Achieving Multi-Tasking Robots in Multi-Robot Tasks

Yu Zhang and Winston Smith

Abstract—One simplifying assumption made in distributed robot systems is that the robots are single-tasking: each robot operates on a single task at any time. While such a sanguine assumption is innocent to make in situations with sufficient resources so that the robots can operate independently, it becomes impractical when they must share their capabilities. In this paper, we consider multi-tasking robots with multi-robot tasks. Given a set of tasks, each achievable by a coalition of robots, our approach allows the coalitions to overlap and task synergies to be exploited by reasoning about the physical constraints that can be synergistically satisfied for achieving the tasks. The key contribution of this work is a general and flexible framework to achieve this ability for multi-robot systems in resource-constrained situations to extend their capabilities. The proposed approach is built on the information invariant theory, which specifies the interactions between information requirements. In our work, we map physical constraints to information requirements, thereby allowing task synergies to be identified via the information invariant framework. We show that our algorithm is sound and complete under a problem setting with multi-tasking robots. Simulation results show its effectiveness under resource-constrained situations and in handling challenging situations in a multi-UAV simulator.

I. INTRODUCTION

To address a multi-robot task, one simplifying assumption made in the literature is that the robots are single-tasking. This assumption, however, is impractical in situations where the robots must coordinate closely to share their capabilities, such as when a robot has a capability that is required in multiple tasks. To handle such resource-constrained situations, a simple solution is to achieve the tasks sequentially. Unfortunately, such a solution, besides having a negative impact on task efficiency, is feasible only when no concurrent execution is required among the tasks for which capabilities must be shared. As a result, it significantly limits the capabilities of multi-robot systems. Instead, in this paper, we consider multi-tasking robots in multi-robot tasks to enable a robot to operate on multiple tasks at the same time.

While multi-tasking robots are desirable when there are resource contents, the fundamental question regarding its feasibility must be carefully considered. In particular, it is affected heavily by the compatibility of the physical constraints to be satisfied for achieving the tasks. For example, for a robot to share its localization capability, 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 that is assigned to a monitoring task, it must maintain its camera head direction towards the target. Hence, the main challenge to enable multi-tasking robots lies in identifying synergies between the underlying physical constraints for the tasks.

The proposed approach is built on the information invariant theory [1] that specifies interactions between information requirements. In our work, the physical constraints to be satisfied are mapped to information instances, which are categorized by information types. Fig. 1 illustrates a scenario with two tasks: one of them is a centroid task that requires the three robots to maintain their centroid over the base station, which is specified as a constraint on the centroid information over the three robots; the other one is a monitoring task that requires one of the robots to maintain a target within its observation range, which is specified as a constraint on the relative position information from the monitoring robot to the target. The interactions between the constraints can then be described as information interactions, which allows us to identify task synergies using the information invariant framework. More specifically, given a set of physical constraints for the tasks, our approach checks if they are compatible according to a set of rules that govern information interactions. A task synergy is identified, for example, when the constraints for different tasks can be satisfied simultaneously even when there are shared resources (i.e., robots) among them. Fig. 1 illustrates a synergy between the two tasks.

To the best of our knowledge, our work represents the first general framework for achieving multi-tasking robots in multi-robot tasks [2]. It removes a restrictive assumption made in multi-robot systems by enabling overlapping coalitions. A formal framework is presented based on the information invariant theory. Simulation results show that our approach not only achieves better efficiency but also extends the capabilities of such systems in a multi-UAV simulator, especially in situations where resources are limited.

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

Much effort in multi-robot systems has been dedicated to the task allocation problem [2]. 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]. Addressing multi-robot tasks, however, significantly increases the problem complexity [2], for which there has been a noteworthy amount of focus on approximate solutions [5], [6], [7], [8], [9], and realistic concerns while implementing task allocation with distributed robot systems [10], [11], [12], [13], [14], [15], [16], [6], [17], [18], [19]. As far as we know, there exists little discussion on generalizing the problem to multi-tasking robots with instantaneous assignments [2]. Although the focus here is not on task allocation, it is an interesting direction to study how to integrate our method with task allocation algorithms.

The ability to achieve multi-tasking robots may appear similar to the existing notions of task synergy [20], [21] and overlapping coalition structure [22], [23]. However, these prior methods concern mainly with the optimization problem of utility maximization with the influence between the assignments given a priori, i.e., how the assignment of a coalition to a task may contribute to the overall utility given the other assignments. Such influence is often assumed to be captured by a pre-specified heuristic function that is difficult to compute in real-world settings. For one, modeling task synergy involves complex reasoning about the physical constraints as we argued. Our study goes beyond prior work by explicitly modeling the influence of overlapping coalitions on task feasibility due to the consideration of physical constraints, and thus addresses the fundamental question of whether task synergies are present. A similar observation was made in [6] about the impact of physical constraints on task allocation and the issue was addressed by manually specifying a feasibility function. A general framework that can automatically reason over the constraint space is missing.

The information invariant theory is introduced to capture the equivalence of sensori-computational systems [1]. It has since then been used to develop systems that have demonstrated an impressive level of flexibility [18], [24]. Information invariant is well connected to the notion of information space [25], where both must reason about the relationships between different information requirements. The difference being that the latter is often focused on the minimalistic aspect [26]. To the best of our knowledge, however, we are the first to apply the information invariant theory to reason about synergies between multi-robot tasks.

III. APPROACH

A physical constraint in our work is modeled as an information instance or a combination of information instances, which are categorized by information types. Next, we first formally introduce these two notions.

Definition III.1 (Information Instance). An information instance, denoted as $F(E)$, captures the semantics of information where $F$ is the information type and $E$ 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 actual information. For example, $F_R(r_1, r_2)$ is used to refer 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 a specific value of this information. For brevity, we often 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 of an information instance $F$.

Notice that the exact value for a constraint may depend on the environment settings and robot configurations dynamically, and hence is not always specified a priori. For example, in a tracking task, a constraint specified with respect to the target may be influenced by the environment settings (e.g., whether occlusions are likely to occur). This value is assumed to be determined by the execution module (which is not the focus of this work). To infer about information invariant, we define information inference:

Definition III.3 (Inference Rules). Given a set of information instances, $S$, and an information instance $F$, an inference rule defines a relationship such that any value set for $S$, i.e., \( \{f_1 : F_1 \in S\} \), uniquely determines the value of $F$ (i.e., $f$), or written as $S \Rightarrow F$.

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

A. Problem Formulation

The setting of multi-tasking robots with multi-robot tasks (with instantaneous assignment) is given as follows:

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

We further stipulate that the task requirements, manifested as physical constraints, are independent among themselves, which depend only on the environment, task and robot configurations. For example, in our motivating example (Fig. [1]), the constraint introduced by the monitoring task is solely a requirement of the monitoring task, and independent of the centroid task. The problem of enabling MT-MR thus becomes that of determining whether there exists a physical configuration of the robots that satisfies all these constraints at the same time:

Definition III.5 (Compatibility). A set of constraints $S$ is...
TABLE I
EXAMPLES OF INFECTION 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 $X$ ⇒ global position of $X$ |
| $F_R(Y, X) \Rightarrow F_R(X, Y)$ | relative position of $Y$ to $X$ ⇒ 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 $Z$ ⇒ relative position of $X$ to $Y$ |

-compatible if there exists a physical configuration for all the referents in $S$ such that all the constraints are satisfied.

To reason about compatibility, we first consider the inverse when a set of constraints is incompatible. Assuming that the physical constraints are independent and based on the definition above, the set of constraints becomes incompatible if two constraints associated with the same information instance are constrained by different values. We formally define this intuition in the following claim:

Claim III.1. Given a MT-MR setting $\{S_1\}, \{S_2\}$ is incompatible if no two constraints are restricted by the same information with different values.

While the above claim provides a sufficient condition for a set of constraints to be incompatible, it is not a necessary condition. For necessity to hold, however, a process must exist such that two constraints restricted by the same information with different values will always be found whenever a physical configuration does not exist for a given set of constraints. This requires us to reason about the equivalences between constraint systems, or in our case information systems, which are exactly what the information invariant theory [1] deals with! The underlying assumption for realizing this process is the ability to identify all information instances that are relevant and can be derived (or inferred) from known information in the domain. To achieve this, we must first assume that a complete set of inference rules is identified in a domain, which is an implicit assumption made throughout the paper. Next, we incrementally develop a mechanism to implement this process for our problem settings. First, we introduce a new concept:

Definition III.6 (Inference Closure). Denoting the inference closure of a set of information instances $S$ by $C(S)$, any $F$ belongs to $C(S)$ if:

1) $F \in S$ or
2) $C(S) \Rightarrow F$

Note the recursive definition here. Given the inference rules in Tab. I, for example, we can conclude that $C(\{F_G(r_1), F_G(r_2)\}) = \{F_G(r_1), F_G(r_2), F_R(r_1, r_2), F_R(r_2, r_1)\}$. We also refer to any information instance that is in $C(S)$ as “inferable” from $S$, or that $S$ infers it. Based on this definition, $S$ trivially infers any instance already in $S$. We use $\Rightarrow$ to distinguish it from that used in inference rules. Note, however, that $\Rightarrow$ subsumes $\Rightarrow$ is clearly transitive by definition. Next, we more formally define the notion of information inference that links us to inference closure:

**Definition III.7** (Information Inference). Given a set of information instances, $S$, and an information instance $F$, an information inference defines a relationship such that any value set for $S$, i.e., $\{s : F \in S\}$, uniquely determines the value of $F$ (i.e., $f$), or written as $S \Rightarrow F$.

Note the similarity between inference rule and information inference. When $S_1 \Rightarrow S_2$, we refer to them as being equivalent sets. Intuitively, information inference enables us to reason about the constraints that are implicitly “required” as a result of the given set of constraints. 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. A derivative of this result is that if $S_1 \Rightarrow S_2$, a set of constraints on $S_1$ also introduces a 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 in this work, since otherwise the rule can be simplified by removing the instances that are not required. We use $\leftarrow*$ to denote a minimally sufficient inference. In the following, we further simplify our discussion by assuming linear information systems:

**Lemma III.3** (Linear Information Systems). Assuming all inference rules specify linear relationships among the information 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 derive $F$, in the forms of $S_1 \Rightarrow F_1, S_2 \Rightarrow F_2, \ldots, S_k \Rightarrow F_k(F)$, where $S_i \subseteq S \cup_{j<i} S_j$. Since all the inference rules are assumed to be linear, we may replace $F_i$ appearing after the $i$th rule using the $i$th rule for expressing $F_i$, which removes $F_i$ from the equations. After performing this operation sequentially from $i = 1$ to $k - 1$, we end up with an expression of $F$ using only $S$.

The rules in Tab. I define a linear information system.

**Lemma III.4** (Permutability). Assuming a linear informa-
Fig. 2. Illustration of the multi-level graphical structure constructed (from the bottom up) to determine whether a MT-MR setting is compatible. The nodes are labeled by information instances. It shows two cases where a duplicate node is found: one for $F_2$ 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 already stop at level 2 after the incompatibility is detected.

\[ S_1 \cap S_3 \rightarrow F, S_1 \rightarrow F, S_2 \rightarrow F \text{ where } S_1, S_2 \subseteq \bigcup_i S_i \quad (1) \]

**Proof.** For sufficiency, we prove it by contradiction. In particular, we assume that there exist $S_1$ and $S_2$ that satisfy the above conditions and they are compatible. Given that $S_1 \rightarrow F, S_2 \rightarrow F$, we know that there exist subsets $S^*_1$ and $S^*_2$ of $S_1$ and $S_2$, respectively, such that $S^*_1 \rightarrow* F, S^*_2 \rightarrow* F$.

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 fact that $S_1 \cap S_2\rightarrow F$.

For necessity, we must prove that the above conditions must hold for all situations where $\{S_i\}$ is not compatible. Assume a situation where the conditions do 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 infer $F$ (given Claim III.1 and the assumption about a complete set of inference rules), and that $S_a \cap S_b \rightarrow F$ (given the assumption above). In which case, however, $S_a$ and $S_b$ must be compatible for $F$, resulting in a contradiction.

Note that when $S_i$’s overlap, they are trivially incompatible by Claim III.1.

**B. Solution Method**

Brute-forcing the solution is clearly intractable since it requires checking all subset-pairs of $\bigcup_i S_i$, which is exponential. Instead, we propose the following 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 $i + 1$ ($i \geq 0$):** For all inference rules that can be applied to the nodes at levels 0 to $i$, make a parent node for each instance $F$ that can be inferred directly based on an inference rule, if this instance does 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 created, 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 agents, $G$
- 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 \( O(RG^{EN}) \). Hence, the total computation for constructing the graph is bounded by \( O(FRG^{EN}) \). Hence, the computational complexity is only exponential in two constants (determined by the domain), and otherwise polynomial with respect to the number of agents.

**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 implements the same checking process described there. Note, however, when the system returns compatible, it does not necessarily mean that there exists a physical configuration in the current situation, since the environment may also affect the compatibility. However, assuming that the influence of the environment is temporary, a system should be able to recover from an incompatible state. Further discussion on this is delayed to future work.

### IV. Results

#### A. Tasks Considered

We introduced three types of UAV tasks that were considered in our experimental results.

- **Monitoring task:** a target must be monitored within a close proximity by an air vehicle. The constraint for the monitoring task is the relative position \( (FR_T) \) between the vehicle and target, and the global position \( (FR_S) \) of the target (since we do not have 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 \( FC_2 \) or \( FC_3 \)). The centroid information can be derived (inferred) from the global position information of the vehicles involved in the centroid task.

- **Communication maintenance task:** a vehicle must maintain its position in between two other vehicles to maintain communication links. The constraint introduced here is the communication maintenance information (denoted by \( FM \)) that takes 3 vehicles, which can be derived from the relative positions between vehicles 1 and 3, and 2 and 3, assuming that 3 is the vehicle that is between the other two.

#### B. Synthetic evaluation

In this experiment, we tested with the first two 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 did not accept new tasks once they were assigned to a task. In our experiments, we randomly generated sets of tasks for specific sets of agents, and attempted to assign tasks alternately between centroid and monitoring tasks to vehicles, until every generated task has been attempted.

In Fig. 4 (top), we studied the differences of the assignment process between the two approaches in more detail. We generated a new task at each iteration of the assignment process and checked how many tasks were allocated in each iteration accumulatively. We can see that at very low numbers
of tasks relative to the vehicles, the synergy and baseline task assignments assigned the same number of tasks, which was to be expected since we had many more vehicles available. However, as more tasks were added to the assignment, the synergy method pulled ahead of the baseline. Also, our method consistently assigned more tasks in each iteration. In Fig. 4 (bottom), we studied the influence of the ratio between the numbers of the monitoring tasks and centroid tasks in detail. We randomly generated 25 tasks with the ratio between the tasks gradually decreasing, so that initially we had 25 monitoring and 0 centroid task, and 0 monitoring and 25 centroid task at the end. We can see that our method was affected a lot less by the baseline. The performance gap kept increasing as this ratio decreased.

C. Simulation Scenario

1) Simulation Environment and Settings: The OpenA-MASE simulation environment was developed by the Air Force Research Lab[27] as a testing ground for their aerial vehicle control software, UxAS[28]. Together, these two pieces of software form the basis of 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.

2) Simulation Result: In this simulation, we tested our system on a realistic scenario involving a multi-vehicle convoy task with intruder detection. AMASE and UxAS can only handle up to twelve vehicles, and we have limited our simulation to 8 controlled vehicles, 2 intruder vehicles, and 1 static vehicle that simulates a control station. Fig. 5 shows snapshots from running the task. A convoy is formed (centered around vehicle 1) and protected by three vehicles (2, 3, 4), which are assigned to a centroid task with vehicle 1 as the centroid. In the mean time, vehicle 7 must maintain the communication between the convoy and a ground control station (vehicle 8). As the convoy is moving towards its target, 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 two tasks at the same time.

V. Conclusions

In this paper, we set out to address the problem of multi-tasking robots in multi-robot tasks. We observed that the key underlying challenge was to reason about the physical constraints that could be synergistically satisfied. To address the challenge, we developed our method based on the information invariant theory and modeled constraints as information instances. Thereby, a formal and general framework to achieve multi-tasking robots was developed. We showed that our algorithm was sound and complete under our problem settings. Simulation results were provided to show the effectiveness of our approach under resource-constrained situations and in handling challenging situations.
REFERENCES

[1] B. R. Donald, “On information invariants in robotics,” Artificial Intelligence, vol. 72, no. 1-2, 1995.
[2] B. P. Gerkey and M. J. Mataric, “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.
[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, 1998.
[6] L. Vig and J. Adams, “Multi-robot coalition formation,” IEEE Transactions on Robotics, vol. 22, no. 4, pp. 637–640, 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] R. Arkin, “Motor Schema – Based Mobile Robot Navigation,” International Journal of Robotics Research, vol. 8, no. 4, pp. 92–112, Aug. 1989.
[11] S. Botelho and R. Alumi, “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.
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] 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.
[18] ——, “IQ-AsyMTrE: Forming executable coalitions for tightly coupled multirobot tasks,” IEEE Transactions on Robotics, vol. 29, no. 2, pp. 400–416, 2013.
[19] 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.
[20] 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.
[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.