

# Robust Assignment Using Redundant Robots on Transport Networks with Uncertain Travel Time

Amanda Prorok

Department of Computer Science and Technology  
University of Cambridge, UK

asp45@cam.ac.uk

**Abstract**—This paper considers the problem of assigning mobile robots to goals on transport networks with *uncertain and potentially correlated* information about travel times. Our aim is to produce optimal assignments, such that the average waiting time at destinations is minimized. Since noisy travel time estimates result in sub-optimal assignments, we propose a method that offers robustness to uncertainty by making use of *redundant robot assignments*. However, solving the redundant assignment problem optimally is strongly NP-hard. Hence, we exploit structural properties of our mathematical problem formulation to propose a polynomial-time, near-optimal solution. We demonstrate that our problem can be reduced to minimizing a supermodular cost function subject to a matroid constraint. This allows us to develop a greedy assignment algorithm, for which we derive sub-optimality bounds. We demonstrate the effectiveness of our approach with simulations on transport networks with correlated uncertain edge costs and uncertain node positions that lead to noisy travel time estimates. Comparisons to benchmark algorithms show that our method performs near-optimally and significantly better than non-redundant assignment. Finally, our findings include results on the benefit of diversity and complementarity in redundant robot coalitions; these insights contribute towards providing resilience to uncertainty through targeted composition of robot coalitions.

**Note to Practitioners**—This paper is motivated by the problem of assigning mobile robots (e.g., vehicles, drones, etc.) to goals when travel times from robot origins to goal locations are uncertain. Existing robust assignment methods deal with uncertainty by minimizing risk, or by pre-defining acceptable risk thresholds. In this work, we propose a complementary method that offers robustness to uncertainty by making use of robot redundancy. In other words, we assign more robots than necessary to a given goal, in the expectation that one of the redundant robots will reach the goal faster (than the originally assigned robot). However, solving this redundant assignment problem is computationally intractable for large systems. By characterizing the mathematical problem, we show how the redundant assignment problem can be solved efficiently. We apply our assignment algorithm to transport network problems to reduce average waiting times at goal locations, when travel times from vehicle origins to destinations are uncertain and potentially also correlated. Our results show that exploiting robot redundancy is an effective approach to reducing waiting times. In this work, we build on the premise that time is the primary commodity, and we do not explicitly model the additional cost of utilizing redundant robots. Future work should more explicitly address the trade-off between the cost of providing redundancy (e.g., travel costs, robot costs) and performance gains.

**Index Terms**—Task assignment, multi-robot systems, submodular optimization.

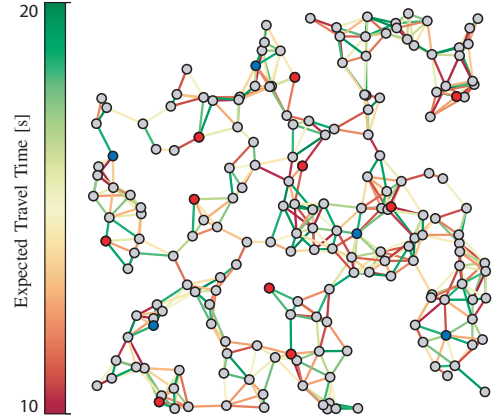


Fig. 1. An instance of a random transport network. Robots are initially located at hubs (red nodes). Task locations (goals) are marked by nodes colored in blue; The travel time along the edges of this transport network is uncertain, and is modeled by a correlated probability distribution that can capture dependencies across edges and nodes; the colormap indicates the expected travel time.

## I. INTRODUCTION

The optimal assignment of mobile robots to tasks at different locations is a fundamental problem that has many applications. In particular, large-scale deployment of robots has the potential of transforming the industries of transport and logistics. To orchestrate the coordination of the robots, centralized communication architectures have become the norm in various instances; representative applications include product pickup and delivery [19], item retrieval in warehouses [10], and mobility-on-demand services [29, 34]. In general, a solution to this problem can be computed by a centralized unit that collects all robot-to-task assignment costs (e.g., expected travel times) to determine the optimal assignment (e.g., by running the Hungarian algorithm). However, the optimality of this assignment hinges on the accuracy of the assignment cost estimates. Despite our best efforts to model any uncertainties, discrepancies between model assumptions and real-life dynamics may arise. For example, in transport scenarios, a robot may encounter an unexpectedly blocked path, and consequently takes significantly longer to reach its destination than anticipated. Travel time uncertainty also arises due to the deterioration of positioning accuracy of GNSS-based solutions in urban canyons [2]. Furthermore, recent methods consider it *desirable* to actively obfuscate true robot

state information (e.g., robot positioning), to ensure privacy across a variety of applications [6, 31]. Yet irrespective of the source of uncertainty, it follows that any discrepancies around true robot states cause a degradation in the system’s overall performance, and can lead to cascading effects.

We are interested in applications that require a fast arrival of robots at their destinations. Due to uncertain robot locations or uncertain path conditions, the knowledge about expected travel times may be imprecise or incomplete. This compounds the difficulty of determining an optimal robot-to-goal assignment. Although assignments under random costs have gained a considerable amount of attention [21, 24, 25], the focus has primarily been on providing analyses of the performance under noisy conditions. In this work, we propose a complementary method that provides robustness to noisy travel time estimates by making use of *robot redundancy*. In other words, the core idea of our work is to exploit redundancy to counter uncertainty and redeem performance. Although the idea of engineering robust systems with redundant resources is not new in a broad sense [17, 22], we are the first to consider redundant mechanisms for the problem of mobile robot assignment under uncertainty, with arbitrary and potentially correlated probability distributions. We believe that providing redundant robots will be a fundamental design feature for systems where time is the primary asset (e.g., rescue scenarios), and where an over-provisioning with respect to the number of robots is a minor concern.

The premise of this work is that time is the most valuable asset in the system. Our idea is led by a ‘*first-come, first-to-serve*’ principle, by which only the fastest robot to reach a task actually services it. Redundancy allows us to reduce the waiting time at the goal locations: when multiple robots travel to the same destination, only the travel time of the fastest robot counts. Consequently, this paper presents a novel and efficient method by which robots are redundantly matched to individual goals, such that the average waiting time over all goal locations is minimized.

Our problem belongs to the class of Single-Task Robots, Multi-Robot Tasks (ST-MR) [16], which considers the assignment of groups of robots that have a combined utility for any given task. This task utility can be uniquely defined for a given robot group and task, and is not necessarily linear over the individual robot utilities. The aim is to split the set of robots to form task-specific coalitions, such that the average task utilities over all coalition-to-task assignments is maximized (or, in our case, the average cost minimized). Formally, this can be cast as a set-partitioning problem, which considers a set of robots that needs to be partitioned into a family of subsets with maximum utility over the set of tasks that they are assigned to. The set-partitioning problem is strongly NP-hard [15]. Research in this domain has generated heuristic solutions as well as methods that facilitate the combinatorial search [3, 7, 36]. However, if we pay special attention to the objective function, the problem may reveal additional structure that can be exploited to find near-optimal solutions. In particular, for utility functions that satisfy a property of *diminishing returns*, otherwise known as *submodularity*, near-optimal approximations can be found [14]. In one of the key

contributions of this work, we show that our approach provides diminishing returns in the number of redundant robots assigned to each goal, yielding a *supermodular* cost function. Thus, we pose a set optimization problem under cardinality constraints, and employ a method based on supermodular cost minimization under a matroid constraint. This framework leads to a polynomial-time algorithm that has a provable bound on the difference between the optimal waiting time and the waiting time resulting from our algorithm.

## II. RELATED WORK

Our problem is related to the general class of submodular welfare problems [23, 37], which stems from the field of combinatorial auctions. The welfare problem considers a set of items and a set of players, and seeks a partition of the items into disjoint sets assigned to players in order maximize the total welfare over the players. The *welfare* is equivalent to the sum of utilities over all sets. The utility functions satisfy the property of diminishing returns, and hence, the problem can be formulated as one of submodular maximization. In contrast to our work, the welfare problem does not prescribe any explicit form for the submodular function, nor does it explicitly consider uncertainty. Instead, it assumes a *value oracle model*, which is a black-box that returns the utility for any given set. In this sense, our problem is a specialization of the general submodular welfare problem. We consider a specific objective function, where the submodularity (or, in our case, the supermodularity) arises due to the redundancy of assigned robots with uncertain travel times. Furthermore, we also provide a specialization of the matroid constraint, which, in our case, relates to the constraint on maximum possible robot deployment sizes.

Another related body of work deals with the weapon-target assignment problem [1], which considers the assignment of weapons to targets so that the total expected survival value of the targets is minimized. Similar to our problem, this problem considers the assignment of a redundant number of items (i.e., weapons) to a single goal (i.e., target), where assignment costs are uncertain (i.e., probability of target survival). In other words, the compound probability of successfully hitting a target is increased as the number of resources (i.e., weapons) is increased. In contrast to our work, however, the weapon-target assignment problem only applies to independent noise from binomial distributions that model the outcome (i.e, survival vs. destruction) of each target as a Bernoulli random variable. Instead, we are interested in quantifying travel time, and our algorithm is capable of dealing with arbitrary (and potentially correlated) probability distributions that describe these assignment costs. In that sense, our problem is a generalization of the weapon-target assignment problem.

The work in [18] develops an adaptive optimization approach under partial observability of the cost function. The authors introduce the concepts of expected marginal gain and adaptive submodularity, which they leverage to develop an adaptive greedy algorithm. Although these theoretical results are amenable to the types of problems we are interested in (i.e., assignment under uncertainty), they are founded on an

orthogonal approach, which consists of adapting cost estimates online, instead of hedging against uncertainty by redundancy a-priori, as we do.

Submodular optimization for combinatorial problems has also gained considerable traction in the domain of multi-robot systems. Applications include coordinated robot routing for environmental monitoring [32], leader selection in leader-follower systems [8], sensor scheduling for localization [33], path planning for orienteering missions [20], exploration [9] and constrained task allocation [38]. Typically, the aforementioned studies develop explicit objective functions and constraints that are specific to the considered problem domains. The authors then go about proving the submodularity property, deriving sub-optimality bounds, and devising the appropriate assignment algorithms. This general methodology is similar to the one presented in our paper.

In summary, the particularity of our work lies in the specificity of our objective function for redundant robot assignments under uncertain travel times on random transport networks. Robot redundancy plays a key role in our theoretical developments; our submodularity property is tightly coupled to travel applications, and hinges on the ‘first-come, first-to-serve’ principle. Finally, our findings include results on the benefit of diversity and complementarity in redundant robot coalitions; these are unprecedented insights within this context, and contribute towards providing robustness against uncertainty through targeted composition of robot coalitions.

*Contributions.* The main contribution of this work is a supermodular optimization framework that selects redundant robot-to-goal matchings in order to minimize the average waiting time at the task locations. Additionally, we pose a variant of this optimization problem such that the number of assigned robots is minimized, subject to a maximum allowable cost budget. Our framework is underpinned by the novel insight that the deployment of redundant robots under uncertain travel times is supermodular in the number of redundant robots. This insight is derived from a *first-come, first-to-serve* principle, which is formally introduced in this paper through an aggregate function that considers only the *minimum* travel time among the assigned robot coalition. Furthermore, we formalize the cardinality and assignment constraints by a matroid. These combined results (optimization of a supermodular function under a matroid constraint) allow us to derive sub-optimality bounds on the performance of our method.

Our second contribution is the development of a dynamic programming algorithm that reduces the number of calls to the objective function, and allows us to compute task utilities efficiently in polynomial time. This algorithm leverages the concept of aggregate functions over robot coalitions, which we formally introduce in this work. A key consideration is that the performance at each goal is measured by an aggregate cost function that considers the joint performance of all assigned robots at that goal.

Finally, we apply the redundant assignment solution to random instances of transport networks in planar space. In particular, we consider two causes for uncertain travel time: uncertainty in the robot origin positions, and uncertainty in the road travel time of the transport network (which was

considered in our prior work [30]). We compare our method with an optimal, exhaustive combinatorial search algorithm to show that our method performs very close to the optimum. Our approach is also compared to several benchmark algorithms, and is evaluated under varying noise conditions. We show that in all considered cases, redundant assignment effectively increases robustness against uncertainty by reducing the performance gap to an ideal, non-noisy system.

### III. PROBLEM STATEMENT

We consider a system composed of  $M$  goals and  $N$  available robots. Our problem considers the assignment of robots to goals via a path, where, for each robot-to-goal assignment, multiple possible paths with random costs exist. We seek to find a minimum cost matching, such that all goals are covered, any goal may be assigned multiple robots, and each robot uses one of  $K$  possible paths to reach its assigned goal.

The mathematical model and algorithms presented in this paper are general, and can accommodate uncertainties defined on arbitrary task assignment problems. In order to provide specific examples and their implementations, we apply the assignment solution to a transport scenario (in planar space), where imprecise information about robot positions or road travel times leads to uncertainty in the robot travel time estimates (and hence, assignment cost uncertainty). In specific, we consider two problem variants (see Problem 1 and Problem 2). The first problem aims to minimize the number of deployed robots, while respecting a limit on the maximum admissible average assignment cost,  $\xi$ . The second problem aims to minimize the average assignment cost over all goals, while respecting a limit on the maximum deployment size  $N_d$ . For each of these problems, we consider that assignment cost uncertainty arises due to two main causes: (i) robot position uncertainty, and (ii) road travel-time uncertainty. Sections IV-A and IV-B elaborate these scenarios.

*Assumptions:* In this work, we make three key assumptions. First, we assume that an initial, non-redundant robot assignment is made by means of a standard assignment method (e.g., through the Hungarian algorithm). This assumption is key to enabling a tractable approach—the technical reasons are elaborated in further detail in Section III. Second, we assume that  $K$  possible paths exist between each robot and each goal, and that knowledge of these  $K$  feasible paths can be readily computed. We acknowledge that *not* assuming  $K$  paths—that is, evaluating *all* possible paths—would potentially lead to better solutions, however, this comes at the cost of computational complexity, hence our decision to limit this choice. Finally, we assume that it is possible to constrain our problem, either (i) by assuming a known budget  $\xi$  on the desired performance (see Problem 1), or else, (ii), by assuming that a limited number  $N_d$  robots are readily available for assignment (see Problem 2).

#### A. Redundant Assignment with Random Costs

Consider a graph  $\mathcal{B} = (\mathcal{U}, \mathcal{F}, \mathcal{C})$ . The set of vertices  $\mathcal{U}$  is partitioned into two subsets  $\mathcal{U}_r$  and  $\mathcal{U}_g$ , such that  $\mathcal{U}_r = \bigcup_i r_i$ ,  $i = 1, \dots, N$  contains all robot nodes,  $\mathcal{U}_g = \bigcup_j g_j$ ,  $j = 1, \dots, M$  contains all goal nodes, and



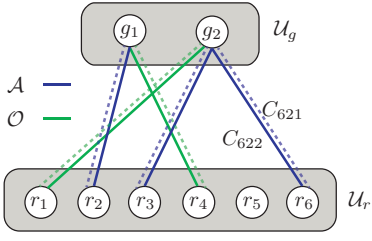


Fig. 2. Sketch of redundant robot-to-goal matching, for a total of  $N = 6$  available robots,  $M = 2$  goals,  $K = 2$  path options, and a deployment size of  $N_d = 5$ . Edges in  $\mathcal{O}$  represent the initial assignment, and edges in  $\mathcal{A}$  represent the redundant assignment. The random variable  $C_{ijk}$  represents the uncertain travel time it takes robot  $i$  to reach goal  $j$  via path  $k$ .

$\mathcal{U} = \mathcal{U}_r \cup \mathcal{U}_g$ ,  $\mathcal{U}_r \cap \mathcal{U}_g = \emptyset$ . We define a fixed number  $K$  of possible path routes that lead any robot to any goal. The edge set  $\mathcal{F} = \{(i, j, k) | i \in \mathcal{U}_r, j \in \mathcal{U}_g, k = 1, \dots, K, \forall i, j, k\}$  is complete, meaning that any robot can reach any goal node, and that up to  $K$  possible path choices exist for any pair  $(i, j)$ . We note that for  $K = 1$ , the problem becomes equivalent to a bi-partite graph matching problem with uncertain costs. The tuple  $(i, j, k)$  indicates that robot  $i$  is assigned to goal  $j$  through path  $k$ . Since the travel time for a robot to reach its goal is uncertain, we represent the weight of each edge  $(i, j, k)$  by a random variable  $C_{ijk} \in \mathcal{C}$ , where  $\mathcal{C}$  is a set of random variables. The set  $\mathcal{C}$  has a joint distribution  $\mathcal{D}$ . Hence,  $C_{ijk}$  can be arbitrarily defined for any edge (admitting  $C_{ijk} > 0$ ); in particular, edge costs may be correlated, and we do not make use of *i.i.d.* assumptions.

We consider the case where an initial non-redundant assignment has been made. The initial assignment is  $\mathcal{O} \subset \mathcal{F}$ , such that  $\forall j |\{i | (i, j, k) \in \mathcal{O}\}| = 1$ ,  $\forall i |\{j | (i, j, k) \in \mathcal{O}\}| \leq 1$ , and  $\forall i, j |\{k | (i, j, k) \in \mathcal{O}\}| \leq 1$ . In other words,  $\mathcal{O}$  covers every goal with one robot, and any robot is assigned to at most one goal through one given path. Given an initial assignment, the aim of this paper is to find an optimal set of assignments  $\mathcal{A}^* \subset \mathcal{F} \setminus \mathcal{O}$  for the remaining  $N_d - M$  robots<sup>1</sup>. From here on, we denote  $\mathcal{F} \setminus \mathcal{O}$  by  $\mathcal{F}_{\mathcal{O}}$ . Fig. 2 illustrates a simple robot-to-goal matching.

The main novelty of our approach is the use of redundant robots to help counter the adverse effect of uncertainty. If the system admits a sufficiently large number of robots (i.e.,  $N_d - M > 0$ ), we can assign multiple robots to the same goal, while still ensuring that all goals are assigned at least one robot. A key consideration is that the performance at each goal is measured by an *aggregate cost* function that considers the joint performance of all assigned robots at that goal.

**Definition 1** (Aggregate Cost). *We define an aggregate function  $\Lambda : 2^{\mathcal{F}} \mapsto \mathbb{R}$  that operates over the set of edges incident to a given node, and returns a scalar that represents the aggregate cost over the weights of these edges. If  $I_j(\mathcal{A})$  is the set of incident edges to node  $j$  in the set of edges  $\mathcal{A}$ ,*

<sup>1</sup>Without  $\mathcal{O}$ , any solution that is smaller in size than  $M$  would lead to an infinite waiting time, and hence, the objective function loses its supermodular property (meaning that our approach relies on this assumption). It follows that *without* assuming an initial assignment, we would have to fall back to a formulation of the classical set-partitioning problem, as introduced in Sec. I, which is strongly NP-hard. We further discuss the ramifications of our assumption in Sec. VIII.

and is equal to  $\{(i, j, k) | \forall i, k \text{ with } (i, j, k) \in \mathcal{A}\}$ , then we can write the aggregate cost for goal  $j$  as

$$\Lambda(I_j(\mathcal{A})). \quad (1)$$

We instantiate the definition for  $\Lambda$  later, in Sec. IV. In the following, we also refer to the set of all assigned robots to one goal as a *robot coalition*, following the terminology originally introduced in [5] and later adapted to the robot domain in [26].

## B. Optimization Problems

The definitions above allow us to formulate our objective function. For a given set  $\mathcal{A}$  of edges that define robot to goal assignments, we wish to measure the average aggregate cost over all goals, in expectation over the random assignment costs:

$$J_{\mathcal{O}}(\mathcal{A}) = \frac{1}{M} \sum_{j=1}^M \mathbb{E}_{\mathcal{C}} [\Lambda(I_j(\mathcal{A} \cup \mathcal{O}))]. \quad (2)$$

We note that when no redundant robots are deployed, the assignment is reduced to the set  $\mathcal{O}$ , for which the performance is measured as

$$J_0 = J_{\mathcal{O}}(\emptyset) = \frac{1}{M} \sum_{j=1}^M \mathbb{E}_{\mathcal{C}} [C_{ijk} | (i, j, k) \in \mathcal{O}]. \quad (3)$$

For practical reasons, we further specify our problem to constrain the average aggregate cost (i.e., the performance), while aiming to minimize the robot deployment size. We assume that this cost budget  $\xi$  is pre-defined by the user. Such a constraint is useful when a known performance guarantee is required, and when we wish to deploy the least possible redundant resources (i.e., robots) to meet this requirement. We formalize this first problem as follows.

**Problem 1.** *Optimal matching of redundant robots under a cost budget: Given  $N$  available robots,  $M$  goals with uncertain robot-to-goal assignment costs, and an initial assignment  $\mathcal{O}$ , find a minimally sized matching  $\mathcal{A} \subseteq \mathcal{F}_{\mathcal{O}}$  of redundant robots to goals, such that the average aggregate cost over all goals  $J_{\mathcal{O}}(\mathcal{A})$  is smaller than a given cost budget  $\xi \in \mathbb{R}$ . This is formally stated as:*

$$\underset{\mathcal{A} \subseteq \mathcal{F}_{\mathcal{O}}}{\operatorname{argmin}} \quad |\mathcal{A}| \quad (4)$$

$$\text{subject to} \quad \forall i |\{j | (i, j) \in \mathcal{A} \cup \mathcal{O}\}| \leq 1 \quad (5)$$

$$J_{\mathcal{O}}(\mathcal{A}) \leq \xi \quad (6)$$

Our second problem is closely related to the first. Instead of specifying a cost budget,  $\xi$ , we specify a cap on the maximum number of assignable robots (constrained by  $N_d$ , with  $M \leq N_d \leq N$ ). This consideration is particularly relevant when a known, fixed number of robots is readily available, or, when no meaningful cost budget can be derived. Relevant applications include those with continuous and fluctuating task demands. In such applications, the consideration of a limited number of assignable robots allows us to keep some robots in reserve for future deployments. It also allows operators to limit the cost of redundant robot deployments, for example, by monitoring and capping energy consumption. Formally, we capture this

additional constraint by a *matroid* [27] (see also Def. 6 and Sec. VI-A), which is an abstract structure that generalizes the notion of linear independence to set systems. We formalize this second problem as follows.

**Problem 2.** Optimal matching of redundant robots under cardinality constraints: *Given  $N$  available robots,  $M$  goals with uncertain robot-to-goal assignment costs, and an initial assignment  $\mathcal{O}$ , find a matching  $\mathcal{A} \subseteq \mathcal{F}_{\mathcal{O}}$  of redundant robots to goals such that the average cost over all goals is minimized, and the total number of robots deployed is  $N_d$ . This is formally stated as:*

$$\underset{\mathcal{A} \subseteq \mathcal{F}_{\mathcal{O}}}{\operatorname{argmin}} \quad J_{\mathcal{O}}(\mathcal{A}) \quad (7)$$

$$\text{subject to} \quad \forall i | \{j | (i, j) \in \mathcal{A} \cup \mathcal{O}\} | \leq 1 \quad (8)$$

$$|\mathcal{A}| = N_d - M \quad (9)$$

We note that, in addition to the applications listed above, Problem 2 is particularly relevant in scenarios where the loss incurred when *not* solving the problem effectively far outweighs the cost of deploying redundant robots. Such scenarios include applications of Search-and-Rescue (e.g., the cost of a life far outweighs the cost of a robot).

#### IV. APPLICATION TO TRANSPORT NETWORKS

We are interested in applications on transport networks, and use graphs to represent possible robot routes (from origins to goals). We represent routes via a weighted directed graph,  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ . Vertices in the set  $\mathcal{V}$  represent geographic locations. Nodes  $u$  and  $v$  are connected by an edge if  $(u, v) \in \mathcal{E}$ . We assume the graph  $\mathcal{G}$  is a strongly connected graph, i.e., a path exists between any pair of vertices. Paths between a same origin  $i$  and goal  $j$  are distinct if there is at least one edge in one path that is not present in the other path.

The random variable  $C_{ijk}$  captures the estimated travel time for robot  $i$  to reach a goal  $j$  via path  $k$ . The following two sections elaborate specific implementations of  $\mathbb{E}[C_{ijk}]$  for the two causes of uncertainty that we consider in this work. Furthermore, they elucidate a *first-come, first-to-serve* concept, by which only the fastest robot to reach a task actually services it. Redundancy, as defined in Def. 1, allows us to reduce the waiting time at the goal locations: when multiple robots travel to the same destination, only the travel time of the fastest robot counts. This is specified in the following definition.

**Definition 2** (Effective Waiting Time). *Since only the first robot's arrival defines the effective waiting time at a goal node  $j$ , the aggregate cost function  $\Lambda$  (see Def. 1) is equivalent to the minimum operator. The effective waiting time (cost) at goal  $j$  is*

$$\Lambda(I_j(\mathcal{A})) \triangleq \min\{C_{ijk} | (i, j, k) \in I_j(\mathcal{A})\}. \quad (10)$$

##### A. Application to Transport with Uncertain Travel Time

We consider a robot  $i$  at node  $r_i$  and a goal  $j$  at node  $g_j$ . We define a random variable  $w_{uv} \in \mathcal{W}$  that represents the time needed to traverse an edge  $(u, v)$ . The set of weights

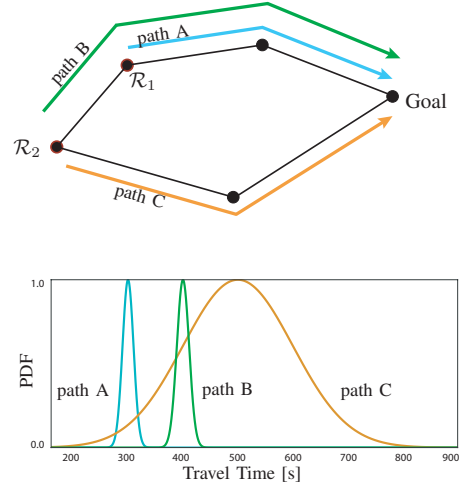


Fig. 3. Example scenario. Robot  $\mathcal{R}_1$  is assigned to the goal via path A. Robot  $\mathcal{R}_2$  is a redundant robot, and must choose between paths B and C.

$\mathcal{W}$  can be sampled from a distribution  $\mathcal{D}_w$ . We formulate the estimated travel time from node  $r_i$  to goal  $g_j$  on path  $k$  as:

$$C_{ijk} = \sum_{(u,v) \in \mathcal{S}_{r_i, g_j, k}} w_{uv}, \quad \text{and thus} \quad (11)$$

$$\mathbb{E}[C_{ijk}] = \sum_{(u,v) \in \mathcal{S}_{r_i, g_j, k}} \mathbb{E}[w_{uv}],$$

where  $\mathcal{S}_{r_i, g_j, k}$  is the set of edges on path  $k$  between nodes  $r_i$  and  $g_j$ . The distribution  $\mathcal{D}$ , from which the  $C_{ijk}$  are sampled, is defined through (11) and the distribution  $\mathcal{D}_w$ .

*First-come, first-to-serve:* The following example illustrates this principle when travel time on transport edges is uncertain. We consider a simple redundant assignment problem, as illustrated in Fig. 3, with multiple path choices. In this particular example, robot  $\mathcal{R}_1$  has already been assigned to the goal and  $\mathcal{R}_2$  must choose between paths B and C. Path B is correlated with path A (they share two edges). Path C appears to take longer, and the intuitive choice would be path B. However, as Fig. 4 shows, there is a small chance that path C leads to an improved waiting time at the goal. In other words, if robot  $\mathcal{R}_2$  selects path C, there is a small chance that it is the first robot to arrive at the goal, improving upon the performance of robot  $\mathcal{R}_1$  (i.e., the arrival of robot  $\mathcal{R}_2$  at the goal defines the effective waiting time, and it is the *first-to-serve*).

##### B. Application to Transport with Uncertain Robot Positions

We consider that robot  $i$  estimates its origin node to be  $\tilde{r}_i$ . The true origins of all robots can hence be defined as random variables  $\{r_i | i = 1, \dots, N\}$ , sampled from a distribution  $\mathcal{D}_v$ . Similar to the above, we formulate the travel time of robot  $i$  to goal  $j$  on path  $k$  as

$$C_{ijk} = \sum_{(u,v) \in \mathcal{S}_{r_i, g_j, k}} w_{uv}, \quad (12)$$

where  $\mathcal{S}_{r_i, g_j, k}$  is the set of edges on path  $k$  between the robot  $i$ 's possible origin  $r_i$ , and goal  $g_j$ . When  $w_{uv}$  are fixed, the only source of uncertainty comes from noisy position estimates

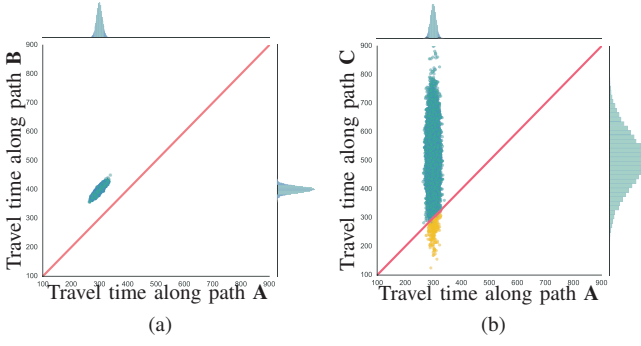


Fig. 4. Joint distributions of travel times, for (a) paths A and B, and (b) paths A and C. The Pearson correlation coefficient is (a) 0.89 and (b) 0.0. The red line shows axis equality (i.e., equal travel times). If robot  $\mathcal{R}_2$  chooses path C, there is a small chance (see datapoints in yellow) that the waiting time at the goal will be improved upon, despite being slower than path B on average.

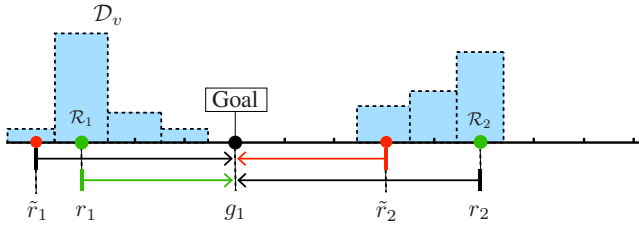


Fig. 5. One-dimensional sketch of a redundant assignment of two robots to one goal. Based on the noisy positions (in red), robot  $\mathcal{R}_2$  is closer to the goal. However, based on the true positions (in green), robot  $\mathcal{R}_1$  is closer to the goal, and is effectively the one to service the task at the goal.

of robots at their origin nodes. In this case, the average travel time for robot  $i$  is

$$\mathbb{E}[C_{ijk}] = \sum_{r_i \in \mathcal{V}} P(r_i | \tilde{r}_i) \sum_{(u,v) \in \mathcal{S}_{r_i, g_j, k}} w_{uv},$$

where  $P(r_i | \tilde{r}_i)$  is the marginal probability that the true origin of robot  $i$  is  $r_i$  when its estimated position is  $\tilde{r}_i$ . We note that when there is no noise on the transport network edges (i.e.,  $w_{uv}$  are fixed), we will assume that each robot takes the shortest path to its assigned goal, and hence,  $K = 1$ . The distribution  $\mathcal{D}$ , from which the  $C_{ijk}$  are sampled, is defined through (12) and the distribution  $\mathcal{D}_v$ .

*First-come, first-to-serve:* Fig. 5 illustrates this principle when robot positions are uncertain. Based on the noisy position estimates, robot  $\mathcal{R}_2$  is closer to the goal, and is assigned to it. A redundant assignment of both robots to the goal reduces the effective waiting time at the goal, with robot  $\mathcal{R}_1$  being the first robot to arrive, since, based on the true positions, robot  $\mathcal{R}_1$  is closer to the goal than robot  $\mathcal{R}_2$ .

## V. ANALYSIS

Our method is underpinned by the following key insight: as we assign additional robots to a given goal, the waiting time at that goal decreases by durations of diminishing length. This property is known as *supermodularity*. In the following, we begin by introducing the fundamental concepts, and demonstrate the supermodular structure of our problem. Building on this, in Section VI, we elaborate a polynomial-time algorithm that addresses Problem 1 and Problem 2.

### A. Preliminaries

A supermodular function is a set function. Given a finite set  $\mathcal{F}$ , it is defined as  $J : 2^{\mathcal{F}} \mapsto \mathbb{R}$ , assigning a scalar to any subset of  $\mathcal{F}$ .

**Definition 3** (Marginal decrease). *For a finite set  $\mathcal{F}$  and a given set function  $J : 2^{\mathcal{F}} \mapsto \mathbb{R}$ , the marginal decrease of  $J$  at a subset  $\mathcal{A} \subseteq \mathcal{F}$  with respect to an element  $x \in \mathcal{F} \setminus \mathcal{A}$  is:*

$$\Delta_J(x|\mathcal{A}) \triangleq J(\mathcal{A}) - J(\mathcal{A} \cup \{x\}). \quad (13)$$

**Definition 4** (Supermodular). *Let  $J : 2^{\mathcal{F}} \mapsto \mathbb{R}$  and  $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{F}$ . The set function  $J$  is supermodular if and only if for any  $x \in \mathcal{F} \setminus \mathcal{B}$ :*

$$\Delta_J(x|\mathcal{A}) \geq \Delta_J(x|\mathcal{B}) \quad (14)$$

The definition implies that adding an element  $x$  to a set  $\mathcal{A}$  results in a larger or equal marginal decrease than when  $x$  is added to a superset of  $\mathcal{A}$ . This property is known as a property of diminishing returns from an added element  $x$  as the set it is added to grows larger.

**Remark 1.** A function  $J$  is submodular if  $-J$  is supermodular.

**Definition 5** (Monotone non-increasing set function). *A set function  $J : 2^{\mathcal{F}} \mapsto \mathbb{R}$  is monotone non-increasing if for any  $\mathcal{A} \subseteq \mathcal{B}$ ,  $J(\mathcal{A}) \geq J(\mathcal{B})$ .*

**Definition 6** (Matroid). *Given a finite ground set  $\mathcal{F}$  and  $\mathcal{I} \subseteq 2^{\mathcal{F}}$  a family of subsets of  $\mathcal{F}$ , an independence system is an ordered pair  $(\mathcal{F}, \mathcal{I})$  with the following two properties: (i)  $\emptyset \in \mathcal{I}$ , and (ii) for every  $\mathcal{A} \in \mathcal{I}$ , with  $\mathcal{B} \subseteq \mathcal{A}$ , implies that  $\mathcal{B} \in \mathcal{I}$ . The second property is known as downwards-closed — in other words, every subset of an independent set is independent. An independence system  $(\mathcal{F}, \mathcal{I})$  is a matroid if it also satisfies the augmentation property, that is, for every  $\mathcal{A}, \mathcal{B} \in \mathcal{I}$ , with  $|\mathcal{A}| > |\mathcal{B}|$ , there exists an element  $a \in \mathcal{A} \setminus \mathcal{B}$  such that  $\{a\} \cup \mathcal{B} \in \mathcal{I}$ .*

For an optimization problem  $\min J(\mathcal{A})$  such that  $\mathcal{A} \in \mathcal{I}$ , where  $J$  is supermodular and  $\mathcal{I}$  is an independence system, we can apply a *greedy* approximation algorithm. This approach works as follows. At each iteration  $k$ , an element  $a$  is added to the solution set  $\mathcal{A}_{k-1}$  such that it maximizes the marginal decrease given this current set,

$$a \leftarrow \operatorname{argmax}_{x \in \mathcal{F}_E(\mathcal{A}_{k-1}, \mathcal{I})} \Delta_J(x|\mathcal{A}_{k-1}) \quad (15)$$

where  $\mathcal{F}_E$  denotes the set of eligible elements, given the independence system  $\mathcal{I}$  and ground set  $\mathcal{F}$ , defined as

$$\mathcal{F}_E(\mathcal{A}_k, \mathcal{I}) \triangleq \{x \in \mathcal{F} \setminus \mathcal{A}_k \mid \mathcal{A}_k \cup \{x\} \in \mathcal{I}\}. \quad (16)$$

A key property is that optimization with this algorithm yields a  $1/2$  approximation ratio [13]. In our case, this is equivalent to *Greedy* returning a set  $\mathcal{A}_G$  with ratio  $J(\mathcal{A}_G) \leq \frac{1}{2} (J^* + J_0)$  where  $J^*$  is the optimal cost, and  $J_0$  is the system's baseline performance without redundant assignments (see also Proposition 2). For matroid rank  $r$  and ground set size  $n$ , *Greedy* requires  $O(nr)$  calls to the objective function.

### B. Supermodularity of cost functional

The following derivations prove the supermodularity of our cost function.

**Lemma 1.** (From [14], Section 3.1 (c)) *Any nonnegative finite weighted sum of supermodular functions is supermodular.*

**Lemma 2.** *Given two sets  $\mathcal{A}$  and  $\mathcal{B}$  that are composed of random variables, we have  $\mathbb{E}[\min \mathcal{A}] \geq \mathbb{E}[\min(\mathcal{A} \cup \mathcal{B})]$ .*

*Proof:* Consider  $X$  and  $Y$  two random variables. Since  $\min\{X\} \geq \min\{X, Y\}$ , we have that  $\mathbb{E}[\min\{X\}] \geq \mathbb{E}[\min\{X, Y\}]$ , which follows directly from the monotonicity of the expectation operator. We can now see that  $\mathbb{E}[\min \mathcal{A}] = \mathbb{E}[\min\{\min \mathcal{A}\}] \geq \mathbb{E}[\min\{\min \mathcal{A}, \min \mathcal{B}\}] = \mathbb{E}[\min(\mathcal{A} \cup \mathcal{B})]$ , since both  $\min \mathcal{A}$  and  $\min \mathcal{B}$  are random variables. ■

Based on these definitions, we show that our cost function  $J_{\mathcal{O}}(\mathcal{A})$  in (7) and (6) has the following property:

**Theorem 1.** *The cost function  $J_{\mathcal{O}}(\mathcal{A})$  is a non-increasing supermodular function of the set  $\mathcal{F}_{\mathcal{O}}$ .*

*Proof:* Based on Lemma 1, it suffices to show that  $\mathbb{E}[\Lambda(I_j(\mathcal{A} \cup \mathcal{O}))] = \mathbb{E}[\min\{C_{ijk} | (i, j, k) \in \mathcal{A} \cup \mathcal{O}\}]$  is supermodular for all  $j$ . This is equivalent to showing that  $\mathbb{E}[\min \mathcal{A}]$  is a supermodular monotone non-increasing function of  $\mathcal{A}$ , where  $\mathcal{A}$  denotes a set of random variables.

By Def. 4, this is true if and only if

$$\begin{aligned} & \mathbb{E}[\min \mathcal{A}] - \mathbb{E}[\min(\mathcal{A} \cup \{x\})] \\ & \geq \mathbb{E}[\min \mathcal{B}] - \mathbb{E}[\min(\mathcal{B} \cup \{x\})] \end{aligned} \quad (17)$$

with  $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{F}_{\mathcal{O}}$  and  $x \in \mathcal{F}_{\mathcal{O}} \setminus \mathcal{B}$ , where  $\mathcal{F}_{\mathcal{O}}$  denotes the ground set. Consider  $\mathcal{B} = \mathcal{A} \cup \mathcal{Y}$  where  $\mathcal{Y} \subset \mathcal{F}_{\mathcal{O}}$ . Given the linearity of the expectation operator, we rewrite (17) as

$$\begin{aligned} & \mathbb{E}[\min \mathcal{A} - \min(\mathcal{A} \cup \{x\})] \\ & \geq \mathbb{E}[\min(\mathcal{A} \cup \mathcal{Y}) - \min(\mathcal{A} \cup \mathcal{Y} \cup \{x\})] \end{aligned} \quad (18)$$

By Lemma 2 we know that  $\min \mathcal{A} \geq \min(\mathcal{A} \cup \mathcal{Y} \cup \{x\})$ . Hence, it remains to show that the difference on the left side of the inequality (18) is greater or equal to the difference on its right side. To do this, we consider the following two possible cases.

*Case 1:*  $\min(\mathcal{A} \cup \{x\}) \geq \min(\mathcal{A} \cup \mathcal{Y})$ . This implies that  $\min(\mathcal{A} \cup \mathcal{Y}) - \min(\mathcal{A} \cup \mathcal{Y} \cup \{x\}) = 0$ . By Lemma 2, we have that  $\min(\mathcal{A} \cup \{x\}) \leq \min \mathcal{A}$ . Thus,  $\min \mathcal{A} - \min(\mathcal{A} \cup \{x\}) \geq 0$ , and given the monotonicity of the expectation operator, the inequality in (18) holds.

*Case 2:*  $\min(\mathcal{A} \cup \{x\}) < \min(\mathcal{A} \cup \mathcal{Y})$ . This implies that  $\min(\mathcal{A} \cup \mathcal{Y} \cup \{x\}) = \min(\mathcal{A} \cup \{x\})$ . By Lemma 2, we have that  $\min(\mathcal{A} \cup \mathcal{Y}) \leq \min \mathcal{A}$ . Thus, we have that  $\min \mathcal{A} - \min(\mathcal{A} \cup \{x\}) \geq \min(\mathcal{A} \cup \mathcal{Y}) - \min(\mathcal{A} \cup \{x\})$ , and given the monotonicity of the expectation operator, the inequality in (18) holds.

To show that  $J_{\mathcal{O}}$  is a monotone non-increasing set function (as defined by Def. 5), it suffices to substitute  $\mathbb{E}[\min \mathcal{A}]$  into the inequality of Lemma 2. ■

The results above establish our problem of selecting redundant robot assignments to minimize the effective waiting time at destinations as a problem of supermodular minimization.

## VI. ALGORITHMIC APPROACH

If we are given a supermodular objective function that satisfies a matroid constraint, we can employ a greedy algorithm to solve our problem within known optimality bounds. However, to maintain the efficiency of such a greedy assignment algorithm, we need to ensure that the evaluation of the objective function itself is efficient (and can be computed in polynomial time). Towards this end, we develop a dynamic programming (DP) approach that hinges on a definition of *incrementally computable functions*, which we apply to our redundant assignment problem. The following paragraphs elaborate our methodology — first, in Sec. VI-A, we show that the matroid constraint applies to our problem setting, and second, in Sec. VI-B, we show how Greedy is implemented efficiently through a dynamic programming approach. The resulting routine is shown in Algorithm 1.

### A. Matroid Constraint

In the following we show that the problem of assigning redundant robots with multiple path options satisfies the properties of a matroid. Following constraints (8) and (9), our problem considers the matroid  $(\mathcal{F}_{\mathcal{O}}, \mathcal{I}_{\mathcal{O}})$ , with

$$\begin{aligned} \mathcal{I}_{\mathcal{O}} \triangleq & \{ \mathcal{A} | \mathcal{A} \subseteq \mathcal{F}_{\mathcal{O}} \wedge |\mathcal{A}| \leq N_d - M \\ & \wedge \forall i | \{j | (i, j, k) \in \mathcal{A} \cup \mathcal{O}\} | \leq 1 \}. \end{aligned} \quad (19)$$

By the definition of a matroid, any valid assignment must be an element of the family of independent sets  $\mathcal{I}_{\mathcal{O}}$ . Firstly, the empty set is a valid solution, in which case our objective function is reduced to  $J_0$ , as given by (3). Secondly, our system is downwards-closed: for any valid robot-to-goal assignment  $\mathcal{A} \in \mathcal{I}_{\mathcal{O}}$ , any subset of assignments  $\mathcal{B} \subseteq \mathcal{A}$  is also a valid assignment by (8). Thirdly, we can show that our system satisfies the augmentation property. For any two valid assignments  $\mathcal{A}$  and  $\mathcal{B}$ ,  $|\mathcal{A}| > |\mathcal{B}|$  implies that there is at least one robot assigned to a goal in set  $\mathcal{A}$  that is unassigned in set  $\mathcal{B}$ , irrespective of what path was selected. Hence, adding that robot-to-goal assignment to set  $\mathcal{B}$  still satisfies (8) and maintains the validity of the solution. We note that the augmentation property implies that all maximal solution sets have the same cardinality  $N_d - M$ , which corresponds to the *rank* of our matroid.

### B. Greedy Assignment with Dynamic Programming

Our work considers uncertainty models, represented by arbitrary distributions that are also capable of capturing correlations between random variables. Our approach is to consider a sampling-based method that takes  $S$  samples from the  $MNK$ -dimensional joint distribution<sup>2</sup>  $\mathcal{D}$ . Our aim is to ensure that the computation of the aggregate cost  $\Lambda$  that assembles the performance of the robot coalition (see Def. 1) does not incur additional complexity that depends on the number of robots  $N$ , the deployment size  $N_d$ , or number of tasks  $M$ . Our insight is that, in a number of practical cases,  $\Lambda$  is a *distributive aggregate function* and is incrementally computable [28]. This

<sup>2</sup>We note that if an analytical model is known, this can be used instead.



---

**Algorithm 1** Greedy Redundant Assignment with DP
 

---

**Input:** Graph  $\mathcal{B} = (\mathcal{U}, \mathcal{F}, \mathcal{C})$ , initial assignment  $\mathcal{O}$   
 Problem 1: size of deployment  $N_d = N$ , cost budget  $\xi$   
 Problem 2: size of deployment  $N_d$ , cost budget  $\xi = 0$   
**Output:** Set of edges  $\mathcal{A}$  defining redundant assignments

```

1:  $\mathcal{A}_G \leftarrow \emptyset$  // set of assignment solutions
2:  $\mathcal{F}_O \leftarrow \mathcal{F} \setminus \mathcal{O}$  // set of eligible assignments
3:  $\mathcal{I}_O \leftarrow \text{Eq. (19)}$  // set of independent assignments
4:  $\hat{\mathcal{C}} \leftarrow \text{sample } S \text{ samples from } MNK\text{-dim. distrib. } \mathcal{D}$ 
5: for  $\hat{C}_{ijk} \in \hat{\mathcal{C}}$  do
6:    $\text{samples}[(i, j, k)] \leftarrow \hat{C}_{ijk}$ 
7: end for
8: for  $(i, j, k) \in \mathcal{O}$  do
9:    $\text{state}[j] \leftarrow \text{samples}[(i, j, k)]$ 
10: end for
11: for  $d \in \{1, \dots, N_d - M\}$  do
12:   if  $J_O(\mathcal{A}_G) \leq \xi$  then
13:     break
14:   end if
15:    $\Delta_{J_O}^* \leftarrow -\infty$ 
16:    $\mathcal{F}_{O,E} \leftarrow \{(i, j, k) \in \mathcal{F}_O \setminus \mathcal{A}_G \mid \mathcal{A}_G \cup \{(i, j, k)\} \in \mathcal{I}_O\}$ 
17:   for  $(i, j, k) \in \mathcal{F}_{O,E}$  do
18:      $\text{curr} \leftarrow \frac{1}{S} \sum_{z=1}^S \text{state}[j]_z$ 
19:      $\text{new} \leftarrow \frac{1}{S} \sum_{z=1}^S \Lambda(\text{state}[j]_z, \text{samples}[(i, j, k)]_z)$ 
20:      $\Delta_{J_O} \leftarrow \text{curr} - \text{new}$ 
21:     if  $\Delta_{J_O} > \Delta_{J_O}^*$  then
22:        $\Delta_{J_O}^* \leftarrow \Delta_{J_O}$ 
23:        $(i^*, j^*, k^*) \leftarrow (i, j, k)$ 
24:     end if
25:   end for
26:    $\mathcal{A}_G \leftarrow \mathcal{A}_G \cup (i^*, j^*, k^*)$ 
27:    $\text{state}[j] \leftarrow \Lambda(\text{state}[j], \text{samples}[(i^*, j^*, k^*)])$ 
28: end for
29: return  $\mathcal{A}_G$ 

```

---

allows us to implement a dynamic programming approach, as shown in Algorithm 1.

**Definition 7** (Distributive Aggregate Function). We define a class  $\Delta$  of distributive aggregate functions  $\delta : A \mapsto \mathbb{R}$ , for  $A \subset \mathbb{R}$ , such that  $\delta \in \Delta$  only if  $\delta(A \cup x)$  can be computed incrementally, as a function of the old value  $\delta(A)$  and new value  $x$  only.

Algorithm 1 solves both Problems 1 and 2. After initializing its data-structures, it proceeds with a greedy assignment of available robots, while respecting constraints (on budget  $\xi$  and available robots  $N_d$ ). This is done by evaluating eligible assignments, and iteratively choosing the one that contributes most to a cost reduction.

The details of the algorithm are as follows. First, the input designates two options: for Problem 2, we set a maximum deployment size and do not consider a cost budget; for Problem 1, the maximum deployment size is limited only by the total number of robots  $N$ , and the cost budget  $\xi$  is set to some feasible value (e.g., within a 1/2 approximation ratio of the optimal performance). Lines 1-10 initialize the data structures. In particular, we pre-sample a fixed set of  $S$  samples (which amounts to sampling  $S$  values for each of the  $MNK$  matchings). Pre-sampling allows the algorithm to maintain the supermodularity property (since, without a fixed sample set, we cannot guarantee diminishing returns). For the remaining

number of robots to be deployed, we proceed with a greedy assignment. We note that lines 12- 14 relate to Problem 1, and can be omitted if only Problem 2 is to be solved. Similarly, for Problem 1,  $N_d$  is set to  $N$ . Line 16 constructs the set of eligible assignments, as in Eq. (16). Then, for all eligible assignment candidates, we compute the marginal cost decrease incurred by adding that assignment to goal  $j$ . In order to do this, Line 19 computes the new aggregate cost function. This is done incrementally, since  $\Lambda$  is a distributive aggregate function. Overall, this inner for-loop is equivalent to Eq. (15), which allows line 22 to retain the best assignment candidate. We then add the best candidate to the current solution, and update the aggregate cost incurred at the goal the new robot was assigned to. Line 27 uses an element-wise operator.

Our approach requires  $O((N_d - M)NMKS)$  calls to the objective function, and the objective function is computed in  $O(S)$ . We summarize the properties of our algorithm in the following proposition.

**Proposition 1.** Algorithm 1 is a valid instantiation of Greedy, and has complexity  $O((N_d - M)NMKS)$ , if (i)  $\Lambda$  is a distributive aggregate function, (ii)  $J_O$  is supermodular, and (iii)  $(F_O, \mathcal{I}_O)$  is a matroid constraint.

The transport network application described in Sec. IV satisfies the conditions in Proposition 1: Since (10) is supermodular, and the *minimum* operator in Def. 2 is a distributive aggregate function. It follows that the objective of minimizing the average effective waiting time is supermodular. The matroid constraint is trivially satisfied.

**Proposition 2.** Consider  $J_0$ , given by (3), as the maximum possible cost of the system, when no redundant robots are deployed, and each goal is assigned exactly one robot. Let  $J_O^*$  be the optimal value of (7). Then, Algorithm 1 returns a set  $\mathcal{A}_G$  satisfying

$$J_O(\mathcal{A}_G) \leq \frac{1}{2} (J_O^* + J_0). \quad (20)$$

The randomized continuous greedy algorithm (proposed in [4, 12]) satisfies

$$J_O(\mathcal{A}_G) \leq \left(1 - \frac{1}{e} - \epsilon\right) J_O^* + \left(\frac{1}{e} + \epsilon\right) J_0. \quad (21)$$

*Proof:* Consider the function  $Q(\mathcal{A}) = J_0 - J_O(\mathcal{A})$ . Since  $J_O(\mathcal{A})$  is non-increasing and supermodular by Theorem 1, then, by Remark 1,  $Q(\mathcal{A})$  is a monotone non-decreasing normalized submodular function. Hence, minimizing  $J_O(\mathcal{A})$  is equivalent to maximizing  $Q(\mathcal{A})$ .

By Theorem 1.1 of [13], for a monotone nondecreasing submodular function  $Q(\mathcal{A})$  subject to a matroid constraint, the greedy algorithm returns a set  $\mathcal{A}_G$  satisfying  $Q(\mathcal{A}_G) \geq Q^*/2$ , with  $Q^* \triangleq \max\{Q(\mathcal{A}) \mid \mathcal{A} \in \mathcal{I}\}$  and where  $M = (\mathcal{F}, \mathcal{I})$  is a matroid given by a membership oracle. Consequently, the solution  $\mathcal{A}_G$  returned by Algorithm 1 satisfies  $Q(\mathcal{A}_G) \geq Q^*/2$ . We substitute the definition of  $Q(\mathcal{A})$  into this equation to yield the result in (20).

By Theorem 1.3 of [4], for a monotone nondecreasing submodular function  $Q(\mathcal{A})$  subject to a matroid constraint, the randomized continuous greedy algorithm returns a set  $\mathcal{A}_G$



satisfying  $Q(\mathcal{A}_G) \geq (1 - 1/e + \epsilon) Q^*$ . By the same procedure as above, substituting the definition of  $Q(\mathcal{A})$  into this equation yields the result in (21). ■

For completeness, we also pose a bound for the set size. Under properties (ii) and (iii) of Proposition 1, the following is known to be true [11]:

**Proposition 3** (following [11, 35]). *Consider  $\mathcal{A}^*$  as the optimal solution to Problem 1, and consider  $\mathcal{A}_0, \mathcal{A}_1, \dots$  the sequence of sets selected by Algorithm 1. Let  $k$  be the smallest index such that  $J_{\mathcal{O}}(\mathcal{A}_k) \leq \xi$ . Then,*

$$\frac{|\mathcal{A}_k|}{|\mathcal{A}^*|} \leq 1 + \log \frac{J_{\mathcal{O}}(\mathcal{F}_{\mathcal{O}}) - J_{\mathcal{O}}(\emptyset)}{J_{\mathcal{O}}(\mathcal{F}_{\mathcal{O}}) - J_{\mathcal{O}}(\mathcal{A}_{k-1})}. \quad (22)$$

## VII. EVALUATION

We evaluate our algorithm through a series of simulations. Section VII-A introduces the benchmark algorithms against which we compare our method. The following section, Sec. VII-B, compares our method to an exhaustive, optimal search strategy. The final two sections, Sec. VII-C and VII-D, discuss the performance of our method for node position uncertainty and edge cost uncertainty, respectively.

As shown in Algorithm 1, the solutions to Problem 1 and 2 are coupled. In particular, it is possible to find an  $N_d$  such that the solution to Problem 2 is the same as to Problem 1, and vice-versa. Hence, the following results focus mainly on Problem 2; we include results for Problem 1 at the end of this section.

### A. Benchmark Algorithms

The performance of our method (*Greedy*) is compared to four alternate assignment algorithms: (1) *Hungarian*: We implement the Hungarian method on expected waiting times for a non-redundant assignment of  $N_d = M$  robots (i.e.,  $\mathcal{A} = \emptyset$ ). This represents the initial assignment  $\mathcal{O}$ , and is used as the baseline for all following (redundant) assignment algorithms. (2) *Random*: A random algorithm assigns the redundant  $N_d - M$  robots randomly to goals. (3) *Repeated Hungarian*: We implement repeated iterations of the Hungarian assignment algorithm (at each iteration, assigning  $M$  redundant robots in one go), until the cap  $N_d - M$  is reached. (4) *Best a-posteriori*: This corresponds to the best a-posteriori performance for a given set of robot origins and goal destinations, based on true (observed) travel times, on which we run the Hungarian method with  $N_d = M$  robots. (5) *Optimal*: We implement an exhaustive search strategy using dynamic programming, which makes  $O(M2^N)$  calls to the objective function (assuming an initial assignment  $\mathcal{O}$  has already been made)<sup>3</sup>.

For each of the simulation series below, we define an underlying noise distribution  $\mathcal{D}$  that describes the uncertainty around our travel time estimates. Using this distribution  $\mathcal{D}$ , we also sample ‘true’ (observed) values, which we use for our performance evaluation. This value is unknown to all algorithms except *Best a-posteriori*.

<sup>3</sup>Evaluations show that implementing *Optimal* with a choice of  $N$  robots (instead of  $N - M$ ) gives imperceptible performance gains.

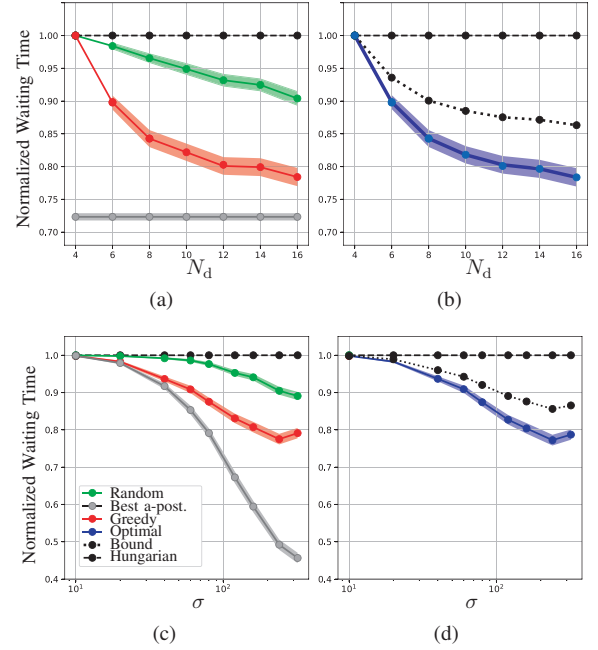


Fig. 6. Performance of the assignment algorithm as measured by normalized waiting time, averaged over 500 iterations. The shaded area shows a 95% confidence interval. The algorithms were evaluated on a  $16 \times 16$  grid with a 50 m separation. The robot origins are perturbed through added Gaussian noise with  $\sigma = 100$ . We have  $N_d = \{4, 6, \dots, 16\}$ ,  $M = 4$ ,  $N = 16$ . Goal locations are sampled randomly within the gridmap for each iteration. Waiting times are normalized by the *Hungarian*, and thus show the improvement w.r.t. the initial non-redundant assignment. For clarity, we separate the results into two subfigures: figures (a) and (c) show results for *Greedy*, *Random*, *Best a-posteriori*, and figures (b) and (d) show results for *Optimal* and its derived bound.

### B. Comparison to Optimal

We perform simulations that compare our algorithm to *Optimal* for  $N_d = \{4, 6, \dots, 16\}$ ,  $M = 4$ ,  $N = 16$ . For this comparison, we consider uncertainty on robot origin nodes. For all algorithms except *Best a-posteriori*, we add noise to the origins of the robots. The noise is sampled from a 2D Gaussian, with uncorrelated noise with a standard deviation  $\sigma = 100$ . We add this noise to a given robot origin  $r_i$ ; the noisy position  $\tilde{r}_i$  is the node that is closest. Robot origin and goal locations are randomly positioned on a  $16 \times 16$  regular grid with 50 m separation. Travels speeds are drawn from a normal distribution with mean 10 m/s and standard deviation 2 m/s.

Fig. 6(a) shows the normalized waiting time  $J/J_0$ , as a function of the maximum deployment size  $N_d$ . The results show that *Greedy* performs near-optimally and well below the bound, with the mean values coinciding with *Optimal* for all values of  $N_d$ . Redundant assignment clearly improves upon non-redundant assignment. Further, for the same deployment size, our algorithm performs significantly better than randomly assigned redundant robots. Fig. 6(b) compares redundant with non-redundant assignment for varying Gaussian noise values. The results show how the improvement of *Greedy* over *Hungarian* increases as the noise increases. For very large noise values (w.r.t. the size of the workspace), the performance of *Random* approaches that of *Greedy*. In summary, these results confirm the benefit of redundant assignments, and demonstrate

the near-optimality of the proposed approach.

### C. Independent Noise with Node Uncertainty

In this section, we analyze the effect of node position uncertainty (at robot origins), on random transport networks. First, we assume that the noise is independent across the nodes – this assumption will be relaxed in the following section. Fig. 7 reports a series of simulations in which we test our algorithm on transport graphs with travel time uncertainty due to uncertain robot positioning. We evaluate Algorithm 1 on a set of 500 random undirected connected transport networks (of which Fig. 1(a) shows an example) with 200 nodes, drawn uniformly at random in a bounded planar area. Our default values are  $N = 25$  robots,  $N_d = 20$  robots,  $M = 5$  goals,  $S = 200$  samples,  $K = 1$  path options. The noisy node positions are generated by selecting one node (uniformly at random) out of the set of nearest neighboring nodes based on Euclidean distance, for a given neighborhood size. The default neighborhood size is 4, which means that one out of 5 nodes is randomly sampled. Robots are initially located at 10 randomly selected hubs.

Fig. 7(a) shows the normalized waiting time as a function of the deployment size. The results show that *Greedy* performs significantly better and *Random* and *Repeated Hungarian*. As the maximum deployment size tends toward the total number of robots, the performance tends toward the ideal setting (*Best a-post.*), and the difference in performance between the algorithm variants decreases.

Fig. 7(b) shows the normalized waiting time as a function of the positioning noise. *Greedy* performs better than the variant methods, with the performance difference decreasing as the noise increases. This trend is analogous with the trend in Fig. 6(b).

### D. Correlated Noise with Edge Uncertainty

In this section, we analyze the effect of uncertain edge costs. In other words, the travel time along the edges of the transport network is uncertain. In particular, we assume that this noise can be correlated across the edges. This assumption reflects real-world settings where travel time uncertainty is affected by causes that are correlated across the transport network (e.g., congestion).

As in Sec. VII-C, we evaluate Algorithm 1 on a set of random undirected connected graphs with 200 nodes. Our default values are  $N = 25$  robots,  $N_d = 20$  robots,  $M = 5$  goals,  $S = 200$  samples,  $K = 4$  path options. We generate the  $K$  path options by taking the shortest (in average)  $K$  paths from  $i$  to  $j$ . Robots are initially located at 10 randomly selected hubs. The joint distribution of travel times along all  $MNK$  possible paths is modeled as a multi-variate Gaussian (truncated at 0) with a mean sampled uniformly at random between 10 and 20. Its covariance matrix is such that the diagonal entries are sampled uniformly between 25 and 100, and the off-diagonal correlation factors are generated using a random lower-triangular matrix corresponding to its Cholesky decomposition. This allows us to sample from the underlying distribution  $\mathcal{D}$ .

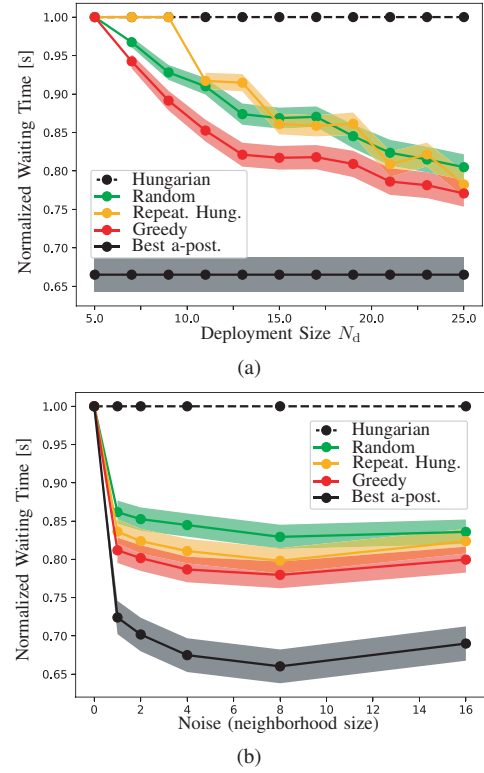


Fig. 7. Performance of the assignment strategies, as measured by normalized waiting time. Each data-point is averaged over 500 runs (random graphs), and the shaded areas show 95% confidence intervals. (a) Performance as a function of deployment size  $N_d$  for  $N = 25$ . (b) Performance as a function of the noise on node positions (implemented through a neighborhood method).

Fig. 8 shows the performance of our algorithm, as measured by the normalized waiting time  $J/J_0$ . Fig. 8(a) shows how, as we increase the total robot deployment  $N_d$ , the waiting time decreases, approaching the lower bound (*Best a-posteriori*). Fig. 8(b) shows how, as we increase the number of path options  $K$  to be considered by the assignment algorithm, performance improves initially, but then flattens out. This validates our usage of a fixed cap ( $K$ ) on the number of paths to be considered by the algorithm. We see that any redundant assignment strategy improves upon non-redundant assignment. Our solution *Greedy* improves significantly upon the benchmarks *Random* and *Repeated Hungarian*.

Fig. 9 shows the correlation of paths (within robot coalitions) in the solutions found by three strategies (*Greedy*, as well as *Repeated Hungarian* and *Random*). For each robot coalition assigned to one goal, we compute the average pairwise correlation between all pairs of paths found for the robots belonging to that coalition. The latter value is averaged over all coalitions. We observe that the correlation of paths within coalitions generated by *Greedy* is lower than that of both *Random* and *Repeated Hungarian*, across all uncertainty distribution correlation values. This indicates that paths selected by *Greedy* tend to be more diverse.

Figure 10 shows results for Problem 1, where we find a solution to the number of robots needed,  $|\mathcal{A}_G|$ , as a function of the percent improvement over a non-redundant assignment (Hungarian method). *Greedy* achieves a much higher improve-

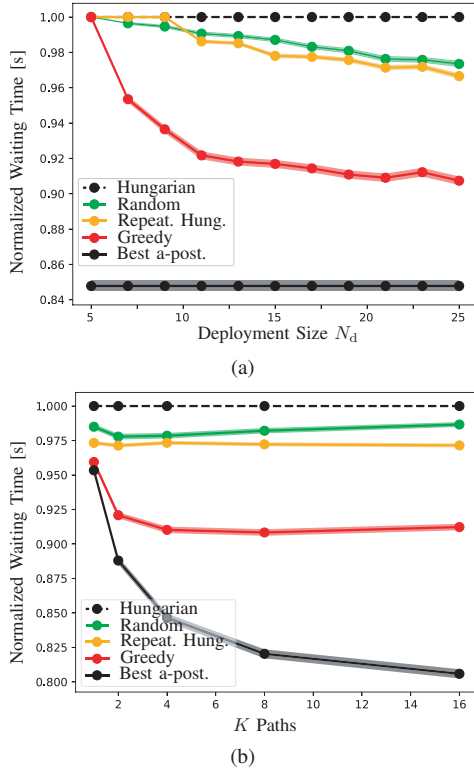


Fig. 8. Performance of the assignment strategies, as measured by normalized waiting time. Each data-point is averaged over 500 runs, and the shaded areas show 95% confidence intervals. (a) Performance as a function of deployment size  $N_d$  for  $N = 25$ . (b) Performance as a function of  $K$  paths,  $N_d = 20$ .

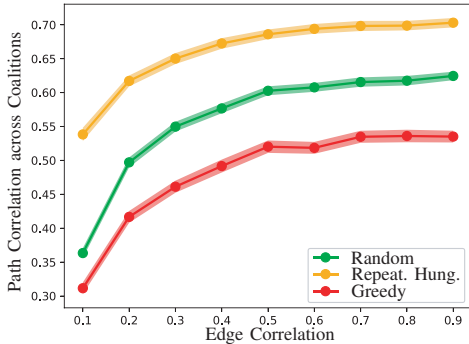


Fig. 9. Correlation across the paths selected by the robots in redundant coalitions. Each data-point is averaged over 500 runs, over all robot coalitions formed. The shaded areas show a 95% confidence interval.

ment than *Random* and *Repeated Hungarian*, for the same number of robots deployed.

## VIII. DISCUSSION

Although the idea of engineering systems with *redundant resources* to increase reliability, robustness and resilience is not new [17, 22], our ideas in this work provide a different take on the concept of redundancy. In particular, we provide a mathematical framework that allows us to reason about the added value of redundancy for mobile systems where the cost of time comes at the highest premium. Although we do provide an option to limit the number of additional robots to deploy, future work should more explicitly address

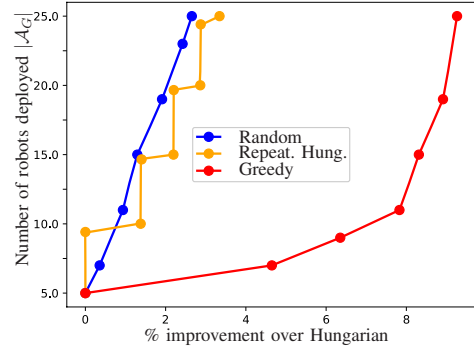


Fig. 10. The number of robots deployed,  $|A_G|$ , as a function of the percent improvement over a non-redundant assignment (Hungarian method). Given a target improvement over the Hungarian method, each data-point represents the average performance across 500 runs on random transport networks.

the trade-off between the cost of providing redundancy (e.g., operational costs, maintenance costs) and performance gains. Within this context, future studies should analyze the economy of on-demand task assignment systems, analyzing how much more users are willing to pay for improved quality of service and reduced waiting times. Richer variants of the problem statement would consider more comprehensive budget constraints as well as heterogeneous robots with different costs. A potential limitation of our approach is that we do not explicitly model the redistribution of robots due to the redundant assignment scheme. Whether this robot re-balancing is beneficial remains to be studied. We note that our current work only focuses on static assignment for a given batch of tasks. For task assignment problems with a continuous influx of tasks, our framework would need to be extended. In previous work, we implemented such a mechanism through a sliding window approach that keeps a reserve of robots to accommodate future (unknown) task demands [31]. This idea should be further extended to optimize the re-balancing of the robot distribution, as a function of predictions of spatio-temporal demand distributions.

It would also be interesting, in future work, to study whether there are approaches that solve our problem while further relaxing the assumptions, as introduced in Sec. III. The challenge would consist of developing approaches that remain tractable. We acknowledge that *not* assuming an initial assignment, and assigning all redundant resources from scratch, with the same method, could potentially lead to a solution that is closer to the optimum. The issue in doing so, however, is that we lose the computational gains of our approach. In specific, we would need to fall back to a formulation of the classical set-partitioning problem, which is strongly NP-hard. Similarly, we note that finding an optimal selection of  $K$  paths is hard, and not attempting to do so does indeed simplify the problem significantly. In our approach to the problem, we take the position that there is a finite set of reasonable paths (e.g., a taxi driver in a city will only consider a few options at any given point in time). We acknowledge that alternative approaches or heuristics for selecting  $K$  paths could be implemented. For example, more risk-averse or optimistic variations are possible. We leave such analyses for future work.



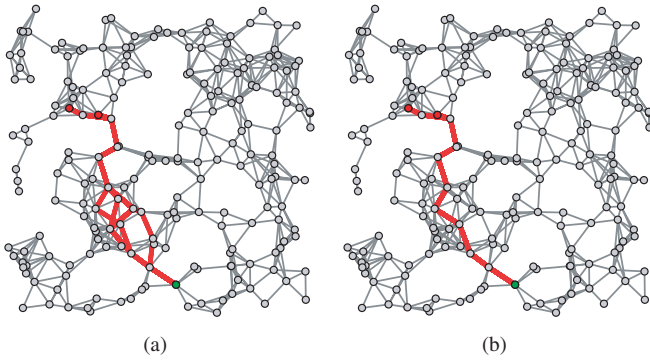


Fig. 11. Paths selected by a robot coalition initially located at two separate hubs (red nodes), and assigned to a goal (green node) for (a) *Greedy* and (b) *Repeated Hungarian*.

The results in Sec. VII-D indicate an interesting connection between resilience and diversity: the paths selected by *Greedy* in the redundant robot coalitions tend to be more diverse (and correlate with better performance). This insight is illustrated in Fig. 11, which shows two instances of paths selected by a redundant coalition of 5 robots, using *Greedy* in Fig. 11(a) and *Repeated Hungarian* in Fig. 11(b). The *Repeated Hungarian* method sends robots along the same perceived best path, whereas *Greedy* evaluates the added gain for each new redundant robot, and hence, diversifies the selected paths. Although these results provide insights to the algorithm’s inner workings, more work needs to be done in order to explicitly exploit the coupling between robustness and diversity.

Finally, we highlight that, although we focus on transport networks and travel in particular, the underlying concepts of this work can be analogously applied to a broad class of multi-agent or multi-robot problems, where there are dependent costs (e.g., risks) or dependent benefits (e.g., information gain) among agents.

## IX. CONCLUSION

In this work, we provided a framework for robot-to-goal assignment that is robust to uncertainty over robot travel times. The main novelty is the exploitation of robot redundancy, and the formulation of a supermodular optimization framework that efficiently and near-optimally selects redundant robot matchings to minimize the average waiting time at the goal locations. Our first-come-first-to-serve principle implies a *minimum* aggregation over redundant assignments. This allows us to compute our objective function efficiently by dynamic programming, leading to a polynomial-time algorithm that can be run in real-time, even for large numbers of robots, goals, and graph nodes. Our results show that redundant assignment reduces waiting time with respect to non-redundant assignments. This performance gap between redundant and non-redundant assignment increases with increasing noise levels. The proposed redundant assignment algorithm is valid for the general problem of uncertain travel times. In particular, we do not make any assumptions on the underlying uncertainty models, i.e., distributions can be arbitrary and also correlated across edges and nodes. Finally, our findings include results on the benefit of diversity and complementarity in redundant robot

coalitions; these are unprecedented insights within this context, and contribute towards providing robustness and resilience to uncertainty through targeted robot team compositions.

## ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers for their constructive feedback. The author would also like to thank Matthew Malencia for his careful review of this paper.

This work was supported by ARL DCIST CRA W911NF-17-2-0181, by the Centre for Digital Built Britain, under InnovateUK grant number RG96233, for the research project “Co-Evolving Built Environments and Mobile Autonomy for Future Transport and Mobility”, and by the Engineering and Physical Sciences Research Council (grant EP/S015493/1).

## REFERENCES

- [1] R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin. Exact and heuristic algorithms for the weapon-target assignment problem. *Operations Research*, 55(6):1136–1146, 2007.
- [2] A. Angrisano, S. Gaglione, and C. Gioia. Performance assessment of GPS/GLONASS single point positioning in an urban environment. *Acta Geodaetica et Geophysica*, 48(2):149–161, 2013.
- [3] A. Atamtürk, G. L. Nemhauser, and M. W. Savelsbergh. A combined lagrangian, linear programming, and implication heuristic for large-scale set partitioning problems. *Journal of heuristics*, 1(2):247–259, 1996.
- [4] A. Badanidiyuru and J. Vondrák. Fast algorithms for maximizing submodular functions. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1497–1514. Society for Industrial and Applied Mathematics, 2014.
- [5] S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1): 135–153, 2001.
- [6] T. Chakraborti, A. Kulkarni, S. Sreedharan, D. E. Smith, and S. Kambhampati. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 86–96, 2019.
- [7] N. Cho and J. Linderoth. Row-partition branching for set partitioning problems. In *Proceedings of the INFORMS Computing Society Meeting*, pages 119–133, 2015.
- [8] A. Clark, L. Bushnell, and R. Poovendran. A supermodular optimization framework for leader selection under link noise in linear multi-agent systems. *IEEE Transactions on Automatic Control*, 59(2):283–296, 2014.
- [9] M. Corah and N. Michael. Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice. *Autonomous Robots*, 43(2):485–501, 2019.
- [10] J. Enright and P. R. Wurman. Optimization and coordinated autonomy in mobile fulfillment systems. In *Automated action planning for autonomous mobile robots*, pages 33–38, 2011.
- [11] U. Feige. A Threshold of  $\ln N$  for Approximating Set Cover. *J. ACM*, 45(4):634–652, July 1998. ISSN 0004-5411. doi: 10.1145/285055.285059. URL <http://doi.acm.org/10.1145/285055.285059>.
- [12] Y. Filmus and J. Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM Journal on Computing*, 43(2):514–542, 2014.
- [13] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions-II. *Polyhedral combinatorics*, pages 73–87, 1978.
- [14] S. Fujishige. *Submodular Functions and Optimization*, volume 58. Elsevier, 2005.

- [15] M. R. Garey and D. S. Johnson. “Strong” NP-completeness results: Motivation, examples, and implications. *Journal of the ACM (JACM)*, 25(3):499–508, 1978.
- [16] B. P. Gerkey and M. J. Mataric. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.
- [17] P. Ghare and R. Taylor. Optimal redundancy for reliability in series systems. *Operations research*, 17(5):838–847, 1969.
- [18] D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.
- [19] P. Grippa, D. A. Behrens, C. Bettstetter, and F. Wall. Job selection in a network of autonomous uavs for delivery of goods. *Robotics: Science and Systems*, 2017.
- [20] S. Jorgensen, R. H. Chen, M. B. Milam, and M. Pavone. The matroid team surviving orienteers problem: Constrained routing of heterogeneous teams with risky traversal. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5622–5629, 2017. doi: 10.1109/IROS.2017.8206450.
- [21] P. A. Krokhmal and P. M. Pardalos. Random assignment problems. *European Journal of Operational Research*, 194(1): 1–17, 2009.
- [22] S. Kulturel-Konak, A. E. Smith, and D. W. Coit. Efficiently solving the redundancy allocation problem using tabu search. *IIE transactions*, 35(6):515–526, 2003.
- [23] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 18–28. ACM, 2001.
- [24] C. Nam and D. A. Shell. When to do your own thing: Analysis of cost uncertainties in multi-robot task allocation at run-time. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1249–1254, 2015.
- [25] C. Nam and D. A. Shell. Analyzing the sensitivity of the optimal assignment in probabilistic multi-robot task allocation. *IEEE Robotics and Automation Letters*, 2(1):193–200, 2017.
- [26] M. Ouimet and J. Cortés. Hedonic coalition formation for optimal deployment. *Automatica*, 49(11):3234–3245, 2013.
- [27] J. G. Oxley. *Matroid theory*, volume 3. Oxford University Press, USA, 2006.
- [28] T. Palpanas, R. Sidle, R. Cochrane, and H. Pirahesh. Incremental maintenance for non-distributive aggregate functions. In *Proceedings of the 28th International Conference on Very Large Data Bases*, pages 802–813, 2002.
- [29] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus. Robotic load balancing for mobility-on-demand systems. *The International Journal of Robotics Research*, 31(7):839–854, 2012.
- [30] A. Prorok. Redundant robot assignment on graphs with uncertain edge costs. In *Distributed Autonomous Robotic Systems*, pages 313–327. Springer, 2019.
- [31] A. Prorok and V. Kumar. Privacy-preserving vehicle assignment for mobility-on-demand systems. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1869–1876, Sept 2017. doi: 10.1109/IROS.2017.8206003.
- [32] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, 34:707–755, 2009.
- [33] P. Singh, M. Chen, L. Carlone, S. Karaman, E. Frazzoli, and D. Hsu. Supermodular mean squared error minimization for sensor scheduling in optimal kalman filtering. In *American Control Conference (ACC)*, 2017, pages 5787–5794. IEEE, 2017.
- [34] K. Spieser, S. Samaranayake, W. Gruel, and E. Frazzoli. Shared-vehicle mobility-on-demand systems: a fleet operator’s guide to rebalancing empty vehicles. In *Transportation Research Board*, 2016.
- [35] V. Tzoumas, A. Jadbabaie, and G. J. Pappas. Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms. In *2016 American Control Conference (ACC)*, pages 191–196, July 2016. doi: 10.1109/ACC.2016.7524914.
- [36] S. Umetani. Exploiting variable associations to configure efficient local search in large-scale set partitioning problems. In *AAAI*, pages 1226–1232, 2015.
- [37] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.
- [38] R. K. Williams, A. Gasparri, and G. Ulivi. Decentralized matroid optimization for topology constraints in multi-robot allocation problems. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 293–300. IEEE, 2017.