# Constrained Robust Submodular Sensor Selection with Applications to Multistatic Sonar Arrays

Thomas Powers*, Jeff Bilmes*, David W. Krout† and Les Atlas*
*Department of Electrical and Engineering
University of Washington, Seattle, USA
Emails: {tcpowers, bilmes, atlas} @uw.edu
†Applied Physics Laboratory
University of Washington, Seattle, USA
Email: dkrout@apl.washington.edu

***Abstract*—We develop a framework to select a subset of sensors from a field in which the sensors have an ingrained independence structure. Given an arbitrary independence pattern, we construct a graph that denotes pairwise independence between sensors, which means those sensors may operate simultaneously. The set of all fully-connected subgraphs (cliques) of this independence graph forms the independent sets of a matroid over which we maximize the minimum of a set of submodular objective functions. We propose a novel algorithm called MatSat that exploits submodularity and, as a result, returns a near-optimal solution with approximation guarantees that are within a small factor of the average-case scenario. We apply this framework to ping sequence optimization for active multistatic sonar arrays by maximizing sensor coverage and derive lower bounds for minimum probability of detection for a fractional number of targets. In these ping sequence optimization simulations, MatSat exceeds the fractional lower bounds and reaches near-optimal performance.**

## I. Introduction

Subset selection problems are important for many applications in areas such as wireless communications, environmental monitoring, and speech processing [1]–[4]. This paper addresses general sensor selection, with a specific focus on sensor networks with interfering sensors, and we demonstrate improved performance on active multistatic sonar arrays [5]. As is the case with most optimization problems, it would be advantageous for the problem to be convex. However, framing sensor selection problems as convex has two main problems. First, sensor selection is inherently a discrete optimization problem since selecting a sensor is an absolute choice. One cannot partially choose a sensor, and convex optimization utilizes continuous variables. Second, convex optimization is unable to handle dependence constraints between variables in a way that is appropriate for our task. Specifically, there is no known way to enforce dependent values between variables in a convex framework such that if a pair of sensors interfere, the two sensors will not be present in the solution together.

Submodular function optimization (SFO) provides a more intuitive framework for handling these two problems, since it inherently uses set functions and can be constrained to optimize over the independence sets of matroids. Matroids are a structure that generalize the notion of linear independence from vector spaces to set systems, and can be used to form constraints in SFO. They will be addressed further in Section II.

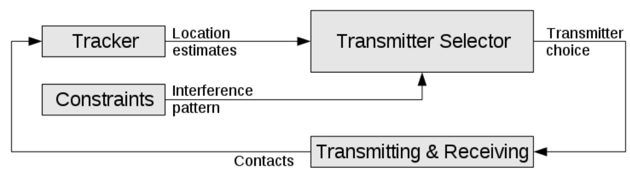



Fig. 1: General system diagram for sensor selection.

The main focus of this paper is a novel optimization algorithm, MatSat, that is a modification of SATURATE [3]. MatSat maximizes the minimum of a set of submodular functions subject to matroid independence constraints rather than cardinality constraints, and as a result offers a new strategy for robust submodular optimization over constraint sets (such as matroids) that are not easily expanded as in the case of a cardinality constraints. We also investigate modeling sensor networks as graphs and then using the graph structure to form matroid independence sets for use as constraints in submodular function optimization. SFO can handle constraints that make problems nonconvex or non-polynomial (NP) hard and find polynomial time solutions that are provably near-optimal with performance guarantees [6]. Section II describes how to model sensor network interference patterns as an independence graph can be folded directly into SFO as a series of matroids [7], [8]. This new approach is applied to scheduling active multistatic deep-water sonar arrays, or ping sequence optimization (PSO), in which we repeatedly optimize to select a subset of buoys that maximize a probabilistic coverage metric. We detail this application in Section III. In order to demonstrate the specific advantages of MatSat, we compare its performance to SFO-Greedy and exhaustive search approaches in Section V.

## II. Submodularity and Independence Graphs

The discrete nature of sensor selection makes optimization difficult. Typically, one represents the sensor nodes in an indicator vector with a selected sensor node as ones and unselected sensor nodes as zeros. These independence constraints make optimization problems nonconvex. One of the main contributions of this paper is modeling independence constraints on the sensor networks.

Submodularity is a property that describes set functions similar to how convexity describes functions in a continuous space. For ping sequence optimization, submodular functions can be used to find optimal subsets of buoys to achieve objectives like maximizing coverage of non-interfering buoys, or maximizing probability of target detection in a target tracking scenario. Rather than exhaustively searching over all combinations of subsets, submodular functions provide a fast and tractable framework to compute a solution [6], [9], [10].

Let the set of available objects, known as the ground set, be denoted as $V$. A submodular function $f$ maps a set of objects denoted by a binary indicator vector of length $V$ to a real number. The binary indicator vector is represented by the expression $2^V$ since the variable can take two values and is indexed by elements of set $V$. As mentioned previously, a value of 1 or 0 for the $i^{th}$ element of the indicator vector denotes the inclusion or exclusion of the $i^{th}$ element of the ground set $V$, and therefore any subset $A \subseteq V$ can be placed in one-to-one correspondence with incidence vectors.

A submodular function $f$ is defined as one with the following property: for all $A, B \subseteq V$,

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{1}$$

In light of the above equivalence between subsets and incidence vectors, a submodular function can also seen to be one that operates on 0/1-vectors with entries elements indexed by elements of $V$. Submodularity is sometimes viewed as a discrete analog to convexity [9], although it should be noted that submodularity and convexity are quite distinct for many reasons.

Submodularity can equivalently, and perhaps more intuitively, be expressed via the notion of diminishing returns, i.e., the incremental gain of the objective diminishes as the context grows. Defining the incremental gain of adding $v$ to $A$ as $f(v|A) = f(A \cup \{v\}) - f(A)$, then submodularity is defined as any function with $f(v|A) \geq f(v|B)$. for all $A \subseteq B \subset V$ and a $v \notin B$.

Submodularity is very closely tied to structures known as matroids, which generalize the notion of linear independence in vector spaces [7]. One can think of matroids as a generalization of matrices, which extend the definition of rank beyond column vectors of a matrix to more general independent subsets over a finite ground set. More importantly, SFO allows for matroid independence constraints to be placed on the problem, which means complicated variable dependence patterns can be encoded into the problem and polynomial time solutions can be obtained. Given a finite set $V$ and a finite set of subsets $\mathcal{I} = \{I_1, I_2, \dots\}$, the pair $(V, \mathcal{I})$ is said to be a matroid when the family of sets $\mathcal{I}$ satisfies the following three properties:

1) $\emptyset \in \mathcal{I}$
2) $I_1 \subseteq I_2 \in \mathcal{I}$
3) $I_1, I_2 \in \mathcal{I}, |I_1| < |I_2| \implies \exists v \in I_2 \backslash I_1 : I_1 \cup v \in \mathcal{I}$

This leads us to the independence graphs, where nodes on the graph represent sensors and edges denote pairwise independence between sensors. An edge between two sensors, in other words, means both sensors can be used at the same time. For this setup, the nodes in any fully connected subgraph (clique) is an allowable subset. The set of all cliques, also called the clique complex, from this independence graph, denoted by $G$, can form a partition matroid if the stable sets of the complimentary graph $\bar{G}$ form a partition. This is not true in general for the independence graphs generated from the interference patterns in a sensor field. However, the clique complex can be modeled as the intersection of $k$ matroids [11]. Kashiwabara et al. give the following algorithm to construct the $k$ matroids.

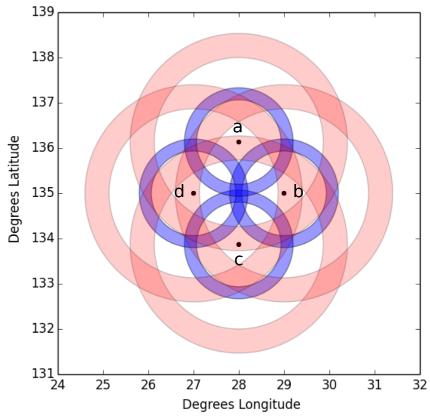
**Algorithm 1** Clique Complex to Matroids

1: Construct the stable-set graph $\mathscr{S}(G)$
2: Find a $k$-coloring of $\mathscr{S}(G)$
3: The nodes of each color form the partitions of one of the $k$ matroids

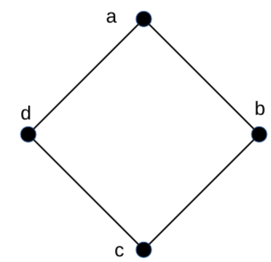
The stable-set graph of $G$, denoted $\mathscr{S}(G)$, is a graph whose vertices are the maximal stable, or independent, sets of $G$. An edge exists between vertices that share vertices of $G$. The $k$-coloring problem, in which the goal is to assign each vertex one of $k$ colors such that adjacent vertices are different colors. In terms of $\mathscr{S}(G)$, vertices of the same color do not share any vertices from $G$, and thus each vertex forms a partition. Kashiwabara et al. also prove that the clique complex of a graph of $n$ vertices can be represented by the intersection of at most $n-1$ partition matroids [11]. By turning the interference pattern of a sensor field into a set of matroid constraints, we can guarantee that two interfering sensors will not be chosen in the solution.

## III. Application to Ping Sequence Optimization

We apply this sensor selection framework to active sonar arrays, where each buoy has a co-located transmitter and receiver that operates monostatically. However, since SFO allows for multiple buoys to be selected, the array functions multistatically in that multiple receivers are operating simultaneously and at potentially overlapping regions. An example of a spatial buoy arrangement where some of the buoys interfere can be found in Fig. 2a. The four buoys are arranged in a diamond pattern with locations represented by black dots. In Fig. 2a, the blue rings denote the coverage regions for each buoy and the red rings denote the regions where another buoy will interfere with a given buoy. Coverage is defined by the probability of target detection for a buoy. If two interfering buoys transmit simultaneously, the direct path signal from the first will arrive at the second when the second buoy's reflections would arrive. Fig. 2a also shows the relationship between the coverage and interference regions for the buoys. In this arrangement, the buoys across from each other, i.e. the top and bottom pair and left and right pair, will interfere with each other, since the buoys in each pair are in the red interference region of the other buoy. However, any other pair of buoys can ping simultaneously [5].




(a) Four buoys (black dots) with coverage regions (blue rings) and interference regions (red rings).




(b) Independence graph for four buoys as nodes with edges that signify pairwise independence.

Fig. 2: Relationship between the independence graph, coverage regions and interference regions for a four buoy arrangement.

In order to find out the maximum number of buoys that can ping simultaneously, the largest set of nodes is picked such that all the nodes in the set are connected to every node in the set. Note that self-loops are implied, since a buoy does not interfere with itself. The problem of finding the largest subset of fully connected nodes is a well known problem in computer science [12]. Exact methods for solving this problem run in exponential time, but for reasonable graph sizes (a hundred vertices), the algorithm runs fairly quickly. For example, if the graph meets certain conditions, i.e. if the graph is "planar" or "perfect," finding the largest clique can be solved in polynomial time [13]. For the arrangement in Fig. 2a, there is a four-way tie for largest clique, which are the adjacent pairs (top and left buoys, left and bottom buoys, bottom and right buoys, and right and top buoys). The independence graph for this arrangement is depicted in Fig. 2b. In a real scenario, the detection regions will not be perfect rings, so one of the pairs might have better coverage than the others. A more complicated interference pattern will emerge as the number of buoys is increased, which is demonstrated in Fig. 5b.

Our objective function is a variant of probabilistic coverage. It utilizes target state estimates to help determine which buoys are selected. Let $V$ be the set of $N$ buoys $b_i, i = 1...N$. Let $B \subseteq V$ such that $B$ is a clique of $G$, where $G$ is the independence graph determined by the interference pattern of all the buoys $b_i$ in $V$. Let the set of all sets of sensors that form cliques on the graph be a partition matroid $\mathcal{I}$. Coverage is a positive, non-decreasing objective, so the goal is to maximize the objective function. Two different problem formulations can be used, which correspond to maximizing the average coverage over a set of target locations [14], Equation (2), and maximizing the worst-case coverage over a set of target locations, Equation (3). Then the optimal set of buoys for the average-case scenario is given by

$$B^* \in \operatorname*{argmax}_{B \in \mathcal{I}} \frac{1}{M} \sum_{i=1}^{M} f_i\left(B\right). \tag{2}$$

The worst-case scenario is given by

$$B^* \in \operatorname*{argmax}_{B \in \mathcal{I}} \min_i f_i\left(B\right). \tag{3}$$

In both cases, $i = 1, ..., M$ corresponds to the predicted target locations and $M$ is the number of targets, and the functions $f_i : 2^V \to \mathbb{R}$ are given by the equation

$$f_i\left(B\right) = 1 - \prod_{b_i \in B} \left(1 - P_{i,b_i}\right) \tag{4}$$

where $P_{i,b_i}$ is the probability of detection of buoy $b_i$ at location $i$ determined by a table look-up for pre-computed probability of detection maps for each buoy.

Our approach emphasizes tracking in that the objective prioritizes covering areas where known targets are located, but it provides good coverage as well. After the algorithm addresses coverage of the known targets, it adds as many non-interfering buoys as are available, and thus provides an effective simultaneous track and search framework.

## IV. Matroid Constrained Saturate (MatSat)

To optimize the Equation (3), we develop a novel algorithm MatSat, which generalizes the Saturate algorithm created by Krause et al. [3]. Krause et al. use Saturate to optimize an objective function of the form

$$A^* = \operatorname*{argmax}_{|A| \leq k} \min_i f_i\left(A\right) \tag{5}$$

where $f_i\left(A\right)$ is a set of monotone submodular functions. Saturate solves this worst-case optimization problem by proposing an alternative formulation and relaxing the cardinality constraint from $|A| \leq k$ to $|A| \leq \alpha k$. As long as $\alpha$ is large enough, the solution $\hat{A}$ from the SATURATE algorithm guarantees that

$$\min_i f_i(\hat{A}) \geq \underset{|A|\leq k}{\operatorname{argmax}} \min_i f_i\left(A\right) \quad \text{and} \quad |\hat{A}| \leq \alpha k.$$

Krause et al. claim that the only way to achieve a non-trivial guarantee is to relax the constraint, which limits both the types of constraints that can be applied to the problem as well as the values the objective functions can take, i.e. integral or rational valued objective functions. Matroid constraints, for instance, have no immediately obvious relaxation. One way to relax the matroid constraint, however, might be to expand the bases of the matroid. For our task, this leads to undesirable solutions. For example, for ping sequence optimization, the resulting solutions might contain interfering buoys. However, there is another way to achieve non-trivial guarantees, which is to relax the objective itself, leaving the constraints intact, and produce a fractional bound on the objective function, something that is made possible thanks to the use of submodularity. The proposed algorithm, MatSat, uses this alternative approach to find a solution such that a fraction $\gamma$ of the submodular functions are above a minimum value $\beta$. Moreover, the user can set particular values of $\beta$ or $\gamma$, as long as $\beta$ is less than the submodular guarantee $\alpha$ and $\gamma < 1$. The derivation for the lower bound is given below.

By relaxing the problem, we can consider the following constrained optimization problem:

$$A^* = \underset{A\in\mathcal{I}}{\operatorname{argmax}} \min_i f_i\left(A\right) \tag{6}$$

where $\mathcal{I}$ are the independent sets of a matroid. For a fixed value of $c$, which can be thought of as the saturation level, we can determine if $f_i\left(A\right) \geq c$ via submodular maximization of the following surrogate function:

$$f^c\left(B\right) = \frac{1}{M}\sum_{i=1}^{M} \min\{f_i\left(B\right), c\}. \tag{7}$$

$f^c(B)$ is a submodular function, because it is a non-negatively weighted sum of functions $\min\{f_i\left(B\right), c\}$ which are submodular [15]. At each iteration of MatSat, we run a greedy algorithm described in Algorithm 2. Basically, the greedy approach selects the buoy which provides the best incremental gain in coverage and does not interfere with any of the other previously selected buoys.

**Algorithm 2** Greedy $\left(f^c, c\right)$

$B \leftarrow \emptyset$
**while** $\exists b \in V \backslash B$ s.t. $B \cup \{b\} \in \mathcal{I}$ and $f^c\left(b|B\right) > 0$ **do**
  $S \leftarrow \{b \in V\backslash B :\ B \cup \{b\} \in \mathcal{I}\}$
  $s^* \leftarrow \operatorname{argmax}_{s\in S} f^c\left(b|B\right)$
  $B \leftarrow B \cup \{s^*\}$
**end while**
**return** $B$

MatSat is outlined in Algorithm 3. Given monotone submodular functions $\left(f_1, ..., f_M\right)$, approximation guarantee $\alpha$ for the matroid constrained submodular maximization problem, and tolerance threshold $\epsilon$, we first set $c_{min}$ and $c_{max}$ to values that ensure the true optimal value lies in the interval. While performing a binary search over $c$, we test the value of the approximate solution $f^c(\hat{B})$ against the lower bound $\alpha c$. If the approximate solution is less than the lower bound, we know that the true optimal is less than $c$, so we limit the search to the lower half of the interval. Likewise, if the lower bound is met, we store the solution (which, as we describe below, is fractionally good w.r.t. the current $c$) and then continue attempt to find a better one (higher $c$) by searching over the upper half of the interval. We stop when the range falls within the tolerance.

**Algorithm 3** MatSat $\left(f_1, ..., f_M, \alpha, \epsilon\right)$

1: $c_{min} \leftarrow 0, c_{max} \leftarrow \min_i f_i\left(V\right)$
2: **while** $\left(c_{max} - c_{min}\right) > \epsilon$ **do**
3:   $c \leftarrow \left(c_{max} - c_{min}\right)/2$
4:   $f^c\left(B\right) \leftarrow \frac{1}{M}\sum_{i=1}^{M} \min\{f_i\left(B\right), c\}$
5:   $\hat{B} \leftarrow$ Greedy $\left(f^c, c\right)$
6:   **if** $f^c(\hat{B}) < \alpha c$ **then**
7:     $c_{max} \leftarrow c$
8:   **else**
9:     $c_{min} \leftarrow c, \quad B_{best} \leftarrow \hat{B}$
10:   **end if**
11: **end while**
12: **return** $B_{best}$

**Theorem 1.** *Given a value $\beta < \alpha$, MatSat finds a solution $\hat{B}$ that guarantees the following fraction $\gamma$ of the $M$ functions min$\{f_i\left(\hat{B}\right), c\} \geq \beta c$:*

$$\gamma \geq \frac{\alpha - \beta}{1 - \beta}$$

*where $\alpha$ is the approximation guarantee for matroid independence set constrained submodular maximization problem.*

***Proof*** $f^c(B) \geq c$ only if $\min_i f_i(B) \geq c$. When all $f_i(B) \geq c$, then $f^c(B) = c$. Likewise, when any $f_i(B) < c$ then the $f^c(B) < c$, since the maximum value of $f^c(B)$ is $c$. The greedy solution $\hat{B}$ for maximizing a monotone submodular function subject to a matroid constraint is $f^c(\hat{B}) \geq \alpha f^c(B^*)$, where $\alpha$ depends on the algorithm chosen. If line 6 of Algorithm 3 is true, then $f^c(\hat{B}) < \alpha c$ which implies that $\min_i \ f_i(B^*) < c$. Line 6 being true also implies that $c$ is too large, so $c_{max} \leftarrow c$. At line 12 of Algorithm 3, $\left(c_{max} - c_{min}\right) \leq \epsilon$ and the true optimal value $\min_i \ f_i(B^*)$ is in the interval $[c_{min}, c_{max}]$. The submodular approximation guarantee ensures that $f^c(\hat{B}) \geq \alpha f^c(B^*)$ and $f^c(\hat{B}) \geq \alpha c$. Given a value for $\beta$, the terms of $f^c(\hat{B})$ can be split into two groups, one that have value less than $\beta c$ and the other greater than or equal to $\beta c$:

$$f^c(\hat{B}) = \frac{1}{M}\sum_{i:\min\{f_i(B),c\}<\beta c} \min\{f_i\left(B\right), c\} + \frac{1}{M}\sum_{i:\min\{f_i(B),c\}\geq\beta c} \min\{f_i\left(B\right), c\}.$$

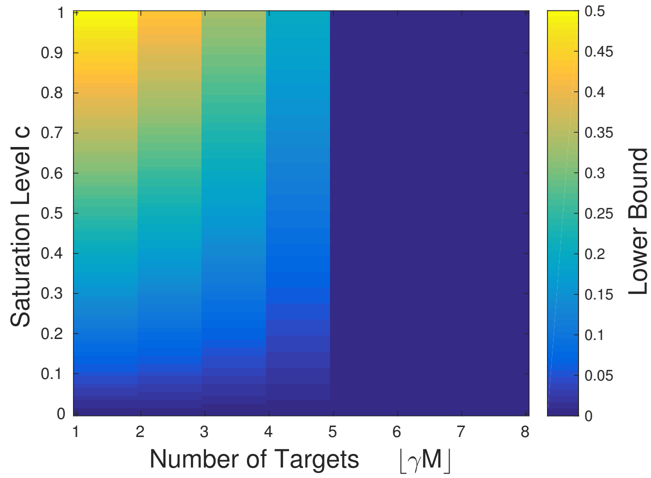



Fig. 3: Visualization of lower bound as the saturation level $c$ and the fractional guarantee $\lfloor \gamma M \rfloor$ vary. The submodular guarantee is fixed at $\alpha = \frac{1}{2}$.

Let $\gamma$ be the fraction of terms that meet the $\beta c$ threshold. Then, the two summation terms become $f^c(\hat{B}) = (1-\gamma)\,\beta c + \gamma c \geq \alpha c$. Rearranged, the expression becomes $\gamma \geq \frac{\alpha-\beta}{1-\beta}$. A visualization of the lower bound over values of the saturation level $c$ and fractional number of targets $\lfloor \gamma M \rfloor$ is shown in Fig. 3. In this case, the submodular guarantee for a monotone submodular function constrained by a single matroid via the greedy algorithm, which is $\alpha = 1/2$.

## V. Experimental Results

In this section, we compare the performance of the proposed ping sequence optimization algorithm MatSat to the previously proposed SFO-Greedy and exhaustive search. MatSat and exhaustive search are optimized with respect to Equation (3), while SFO-Greedy is optimized with respect to Equation (2). If SFO-Greedy were to be optimized with respect to Equation (3), the algorithm would always select the empty set, except in the degenerate case where there is a buoy that can detect every single target. For the first experiment, there are nine buoys in a grid pattern spaced 128 km away from their neighbors and seven targets. The nine coverage regions and seven target locations are represented by blue rings the red triangles, respectively, in Figure 4. In this experiment, we allow the targets to have unequal probabilities of detection. Ordered from top to bottom and left to right, the targets have probability of detection of $P = \{0.6, 0.6, 0.9, 0.2, 0.6, 0.9, 0.9\}$ in each buoy's coverage region and $P = 0$ everywhere else. The independence graph for the buoys can be found in Fig. 5b, and the interference pattern in Fig. 5a. By applying Kashiwabara's method for creating a set of matroid constraints to the interference graph in Fig. 5b, the clique complex can be represented as the intersection of four partition matroids, whose partitions are given as $\{ad, be, cf, hi, g\}$, $\{ab, de, fi, gh, c\}$, $\{bc, ef, dg, a, h, i\}$, and $\{eh, a, b, c, d, e, f, i\}$.

The results for experiment one can be found in Table I. MatSat performs as well as the exhaustive search, which is to say it achieved the optimal solution both in terms of worst case and average coverage objectives. SFO-Greedy, however, chooses the buoys $\{a, f, h\}$ instead of $\{b, d, f, h\}$, which misses one of the targets and results in suboptimal worst case and average coverage.

| Method | **MatSat** | SFO-Greedy | Exhaustive search |
|---|---|---|---|
| Min PD | **0.200** | 0.000 | 0.200 |
| Mean PD | **0.671** | 0.586 | 0.671 |
| Buoys Selected | $\{b, d, f, h\}$ | $\{a, f, h\}$ | $\{b, d, f, h\}$ |

TABLE I: Probability of detection (PD) results for seven targets and nine buoys.

For the second experiment, we have thirty two buoys in a ring. The corresponding clique complex of the independence graph can be represented as the intersection of four matroids. In this experiment, the targets have an equal probability of detection of $P = 0.8$ in each buoy's coverage region and $P = 0$ everywhere else. We assume here that there is no sensor drift during the experiment. Two targets with random initial location, constrained to be within the buoy array's detection area, and constant velocities are present for each trial. The second experiment consists of two hundred trials with each trial lasted sixty-four time-steps or until a target moved out of the array's detection area.

For each trial, we initialize the target location and velocity and pass the initial state estimates into the SFO algorithm. Based on the objective function output for each target, we sample the probability that each target has a successful detection at the next time step and pass in the updated state estimates for the detected targets. Over the course of the trial, we accumulate the objective function values which form a

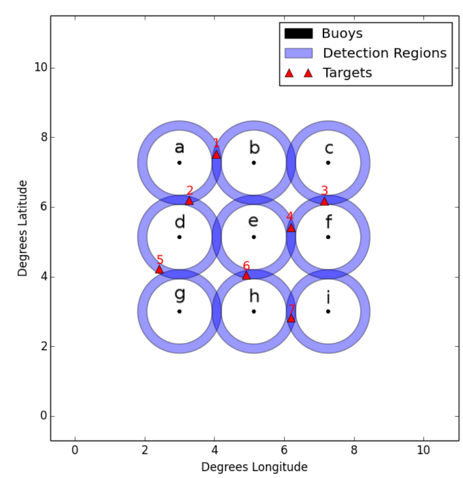
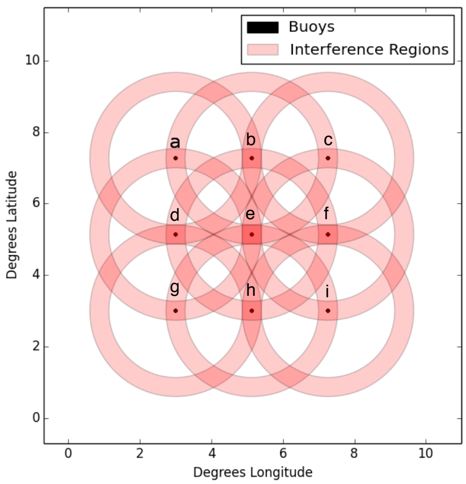
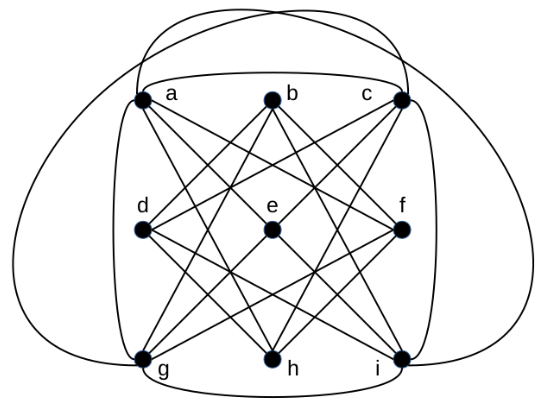



Fig. 4: Coverage pattern for the nine buoys in a three by three grid and target locations.




(a) Interference pattern for the nine buoys in a three by three grid.




(b) Independence graph for the nine buoys in a three by three grid.

Fig. 5: Interference regions and independence graph for a nine buoy experiment.

cumulative probability of detection (CPD) score for the two algorithms.

The results of the second experiment can be found in Fig. **??**. Both MatSat and SFO-Greedy have significantly higher fractional worst-case probability of detection than the lower bound across all fractions of the eight targets $\lfloor \gamma M \rfloor$. Even when the lower-bound provides a trivial guarantee, $\lfloor \gamma M \rfloor \geq 3$, both MatSat and SFO-Greedy match the optimal performance. In fact, the performance of MatSat and SFO-Greedy are practically indistinguishable from exhaustive search for any fraction of the targets. Fig. **??** zooms in on Fig. **??** to show the minute differences between the approximate solutions of the proposed algorithms and the true optimal. The only differences are at $\lfloor \gamma M \rfloor < 3$, where MatSat and SFO-Greedy have slightly lower minimum probabilities of detection. While Fig. **??** also shows that MatSat and SFO-Greedy have equivalent minimum probabilities of detection, MatSat provides a better theoretical guarantee than SFO-Greedy. The lower bound is shown by the red line in Fig. **??**. The bound reflects the submodular guarantee of the forward greedy algorithm used in the experimental code, which is $\frac{1}{k+1}$ [16]. Since the clique complex can be represented as the intersection of four matroids, the guarantee is 0.2.

## VI. Conclusion

In this paper, we propose a new optimization algorithm MatSat to solve a relaxed worst-case subset selection problem subject to a matroid constraint and derive lower bounds on the average performance as well as fractional worst-case performance. In applying independence graphs to a sensor selection problem, we demonstrate the utility of submodular function optimization (SFO) to the problem domain. Specifically for ping sequence optimization (PSO), SFO allows us go beyond the standard approach for buoy selection by allowing for simultaneous pinging. By posing the PSO as a submodular optimization problem or as a discrete problem with submodular structure, we are able to derive near-optimal solutions for combinatorial problems. We can select buoys based on target state that significantly improve the probability of detecting targets over a standard approach and achieve equivalent performance to an optimal exhaustive search approach. Moreover, our approach allows for simultaneous search and track objectives within the system.

## Acknowledgment

This work was funded by the U.S. Office of Naval Research, Contract Number N00014-12-G-0078, Delivery Order #24, and the ARO grant number W911NF1510450. We acknowledge the helpful discussions with Scott Wisdom and Rishabh Iyer of the University of Washington Department of Electrical Engineering.

## References

[1] R.K. Iyer and J.A. Bilmes, "Submodular optimization with submodular cover and submodular knapsack constraints," in *Advances in Neural Information Processing Systems 26*, pp. 2436–2444. Curran Associates, Inc., 2013.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications magazine, IEEE*, vol. 40, no. 8, pp. 102–114, 2002.

[3] A. Krause, B. McMahan, Guestrin C., and Gupta A., "Robust submodular observation selection," *Journal of Machine Learning Research (JMLR)*, vol. 9, pp. 2761–2801, December 2008.

[4] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *Signal Processing, IEEE Transactions on*, vol. 57, no. 2, pp. 451–462, Feb 2009.

[5] D.W. Krout and T. Powers, "Sensor management for multistatics," in *Information Fusion (FUSION), 2014 17th International Conference on*, July 2014, pp. 1–6.

[6] S. Fujishige, *Submodular functions and optimization*, vol. 58, Elsevier, 2005.

[7] J. Edmonds, "Matroids and the greedy algorithm," *Mathematical Programming*, vol. 1, no. 1, pp. 127–136, 1971.

[8] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a submodular set function subject to a matroid constraint," in *Integer programming and combinatorial optimization*, pp. 182–196. Springer, 2007.

[9] L. Lovsz, "Submodular functions and convexity," in *Mathematical Programming The State of the Art*, Achim Bachem, Bernhard Korte, and Martin Grtschel, Eds., pp. 235–257. Springer Berlin Heidelberg, 1983.

[10] A. Krause, "SFO: A toolbox for submodular function optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 1141–1144, Mar. 2010.

[11] K. Kashiwabara, Y. Okamoto, and T. Uno, "Matroid representation of clique complexes," *Discrete Applied Mathematics*, vol. 155, no. 15, pp. 1910–1929, 2007.

[12] C. Bron and J. Kerbosch, "Algorithm 457: finding all cliques of an undirected graph," *Communications of the ACM*, vol. 16, no. 9, pp. 575–577, 1973.

[13] M.R. Garey, D.S. Johnson, and L. Stockmeyer, "Some simplified np-complete graph problems," *Theoretical computer science*, vol. 1, no. 3, pp. 237–267, 1976.

[14] T. Powers, D.W. Krout, and L. Atlas, "Sensor selection from independence graphs using submodularity," in *Information Fusion (Fusion), 2015 18th International Conference on*, July 2015, pp. 333–337.

[15] Toshihiro Fujito, "Approximation algorithms for submodular set cover with applications," *IEICE Transactions on Information and Systems*, vol. 83, no. 3, pp. 480–487, 2000.

[16] M.L. Fisher, G.L. Nemhauser, and L.A. Wolsey, "An analysis of approximations for maximizing submodular set functions-ii," in *Polyhedral Combinatorics*, M.L. Balinski and A.J. Hoffman, Eds., vol. 8 of *Mathematical Programming Studies*, pp. 73–87. Springer Berlin Heidelberg, 1978.