Probabilistic Maximum Set Cover with Path Constraints for Informative Path Planning

Graeme Best[1] and **Robert Fitch**[1,2]
[1]Australian Centre for Field Robotics, The University of Sydney, Australia
{g.best,rfitch}@acfr.usyd.edu.au
[2]Centre for Autonomous Systems, University of Technology Sydney, Australia

Abstract

We pose a new formulation for informative path planning problems as a generalisation of the well-known maximum set cover problem. This new formulation adds path constraints and travel costs, as well as a probabilistic observation model, to the maximum set cover problem. Our motivation is informative path planning applications where the observation model can be naturally encoded as overlapping subsets of a set of discrete elements. These elements may include features, landmarks, regions, targets or more abstract quantities, that the robot aims to observe while moving through the environment with a given travel budget. This formulation allows directly modelling the dependencies of observations from different viewpoints. We show this problem is NP-hard and propose a branch and bound tree search algorithm. Simulated experiments empirically evaluate the bounding heuristics, several tree expansion policies and convergence rate towards optimal. The tree pruning allows finding optimal or bounded-approximate solutions in a reasonable amount of time, and therefore indicates our work is suitable for practical applications.

1 Introduction

Many important robotic tasks belong to the information gathering class of problems, including object classification [Patten *et al.*, 2016], exploration and mapping [Kim and Eustice, 2015], target search and tracking [Cliff *et al.*, 2015], data collection in sensor networks [Faigl and Hollinger, 2014] and environmental monitoring [Das *et al.*, 2015]. These fundamental tasks appear in a diverse range of applications, for example outdoor robotic perception for agriculture, defence, infrastructure monitoring and environmental science. The objective in information gathering problems is generally to increase knowledge of quantities of interest by making informative observations of the environment. Informative path planning is the problem of optimising the robot's path to maximise the information gathered. Various models of information can be used in the planning objective, which is dependent on the class of quantities being estimated, such as an environment map, target locations, object features or an ecosystem process. Most existing methods assume a general model for information gain that is monotone submodular [Meliou *et al.*, 2007; Hollinger and Sukhatme, 2014], and specific instances of this broad class of models need to be defined for different applications, such as Gaussian processes for temperature sensing [Garg and Ayanian, 2014]. In this paper we propose a new formulation that models information as overlapping probabilistic-sets of discrete elements. This formulation naturally models many important problems, such as object classification, data collection, area coverage and precision agriculture tasks.

The *set cover problem* [Vazirani, 2001] is a computational problem where the aim is to find an optimal selection of subsets that covers a universe set of elements. The maximum set cover problem [Khuller *et al.*, 1999] is a well-studied variant that optimises the measure of coverage given a budget on the number of selected subsets. This concept naturally relates to informative path planning where the objective is to maximise the collection of information for a given travel budget. Additionally, the coverage objectives can directly model the information overlap of observations at multiple locations.

We formulate a generalisation of the maximum cover problem for path planning with uncertain observations. The objective is to find a path through the environment that maximises the expected weighted set cover, while constrained to a travel budget and subject to path constraints. This set cover formulation is motivated by applications where the observation model is naturally encoded as overlapping subsets of a set of discrete elements. The elements may include features, landmarks, regions, targets or more abstract quantities that are typi-

cally used to model observations, and therefore can aptly model a wide range of problems.

This formulation presents an optimisation problem to be solved. We begin by showing the problem is NP-hard as there exists a reduction from the maximum set cover problem. We investigate an adapted set cover greedy approach for the generalised problem and show the typical approximation bounds do not hold. We extend the greedy approach with a branch and bound (BnB) algorithm. Unlike most alternative approaches to informative path planning, the BnB algorithm has several useful properties that are important for practical scenarios: it is any-time, approximation bounds are computed for intermediate solutions, it converges towards the optimal solution and it can efficiently perform an optimal exhaustive search. The approximation bounds for partial solutions are used to improve the efficiency by adaptively pruning the search space. We propose and analyse several tree expansion policies for guiding the pruning.

We are interested specifically in performance properties in practice in order to enable applications. We describe simulation experiments for scenarios motivated by communication with a sensor field. We empirically evaluate various tree expansion policies for the algorithm and the convergence rate towards optimal. The algorithm found the optimal solution after exploring only 15 % of the search space on average. Near-optimal solutions with optimality bounds are found in a reasonable amount of time (milliseconds to minutes), and therefore indicates the approach is suitable for practical applications.

1.1 Applications

We address the problem as an abstract computational problem, but we first detail motivating applications. The most apparent application is coverage tasks where the objective is to view a set of landmarks or regions [Dornhege *et al.*, 2016]. Multiple landmarks can be viewed from each viewpoint and each landmark can be viewed from multiple viewpoints. Similarly, in precision weeding, spraying and pruning tasks each target can be reached by multiple configurations of a robot arm [Lee *et al.*, 2014]. A related problem is data collection and resource replenishment in sensor networks [Faigl and Hollinger, 2014]. The landmarks may evolve over time, such as moving targets or temporal processes [Best *et al.*, 2015; Mathew *et al.*, 2015].

Another interesting application area is where multiple elements describe each quantity of interest. This is particularly evident in feature-based object classification problems where an object is viewed from multiple angles to improve classification confidence [Patten *et al.*, 2016; Best *et al.*, 2016b; Hollinger *et al.*, 2011]. Observing discriminative features generally increases classification performance. An observation model can be derived from probabilistic estimates of properties such as object identities and the presence of occlusions.

2 Related Work

In this section we first review related approaches to informative path planning. Secondly, we provide an extended discussion of the standard set cover problem and relevant variants in order to set the context for our new variant. Lastly, we discuss relationships to variants of the travelling salesman problem and the orienteering problem.

2.1 Informative Path Planning

Informative path planning has been studied widely in the literature, with various approaches and applications. One approach is to first formulate it is as a sensor placement problem [Krause *et al.*, 2008] then plan a path through the selected locations [Hollinger *et al.*, 2012]. This approach makes guarantees on the informativeness of the path but does not directly consider the movement costs or a travel budget. Other approaches [Meliou *et al.*, 2007; Lim *et al.*, 2016] find paths with approximately optimal cost with a guaranteed level of informativeness. Branch and bound algorithms [Binney and Sukhatme, 2012; Singh *et al.*, 2009], RRT variants [Hollinger and Sukhatme, 2014], Monte Carlo tree search [Best *et al.*, 2016a], recursive greedy [Chekuri and Pal, 2005] and self-organising maps [Best *et al.*, 2016b] allow planning while directly addressing a limited travel budget.

Branch and bound (BnB) methods are advantageous since they output calculated approximation bounds for partial solutions and they efficiently perform an exhaustive search given sufficient time. We propose a BnB algorithm which, in contrast to most BnB algorithms (e.g., [Binney and Sukhatme, 2012]), uses a heuristic to compute a lower bound at each node, in addition to the upper bound. This allows more aggressive pruning of the search space while still providing guarantees. A similar approach is proposed in [Singh *et al.*, 2009], although we use heuristics adapted for our problem: lower bounds using an adapted set cover greedy algorithm [Vazirani, 2001], and upper bounds using the reachability subject to the travel budget [Binney and Sukhatme, 2012]. Additionally, we refine the bounds as descendants are explored, which may be exploited by a best-first tree expansion policy [Mehlhorn and Sanders, 2008]. We empirically evaluate several such expansion policies.

Informative path planning is driven by a measure of the information gained by taking observations. In many cases, the measure describes the reduction in uncertainty of a process with spatial correlations, such as Gaussian processes [Garg and Ayanian, 2014; Hollinger and Sukhatme, 2014]. This class of models is well suited for continuous processes, e.g. ocean modelling [Das *et al.*, 2015; Binney and Sukhatme, 2012]. Typically, the information

measure belongs to the class of functions that satisfy the monotone submodularity property [Meliou *et al.*, 2007; Hollinger and Sukhatme, 2014]. We formulate a new instance of this class of information measures motivated by the applications described in Sec. 1.1. The objective in these applications is intermediately modelled as discrete quantities, without any clear underlying continuous process linking all quantities. Subsets of these quantities may be observed simultaneously, such as a subset of features viewed from particular viewpoints [Best *et al.*, 2016b; Hollinger *et al.*, 2011]. We formulate the objective for this class of problems explicitly as maximally collecting these quantities of interest. The dependencies between observations are modelled directly as overlap between the subsets for different viewpoints.

2.2 The Set Cover Problem

The set cover problem exists in various forms, both for the development of fundamental algorithmic techniques and for its applicability to a range of problems [Vazirani, 2001; Hochbaum, 1997]. The classic set cover problem is posed as follows. Given a universe set of elements, and a collection of subsets of the universe set with associated costs, find a minimum-cost subcollection of subsets that covers all elements of the universe set.

The maximum cover problem [Vazirani, 2001; Khuller *et al.*, 1999] is a variant where the objective and constraint of the set cover problem are swapped. The goal is to find the subcollection that maximally covers the universe set and has cost less than a given budget. Generalisations of this problem include cases where the elements have non-uniform element weights and the subsets have non-uniform costs. The concept of selecting from a set of discrete choices given a budget naturally relates to informative path planning problems where a robot chooses viewpoints given an observation or travel budget. This motivates our generalisation of the maximum cover problem, and in Sec. 3.5 we show that the weighted maximum cover problem reduces to our problem.

The set cover problem and its variants are NP-hard but there exists polynomial-time approximation algorithms [Hochbaum, 1997]. Typically, the most effective approach is a greedy algorithm where subsets are selected sequentially based on the relative effectiveness of choosing a subset given a partial solution. For the standard set cover problem with n subsets, this algorithm achieves a $\log n$ approximation bound, and is the best that can be achieved [Chvatal, 1979]. For the maximum cover problem, an adapted greedy approach achieves a $1-1/e$ approximation bound [Hochbaum, 1997; Khuller *et al.*, 1999]. The greedy approach can be formulated for our problem, however we show the typical approximation bounds do not hold. A similar result is shown in [Singh *et al.*, 2009] for a related problem.

2.3 Travelling Salesman Problem and Orienteering Problem Variants

The objective of the generalised TSP (GTSP) is to find a minimum cost circuit that visits at least one out of every vertex in a collection of vertex sets. One approach to this problem transforms the graph into an instance of the standard TSP [Noon and Bean, 1989]. Techniques need to be applied to take into account various classes of problem instances, such as if the vertex sets are overlapping or if the circuit is allowed to visit a vertex set multiple times. The GTSP has been used in robotics problems, such as persistent recharging of mobile agents [Mathew *et al.*, 2015]. The GTSP is closely related to the standard set cover problem since the objective is to complete a minimum-cost coverage of all subsets.

The maximum set cover problem is more closely related to the orienteering problem (OP) [Gunawan *et al.*, 2016]. The objective is to find a path or tour through a graph that visits the largest number of weighted vertices, constrained to a maximum travel budget. The OP is closely related to our set cover generalisation for the case where each subset contains one element and each element is in one subset. The generalised OP (GOP) [Geem *et al.*, 2005] extends the OP to set-based objectives in a similar way to the GTSP. Deterministic instances of our problem are equivalent to the GOP when $k \to \infty$ in their objective function. The OP with neighbourhoods [Faigl *et al.*, 2016] defines the sets as continuous regions rather than discrete sets, and this formulation has been extended for multi-robot active perception [Best *et al.*, 2016b]. Our problem differs from these OP variants primarily due to defining the sets probabilistically.

3 Probabilistic Maximum Set Cover with Path Constraints

In this section we formulate the proposed generalised set cover problem. The environment is represented by a graph, such that vertices represent viewpoints, and paths through the graph represent robot trajectories. Each viewpoint has an associated predicted observation defined as a probabilistic subset for a weighted set cover problem. The objective is to plan a path that maximises the expected weighted set cover, while constrained to a travel budget. We show that this problem is NP-hard.

3.1 Environment Representation

The robot's configuration space and available actions are represented by a graph with vertices $v_i \in \mathcal{V}$ and edges $e_{ij} := \langle v_i, v_j \rangle \in \mathcal{V} \times \mathcal{V} \in \mathcal{E}$. Each vertex v_i describes a viewpoint where the robot may make an observation. Each edge e_{ij} describes the action of moving from v_i to v_j and has an associated travel cost $c\langle v_i, v_j \rangle := c_{ij} \geq 0$. A path X through the environment is described as a

sequence of non-repeating vertices $V = (v_a, v_b, v_c, ...)$, or equivalently as the sequence of edges $E = (e_{ab}, e_{bc}, ...)$. We assume the start vertex is fixed as $v_a = v_0$. The total travel cost of a path X is

$$C(X) := \sum_{e_{ij} \in X} c_{ij}. \tag{1}$$

The set of all possible paths is denoted $\mathcal{X}$. In the proposed algorithms we assume that the graph is complete, however this is not restrictive.

3.2 Information and Observations

The information that the robot may gather is represented by a finite universe set $\mathcal{U}$ of discrete elements $u_i \in \mathcal{U}$. Every element u_i has an associated weight $w_i > 0$ which represents a reward for observing u_i.

At each viewpoint $v_i \in \mathcal{V}$, the robot makes an observation S_i. An observation is encoded as a subset of elements $S_i \subseteq \mathcal{U}$. The set of observations over all vertices is denoted $\mathcal{S}$. The weight (reward) of a single observation $S_i \subseteq \mathcal{U}$ is given by the weighted set cover:

$$W(S_i) = \sum_{u_j \in S_i} w_j.$$

A path X through the graph makes the sequence of observations $(S_a, S_b, S_c, ...) = \mathcal{S}^X$ with a combined weight calculated as the weighted set cover of the union of all elements observed on this path:

$$\begin{aligned} W(X) &:= W\left(\bigcup_{S_i \in \mathcal{S}^X} S_i\right) \\ &= \sum_{u_j} w_j, \quad u_j \in \bigcup_{S_i \in \mathcal{S}^X} S_i. \end{aligned}$$

For convenience, $W(X)$ is equivalently rewritten as

$$W(X) = \sum_{u_j \in \mathcal{U}} \left[w_j \left(1 - \prod_{S_i \in \mathcal{S}^X} \left(1 - \mathbb{1}_{S_i}(u_j) \right) \right) \right], \tag{2}$$

where $\mathbb{1}$ is the indicator function for set membership:

$$\mathbb{1}_S(u) := \begin{cases} 1 & \text{if } u \in S, \\ 0 & \text{if } u \notin S. \end{cases}$$

In (2), the $1 - \prod(\cdot)$ is effectively performing a logical disjunction operation for u_j over all visited sets.

3.3 Uncertain Predicted Observations

In most robotic applications, the observation from a viewpoint is not known precisely prior to making the observation, and instead may be predicted probabilistically. We therefore assume that the subset S_i of elements observed by making an observation at viewpoint v_i is defined by the probability of set memberships. More specifically, the observation belief is modelled by given probabilities that an element u will be observed from a viewpoint v_i, i.e.,

$$p(u, S_i) := \mathbb{P}(u \in S_i), \quad \forall u \in \mathcal{U}, v_i \in \mathcal{V}.$$

Therefore, by generalising (2), the weight (reward) for a path is given by the expected weighted set cover:

$$\begin{aligned} \bar{W}(X) &:= \mathbb{E}[W(X)] \\ &= \sum_{u_j \in \mathcal{U}} \left[w_j \left(1 - \prod_{S_i \in \mathcal{S}^X} \Big(1 - p(u_j, S_i) \Big) \right) \right]. \end{aligned} \tag{3}$$

3.4 Problem Statement

The goal of the probabilistic maximum set cover with path constraints problem is to plan a path that maximises the expected weighted set cover (3) given that the length of the path (1) can not be longer than a given travel budget L. More formally, the optimisation problem is stated as follows.

Problem 1. *Given a set of elements $u_i \in \mathcal{U}$ with element weights w_i, a set of vertices $v_j \in \mathcal{V}$ with subsets $S_j \subseteq \mathcal{U}$ defined by probabilities of predicted set memberships $p(u, \mathcal{S}_j)$, a start vertex $v_0 \in \mathcal{V}$, a set of edges $e_{jk} \in \mathcal{E}$ with edge costs c_{jk}, and a travel budget L, find the optimal path X^*, where*

$$\begin{aligned} X^* := \ & \operatorname*{argmax}_{X \in \mathcal{X}} \left(\bar{W}(X) \right), \\ & \textit{such that } C(X) \leq L. \end{aligned}$$

3.5 Complexity

In the Appendix we show that the problem is NP-hard as there is a reduction from the maximum set cover problem [Khuller *et al.*, 1999]. As indicated in Sec. 2.3, reductions can also readily be applied from variants of the orienteering problem [Gunawan *et al.*, 2016].

4 Greedy Inapproximability

An exact solution to the problem can be found by performing a naive exhaustive search that enumerates all feasible paths; however, this is clearly intractable for large problems. Given that the problem is NP-hard, we turn our attention to approximation algorithms. In this section we introduce the standard greedy approach to the set cover problems adapted for the generalised formulation. However, we show with a counter-example that the typical approximation bounds do not hold. Later in Sec. 5 we propose a branch and bound algorithm that extends the greedy approach.

Algorithm 1 Greedy algorithm.

function GREEDY ($\mathcal{U}, \{w_i\}, \mathcal{V}, \mathcal{S}, v_0, \mathcal{E}, \{c_{jk}\}, L$)
  $i \leftarrow 0$
  $X_0 \leftarrow S_0$ ▷ Partial solution
  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_0\}$ ▷ Candidate subsets
  repeat
    $G \leftarrow \underset{S_j \in \mathcal{S}}{\operatorname{argmax}} \left(\frac{\bar{W}(X \cup S_j) - \bar{W}(X)}{c\langle X_i, v_j \rangle} \right)$
    if $C(X) + c\langle X_i, G \rangle \leq L$ **then**
      $i \leftarrow i + 1$
      $X_i \leftarrow G$
    $\mathcal{S} \leftarrow \{\mathcal{S}\} \setminus \{G\}$
  until $\mathcal{S} = \emptyset$
  return $X = (X_0, X_1, X_2, ...)$

4.1 Greedy Algorithm

Alg. 1 shows the standard greedy approach for set cover problems adapted for the generalised problem. At each iteration, the subset G that maximises effectiveness and does not exceed the budget L is added to the partial solution path X. We define the effectiveness of adding subset G to a partial solution X as the ratio of the increase in weight to the increase in cost. The key difference to standard set cover problems is that the cost of selecting a candidate subset S_j is the edge cost $c\langle X_i, v_j \rangle$, which changes at each iteration based on the previously selected set X_i. This algorithm has $\mathcal{O}(|\mathcal{S}|^2)$ runtime in the number of subsets.

4.2 Counter-example for Bounds

For most variants of the set cover problem the greedy approach is a $(1 - 1/e)$ constant bounded approximation. We show with a counter-example that this guarantee does not hold for the generalised problem, even when the element weights are uniform and the sets are deterministic, due to the non-constant cost per subset that varies dependent on the previously chosen subset.

Consider a problem instance with $N + 1$ vertices $\{v_0, v_1, v_2, ...v_N\} = \mathcal{V}$, where the start vertex v_0 has an empty associated subset $S_0 = \emptyset$ and all others have single element subsets $S_i = \{u_i\}, \forall i > 1$. The elements have uniform weights $w_i = 1, \forall i$. The edge costs are defined as follows, with $0 < \varepsilon \ll L$:

$$\begin{cases} c_{0i} = L - (N - i)\varepsilon, & \forall i : 1 \leq i \leq N - 1, \\ c_{0N} = \varepsilon, & \\ c_{ij} = \varepsilon, & \forall i, j : j = i + 1,\ 1 \leq i \leq N, \\ c_{ji} = L, & \forall i, j : j = i + 1,\ 1 \leq i \leq N, \end{cases}$$

and all other edge costs are given by the shortest path through the edges defined above. Figure 1 illustrates the edge costs for this problem instance with $N = 3$.

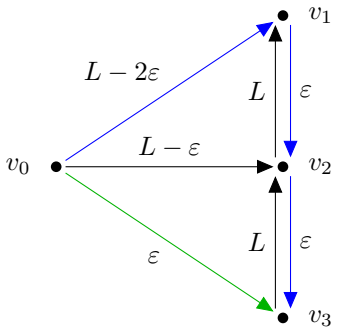


Figure 1: A problem instance for the generalised maximum set cover problem, as described in the text for $N = 3$, where the greedy approach (green) performs poorly compared to the optimal solution (blue). Each subset associated with v_1, v_2 and v_3 contains a single unique element with unit weight.

The optimal solution is clearly $X^* = (v_0, v_1, v_2, ...v_N)$, which has a cost of L and a set cover weight of $\bar{W}(X^*) = N$. However the greedy algorithm will select $X' = (v_0, v_N)$ since e_{0N} is the lowest cost edge from v_0 and then all edges from v_N are over-budget. The greedy solution has a set cover weight of $\bar{W}(X') = 1 = \bar{W}(X^*)/N$.

This large error is because the solution cost is defined for the ordered sequence of subset choices rather than the unordered set of choices as in the typical set cover problems. Observe that for this problem instance the (unordered) set of greedy selections is a subset of the optimal selection $\{X'\} \subset \{X^*\}$. If this occurred in the typical maximum set cover problems this would mean that it must be feasible to add further subsets to X' since the superset X^* still meets the budget constraint. However, when edge costs are considered, it is not feasible to append subsets to the sequence X' after v_N.

The approximation error can be larger when considering instances with multiple elements per subset, non-uniform element weights, or probabilistically defined subsets. In [Khuller *et al.*, 1999], a revised greedy algorithm is proposed for the maximum set cover problem where the elements have non-uniform weights. Unfortunately, their modification is not directly applicable for our generalisation since we have shown that the bounds are not constant even for the case with uniform weights.

5 Branch and Bound Algorithm

The inapproximability of greedy motivates the development of non-myopic planners. We propose a branch and bound algorithm that extends the greedy approach, and performs a tree search over the solution space. The algorithm iteratively grows a tree, where each node in the tree represents a partial solution path. Lower and upper bounds computed at each node give guarantees on the quality of partial solutions. These bounds can not

Algorithm 2 Branch and Bound algorithm.

1: **function** BnB ($\mathcal{U}, \{w_i\}, \mathcal{V}, \mathcal{S}, v_0, \mathcal{E}, \{c_{jk}\}, L$)
2: $X^{root} \leftarrow \{v_0\}$ ▷ Root node
3: $X^* \leftarrow X^{root}$ ▷ Best solution so far
4: $LB^* \leftarrow \bar{W}(X^{root})$ ▷ Current lower bound
5: $PriorityQueue$.INSERT($root$)
6: **repeat**
7: ▷ **Select unexplored leaf node**
8: $n \leftarrow PriorityQueue$.PULLMAXNODE
9: ▷ **Evaluate bounds for selected node**
10: $X^n \leftarrow$ GREEDY $(\mathcal{U}, \{w_i\}, \mathcal{V}, \mathcal{S}, V^n, \mathcal{E}, \{c_{jk}\}, L^n)$
11: $LB^n \leftarrow \bar{W}(X^n)$
12: **if** $LB^n > LB^*$ **then**
13: $X^* \leftarrow X^n$
14: $LB^* \leftarrow LB^n$
15: $\hat{V}^n \leftarrow$ UB $(V^n, \mathcal{V}, \mathcal{S}, v^n, L^n)$
16: $UB^n \leftarrow \bar{W}(\hat{V}^n)$
17: ▷ **Back-propagate bounds and prune**
18: $p \leftarrow n$
19: **while** $p \neq \emptyset$ **do**
20: **if** $LB^n > LB^p$ **then**
21: $LB^p \leftarrow LB^n$
22: **if** $c = evaluated,\ \forall c \in children(p)$ **then**
23: $UB^p \leftarrow \max_c(UB^c)$
24: **if** $UB^p < LB^*$ **then**
25: PRUNE(p)
26: $p \leftarrow parent(p)$
27: ▷ **Add children within budget**
28: **for each** $v \in \mathcal{S} \setminus \{V^n\}$ **do**
29: $L' \leftarrow L^n - c\langle v^n, v\rangle$
30: **if** $L' \geq 0$ **then**
31: $c \leftarrow n$.ADDCHILD$(\{V^n \cup v\}, v, L')$
32: $PriorityQueue$.INSERT(c)
33: **until** $LB^* = UB^{root}$ or computation budget
34: **return** X^*, LB^*, UB^{root}

only inform a user about whether it is worth continuing the search, but are also used directly by the algorithm to prune suboptimal sub-trees from the search space and therefore improve efficiency. Pseudocode is listed in Alg. 2 and we describe the algorithm as follows.

Tree expansion

At each iteration, an unexplored leaf node n is pulled from a priority queue (line 8), evaluated and added to the tree. The order in which nodes are added, i.e., the expansion policy, is determined by the keys in the priority queue. The most suitable choice for the keys is problem dependent; in Sec. 6.2 we empirically analyse several possible expansion policies.

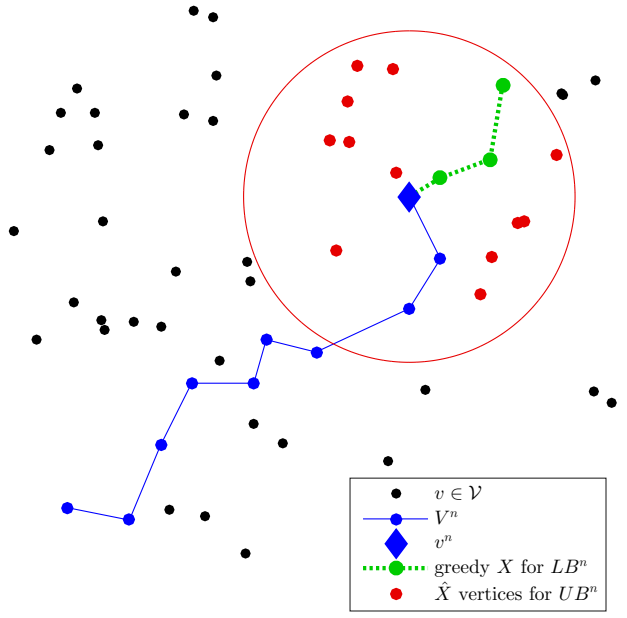


Figure 2: Example bounds calculation for node n with partial solution path V^n and current location v^n. The greedy path (green) is used as the lower bound LB^n. The weighted set cover of the union of V^n with all vertices within budget (red, and including the greedy selection) gives the upper bound UB^n.

Tree nodes

A node n represents a feasible partial solution V^n. For node n, a lower bound LB^n and upper bound UB^n is computed for the reward $\bar{W}$ of the best feasible solution that begins with the partial solution V^n. These bounds are first evaluated when the node is added (lines 10-16) and later refined as its descendants are expanded (lines 20-23). We calculate LB^n using the adapted set cover greedy algorithm and UB^n using the union of all reachable vertices subject to the travel budget. These bounds are illustrated in Fig. 2 and described below.

Lower bounds

In standard BnB algorithms, the lower bound typically provides a guarantee on solution quality for *all* solutions that are descendants of a node. We instead use a tighter bound that provides a guarantee for the *best* solution that is a descendant of a node. This is desirable since it allows more aggressive pruning while guaranteeing not to prune the optimal solution. For this purpose, we calculate the lower bound LB^n using the Alg. 1 greedy algorithm from Sec. 4.1. Observe that the optimal solution relative to a partial solution V^n is guaranteed to have a value $\bar{W}$ greater than or equal to the greedy solution, and thus is a feasible LB^n. Despite the inapproximability of

Algorithm 3 Upper Bound used by BnB.

```
1: function UB ( V̂, 𝒱, 𝒮, v, L )
2:     for each v_j ∈ 𝒱 do
3:         if c⟨v, v_j⟩ ≤ L then
4:             V̂ ← V̂ ∪ {S_j}
5:     return V̂
```

greedy (Sec. 4.2), this approach will typically perform reasonably well in practice, and therefore is suitable.

Upper bounds

The upper bound UB^n is computed according to Alg. 3. This computation considers the set of vertices $\hat{V}$ defined as the union of all vertices in the partial solution V^n and all vertices reachable within the remaining budget L' from the previous selection v^n. The upper bound UB^n is computed as the weighted set cover $\bar{W}$ of the subsets associated with $\hat{V}^n$. Since the set weights are positive, the objective function $\bar{W}$ is monotone, and therefore all descendants of n are guaranteed to have a $\bar{W}$ less than or equal to UB^n. Although this bound may be weak in some cases (e.g., when L' is large), it is difficult to efficiently find tighter bounds without further assumptions.

Back-propagation

Refinement of the bounds is performed while back-propagating from the current node n back to the root node (lines 18-26). This refinement is useful, although not strictly necessary, when the priority queue keys (i.e., the expansion policy) are a function of the bounds. The lower bound LB^p of a parent node p is maintained as the maximum of the previously calculated LB^p and the lower bounds LB^c of its children c (lines 21). Once all children c have been evaluated, the upper bound UB^p is maintained as the maximum UB^c of its children (line 23).

Pruning

Pruning of the search tree is performed by considering the bounds at each node. Pruning is desirable since this effectively reduces the size of the search space while guaranteeing optimality. This is achieved by firstly remembering the best lower bound LB^* achieved so far by any node (line 14). A node p is then pruned if its upper bound UB^p is less than LB^* (line 24-25). Once pruned, a node and its descendants will never be visited again. This is reasonable since the bounds guarantee that any descendent is not an optimal solution. If the current node n was not immediately pruned, then all feasible children of n that meet the budget constraint are inserted into the priority queue (lines 28-32).

Termination criteria

The optimal solution is guaranteed to have been found if the lower bound is equal to the upper bound (line 33). The tree may continue growing until this termination criteria is reached. The algorithm is any-time, and therefore may be halted early if a computation budget is exhausted, and return the best found solution and its approximation bounds. The first node added to the tree computes the greedy solution and therefore BnB is guaranteed to give a solution as good as or better than greedy.

6 Empirical Analysis

In this section we investigate the behaviour of the algorithm with simulated random environments. In particular, we analyse the rate of convergence of the bounds, compare to greedy, and study a variety of expansion policies. The first experiments involve random environments, while the second experiments simulate a probabilistic range-sensing model in an occluded environment.

6.1 Convergence Towards Optimal

We begin by analysing the convergence of the bounds to the optimal solution for random problem instances where an exhaustive search is feasible.

Experimental setup

The random environments consist of 12 vertices placed uniform randomly in a continuous rectangular environment, with Euclidean distance edge costs. There are 100 elements with weights assigned uniform randomly between 1 and 10. The probability $p(u, S_i)$ of each element being an element of each subset is drawn from a beta distribution with $\alpha = 1$ and $\beta = 5$ and therefore has a mean probability of 0.16. The robot starts in the centre of the environment and has a travel budget equal to the width of the environment. On average, the optimal solution is 70 % of the maximum possible coverage if the budget constraints were removed. These experiments were performed using the max-UB expansion policy introduced later in Sec. 6.2. The average runtime for each instance was 500 ms when performed on a single core of a standard desktop computer with an Intel i7 processor.

Results

Figure 3 shows the convergence of the lower and upper bounds towards the optimal solution for 50 random environments. The LB generally converged faster towards the optimal than the UB, which indicates the LB is a stronger heuristic for this problem. The number of iterations before the optimal solution was found was on average 27 % of the number of iterations before optimality was guaranteed. The average number of nodes visited by the search was 15 % of the exhaustive search tree. The greedy algorithm (i.e., the first iteration of BnB) found the optimal solution in 28 % of the instances. However, as discussed in Sec. 4.2, greedy can perform arbitrarily poorly; the worst case for these problem instances achieved a weight 25% worse than optimal.

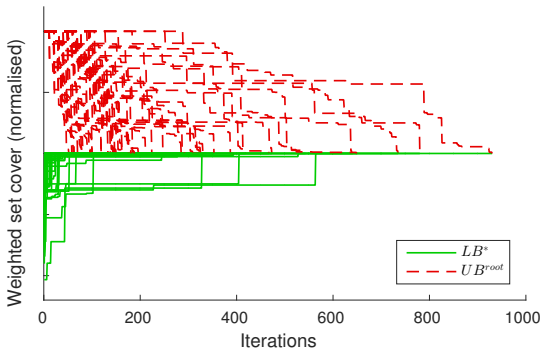


Figure 3: Convergence of the bounds for 50 environments. The LB generally converges faster towards optimal than UB. Vertical axis is normalised for each instance to give a constant $\bar{W}(X^*)$ and initial UB^{root}.

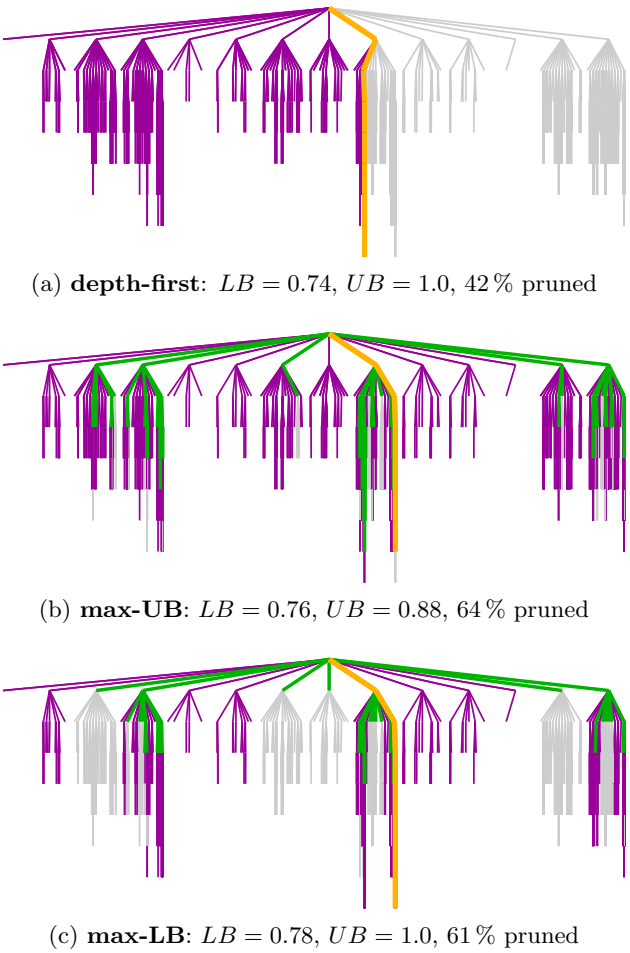
6.2 Expansion Policies

The expansion policy is dictated by the choice of keys in the priority queue. At each iteration, the unexplored node in the priority queue with the largest key is added to the tree (Alg. 2 line 9). In these experiments, we evaluate and compare the following expansion policies.

- **breadth-first -** Perform a breadth-first traversal.
- **depth-first -** Perform a depth-first traversal.
- **max-UB -** Select an unexplored node c whose parent p has the largest UB^p.
- **max-LB -** Select an unexplored node c whose parent p has the largest LB^p.
- **max-mean -** Select an unexplored node c whose parent p has the largest $(UB^p + LB^p)/2$.
- **max-bound -** Select an unexplored node c whose parent p has the largest $UB^p - LB^p$.
- **min-bound -** Select an unexplored node c whose parent p has the smallest $UB^p - LB^p$.
- **random -** Select a random unexplored node.

Experimental setup

We perform experiments with a more complex environment and realistic observation model. The environments consist of 50 vertices, 50 small random walls and 100 random points representing elements. The observation model is range-dependent and occluded by walls. Each element-vertex pair has observation probability $p(u, S_i) = 0$ if the connecting line segment intersects a wall. When there is no intersection, then $p(u, S_i)$ has high probability at close distances, and low probability at far distances. More specifically, $p(u, S_i)$ is defined as a zero-mean Gaussian function of the element-vertex distance, with random variance, and normalised to 1 when the distance is zero. Edge costs are defined as Euclidean distance plus a constant, and the robot moves at 50 % speed if it passes through a wall. For these experiments, the search continued until 15 % of the full tree had been visited, which took an average of 10 s per instance.



(a) **depth-first**: $LB = 0.74$, $UB = 1.0$, 42 % pruned

(b) **max-UB**: $LB = 0.76$, $UB = 0.88$, 64 % pruned

(c) **max-LB**: $LB = 0.78$, $UB = 1.0$, 61 % pruned

Figure 4: Illustration of search trees expanded by different expansion policies after 1000 iterations for a small problem instance. Gold path is best solution found so far; green is current search tree; purple is pruned search space; gray has not yet been explored or pruned.

Results

Figure 4 shows a snapshot of the search tree for 3 different policies for a smaller problem instance after 1000 iterations (25 % of the full tree). The depth-first policy (a) searches arbitrarily from left to right and has pruned 42 % of the space, but the UB has not been refined. The max-UB policy (b) selects nodes that are currently contributing most to the UB. This results in a tighter UB than (a) and more pruning of the search space. The max-LB policy (c) instead focuses on improving LB, and it achieved the highest LB out of these three policies. However, the UB had not yet been refined since some sub-trees have not been explored beyond depth 1.

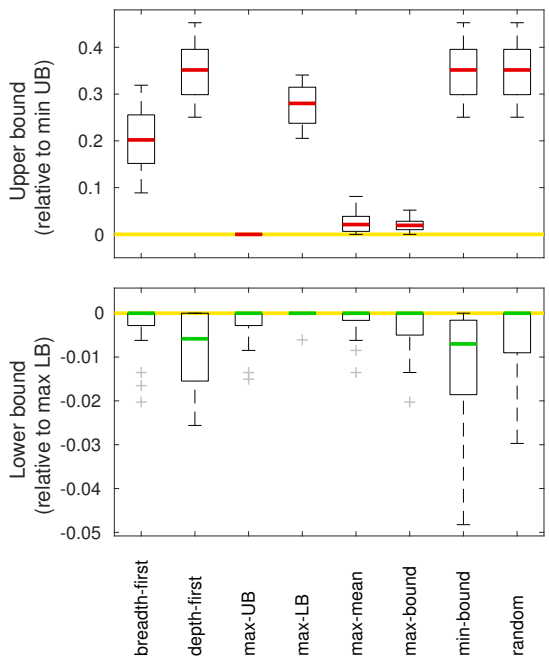


Figure 5: Comparison of the bounds achieved by different expansion policies after exploring 15 % of the search space. Ideally we want the upper bound to be as low as possible, and the lower bound to be as high as possible. The bounds shown are measured relative to the best bounds achieved in each instance (gold line).

Figure 5 shows the bounds achieved by 8 policies for 20 instances. The bounds shown were normalised by subtracting the best bounds achieved by any of the policies in each instance. Similar trends are seen to the Fig. 4 illustration: the best UB was usually achieved by max-UB, while the best LB was usually achieved by max-LB. The max-mean and max-bound policies achieved a balance between a reasonable LB and UB. The min-bound policy performed poorly since it focuses the computation on areas of the tree that already have the tightest bounds. The random, breadth-first and depth-first policies explore nodes in an arbitrary order and thus generally achieved poorer results. The depth-first policy is known to work well for other problems [Mehlhorn and Sanders, 2008]; however, our greedy LB heuristic is effectively already performing a single depth-first rollout during each iteration and is directly leveraged by the other expansion policies that consider LB and UB.

7 Conclusions and Future Work

We have formulated a generalisation of the set cover problem and a branch and bound solution algorithm for informative path planning applications. In future work we wish to investigate other suitable algorithms such as Monte Carlo tree search [Best *et al.*, 2016a]. It would also be interesting to consider further generalisations such as multi-robot scenarios, time-varying extensions and adaptive replanning, as well as generalisations motivated by set cover variants, such as minimum entropy set cover [Halperin and Karp, 2005] and set multicover [Berman *et al.*, 2007]. Additional future work is to solve specific applications formulated as instances of our problem, such as area exploration, active object recognition and precision agriculture.

Appendix

Reduction from Maximum Set Cover

Any instance of the maximum set cover problem can be transformed into an instance of the proposed generalised problem as follows. Create a complete graph with one vertex associated with each subset. These subsets are defined deterministically and therefore this is a special case where $p\left(u, S_i\right) \in \{0, 1\}, \forall u, S_i$. Let all incoming edges into a vertex have a cost equal to the cost of the associated subset. Create a single start vertex with zero-cost outgoing edges and whose associated subset is the empty set. Every candidate solution path in the generalised form can be transformed to the set cover form by selecting all subsets that are associated with the vertices along the path. The path will have the same cost and set cover as in the original form. Conversely, each candidate solution in the set cover form is equivalent to all paths through the graph that visit the vertices associated with the selected subsets. These paths have the same cost and set cover as in the original form. Therefore any problem instance of the maximum set cover problem can be solved using a reduction to the proposed generalised problem.

References

[Berman *et al.*, 2007] P. Berman, B. DasGupta, and E. Sontag. Randomized approximation algorithms for set multicover problems with applications to reverse engineering of protein and gene networks. *Discrete Appl. Math.*, 155(67):733–749, 2007.

[Best *et al.*, 2015] G. Best, W. Martens, and R. Fitch. A spatiotemporal optimal stopping problem for mission monitoring with stationary viewpoints. In *Proc. of Robotics: Science and Systems*, 2015.

[Best *et al.*, 2016a] G. Best, O.M. Cliff, T. Patten, R.R. Mettu, and R. Fitch. Decentralised Monte Carlo tree search for active perception. In *Proc. of WAFR*, 2016.

[Best *et al.*, 2016b] G. Best, J. Faigl, and R. Fitch. Multi-robot path planning for budgeted active perception with self-organising maps. In *Proc. of IEEE/RSJ IROS*, 2016.

[Binney and Sukhatme, 2012] J. Binney and G. Sukhatme. Branch and bound for informative path planning. In *Proc. of IEEE ICRA*, pages 2147–2154, 2012.

[Chekuri and Pal, 2005] C. Chekuri and M. Pal. A recursive greedy algorithm for walks in directed graphs. In *Proc. of IEEE FOCS*, pages 245–253, 2005.

[Chvatal, 1979] V. Chvatal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4(3):233–235, 1979.

[Cliff *et al.*, 2015] O.M. Cliff, R. Fitch, S. Sukkarieh, D.L. Saunders, and R. Heinsohn. Online localization of radio-tagged wildlife with an autonomous aerial robot system. In *Proc. of Robotics: Science and Systems*, 2015.

[Das *et al.*, 2015] J. Das, F. Py, J.B.J. Harvey, J.P. Ryan, A. Gellene, R. Graham, D.A. Caron, K. Rajan, and G.S. Sukhatme. Data-driven robotic sampling for marine ecosystem monitoring. *Int. J. Robot. Res.*, 34(12):1435–1452, 2015.

[Dornhege *et al.*, 2016] C. Dornhege, A. Kleiner, A. Hertle, and A. Kolling. Multirobot coverage search in three dimensions. *J. Field Robot.*, 33(4):537–558, 2016.

[Faigl and Hollinger, 2014] J. Faigl and G.A. Hollinger. Unifying multi-goal path planning for autonomous data collection. In *Proc. of IEEE/RSJ IROS*, pages 2937–2942, 2014.

[Faigl *et al.*, 2016] J. Faigl, R. Pěnička, and G. Best. Self-organizing map-based solution for the orienteering problem with neighborhoods. In *Proc. of IEEE SMC*, 2016.

[Garg and Ayanian, 2014] S. Garg and N. Ayanian. Persistent monitoring of stochastic spatio-temporal phenomena with a small team of robots. In *Proc. of Robotics: Science and Systems*, 2014.

[Geem *et al.*, 2005] Z.W. Geem, C.-L. Tseng, and Y. Park. Harmony search for generalized orienteering problem: Best touring in china. In *Proc. of ICNC*, pages 741–750, 2005.

[Gunawan *et al.*, 2016] A. Gunawan, H.C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *Eur. J. Oper. Res.*, 255(2):315 – 332, 2016.

[Halperin and Karp, 2005] E. Halperin and R.M. Karp. The minimum-entropy set cover problem. *Theoretical Computer Science*, 348(23):240–250, 2005.

[Hochbaum, 1997] D.S. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., Boston, MA, USA, 1997.

[Hollinger and Sukhatme, 2014] G.A. Hollinger and G.S. Sukhatme. Sampling-based robotic information gathering algorithms. *Int. J. Robot. Res.*, 2014.

[Hollinger *et al.*, 2011] G.A. Hollinger, U. Mitra, and G.S. Sukhatme. Active classification: Theory and application to underwater inspection. In *Proc. of ISRR*, 2011.

[Hollinger *et al.*, 2012] G.A. Hollinger, B. Englot, F.S. Hover, U. Mitra, and G.S. Sukhatme. Active planning for underwater inspection and the benefit of adaptivity. *Int. J. Robot. Res.*, 2012.

[Khuller *et al.*, 1999] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inform. Process. Lett.*, 70(1):39–45, 1999.

[Kim and Eustice, 2015] A. Kim and R.M. Eustice. Active visual slam for robotic area coverage: Theory and experiment. *Int. J. Robot. Res.*, 34(4-5):457–475, 2015.

[Krause *et al.*, 2008] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, 2008.

[Lee *et al.*, 2014] J.J.H. Lee, K. Frey, R.C. Fitch, and S. Sukkarieh. Fast path planning for precision weeding. In *Proc. of ARAA ACRA*, 2014.

[Lim *et al.*, 2016] Z.W. Lim, D. Hsu, and W.S. Lee. Adaptive informative path planning in metric spaces. *Int. J. Robot. Res.*, 35(5):585–598, 2016.

[Mathew *et al.*, 2015] N. Mathew, S.L. Smith, and S.L. Waslander. Multirobot rendezvous planning for recharging in persistent tasks. *IEEE Trans. Robot.*, 31(1):128–142, 2015.

[Mehlhorn and Sanders, 2008] K. Mehlhorn and P. Sanders. *Algorithms and Data Structures*, chapter Generic Approaches to Optimization, pages 233–262. Springer, 2008.

[Meliou *et al.*, 2007] A. Meliou, A. Krause, C. Guestrin, and J.M. Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *Proc. of AAAI*, pages 602–607, 2007.

[Noon and Bean, 1989] C.E. Noon and J.C. Bean. An efficient transformation of the generalized traveling salesman problem. Technical Report 89-36, Department of Industrial and Operations Engineering, University of Michigan, 1989.

[Patten *et al.*, 2016] T. Patten, M. Zillich, R. Fitch, M. Vincze, and S. Sukkarieh. Viewpoint evaluation for online 3-D active object classification. *IEEE Robot. Autom. Lett.*, 1(1):73–81, 2016.

[Singh *et al.*, 2009] A. Singh, A. Krause, C. Guestrin, and W.J. Kaiser. Efficient informative sensing using multiple robots. *J. Artif. Intell. Res.*, 34(2):707, 2009.

[Vazirani, 2001] V.V. Vazirani. *Approximation algorithms*. Springer, 2001.