Fair Robust Assignment Using Redundancy

Matthew Malencia, Vijay Kumar, George Pappas, and Amanda Prorok

***Abstract*—We study the consideration of fairness in redundant assignment for multi-agent task allocation. It has recently been shown that redundant assignment of agents to tasks provides robustness to uncertainty in task performance. However, the question of how to *fairly* assign these redundant resources across tasks remains unaddressed. In this letter, we present a novel problem formulation for fair redundant task allocation, which we cast as the optimization of worst-case task costs under a cardinality constraint. Solving this problem optimally is NP-hard. We exploit properties of supermodularity to propose a polynomial-time, near-optimal solution. In supermodular redundant assignment, the use of additional agents always improves task costs. Therefore, we provide a solution set that is α times larger than the cardinality constraint. This constraint relaxation enables our approach to achieve a super-optimal cost by using a sub-optimal assignment size. We derive the sub-optimality bound on this cardinality relaxation, α. Additionally, we demonstrate that our algorithm performs near-optimally without the cardinality relaxation. We show simulations of redundant assignments of robots to goal nodes on transport networks with uncertain travel times. Empirically, our algorithm outperforms benchmarks, scales to large problems, and provides improvements in both fairness and average utility.**

***Index Terms*—Ethics and philosophy, fairness, multi-robot systems, submodular optimization, task planning.**

I. Introduction

FAIRNESS in algorithms has received increasing attention in both research [3], [4], [31] and policy [9], [27], [32]. Multi-agent task allocation algorithms distribute resources among *human-centric* tasks, thus requiring the consideration of fairness. Bias in these systems has myriad sources, such as design choices [18], models and data [17], and interpretation of results [8]. In this letter, we focus on the fairness of the objective function. We reevaluate the objective function of redundant multi-agent task allocation under the consideration of fairness and propose an algorithm for solving for this fair objective.

Manuscript received October 15, 2020; accepted February 19, 2021. Date of publication March 18, 2021; date of current version April 8, 2021. This letter was recommended for publication by Associate Editor B. Wang and Editor M. Ani Hsieh upon evaluation of the reviewers' comments. This work was supported by ARL Grant DCIST CRA W911NF-17-2-0181, NSF Grant CNS-1521617, ARO Grant W911NF-13-1-0350, ONR Grants N00014-20-1-2822 and N00014-20-S-B001, and Qualcomm Research. The work of Matthew Malencia was supported by the National Science Foundation Graduate Research Fellowship under Grant DGE-1845298. *(Corresponding author: Matthew Malencia.)*

Matthew Malencia is with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104-6228 USA, and also with the Faculty of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, U.K. (e-mail: malencia@seas.upenn.edu).

Vijay Kumar and George Pappas are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, PA 19104-6228 USA (e-mail: kumar@cis.upenn.edu; pappasg@seas.upenn.edu).

Amanda Prorok is with the Faculty of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, U.K. (e-mail: asp45@cl.cam.ac.uk).

Digital Object Identifier 10.1109/LRA.2021.3067283

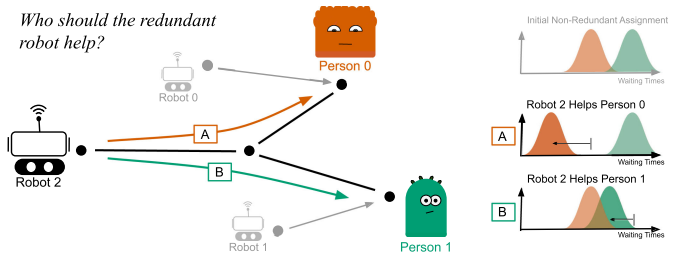


Fig. 1. Redundant assignment answers the question, *which task should receive extra resources?* Given an existing non-redundant assignment and respective cost distributions (*top right*), a single redundant robot, Robot 2, can be assigned to either task. Utilitarian approaches assign Robot 2 to Person 0 because of the higher improvement in cost (*plot A*), whereas fair approaches assign Robot 2 to Person 1 because of their higher need (*plot B*).

Multi-agent task allocation is studied across domains ranging from operations research to robotics [10], [20], [35]. Because finding optimal allocations in this combinatorial problem is challenging, most approaches assume deterministic costs. However, real-world costs are uncertain. For example, in assigning agents to spatially located goals, travel times and agent locations are not deterministic [25]. To account for this uncertainty, previous approaches define risk-based objective functions [37] or use risk constraints [35]. In recent work, Prorok shows that redundant assignment provides robustness against uncertainty and is complementary to existing approaches [24]. For example, in *time-sensitive* applications such as rescue scenarios and robots delivering life saving supplies [1], redundancy improves performance because tasks are completed by the first robot to arrive based on the *first-come first-to-serve* principle [24].

The redundant assignment problem is classified as Single-Task Robots, Multi-Robot Tasks with Instantaneous Assignment (ST-MR-IA) because each robot can complete all tasks (ST) and multiple robots can be assigned to the same task (MR) [12]. This problem can be cast as a set-partitioning problem which has been proven to be strongly NP-hard [11]. ST-MR-IA problems require a global optimization function. Traditional methods that optimize utilitarian welfare [13] (the expected sum of costs) provide no guarantees for individual fairness [26]. Fig. 1 shows that utilitarian approaches seek the best absolute improvement in task cost without accounting for individual task needs. In *human-centric* applications such as rescue scenarios, delivery systems, and ride-sharing, disparities in task performance yield disparities in the treatment of individual people. This unfairness in utilitarian assignments makes the system unfit with respect to broader social structures and causes a lack of trust in the system. For this reason, it is crucial to consider *fair* optimization objectives.

Fairness in redundant assignment is an especially important consideration because the tasks served in these allocation problems are often both human-centric and high-risk. For example,

2377-3766 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:09:50 UTC from IEEE Xplore. Restrictions apply.

in rescue scenarios such as robots delivering life saving supplies, the choice of how robots are allocated impacts how long different individuals must wait for the life saving supplies. This allocation strategy stems directly from the objective function chosen by the system designer.

There exist many definitions of fair objectives [6], [19], [36]. For example, proportional division seeks assignments that provide agents with equal proportional utility but has no guarantees for indivisible resources [29], and envy-freeness seeks assignments such that no agent prefers another agent's assignment over its own [2]. While fair objectives are often *intra*-personal comparisons, where agents evaluate others' assignments according to their own utility function, experimental work shows that human behavior is better predicted by *inter*-personal comparisons, where agents compare outcomes [14]. For redundant assignment, we seek to minimize the maximum cost among the set of tasks. Specifically, we answer the question: *how can we find a redundant assignment that is fair to the worst-off task?* Hence, we define an inter-task criteria based on John Rawls' *veil of ignorance* [26].

The 'veil of ignorance' thought experiment for the example of emergency supply delivery is as follows. Imagine you get to choose the assignment of delivery robots *and* you will be on the receiving end of this assignment, i.e., one of the humans in need of life-saving supplies. However, you are *ignorant* of your location. Rawls concludes that, under this *veil of ignorance*, the rational conclusion is to choose an assignment that benefits the person with the longest waiting time, since this person could be you. This definition of fairness is often formulated as a minimax or maximin problem [36]. We cast this problem as the optimization of worst-case expected task cost.

A. Related Work

Noise and uncertainty in real-world applications require managing risk (known unknowns) [5], [7], [16]. A common risk measure is Conditional Value at Risk (CVaR), which is the expectation of the tail of a distribution. Wilder [33] and Zhou et. al [37] use CVaR in approaches to submodular optimization. In addition to risk measures, redundancy provides robustness to uncertainty. Prorok [24] shows that redundant assignment improves task performance under uncertainty and is complementary to traditional risk measures. We leverage redundancy in this work to maintain the guarantees for satisfying all constraints of the assignment problem while not precluding the incorporation of traditional risk measures.

Most assignment problems, as well as many problems in robotics, optimize the average expected utility. This default objective is mathematically convenient but has been questioned under considerations such as fairness. Zhang and Shah [36] define four maximin fairness criteria for multi-agent Markov Decision Processes. These definitions of fairness are based on the Rawlsian theory of justice [26]. The fairness criteria in this letter similarly applies Rawlsian justice to redundant assignment. Mathematically, this definition is similar to providing robustness to a worst-case objective. Therefore, we look to insights from prior work solving minimax problems.

Minimax problems are difficult to solve because they are highly nonlinear and combinatorial in nature. Uncertainty adds additional complexity to this problem. Because it is NP-hard to solve these problems optimally, approaches exploit mathematical properties of problem formulations to produce polynomial-time, near-optimal solutions. One such property is supermodularity, the property of diminishing returns (see Appendix VII-A). Supermodular objectives (or in many cases, submodular) can be solved using simple greedy algorithms with a $\frac{1}{2}$ bound on optimality [21]. Many objective functions are naturally supermodular, such as information gain [28] and real-world monetary costs [22]. Submodularity has also been exploited to create robustness to worst-case objectives [15] and arises naturally in redundant task assignments as it captures the attribute of diminishing returns [24].

In a closely related work, Krause [15] defines the Robust Submodular Observation Selection problem, where the goal is to select an observation set that is robust against the worst-case of multiple objective functions. This work is grounded in the example of sensor placement for monitoring an environmental process. The problem involves finding a set of sensor locations, $\mathcal{A}$, that is a subset of all possible sensor locations, V. There exists a set of $J_j(\mathcal{A})$ that measure the variance reduction at locations j. The objective is to choose k sensor locations that maximize the minimum variance reduction (equivalent to minimizing the maximum variance). This problem is formulated as $\max_{\mathcal{A}\subseteq V} \ \min_j J_j(\mathcal{A}) \ \text{ s.t. } \ |\mathcal{A}| \leq k$, where each $J_1, \ldots, J_M$ are normalized monotonic submodular functions.

To solve this problem, Krause [15] proposes the SATURATE algorithm, which finds an approximate solution by relaxing the cardinality constraint, with guarantees on this relaxation. Powers et. al [23] extend this work for any matroid constraint. They propose the GENERALIZED SATURATION algorithm to solve this problem by relaxing the objective. They guarantee that only a fraction of the set of objective functions exceed a threshold. In contrast to these works, we solve a problem under *both* assignment constraints and a cardinality constraint whereby our algorithm relaxes the constraint on cardinality.

B. Contributions

We propose a novel problem formulation that (for the first time) formalizes the *fair* redundant task allocation problem. Redundancy serves to improve performance under risk. An assumption of supermodularity enables performance guarantees. Notably, we are the first to develop an algorithm for solving a fair redundant assignment problem. Our main contributions are as follows:

- We define the Fair Supermodular Redundant Assignment Problem as the minimization of the maximum expected cost across tasks under assignment constraints and cardinality constraints.
- We propose an efficient algorithm to solve the Fair Supermodular Redundant Assignment problem approximately by relaxing the cardinality constraint.
- We prove a bound on this relaxation and the computational complexity of our approach.

We analyze our algorithm in simulation, showcasing our theory empirically, demonstrating that our algorithm outperforms benchmarks, and analyzing a case study to compare the fairness of our algorithm to prior work.

II. PRELIMINARIES

The problem space in this letter is represented by a graph $\mathcal{B} = (\mathcal{U}, \mathcal{F}, \mathcal{C})$. The set of vertices $\mathcal{U}$ is partitioned into two disjoint sets $\mathcal{U}_a$ and $\mathcal{U}_t$, the sets of agent nodes and task nodes,

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:09:50 UTC from IEEE Xplore. Restrictions apply.

respectively. The edge set $\mathcal{F} = \{(i,j) \mid i \in \mathcal{U}_a, j \in \mathcal{U}_t,\ \forall\, i,j\}$ defines all possible agent-task pairings available for assignment. For each (i,j) pair in $\mathcal{F}$, there exists a corresponding cost $\mathcal{C}_{i,j}$, which is a random variable. The set of all $\mathcal{C}_{i,j}$ is the cost set, $\mathcal{C}$. While specific instances of this problem can constrain $\mathcal{C}$, we make no assumptions on the form, independence, or correlations of the elements of $\mathcal{C}$.

There are M tasks and N agents, and a total deployment size, N_d, specified as an input. Each task j has a corresponding cost function $J_j(\cdot)$ mapping the cost set $\mathcal{C}$ to a real number.[1] For example, $J_j(\cdot)$ could be the sum, minimum, or product of the costs $\mathcal{C}_{i,j}$ for agents i assigned to task j. These cost functions represent how a set of redundant agents complete a task; the functions map a subset of the assignment to a real valued cost. In this letter we consider the minimum operator, representing the first-come first-to-serve principle [24].

Fair assignment seeks a pairing of agents to tasks that minimizes the maximum task cost, where agents cannot be assigned multiple tasks, tasks must be assigned at least one agent, and the number of agents is no more than N_d. This problem is formulated as finding a subset of the ground set that minimizes the maximum cost: $\min_{\mathcal{A}\subseteq\mathcal{F}} \max_j J_j(\mathcal{A})$. This assignment, $\mathcal{A}$, has a constrained cardinality of $|\mathcal{A}| \leq N_d$, and assignment constraints per agent and task. The optimization is formulated with the introduction a second decision variable ξ and constraint of all $J_j(\mathcal{A})$ by ξ. Finding the minimum ξ such that all $J_j(\mathcal{A}) \leq \xi$ equivalently minimizes $\max_j J_j(\mathcal{A})$.

Definition 1: Fair Assignment: Given N agents and M tasks with uncertain agent-task assignment costs, find an assignment $\mathcal{A} \subseteq \mathcal{F}$ no larger than $N_d \geq M$ and minimum cost bound $\xi \in \mathbb{R}$, such that no costs $J_j(\mathcal{A})$ exceed ξ. Formally:

$$\begin{aligned} \min_{\xi,\mathcal{A}\subseteq\mathcal{F}} \quad & \xi \\ \text{s.t.} \quad & J_j(\mathcal{A}) \leq \xi, \quad \forall j \\ & |\mathcal{A}| \leq N_d \\ & \forall\, i,\ |\{j \mid (i,j) \in \mathcal{A}\}| \ \leq 1 \\ & \forall\, j,\ |\{i \mid (i,j) \in \mathcal{A}\}| \ \geq 1 \end{aligned} \tag{1}$$

Fair assignment is a set-partitioning problem and has been shown to be strongly NP-hard [11]. Approaches that solve similar problems do not have guarantees on all of these constraints [37], [38]. To ensure that both assignment constraints are met and to reduce uncertainty through redundancy, we define the Fair Supermodular Redundant Assignment problem.

III. Problem Statement

Consider a system of M tasks and N agents, where each agent-task pairing has random cost, represented by an arbitrary probability distribution.[2] For each of the M tasks, a cost function is defined. Our goal is to find an assignment that:

- minimizes the maximum task cost,
- uses at most N_d agents,
- assigns each agent to at most one task, and
- ensures no task is unassigned ($N_d \geq M$).

Redundant assignment, where multiple agents can be assigned the same task, provides robustness against uncertainty and improvements to expected performance of tasks [24]. If tasks have deterministic costs, this problem simplifies to the linear bottleneck assignment problem which can be solved using the thresholding algorithm [30]. The presence of uncertainty, however, requires a new approach to this optimization.

This section describes a flexible mathematical formulation of fair assignment under two assumptions:

Assumptions. (1) all cost functions, $J_j(\cdot)$, must be *supermodular*,[3] and (2) a prior non-redundant assignment exists.

The first assumption is that all cost functions are supermodular, i.e., they have the property of diminishing return. However, we are solving for the maximum of a set of supermodular functions, which is generally *not supermodular* (see Appendix VII-A). Therefore, the greedy algorithm has no performance guarantees for this problem. Simple examples, as shown by Krause, prove that the greedy algorithm applied to this problem can perform arbitrarily badly [15].

The second assumption requires an initial non-redundant assignment exists, denoted $\mathcal{O}$, such as one found through standard assignment methods (e.g., threshold algorithm for the linear bottleneck assignment problem). This assumption guarantees that all tasks are assigned at least one agent, thus eliminating this constraint. Additionally, this assumption enables tractability and ensures the supermodularity of cost functions (e.g., unassigned tasks potentially have infinite cost, breaking supermodularity). With this assumption, we focus strictly on assigning the redundant agents. Therefore, the ground set becomes $\mathcal{F}\backslash\mathcal{O}$, denoted $\mathcal{F}_{\mathcal{O}}$. Cost functions remain denoted $J_j(\mathcal{A})$ though they are a function of $\mathcal{A}$ given $\mathcal{O}$.

These two assumptions on Definition 1 yield Problem 1, the Fair Supermodular Redundant Assignment problem. The constraint in Definition 1 that all tasks be completed is guaranteed by the initial assignment (Assumption 2), and therefore omitted. All cost functions, $J_j(\mathcal{A})$, are assumed to be supermodular and are a function of $\mathcal{A}$ given $\mathcal{O}$.

Problem 1: Fair Supermodular Redundant Assignment: Given N agents, M tasks with uncertain agent-task assignment costs, and an initial assignment $\mathcal{O}$, find an assignment $\mathcal{A} \subseteq \mathcal{F}_{\mathcal{O}}$ and minimum cost bound $\xi \in \mathbb{R}$, such that no cost $J_j(\mathcal{A})$ exceeds ξ, assuming all functions $J_j(\mathcal{A})$ are supermodular. Formally:

$$\begin{aligned} \min_{\xi,\mathcal{A}\subseteq\mathcal{F}_{\mathcal{O}}} \quad & \xi \\ \text{s.t.} \quad & J_j(\mathcal{A}) \leq \xi, \quad \forall j \\ & |\mathcal{A}| \leq N_d - M \\ & \forall\, i,\ |\{j \mid (i,j) \in \mathcal{A} \cup \mathcal{O}\}| \ \leq 1 \end{aligned} \tag{2}$$

This formulation is general and flexible enough to be applied to many assignment problems, as long as the cost functions are supermodular. One such example is the assignment of agents with uncertain travel times to spatially located tasks. In this example, the cost at each node is defined as the expected waiting time, $J_j(\mathcal{A}) = \mathop{\mathbb{E}}_{c}[\min_i\{C_{ij} | (i,j) \in \mathcal{A} \cup \mathcal{O}\}]$. The minimum operator here represents the first-come first-to-serve principle introduced by Prorok [24], who showed that this function is supermodular.

[1] We later assume that all $J_j(\cdot)$ are supermodular.

[2] Specific instances of this problem can restrict these distributions, but we make no assumptions on their form, independence, or correlations.

[3] $J_j(\cdot)$ can each be different functions but should be similarly scaled so that comparisons among different cost functions are meaningful.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:09:50 UTC from IEEE Xplore. Restrictions apply.

IV. Approach

The algorithm presented in this section provides an approximate solution, $\mathcal{A}_f$, to the Fair Supermodular Redundant Assignment problem. The solution is approximate because we relax the cardinality constraint for tractability purposes: the solution $\mathcal{A}_f$ can be α times larger than the size of the optimal assignment (this section elaborates on the bound on α). By relaxing the size of the solution, this assignment is guaranteed to have a maximum cost that is at most equal to the maximum cost of the optimal solution $\mathcal{A}^*$. That is, **the approximate solution $\mathcal{A}_f$ has a cost that is less than that of $\mathcal{A}^*$ but a size that is larger**. This section outlines the algorithm, a bound on the cardinality relaxation α, the limits on the deployment size N_d, and the computational complexity of this approach.

A. The Relaxed Fair Supermodular Redundant Assignment Problem

The basic premise of our approach to solving Problem 1 is to conduct half-interval search on possible values of ξ and to solve a sub-problem to determine which half-interval cannot contain the target solution. Suppose there exists an algorithm that finds the smallest assignment that yields a maximum task cost of at most ξ, which is given as an input:

$$\begin{aligned} \underset{\mathcal{A}\subseteq\mathcal{F}_\mathcal{O}}{\text{argmin}} \quad & |\mathcal{A}| \\ \text{s.t.} \quad & J_j(\mathcal{A}) \leq \xi, \quad \forall j \\ & \forall\, i,\ |\{j \mid (i,j) \in \mathcal{A} \cup \mathcal{O}\}| \ \leq 1 \end{aligned} \tag{3}$$

This sub-problem returns a candidate solution $\mathcal{A}_s$. If $\mathcal{A}_s$ has a cardinality that is at most $N_d - M$, then $\mathcal{A}_s$ and the given value of ξ are feasible for Problem 1. If $|\mathcal{A}_s| > N_d - M$, then $\mathcal{A}_s$ and the ξ value are not feasible. Solving this sub-problem determines whether a given value of ξ is feasible. Therefore, half-interval search can find the optimal (minimum feasible) value of ξ. Given a starting range of possible ξ values, the midpoint is tested for feasibility. If feasible, the half-interval below the midpoint contains the optimal value and thus becomes the new range of possible ξ values. Otherwise, the higher half-interval becomes the new range. This procedure repeats until convergence, at which point the returned $\mathcal{A}_s$ and ξ are guaranteed to be optimal.

However, Krause proves with Theorem 3 [15] that unless $P = NP$, there cannot exist any polynomial time approximation algorithm for Problem 1. In other words, unless $P = NP$, the sub-problem in equation (3) cannot be solved in polynomial time. To enable a polynomial time approximation algorithm for Problem 1, the cardinality constraint is relaxed by α. This results in the *relaxed* Fair Supermodular Redundant Assignment problem, with $\alpha > 1$:

$$\begin{aligned} \min_{\xi, \mathcal{A}\subseteq\mathcal{F}_\mathcal{O}} \quad & \xi \\ \text{s.t.} \quad & J_j(\mathcal{A}) \leq \xi, \quad \forall j \\ & |\mathcal{A}| \leq \alpha(N_d - M) \\ & \forall\, i,\ |\{j \mid (i,j) \in \mathcal{A} \cup \mathcal{O}\}| \ \leq 1 \end{aligned} \tag{4}$$

This relaxed problem can now be solved with the aforementioned half-interval procedure. Given a value ξ, the sub-problem of the relaxed Fair Supermodular Redundant Assignment problem is still equation (3). However, the relaxation on the cardinality constraint changes the update step of the half-interval search. We begin with a range of values $[\xi_{\text{min}}, \xi_{\text{max}}]$, where $\xi_{\text{min}} = 0$ because costs are strictly positive and ξ_{max} is the worst waiting time after the initial assignment, $\max_j J_j(\emptyset)$. Given the mid-point of this range, the sub-problem returns a candidate solution $\mathcal{A}_s$. Previously, the cardinality of $\mathcal{A}_s$ was compared to $N_d - M$ to determine its feasibility. Because of the cardinality relaxation, the cardinality of $\mathcal{A}_s$ now informs either the *infeasibility* of the original problem or the *feasibility* of the relaxed problem, adjusting the respective bounds:

- If $|\mathcal{A}_s| > \alpha(N_d - M)$, then ξ is not a feasible solution to the original problem.[4] Set $\xi_{\text{min}} = \xi$.
- If $|\mathcal{A}_s| \leq \alpha(N_d - M)$, then ξ is a feasible solution to the relaxed problem. Set $\xi_{\text{max}} = \xi$.

To guarantee a solution, the number of agents used after the relaxation, $\alpha(N_d - M)$, must not exceed the number of agents available, $N - M$. Thus, set $N_d \leq \frac{N-M}{\alpha} + M$.

Through this point, we have assumed that there exists an approach to solving the sub-problem defined in equation (3) above. We have also assumed that this solution satisfies the relaxed cardinality constraint α. The rest of this section describes both an algorithm that solves this sub-problem and the bound on the cardinality relaxation α.

B. Algorithm Details

We seek an algorithm to sub-problem (3) that finds the smallest set $\mathcal{A}$ such that all cost functions $J_j(\mathcal{A})$ are below a given cost budget ξ. We build on the prior work of Prorok on *optimal matching of redundant agents under a cost budget* [24]. Leveraging this work requires two challenges to be addressed. First, Prorok uses a single supermodular function constraint while we constrain a set of M supermodular functions. Second, the relaxation of this sub-problem must hold for all iterations of the half-interval search.

We transform equation 3 to have a single supermodular cost constraint. Let $\hat{J}_j(\mathcal{A})$ be the truncated function $\max\{J_j(\mathcal{A}), \xi\}$ and form a single constraint of the average of the set of $\hat{J}_j(\mathcal{A})$. This constraint is equivalent to the original set of cost constraints. The average of truncated cost functions less than ξ implies that all cost functions are less than ξ. Additionally, truncation and the average function preserves supermodularity. Because the average of the truncated cost functions has a minimal value of ξ, the inequality constraint becomes an equality constraint.

$$\begin{aligned} \underset{\mathcal{A}\subseteq\mathcal{F}_\mathcal{O}}{\text{argmin}} \quad & |\mathcal{A}| \\ \text{s.t.} \quad & \frac{1}{M}\sum_j \hat{J}_j(\mathcal{A}) = \xi \\ & \forall\, i,\ |\{j \mid (i,j) \in \mathcal{A} \cup \mathcal{O}\}| \ \leq 1 \end{aligned} \tag{5}$$

Prorok's greedy algorithm [24] is shown to solve this problem in polynomial time with a bound (which we call α) on the ratio of the solution set size to the optimal solution set size. However, to use this approximation algorithm in the half-interval search, the bound on α must be independent of ξ. Theorem 1 below details a ξ-independent bound on α.

[4] Or if there does not exist a feasible $\mathcal{A}_s$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:09:50 UTC from IEEE Xplore. Restrictions apply.

We solve Problem 1 with Algorithm 1, Fair Supermodular Redundant Assignment. After initialization, Algorithm 1 conducts a half-interval search over ξ. For each value of ξ, Prorok's Greedy Redundant Assignment (GRA) with dynamic programming [24] greedily assigns agent-task pairs until either all costs are below ξ or all agents are assigned. The output of GRA is then used to update the range of ξ to continue the half-interval search until convergence.

Algorithm description. The inputs required are M cost functions J_j that are all supermodular, a deployment size $N_d \geq M$, a relaxation constant $\alpha \geq 1$, and an initial non-redundant assignment $\mathcal{O}$ that can be found using traditional methods such as the thresholding algorithm for the linear bottleneck assignment problem. Lines 1-3 initialize the range of ξ and the output solution $\mathcal{A}_f$. The *while* loop beginning at line 5 is the half-interval search, whose interval bounds are updated in lines 8-13. Given a value ξ (line 6) and the average truncated cost function defined in line 1, the sub-problem in equation 5 is solved (line 7). The solution of this sub-problem, $\mathcal{A}_s$, is then used to update the half-interval search range. If $|\mathcal{A}_s| \leq \alpha(N_d - M)$, ξ is a feasible approximate solution to Problem 1. ξ_{max} is updated and $\mathcal{A}_s$ is kept as the best candidate solution, $\mathcal{A}_f$. If $|\mathcal{A}_s| > \alpha(N_d - M)$, there is no feasible solution to Problem 1 for the given value of ξ, and ξ_{min} is updated. For every iteration of the half-interval search, ξ_{min} is guaranteed to be infeasible, i.e., lower than minimum ξ in Problem 1. Therefore, the half-interval search converges to $\xi \leq \xi^*$ and the returned solution $\mathcal{A}_f$ is guaranteed to satisfy $|\mathcal{A}_f| \leq \alpha(N_d - M)$.

Theorem 1 states that the Fair Supermodular Redundant Assignment algorithm returns a solution $\mathcal{A}_f$ with a cost that is at worst equivalent to the cost achieved by $\mathcal{A}^*$ but a solution size that is up to α time larger, as defined in equation (6). The algorithm works for any value of $\alpha \geq 1$, but values differing from that defined in equation (6) will not have the guarantees stated in Theorem 1.

Theorem 1: The Fair Supermodular Redundant Assignment algorithm guarantees that

$$\max_j J_j(\mathcal{A}_f) \leq \min_{|\mathcal{A}| \leq N_d} \max_j J_j(\mathcal{A}) \quad \text{and} \quad |\mathcal{A}_f| \leq \alpha N_d$$

for a given N_d and $\alpha = 1 + \log(\max_j J_j(\emptyset) - 1)$. The number of function evaluations is

$$O\left((1 + \log_2(M \max_j J_j(\emptyset))\ (N_d - M)NMS \right)$$

where S is the number of samples taken from the cost distributions.

Proof: Let $\bar{J}(\mathcal{A}) = \frac{1}{M} \sum_j \hat{J}_j(\mathcal{A})$. Prorok's Proposition 3 [24] shows that for our sub-problem, $\alpha = 1 + \log \frac{\bar{J}(\emptyset) - \xi}{\bar{J}(\mathcal{A}_{k-1}) - \xi}$, where $\mathcal{A}_{k-1}$ is the set chosen by the greedy algorithm in its penultimate step (i.e., the next assignment by the greedy algorithm achieves ξ). Because $\bar{J}(\mathcal{A}_{k-1})$ can be arbitrarily close to ξ, the bound can tend towards infinity, preventing a ξ-independent bound on α. Assuming $J_j(\mathcal{A})$ only takes integer values, though, limits $\bar{J}(\mathcal{A}_{k-1}) - \xi$ to minimum value of one, and ensures $\xi \geq 1$ because costs are strictly positive, thus yielding a ξ-independent bound of

$$\alpha = 1 + \log\left(\max_j J_j(\emptyset) - 1 \right) \geq 1 + \log\left(\bar{J}(\emptyset) - \xi \right) \quad (6)$$

Algorithm 1: Fair Supermodular Redundant Assignment.

Input: $J_1, \ldots, J_M, N_d, \alpha, \mathcal{O}$
Output: Set of fair redundant assignments, $\mathcal{A}_f$
1: Define $\bar{J}(\mathcal{A}, \xi) = \frac{1}{M} \sum_j \max\{J_j(\mathcal{A}), \xi\}$
2: $\xi_{\text{min}} \leftarrow 0$
3: $\xi_{\text{max}} \leftarrow \max_j J_j(\emptyset)$
4: $\mathcal{A}_f \leftarrow \emptyset$
5: **while** $\xi_{\text{max}} - \xi_{\text{min}} \geq \frac{1}{M}$ **do**
6: $\quad \xi \leftarrow \frac{\xi_{\text{min}} + \xi_{\text{max}}}{2}$
7: $\quad \mathcal{A}_s \leftarrow \text{GRA}(\bar{J}, \xi, \mathcal{O})$
8: $\quad$ **if** $|\mathcal{A}_s| \leq \alpha(N_d - M)$ **then**
9: $\quad\quad \xi_{\text{max}} \leftarrow \xi$
10: $\quad\quad \mathcal{A}_f \leftarrow \mathcal{A}_s$
11: $\quad$ **else**
12: $\quad\quad \xi_{\text{min}} \leftarrow \xi$
13: $\quad$ **end if**
14: **end while**
15: **return** $\mathcal{A}_f$

This result is similar to that derived by Wolsey [34]. Additionally, Krause builds on Wolsey's result to show in Section 7.1 [15] that integer objective functions can be easily extended to take rational values by rounding by their highest order bits, allowing a small additive approximation error. Simulations in Section V show near-optimal performance without the need for this rounding.

The computational complexity comes from the number of times the half-interval search runs $(1 + \log_2(M \max_j J_j(\emptyset))$ [15] and the $(N_d - M)NMS$ evaluations in the inner loop sub-problem [24]. ■

V. Evaluation

We present Algorithm 1 as it applies the case study of robots with uncertain travel times assigned to spatially located tasks, such as in emergency supply delivery. The cost at each node is defined as the expected waiting time, $J_j(\mathcal{A}) = \mathbb{E}_c[\min_i\{C_{ij} | (i,j) \in \mathcal{A} \cup \mathcal{O}\}]$. The *minimum* operator here represents the *first-come first-to-serve* principle [24]. In other words, all robots have the emergency supplies (e.g., water or medicine) to delivery to any task, and whichever robot arrives to a task location first provides the supply thus accomplishing the task. This case study seeks to minimize expected waiting time of the worst task while ensuring the total deployment size (initial plus redundant assignment) is less than N_d. Our case study includes three sets of simulations.

The first two sets of simulations use random bipartite graphs to empirically showcase Theorem 1 and to demonstrate that Algorithm 1 outperforms benchmark algorithms. The bipartite graphs in these simulations are abstract representations of any problem that uses the first-come first-to-serve principle. The last set of simulations uses a random transport network to analyze a use case of Algorithm 1 and highlight the unfairness that can arise from utilitarian assignment.

A. Empirical Display of Theorem 1

In this section, we display and visualize Theorem 1 empirically by showcasing that Algorithm 1 yields assignments with more-than-optimal cost when using the cardinality relaxation

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:09:50 UTC from IEEE Xplore. Restrictions apply.

of $\alpha = 1 + \log(\max_j J_j(\emptyset) - 1)$ and near optimal cost when $\alpha = 1$. The former achieves super-optimality by using a deployment size that is larger than the desired deployment size, while the latter respects the desired deployment size but has no theoretical guarantees on performance (though empirically near-optimal in most cases).

To show these results, Algorithm 1 must be compared to the optimal assignment. Finding the optimal is extremely computationally expensive because it requires brute force search over the power set. Therefore, this simulation set is constrained to small examples. We represent a problem with 18 agents, 2 tasks, and deployment sizes $[3, 4, 5, 6, 7]$ as a random bipartite graph, $\mathcal{B} = (\mathcal{U}, \mathcal{F}, \mathcal{C})$. The small number of tasks and deployment sizes are chosen for computational tractability of the optimal solution, and the number of agents is chosen to allow for the cardinality relaxation.

The set of vertices $\mathcal{U}$ is partitioned into two disjoint sets $\mathcal{U}_a$ and $\mathcal{U}_t$, the set of agent nodes (size 18) and task nodes (size 2), respectively. The graph is fully connected, i.e., every node in $\mathcal{U}_a$ is connected to all nodes in $\mathcal{U}_t$. Each edge has a corresponding cost random variable: a truncated Gaussian distribution.[5] These edges represent the distribution of travel times from the agent location to the task location (e.g., in minutes). Means are uniformly sampled from the range $[15, 20]$, standard deviations uniformly sampled from the range $[5, 10]$, and all distributions are truncated at 5 (restricting the minimum travel time between any agent and task to 5).

Using these parameters, we create 1000 random bipartite graphs and implement Algorithm 1 using $\alpha = 1 + \log(\max_j J_j(\emptyset) - 1)$ and $\alpha = 1$, then calculate the brute force optimal solution. Fig. 2 shows the aggregate results over the 1000 trials. Because $\alpha = 1 + \log(\max_j J_j(\emptyset) - 1)$ depends on the problem instance, the value of α is not constant among trials and ranges from $[3.8, 4.2]$. The left plots show that the cardinality relaxation yields assignment sizes that are larger than the desired size but costs that are better than optimal. On the right, using $\alpha = 1$ respects the desired deployment size while achieving near-optimal performance empirically. [6]

B. Comparing Algorithm 1 to Benchmarks

In the following, we show that Algorithm 1 with $\alpha = 1$ outperforms three benchmarks: **(1)** non-redundant assignment using the thresholding algorithm, **(2)** random assignment of redundant agents, **(3)** repeated iterations of the thresholding algorithm for redundant assignment. We use $\alpha = 1$ since it is shown above to be near optimal without the need for cardinality relaxations. The results in Fig. 3 show that redundant agents improve the performance of non-redundant assignments and that Algorithm 1 outperforms all three benchmarks.

We create 1000 random bipartite graphs with the cost random variable parameters listed above. Because calculating the optimal is not needed in these simulations, larger graphs are studied.[7] We first consider 40 agents and 10 tasks, with increasing deployment sizes. These trials restrict the deployment size such that some agents are unused. Next, we consider 10 tasks with varying number of agents where all available agents are deployed, i.e., $N = N_d$. These two different problem setups are chosen to show that Algorithm 1 performs well in scenarios when partial and full deployment is desired. In both cases, Algorithm 1 outperforms the benchmarks. [8]

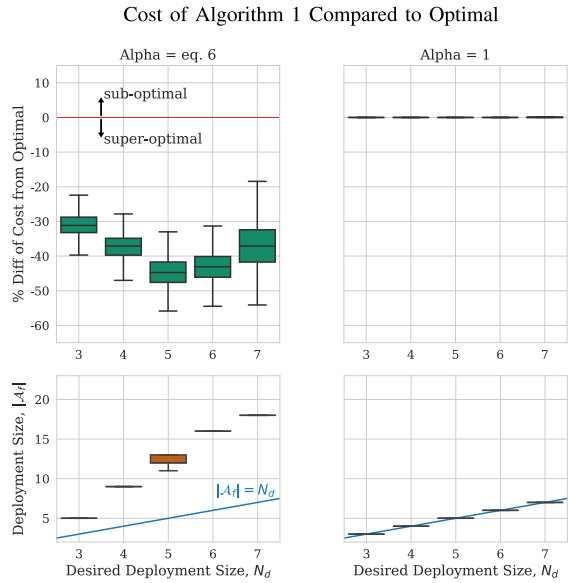


Fig. 2. *Top.* The percent difference in cost between the optimal assignment and the assignments produced by Algorithm 1 displays our above proof that the α value proven in Theorem 1 (*left*) yields super-optimal cost by using larger than optimal assignments,and shows that $\alpha = 1$ (*right*) yields near optimal results. Note that **positive values are sub-optimal** (*top*) because the problem is a minimization. Outliers are omitted. *Bottom.* The comparison of the actual deployment size to the deployment size constraint shows that super-optimality is achieved by relaxing the constraint, yielding larger assignments and that $\alpha = 1$ respects the deployment size constraint.

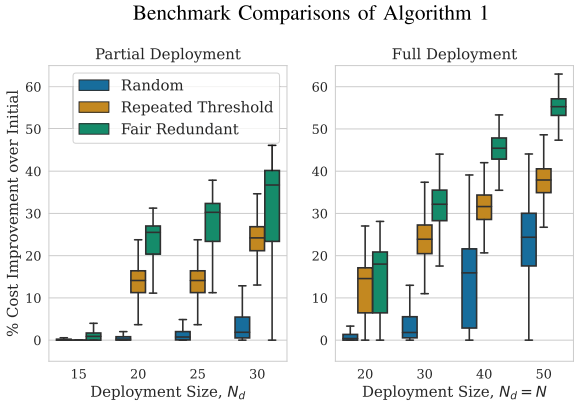


Fig. 3. Fair redundant assignment yields the largest improvement in maximum task cost compared to benchmarks. The plots above shows the improvement in cost for random redundant assignment, repeated threshold assignment, and fair redundant assignment. The data show the percent difference between the respective assignment and the initial non-redundant assignment. *Left.* $M = 10$ tasks, $N = 40$ agents, and varying N_d deployment sizes (*x-axis*). *Right.* $M = 10$ tasks with varying number of agents, where the deployment size equals the number of agents, $N = N_d$.

[5]Algorithm 1 does not rely on any Gaussian assumption. Any arbitrary distribution, discrete or continuous, can be used.

[6]The majority of trials using $\alpha = 1$ are optimal; only the unshown outliers are sub-optimal, which lie within 10% of optimal in these simulations.

[7]Graphs with tens of agents and tasks are shown in this letter, but the algorithm has been run on examples with hundreds of agents and tasks.

[8]Not shown here is Algorithm 1 with the α from equation (6), which is guaranteed perform at least as good as $\alpha = 1$, but comes at the cost of larger assignments. In the right of Fig. 3, both α values perform equally because there are no available agents to be used by the cardinality relaxation.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:09:50 UTC from IEEE Xplore. Restrictions apply.

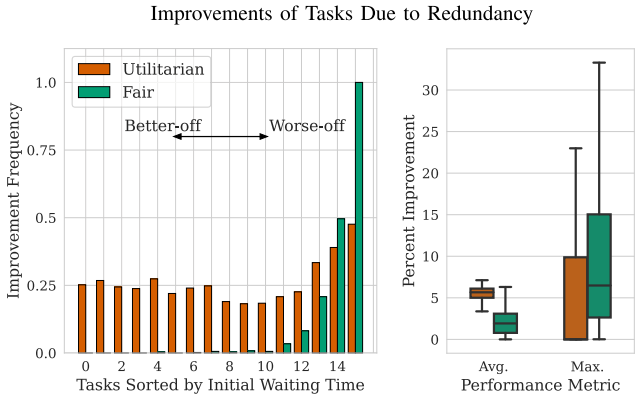


Fig. 4. *Left.* The frequency at which different tasks are improved by redundant assignment under the utilitarian and fair algorithms across 500 examples. The tasks are ordered by increasing expected cost after the initial assignment, showing that the fair approach always helps the tasks that are worse off while the greedy approach more uniformly helps tasks. *Right.* The percent improvement in the average and maximum task cost compared to those of the initial assignment shows that the fair assignment does not improve the average as much as does the utilitarian assignment but strongly outperforms in improvement of the maximum cost.

C. Random Transport Network Case Study

We instantiate 500 random transport networks with 32 agents and 16 tasks randomly located at different nodes where the travel time along each edge is represented by a Gaussian random variable truncated at zero with mean uniformly sampled from the range $[10, 20]$ and standard deviation uniformly sampled from the range $[5, 10]$. Assignments of size 20 are found using Algorithm 1 and the utilitarian approach.[9]

Fig. 4 shows that the utilitarian approach nearly uniformly helps tasks, while Algorithm 1 focuses the redundant agents on helping tasks most in need (those with the highest task cost). Additionally, for more than half the trials, the utilitarian approach does not help the worst-off task. All redundant assignment is guaranteed to improve the average task cost because the redundant agents improve the performance of their assigned tasks. While utilitarian redundant assignment better improves the average task cost, it does not necessarily improve the maximum cost. Fair redundant assignment is guaranteed to improve both the maximum and the average task cost.

VI. Discussion

This work is inspired by robotic applications such as rescue and medical or emergency supplies delivery. The objectives of such applications focus on task completion time and omit other costs (e.g., energy used). While previous work focuses on average performance, we observe that the human-centric nature of these applications requires considerations of fairness. While our work contributes to fairness in redundant assignment by improving the objective function, we acknowledge that fairness is more than a mathematical formulation and we welcome discussions of the broader impact of these algorithms.

For practitioners, note the similarity of our problem formulation and approach to works on providing robustness to a worst-case objective. This formulation is flexible in that it has few assumptions, such as the supermodularity of the cost functions. Thus far, work in redundant assignment has focused on the first-come first-to-serve principle. However, Algorithm 1 can be applied to many multi-agent problems where uncertainty impacts task performance, requiring new collaboration functions, $J_j(\mathcal{A})$.

When studying other applications, scalability must be considered. Theorem 1 shows that the complexity of Algorithm 1 is dominated by the greedy term, which scales linearly with the dominant term, N, because the number of tasks, M, and degree of redundancy, N_d are small. In instances where both N_d and N are large, the scaling is approximately quadratic. In such cases, our centralized approach is less tractable, and therefore could be adapted via hierarchical approaches or by developing new decentralized algorithms.

Last, we note that existing approaches in redundant assignment, including this work, assume an initial non-redundant assignment. Therefore, no theoretical guarantees have been proven for the full assignment of initial and redundant agents thus far. Doing so will require new algorithms that do not rely on the initial assignment to maintain supermodularity and that ensure that all tasks are assigned at least one agent.

VII. Conclusion

Redundant assignment provides robustness against uncertainty and improves task performance. While redundancy comes at the cost of using more robots, it is beneficial in time sensitive scenarios such as rescue or emergency delivery.

Inspired by the human-centric nature of these application spaces (e.g., rescue), we build on previous work in redundant assignment to consider fairness; notably, we are first to formalize fairness in this space. Using a Rawlsian approach to fairness, we formulate fair redundant task allocation as the optimization of worst-case task cost with a cardinality constraint, a problem that is NP-Hard. We exploit the natural supermodularity of the problem to propose a tractable solution.

Algorithm 1 is a binary search where the target value is determined by greedily solving the relaxed sub-problem shown in 5. This algorithm converges to an approximate solution: an assignment that is α times larger than the cardinality constraint. We prove performance guarantees of this algorithm, showing that the returned approximate solution has a cost that is less than that of the optimal solution but a solution size that is α times larger. Theorem 1 proves bounds on this α value.

Additionally, we show empirically that Algorithm 1 with $\alpha = 1$ (meaning the cardinality constraint remains satisfied) provides near optimal results, despite having no theoretical guarantees. Simulated experiments show that this approach outperforms benchmarks, scales to large problem instances, and provides both increased fairness and increased utilitarian social welfare over non-redundant assignments.

Appendix

A. Supermodularity

Supermodularity 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 is known as the property of diminishing returns; there are diminishing returns from an added element x as the set it is added to grows larger. Below are the formal definitions of supermodularity [24].

[9]Both algorithms require an initial non-redundant assignment. To compare the results on the same footing, both algorithms are given the same initial assignment: Hungarian assignment on mean travel times.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:09:50 UTC from IEEE Xplore. Restrictions apply.

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 $x \in \mathcal{F} \backslash \mathcal{A}$ is:

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

Definition 3: Supermodularity. Let $J : 2^{\mathcal{F}} \mapsto \mathbb{R}$ and $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{F}$. The set function J is supermodular[10] if and only if for any $x \in \mathcal{F} \backslash \mathcal{B}$

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

Lemma 1: The maximum of a set of supermodular functions is not supermodular.

Proof: Consider an assignment set $\mathcal{A}$ and an element x to be added. Let x decrease the value of $J_1(\mathcal{A})$. Here, x is an assignment of a new agent to task 1, thus decreasing the expected cost of task 1. If J_1 is not the maximum of the all J_j, the addition of x to $\mathcal{A}$ has no effect on $\max_j(J_j(\mathcal{A}))$. In other words, its marginal decrease of $\max_j(J_j(\mathcal{A}))$ with respect to x is zero. By Definition 3, $\max_j(J_j)$ is only supermodular if the marginal decrease of $\max_j(J_j)$ with respect to x is also zero for **all supersets** of $\mathcal{A}$.

Let $\mathcal{B}$ be a superset of $\mathcal{A}$ that improves upon all task costs except task 1, such that the maximum cost of $\mathcal{B}$ is J_1. Here $\mathcal{B} \supseteq \mathcal{A}$ such that $\max_j(J_j(\mathcal{B})) = J_1(\mathcal{B}) = J_1(\mathcal{A})$.[11] Since the assignment x decreases the value of $J_1(\mathcal{A})$, it decreases the value of $\max_j(J_j(\mathcal{B}))$. Therefore, the marginal decrease of adding x to $\mathcal{B}$ is nonzero, meaning $\max_j(J_j)$ is not supermodular. ■

REFERENCES

[1] E. Ackerman and E. Strickland, "Medical delivery drones take flight in east africa," *IEEE Spectr.*, vol. 55, no. 1, pp. 34–35, Jan. 2018.

[2] C. Arnsperger, "Envy-freeness and distributive justice," *J. Econ. Surv.*, vol. 8, no. 2, pp. 155–186, 1994.

[3] S. Barocas, M. Hardt, and A. Narayanan, "Fairness in machine learning," *Neural Inf. Process. Syst. Tut.*, vol. 1, p. 2, 2017.

[4] R. K. Bellamy *et al.*, "AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias," *IBM J. Res. Develop.*, vol. 63, no. 4, pp. 4:1–4:15, July-Sep.

[5] V. Borkar and R. Jain, "Risk-constrained Markov decision processes," *IEEE Trans. Autom. Control*, vol. 59, no. 9, pp. 2574–2579, Feb. 2014.

[6] S. J. Brams and A.D. Taylor, *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[7] S. M. Chaves, J. M. Walls, E. Galceran, and R. M. Eustice, "Risk aversion in belief-space planning under measurement acquisition uncertainty," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots Syst.*, 2015, pp. 2079–2086.

[8] M. L. Cummings and S. Li, "Subjectivity in the creation of machine learning models," *J. Data Inf. Qual.*, 2020. [Online]. Available: https://hal.pratt.duke.edu/publications

[9] L. Edwards and M. Veale, "Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for," *Duke L. Tech. Rev.*, vol. 16, pp. 18, 2017.

[10] D. Gale, *The Theory of Linear Economic Models*. Chicago, IL, USA: Univ. of Chicago Press, 1989.

[11] M. R. Garey and D. S. Johnson, "Strong NP-completeness results: Motivation, examples, and implications," *J. ACM (JACM)*, vol. 25, no. 3, pp. 499–508, 1978 .

[12] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int. J. Robot. Res.*, vol. 23, no. 9, pp. 939–954, 2004.

[13] J. C. Harsanyi, "Bayesian decision theory and utilitarian ethics," *Amer. Econ. Rev.*, vol. 68, no. 2, pp. 223–228, 1978.

[14] D. K. Herreiner and C. D. Puppe, "Envy freeness in experimental fair division problems," *Theory Decis.*, vol. 67, no. 1, pp. 65–100, 2009.

[15] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *J. Mach. Learn. Res.*, vol. 9, pp. 2761–2801, Dec. 2008.

[16] A. Majumdar and M. Pavone, "How should a robot assess risk? Towards an axiomatic theory of risk in robotics," in *Proc. Int. Symp. on Robotics Research*, Puerto Varas, Chile, Dec. 2017. [Online]. Available: /wp-content/papercite-data/pdf/Majumdar.Pavone.ISRR17.pdf

[17] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," In*CoRR*. 2019, *arXiv:1908.09635*.

[18] B. D. Mittelstadt, P. Allo, M. Taddeo, S. Wachter, and L. Floridi, "The ethics of algorithms: Mapping the debate," *Big Data Soc.*, vol. 3, no. 2, 2016, Art. no. 2053951716679679.

[19] H. Moulin, *Fair Division and Collective Welfare*. Cambridge, MA, USA: MIT Press, 2004.

[20] C. Nam and D. A. Shell, "Analyzing the sensitivity of the optimal assignment in probabilistic multi-robot task allocation," *IEEE Robot. Automat. Lett.*, vol. 2, no. 1, pp. 193–200, Jan. 2017.

[21] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions–I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.

[22] A. C. Pigou, "The laws of diminishing and increasing cost," *Econ. J.*, vol. 37, no. 146, pp. 188–197, 1927.

[23] T. Powers, J. Bilmes, S. Wisdom, D. W. Krout, and L. Atlas, "Constrained robust submodular optimization," in *NIPS OPT2016 Workshop*, 2016.

[24] A. Prorok, "Robust assignment using redundant robots on transport networks with uncertain travel time," *IEEE Trans. Automat. Sci. Eng.*, vol. 17, no. 4, pp. 2025–2037, Oct. 2020.

[25] A. Prorok and V. Kumar, "Privacy-preserving vehicle assignment for mobility-on-demand systems," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots Syst.*, 2017, pp. 1869–1876.

[26] J. Rawls, *A Theory of Justice*. Cambridge, MA, USA: Harvard Univ. Press, 2009.

[27] K. Shahriari and M. Shahriari, "IEEE standard review – ethically aligned design: A vision for prioritizing human wellbeing with artificial intelligence and autonomous systems," in *Proc. IEEE Canada Intl. Humanitarian Technol. Conf.*, 2017, pp. 197–201.

[28] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.

[29] W. Suksompong, "Asymptotic existence of proportionally fair allocations," *Math. Social Sci.*, vol. 81, pp. 62–65, 2016.

[30] W. Szpankowski, "On the bottleneck and capacity assignment problems," *Dept. Comput. Sci. Tech. Reports*, 1988, Paper 719.

[31] S. Verma and J. Rubin, "Fairness definitions explained," in *Proc. IEEE/ACM Intl. Workshop Softw. Fairness*, 2018, pp. 1–7.

[32] S. Wachter, B. Mittelstadt, and L. Floridi, "Why a right to explanation of automated decision-making does not exist in the general data protection regulation," *Intl. Data Privacy Law*, vol. 7, no. 2, pp. 76–99, 2017.

[33] B. Wilder, "Risk-sensitive submodular optimization," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6451–6458.

[34] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[35] F. Yang and N. Chakraborty, "Algorithm for optimal chance constrained linear assignment," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 801–808.

[36] C. Zhang and J. A. Shah, "On fairness in decision-making under uncertainty: definitions, computation, and comparison," in *Proc. AAAI Conf. Artif. Intell.*, 2015.

[37] L. Zhou and P. Tokekar, "An approximation algorithm for risk-averse submodular optimization," in *Proc. Intl. Workshop Algorithmic Foundations Robot.*, Berlin, Germany: Springer, 2018, pp. 144–159.

[38] L. Zhou and P. Tokekar, "Sensor assignment algorithms to improve observability while tracking targets," *IEEE Trans. Robot.*, vol. 35, no. 5, pp. 1206–1219, Oct. 2019.

[10] Note, a supermodular function J implies $-J$ is submodular.

[11] This is feasible because the cost distributions are unconstrained.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:09:50 UTC from IEEE Xplore. Restrictions apply.