An Approximation Algorithm for Risk-averse Submodular Optimization

Lifeng Zhou and Pratap Tokekar

Department of Electrical & Computer Engineering, Virginia Tech, USA.
{lfzhou, tokekar}@vt.edu

Abstract. We study the problem of incorporating risk while making combinatorial decisions under uncertainty. We formulate a discrete submodular maximization problem for selecting a set using Conditional-Value-at-Risk (CVaR), a risk metric commonly used in financial analysis. While CVaR has recently been used in optimization of linear cost functions in robotics, we take the first stages towards extending this to discrete submodular optimization and provide several positive results. Specifically, we propose the Sequential Greedy Algorithm that provides an approximation guarantee on finding the maxima of the CVaR cost function under a matroidal constraint. The approximation guarantee shows that the solution produced by our algorithm is within a constant factor of the optimal and an additive term that depends on the optimal. Our analysis uses the curvature of the submodular set function, and proves that the algorithm runs in polynomial time. This formulates a number of combinatorial optimization problems that appear in robotics. We use two such problems, vehicle assignment under uncertainty for mobility-on-demand and sensor selection with failures for environmental monitoring, as case studies to demonstrate the efficacy of our formulation.

1 Introduction

Combinatorial optimization problems find a variety of applications in robotics. Typical examples include:

- *Sensor placement:* Where to place sensors to maximally cover the environment [1] or reduce the uncertainty in the environment [2]?
- *Task allocation:* How to allocate tasks to robots to maximize the overall utility gained by the robots [3]?
- *Combinatorial auction:* How to choose a combination of items for each player to maximize the total rewards [4]?

Algorithms for solving such problems find use in sensor placement for environment monitoring [1, 2], robot-target assignment and tracking [5–7], and informative path planning [8]. The underlying optimization problem in most cases can be written as:

$$\max_{\mathcal{S}\in\mathcal{I},\mathcal{S}\in\mathcal{X}} f(\mathcal{S}), \tag{1}$$

where $\mathcal{X}$ denotes a ground set from which a subset of elements S must be chosen. f is a monotone submodular utility function [9, 10]. Submodularity is the property of diminishing returns. Many information theoretic measures, such as mutual information [2], and geometric measures such as the visible area [11], are known to be submodular. $\mathcal{I}$ denotes a matroidal constraint [9, 10]. Matroids are a powerful combinatorial tool that can represent constraints on the solution set, e.g., cardinality constraints ("place no more than k sensors") and connectivity constraints ("the communication graph of the robots must be connected") [12]. The objective of this problem is to find a set $\mathcal{S}$ satisfying a matroidal constraint $\mathcal{I}$ and maximizing the utility $f(\mathcal{S})$. The general form of this problem is NP-complete. However, a greedy algorithm yields a constant factor approximation guarantee [9, 10].

In practice, sensors can fail or get compromised [13] or robots may not know the exact positions of the targets [14]. Hence, the utility $f(\mathcal{S})$ is not necessarily deterministic but can have uncertainty. Our main contribution is to extend the traditional formulation given in Eq. 1 to also account for the uncertainty in the actual cost function. We model the uncertainty by assuming that the utility function is of the form $f(\mathcal{S}, y)$ where $\mathcal{S} \in \mathcal{X}$ is the decision variable and $y \in \mathcal{Y}$ represents a random variable which is independent of $\mathcal{S}$. We focus on the case where $f(\mathcal{S}, y)$ is monotone submodular in $\mathcal{S} \in \mathcal{X}$ and integrable in y.

The traditional way of stochastic optimization is to use the expected utility as the objective function: $\max_{\mathcal{S}\in\mathcal{I},\mathcal{S}\in\mathcal{X}} \mathbb{E}_y[f(\mathcal{S}, y)]$. Since the sum of the monotone submodular functions is monotone submodular, $\mathbb{E}_y[f(\mathcal{S}, y)]$ is still monotone submodular in $\mathcal{S}$. Thus, the greedy algorithm still retains its constant-factor performance guarantee [9, 10]. Examples of this approach include influence maximization [15], moving target detection and tracking [14], and robot assignment with travel-time uncertainty [16].

While optimizing the expected utility has its uses, it also has its pitfalls. Consider the example of mobility-on-demand where two self-driving vehicles, v_1 and v_2, are available to pick up the passengers at a demand location (Fig. 1). v_1 is closer to the demand location, but it needs to cross an intersection where it may need to stop and wait. v_2 is further from the demand location but there is no intersection along the path. The travel time for v_1 follows a bimodal distribution (with and without traffic stop) whereas that for v_2 follows a unimodal distribution with a higher mean but lower uncertainty. Clearly, if the passenger uses the expected travel

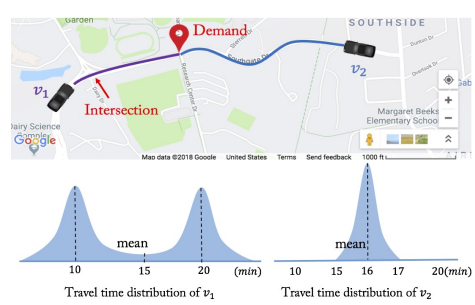


Fig. 1. Mobility on demand with travel time uncertainty of self-driving vehicles.

time as the objective, they would choose v_1. However, they will risk waiting a much longer time, i.e., $17 \sim 20$ *min* about half of the times. A more risk-averse passenger would choose v_2 which has higher expected waiting time 16 *min* but a lesser risk of waiting longer.

Thus, in these scenarios, it is natural to go beyond expectation and focus on a risk-averse measure. One popular coherent risk measure is *Conditional-Value-at-Risk* (CVaR) [17, 18]. CVaR takes a risk level α which is the probability of the worst α-tail cases. Loosely speaking, maximizing CVaR is equivalent to maximizing the expectation of the worst α-tail scenarios.[1] This risk-averse decision is rational especially when the failures can lead to unrecoverable consequences, such as a sensor failure.

Related work. Yang and Chakraborty studied a chance-constrained combinatorial optimization problem that takes into account the risk in multi-robot assignment [19]. They later extended this to knapsack problems [20]. They solved the problem by transforming it to a risk-averse problem with mean-variance measure [21]. Chance-constrained optimization is similar to optimizing the Value-at-Risk (VaR), which is another popular risk measure in finance [22]. However, Majumdar and Pavone argued that CVaR is a better measure to quantify risk than VaR or mean-variance based on six proposed axioms in the context of robotics [23].

Several works have focused on optimizing CVaR. In their seminal work [18], Rockafellar and Uryasev presented an algorithm for CVaR minimization for reducing the risk in financial *portfolio optimization* with a large number of instruments. Note that, in portfolio optimization, we select a distribution over available decision variables, instead of selecting a single one. Later, they showed the advantage of optimizing CVaR for general loss distributions in finance [24].

When the utility is a discrete submodular set function, i.e., $f(\mathcal{S}, y)$, Maehara presented a negative result for maximizing CVaR [25]— there is no polynomial time multiplicative approximation algorithm for this problem under some reasonable assumptions in computational complexity. To avoid this difficulty, Ohsaka and Yoshida in [26] used the same idea from portfolio optimization and proposed a method of selecting a distribution over available sets rather than selecting a single set, and gave a provable guarantee. Following this line, Wilder considered a CVaR maximization of a continuous submodular function instead of the submodular set functions [27]. They gave a $(1 - 1/e)$–approximation algorithm for continuous submodular functions and also evaluated the algorithm for discrete submodular functions using portfolio optimization [26].

Contributions. We focus on the problem of selecting a single set, similar to [25], to maximize CVaR rather than portfolio optimization [26, 27]. This is because we are motivated by applications where a one-shot decision (placing sensors and assigning vehicles) must be taken. Our contributions are as follows:

- We propose the Sequential Greedy Algorithm (SGA) which uses the deterministic greedy algorithm [9, 10] as a subroutine to find the maximum value of CVaR (Algorithm 1).

[1] We formally review CVaR and other related concepts in Section 2.1

- We prove that the solution found by SGA is within a constant factor of the optimal performance along with an additive term which depends on the optimal value. We also prove that SGA runs in polynomial time (Theorem 1) and the performance improves as the running time increases.
- We demonstrate the utility of the proposed CVaR maximization problem through two case studies (Section 3.2). We evaluate the performance of SGA through simulations (Section 5).

Organization of rest of the paper. We give the necessary background knowledge for the rest of the paper in Section 2. We formulate the CVaR submodular maximization problem with two case studies in Section 3. We present SGA along with the analysis of its computational complexity and approximation ratio in Section 4. We illustrate the performance of SGA to the two case studies in Section 5. We conclude the paper in Section 6.

2 Background and Preliminaries

We start by defining the conventions used in the paper.

Calligraphic font denotes a set (e.g., $\mathcal{A}$). Given a set $\mathcal{A}$, $2^{\mathcal{A}}$ denotes its power set. $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$. Given a set $\mathcal{B}$, $\mathcal{A} \setminus \mathcal{B}$ denotes the set of elements in $\mathcal{A}$ that are not in $\mathcal{B}$. $\Pr[\cdot]$ denotes the probability of an event and $\mathbb{E}[\cdot]$ denotes the expectation of a random variable. $\lceil x \rceil = \min\{n \in \mathbb{Z} | x \leq n\}$ where $\mathbb{Z}$ denotes the set of integers.

Next, we give the background on set functions (in the appendix file) and risk measures.

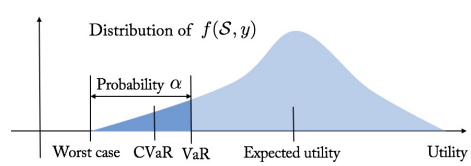


Fig. 2. An illustration of risk measures: VaR and CVaR.

2.1 Risk measures

Let $f(\mathcal{S}, y)$ be a utility function with decision set $\mathcal{S}$ and the random variable y. For each $\mathcal{S}$, the utility $f(\mathcal{S}, y)$ is also a random variable with a distribution induced by that of y. First, we define the Value-at-Risk at risk level $\alpha \in (0, 1]$.
Value at Risk:

$$\mathrm{VaR}_\alpha(\mathcal{S}) = \inf\{\tau \in \mathbb{R}, \Pr[f(\mathcal{S}, y) \leq \tau] \geq \alpha\}. \tag{2}$$

Thus, $\mathrm{VaR}_\alpha(\mathcal{S})$ denotes the left endpoint of the α-quantile(s) of the random variable $f(\mathcal{S}, y)$. The Conditional-Value-at-Risk is the expectation of this set of α-worst cases of $f(\mathcal{S}, y)$, defined as:

Conditional Value at Risk:

$$\mathrm{CVaR}_\alpha(\mathcal{S}) = \mathop{\mathbb{E}}_{y}[f(\mathcal{S}, y) | f(\mathcal{S}, y) \leq \mathrm{VaR}_\alpha(\mathcal{S})]. \tag{3}$$

Fig. 2 shows an illustration of $\mathrm{VaR}_\alpha(\mathcal{S})$ and $\mathrm{CVaR}_\alpha(\mathcal{S})$. $\mathrm{CVaR}_\alpha(\mathcal{S})$ is more popular than $\mathrm{VaR}_\alpha(\mathcal{S})$ since it has better properties [18], such as *coherence* [28].

When optimizing $\mathrm{CVaR}_\alpha(\mathcal{S})$, we usually resort to an auxiliary function:

$$H(\mathcal{S}, \tau) = \tau - \frac{1}{\alpha}\mathbb{E}[(\tau - f(\mathcal{S}, y))_+].$$

We know that optimizing $\mathrm{CVaR}_\alpha(\mathcal{S})$ over $\mathcal{S}$ is equivalent to optimizing the auxiliary function $H(\mathcal{S}, \tau)$ over $\mathcal{S}$ and τ [18]. The following lemmas give useful properties of the auxiliary function $H(\mathcal{S}, \tau)$.

Lemma 1. *If $f(\mathcal{S}, y)$ is normalized, monotone increasing and submodular in set $\mathcal{S}$ for any realization of y, the auxiliary function $H(\mathcal{S}, \tau)$ is monotone increasing and submodular, but not necessarily normalized in set $\mathcal{S}$ for any given τ.*

We provide the proofs for all the Lemmas and Theorem in the appendix.

Lemma 2. *The auxiliary function $H(\mathcal{S}, \tau)$ is concave in τ for any given set $\mathcal{S}$.*

Lemma 3. *For any given set $\mathcal{S}$, the gradient of the auxiliary function $H(\mathcal{S}, \tau)$ with respect to τ fulfills: $-(\frac{1}{\alpha} - 1) \leq \frac{\partial H(\mathcal{S},\tau)}{\partial \tau} \leq 1$.*

3 Problem Formulation and Case Studies

We first formulate the CVaR submodular maximization problem and then present two applications which we use as case studies.

3.1 Problem Formulation

CVaR Submodular Maximization: We consider the problem of maximizing $\mathrm{CVaR}_\alpha(\mathcal{S})$ over a decision set $\mathcal{S} \subseteq \mathcal{X}$ under a matroid constraint $\mathcal{S} \in \mathcal{I}$. We know that maximizing $\mathrm{CVaR}_\alpha(\mathcal{S})$ over $\mathcal{S}$ is equivalent to maximizing the auxiliary function $H(\mathcal{S}, \tau)$ over $\mathcal{S}$ and τ [18]. Thus, we propose the maximization problem as:

Problem 1.

$$\begin{aligned} &max \;\; \tau - \frac{1}{\alpha}\mathbb{E}[(\tau - f(\mathcal{S}, y))_+] \\ &s.t. \;\; \mathcal{S} \in \mathcal{I}, \mathcal{S} \subseteq \mathcal{X}, \tau \in [0, \Gamma], \end{aligned} \tag{4}$$

where Γ is the upper bound of the parameter τ. Problem 1 gives a risk-averse version of maximizing submodular set functions.

3.2 Case Studies

The risk-averse submodular maximization has many applications, as it has been written in Section 3.2. We describe two specific applications which we will use in the simulations.

Resilient Mobility-on-Demand Consider a mobility-on-demand problem where we assign R vehicles to N demand locations under arrival-time uncertainty. An example is shown in Fig. 3 where seven self-driving vehicles must be assigned to three demand locations to pick up passengers. We follow the same constraint setting in [16]— each vehicle can be assigned to at most one demand but multiple vehicles can be assigned to the same demand. Only the vehicle that arrives first is chosen for picking up the passengers. Note that the advantage of the redundant assignment to each demand is that it counters the effect of uncertainty and reduces the waiting time at demand locations [16]. This may be too conservative for consumer mobility-on-demand services but can be crucial for urgent and time-critical tasks such as delivery medical supplies [29].

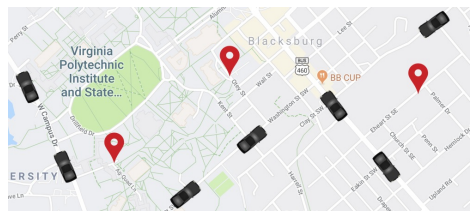
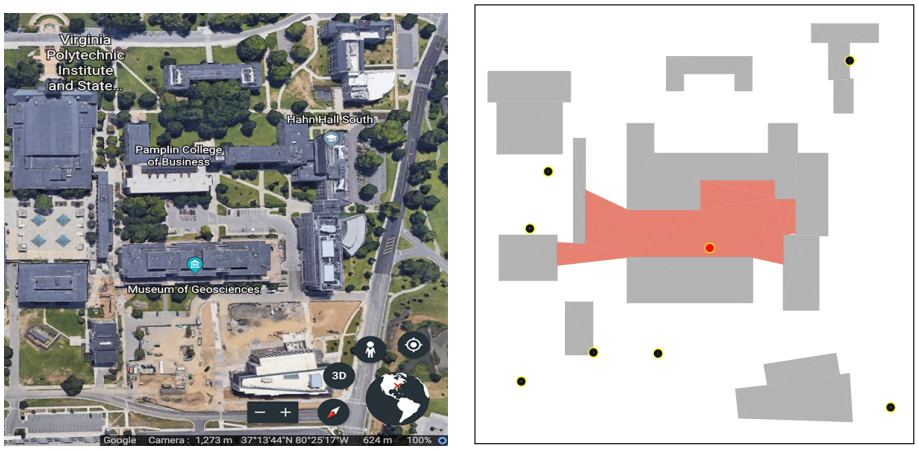


Fig. 3. Mobility-on-demand with multiple demands and multiple self-driving vehicles.

Assume the arrival time for the robot to arrive at demand location is a random variable. The distribution can depend on the mean-arrival time. For example, it is possible to have a shorter path that passes through many intersections, which leads to an uncertainty on arrival time. While a longer road (possibly a highway) has a lower arrival time uncertainty. Note that for each demand location, there is a set of robots assigned to it. The vehicle selected at the demand location is the one that arrives first. Then, this problem becomes a minimization one since we would like to minimize the arrival time at all demand locations. We convert it into a maximization one by taking the reciprocal of the arrival time. Specifically, we use the arrival *efficiency* which is the reciprocal of arrival time. Instead of selecting the vehicle at the demand location with minimum arrival time, we select the vehicle with maximum arrival efficiency. The arrival efficiency is also a random variable, and has a distribution depending on mean-arrival efficiency. Denote the arrival efficiency for robot $j \in \{1, ..., R\}$ arriving at demand location $i \in \{1, ..., N\}$ as e_{ij}. Denote the assignment utility as the arrival efficiency at all locations, that is,

$$f(\mathcal{S}, y) = \sum_{i \in N} \max_{j \in \mathcal{S}_i} e_{ij} \tag{5}$$

with $\bigcup_{i=1}^{N} \mathcal{S}_i = \mathcal{S}$ and $\mathcal{S}_i \cap \mathcal{S}_k = \emptyset,\ i, k \in \{1, \cdots, N\}$. $\mathcal{S}_i \cap \mathcal{S}_k = \emptyset$ indicates the selected set $\mathcal{S}$ satisfies a partition matroid constraint, $\mathcal{S} \in \mathcal{I}$, which represents that each robot can be assigned to at most one demand. The assignment utility $f(\mathcal{S}, y)$ is monotone submodular in $\mathcal{S}$ due to the "max" function. $f(\mathcal{S}, y)$ is normalized since $f(\emptyset, y) = 0$. Here, we regard the uncertainty as a risk. Our risk-averse assignment problem is a trade-off between efficiency and uncertainty. Our goal is to maximize the total efficiencies at the demand locations while considering the risk from uncertainty.



(a) Part of Virgina Tech campus from Google Earth.

(b) Top view of part of a campus and ground sensor's visibility region.

Fig. 4. Campus monitoring by using a set of sensors with visibility regions.

Robust Environment Monitoring Consider an environment monitoring problem where we monitor part of a campus with a group of ground sensors (Fig. 4). Given a set of N candidate positions $\mathcal{X}$, we would like to choose a subset of M positions $\mathcal{S} \subseteq \mathcal{X}, M \leq N$, to place visibility-based sensors to maximally cover the environment. The visibility regions of the ground sensors are obstructed by the buildings in the environment (Fig. 4-(b)). Consider a scenario where the probability of failure of a sensor depends on the area it can cover. That is, a sensor covering a larger area has a larger risk of failure associated with it. This may be due to the fact that the same number of pixels are used to cover a larger area and therefore, each pixel covers proportionally a smaller footprint. As a result, the sensor risks missing out on detecting small objects.

Denote the probability of success and the visibility region for each sensor $i, i \in \{1, ..., N\}$ as p_i and v_i, respectively. Thus, the polygon each sensor i monitors is also a random variable. Denote this random polygon as A_i and denote the

selection utility as the joint coverage area of a set of sensors, $\mathcal{S}$, that is,

$$f(\mathcal{S}, y) = \text{area}(\bigcup_{i=1:M} A_i),\ i \in \mathcal{S}, \mathcal{S} \subseteq \mathcal{I}. \tag{6}$$

The selection utility $f(\mathcal{S}, y)$ is monotone submodular in $\mathcal{S}$ due to the overlapping area. $f(\mathcal{S}, y)$ is normalized since $f(\emptyset, y) = 0$. Here, we regard the sensor failure as a risk. Our robust environment monitoring problem is a trade-off between area coverage and sensor failure. Our goal is to maximize the joint-area covered while considering the risk from sensor failure.

4 Algorithm and Analysis

We present the Sequential Greedy Algorithm (SGA) for solving Problem 1 by leveraging the useful properties of the auxiliary function $H(\mathcal{S}, \tau)$. The pseudo-code is given in Algorithm 1. SGA mainly consists of searching for the appropriate value of τ by solving a subproblem for a fixed τ under a matroid constraint. Even for a fixed τ, the subproblem of optimizing the auxiliary function is NP-complete. Nevertheless, we can employ the greedy algorithm for the subproblem, and sequentially apply it for searching over all τ. We explain each stage in detail next.

4.1 Sequential Greedy Algorithm

These are four stages in SGA:

a) Initialization (line 1): Algorithm 1 defines a storage set $\mathcal{M}$ and initializes it to be the empty set. Note that, for each specific τ, we can use the greedy algorithm to obtain a near-optimal solution $\mathcal{S}^G$ based on the monotonicity and submodularity of the auxiliary function $H(\mathcal{S}, \tau)$. $\mathcal{M}$ stores all the $(\mathcal{S}^G, \tau)$ pairs when searching all the possible values of τ.

*b) Searching for τ (**for** loop in lines 2–10):* We use a user-defined separation Δ (line 3) to sequentially search for all possible values of τ within $[0, \Gamma]$. Γ is an upper bound on τ and can be set by the user based on the specific problem at hand. We show how to find Γ for the specific cases in Section 5.

c) Greedy algorithm (lines 4–8): For a specific τ, say τ_i, we use the greedy approach to choose set $\mathcal{S}_i^G$. We first initialize set $\mathcal{S}_i^G$ to be the empty set (line 4). Under a matroid constraint, $\mathcal{S}_i^G \in \mathcal{I}$ (line 5), we add a new element s which gives the maximum marginal gain of $H(\mathcal{S}_i^G, \tau_i)$ (line 6) into set $\mathcal{S}_i^G$ (line 7) in each round.

d) Find the best pair (line 11): Based on the collection of pairs $\mathcal{M}$ (line 9), we pick the pair $(\mathcal{S}_i^G, \tau_i) \in \mathcal{M}$ that maximizes $H(\mathcal{S}_i^G, \tau_i)$ as the final solution S_i^G. We denote this value of τ by τ^G.

Algorithm 1: Sequential Greedy Algorithm for Problem 1.

Input:
- Ground set $\mathcal{X}$ and matroid $\mathcal{I}$
- User-defined risk level $\alpha \in [0,1]$
- Range of the parameter $\tau \in [0, \Gamma]$ and discretization stage $\Delta \in (0, \Gamma]$
- An oracle $\mathcal{O}$ that evaluates $H(\mathcal{S}, \tau)$

Output:
- Selected set $\mathcal{S}^G$ and corresponding parameter τ^G

$\mathcal{M} \leftarrow \emptyset$
for $i = \{0, 1, \cdots, \lceil \frac{\Gamma}{\Delta} \rceil\}$ **do**
  $\tau_i = i\Delta$
  $\mathcal{S}_i^G \leftarrow \emptyset$
  while $\mathcal{S}_i^G \in \mathcal{I}$ **do**
    $s = \underset{s \in \mathcal{X} \setminus \mathcal{S}_i^G, \mathcal{S}_i^G \cup \{s\} \in \mathcal{I}}{\operatorname{argmax}} H((\mathcal{S}_i^G \cup \{s\}), \tau_i) - H(\mathcal{S}_i^G, \tau_i)$
    $\mathcal{S}_i^G \leftarrow \mathcal{S}_i^G \cup \{s\}$
  end while
  $\mathcal{M} = \mathcal{M} \cup \{(\mathcal{S}_i^G, \tau_i)\}$
end for
$(\mathcal{S}^G, \tau^G) = \underset{(\mathcal{S}_i^G, \tau_i) \in \mathcal{M}}{\operatorname{argmax}} H(\mathcal{S}_i^G, \tau_i)$

Designing an Oracle: Note that an oracle $\mathcal{O}$ is used to calculate the value of $H(\mathcal{S}, \tau)$. We use a sampling based method to approximate this oracle. Specifically, we sample n_s realizations $\tilde{y}(s)$ from the distribution of y and approximate $H(\mathcal{S}, \tau)$ as $H(\mathcal{S}, \tau) \approx \tau - \frac{1}{n_s \alpha} \sum_{\tilde{y}} [(\tau - f(\mathcal{S}, \tilde{y}))_+]$. According to [26, Lemma 4.1], if the number of samples is $n_s = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta}), \delta, \epsilon \in (0,1)$, the CVaR approximation error is less than ϵ with the probability at least $1 - \delta$.

4.2 Performance Analysis of SGA

Theorem 1. *Let $\mathcal{S}^G$ and τ^G be the set and the scalar chosen by the SGA, and let the $\mathcal{S}^\star$ and $\tau^\star$ be the set and the scalar chosen by the OPT, we have*

$$H(\mathcal{S}^G, \tau^G) \geq \frac{1}{1+k_f}(H(\mathcal{S}^\star, \tau^\star) - \Delta) - \frac{k_f}{1+k_f}\Gamma(\frac{1}{\alpha} - 1), \tag{7}$$

where $k_f \in [0,1]$ is the curvature of the $H(\mathcal{S}, \tau)$ in set $\mathcal{S}$. Please see the detailed definition of the curvature in the appendix. The computational time is $O(\lceil \frac{\Gamma}{\Delta} \rceil |\mathcal{X}|^2 n_s)$ where Γ and Δ are the upper bound on τ and searching separation parameter, $|\mathcal{X}|$ is the cardinality of the ground set $\mathcal{X}$ and n_s is the number of the samplings used by the oracle.

SGA gives $1/(1+k_f)$ approximation of the optimal with two approximation errors. One approximation error comes from the searching separation Δ. We can make this error very small by setting Δ to be close to zero with the cost of

increasing the computational time. The second approximation error comes from the additive term,

$$H^{\text{add}} = \frac{k_f}{1+k_f}\Gamma(\frac{1}{\alpha}-1), \tag{8}$$

which depends on the curvature k_f and the risk level α. When the risk level α is very small, this error is very large which means SGA may not give a good performance guarantee of the optimal. However, if the function $H(\mathcal{S},\tau)$ is close to modular in $\mathcal{S}$ ($k_f \to 0$), this error is close to zero. Notably, when $k_f \to 0$ and $\Delta \to 0$, SGA gives a near-optimal solution ($H(\mathcal{S}^G,\tau^G) \to H(\mathcal{S}^\star,\tau^\star)$).

Next, we prove Theorem 1. We start with the proof of approximation ratio, then go to the analysis of the computational time. We first present the necessary lemmas for the proof of the approximation ratio.

Lemma 4. *Let $\mathcal{S}_i^\star$ be the optimal set for a specific τ_i that maximizes $H(\mathcal{S},\tau=\tau_i)$. By sequentially searching for $\tau \in [0,\Gamma]$ with a separation Δ, we have*

$$\max_{i\in\{0,1,\cdots,\lceil\frac{\Gamma}{\Delta}\rceil\}} H(\mathcal{S}_i^\star,\tau_i) \geq H(\mathcal{S}^\star,\tau^\star)-\Delta. \tag{9}$$

Next, we build the relationship between the set selected by the greedy approach, $\mathcal{S}_i^G$, and the optimal set $\mathcal{S}_i^\star$ for τ_i.

Lemma 5. *Let $\mathcal{S}_i^\star$ and $\mathcal{S}_i^G$ be the sets selected by the greedy algorithm and the optimal approach for a fixed τ_i that maximizes $H(\mathcal{S},\tau=\tau_i)$. We have*

$$H(\mathcal{S}_i^G,\tau_i) \geq \frac{1}{1+k_f}H(\mathcal{S}_i^\star,\tau_i) - \frac{k_f}{1+k_f}\Gamma(\frac{1}{\alpha}-1). \tag{10}$$

where k_f is the curvature of the function $H(\mathcal{S},\tau)$ in $\mathcal{S}$ with a matroid constraint $\mathcal{I}$. Γ is the upper bound of parameter τ.

We leverage Lemma 4 and Lemma 5 to prove the approximation ratio in Theorem 1.

5 Simulations

We perform numerical simulations to verify the performance of SGA in resilient mobility-on-demand and robust environment monitoring. Our code is available online.[2]

[2] https://github.com/raaslab/risk_averse_submodular_selection.git

5.1 Resilient Mobility-on-Demand under Arrival Time Uncertainty

We consider assigning $R = 6$ supply vehicles to $N = 4$ demand locations in a 2D environment. The positions of the demand locations and the supply vehicles are randomly generated within a square environment of 10 units side length. Denote the Euclidean distance between demand location $i \in \{1, ..., N\}$ and vehicle position $j \in \{1, ..., R\}$ as d_{ij}. Based on the distribution discussion of the arrival efficiency distribution in Section 3.2, we assume each arrival efficiency e_{ij} has a uniform distribution with its mean proportional to the reciprocal of the distance between demand i and vehicle j. Furthermore, the uncertainty is higher if the mean efficiency is higher. Note that, the algorithm can handle other, more complex, distributions for arrival times. We use a uniform distribution for ease of exposition. Specifically, denote the mean of e_{ij} as $\bar{e}_{ij}$ and set $\bar{e}_{ij} = 10/d_{ij}$. We model the arrival efficiency distribution to be a uniform distribution as follows:

$$e_{ij} = [\bar{e}_{ij} - \bar{e}_{ij}^{2.5}/\max\{\bar{e}_{ij}\}, \bar{e}_{ij} + \bar{e}_{ij}^{2.5}/\max\{\bar{e}_{ij}\}],$$

where $\max\{\bar{e}_{ij}\} = \max_{i,j} e_{ij}, i \in \{1, ..., N\}, j \in \{1, ..., R\}$.

From the assignment utility function (Eq. 5), for any realization of y, say $\tilde{y}$,

$$f(\mathcal{S}, \tilde{y}) := \sum_{i \in N} \max_{j \in \mathcal{S}_i} \tilde{e}_{ij}$$

where $\tilde{e}_{ij}$ indicates one realization of e_{ij}. If all vehicle-demand pairs are independent from each other, y models a multi-independent uniform distribution. We sample n_s times from underlying multi-independent uniform distribution of y and approximate the auxiliary function $H(\mathcal{S}, \tau)$ as

$$H(\mathcal{S}, \tau) \approx \tau - \frac{1}{n_s \alpha} \sum_{\tilde{y}} [(\tau - \sum_{i \in N} \max_{j \in \mathcal{S}_i} \tilde{e}_{ij})_+].$$

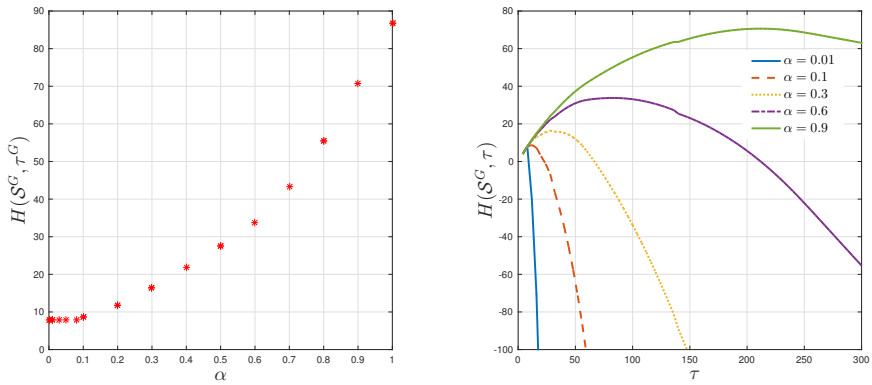
We set the upper bound of the parameter τ as $\Gamma = N \max(\tilde{e}_{ij}), i = \{1, ...N\}, j = \{1, ..., R\}$, to make sure $\Gamma - f(\mathcal{S}, y) \geq 0$. We set the searching separation for τ as $\Delta = 1$.

After receiving the pair $(\mathcal{S}^G, \tau^G)$ from SGA, we plot the value of $H(\mathcal{S}^G, \tau^G)$ and $H(\mathcal{S}^G, \tau)$ with respect to different risk levels α in Fig. 5. Fig. 5-(a) shows that $H(\mathcal{S}^G, \tau^G)$ increases when α increases. This suggests that SGA correctly maximizes $H(\mathcal{S}, \tau)$. Fig. 5-(b) shows that $H(\mathcal{S}^G, \tau)$ is concave or piecewise concave, which is consistent with the property of $H(\mathcal{S}, \tau)$.

We plot the distribution of assignment utility,

$$f(\mathcal{S}^G, y) = \sum_{i \in N} \max_{j \in \mathcal{S}_i^G} e_{ij}$$

in Fig. 6 by sampling $n_s = 1000$ times from the underlying distribution of y. $\mathcal{S}_i^G$ is a set of vehicles assigned to demand i by SGA. $\mathcal{S}^G = \cup_{i=1}^N \mathcal{S}_i^G$. When the risk level α is small, vehicle-demand pairs with low efficiencies (equivalently, low uncertainties) are selected. This is because small risk level indicates the



(a) The value of $H(\mathcal{S}^G, \tau^G)$ with respect to several risk levels α.

(b) Function $H(\mathcal{S}^G, \tau)$ with respect to several risk levels α.

Fig. 5. The value of $H(\mathcal{S}, \tau)$ by SGA with respect to different risk confidence levels.

assignment is conservative and only willing to take a little risk. Thus, lower efficiency with lower uncertainty is assigned to avoid the risk induced by the uncertainty. In contrast, when α is large, the assignment is allowed to take more risk to gain more assignment utility. Vehicle-demand pairs with high efficiencies (equivalently, high uncertainties) are selected in such a case. Note that, when the risk level is close to zero, SGA may not give a correct solution because of a large approximation error (Fig. 7). However, this error decreases quickly to zero when the risk level increases.

We also compare SGA with CVaR measure with the greedy algorithm with the expectation, i.e., risk-neutral measure [16] in Fig. 8. Note that risk-neutral measure is a special case of $\mathrm{CVaR}_\alpha(\mathcal{S})$ measure when $\alpha = 1$. We give an illustrative example of the assignment by SGA for two extreme risk levels, $\alpha = 0.1$ and $\alpha = 1$. When α is small ($\alpha = 0.1$), the assignment is conservative and thus further vehicles (with lower efficiency and lower uncertainty) are assigned to each demand (Fig. 8-(a)). In contrast, when $\alpha = 1$, nearby vehicles (with higher efficiency and higher uncertainty) are selected for the demands (Fig. 8-(b)). Even though the mean value of the assignment utility distribution is larger at $\alpha = 1$ than $\alpha = 0.1$, it is exposed to the risk of receiving lower utility since the mean-std bar at $\alpha = 1$ has smaller left endpoint than the mean-std bar at $\alpha = 0.1$ (Fig. 8-(c)).

5.2 Robust Environment Monitoring

We consider selecting $M = 4$ locations from $N = 8$ candidate locations to place sensors in the environment (Fig. 4). Denote the area of the free space as v^{free}. The positions of N candidate locations are randomly generated within the free space v^{free}. We calculate the visibility region for each sensor v_i by using the

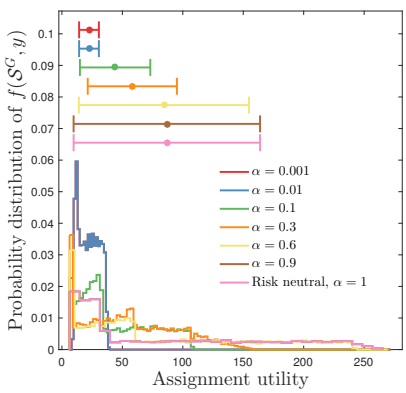


Fig. 6. Distribution of the selection utility $f(\mathcal{S}^G, y)$ by SGA.

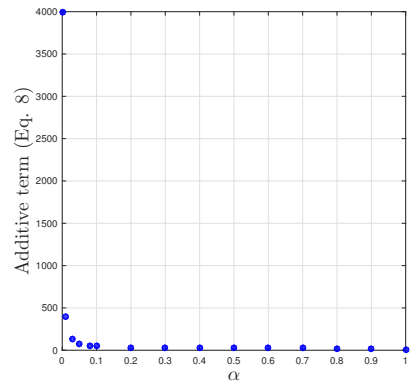


Fig. 7. Additive term in the approximation ratio with respect to risk level α.

VisiLibity library [30]. Based on the sensor model discussed in Section 3.2, we set the probability of success for each sensor i as

$$p_i = 1 - v_i / v^{\text{free}},$$

and model the working of each sensor as a Bernoulli distribution with p_i probability of success and $1 - p_i$ probability of failure. Thus the random polygon monitored by each sensor A_i, follows the distribution

$$\begin{cases} \Pr[A_i = v_i] = p_i, \\ \Pr[A_i = 0] = 1 - p_i. \end{cases} \tag{11}$$

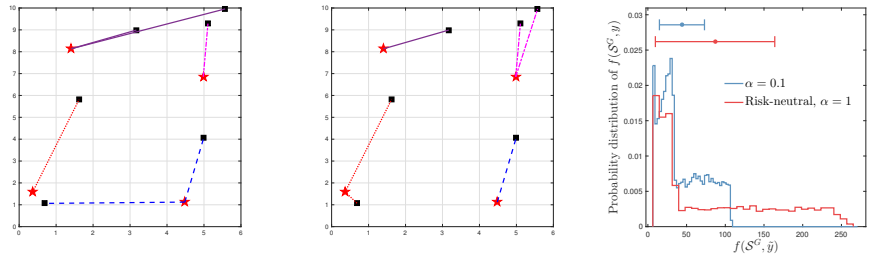
From the assignment utility function (Eq. 6), for any realization of y, say $\tilde{y}$,

$$f(\mathcal{S}, y) = \text{area}(\bigcup_{i=1:M} \tilde{A}_i),$$

where $\tilde{A}_i$ indicates one realization of A_i by sampling y. If all sensors are independent of each other, we can model the working of a set of sensors as a multi-independent Bernoulli distribution. We sample $n_s = 1000$ times from the underlying multi-independent Bernoulli distribution of y and approximate the auxiliary function $H(\mathcal{S}, \tau)$ as

$$H(\mathcal{S}, \tau) \approx \tau - \frac{1}{n_s \alpha} \sum_{\tilde{y}} [(\tau - \bigcup_{i=1:M} \tilde{A}_i)_+],$$

where $\tilde{A}_i$ is one realization of A_i by sampling y. We set the upper bound for the parameter τ as the area of all the free space v^{free} and set the searching separation for τ as $\Delta = 1$.

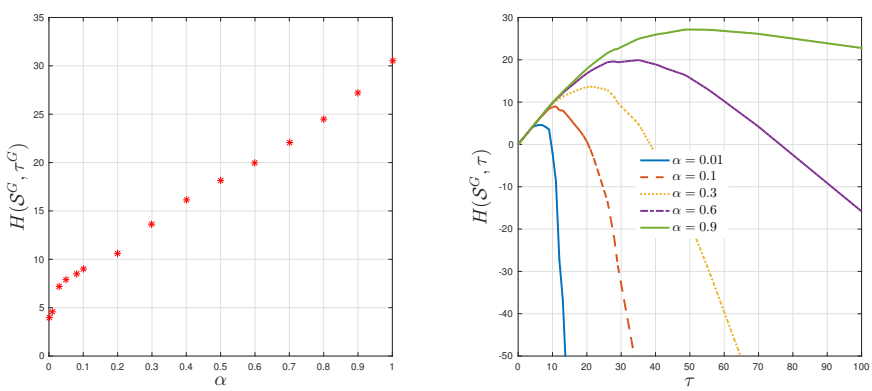


(a) Assignment when $\alpha = 0.1$.

(b) Assignment when $\alpha = 1$ (Risk-neutral).

(c) Assignment utility distributions at $\alpha = 0.1$ and $\alpha = 1$.

Fig. 8. Assignments and utility distributions by SGA with two extreme risk level values. The red solid star represents the demand location. The black solid square represents the vehicle position. The line between the vehicle and the demand represents an assignment.



(a) The value of $H(\mathcal{S}^G, \tau^G)$ with respect to different risk levels α.

(b) Function $H(\mathcal{S}^G, \tau)$ with respect to several risk levels α.

Fig. 9. The value of $H(\mathcal{S}, \tau)$ by SGA with respect to different risk confidence levels.

We use SGA to find the pair $(\mathcal{S}^G, \tau^G)$ with respect to several risk levels α. We plot the value of $H(\mathcal{S}^G, \tau^G)$ for several risk levels in Fig. 9-(a). A larger risk level gives a larger $H(\mathcal{S}^G, \tau^G)$, which means the pair $(\mathcal{S}^G, \tau^G)$ found by SGA correctly maximizes $H(\mathcal{S}, \tau)$ with respect to the risk level α. Moreover, we plot functions $H(\mathcal{S}^G, \tau)$ for several risk levels α in Fig. 9-(b). Note that $\mathcal{S}^G$ is computed by SGA at each τ. For each α, $H(\mathcal{S}^G, \tau)$ shows the concavity or piecewise concavity of function $H(\mathcal{S}, \tau)$.

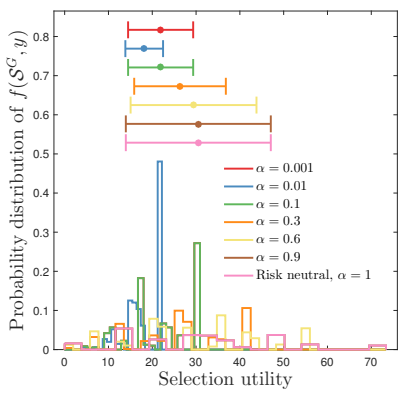


Fig. 10. Distribution of the selection utility $f(\mathcal{S}^G, y)$ by SGA.

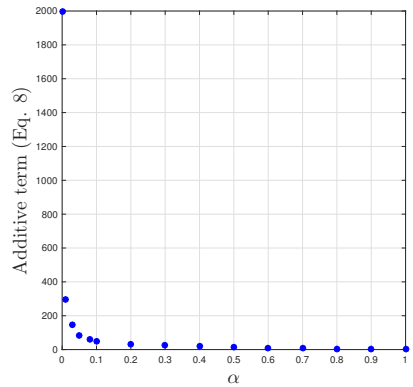


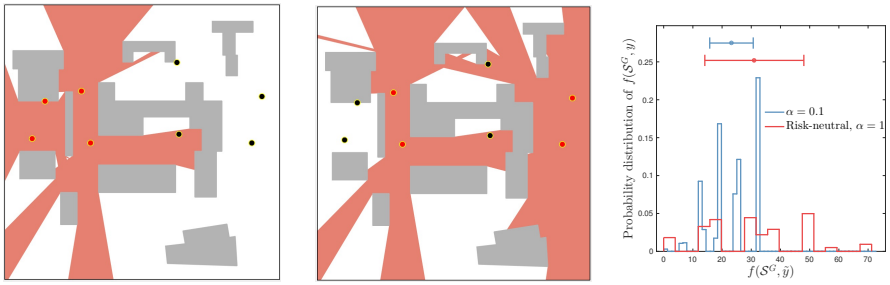
Fig. 11. Additive term in the approximation ratio with respect to risk level α.

Based on the $\mathcal{S}^G$ calculated by SGA, we sample $n_s = 1000$ times from the underlying distribution of y and plot the distribution of the selection utility,

$$f(\mathcal{S}^G, y) = \bigcup_{i=1:M} A_i, i \in \mathcal{S}^G$$

in Fig. 10. Note that, when the risk level α is small, the sensors with smaller visibility region and a higher probability of success should be selected. Lower risk level suggests a conservative selection. Sensors with a higher probability of success are selected to avoid the risk induced by sensor failure. In contrast, when α is large, the selection would like to take more risk to gain more monitoring utility. The sensors with larger visibility region and a lower probability of success should be selected. Fig. 10 demonstrates this behavior except between $\alpha = 0.001$ to $\alpha = 0.01$. This is because when α is very small, the approximation error (Eq. 8) is very large as shown in Fig. 11, and thus SGA may not give a good solution.

We also compare SGA by using CVaR measure with the greedy algorithm by using the expectation, i.e., risk-neutral measure (mentioned in [2, Section 6.1]) in Fig. 12. In fact, the risk-neutral measure is equivalent to case of $\mathrm{CVaR}_\alpha(\mathcal{S})$ when $\alpha = 1$. We give an illustrative example of the sensor selection by SGA for two extreme risk levels, $\alpha = 0.1$ and $\alpha = 1$. When risk level α is small ($\alpha = 0.1$), the selection is conservative and thus the sensors with small visibility region are selected (Fig. 12 -(a)). In contrast, when $\alpha = 1$, the risk is neutral and the selection is more adventurous, and thus sensors with large visibility region are selected (Fig. 12 -(b)). The mean-std bars of the selection utility distributions in Fig. 12 -(c) show that the selection utility at the expectation ($\alpha = 1$) has larger mean value than the selection at $\alpha = 0.1$. However, the selection at $\alpha = 1$ has the risk of gaining lower utility since the left endpoint of mean-std bar at $\alpha = 1$ is smaller than the left endpoint of mean-std bar at $\alpha = 0.1$.



(a) Selection when $\alpha = 0.1$.

(b) Selection when $\alpha = 1$ (Risk-neutral).

(c) Selection utility distributions at $\alpha = 0.1$ and $\alpha = 1$.

Fig. 12. Sensor selection and utility distributions by SGA with two extreme risk level values. The red solid circle represents the sensor selected by SGA.

6 Conclusion and Discussion

We studied a risk-averse discrete submodular maximization problem. We provide the first positive results for discrete CVaR submodular maximization for selecting a set under matroidal constraints. In particular, we proposed the Sequential Greedy Algorithm and analyzed its approximation ratio and the running time. We demonstrated the two practical use-cases of the CVaR submodular maximization problem.

Notably, our Sequential Greedy Algorithm works for any matroid constraint. In particular, the multiplicative approximation ratio can be improved to $1/k_f(1-e^{-k_f})$ if we know that the constraint is a uniform matroid [31, Theorem 5.4].

The additive term in our analysis depends on α. This term can be large when the risk level α is very small. Our ongoing work is to remove this dependence on α, perhaps by designing another algorithm specifically for low risk levels. We note that if we use an optimal algorithm instead of the greedy algorithm as a subroutine, then the additive term disappears from the approximation guarantee. The algorithm also requires knowing Γ. We showed how to find Γ (or an upper bound for it) for the two case studies considered in this paper. Devising a general strategy for finding Γ is part of our ongoing work.

Our second line of ongoing work focuses on applying the risk-averse strategy to multi-vehicle routing, patrolling, and informative path planning in dangerous environments [32] and mobility on demand with real-world data sets (2014 NYC Taxicab Factbook).[3]

7 Acknowledgements

This work was supported by NSF award IIS-1637915 and ONR Award N00014-18-1-2829.

[3] http://www.nyc.gov/html/tlc/downloads/pdf/2014_taxicab_fact_book.pdf

8 Appendix

Background on set functions:

8.1 Monotonicity, Submodularity, Matroid and Curvature

We begin by reviewing useful properties of a set function $f(\mathcal{S})$ defined for a finite ground set $\mathcal{X}$ and matroid constraints.

Monotonicity [9]: A set function $f : 2^{\mathcal{X}} \mapsto \mathbb{R}$ is monotone (non-decreasing) if and only if for any sets $\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{X}$, we have $f(\mathcal{S}) \leq f(\mathcal{S}')$.

Normalized Function [10]: A set function $f : 2^{\mathcal{X}} \mapsto \mathbb{R}$ is called normalized if and only if $f(\emptyset) = 0$.

Submodularity [9, Proposition 2.1]: A set function $f : 2^{\mathcal{X}} \mapsto \mathbb{R}$ is submodular if and only if for any sets $\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{X}$, and any element $s \in \mathcal{X}$ and $s \notin \mathcal{S}'$, we have: $f(\mathcal{S} \cup \{s\}) - f(\mathcal{S}) \geq f(\mathcal{S}' \cup \{s\}) - f(\mathcal{S}')$. Therefore the marginal gain $f(\mathcal{S} \cup \{s\}) - f(\mathcal{S})$ is non-increasing.

Matroid [33, Section 39.1]— Denote a non-empty collection of subsets of $\mathcal{X}$ as $\mathcal{I}$. The pair $(\mathcal{X}, \mathcal{I})$ is called a matroid if and only if the following conditions are satisfied:

- for any set $\mathcal{S} \subseteq \mathcal{X}$ it must hold that $\mathcal{S} \in \mathcal{I}$, and for any set $\mathcal{P} \subseteq \mathcal{S}$ it must hold that $\mathcal{P} \in \mathcal{I}$.
- for any sets $\mathcal{S}, \mathcal{P} \subseteq \mathcal{X}$ and $|\mathcal{P}| \leq |\mathcal{S}|$, it must hold that there exists an element $s \in \mathcal{S} \backslash \mathcal{P}$ such that $\mathcal{P} \cup \{s\} \in \mathcal{I}$.

We will use two specific forms of matroids that are reviewed next.

Uniform Matroid: A *uniform matroid* is a matroid $(\mathcal{X}, \mathcal{I})$ such that for a positive integer κ, $\{\mathcal{S} : \mathcal{S} \subseteq \mathcal{X}, |\mathcal{S}| \leq \kappa\}$. Thus, the uniform matroid only constrains the cardinality of the feasible sets in $\mathcal{I}$.

Partition Matroid: A *partition matroid* is a matroid $(\mathcal{X}, \mathcal{I})$ such that for a positive integer n, disjoint sets $\mathcal{X}_1, ..., \mathcal{X}_n$ and positive integers $\kappa_1, ..., \kappa_n$, $\mathcal{X} \equiv \mathcal{X}_1 \cup \cdots \mathcal{X}_n$ and $\mathcal{I} = \{\mathcal{S} : \mathcal{S} \subseteq \mathcal{X}, |\mathcal{S} \cap \mathcal{X}_i| \leq \kappa_i \text{ for all } i = 1, ..., n\}$.

Curvature [31]: consider a matroid $\mathcal{I}$ for $\mathcal{X}$, and a non-decreasing submodular set function $f : 2^{\mathcal{X}} \mapsto \mathbb{R}$ such that (without loss of generality) for any element $s \in \mathcal{X}$, $f(s) \neq 0$. The curvature measures how far f is from submodularity or linearity. Define *curvature* of f over the matroid I as:

$$k_f \triangleq 1 - \min_{s \in \mathcal{S}, \mathcal{S} \in \mathcal{I}} \frac{f(\mathcal{S}) - f(\mathcal{S} \setminus \{s\})}{f(s)}. \tag{12}$$

Note that the definition of curvature k_f (Equation 12) implies that $0 \leq k_f \leq 1$. Specifically, if $k_f = 0$, it means for all the feasible sets $\mathcal{S} \in \mathcal{X}$, $f(\mathcal{S}) = \sum_{s \in \mathcal{S}} f(s)$.

In this case, f is a modular function. In contrast, if $k_f = 1$, then there exist a feasible $\mathcal{S} \in \mathcal{I}$ and an element $s \in \mathcal{X}$ such that $f(\mathcal{S}) = f(\mathcal{S} \setminus \{s\})$. In this case, the element s is redundant for the contribution of the value of f given the set $\mathcal{S} \setminus \{s\}$.

8.2 Greedy Approximation Algorithm

In order to maximize a set function f, the greedy algorithm selects each element s of $\mathcal{S}$ based on the maximum marginal gain at each round.

We consider maximizing a normalized monotone submodular set function f. For any matroid, the greedy algorithm gives a 1/2 approximation [10]. In particular, the greedy algorithm can give a $(1 - 1/e)$–approximation of the optimal solution under the uniform matroid [9]. If we know the curvature of the set function f, we have a $1/(1 + k_f)$ approximation for any matroid constraint [31, Theorem 2.3]. That is,

$$\frac{f(\mathcal{S}^G)}{f^\star} \geq \frac{1}{1 + k_f}.$$

where $\mathcal{S}^\mathcal{G} \in \mathcal{I}$ is the set selected by the greedy algorithm, $\mathcal{I}$ is the uniform matroid and $f^\star$ is the function value with optimal solution. Note that, if $k_f = 0$, which means f is modular, then the greedy algorithm reaches the optimal. If $k_f = 1$, then we have the 1/2–approximation.

Proof of Lemma 1:

Proof. $H(\mathcal{S}, \tau) = \tau - \frac{1}{\alpha}\mathbb{E}[\max(\tau - f(\mathcal{S}, y), 0)]$. Since $f(\mathcal{S}, y)$ is monotone increasing and submodular in $\mathcal{S}$, $\max\{\tau - f(\mathcal{S}, y), 0\}$ is monotone decreasing and supermodular in $\mathcal{S}$, and its expectation is also monotone decreasing and supermodular in $\mathcal{S}$. Then $H(\mathcal{S}, \tau)$ is monotone increasing and submodular in $\mathcal{S}$.

$H(\emptyset, \tau) = \tau(1 - \frac{1}{\alpha})$ given $f(\mathcal{S}, y)$ is normalized ($f(\emptyset, y) = 0$). Thus, $H(\mathcal{S}, \tau)$ is not necessarily normalized since τ is not necessarily zero. See a similar proof in [18, 25].

□

Proof of Lemma 2:

Proof. $H(\mathcal{S}, \tau) = \tau - \frac{1}{\alpha}\mathbb{E}[\max(\tau - f(\mathcal{S}, y), 0)]$. Since $\max(\tau - f(\mathcal{S}, y), 0)$ is convex in τ, its expectation is also convex in τ. Then $-\frac{1}{\alpha}\mathbb{E}[\max(\tau - f(\mathcal{S}, y), 0)]$ is concave in τ and $H(\mathcal{S}, \tau)$ is concave in τ.

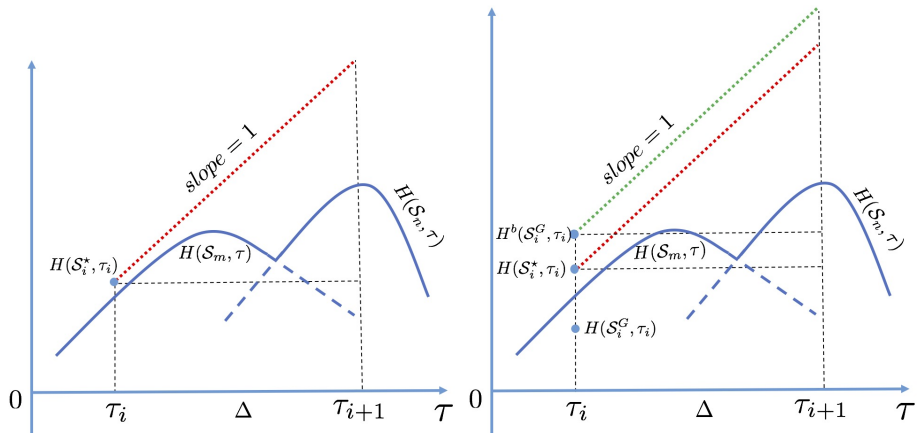
□

Proof of Lemma 3:

Proof. By using the result in [18, Lemma 1 and Proof of Theorem 1], we know that $H(\mathcal{S}, \tau)$ is concave and continuously differentiable with derivative given by

$$\frac{\partial H(\mathcal{S}, \tau)}{\partial \tau} = 1 - \frac{1}{\alpha}(1 - \Phi(f(\mathcal{S}, y)))$$

where $\Phi(f(\mathcal{S}, y))$ is the cumulative distribution function of $f(\mathcal{S}, y)$. Thus, $0 \leq \Phi(f(\mathcal{S}, y))) \leq 1$, which proves the lemma.



(a) $H(\mathcal{S}, \tau)$ is under the red dotted line. (b) $H(\mathcal{S}, \tau)$ is under the red dotted line and green dotted line.

Fig. 13. Illustration of $H(\mathcal{S}, \tau)$ within $\tau \in [\tau_i, \tau_{i+1}]$.

□

Proof of Lemma 4:

Proof. Denote $H_i^\star = \max\ H(\mathcal{S}, \tau)$ with $\tau \in [i\Delta, (i+1)\Delta), \mathcal{S} \in \mathcal{I}$. From Lemmas 2 and 3, we know $H(\mathcal{S}, \tau)$ is concave in τ and $\frac{\partial H(\mathcal{S},\tau)}{\partial \tau} < 1$. The properties of concavity and bound on the gradient give

$$H_i^* - H(\mathcal{S}_i^\star, \tau_i) \leq \Delta.$$

We illustrate this claim by using Figure 13-(a). Since $\mathcal{S}_i^\star$ is the optimal set at τ_i for maximizing $H(\mathcal{S}, \tau)$, the value of $H(\mathcal{S}, \tau)$ with any other set $\mathcal{S} \in \mathcal{I}$ at τ_i is at most $H(\mathcal{S}_i^\star, \tau_i)$. That is, $H(\mathcal{S}, \tau_i) \leq H(\mathcal{S}_i^\star, \tau_i)$. Since $H(\mathcal{S}, \tau)$ is a concave function of τ for any specific $\mathcal{S}$, $H(\mathcal{S}, \tau)$ can be a single concave function, i.e., $H(\mathcal{S}_m, \tau)$ or $H(\mathcal{S}_n, \tau)$ or a piecewise concave function by a combination of several concave functions, i.e., a combination of $H(\mathcal{S}_m, \tau)$ and $H(\mathcal{S}_n, \tau)$ during $\tau \in [\tau_i, \tau_{i+1}]$ (Figure 13-(a)). In either case, $H(\mathcal{S}, \tau)$ is below the line starting at $H(\mathcal{S}_i^\star, \tau_i)$ with *slope* $= 1$ during $\tau \in [\tau_i, \tau_{i+1}]$ (the red dotted line in Figure 13-(a)). Since $H(\mathcal{S}, \tau_i) \leq H(\mathcal{S}_i^\star, \tau_i)$ and $H(\mathcal{S}, \tau)$ has a bounded gradient $\frac{\partial H(\mathcal{S},\tau)}{\partial \tau} \leq 1$. Thus, $H_i^\star - H(\mathcal{S}_i^\star, \tau_i) \leq \frac{\partial H(\mathcal{S},\tau)}{\partial \tau}\Delta = \Delta,\ \forall i \in \{0, 1, \cdots, \lceil \frac{\Gamma}{\Delta} \rceil\}$.

Then we have $H_i^\star - \max_i H(\mathcal{S}_i^\star, \tau_i) \leq \Delta,\ \forall i \in \{0, 1, \cdots, \lceil \frac{\Gamma}{\Delta} \rceil\}$. Note that $H_i^\star$ is the maximum value of $H(\mathcal{S}, \tau)$ at each interval $\tau \in [i\Delta, (i+1)\Delta)$. The maximum value of $H(\mathcal{S}, \tau)$, $H(\mathcal{S}^\star, \tau^\star)$ is equal to one of $H_i^\star, i \in \{0, 1, \cdots, \lceil \frac{\Gamma}{\Delta} \rceil\}$. Thus, we reach the claim in Lemma 4.

□

Proof of Lemma 5:

Proof. We use a the previous result [31, Theorem 2.3] for the proof of this claim. We know that for any given τ, $H(\mathcal{S}, \tau)$ is a non-normalized monotone submdoular

function in $\mathcal{S}$ (Lemma 1). For maximizing normalized monotone submodular set functions, the greedy approach can give a $1+1/k_f$ approximation of the optimal performance with any matroid constraint [31, Theorem 2.3]. After normalizing $H(\mathcal{S},\tau)$ by $H(\mathcal{S},\tau)-H(\emptyset,\tau)$, we have

$$\frac{H(\mathcal{S}_i^G,\tau_i)-H(\emptyset,\tau_i)}{H(\mathcal{S}_i^\star,\tau_i)-H(\emptyset,\tau_i)} \geq \frac{1}{1+k_f}, \tag{13}$$

with any matroid constraint. Given $0 \leq k_f \leq 1$ and $H(\emptyset,\tau) = -\tau(\frac{1}{\alpha}-1)$, we transform Equation 13 into,

$$\begin{aligned} H(\mathcal{S}_i^G,\tau_i) &\geq \frac{1}{1+k_f}H(\mathcal{S}_i^\star,\tau_i) - \frac{k_f}{1+k_f}\tau_i(\frac{1}{\alpha}-1) \\ &\geq \frac{1}{1+k_f}H(\mathcal{S}_i^\star,\tau_i) - \frac{k_f}{1+k_f}\Gamma(\frac{1}{\alpha}-1) \end{aligned} \tag{14}$$

where Equation 14 holds since Γ is the upper bound of τ. Thus, we prove the Lemma 5.

□

Proof of Theorem 1:

Proof. From Equation 10 in Lemma 5, we have $H(\mathcal{S}_i^\star,\tau_i)$ is bounded by

$$H(\mathcal{S}_i^\star,\tau_i) \leq (1+k_f)H(\mathcal{S}_i^G,\tau_i) + k_f\Gamma(\frac{1}{\alpha}-1). \tag{15}$$

Denote this upper bound as

$$H^b(\mathcal{S}_i^G,\tau_i) := (1+k_f)H(\mathcal{S}_i^G,\tau_i) + k_f\Gamma(\frac{1}{\alpha}-1).$$

We know $H(\mathcal{S},\tau)$ is below the line starting at $H(\mathcal{S}_i^\star,\tau_i)$ with *slope* $=1$ during $\tau \in [\tau_i,\tau_{i+1}]$ (the red dotted line in Figure 13-(a)/(b)) (Lemma 4). $H(\mathcal{S},\tau)$ must be also below the line starting at $H^b(\mathcal{S}_i^G,\tau_i)$ with *slope* $=1$ during $\tau \in [\tau_i,\tau_{i+1}]$ (the green dotted line in Figure 13-(b)). Similar to the proof in Lemma 4, we have $H_i^\star - H^b(\mathcal{S}_i^G,\tau_i) \leq \Delta$ and

$$\max_{i\in\{0,1,\cdots,\lceil\frac{\Gamma}{\Delta}\rceil\}} H^b(\mathcal{S}_i^G,\tau_i) \geq H(\mathcal{S}^\star,\tau^\star) - \Delta. \tag{16}$$

SGA selects the pair $(\mathcal{S}^G,\tau^G)$ as the pair $(\mathcal{S}_i^G,\tau_i)$ with $\max_i H(\mathcal{S}_i^G,\tau_i)$. Then by Inequalities 15 and 16, we have

$$(1+k_f)H(\mathcal{S}^G,\tau^G) + k_f\Gamma(\frac{1}{\alpha}-1) \geq H(\mathcal{S}^\star,\tau^\star) - \Delta. \tag{17}$$

By rearranging the terms, we get the approximation ratio in Theorem 1.

Next, we give the proof of the computational time of SGA in Theorem 1. We verify the computational time of SGA by following the stages of the pseudo code in SGA. First, from line 2 to 10, we use a "for" loop for searching τ which takes

$\lceil \frac{\Gamma}{\Delta} \rceil$ evaluations. Second, within the "For" loop, we use the greedy algorithm to solve the subproblem (lines 4–8). In order to select a subset $\mathcal{S}$ with size $|\mathcal{S}|$ from a ground set $\mathcal{X}$ with size $|\mathcal{X}|$, the greedy algorithm takes $|\mathcal{S}|$ rounds (line 5), and calculates the marginal gain of the remaining elements in $\mathcal{X}$ at each round (line 6). Thus, the greedy algorithm takes $\sum_{i=1}^{|\mathcal{S}|} |\mathcal{X}| - i$ evaluations. Thus, the greedy algorithm takes $|\mathcal{S}|(\sum_{i=1}^{|\mathcal{S}|} |\mathcal{X}| - i)$ evaluations. Third, by calculating the marginal gain for each element, the oracle $\mathcal{O}$ samples n_s times for computing $H(\mathcal{S})$. Thus, overall, the "for" loop containing the greedy algorithm with the oracle sampling takes $\lceil \frac{\Gamma}{\Delta} \rceil |\mathcal{S}|(\sum_{i=1}^{|\mathcal{S}|} |\mathcal{X}| - i) n_s$ evaluations. Last, finding the best pair from storage set $\mathcal{M}$ (line 11 of Alg. 1) takes $O(\lceil \frac{\Gamma}{\Delta} \rceil)$ time. Therefore, the computational complexity for SGA is,

$$\lceil \frac{\Gamma}{\Delta} \rceil |\mathcal{S}|(\sum_{i=1}^{|\mathcal{S}|} |\mathcal{X}| - i) n_s + O(\lceil \frac{\Gamma}{\Delta} \rceil) = O(\lceil \frac{\Gamma}{\Delta} \rceil |\mathcal{X}|^2 n_s),$$

given $|\mathcal{S}| \leq |\mathcal{X}|$.

□

References

1. Joseph O'Rourke. *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford, 1987.
2. Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.
3. Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.
4. Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.
5. John R Spletzer and Camillo J Taylor. Dynamic sensor planning and control for optimally tracking targets. *The International Journal of Robotics Research*, 22(1):7–20, 2003.
6. Onur Tekdas and Volkan Isler. Sensor placement for triangulation-based localization. *IEEE transactions on Automation Science and Engineering*, 7(3):681–685, 2010.
7. Pratap Tokekar, Volkan Isler, and Antonio Franchi. Multi-target visual tracking with aerial robots. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3067–3072. IEEE, 2014.
8. Amarjeet Singh, Andreas Krause, Carlos Guestrin, and William J Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, 34:707–755, 2009.
9. George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions–i. *Mathematical programming*, 14(1):265–294, 1978.

10. Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. An analysis of approximations for maximizing submodular set functionsii. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.
11. Huanyu Ding and David Castanón. Multi-agent discrete search with limited visibility. In *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, pages 108–113. IEEE, 2017.
12. Ryan K Williams, Andrea Gasparri, and Giovanni Ulivi. Decentralized matroid optimization for topology constraints in multi-robot allocation problems. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 293–300. IEEE, 2017.
13. Anthony D Wood and John A Stankovic. Denial of service in sensor networks. *computer*, 35(10):54–62, 2002.
14. Philip Dames, Pratap Tokekar, and Vijay Kumar. Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots. *The International Journal of Robotics Research*, 36(13-14):1540–1553, 2017.
15. David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.
16. Amanda Prorok. Supermodular optimization for redundant robot assignment under travel-time uncertainty. *arXiv preprint arXiv:1804.04986*, 2018.
17. Georg Ch Pflug. Some remarks on the value-at-risk and the conditional value-at-risk. In *Probabilistic constrained optimization*, pages 272–281. Springer, 2000.
18. R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.
19. Fan Yang and Nilanjan Chakraborty. Algorithm for optimal chance constrained linear assignment. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 801–808. IEEE, 2017.
20. Fan Yang and Nilanjan Chakraborty. Algorithm for optimal chance constrained knapsack with applications to multi-robot teaming. In *Robotics and Automation (ICRA), 2018 IEEE International Conference on*, to appear.
21. Harry Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.
22. JP Morgan. Riskmetrics technical document. 1996.
23. Anirudha Majumdar and Marco Pavone. How should a robot assess risk? towards an axiomatic theory of risk in robotics. *arXiv preprint arXiv:1710.11040*, 2017.
24. R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.
25. Takanori Maehara. Risk averse submodular utility maximization. *Operations Research Letters*, 43(5):526–529, 2015.
26. Naoto Ohsaka and Yuichi Yoshida. Portfolio optimization for influence spread. In *Proceedings of the 26th International Conference on World Wide Web*, pages 977–985. International World Wide Web Conferences Steering Committee, 2017.
27. Bryan Wilder. Risk-sensitive submodular optimization. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.
28. Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical finance*, 9(3):203–228, 1999.
29. Evan Ackerman and Eliza Strickland. Medical delivery drones take flight in east africa. *IEEE Spectrum*, 55(1):34–35, 2018.
30. K. J. Obermeyer and Contributors. VisiLibity: A c++ library for visibility computations in planar polygonal environments. `http://www.VisiLibity.org`, 2008. R-1.

31. Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete applied mathematics*, 7(3):251–274, 1984.
32. Stefan Jorgensen, Robert H Chen, Mark B Milam, and Marco Pavone. The team surviving orienteers problem: routing teams of robots in uncertain environments with survival constraints. *Autonomous Robots*, 42(4):927–952, 2018.
33. Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.