# A mathematical programming approach to collaborative missions with heterogeneous teams

**Conference Paper** · September 2014

DOI: 10.1109/IROS.2014.6942590



**3 authors:**

Eduardo Feo Flushing
Carnegie Mellon University Qatar


Luca Maria Gambardella
University of Applied Sciences and Arts of Southern Switzerland


Gianni A. Di Caro
Carnegie Mellon University

# A Mathematical Programming Approach to Collaborative Missions with Heterogeneous Teams

Eduardo Feo Flushing, Luca M. Gambardella, Gianni A. Di Caro

***Abstract*— We consider the problem of the joint mission planning in teams of heterogeneous physical agents. The type of missions that we consider are composed of spatially distributed tasks that need to be selected and assigned to the agents for dealing with them. A plan consists of a set of directives specifying who does what, where, when, and for how long. The aim is to optimize system-level performance by explicitly taking into account and exploiting the different sensory-motor characteristics of the agents. We tackle this general problem by proposing a mixed integer linear formulation of it which includes several aspects/constraints that closely model features and requests of realistic scenarios. We present a solution approach based on the combination of a metaheuristic and mathematical programming method that allows to compute high-quality plans within short time, and with formal guarantees on their optimality. The application of the framework is validated in the context of search and rescue missions.**

## I. INTRODUCTION

Teams composed by physical agents with different cognitive and sensory-motor skills (e.g., robots, humans, animals) naturally provide heterogeneity and redundancy of resources, parallelism, and distributedness. Moreover, they can be designed to produce effective synergies through agent cooperation. All these aspects makes heterogeneous teams highly suited for a number of important real-world problems, such as search and rescue, environmental monitoring, surveillance, and other similar problems which are spatially distributed and can profit from the presence of a diversity of skills.

In this paper, we study the problem of planning the activities of such heterogeneous systems. We deal with missions that are defined by a set of *spatially distributed tasks*. A *task* represents a *localized activity*: it is directly associated to a specific location (or portion) of the environment. For instance, when looking for a missing person in the wilderness inside a given an area $A$, a task might consist in *searching the person in a specific portion of $A$*. We assume that the initial set of tasks is given, together with a finite temporal horizon for team's operations. Then, given a team of heterogeneous agents, the problem consists in selecting which tasks to perform, assigning them to specific agents within the team, and appointing the duration and schedule of the services provided to each one of the selected tasks. The objective is to optimize team-level performance by explicitly taking into account the different agents' skills, their mutual interactions and dependencies, and potential synergies.

An important characterization on the type of tasks, is that we distinguish between *atomic* and *non-atomic* tasks. Atomic tasks are assigned as a whole, meaning that, once allocated, the effort put into it must be equal to that required for its completion. Furthermore, only two *states* can be associated to these tasks: *completed*, or *not completed*. Instead, a non-atomic task can be carried out in an incremental manner over disjoint periods of time, and therefore, at any time, these can be in a *partially completed* state. Moreover, this state can be the result from the service provided by an arbitrary number of agents, each contributing with possibly different levels of efficiency. Most works in this context are restricted to atomic tasks (e.g., [1], [2], [3]). Here, we consider more generically mission plans that can accommodate non-atomic tasks that can be *partially* or completely fulfilled through the service provided by *one or more agents during the course of the mission*. Since we are dealing with heterogeneous teams, the relationship between the allotted effort (i.e., devoted time) and the amount of service provided to a task by an agent depends on the matching between the characteristics of the task and the capabilities of the agent. Addressing these issues and optimizing this matching are some of the core challenges that we focus on this work.

We formalize this mission planning problem by means of a *mixed-integer linear programming* (MIP) formulation, The MIP formulation explicitly takes into account the intrinsically different skills of the team. It also provides means for granting robustness against deviations in the agents' response arising as a consequence of the presence of partially controllable agents, such as dogs, agents that have a strong autonomy, like humans, or agents that might incur in sensing/actuation problems, like robots. Moreover, the formulation comes with a set of modeling tools that enable the user (the mission controller) to realize specific mission strategies through the management of the *spatio-temporal interactions and dependencies* between the agents.

It is important to remark that the adoption of a linear, MIP formulation has some important advantages [4]: it can be solved to optimality with formal guarantees, a using standard tools they can be solved with any-time properties: solutions are progressively and monotonically improved over computation time and can be retrieved with formal error bounds on optimality. Moreover, the optimal solution is guaranteed to be found in finite time. However, finding the optimal solution for large instances, might require significant computational efforts. We address this issue by adopting a

The authors are with the Dalle Molle Institute for Artificial Intelligence (IDSIA) / SUPSI / Università della Svizzera Italiana (USI), Lugano, Switzerland. E-mail: {eduardo,luca,gianni}@idsia.ch

This research has been partially funded by the Swiss National Science Foundation (SNSF) Sinergia project SWARMIX, project number CRSI22_133059

*matheuristic* methodology [5] that combines computational affordability with *proven bounds on optimality*.

To summarize, the contributions of this paper comprise the following: (i) a mathematical programming model for collaborative mission planning using heterogeneous multi-agent systems that addresses the challenges of heterogeneity, incorporates mechanisms to manage agents' mutual interactions and dependencies, and provides an efficient way to deal with partial controllability of agents in task execution; (ii) an any-time, effective solution approach that allows to compute high-quality plans within short time, and with formal guarantees; (iii) the application of the proposed framework in the context of search and rescue missions and the validation of different aspects in simulation.

## II. Related work

The problem of multi-agent mission planning that we consider is a combination of *task allocation*, *scheduling*, and *routing*. Individually, these topics are the subject of a large amount of research in operations research (OR) and multi-robot systems. In particular, as vehicle routing and machine scheduling (in OR) and multi-robot task allocation problems.

In the field of OR, Vehicle Routing Problem with Profits [6], consists in determining an optimal set of time-constrained routes for multiple vehicles that maximizes the total collected reward from the visited customer. In the other hand, machine scheduling problems [7], in a wide sense, aims at allocating resources (i.e., agents) to a predefined set of time dependent activities (i.e., tasks), while respecting precedence constraints between the jobs, and with the objective of minimizing the project make-span. Due to the integration of planning, task allocation, scheduling, and routing decisions, and the inclusion of spatio-temporal interdependencies between agents in our formulation, the existing mathematical models for these problems cannot be directly applied to the problem we are considering in this work.

In relation to research in robotics, the class of problems closest to ours is that of *multi-robot task allocation* (MRTA), which is the problem of determining which robots should execute which tasks in order to achieve the overall system goals. According to the categorization of Gerkey and Mataric [8], our planning problem covers both *single-task robots* (ST) and *multi-task robots* (MT), and considers *single-robot tasks* (SR), for a *time-extended assignment* (TA). Most of the studies on MRTA problems are focused on *decentralized* solution approaches [4], whose appealing characteristics (e.g., fault tolerance, resiliency, simplicity) often come at the cost of lower quality solutions, [4], and rough or non-existent guarantees about the quality of the solutions provided [9]. *Centralized* approaches, as the one we propose, on the other hand, can satisfy the need for *optimality*, typically provide solutions of better quality than their decentralized counterparts, and can also exhibit an *anytime* property. These approaches, including ours, are typically grounded on mathematical programming frameworks, and make use of exact algorithms such as branch-and-bound [3], and branch-and-price [1]. However, these previous works have been limited to domains where tasks can only be assigned as a whole (i.e., atomic tasks), meaning that the assignment of a task to an agent is made with the assumption that it will spend an uninterrupted effort equal to that required for the completion of the task. This condition precludes (possibly more efficient) solutions in which tasks may be accomplished in an incremental manner over disjoint periods of time during which different agents devoted a certain effort into it, and it also prevents the partial fulfillment of tasks, a condition that enables the agents to better distribute their effort over a larger number of tasks, given time limitations.

Additionally, in our work we consider additional factors related to dependencies among the schedules of the agents that can affect the efficiency of mission performance and/or the feasibility mission plans. These aspects have been recently cataloged as *cross-schedule dependencies* [1]. but their treatment have been limited to *inter-task* dependencies, such as precedence and synchronization constraints [1], (e.g., certain tasks that must be executed simultaneously, or with relative ordering conditions). In contrast, we consider *inter-agent* dependencies, and we introduce a flexible framework for the definition of directives that can be used to promote or enforce of spatio-temporal proximity relations among the trajectories of the agents (i.e., the sequence of locations of their assigned tasks) as a means of managing the *agents' interactions*. Types of constraints that can be expressed here are, for instance, keeping certain agents close to each other while they execute their tasks (e.g., for communication provisioning), or ensuring that some agents remain distant from each other (e.g, for safety reasons).

Deterministic approaches, like all those described above, assume that the effect of the actions of the agents is known. Another class of methods that can be used to tackle MRTA problems that are subject to uncertainties are stochastic planning approaches such as Markov Decision Processes (MDPs) [10]. The application of these methods allows to directly accommodate uncertain aspects related to the environment and to the behavior of the agents, assuming that a world model is provided. However, the resolution of these approaches is notoriously difficult, particularly in the case of multiple agents. Moreover, their increased computational complexity is not justified in scenarios where the only sources of *uncertainty* are the possible deviations in the execution of plans, such as the ones we are considering. As an alternative and novel approach, in this work we propose a simple but yet effective way to define plans that are robust against bounded deviations in their execution. It allows to integrate rough estimates about the possible effects of deviations the agent might incur, without adding further complexity to the already complex problem.

## III. Problem Statement

A team of heterogeneous mobile agents ($\mathcal{A}$) is available to perform a *joint mission* in a distributed environment of a specified dimension. The mission has been decomposed into a set $\mathcal{T}$ of *spatially distributed, location-dependent tasks*. The tasks have the general characteristics described in the Introduction: they can be non-atomic, provide a reward proportional to the progress achieved in its completion, and can be eventually brought to an end. Individual tasks are independent from each other, and the success of the mission depends on how which tasks are selected and on the quality each task is performed, as it is explained in the following.

A task corresponds to the execution of a *particular action* at a *specific location* (or portion) of the environment. Let us assume as given the discretization of the environment into a finite set of locations $\mathcal{L} = \{l_1, \ldots, l_m\}$, and the set of possible actions, $\mathcal{O} = \{o_1, \ldots, o_n\}$. Then, the set of tasks composing the mission is $\mathcal{T} \subseteq \mathcal{O} \times \mathcal{L}$. The complete execution of each task $\tau \in \mathcal{T}$ provides an *utility*, or *reward* the task), indicated with $R_\tau$. The total amount of reward which is provided by a task depends on how important the task is for the mission (e.g., in a search and rescue mission, some locations might be ore important to search accurately than others based on some a priori knowledge).

Due to team heterogeneity, agents have different capabilities, and not all of them are able to perform all actions in $\mathcal{O}$. Moreover, the diverse characteristics of the environment can prevent some agents from reaching certain parts of the area and/or perform actions in it (e.g., a flying robot alone cannot enter into a locked room, a dog cannot dive/search into a lake, a man alone cannot overcome a deep cliff). All these conditions imply that not all the agents are able to execute all tasks in $\mathcal{T}$. Assuming that $\mathcal{O}^k \subseteq \mathcal{O}$ denotes the set of actions that agent $k$ can perform, and $\mathcal{L}^k \subseteq \mathcal{L}$ denotes the set of locations that $k$ can reach, then the set of tasks that can be *feasibly assigned* to agent $k$ is $\Gamma_k = \mathcal{T} \cap \left(\mathcal{O}^k \times \mathcal{L}^k\right)$, with $\bigcup_{k \in \mathcal{A}} \Gamma_k = \mathcal{T}$.

If we look at the tasks as requests to be possibly serviced, and at the agents as the entities with the capabilities to service the requests, the "quality" with which a specific request is serviced depends, first, on the skills of the agents, and then on the time/effort the agents is spending to provide the service. Adopting this point of view, we consider the *service time* as the measure of effort spent by an agent, with the increase in the level of completion of a task which is proportional to the time devoted to it. Since we deal with heterogeneous teams, different agents may demonstrate different levels of performance in accomplishing the same task, due to their potentially different skills in relation to the specific local characteristics of the task. For instance, the time effort required for pushing a box across a room is dependent upon how much force the agent is able to apply to the box and on well it can detect possible obstacles.

We model the difference in performance among the agents through the *completion rate function* $\phi : \mathcal{A} \times \mathcal{T} \mapsto \mathbb{R}$, which specifies, for each one of the agents, the amount of effort (measured as service time in seconds) required for the completion of each task in $\mathcal{T}$. When an agent $k \in \mathcal{A}$ performs a task $\tau \in \mathcal{T}$ for $t$ seconds, it contributes with $100 \cdot \phi^k(\tau) \cdot t$ percent of *task completion*. Thus, the amount of time needed for agent $k$ to complete a task $\tau$ is $\frac{1}{\phi^k(\tau)}$. In other words, $\phi^k(\tau)$ represents a linear measure of the *efficacy* of agent $k$ for performing task $\tau$. In the following, for the reason of reducing computational complexity, we assume that the whole mission time is discretized into *mission intervals* of equal length $\Delta_T$ seconds. Consequently, the completion rate functions are scaled and turned into what we term the *performance functions* $\varphi_k(\tau)$, precisely representing efficacy of agent $k \in \mathcal{A}$ when executing task $\tau \in \mathcal{T}$:

$$\varphi_k(\tau) = \begin{cases} \phi_k(\tau)\Delta_T & \text{if } \tau \in \Gamma_k \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

The *completion map* $C_m : \mathcal{T} \mapsto [0, 1]$ expresses the level of completion required for each one of the tasks $\tau$ included in the mission specification $\mathcal{T}$. For instance, a value of 0 indicates that task $\tau$ has been completed and, therefore, no further efforts from the agents are required. If an agent attempts to further deal with the task it will receive no additional reward, which will amount a waste of time and resources. A service provided to task $\tau$ that decreases its required completion $C_m(\tau)$ of a fraction $p$ (i.e., executing $p \cdot 100$ percent of $\tau$). provides a partial utility of $pR_\tau$.

For each agent $k \in \mathcal{A}$, we also define a directed *traversability graph* $G_k = (\Gamma_k, E_k)$ where $E_k$ contains an arc $(i, j)$ if task $j$ can be scheduled right after task $i$. In the general case, graph $G_k$ is complete (i.e., $E_k = \Gamma_k \times \Gamma_k$). However, specific scenarios may impose constraints over the sequences of tasks that can be executed. For instance, when some tasks cannot be designated immediately after others (e.g., due to mobility constraints), or when specific tasks must be serviced immediately before servicing others (e.g., unblocking a road in order to reach parts of the area).

Based on the above notions and specifications, the *Multi-Agent Mission Planning Problem* (MMPP) can be stated as follows. Given a set of agents, a set of assignable tasks and a traversability graph for each agent, and a given *limited time budget* $T$, the MMPP consists in determining joint plans for the activities of the agents in the environment which enable the timely selection of the tasks to perform, and aiming to maximize the overall mission utility (i.e., the sum of all gathered rewards). A *solution to the MMPP* (i.e., a mission plan) consists of time-constrained sequences of task, one for each agent, that can be represented as paths in $G_k$. In addition, plans also define how much effort (i.e., devoted time) each of the selected tasks will receive. Due to time limitations not all tasks might be performed: a plan implicitly defines a selection among all tasks in $\mathcal{T}$.

Figure 1 shows an example plan for three agents. A set of 16 tasks are located in rectangular grid. The traversability graph allows movements between adjacent tasks in the Moore neighborhood. Since in this example each task is associated to a unique location, we denote tasks as $\tau_{ij}$, where $0 \leq i,\ j < 4$ indicate the column and row of the location, respectively, of the task in the grid. The top of the figure shows the performance functions $\varphi$ of each agent. For instance, the value $\varphi_1(\tau_{00}) = 1$ indicates that agent 1 is able to complete task $\tau_{00}$ in a single mission interval, while $\varphi_3(\tau_{00}) = \frac{1}{4}$ indicates that agent 3 is able to complete one-fourth of the task in the same amount of time.

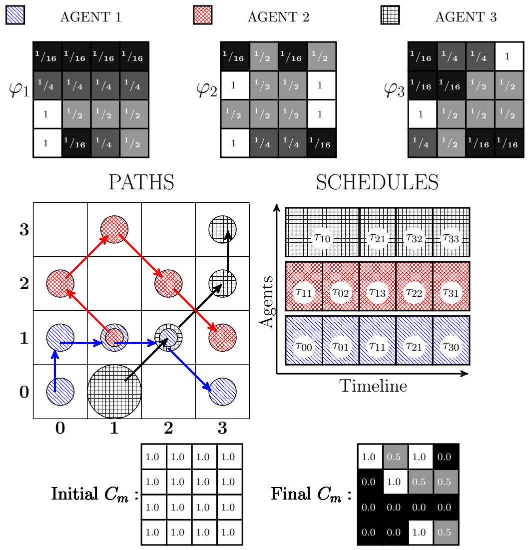


Fig. 1: Example plan for 3 agents. On the top, the performance functions $\varphi_k$ for each agent. The example shows a mission plan with $T = 5$. Paths are depicted in the left, while schedules are depicted as a timeline in the right. On the bottom, the initial (left), and final (right) coverage maps.

All agents are initially located at the bottom-left corner, and paths can start from $\tau_{00}$ or any of its adjacent tasks. The mission plan is defined for a time budget of $T = 5$ mission intervals and determines the following paths: $\langle \tau_{00}, \tau_{01}, \tau_{11}, \tau_{21}, \tau_{30} \rangle$, $\langle \tau_{11}, \tau_{02}, \tau_{13}, \tau_{22}, \tau_{31} \rangle$, $\langle \tau_{10}, \tau_{21}, \tau_{32}, \tau_{33} \rangle$, for agents 1,2,and 3, respectively. The schedules of all agents allocate one time unit (i.e., mission interval) for all tasks with the sole exception of $\tau_{10}$, which is assigned to agent 3 for two mission intervals. Initially, all tasks require full service: $C_m(\tau) = 1$, $\forall \tau$. The execution of the mission plan decreases the level of completion required of the tasks involved in the plan. For instance, task $\tau_{00}$ is completed by agent 1, and task $\tau_{11}$ is completed after being serviced by agents 1 and 2. Note that some of the tasks do not receive any service (e.g., $\tau_{03}$), while others are completed (e.g., $\tau_{00}$), or partially completed (e.g., $\tau_{30}$).

## IV. Multi-Agent Mission Planning as a MIP

In this section, we formulate the MMPP as a mathematical optimization problem. We start with a reference *mixed-integer linear program* (MIP) model that can be used to determine plans for each one of the agents with the aim of maximizing the gathered reward. From this reference formulation, we incrementally add features as extensions/modifications to the MIP model addressing important aspects that need to be accounted for in order to be able to deal with the complexity of the real world. These additional aspects are included as integral part of the problem, but in a sense optional from the point of view of the main controller of the planning system. These include: (i) The grouping of potentially close tasks together into single planning units, which allows to define plans in terms of groups of tasks, (instead of single tasks); (ii) the need to be robust to potential *deviations in the way plans might be actuated* by the agents; (iii) the enforcement or promotion of spatio-temporal relations and dependencies among agents' plans to, e.g., *promote synergies* or *minimize task interference*.

The following *decision variables* are employed to build the MIP optimization model for the MMPP presented in (2)-(12):

$x_{ijk}$: binary, equals 1 if agent $k$ traverses arc $(i, j) \in E_k$;
$y_{ik}$: binary, equals 1 if agent $k$ is assigned to task $i \in \Gamma_k$;
$\Phi_\tau$: service provided to task $\tau \in \mathcal{T}$ by all agents;
$t_{ik}$: starting time of execution of task $i \in \Gamma_k$ by agent $k$;
$w_{ik}$: time assigned to task $i \in \Gamma_k$ for agent $k$.

$$\text{maximize} \quad \sum_{\tau \in \mathcal{T}} R_\tau \Phi_\tau \tag{2}$$

subject to

$$\sum_{(0,j)\in E_k} x_{0jk} = 1 \qquad k \in \mathcal{A} \tag{3}$$

$$\sum_{(i,0)\in E_k} x_{i0k} = 1 \qquad k \in \mathcal{A} \tag{4}$$

$$\sum_{(i,j)\in E_k} x_{ijk} = \sum_{(j,i)\in E_k} x_{jik} = y_{jk} \qquad k \in \mathcal{A},\ j \in \Gamma_k \tag{5}$$

$$t_{ik} + w_{ik} - t_{jk} \le (1 - x_{ijk})T \qquad k \in \mathcal{A}, (i,j) \in E_k,\ i,j \ne 0 \tag{6}$$

$$y_{ik} \le t_{ik}, w_{ik} \le T y_{ik} \qquad k \in \mathcal{A},\ i \in \Gamma_k \tag{7}$$

$$\Phi_\tau \le \sum_{k\in\mathcal{A}} \sum_{i\in\Gamma_k, i=\tau} \varphi_k(i) w_{ik} \qquad \tau \in \mathcal{T} \tag{8}$$

$$0 \le \Phi_\tau \le C_m(\tau) \qquad \tau \in \mathcal{T} \tag{9}$$

$$\Phi_\tau \in \mathbb{R} \qquad \tau \in \mathcal{T} \tag{10}$$

$$t_{ik},\ w_{ik} \in \mathbb{N} \qquad k \in \mathcal{A}, i \in \Gamma_k \tag{11}$$

$$x_{ijk},\ y_{jk} \in \{0,1\} \qquad k \in \mathcal{A},\ i,j \in \Gamma_k \tag{12}$$

The objective function (2) defines the quality of a mission plan in terms of its utility, quantifying the expected effect of agents' activities over the current state of the completion map $C_m$. A dummy vertex (denoted by 0) represents the starting point and ending point of the agent paths. To this end, graphs $G_k$ are extended with arcs from 0 to each of the initially accessible tasks and from all elements to the dummy 0. Constraints (3-4) ensure that paths start and end at the dummy vertex 0. Path continuity is guaranteed by constraints (5). Constraints (6) eliminate sub-tours [11] and, together with (7), they define the bounds of variables $t$ and $w$. The completion level of each task is bounded by constraints (8-9) These bounds ensure that a task provides a maximum reward equal to $R_\tau C_m(\tau)$, and the utility of a plan is contributed with $R_\tau$ scaled by the completion of $\tau$ (i.e., $\Phi_\tau$). Finally, constraints (10-12) set the real, integer, and binary requirements on the model variables.

This formulation represents the core of the model which can be used to maximize the utility of joint agent plans.

### A. Addressing heterogeneity with macro-tasks

The first extension to the model considers grouping subsets of tasks into single planning units. The main motivation of this way of proceeding is to address the issue of having heterogeneous agents, with different mobility and sensing and actuation capabilities. In fact, given that the planning must be carried out over a predefined, task decomposition (i.e., the set of tasks) and a common mission interval, it might no be feasible or reasonable to assign a single task for a mission interval to every agent. For instance, when searching an area, be the given spatial decomposition equivalent to cells of size 50× 50 m$^2$, each corresponding to a *searching task*. If the mission interval is set to 5 minutes, a human can be allocated one task per time step, while the same would not be reasonable for an aerial robot, that in same amount of time could easily search areas of size 200× 200 m$^2$, corresponding to 16 tasks

To deal with these issues, we introduce the concept of *macro-tasking*: given the initial task decomposition, a preprocessing is performed to define the set of bundles of tasks that can be assigned in one planning step to each agent. Based on the agents' specific characteristics, these bundles are potentially different from agent to agent. A bundle is regarded as a planning unit: the agent autonomously will decide how to allocate time among the composing tasks, and in which order they will be serviced. We refer to a bundle as a *macro-task*, and the set of all predefined macro-tasks is denoted by $\mathcal{T}^M$, which is a subset of the power set of $\mathcal{T}$. Therefore, in our notation, the set of tasks that can be assigned to agent $k \in \mathcal{A}$ is still denoted by $\Gamma_k$ but now includes all macro-tasks that can be assigned to it.

In order to fit the use of macro-tasks into the model, we require to precisely define their utility, based on the fact that the time effort must be split among the composing tasks. Therefore the question is about which fraction of time will be spent on each task composing a macro-task. We assume that estimates of the *effort distribution schemes* within each macro-task, for each one of the agents, are given and defined by $\sigma_k : \mathcal{T}^M \times \mathcal{T} \mapsto [0,1]$, with $\sum_{\tau\in\omega} \sigma_k(\omega,\tau) = 1$, $\forall k \in \mathcal{A}$, $\omega \in \Gamma_k$, and that, during the execution of macro-task $\omega$ by agent $k$, the agent spends $100\sigma_k(\omega,\tau)$ percent of the assigned time doing task $\tau \in \omega$. In the experiments we adopt a uniform distribution of the effort among all tasks composing a macro-task: $\sigma_k(\omega,\tau) = (\Delta_T/|\omega|)$.

### B. Robustness against deviations in execution

In practice, mission plans are computed assuming that their corresponding nominal performance reflected in execution. However, when plans are not executed as expected, their actual performance can experience a significant degradation.

Given that plans are expressed in terms of sequences of spatial tasks, they also describe trajectories in the environment, which are expected to be followed by the agents during the execution of the plans. Yet, in many real-world situations, agents might deviate from their assigned trajectories. As a consequence, some tasks, not included in the plan and located around the specified trajectory, may receive some unplanned amount of service, while the originally assigned tasks receive lesser amount of effort than they were suppose to.

In this work we address these issues by anticipating the effect that deviations could have over mission plans. The goal is to obtain robust plans that guarantee an optimal use of all the mission assets even in the case that the execution undergoes some deviations. The method exploits the flexibility of the macro-task concept in order to accommodate localized uncertain movements around the prescribed plans. It consists in the following preprocessing of tasks.

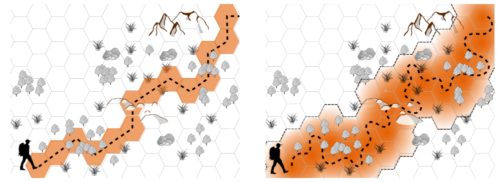

Fig. 2: Providing robustness using macro-tasking: the plan (left) is enlarged including proximal tasks (right).

For any task $\omega \in \Gamma$ that can be subject to deviations in execution, we enlarge its geographical region in order to also include *proximal tasks*. To this end, we define a mapping between the original task and its geographical extensions, and we indicate with $U(\omega)$ an extended macro-task that corresponds to the original task $\omega$ plus the proximal tasks. The rationale behind this way of proceeding consists in assuming that, with some uniform probability, an agent deviates from its assigned path and devote some time to any of these proximal tasks. Therefore, we consider that deviations in the execution of $\omega$ translate into a fraction of time being spent on some of the tasks in $U(\omega) \setminus \{\omega\}$.

We assume that an estimate of the probability of deviation from task $\omega$ is given, and denoted as $p_u$. Then, in order to accommodate the estimated effect of deviations, we define the effort distribution scheme of the corresponding macro-task $U(\omega)$ in the following way:

$$\sigma(U(\omega),\tau) = \begin{cases} (1-p_u) & \text{if } \tau = \omega \\ p_u/|U(\omega)\setminus\{\omega\}| & \text{if } \tau \in U(\omega)\setminus\{\omega\} \end{cases} \tag{13}$$

Note that planning in terms of macro-tasks $U(\omega)$ is equivalent to optimizing mission plans assuming one of the possible deviation scenarios: the agent deviates from the original plan and devotes $100 \cdot p_u\%$ of the allocated time to all the extended tasks. Moreover, this amount of time is distributed uniformly among them. This approach shares some similarities with Robust Optimization methods [12], in which uncertainties are also tackled, in a deterministic way, assuming realizations of the random variables.

Figure 2 illustrates the approach. In this example, a human agent performs sensing activities (e.g., finding a missing object), and plans are defined in terms of tasks at one of the cells. The track specified by a plan is indicated in the figure on the left. In the figure on the right, the same plan is now defined in terms of macro-tasks, that extend the original tasks to cover a cluster of 7 adjacent cells. The nominal performance of each plan is determined by quantifying the amount of service provided to each of the involved tasks, which are highlighted in the figures. As noted, the nominal performance of the second plan considers possible deviations from the original track, which may involve the exploration of cells that do not belong to the original plan.

### C. Proximity constraints between groups of agents

Direct *dependencies in space and time* among the plans of individual agents are common in the domains that we are targeting. For instance, in a search and rescue scenario the mission commander might request mission plans in which every agent is *always located within the communication range* of a mobile wireless relay unit, to allow a reliable tracking; or plans that keep air-scent dogs, if used, *always separated* from each other by a minimum distance, to avoid mutual task interference. Both situations represent use cases of *proximity constraints*: controlling the minimum and maximum distance between groups of agents. To control these aspects, we introduce a general class of *proximity constraints*.

Let $\psi_{kilj}$ be the estimated distance, in meters, between two agents $k, l \in \mathcal{A}$ carrying out activities at the location of tasks $i \in \Gamma_k,\ j \in \Gamma_l$. Given two disjoint subset of agents $\mathcal{A}', \mathcal{A}'' \subset \mathcal{A}$, we introduce two variables; the *minimum distance* between $\mathcal{A}'$ and $\mathcal{A}''$ at time $t$ denoted by $\Theta^t_{\mathcal{A}',\mathcal{A}''}$, and the *maximum distance*, represented by $\Psi^t_{\mathcal{A}',\mathcal{A}''}$. Let $p_t : \mathcal{A} \mapsto \Gamma$ be the *position* (i.e., the assigned task) of an agent at time step $t$ in the current solution. Using this notation, the model variables for $\Theta^t$ and $\Psi^t$ are defined as:

$$\Theta^t_{\mathcal{A}'\mathcal{A}''} = \min_{k\in\mathcal{A}',\ l\in\mathcal{A}'', i=p_t(k),\ j=p_t(l)} \psi_{ij} \tag{14}$$

$$\Psi^t_{\mathcal{A}'\mathcal{A}''} = \max_{k\in\mathcal{A}',\ l\in\mathcal{A}'', i=p_t(k),\ j=p_t(l)} \psi_{ij} \tag{15}$$

A *proximity constraint* consists in establishing a lower or a upper limit to one of the $\Theta^t$ or $\Psi^t$ variables for a specific time step $t$. We classify, and name, the proximity constraints according to the effect that can be achieved through their use. The first type, called *coalition* constraints, correspond to those that set upper limits to the maximum distance ($\Psi^t \leq$). Using these constraints, two groups of agents can be kept close to each other, as forming a temporary *coalition*. The *network* constraints set upper limits to the minimum distance ($\Theta^t \leq$). These constraints are typically employed to enhance and promote communication (i.e., *network connectivity* [13]) when the possibility of communication between two nodes is directly related to their physical separation distance. Next, the *interference avoidance* constraints, are those that can be used to keep two groups of agents distant from each other such as to *avoid task interference* [14], dangerous situations, and other undesirable events that are likely to occur when these agents get closer to each other. These constraints are defined by setting lower limits to the minimum distance ($\leq \Theta^t$). Finally, the *sparsity* constraints, are those where the maximum distance must be greater than a certain value ($\leq \Psi^t$). Their effect is the enlarging of the area covered by members belonging to the corresponding groups.

Due to space limitations, the reader is referred to [15] for a detailed description of the MIP model.

## V. Solution approach

Optimal solutions to the MIP model can be found using a general, out-of-the-box solver. However, since a MIP solver does not rely on any domain specific knowledge about the problem, the progressive improvement over time of the quality of the best solution available (i.e., incumbent solution) tends to be slow. To address this issue, and therefore improve the anytime behavior, we developed a novel problem solving approach that synergistically combines an exact mathematical solver and a hybrid metaheuristic algorithm.

### A. Hybrid meta-heuristic: GA + ILS

The basis of the heuristic approach is the observation that the MMPP can be decomposed into two distinct, but interconnected problems, namely (i) the selection of sequences of tasks and (ii) the service scheduling. Accordingly, we tackle the problem using a two-level decomposition. At the top level, we explore possible sequences of tasks for each agent using a genetic algorithm (GA). At the bottom level, an iterated local search method (ILS) is employed to assess the *utility*, in terms of mission performance, of each of the considered sequences. The integration of ILS into the GA represents a form of hybridization, therefore we refer to it as a *hybrid meta-heuristic method*.

### B. Shared Incumbent Environment

A shared incumbent environment (SIE) is a general methodology to realize collaborative combinations of mathematical programming and metaheuristic approaches [16]. We implement a SIE as two independent processes, each running a different solution method (i.e., a MIP solver and the hybrid metaheuristic) on the same problem instance, and able to *communicate* with each other. The cooperation scheme between both solvers consists in the continuous exchange of their best found solutions so far. From the MIP solver side, this corresponds to the best upper bound (also known as the *incumbent solution*). In the metaheuristic, it is best individual that has been evaluated so far. The exchanged information is then used by the MIP solver to improve its current incumbent and prune the branch and bound tree, and by the metaheuristic to guide the search to more promising regions of the solution space and to avoid getting stuck in local minima. As a result, we obtain a solution method that speeds-up the computation of high-quality mission plans, and at the same time preserves the anytime property and the formal error bounds on optimality that a mathematical solver provides.

## VI. Evaluation

In this section we evaluate the proposed mission planning approach in the context of wilderness search and rescue scenarios (WiSAR). This application is derived from our previous works [17], [13]. We consider a single action to be performed by the agents, that is *searching for a missing target*. The set of possible locations at which this action can be performed is obtained through a cellular grid decomposition of the area. In line with our mission planning model, the completion map $C_m$ relates cells to numerical values representing the amount of coverage required.

The problem instances are based on an area of size 700 $\times$ 700 $m^2$. The area has been decomposed into cells of 100 $\times$ 100 $m^2$. We consider three different models of agents, that correspond to types of agents that are commonly used

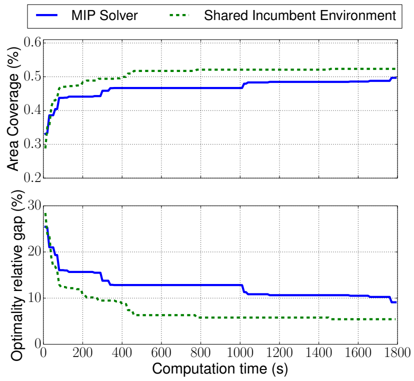


Fig. 3: Comparison of the anytime behavior of the MIP solver vs. the shared incumbent environment.

in WiSAR missions, namely, (i) aerial robotic platforms, more specifically quad-rotors, (ii) human rescuers, and (iii) air-scent dogs. To account for the increased mobility of aerial robots compared to dogs and humans, we define their possible sectors of size 200 $\times$ 200 $m^2$ (i.e., all clusters are of 2 cells $\times$ 2 cells). For human and air-scent dog agents, the sectors are in one-to-one correspondence with the cells. For each agent, we defined a *search efficacy*, that relates its skills, as well as the effect of local conditions, to the amount of area covered over time [18]. The search efficacy plays the role of the completion rate ($\phi^k$) in our formulation.

*a) Performance of solution approaches:* First, we assess the performance of the proposed solution approach. We computed solutions using only a MIP solver, and using the SIE approach. As MIP solver we use CPLEX®. In Figure 3, we show the performance over time of both methods while solving a problem instance consisting of 12 agents and a mission time span of 10 time steps. On top, we can appreciate the progress over the quality of the best current solutions, in terms of area coverage of the corresponding mission plans. Below, the optimality gap, that is, the proven relative gap between the value of the current solution and the bound on the optimal solution. As noted, the SIE exhibits a better anytime behavior, and is able to find solutions with optimality gaps under 10% after a few minutes, while the MIP solver struggles to achieve similar solution quality. This indicates that the proposed SIE allows to speed up the generation of high-quality mission plans.

*b) Partial fulfillment of tasks:* One of the aspects that differentiates our planning approach from previous works is the capability of describing plans in terms of non-atomic tasks, which accepts the condition of leaving some tasks *partially fulfilled* by the end of the planning horizon. We therefore compare the performance of plans that consider non-atomic tasks against the performance obtained by plans that restrict the agents to fully complete the assigned tasks. The atomicity of tasks is ensured adding the following set of constraints into the model:

$$T_i^k y_{ik} \leq w_{ik} \quad \forall k \in \mathcal{A},\ i \in \Gamma_k \tag{16}$$

where $T_i^k$ is the amount of time that agent $k$ requires to complete task $i \in \Gamma_k$. In other words, the previous set of constraints ensures that, once a task is assigned to an agent, that agent will bring that task to completion.

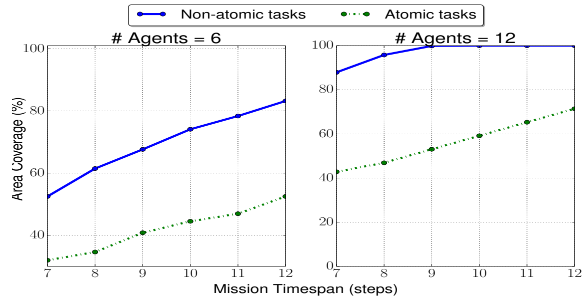


Fig. 4: Performance of Non-atomic tasks vs. Atomic tasks.

In the evaluation, we have considered teams of 6 and 12 agents, with equal number of agents of each type, and a mission time-span ranging from 7 up to 12 time steps. Figure 4 shows the performance (in terms of area coverage) of plans computed with the restriction of atomic tasks, and of plans computed using non-atomic tasks. In all the scenarios the performance of plans using non-atomic tasks is significantly higher than the performance of those with atomic tasks. This suggests that the partial fulfillment condition enables the agents to better distribute their effort over a larger number of tasks, and as consequence, achieving higher performance levels. These results encourage the use of non-atomic tasks to define mission plans.

*c) Management of agents' interactions:* In the following, we illustrate and validate the use of proximity constraints as a means of managing the agents' mutual interactions. We consider a typical issue arising in WiSAR missions involving the participation of air-scent dogs, namely the problem of task interference. Task interference is caused when the presence of other agents affects the behavior of dogs and deviates their focus, therefore degrading their performance in the field [14]. In order to prevent this issue, it is of interest to obtain mission plans that keep dogs separated from other ground agents, by a predefined minimum distance. We considered teams composed by 6, and 12 agents, each with an equal number of human rescuers and air-scent dogs. We compute mission plans with and without *interference constraints*. We consider different values for the minimum distance required between the agents. Specifically, three different cases, where the desired minimum distance was set to 100m, 150m, and 200m, respectively. The constraints were included in the model as *soft constraints*, meaning that their violation was penalized in the objective function.

We simulate the execution of plans using stochastic mobility models derived from our previous work [18], [17], and measure the distances between the agents during the mission. Results in Figure 5 show the distances between each dog and its closest ground agent, averaged over all the mission (top), and the nominal area coverage provided by the mission plans (below). We can appreciate that the use of proximity constraints allows to increase the distance between dogs and possible interferes, at a small expense in area coverage. Therefore, these plans will diminish the effect of interference while still exhibiting good performance.

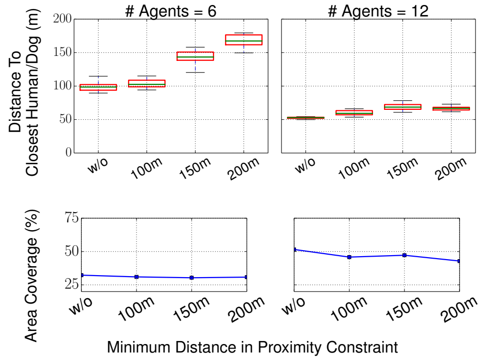


Fig. 5: Controlling task interference through the use of proximity constraints.

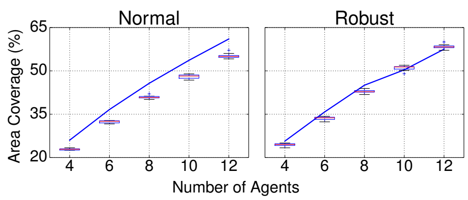


Fig. 6: Validation of robust mission plans.

*d) Robustness of plans:* Finally, we validate the use of the robust approach to address deviations in execution of plans. We compute mission plans using extended sectors that include adjacent cells and assume that agents deviate to these cells with probability $p_u = 0.4$. We take the obtained plans, in terms of original sectors, and simulated their execution. We perform 10 simulation runs for each plan and compute the final state of the coverage map using the agent traces. Figure 6 shows the final coverages in simulation (boxplot), together with the nominal coverage of the plans (solid line). After analyzing the results, we make the following observations. First, the simulated performance of plans obtained without robustness considerations is significantly lower than their nominal coverage . In contrast, this difference is reduced for the robust plans. The more precise assessment of robust plans suggests that the proposed method captures indeed the possible deviations. Secondly, we note that the differences in execution performance between the normal plans and their robust counterparts in these particular instances are not significant. However, the first observation also suggests that the performance degradation that robust plans might exhibit in other types of scenarios can be expected to be lower than the degradation that normal plans might suffer.

## VII. Conclusions and Future Work

We presented a mixed integer linear formulation for mission planning in heterogeneous teams of physical agents. We targeted missions which can be seen as composed of a set of spatially distributed tasks. The aim of the mathematical formulation is to assign plans to the agents by exploiting their specific characteristics in relation to the tasks, and promoting their synergies. We included in the formulation a number of different aspects derived from real-world mission planning scenarios, which include the ability to enforce spatio-temporal relations among groups of agents, dealing in a robust way with the uncertainty in plan execution, letting open the possibility to complete a task incrementally, by different agents in different times. As a result the model is highly realistic and flexible. We also presented a computationally effective solution approach, that preserves some optimality guarantees while saving computations.

Future work will address the definition of more effective and fast solution methodologies, and the definition of decentralized strategies, where each agent autonomously computes its own plan and coordinate with the others.

## References

[1] G. A. Korsah, B. Kannan, B. Browning, A. Stentz, and M. B. Dias, "xBots: An approach to generating and executing optimal multi-robot plans with cross-schedule dependencies," in *Proc. of IEEE ICRA*, 2012, pp. 115–122.

[2] G. Jones, B. Dias, and A. Stentz, "Time-extended multi-robot coordination for domains with intra-path constraints," *Autonomous robots*, vol. 30, no. 1, pp. 41–56, 2011.

[3] M. Koes, I. Nourbakhsh, and K. Sycara, "Constraint optimization coordination architecture for search and rescue robotics," in *Proc. of IEEE ICRA*, 2006, pp. 3977–3982.

[4] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[5] V. Maniezzo, T. Stützle, and S. Voß, Eds., *Hybridizing Metaheuristics and Mathematical Programming*, ser. Annals of Information Systems, vol. 10. Springer, 2009.

[6] D. Feillet, P. Dejax, and M. Gendreau, "Traveling salesman problems with profits," *Transportation Sc.*, vol. 39, no. 2, pp. 188–205, 2005.

[7] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models, and methods," *European Journal of Operational Research*, vol. 112, no. 1, pp. 3–41, 1999.

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

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

[10] C. Guestrin, D. Koller, and R. Parr, "Multiagent planning with factored mdps." in *NIPS*, vol. 1, 2001, pp. 1523–1530.

[11] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," *European Journal of Opererations Research*, vol. 209, no. 1, pp. 1–10, 2011.

[12] A. Ben-Tal and A. Nemirovski, "Robust optimization–methodology and applications," *Mathematical Programming*, vol. 92, no. 3, pp. 453–480, 2002.

[13] E. F. Flushing, M. Kudelski, L. Gambardella, and G. A. Di Caro, "Connectivity-aware planning of search and rescue missions," in *Proc. of IEEE SSRR*, 2013.

[14] E. Feo Flushing, L. Gambardella, and G. A. Di Caro, "Strategic control of proximity relationships in heterogeneous search and rescue teams," in *Proc. of the 3rd IROS Workshop on Robots and Sensors integration in future rescue INformation system (ROSIN)*, 2013.

[15] E. Feo Flushing, G. A. Di Caro, and L. Gambardella, "Collaborative missions with heterogeneous teams: Mathematical model and solution approach," IDSIA, Lugano (Switzerland), Tech. Rep. 02-14, 2014.

[16] E. Feo Flushing and G. A. Di Caro, "Exploiting synergies between exact and heuristic methods in optimization: an application to the relay placement problem in wireless sensor networks," in *Proc. of 7th Int. Conf. on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS'12)*, ser. Springer LNICST, no. 134, 2014.

[17] E. Feo Flushing, L. Gambardella, and G. A. Di Caro, "Search and rescue using mixed swarms of heterogeneous agents: modeling, simulation, and planning," IDSIA, Lugano, Switzerland, Tech. Rep. 05-12, 2012.

[18] E. Feo Flushing, L. Gambardella, and G. A. Di Caro, "GIS-based mission support system for wilderness search and rescue with heterogeneous agents," in *Proc. of 2nd IROS Workshop on Robots and Sensors integration in future rescue INformation system (ROSIN)*, 2012.