Achieving Multitasking Robots in Multi-Robot Tasks

Winston Smith and Yu Zhang

Abstract— **One simplifying assumption made in the existing and well-performing multi-robot systems is that the robots are single-tasking: each robot operates on a single task at any time. While this assumption is innocent to make in situations with sufficient resources such that robots can work independently, it becomes a restriction when they must share capabilities. In this paper, we consider multitasking robots with multi-robot tasks. Given a set of tasks, each achievable by a coalition of robots, our approach allows the coalitions to overlap by exploiting task synergies based on the physical constraints required to maintain these coalitions. The key contribution is a general and flexible framework that extends the current multi-robot systems to enable multitasking. The proposed approach is inspired by the information invariant theory, which orients around the equivalence of different information requirements. We map physical constraints to information requirements in our work, thereby allowing task synergies to be identified by reasoning about the relationships between such requirements. We show that our algorithm is sound and complete. Simulation results show its effectiveness under resource-constrained situations and in handling challenging scenarios in a realistic UAV simulator.**

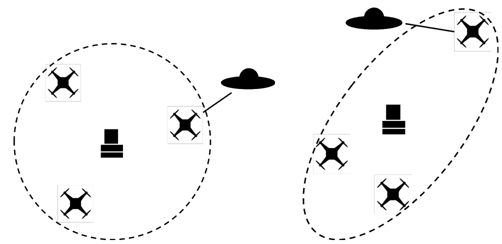

Fig. 1: Scenario that illustrates a synergy between two tasks, a centroid and a monitoring task, where one robot is shared and multitasking. The left and right figures show the change of robot configurations as the target changes its position.

I. INTRODUCTION

To address a multi-robot task, one simplifying assumption made in the literature is that the robots are single-tasking. Wherever robots must coordinate closely to share capabilities, such as when a robot has a capability required in multiple tasks, the solution is often to achieve these tasks sequentially. Such a practice hurts task efficiency (i.e., increasing the makespan) and is feasible only when no concurrent task execution is required. As a result, it significantly limits the current multi-robot systems. In this paper, we consider multi-robot tasks with multitasking robots to enable a robot to operate simultaneously on multiple tasks (also referred to as multi-task robots in [1]).

While multitasking is desirable when there are resource contentions, the fundamental question of its feasibility must be carefully considered. In particular, a robot's ability to perform multitasking is prominently determined by the physical requirements of the tasks that are being taken up simultaneously.[1] For example, for a robot to share its localization capability in a cooperative navigation task, it must stay within the proximity of the robot that requires its assistance; for a UAV to share its camera sensor with a ground vehicle in a cooperative monitoring task, it must maintain its camera head direction towards the target. Hence, the main challenge to enable multitasking robots lies in identifying synergies between the underlying physical constraints across different tasks. Fig. 1 illustrates a scenario with a synergy between the physical constraints across two tasks: one of them is a centroid task that requires three robots to maintain their centroid over the base station. This task introduces a physical constraint on the centroid of the three robots. The other is a monitoring task that requires one of the robots to keep a target within its field of view, which introduces a constraint on the relative position between the monitoring robot and target. There exists a synergy between these constraints since two of the three robots assigned to the centroid task can adjust their positions to maintain the centroid while allowing the third (i.e., the monitoring robot) to track the target.

The proposed approach is inspired by the information invariant theory [2] that orients around the equivalence of different information requirements. In our work, we map physical constraints to information requirements. In such a way, determining the synergies between physical constraints becomes that of reasoning about the compatibility between different information requirements. More specifically, given a set of physical constraints specified as information requirements, our approach checks whether these requirements introduce any conflicts according to a set of inference rules. These general rules are designed to capture the equivalence between different information requirements. A synergy is identified between two tasks if their information requirements introduce no conflicts. For example, in Fig. 1, these rules are used to determine that the centroid requirement does not conflict with the requirement of relative position.

To the best of our knowledge, our work represents the first general framework for achieving multitasking robots in multi-robot tasks. It removes a restrictive assumption made in multi-robot systems by enabling overlapping coalitions. A formal approach is presented based on the notion of information equivalence first introduced in the information in-

Winston Smith and Yu Zhang are with the School of Computing, Informatics and Decision Systems Engineering, Arizona State University {wtsmith7, yzhan442}@asu.edu.

[1]While there may be other factors that affect the feasibility of multitasking, such as a limitation on the communication bandwidth, we focus on physical constraints as the main determinant.

variant theory [2]. Simulation results show that our approach achieves better efficiency statistically under instantaneous task assignments in situations where resources are limited, and that it extends the capabilities of multi-robot systems in challenging scenarios using a realistic multi-UAV simulator.

II. Related Work

Much effort in multi-robot systems has been dedicated to the task allocation problem [1]. When the robots are single-tasking and tasks are single-robot tasks, the problem is also known as the assignment problem, which has efficient solutions [3], [4]. Considering multi-robot tasks, however, significantly increases the problem complexity [1]. There has been a notable amount of focus on approximate solutions [5], [6], [7], [8], [9], [10], and practical issues with applications in distributed robot systems [11], [12], [13], [14], [15], [16], [17], [6], [18], [19], [20], [21]. As far as we know, there exists little discussion on generalizing the problem to multitasking robots with few exceptions discussed below. It is an interesting future direction to study how to efficiently perform task allocation with multitasking robots.

It may be tempting to consider multitasking by applying prior work on task allocation that allows overlapping coalition structures [22], [23]. However, these prior studies are concerned mainly with the utility optimization problem while assuming that the influence of multitasking is captured by the so-called coordination costs provided a priori. In reality, such costs are difficult to estimate for multitasking robots and are generally not fixed values. This is because the feasibility of multitasking, which has a significant impact on the coordination cost, depends on complex reasoning about the physical constraints required for the tasks (that are simultaneously taken up by a multitasking robot). Our study hence goes beyond the prior work by explicitly addressing the query regarding its feasibility. The influence of physical constraints has been identified as a practical issue in distributed robot systems [6] although the focus there is on sensor placements and robots are still single-tasking. The problem is addressed by manually specifying a feasibility function.

The information invariant theory is introduced to reason about the equivalence of sensori-computational systems [2]. It has since been used to develop multi-robot systems that have demonstrated an impressive level of flexibility [19], [24]. Information invariant is well connected to the notion of information space [25]. Both are concerned with the relationship between different information requirements. The difference being that the latter is more focused on the minimalism aspect [26]. To the best of our knowledge, we are also the first to apply information equivalence to reason about synergies between multi-robot tasks to achieve multitasking.

III. Approach

To model information requirements, we use information instances as in [19]. A physical constraint is specified as a constraint on an information instance.

Definition III.1 (Information Instance)**.** *An information instance, denoted as $F(\boldsymbol{E})$, captures the semantics of information where F is the type of the information and $\mathbf{E}$ is an ordered set of referents.* □

Information instances are used to *label* the actual information. In this work, we use capital F to denote information instance and type, and f to denote the *value* of the information. For example, $F_R(r_1, r_2)$ refers to "*the relative position between r_1 and r_2*", where the suffix R denotes relative position; $f_R(r_1, r_2)$ corresponds to the numerical values of this information. For brevity, we sometimes use F without the referents to denote an information instance. Next, we more formally define physical constraint as follows:

Definition III.2 (Physical Constraint)**.** *A physical constraint is a constraint on the value f of an instance F.* □

Notice that the exact value f for a constraint on F may depend on the task, environment settings, and robot configurations dynamically, and hence may not be specified a priori. This means that a physical constraint should be viewed as a placeholder or variable with a label F whose value assumes different values at different times. For example, in a tracking task, a constraint on the relative position to the target may be influenced by the environment settings due to the need for avoiding occlusions. These considerations are assumed to be handled by the execution modules [27], which is out of the scope of this work. To infer about information invariants, we define information inference:

Definition III.3 (Inference Rules)**.** *Given a set of information instances, S, and an information instance F, an inference rule specifies a relationship where any value set for S, i.e., $\{f_i : F_i \in S\}$, uniquely determines the value of F (i.e., f), or written as $S \Rightarrow F$.* □

For example, $\{F_R(r_1, r_2), F_G(r_2)\}$ (i.e., the relative position between r_1 and r_2 and the global position of r_2) can be used to infer $F_G(r_1)$ (i.e., the global position of r_1). See Tab. I for a few more examples of such rules.

A. Problem Formulation

The problem setting of multitasking robots (MT) with multi-robot tasks (MR) with instantaneous assignments[2] is:

Definition III.4 (MT-MR Setting)**.** *A MT-MR setting is given by a set $\{S_i\}$, where S_i is the set of physical constraints to be satisfied by the coalition for task t_i.*

The problem of enabling MT-MR becomes that of determining whether there *always* exists a physical configuration of the robots that satisfies all the constraints simultaneously:

Definition III.5 (Compatibility)**.** *A set of constraints S is compatible if there always exists a physical configuration for the referents in S such that all the constraints are satisfied.*

Instead of checking for compatibility, we check a set of constraints for incompatibility, which is easier. Intuitively, a set of physical constraints is incompatible if two constraints

[2]Our work differs from the task scheduling problem (i.e., with time extended assignments) in that we assume all tasks are executed simultaneously.

TABLE I: Examples of inference rules used in this work

Rule	Description
$\{F_G(X), F_R(Y, X)\} \Rightarrow F_G(Y)$	global position of X & relative position of Y to $X \Rightarrow$ global position of X
$\{F_R(Y, X)\} \Rightarrow F_R(X, Y)$	relative position of Y to $X \Rightarrow$ relative position of X to Y
$\{F_R(X, Z), F_R(Y, Z)\} \Rightarrow F_R(X, Y)$	relative position of X to Z & relative position of Y to $Z \Rightarrow$ relative position of X to Y

associated with the same information instance could be constrained by different values, thus leading to a conflict:

Claim III.1. *A set of physical constraints S is incompatible if any two constraints for the same information instance, as a result of S, could be constrained by different values.*

While the intuition is clear here, the implication of Claim III.1 is that we must be able to determine all the constraints that are *inferable* from S. This ability, in turn, requires the complete set of inference rules for a given domain and an inference process that is sound and complete. We make the implicit assumption here that the complete set of inference rules is given. When it does not hold, our inference process would still be sound in identifying incompatibility but no longer complete. Next, we incrementally develop such an inference process. First, we introduce a new concept:

Definition III.6 (Inference Closure)**.** *Denoting the inference closure of a set of information instances S by $\mathcal{C}(S)$, any $F \in \mathcal{C}(S)$ if:*

1) $F \in S$ *or*
2) $\mathcal{C}(S) \Rightarrow F$

Given the inference rules in Tab. I, for example, we can conclude that $\mathcal{C}(\{F_G(r_1), F_G(r_2)\}) = \{F_G(r_1), F_G(r_2), F_R(r_1, r_2), F_R(r_2, r_1)\}$. We refer to any information instance that is in $\mathcal{C}(S)$ as *inferable* from S, or that S infers it. Based on this definition, S trivially infers any instance already in S. Next, we formally define the notion of information inference based on inference closure:

Definition III.7 (Information Inference)**.** *Given two sets of instances S and S', an information inference is a relationship such that $\forall F \in S', F \in \mathcal{C}(S)$, written as $S \rightarrow S'$.* □

We use $\rightarrow$ to distinguish from $\Rightarrow$ used in inference rules. Note that $\rightarrow$ subsumes $\Rightarrow$. and is transitive by definition. When $S_1 \leftrightarrow S_2$, we refer to them as being equivalent sets. When S' contains a single information instance F above, we simply write $S \rightarrow F$. Intuitively, information inferences enable us to infer about *hidden* constraints given the existing ones. We show it more formally next:

Lemma III.2. *Given an information inference in the form of $S \rightarrow F$, if a set of constraints is defined over S, it also introduces a constraint on F.*

This follows almost immediately from the definition. Given a set of values for S, the value of F is determined from Def. III.7, which implies that F is also constrained according to Def. III.2. Similarly, if $S_1 \rightarrow S_2$, a set of constraints on S_1 also introduces a new set of constraints on S_2.

Definition III.8 (Minimally Sufficient Inference)**.** *$S \rightarrow F$ is a minimally sufficient inference if removing any instance from S would no longer infer F.*

Any inference rule is always assumed to be a minimally sufficient inference, since otherwise the rule can be simplified by removing the instances not required. We use $\rightarrow^*$ to denote a minimally sufficient inference. In this work, we focus on linear information systems where any inference rule specifies a linear relationship among the information instances:

Lemma III.3. *If all inference rules specify linear relationships among the instances, any information inference of the form $S \rightarrow F$ also specifies a linear relationship.*

Proof: Given that $S \rightarrow F$, there must exist a set of inference rules that are sequentially applied to infer F, in the forms of $S_1 \Rightarrow F_1, S_2 \Rightarrow F_2, \ldots, S_k \Rightarrow F_k (F = F_k)$, where $S_i \subseteq S \bigcup_{j<i}\{F_j\}$ and each $S_i \Rightarrow F_i$ specifies a linear equation between S_i and F_i. Since all the inference rules are assumed to be linear, we may replace F_i appearing after the ith rule with F_i's expression in the ith rule, which removes F_i from those equations. After performing this operation sequentially from $i = 1$ to $k - 1$, we end up with a linear expression of F using only instances in S.

The rules in Tab. I define a linear information system when the positions are all specified in the global coordinate system.

Lemma III.4 (Permutability)**.** *Assuming a linear information system, any minimally sufficient inference of the form $S \rightarrow^* F$ is permutable. In other words, it satisfies that $S \cup \{F\} \backslash F_x \rightarrow^* F_x$ for all $F_x \in S$.*

Proof: Given that $S \rightarrow^* F$ specifies a linear relationship (Lemma III.3), the linear expression of F using S as constructed in Lemma III.3 must use all the instances in S. Given a linear relationship, we can swap the positions of any F_x and F in the expression, and the result is still a valid linear equation for expressing F_x. Since F_x expressed by this equation is uniquely determined by $S \cup \{F\} \backslash F_x$ collectively, by Def. III.8, we have $S \cup \{F\} \backslash F_x \rightarrow^* F_x$.

Lemma III.5. *Assuming a linear information system and two sets of constraints S_1 and S_2: if $S_1 \rightarrow^* F$ and $S_2 \rightarrow^* F$ and S_1 and S_2 are compatible, we must have $S_1 = S_2$.*

Proof: We prove it by contradiction. Suppose that S_1 and S_2 are compatible and $S_1 \neq S_2$. Given that $S_1 \rightarrow^* F$, we know that S_1 introduces a constraint on F given Lemma III.2. Hence, to ensure that S_2 is compatible, S_2 must compute the same value f for F.

If S_1 and S_2 are the same, the conclusion trivially holds.

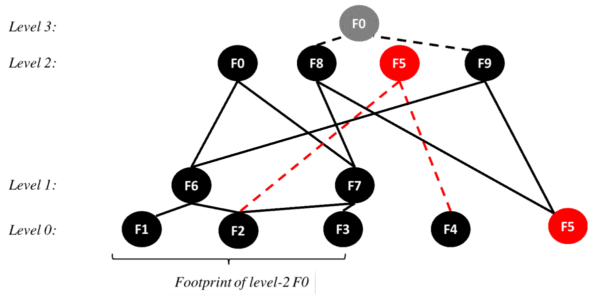

Fig. 2: Illustration of the multi-level graphical structure constructed (from the bottom up) to determine whether a set of constraints is compatible. The nodes are labeled by information instances. It shows two cases where a duplicate node is found: one for F_5 and one for F_0. The F_0 at level 3 is not added and it does not lead to incompatibility. The F_5 at level 2 however leads to incompatibility. In the actual implementation, our algorithm would stop at level 2 after the incompatibility is detected.

Otherwise, if $S_1 \supset S_2$, it results in a contradiction given that both $S_1 \rightarrow^* F$ and $S_2 \rightarrow^* F$ are minimally sufficient. Otherwise, S_2 must contain at least one instance $F_{2,\neg 1}$ that is not present in S_1. In such a case, we can set the value for $F_{2,\neg 1}$ in S_2 independently of S_1. Since updating the value of $F_{2,\neg 1}$ will change the value of F given Lemma III.4, it leads to a contradiction that S_1 and S_2 must compute the same value for F.

Theorem III.6. *Assuming a linear information system and given a MT-MR setting $\{S_i\}$ with non-overlapping S_i's, the following is a* ***necessary and sufficient*** *condition for $\{S_i\}$ to be incompatible:*

$$\exists S_1, S_2 \subseteq \bigcup_i S_i : S_1 \cap S_2 \nrightarrow F, S_1 \rightarrow F, S_2 \rightarrow F \qquad (1)$$

Proof: For sufficiency, we prove it by contradiction. In particular, we assume that there exist S_1 and S_2 that satisfy the above condition and they are compatible. Given that $S_1 \rightarrow F$ and $S_2 \rightarrow F$, we know that there exist subsets S_1^* and S_2^* of S_1 and S_2, respectively, such that $S_1^* \rightarrow^* F$ and $S_2^* \rightarrow^* F$. From Lemma III.5, we know that for them to be compatible (as a result of S_1 and S_2 being compatible), it must satisfy that $S_1^* = S_2^*$. This conflicts with the requirement that $S_1 \cap S_2 \nrightarrow F$.

For necessity, we must prove that the above condition must hold for all situations where $\{S_i\}$ is incompatible. Assume a situation where the condition does *not* hold and $\{S_i\}$ is incompatible. In such a case, there must exist two different sets $S_a \subseteq \bigcup_i S_i$ and $S_b \subseteq \bigcup_i S_i$ that both infer some F given Claim III.1, and that $S_a \cap S_b \rightarrow F$, given the assumption that the condition does not hold above. Denote S_a^* and S_b^* as the subsets of S_a and S_b, respectively, that satisfy $S_a^* \rightarrow^* F$ and $S_b^* \rightarrow^* F$. For S_a and S_b to be incompatible, without loss of generality, there must exist $S_a^! \subset S_a$ such that $S_a^! \neq S_a^*$ and $S_a^! \rightarrow^* F$. In such a case, we can choose S_1 and S_2 in the condition above as $S_a^!$ and S_a^* and it would satisfy, resulting in a contradiction.

Note that when the S_i's overlap, $\{S_i\}$ will be trivially incompatible by Claim III.1.

Example: Let us look at a simple example of how Theorem III.6 works in a cooperative navigation domain:

- One robot r_1 must provide localization assistance to two robots r_2 and r_3 to different goal locations as two tasks. Two coalitions are formed. The constraints ($\bigcup_i S_i$ in Theorem III.6) here are on the global positions of r_2 and r_3 (i.e., $F_G(r_2), F_G(r_3)$), and the relative positions from r_1 to r_2 and r_3 (i.e., $F_R(r_1, r_2), F_G(r_1, r_3)$).
- Two robots r_1 and r_2 must provide localization assistance to two robots r_3 and r_4 to their goal locations as two tasks, respectively. Two coalitions are formed. The constraints here are on the global positions of r_3 and r_4 (i.e., $F_G(r_3), F_G(r_4)$), as well as the relative positions from r_1 to r_2 and from r_3 to r_4 (i.e., $F_R(r_1, r_3), F_G(r_2, r_4)$), respectively.

For the first case, we can see that both $\{F_G(r_2), F_R(r_1, r_2)\}$ (first task) and $\{F_G(r_3), F_R(r_1, r_3)\}$ (second task) can be used to infer $F_G(r_1)$, and their intersection is $\varnothing$. Hence, they are not compatible according to Theorem III.6. For the second case, $\{F_G(r_1), F_R(r_1, r_3)\}$ (first task) and $\{F_G(r_2), F_R(r_2, r_4)\}$ (second task) do not infer the same F. In fact, these two tasks are trivially compatible since there is no multitasking involved. These observations also align with our intuition.

B. Solution Method

Brute-forcing is intractable since it requires checking all subset-pairs of $\bigcup_i S_i$, which is exponential. Instead, we propose a procedure based on a directed and multi-level graphical structure constructed from the bottom up:

- ***Level 0***: Make a node for each $F \in \bigcup_i S_i$ as leaves for the structure.
- ***Level i + 1*** ($i \geqslant 0$): For all inference rules that can be applied to the nodes at levels 0 to i, create a parent node for each instance F that can be inferred directly based on an inference rule, if F did not appear previously in the graph. Otherwise, we compute the intersection of its footprint (all descendant-leaf nodes, see Fig. 2) and that of the previous node, to see if it infers F. If so, continue with building the graph without adding the duplicate node; otherwise, return incompatible.
- ***Stopping criteria***: when no new nodes can be added, return compatible.

Fig. 2 shows an example of the graphical structure constructed to illustrate the compatibility detection process.

C. Solution Analysis

To analyze the complexity of the algorithm, we consider the following variables:

- number of inference rules, R
- number of robots, G
- number of information types, F
- maximum number of referents in instances, E

- maximum number of information instances on the left hand side of an inference rule, N

The maximum number of information instances is bounded by $O(FG^E)$. At any level i, the number of candidate rules to check is bounded by $O(RG^{EN})$. The total computation for constructing the graph is bounded by $O(FG^E RG^{EN})$. Hence, the computational complexity is only exponential in two constants (determined by the domain), or polynomial with respect to the number of robots.

Theorem III.7. *The solution method specified above is both sound and complete for detecting incompatibility in a MT-MR setting with linear information systems.*

This is a direct result of Theorem III.6 since the solution method essentially checks the condition described there. Note, however, when the system returns compatible, it does not guarantee that a physical configuration for satisfying the constraints at any point of time during the execution is always feasible due to, e.g., dynamic and environmental influences. However, assuming such influences are temporary and some fault-tolerance is built in the execution modules, a multi-robot system can often recover from those situations [27].

IV. Results

We introduced four types of UAV tasks that were considered in our experimental results. In this section, "*vehicle*" and "*robot*" are used interchangeably.

- **Monitoring task**: a target must be monitored within a close proximity by an air vehicle. The constraints for the monitoring task include the relative position (F_R) between the vehicle and target and the global position (F_G) of the target (since we have no control over it).
- **Centroid task**: a group of vehicles must maintain their centroid with respect to a specific location or target. The constraint is the centroid information defined over either 2 or 3 robots (denoted by F_{C_2} or F_{C_3}). The centroid information can be inferred from the global position information of the vehicles involved in the centroid task.
- **Formation task**: a group of vehicles must maintain their positions relative to each other. The formation has a designated leader, which may change. The constraints for this task are the relative positions (F_R) between the agents in the formation with respect to the leader.
- **Communication maintenance task**: a vehicle must maintain its position in between two other vehicles to maintain communication links. The constraint here is the communication maintenance information (denoted by F_M) that takes 3 referents, which can be derived from the relative positions between vehicles 1 and 3, and between 2 and 3, assuming that 3 is the maintainer.

A. Synthetic evaluation

In this experiment, we tested with the first three types of tasks only. The goal is to see how beneficial our approach is under resource constrained situations. We ran several experiments to determine the efficacy of our synergistic approach compared to a baseline. In the baseline, vehicles were assumed to be single-tasking, and hence would not accept new tasks once they were assigned to a task. The baseline is clearly inefficient since two vehicles that are assigned to a centroid task could still take on a monitoring task (similar to Fig. 1). In our experiments, we randomly generated a set of tasks for a given set of agents, with tasks alternating between centroid, formation, and monitoring, until every task was attempted and assigned if possible based on each method.

In Fig. 3, we set out to evaluate how our approach performs as the number of vehicles or tasks varies. We show the number of vehicles required to attempt all tasks in a low-demand (resource abundant) environment and a high-demand (resource constrained) environment, and a comparison of how a task environment with increasingly complex tasks affects the performance of each approach in Fig. 3. For the low-demand environment, our experiment generated a number of random centroid and formation tasks equal to $\frac{1}{5}$ of the number of vehicles, and a number of random monitoring tasks equal to $\frac{1}{2}$ of the number of vehicles, with random vehicles associated with each random task. For the high-demand environment, our experiment generated a number of random centroid and formation tasks equal to the number of vehicles, and a number of random monitoring tasks equal to three times the number of vehicles, with random vehicles associated with each random task. For the third experiment, we used 25 vehicles and varied what percentage of tasks required more than one vehicle to be performed. Each data point was given 50 or more iterations and the results were averaged. Our synergy method performed significantly better than the baseline, with roughly 50% more tasks assigned in most cases. In nearly every case, the synergy method assigned as many tasks as it had vehicles available. The baseline method assigned roughly 60% as many tasks as it had vehicles available. In every task environment examined, our approach resulted in more tasks assigned than the baseline approach. Also, the standard deviation of the synergy method is lower because the synergy method always assigned about the same number of tasks as vehicles.

B. Simulation Scenario

1) Simulation Environment and Settings: The OpenAMASE simulation environment was developed by the Air Force Research Lab [28] as a testing ground for their aerial vehicle control software, UxAS [29]. AMASE is a realistic simulator that models 5-DOF (coordinated turning) flight dynamics with self-configured performance. AMASE also handles piloting vehicles to waypoints. Together, these two pieces of software form the simulation environment. AMASE gives access to a GUI and simulates the vehicles over time, and UxAS handles the passing of all relevant messages to and from all modules of the software. Any module can subscribe to any type of message, and will then receive any message of that type sent by any other module. Our software uses the AirVehicleState, containing a "heartbeat" of information about each vehicle for each simulation tick, and the AirVehicleConfiguration, containing capability information about each vehicle, to decide where vehicles should move to.

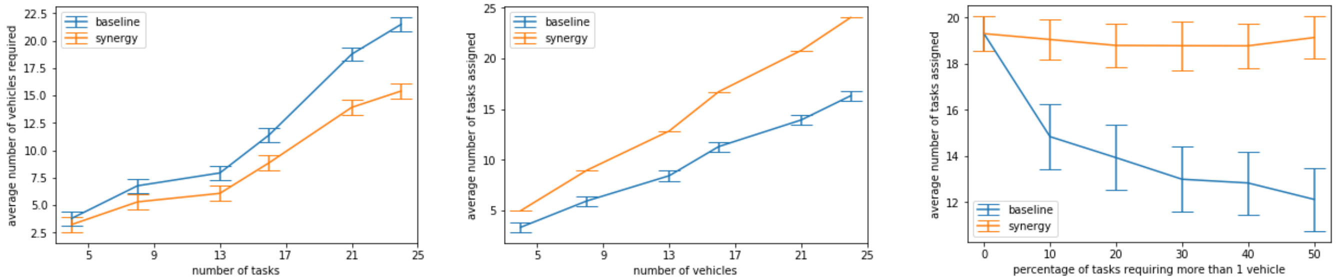

Fig. 3: (left) The number of robots required by each method under a low-task environment (roughly as many vehicles needed in tasks as there are robots) to attempt every task; (middle) the number of tasks assigned by each method given a fixed number of robots and an extremely task-dense environment; (right) a comparison between the two approaches with a fixed number of tasks and robots, but a dynamic ratio of single-robot tasks to multi-robot tasks. Note that for the left plot, lower values are more optimal, and for the middle and right plots, higher values are more optimal.

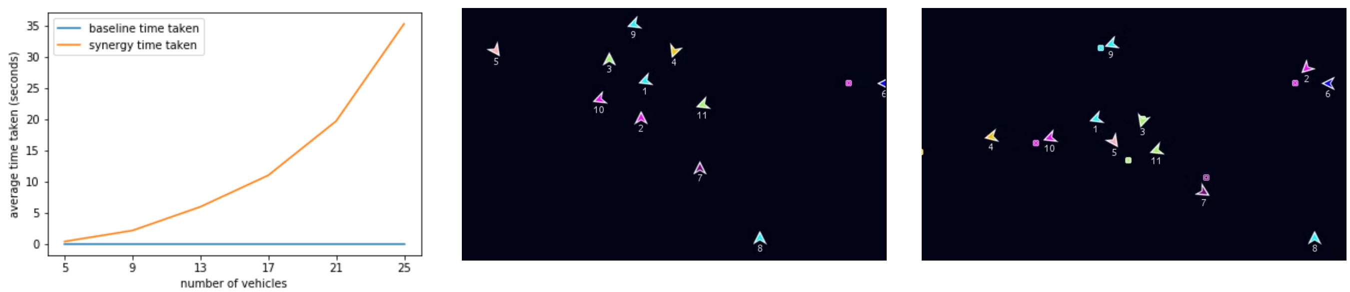

Fig. 4: (left) Computational time analysis of the synergy method (polynomial) and baseline; (middle and right) screenshots of the simulation scenario. The middle screenshot is the starting state of the scenario, and the right is the end state.

2) Simulation Result: In this simulation, we tested our system on a realistic scenario involving a multi-vehicle convoy task with intruder detection. The aim here is to test how our system extends the capability of the state-of-the-art in complex task scenarios. We have simulated 8 controlled vehicles, 2 intruder vehicles, and 1 static vehicle that approximates a control station. Fig. 4 shows snapshots of the start and end states of the simulation. The complete execution of this scenario is submitted as a video attachment.

A convoy is centered around vehicle 1 and protected by three vehicles $(2, 3, 4)$, which are assigned to a centroid task with vehicle 1 as the dynamic centroid. For centroid tasks, we first have any vehicles that are also executing another task calculate their target positions. Any remaining vehicles - at least 1 vehicle per centroid must be "*free*" in this way to satisfy task consistency - will then adjust their target positions to maintain the centroid once all vehicles in the centroid group reach their target positions. Vehicles 1, 9, 10, and 11 are in a formation task. Vehicle 1 is following a set path to its destination, effectively assigning it to a "convoy" task. For formation tasks, we allow all vehicles in the group to loiter in position until one of them is assigned to another task, in which case that vehicle becomes the leader of the group and the rest will maintain their positions relative to the leader. In this case, the vehicles 9, 10, and 11 maintain their positions relative to vehicle 1.

In the meantime, vehicle 7 must maintain communication between the convoy and a ground control station (simulated by a stationary vehicle 8). Additionally, two intruders are detected and two of the vehicles (2 and 3) that are already assigned the centroid task take advantage of the synergy between centroid and monitoring tasks by executing these two tasks at the same time. This assignment is consistent because vehicle 4 is free to move to maintain the centroid, even though 2 and 3 are constrained by their respective monitoring tasks.

V. Conclusions

A fundamental limitation of the existing multi-robot systems is that robots are assumed to be single-tasking. In this paper, we set out to address this limitation by enabling multitasking robots in multi-robot tasks. Inspired by the information invariant theory, we modeled physical constraints as information instances. This allowed us to model the interaction between the physical constraints to determine when they could be synergistically satisfied by reasoning about the relationship between information requirements. We showed that our algorithm was sound and complete for linear information systems, under the assumption that all information inference rules were provided. Simulation results were provided to show the effectiveness of our approach under resource-constrained situations and in handling challenging scenarios in a realistic multi-UAV simulator.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This research is supported in part by the NSF grant IIS-1844524, the NASA grant NNX17AD06G, and the AFOSR grant FA9550-18- 1-0067.

REFERENCES

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

[2] B. R. Donald, "On information invariants in robotics," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 217–304, 1995.

[3] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[4] L. Liu and D. A. Shell, "Large-scale multi-robot task allocation via dynamic partitioning and distribution," *Autonomous Robots*, vol. 33, no. 3, pp. 291–307, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10514-012-9303-2

[5] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 165–200, 1998.

[6] L. Vig and J. Adams, "Multi-robot coalition formation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 637–649, 2006.

[7] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme, "Coalition structure generation with worst case guarantees," *Artificial Intelligence*, vol. 111, no. 1-2, pp. 209–238, 1999.

[8] J. A. Adams *et al.*, "Coalition formation for task allocation: Theory and algorithms," *Autonomous Agents and Multi-Agent Systems*, vol. 22, no. 2, pp. 225–248, 2011.

[9] Y. Zhang and L. E. Parker, "Considering inter-task resource constraints in task allocation," *Autonomous Agents and Multi-Agent Systems*, vol. 26, no. 3, pp. 389–419, 2013.

[10] S. Liemhetcharat and M. Veloso, "Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents," *Artificial Intelligence*, vol. 208, pp. 41–65, 2014.

[11] R. Arkin, "Motor Schema – Based Mobile Robot Navigation," *International Journal of Robotics Research*, vol. 8, no. 4, pp. 92–112, Aug. 1989.

[12] S. Botelho and R. Alami, "M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1234–1239, 1999.

[13] M. Dias and A. Stentz, "A free market architecture for distributed control of a multirobot system," in *Proceedings of the 6th International Conference on Intelligent Autonomous Systems*, 2000, pp. 115–122.

[14] C. Fua and S. Ge, "COBOS: Cooperative backoff adaptive scheme for multirobot task allocation," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1168–1178, 2005.

[15] B. Gerkey and M. Mataric, "Sold!: Auction methods for multi-robot coordination," *IEEE Transactions on Robotics and Automation, Special Issue on Multi-robot Systems*, vol. 18, no. 5, pp. 758–768, 2001.

[16] N. Kalra, D. Ferguson, and A. Stentz, "Hoplites: A market-based framework for planned tight coordination in multirobot teams," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[17] L. Parker and F. Tang, "Building multirobot coalitions through automated task solution synthesis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1289–1305, Jul. 2006.

[18] Y. Zhang and L. Parker, "IQ-ASyMTRe: Synthesizing coalition formation and execution for tightly-coupled multirobot tasks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[19] ——, "IQ-ASyMTRe: Forming executable coalitions for tightly coupled multirobot tasks," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 400–416, 2013.

[20] R. Zlot and A. Stentz, "Complex task allocation for multiple robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Apr. 2005, pp. 1515–1522.

[21] L. E. Parker, "Cooperative robotics for multi-target observation," *Intelligent Automation & Soft Computing*, vol. 5, no. 1, pp. 5–19, 1999.

[22] O. Shehory and S. Kraus, "Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents," in *Proc. of ICMAS-96*. Citeseer, 1996, pp. 330–337.

[23] V. D. Dang, R. K. Dash, A. Rogers, and N. R. Jennings, "Overlapping coalition formation for efficient data fusion in multi-sensor networks," in *AAAI*, vol. 6, 2006, pp. 635–640.

[24] F. Tang and L. E. Parker, "Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 1501–1508.

[25] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[26] B. Tovar, L. Guilamo, and S. M. LaValle, "Gap navigation trees: Minimal representation for visibility-based tasks," in *Algorithmic Foundations of Robotics VI*. Springer, 2004, pp. 425–440.

[27] Y. Zhang and L. E. Parker, "A general information quality based approach for satisfying sensor constraints in multirobot tasks," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 1452–1459.

[28] P. Air Force Research Laboratory, Aerospace System Directorate and C. Division. (2020) Openamase, aerospace multi-agent simulation environment. [Online]. Available: https://github.com/afrl-rq/OpenAMASE/

[29] ——. (2020) Openuxas, project for multi-uav cooperative decision making. [Online]. Available: https://github.com/afrl-rq/OpenUxAS