $t_{CS} = 217s$. This indicates an improvement of 35% for the proposed concurrent solution over the sequential one. As expected through the estimated improvement in Section 5.2.1, this is above the minimum user defined threshold. An animation of the mission execution for the plans found through the proposed concurrent and competing sequential planning methodologies can be found at https://youtu.be/gmM8D8FWOc0.

For an in-depth analysis, let us examine the motion of the sub-coalition of 5 worker robots that leave task V_5 and are allocated to task V_2. This sub-coalition is represented as c_{52}, and is shown through a_{52} of the optimal task-allocation solution, $\boldsymbol{A}^*$, Eq. (43), Fig. 16. The path of this sub-coalition was selected to visit task V_1 when moving from V_5 to V_2, $p^*_{pc_{52}} = \{V_5, V_1, V_2\}$, Eq. (44). This path is completed in two movements: $M_{c_{52}1}$ and $M_{c_{52}2}$, respectively.

Movement $M_{c_{52}1}$ was selected to be executed concurrently with the movement of sub-coalition c_{51} from V_5 to V_1, $M_{c_{51}1}$, Eq. (45), Fig. 18. These two movements begin as soon as V_5 is completed at $t = 62s$, and the associated workers, part of sub-coalitions c_{52} and c_{51}, arrive at V_1 at $t = 95s$. The movements require five support robots. The path of these support robots as they concurrently facilitate movements $M_{c_{52}1}$ and $M_{c_{51}1}$ is shown in Fig. 20(a).

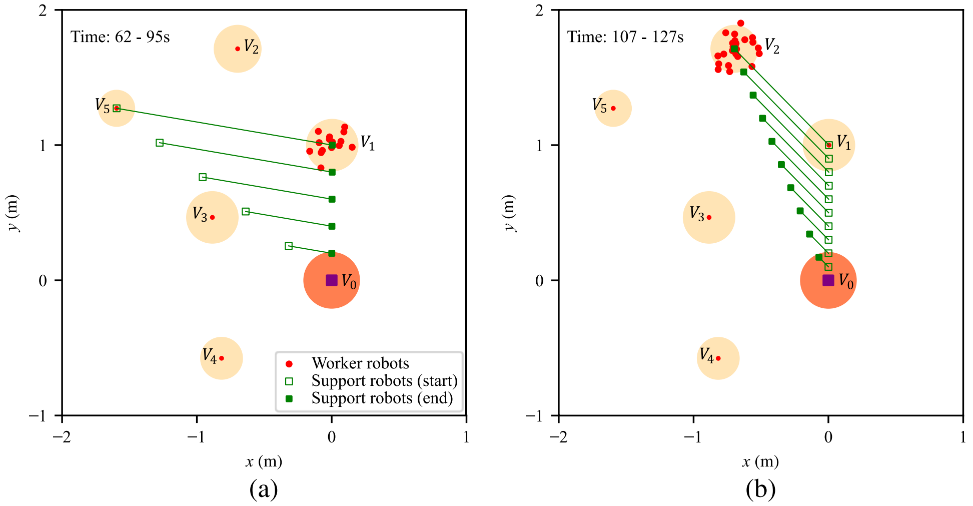


Fig. 20 Path of the support robots as they execute movements (*a*) $M_{c_{52}1}$ and $M_{c_{51}1}$ concurrently, and (*b*) $M_{c_{52}2}$ and $M_{c_{12}1}$ concurrently. The final position of the worker robots for the associated sub-coalitions, c_{52} and c_{51}, and sub-coalitions c_{52} and c_{12}, respectively, are also shown

The second movement of sub-coalition c_{52}, $M_{c_{52}2}$, is planned to be concurrently executed with the movement of sub-coalition c_{12} from V_1 to V_2, $M_{c_{12}1}$, Eq. (45), Fig. 18. Thus, the worker robots of c_{52} must wait until V_1 is completed at $t = 107s$, before they can begin moving to their next task, V_2. Movements $M_{c_{52}2}$ and $M_{c_{12}1}$ begin at $t = 107s$, and the associated worker robots reach task V_2 at $t = 127s$. The movements require 10 support robots. The path of these support robots used to concurrently facilitate these two movements is shown in Fig. 20(b).

Acknowledgements This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Canada Research Chairs program (CRC).

Author Contributions Conceptualization: K.E. and B.B.; Material preparation and data collection: K.E., Analysis: K.E., G.N., B.B., Writing – original draft preparation: K.E.; Writing—review and editing: K.E., G.N., and B.B.; Funding acquisition: G.N. and B.B.; Supervision: G.N. and B.B. All authors read and approved the final manuscript.

Funding This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), NSERC CRD, and the Canada Research Chairs program (CRC).

Declarations

Ethics Approval Not Applicable.

Consent to Participate Not Applicable.

Consent to Publish Not Applicable.

Competing Interests Author Beno Benhabib is an Executive-Board Member for the Journal of Intelligent and Robotic Systems. The authors have no relevant funding, employment, financial interests to disclose.

References

1. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm robotics: A review from the swarm engineering perspective. Swarm Intell. **7**(1), 1–41 (2013)
2. Şahin, E.: Swarm robotics: From sources of inspiration to domains of application. In: Şahin, E., Spears, W.M. (eds.) Swarm Robotics, **3342**, pp. 10–20. Springer, Berlin, Heidelberg (2004)
3. Schranz, M., Umlauft, M., Sende, M., Elmenreich, W.: Swarm robotic behaviors and current applications. Front. Robot. AI **7**, 36 (2020)
4. Macwan, A., Nejat, G., Benhabib, B.: Target-motion prediction for robotic search and rescue in wilderness environments. IEEE Trans. Syst. Man Cybern. Part B. Cybern. **41**(5), 1287–1298 (2011)
5. Bakhtari, A., Naish, M.D., Eskandari, M., Croft, E.A., Benhabib, B.: Active-vision-based multisensor surveillance - An implementation. IEEE Trans. Syst. Man Cybern. Part C. Appl. Rev. **36**(5), 668–680 (2006)
6. Eschke, C., Heinrich, M. K., Wahby, M., and Haman, H.: Self-organized adaptive paths in multi-robot manufacturing: Reconfigurable and pattern-independent fibre deployment. In: IEEE/RSJ Int. Conf. Intel. Robot. Syst., pp. 4086–4091 (2019)
7. Nunes, E., Manner, M., Mitiche, H., Gini, M.: A taxonomy for task allocation problems with temporal and ordering constraints. Robot. Auton. Syst. **90**, 55–70 (2017)
8. Gerkey, B.P., Matarić, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. Int. J. Robot. Res. **23**(9), 939–954 (2004)
9. Koes, M., Nourbakhsh, I., and Sycara, K.: Heterogeneous multi-robot coordination with spatial and temporal constraints. In: Proc. Nat. Conf. Art. Intel. **3**, pp. 1292–1297 (2005)
10. Yao, W., Qi, N., Liu, Y., Xu, S., Du, D.: Homotopic approach for robot allocation optimization coupled with path constraints. IEEE Robot. Autom. Lett. **5**(1), 88–95 (2020)
11. Motes, J., Sandström, R., Lee, H., Thomas, S., Amato, N.M.: Multi-robot task and motion planning with subtask dependencies. IEEE Robot. Autom. Lett. **5**(2), 3338–3345 (2020)
12. Henkel, C., Abbenseth, J., and Toussaint, M.: An optimal algorithm to solve the combined task allocation and path finding problem. In: IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 4140–4146 (2019)
13. Banfi, J., Messing, A., Kroninger, C., Stump, E., Hutchinson, S., Roy, N.: Hierarchical planning for heterogeneous multi-robot routing problems via learned subteam performance. IEEE Robot. Autom. Lett. **7**(2), 4464–4471 (2022)
14. Messing, A., Neville, G., Chernova, S., Hutchinson, S., Ravichandar, H.: GRSTAPS: Graphically recursive simultaneous task allocation, planning, and scheduling. Int. J. Robot. Res. **41**(2), 232–256 (2022)
15. Jones, E.G., Dias, M.B., Stentz, A.: Time-extended multi-robot coordination for domains with intra-path constraints. Auton. Robots **30**(1), 41–56 (2011)
16. Jones, E. G., Dias, M. B., and Stentz, A.: Tiered auctions for multi-agent coordination in domains with precedence constraints: In Proc. 26th Army Sci. Conf., pp. 1–8 (2008)
17. Parker, L.E., Tang, F.: Building multirobot coalitions through automated task solution synthesis. Proc. IEEE **94**(7), 1289–1305 (2006)
18. Zhang, Y. and Parker, L. E.: IQ-ASyMTRe: Synthesizing coalition formation and execution for tightly-coupled multirobot tasks. In: IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 5595–5602 (2010)
19. Zhang, Y., Parker, L.E.: IQ-ASyMTRe: Forming executable coalitions for tightly coupled multirobot tasks. IEEE Trans. Robot. **29**(2), 400–416 (2013)
20. Tang, F. and Parker, L. E.: A Complete Methodology for Generating Multi-Robot Task Solutions using ASyMTRe-D and Market-Based Task Allocation. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 3351–3358 (2007)
21. Korsah, G.A., Stentz, A., Dias, M.B.: A comprehensive taxonomy for multi-robot task allocation. Int. J. Robot. Res. **32**(12), 1495–1512 (2013)
22. Choi, H.-L., Brunet, L., How, J.P.: Consensus-based decentralized auctions for robust task allocation. IEEE Trans. Robot. **25**(4), 912–926 (2009)
23. Nayak, S., Yeotikar, S., Carrillo, E., Rudnick-Cohen, E., Jaffar, M.K.M., Patel, R., Azarm, S., Herrmann, J.W., Xu, H., Otte, M.: Experimental comparison of decentralized task allocation algorithms under imperfect communication. IEEE Robot. Autom. Lett. **5**(2), 572–579 (2020)
24. Khamis, A.M., Elmogy, A.M., Karray, F.O.: Complex task allocation in mobile surveillance systems. J. Intell. Robot. Syst. **64**(1), 33–55 (2011)
25. Ansari, I., Mohamed, A., Flushing, E. F., and Razak, S.: Cooperative and load-balancing auctions for heterogeneous multi-robot teams dealing with spatial and non-atomic tasks. In: IEEE Int. Symp. Safety, Sec., Resc. Robot., pp. 213–220 (2020)
26. Jones, E. G., Bernardine Dias, M., and Stentz, A.: Learning-enhanced market-based task allocation for oversubscribed domains. In: IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 2308–2313 (2007)

27. Berman, S., Halasz, A., Hsieh, M.A., Kumar, V.: Optimized stochastic policies for task allocation in swarms of robots. IEEE Trans. Robot. **25**(4), 927–937 (2009)
28. Lee, W., Vaughan, N., Kim, D.: Task allocation into a foraging task with a series of subtasks in swarm robotic system. IEEE Access **8**, 107549–107561 (2020)
29. Pang, B., Song, Y., Zhang, C., Wang, H., Yang, R.: Autonomous task allocation in a swarm of foraging robots: An approach based on response threshold sigmoid model. Int. J. Control Autom. Syst. **17**(4), 1031–1040 (2019)
30. De Lope, J., Maravall, D., Quiñonez, Y.: Response threshold models and stochastic learning automata for self-coordination of heterogeneous multi-task distribution in multi-robot systems. Robot. Auton. Syst. **61**(7), 714–720 (2013)
31. Mayya, S., Wilson, S., Egerstedt, M.: Closed-loop task allocation in robot swarms using inter-robot encounters. Swarm Intell. **13**(2), 115–143 (2019)
32. Jevtic, A., Gutierrez, Á., Andina, D., Jamshidi, M.: Distributed bees algorithm for task allocation in swarm of robots. IEEE Syst. J. **6**(2), 296–304 (2012)
33. Jha, D. K.: Algorithms for task allocation in homogeneous swarm of robots. In: IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 3771–3776 (2018)
34. Jang, I., Shin, H.-S., Tsourdos, A.: Anonymous hedonic game for task allocation in a large-scale multiple agent system. IEEE Trans. Robot. **34**(6), 1534–1548 (2018)
35. Mazdin, P., Rinner, B.: Distributed and communication-aware coalition formation and task assignment in multi-robot systems. IEEE Access **9**, 35088–35100 (2021)
36. Nedjah, N., de Mendonça, R.M., de Macedo Mourelle, L.: PSO-based distributed algorithm for dynamic task allocation in a robotic swarm. Procedia Comput. Sci. **51**, 326–335 (2015)
37. Dutta, A. and Asaithambi, A.: One-to-many bipartite matching based coalition formation for multi-robot task allocation. In: IEEE Int. Conf. Robot. Autom., pp. 2181–2187 (2019)
38. Chen, J., Yan, X., Chen, H., and Sun, D.: Resource constrained multirobot task allocation with a leader-follower coalition method. In: IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 5093–5098 (2010)
39. Chandarana, M., Hughes, D., Lewis, M., Sycara, K., and Scherer, S.: Hybrid model for a priori performance prediction of multi-job type swarm search and service missions. In: Int. Conf. Adv. Robot., pp. 714–719 (2019)
40. Chandarana, M., Lewis, M., Sycara, K., and Scherer, S.: Determining effective swarm sizes for multi-job type missions. In: IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 4848–4853 (2018)
41. Luna, M. A., Refaat Ragab, A., Ale Isac, M. S., Flores Peña, P., and Cervera, P. C.: A new algorithm using hybrid UAV swarm control system for firefighting dynamical task allocation. In: IEEE Int. Conf. Syst. Man Cyber., pp. 655–660 (2021)
42. Capezzuto, L., Tarapore, D., and Ramchurn, S. D.: Large-scale, dynamic and distributed coalition formation with spatial and temporal constraints. In: Proc. European Conf. Mult. Agent Syst., pp. 108–125 (2021)
43. Hsieh, M.A., Halász, Á., Berman, S., Kumar, V.: Biologically inspired redistribution of a swarm of robots among multiple sites. Swarm Intell. **2**(2), 121–141 (2008)
44. Dutta, A., Ufimtsev, V., and Asaithambi, A.: Correlation clustering based coalition formation for multi-robot task allocation. In: Proc. ACM/SIGAPP Symp. Appl. Comp., pp. 906–913 (2019)
45. Dutta, A., Ufimtsev, V., Said, T., Jang, I., and Eggen, R.: Distributed hedonic coalition formation for multi-robot task allocation. In: IEEE Int. Conf. Autom. Sci. Eng., pp. 639–644 (2021)
46. Autenrieb, J., Strawa, N., Shin, H.-S., and Hong, J.-H.: A mission planning and task allocation framework for multi-UAV swarm coordination. In: Work. Res. Ed. Dev. Unman. Aer. Syst., pp. 297–304 (2019)
47. Atay, N. and Bayazit, B.: Mixed-integer linear programming solution to multi-robot task allocation problem. All Comput. Sci. Eng. Res. Report Number: WUCSE-2006–54, (2006)
48. Sheridan, P.K., Gluck, E., Guan, Q., Pickles, T., Balcıog, B., Benhabib, B.: The dynamic nearest neighbor policy for the multi-vehicle pick-up and delivery problem. Transp. Res. Part Policy Pract. **49**, 178–194 (2013)
49. Notomista, G., Mayya, S., Emam, Y., Kroninger, C., Bohannon, A., Hutchinson, S., Egerstedt, M.: A resilient and energy-aware task allocation framework for heterogeneous multirobot systems. IEEE Trans. Robot. **38**(1), 159–179 (2022)
50. Mayya, S., D'antonio, D.S., Saldaña, D., Kumar, V.: Resilient task allocation in heterogeneous multi-robot systems. IEEE Robot. Autom. Lett. **6**(2), 1327–1334 (2021)
51. Viguria, A., Maza, I., and Ollero, A.: S+T: An algorithm for distributed multirobot task allocation based on services for improving robot cooperation. In: IEEE Int. Conf. Robot. Autom., pp. 3163–3168 (2008)
52. Croft, E.A., Benhabib, B., Fenton, R.G.: Near-time optimal robot motion planning for on-line applications. J. Robot. Syst. **12**(8), 553–567 (1995)
53. Irfan, M. and Farooq, A.: Auction-based task allocation scheme for dynamic coalition formations in limited robotic swarms with heterogeneous capabilities. In: Int. Conf. Intell. System. Eng., pp. 210–215 (2016)
54. Vig, L., Adams, J.A.: Multi-robot coalition formation. IEEE Trans. Robot. **22**(4), 637–649 (2006)
55. Nam, C., Shell, D.A.: Robots in the huddle: Upfront computation to reduce global communication at run time in multirobot task allocation. IEEE Trans. Robot. **36**(1), 125–141 (2020)
56. Emam, Y., Mayya, S., Notomista, G., Bohannon, A., and Egerstedt, M.: Adaptive Task Allocation for Heterogeneous Multi-Robot Teams with Evolving and Unknown Robot Capabilities. In: IEEE Int. Conf. Robot. Autom., pp. 7719–7725 (2020)
57. Borg, J.M., Mehrandezh, M., Fenton, R.G., Benhabib, B.: Navigation-guidance-based robotic interception of moving objects in industrial settings. J. Intell. Robot. Syst. **33**(1), 1–23 (2002)
58. Hujic, D., Croft, E.A., Zak, G., Fenton, R.G., Mills, J.K., Benhabib, B.: The robotic interception of moving objects in industrial settings: strategy development and experiment. IEEE/ASME Trans. Mechatron. **3**(3), 225–239 (1998)
59. Luo, L., Chakraborty, N., Sycara, K.: Provably-good distributed algorithm for constrained multi-robot task assignment for grouped tasks. IEEE Trans. Robot. **31**(1), 19–30 (2015)
60. Wu, D., Zeng, G., Meng, L., Zhou, W., Li, L.: Gini coefficient-based task allocation for multi-robot systems with limited energy resources. J. Autom. Sin. **5**(1), 155–168 (2018)
61. Korsah, G. A., Kannan, B., Browning, B., Stentz, A., and Dias, M. B.: xBots: An approach to generating and executing optimal multi-robot plans with cross-schedule dependencies. In: IEEE Int. Conf. Robot. Autom., pp. 115–122 (2012)
62. Lemaire, T., Alami, R., and Lacroix, S.: A distributed tasks allocation scheme in multi-UAV context. In: IEEE Int. Conf. Robot. Autom., **4**, pp. 3622- 3627 (2004)
63. Suslova, E. and Fazli, P.: Multi-robot task allocation with time window and ordering constraints. In: IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 6909–6916 (2020)
64. Ayari, E., Hadouaj, S., and Ghedira, K.: A dynamic decentralised coalition formation approach for task allocation under tasks priority constraints. In: Int. Conf. Adv. Robot., pp. 250–255 (2017)
65. Mouradian, C., Sahoo, J., Glitho, R. H., Morrow, M. J., and Polakos, P. A.: A coalition formation algorithm for multi-robot task allocation in large-scale natural disasters. In: Int. Wireless Comm. Mob. Comp. Conf., pp. 1909–1914 (2017)

66. Kim, J.Y., Kashino, Z., Colaco, T., Nejat, G., Benhabib, B.: Design and implementation of a millirobot for swarm studies – mROBerTO. Robotica **36**(11), 1591–1612 (2018)
67. Kim, J. Y., Colaco, T., Kashino, Z., Nejat, G., and Benhabib, B.: mROBerTO: A modular millirobot for swarm-behavior studies. In: IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 2109–2114 (2016)
68. Eshaghi, K., Li, Y., Kashino, Z., Nejat, G., Benhabib, B.: mROBerTO 2 0 – An autonomous millirobot with enhanced locomotion for swarm robotics. Robot. Autom. Let **5**(2), 962–969 (2020)
69. Pickem, D., Lee, M., and Egerstedt, M.: The GRITSBot in its natural habitat - A multi-robot testbed. In: IEEE Int. Conf. Robot. Autom., pp. 4062–4067 (2015)
70. Rubenstein, M., Ahler, C., and Nagpal, R.: Kilobot: A low cost scalable robot system for collective behaviors. In: IEEE Int. Conf. Robot. Autom., pp. 3293–3298 (2012)
71. Sabelhaus, A. P., Mirsky, D., Hill, L. M., Martins, N. C., and Bergbreiter, S.: TinyTeRP: A tiny terrestrial robotic platform with modular sensing. In: IEEE Int. Conf. Robot. Autom., pp. 2600–2605 (2013)
72. Arvin, F., Murray, J., Zhang, C., Yue, S.: Colias: An autonomous micro robot for swarm robotic applications. Int. J. Adv. Robot. Syst. **11**(7), 1–10 (2014)
73. Kim, J.Y., Kashino, Z., Pineros, L.M., Bayat, S., Colaco, T., Nejat, G., Benhabib, B.: A high-performance millirobot for swarm-behaviour studies: Swarm-topology estimation. Int. J. Adv. Robot. Syst. **16**(6), 1–18 (2019)
74. Pires, A. G., Macharet, D. G., and Chaimowicz, L.: Exploring heterogeneity for cooperative localization in swarm robotics. In: Int. Conf. Adv. Robot., pp. 407–414 (2015)
75. Li, W., Xiong, Z., Sun, Y., and Xiong, J.: Cooperative positioning algorithm of swarm UAVs based on posterior linearization belief propagation. In: IEEE Inf. Tech. Network. Electron. Autom. Conf. Conf., pp. 1277–1282 (2019)
76. Song, Z. and Mohseni, K.: A distributed localization hierarchy for an AUV swarm. In: American Conf. Conf., pp. 4721–4726 (2014).
77. Loefgren, I., Ahmed, N., Frew, E., Heckman, C., and Humbert, S.: Scalable event-triggered data fusion for autonomous cooperative swarm localization. In: Int. Conf. Inf. Fusion, pp. 1–8 (2019)
78. Yoon, H.J., Eshaghi, K., Nejat, G., Benhabib, B.: Localization and topology estimation of robot swarms using Kalman filters. J. Inst. Control Robot. Syst. **28**(6), 622–631 (2022)
79. Cornejo, A., Nagpal, R.: Distributed range-based relative localization of robot swarms. In: Akin, H.L., Amato, N.M., Isler, V., van der Stappen, A.F. (eds.) Algorithmic Foundations of Robotics XI, **107**, pp. 91–107. Springer International Publishing, Cham, Switzerland (2015)
80. Eshaghi, K., Kashino, Z., Yoon, H.J., Nejat, G., Benhabib, B.: An inchworm-inspired motion strategy for robotic swarms. Robotica **39**(12), 2283–2305 (2021)
81. Eshaghi, K., Rogers, A., Nejat, G., Benhabib, B.: Closed-loop motion control of robotic swarms – A tether-based strategy. IEEE Trans. Robot. **38**(6), 3564–3581 (2022)
82. Tutsoy, O., Barkana, D.E., Balikci, K.: A novel exploration-exploitation-based adaptive law for intelligent model-free control approaches. IEEE Trans. Cybern. **53**(1), 329–337 (2023)
83. Tutsoy, O.: COVID-19 epidemic and opening of the schools: Artificial intelligence-based long-term adaptive policy making to control the pandemic diseases. IEEE Access **9**, 68461–68471 (2021)
84. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd ed. Upper Saddle River, NJ, United States: Prentice Hall PTR (1998)
85. Katoch, S., Chauhan, S.S., Kumar, V.: A review on genetic algorithm: past, present, and future. Multimed. Tools Appl. **80**(5), 8091–8126 (2021)
86. Mohamed, S.C., Fung, A., Nejat, G.: A multirobot person search system for finding multiple dynamic users in human-centered environments. IEEE Trans. Cybern. **53**(1), 628–640 (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Kasra Eshaghi received his B.A.Sc. degree in Mechanical Engineering in 2018 from the University of Toronto. He is currently a Ph.D. candidate at the University of Toronto. His research interests include software and hardware design for robotic swarms.

Goldie Nejat is the Canada Research Chair in Robots for Society and a Professor in the Department of Mechanical and Industrial Engineering at the University of Toronto. She is the Founder and Director of the ASBLab. She is also a Fellow of the Canadian Institute for Advanced Research. Her research interests include intelligent assistive/service robots, human-robot interactions, and semiautonomous/autonomous control. She received her B.A.Sc. and Ph.D. degrees in Mechanical Engineering at the University of Toronto.

Beno Benhabib received his B.Sc., M.Sc., and Ph.D. degrees in Mechanical Engineering in 1980, 1982, and 1985, respectively. He has been a Professor with the Department of Mechanical and Industrial Engineering at the University of Toronto since 1986. His current research interests include the design and control of intelligent autonomous systems.