# Redundant Robot Assignment on Graphs with Uncertain Edge Costs

Amanda Prorok

Department of Computer Science & Technology
University of Cambridge, UK
asp45@cam.ac.uk

**Abstract.** We provide a framework for the assignment of multiple robots to goal locations, when robot travel times are uncertain. Our premise is that *time* is the most valuable asset in the system. Hence, we make use of *redundant* robots to counter the effect of uncertainty and minimize the average waiting time at destinations. We apply our framework to transport networks represented as graphs, and consider uncertainty in the edge costs (i.e., travel time). Since solving the redundant assignment problem is strongly NP-hard, we exploit structural properties of our problem to propose a polynomial-time solution with provable sub-optimality bounds. Our method uses *distributive aggregate functions*, which allow us to efficiently (i.e., incrementally) compute the effective cost of assigning redundant robots. Experimental results on random graphs show that the deployment of redundant robots through our method reduces waiting times at goal locations, when edge traversals are uncertain.

## 1 Introduction

Technological advances are enabling the large-scale deployment of robots to solve various types of logistics problems, such as product delivery [9], warehousing [3], and mobility-on-demand [18]. Robot teams also hold the promise of delivering robust performance in unstructured or extreme environments [22, 11]. The commonality of many of these applications is that they require the assignment of robots to destinations. In general, a solution to the assignment problem can be computed by a centralized unit that collects all robot-to-goal 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, i.e., the cost of traversing any path in the underlying representation. Unpredictable events along robot paths cause uncertain travel times. For example, in structured environments, we may encounter traffic accidents and congestion; in unstructured environments, we may encounter unsafe terrains and risky traversal conditions.

We are interested in applications that require a fast arrival of robots at their destinations, but where knowledge about expected robot travel times may be imprecise or incomplete due to uncertain path conditions. Although robot

assignments under random costs have gained a considerable amount of attention [12, 16, 15], the focus has primarily been on providing analyses of the performance under noisy conditions. In this work, we propose a complementary method by making use of *robot redundancy*. In other words, the core idea of our work is to exploit redundancy to counter the adverse affect of uncertainty. Although the idea of engineering robust systems with redundant resources is not new in a broad sense [13, 8], 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.

**Background.** Our problem belongs to the class of Single-Task Robots, Multi-Robot Tasks (ST-MR) [7], which considers the assignment of groups of robots that have a combined utility for any given task. 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 is strongly NP-hard [6]. 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 [5]. As we show in this work, and in a previous report [19], our approach provides diminishing returns in the number of redundant robots assigned to each goal. This yields a *supermodular* cost function, justifying the usage of efficient, greedy assignment algorithms.

**Contributions.** In this paper, we solve the problem of selecting optimal redundant robot-to-goal matchings on transport graphs with uncertain edge costs. Our objective is to minimize the waiting time at the goals, whilst respecting an upper limit on how many (redundant) robots may be assigned. A previous report proved the supermodularity of our objective [19], and considered the special case where robot positions are uncertain, yet path costs are fixed. The present paper focuses on uncertain edge costs, which implies variances on path costs, and introduces the additional complexity of selecting among numerous partially correlated uncertain path choices. We show that we can capture these additional elements with a matroid constraint, and thus, can provide a polynomial-time algorithm that uses distributive aggregate functions to efficiently assemble the gain of assigning redundant robots. Finally, we provide evaluations on random graphs with uncertain, correlated edge costs.

## 2    Problem Description

We consider a system composed of $M$ goals and $N$ available robots. The size of the total robot deployment is constrained by $N_{\mathrm{d}}$, with $M \leq N_{\mathrm{d}} \leq N$. This constraint is relevant for applications that run continuous assignments with fluc-
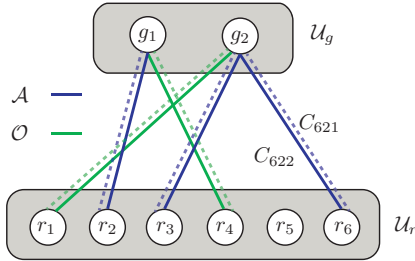
**Fig. 1.** Sketch of redundant robot-to-goal matching, for a total of $N = 6$ available robots, $M = 2$ goals, $K = 2$ path options, and a deployment size of $N_{\mathrm{d}} = 5$. Edges in $\mathcal{O}$ represent the initial assignment, and edges in $\mathcal{A}$ represent the redundant assignment. The random variable $C_{ijk}$ represents the uncertain travel time it takes robot $i$ to reach goal $j$ via path $k$.

tuating task demand, and enables the reservation of some robots for future deployments. Furthermore, it allows operators to limit the cost of redundant robot deployments, for example by monitoring and capping energy consumption. 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. Without loss of generality, we assume that $K$ possible paths exist between each robot and each goal. We seek to find a minimum cost matching, such that all goals are covered, and any goal may be assigned multiple robots, while respecting the limit on the maximum deployment size $N_{\mathrm{d}}$.

**Assignment with Random Path 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 $\mathcal{U} = \mathcal{U}_r \cup \mathcal{U}_g$, $\mathcal{U}_r \cap \mathcal{U}_g = \emptyset$. We define a fixed number $K$ of possible path routes that lead any robot to any goal. The edge set $\mathcal{F} = \{(i, j, k) | i \in \mathcal{U}_r, j \in \mathcal{U}_g, k \in 1, \dots, K, \forall i, j, k\}$ is complete, meaning that any robot can reach any goal node, and that up to $K$ possible path choices exist for any pair $(i, j)$. 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; in particular, edge costs may be correlated, and we do not make use of *i.i.d.* assumptions.

We are interested in settings where an initial non-redundant assignment has been made. We consider an initial assignment $\mathcal{O} \subset \mathcal{F}$, 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 $\mathcal{A}^\star \subset \mathcal{F} \setminus \mathcal{O}$ for the remaining $N_{\mathrm{d}} - M$ robots [1]. From here on, we denote $\mathcal{F} \setminus \mathcal{O}$ by $\mathcal{F}_{\mathcal{O}}$. Fig. 1 illustrates a simple robot-to-goal matching.

---

[1]Without $\mathcal{O}$, any solution that is smaller in size than $M$ would lead to an infinite waiting time, and hence, the objective function looses its supermodular property. The assumption that we already have an initial assignment is necessary, for the developments that follow.

**Optimization Problem.** 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., $N_\mathrm{d} - M > 0$), 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 $\mathcal{A}$, 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*

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

This definition allows 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 edge costs:

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

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

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

We formalize our problem as follows.

*Problem 1. Optimal matching of redundant robots under cardinality constraints and with multiple path choices:* Given $N$ available robots, $M$ goals with uncertain robot-to-goal assignment costs, $K$ possible paths from each robot to each goal, 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 $N_\mathrm{d}$. This is formally stated as:

$$\operatorname*{argmin}_{\mathcal{A} \subseteq \mathcal{F}_{\mathcal{O}}} \; J_{\mathcal{O}}(\mathcal{A}) \tag{4}$$

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

$$|\mathcal{A}| = N_\mathrm{d} - M \tag{6}$$

Fig. 2 shows an example of an assignment problem 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
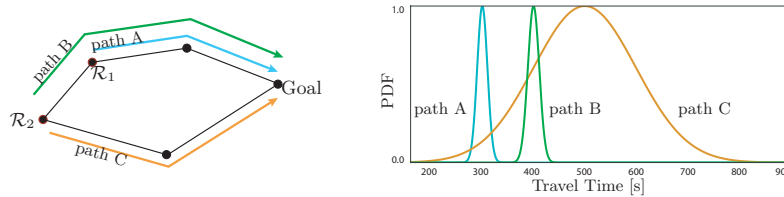
**Fig. 2.** Example scenario. Robot $\mathcal{R}_1$ is assigned to the goal via path A. Robot $\mathcal{R}_2$ is a redundant robot, and must choose between paths B and C.
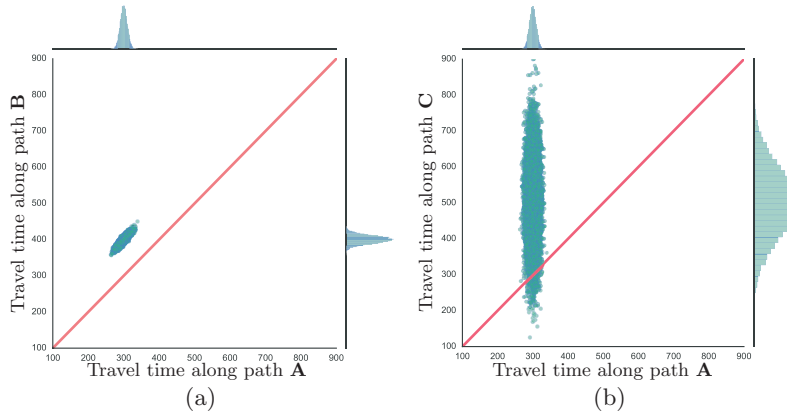


**Fig. 3.** Joint distributions of travel times, for (a) paths A and B, and (b) paths A and C. The Pearson correlation coefficient is (a) 0.89 and (b) 0.0. The red line shows axis equality (i.e., equal travel times). If robot $\mathcal{R}_2$ chooses path C, there is a small chance that the waiting time at the goal will be improved upon, despite being slower than path B on average.

would be path B. However, as Fig. 3 shows, there is a small chance that path C leads to an improved waiting time at the goal, which cannot occur would robot $\mathcal{R}_2$ choose path B. Hence, the goal of our method is to identify combinations of robot assignments and paths that lead to the highest possible performance improvement. In the experimental section (Sec. 6) we show how this leads to a notion of robot complementarity and diversity.

## 3   Preliminaries

Our method is underpinned by the following key insight: as we assign additional robots to a given goal, the assignment cost for that goal decreases by a diminishing amount. This property is known as *supermodularity*. In addition, the cardinality constraint imposed by our maximum deployment size $N_d$ can be represented by a *matroid* that also respects multiple path options. These two properties aid us in finding an efficient algorithm to solve the redundant assign-

ment problem, as elaborated later in Sec. 4. The following definitions introduce the underlying concepts.

**Definition 2 (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:*

$$\Delta_J(x|\mathcal{A}) \triangleq J(\mathcal{A}) - J(\mathcal{A} \cup \{x\}). \tag{7}$$

**Definition 3 (Supermodular function).** *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 $x \in \mathcal{F} \setminus \mathcal{B}$:*

$$\Delta_J(x|\mathcal{A}) \geq \Delta_J(x|\mathcal{B}) \tag{8}$$

The definition implies that adding an element $x$ to a set $\mathcal{A}$ results in a larger 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 4 (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 $\mathcal{A} \subset \mathcal{I}$, 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,

$$a \leftarrow \mathrm{argmax}_{x \in \mathcal{F}_E(\mathcal{A}_{k-1}, \mathcal{I})} \Delta_J(x|\mathcal{A}_{k-1}) \tag{9}$$

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

$$\mathcal{F}_E(\mathcal{A}_k, \mathcal{I}) \triangleq \{x \in \mathcal{F} \setminus \mathcal{A}_k \,|\, \mathcal{A}_k \cup \{x\} \in \mathcal{I}\}. \tag{10}$$

A key property is that optimization with this algorithm yields a $1/2$ approximation ratio [4]. In our case, this is equivalent to *Greedy* returning a set $\mathcal{A}^{\star}$ with ratio $J(\mathcal{A}^{\star}) \leq \frac{1}{2}(J^{\star} + J_0)$ where $J^{\star}$ is the optimal cost, and $J_0$ is the system's baseline performance (without redundant assignments) [19]. For matroid rank $r$ and ground set size $n$, *Greedy* requires $O(nr)$ calls to the objective function.

# 4 Algorithmic Approach

The section above establishes that 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 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. 4.1, we show that the matroid constraint applies to our problem setting, and second, in Sec. 4.2, we show how Greedy is implemented efficiently through a dynamic programming approach. The resulting routine is shown in Algorithm 1.

## 4.1 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 (5) and (6), our problem considers the matroid $(\mathcal{F}_{\mathcal{O}}, \mathcal{I}_{\mathcal{O}})$, with

$$\mathcal{I}_{\mathcal{O}} \triangleq \{\mathcal{A} | \mathcal{A} \subseteq \mathcal{F}_{\mathcal{O}} \wedge |\mathcal{A}| \leq N_{\mathrm{d}} - M \wedge \forall i | \{j | (i, j, k) \in \mathcal{A} \cup \mathcal{O}\} | \leq 1\}. \quad (11)$$

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 (5). 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 (5) and maintains the validity of the solution. We note that the augmentation property implies that all maximal solution sets have the same cardinality $N_{\mathrm{d}} - M$, which corresponds to the *rank* of our matroid.

## 4.2 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 $\mathcal{D}$ [2]. Our aim is to ensure that the computation of the aggregate cost $\Lambda$ that assembles the performances of the redundant robots (see Def. 1) does not incur additional complexity that depends

---

[2]We note that if an analytical model is known, this can be used instead.

---

**Algorithm 1** Greedy Redundant Assignment with Dynamic Programming

---

**Input:** Graph $\mathcal{B} = (\mathcal{U}, \mathcal{F}, \mathcal{C})$, size of deployment $N_\mathrm{d}$, initial assignment $\mathcal{O}$
**Output:** Set of edges $\mathcal{A}$ defining redundant assignments
1: $\mathcal{A} \leftarrow \emptyset$
2: $\mathcal{F}_\mathcal{O} \leftarrow \mathcal{F} \setminus \mathcal{O}$
3: $\mathcal{I}_\mathcal{O} \leftarrow$ Eq. (11)
4: $\hat{\mathcal{C}} \leftarrow$ sample $S$ samples from $MNK-$dimensional distribution $\mathcal{D}$
5: **for** $\hat{C}_{ijk} \in \hat{\mathcal{C}}$ **do**
6: $\quad$ samples$[(i,j,k)] \leftarrow \hat{C}_{ijk}$
7: **end for**
8: **for** $(i,j,k) \in \mathcal{O}$ **do**
9: $\quad$ task_state$[j] \leftarrow$ samples$[(i,j,k)]$
10: **end for**
11: **for** $d \in \{1, \ldots, N_\mathrm{d} - M\}$ **do**
12: $\quad \Delta^\star_{J_\mathcal{O}} \leftarrow -\infty$
13: $\quad \mathcal{F}_{\mathcal{O},E} \leftarrow \{(i,j,k) \in \mathcal{F}_\mathcal{O} \setminus \mathcal{A} \,|\, \mathcal{A} \cup \{(i,j,k)\} \in \mathcal{I}_\mathcal{O}\}$
14: $\quad$ **for** $(i,j,k) \in \mathcal{F}_{\mathcal{O},E}$ **do**
15: $\quad\quad$ curr_state $\leftarrow \frac{1}{S} \sum_{z=1}^{S}$ task_state$[j]_z$
16: $\quad\quad$ new_state $\leftarrow \frac{1}{S} \sum_{z=1}^{S} \Lambda($task_state$[j]_z,$ samples$[(i,j,k)]_z)$
17: $\quad\quad \Delta_{J_\mathcal{O}} \leftarrow$ curr_state $-$ new_state
18: $\quad\quad$ **if** $\Delta_{J_\mathcal{O}} > \Delta^\star_{J_\mathcal{O}}$ **then**
19: $\quad\quad\quad \Delta^\star_{J_\mathcal{O}} \leftarrow \Delta_{J_\mathcal{O}}$
20: $\quad\quad\quad (i^\star, j^\star, k^\star) \leftarrow (i,j,k)$
21: $\quad\quad$ **end if**
22: $\quad$ **end for**
23: $\quad \mathcal{A} \leftarrow \mathcal{A} \cup (i^\star, j^\star, k^\star)$
24: $\quad$ task_state$[j] \leftarrow \Lambda.($task_state$[j],$ samples$[(i^\star,j^\star,k^\star)])$ // element-wise $\Lambda$
25: **end for**
26: **return** $\mathcal{A}$

---

on the number of robots $N$, the deployment size $N_\mathrm{d}$, 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 [17]. This allows us to implement a dynamic programming approach, as shown in Algorithm 1.

**Definition 5 (Distributive Aggregate Function).** *We define a class $\Delta$ of <u>distributive</u> 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.*

Sec. 5 gives an implementation of distributive aggregate functions for the specific application considered in this work.

**Proposition 1.** *Algorithm 1 is a valid instantiation of Greedy, and has complexity $O((N_\mathrm{d} - M)NMKS)$, if* (i) *$\Lambda$ is a distributive aggregate function,* (ii) *$J_\mathcal{O}$ is supermodular, and* (iii) *$(F_\mathcal{O}, \mathcal{I}_\mathcal{O})$ is a matroid constraint.*

Algorithm 1 works as follows. Lines 1-10 initialize the data structures. In particular, lines 4-7 pre-sample a fixed set of $S$ samples (which amounts to sam-

pling $S$ values for each of the $MNK$ edges). Pre-sampling allows the algorithm to maintain the supermodularity property. For the remaining number of robots to be deployed, we proceed with a greedy assignment. Line 13 constructs the set of eligible assignments, as in Eq. (10). 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 16 computes the new aggregate cost function. This is done incrementally, since $\Lambda$ is a distributive aggregate function. Overall, Lines 14-22 are equivalent to Eq. (9), with Lines 18-21 retaining the best assignment candidate. Line 23 adds the best candidate to the current solution, and Line 24 updates the aggregate cost incurred at the goal the new robot was assigned to. Our approach requires $O((N_\mathrm{d} - M)NMK)$ calls to the objective function, and the objective function is computed in $O(S)$.

## 5   Application to Transport Networks

We are interested in applications that use graphs to represent possible robot routes (from origins to goals), where the cost of traversing any individual route (or edge) is *uncertain*. 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. The $k$th path between an origin node $i$ and goal node $j$ is denoted by tuple $(i,j,k)$. 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 $k$. We define a random variable $w_{uv} \in \mathcal{W}$ that represents the time needed to traverse an edge $(u,v)$. We formulate the estimated travel time on a path as:

$$C_{ijk} = \sum_{(u,v)\in\mathcal{S}_{ijk}} w_{uv}, \ \ \text{and thus} \ \ \mathbb{E}[C_{ijk}] = \sum_{(u,v)\in\mathcal{S}_{ijk}} \mathbb{E}[w_{uv}], \qquad (12)$$

where $\mathcal{S}_{ijk}$ is the set of edges on path $k$ between node $i$ and node $j$.

Let us consider a *'first-come, first-to-serve'* principle, 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 further specified in the following definition.

**Definition 6 (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*

$$\Lambda(I_j(\mathcal{A})) \triangleq \min\{C_{ijk}|(i,j,k) \in I_j(\mathcal{A})\}. \qquad (13)$$

This application satisfies the conditions in Proposition 1: In [19] we show that Eq. 13 is supermodular. Further, the *minimum* operator is a distributive aggre-
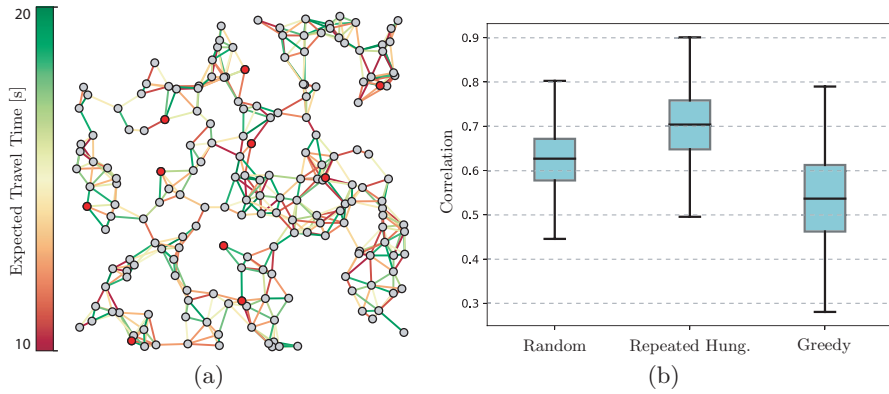
**Fig. 4.** (a) Instance of random graph. Red dots indicate robot hubs. The edges are colored to indicate the expected travel time. (b) Comparison between *Greedy*, *Random*, and *Repeated Hungarian*, with respect to the correlation of paths in the solution set.

gate function. It follows that the objective of minimizing the average effective waiting time is supermodular. The matroid constraint is trivially satisfied.

## 6   Evaluation

We evaluate Algorithm 1 in simulation, on a set of random undirected connected graphs with 200 nodes (of which Fig. 4(a) shows an example). Our default values are $N = 25$ robots, $N_{\mathrm{d}} = 20$ 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 $MNK$ edges 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 $\mathcal{D}$. Using this distribution $\mathcal{D}$, we also sample a 'true' (observed) travel time for each edge, which we use for our performance evaluation (this value is unknown to all algorithms except *Best a-posteriori*, as described below). The performance of our method (*Greedy*) is compared to four alternate assignment algorithms: **(1)** *Hungarian:* We implement the Hungarian method on average waiting times for a non-redundant assignment of $N_{\mathrm{d}} = M$ 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 $N_{\mathrm{d}} - M$ 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 $N_{\mathrm{d}} - M$
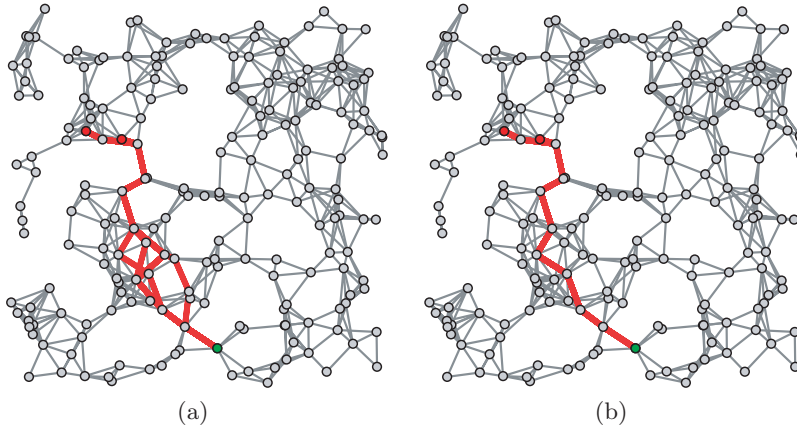
**Fig. 5.** Paths selected by a robot coalition initially located at two separate hubs (red nodes), and assigned to a goal (green node) for (a) *Greedy* and (b) *Repeated Hungarian*.

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.

Fig. 4(b) shows the correlation of paths 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 coalition paths generated by *Greedy* is lower than that of both *Random* and *Repeated Hungarian*. This indicates that paths selected by *Greedy* tend to be more diverse. This is exemplified in Fig. 5, which shows paths selected by a coalition of 5 robots using *Greedy* in Fig. 5(a) and *Repeated Hungarian* in Fig. 5(b).

Fig. 6 shows the performance of our algorithm, as measured by the normalized waiting time $J/J_0$. Fig. 6(a) shows how, as we increase the total robot deployment $N_d$, the waiting time decreases, approaching the lower bound (given best a-posteriori). Fig. 6(b) shows how, as we increase the number of paths $K$ 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. Overall, we see that any redundant assignment strategy improves upon non-redundant assignment. Our solution *Greedy* improves significantly upon the benchmarks *Random* and *Repeated Hungarian*.
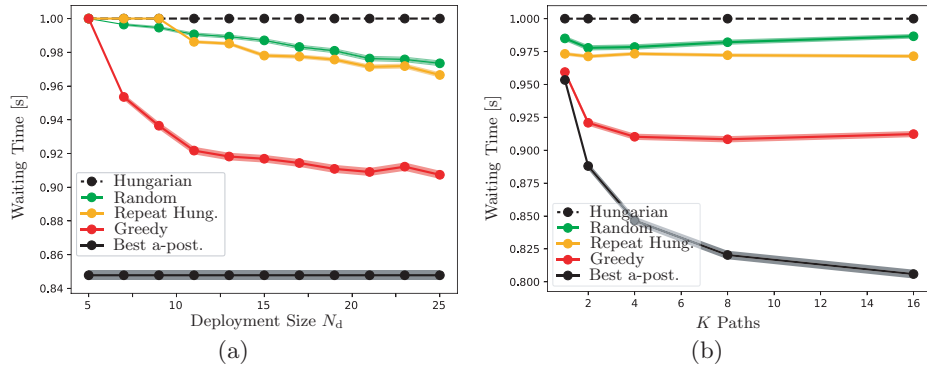
**Fig. 6.** Performance of the assignment strategies, as measured by normalized waiting time. Each data-point is averaged over 500 runs, and the shaded areas show 95% confidence intervals. (a) Performance as a function of deployment size $N_d$ for $N = 25$. (b) Performance as a function of $K$ paths, $N_d = 20$.

## 7   Related Work

Our problem is related to the general class of submodular welfare problems [14, 23], 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. 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 contrast to our work, however, the weapon-target assignment problem only applies to binomial distributions that model the outcome (i.e, survival) of each target as a Bernoulli random variable. Our algorithm is capable of dealing with arbitrary (and potentially correlated) probability distributions that describe the assignment costs. In that sense, our problem is a generalization of the weapon-target assignment problem.

Submodular optimization for combinatorial problems has has also gained considerable traction in the domain of multi-robot systems. Applications include coordinated robot routing for environmental monitoring [20], leader selection in leader-follower systems [2], sensor scheduling for localization [21], and path planning for orienteering missions [10]. Typically, the aforementioned studies develop explicit objective functions that are specific to the considered problem domains. The authors then go about proving the submodularity property and devising the appropriate assignment algorithms. This general methodology is similar to the one presented in our paper. The particularity of our work lies in the specificity of our objective function for redundant robot assignments under uncertain travel times, which is not captured by objective functions developed in prior work.

## 8   Conclusion

In this work, we provided a framework for redundant robot-to-goal assignment. Our main contribution is a supermodular optimization framework that efficiently selects robot matchings and the corresponding paths that minimize the average waiting time at the goal locations when travel times along paths are uncertain. We introduced a polynomial-time algorithm that uses distributive aggregate functions to efficiently assemble the gain of assigning redundant robots. Our evaluations on random graphs with uncertain and partially correlated edge costs showed that redundant assignment reduces the waiting time at goals with respect to non-redundant assignments. In particular, our proposed algorithm significantly outperforms alternate benchmark algorithms. We also demonstrate that our method creates robot coalitions that tend to be more complementary, and thus, more diverse in their selected path options. In conclusion, we show how providing redundant robots can be a key design feature for systems where time is the primary asset, and where an over-provisioning with respect to the number of robots is a minor concern.

## Acknowledgements

# Bibliography

[1] Ahuja RK, Kumar A, Jha KC, Orlin JB (2007) Exact and heuristic algorithms for the weapon-target assignment problem. Operations Research 55(6):1136–1146

[2] Clark A, Bushnell L, Poovendran R (2014) A supermodular optimization framework for leader selection under link noise in linear multi-agent systems. IEEE Transactions on Automatic Control 59(2):283–296

[3] Enright J, Wurman PR (2011) Optimization and coordinated autonomy in mobile fulfillment systems. In: Automated action planning for autonomous mobile robots, pp 33–38

[4] Fisher ML, Nemhauser GL, Wolsey LA (1978) An analysis of approximations for maximizing submodular set functions-II. Polyhedral combinatorics pp 73–87

[5] Fujishige S (2005) Submodular Functions and Optimization, vol 58. Elsevier

[6] Garey MR, Johnson DS (1978) "Strong" NP-completeness results: Motivation, examples, and implications. Journal of the ACM (JACM) 25(3):499–508

[7] Gerkey BP, Mataric MJ (2004) A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. Interntional Journal of Robotics Research 23(9):939–954

[8] Ghare P, Taylor R (1969) Optimal redundancy for reliability in series systems. Operations research 17(5):838–847

[9] Grippa P, Behrens DA, Bettstetter C, Wall F (2017) Job selection in a network of autonomous uavs for delivery of goods. Robotics: Science and Systems

[10] Jorgensen S, Chen RH, Milam MB, Pavone M (2017) 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), pp 5622–5629, DOI 10.1109/IROS.2017.8206450

[11] Kantor G, Singh S, Peterson R, Rus D, Das A, Kumar V, Pereira G, Spletzer J (2003) Distributed search and rescue with robot and sensor teams. In: Field and Service Robotics, Springer, pp 529–538

[12] Krokhmal PA, Pardalos PM (2009) Random assignment problems. European Journal of Operational Research 194(1):1–17

[13] Kulturel-Konak S, Smith AE, Coit DW (2003) Efficiently solving the redundancy allocation problem using tabu search. IIE transactions 35(6):515–526

[14] Lehmann B, Lehmann D, Nisan N (2001) Combinatorial auctions with decreasing marginal utilities. In: Proceedings of the 3rd ACM conference on Electronic Commerce, ACM, pp 18–28

[15] Nam C, Shell DA (2015) 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), pp 1249–1254

[16] Nam C, Shell DA (2017) Analyzing the sensitivity of the optimal assignment in probabilistic multi-robot task allocation. IEEE Robotics and Automation Letters 2(1):193–200

[17] Palpanas T, Sidle R, Cochrane R, Pirahesh H (2002) Incremental maintenance for non-distributive aggregate functions. In: Proceedings of the 28th International Conference on Very Large Data Bases, pp 802–813

[18] Pavone M, Smith SL, Frazzoli E, Rus D (2012) Robotic load balancing for mobility-on-demand systems. The International Journal of Robotics Research 31(7):839–854

[19] Prorok A (2018) Supermodular optimization for redundant robot assignment under travel-time uncertainty. arXiv preprint arXiv:180404986

[20] Singh A, Krause A, Guestrin C, Kaiser WJ (2009) Efficient informative sensing using multiple robots. Journal of Artificial Intelligence Research 34:707–755

[21] Singh P, Chen M, Carlone L, Karaman S, Frazzoli E, Hsu D (2017) Supermodular mean squared error minimization for sensor scheduling in optimal kalman filtering. In: American Control Conference (ACC), 2017, IEEE, pp 5787–5794

[22] Thayer SM, Dias MB, Nabbe B, Digney BL, Hebert M, Stentz A (2001) Distributed robotic mapping of extreme environments. In: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII, International Society for Optics and Photonics, vol 4195, pp 84–96

[23] Vondrák J (2008) Optimal approximation for the submodular welfare problem in the value oracle model. In: Proceedings of the fortieth annual ACM symposium on Theory of computing, ACM, pp 67–74