Exact and Heuristic Algorithms for Risk-Aware Stochastic Physical Search

DANIEL S. BROWN, JEFFREY HUDACK, NATHANIEL GEMELLI
Air Force Research Laboratories, Rome, NY

BIKRAMJIT BANERJEE
University of Southern Mississippi, Hattiesburg, MS

We consider an intelligent agent seeking to obtain an item from one of several physical locations, where the cost to obtain the item at each location is stochastic. We study Risk-Aware Stochastic Physical Search (RA-SPS), where both the cost to travel and the cost to obtain the item are taken from the same budget, and where the objective is to maximize the probability of success while minimizing the required budget. This type of problem models many task planning scenarios, such as space exploration, shopping, or surveillance. In these types of scenarios, the actual cost of completing an objective at a location may only be revealed when an agent physically arrives at the location, and the agent may need to use a single resource to both search for and acquire the item of interest. We present exact and heuristic algorithms for solving RA-SPS problems on complete metric graphs. We first formulate the problem as mixed integer linear programming problem. We then develop custom branch and bound algorithms which result in a dramatic reduction in computation time. Using these algorithms, we generate empirical insights into the hardness landscape of the RA-SPS problem, and compare the performance of several heuristics.

Key words: Risk-aware; stochastic physical search; planning under uncertainty.

1. INTRODUCTION

Intelligent agents operating in uncertain physical environments often must expend resources to obtain an item of interest. In certain domains, the cost to both search for and acquire the item is consumed from the same resource budget. When uncertainty about the location and/or cost of an item of interest exists, agents must carefully plan their search to maximize the probability of successfully obtaining the item of interest while minimizing travel and acquisition costs. Shoppers may need to pay for gas or public transportation in order to visit several stores to find a commodity they can afford. An autonomous Mars rover searching for a specific mineral may want to plan a search path that visits several mineral deposits to maximize the probability of successfully obtaining the sample while minimizing battery usage. A search-and-rescue plane looking for debris from a missing airliner at various locations will want to optimize its chance of success given limited fuel while minimizing the risk of adverse weather. Additionally, intelligence, surveillance, and reconnaissance (ISR) missions may need to operate in regions with limited information, such as satellite images, and uncertainty regarding the amount of time and effort needed to gather intelligence at each location.

These are all examples of stochastic physical search problems, as introduced by Aumann et al. (2008). Aumann et al. define a stochastic physical search problem as the search for an item item of interest over a set of sites. The travel costs between sites are known; however, the actual cost of completing an objective at each site is unknown, but modeled with a probability distribution. This probabilistic cost to obtain the item of interest may reflect information such as prior experience, the dynamics of the environment, or the influence of unknown factors. Because this uncertainty can only be removed through actually visiting locations, an agent

Email addresses: {daniel.brown.81, jeffrey.hudack, nathaniel.gemelli }@us.af.mil, bikramjit.banerjee@usm.edu

must plan its exploration wisely. Our work develops the first exact and heuristic algorithms for solving these types of physical search problems on complete metric graphs.

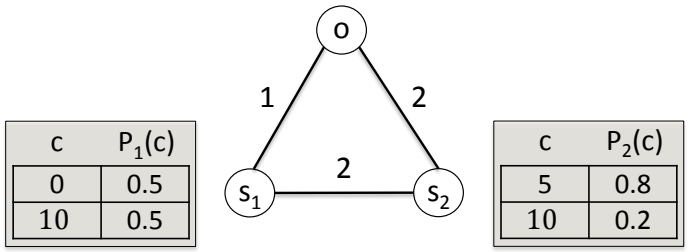
Previous work by Kang et al. (2011) developed exact and heuristic solutions for physical search problems where the objective is to minimize *expected cost*. Expected cost solutions work well for environments where we seek to meet objectives on a recurring basis and is often a reasonable objective since the law of large numbers guarantees that the average performance over many independent executions will converge to this expectation. However, in many instances, plans can only be executed once, either because they take a long time to execute, because we cannot restart if we fail, or because problem instances are always changing and are not independent and identically distributed (Moldovan and Abbeel, 2012). Especially for mission-critical applications where we may only get one chance to succeed, it is important to generate solutions that accommodate for limited resources and an acceptable risk of failure. Additionally, when there is a chance that the item may be unavailable, then the expected cost formulation becomes undefined.

Rather than minimizing expected cost, we focus on problems where there is limited budget and a chance of failure. We call these problems *Risk-Aware Stochastic Physical Search* (RA-SPS) because it takes into account risks of failure when planning. We provide solutions to two dual RA-SPS problems. Given a fixed budget to work with, we may seek to find a *Max-Probability* path that maximizes the probability of success. Alternatively, we may have some threshold on the risk of failure, and seek a *Min-Budget* path that minimizes the budget required to meet that threshold.

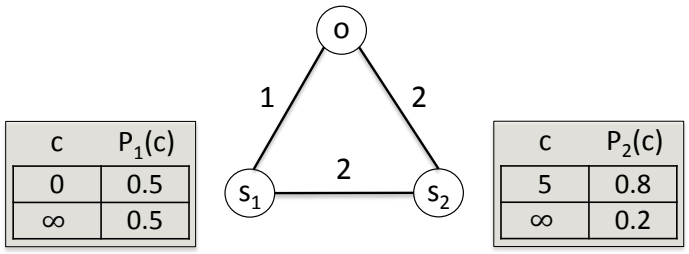
As a motivating example, consider a Mars rover searching for a geological sample. One common goal for rover subsurface exploration is to plan a search path that will allow the rover to obtain certain geological sample for analysis. Geologic and compositional maps produced from orbital remote sensing data can provide the robot with locations where samples can potentially be obtained along with the costs to travel between sites (Wettergreen et al., 2014). However, the actual cost to obtain the sample at each site is unknown until the rover actually arrives and surveys the site. Rovers typically have a suite of different types of equipment to do geological classification and sampling, with each type of equipment requiring a different amount of time and battery power (Wagstaff et al., 2013; Wettergreen et al., 2014). Based on the orbital sensing analysis and previous experience the probability of needing to use each type of equipment can be estimated, along with the amount of battery power that will be required to use that equipment.

As a simple example, consider Figure 1(a). In this example the rover starts at the origin location o. Orbital sensing and imagery analysis has identified two potential locations for obtaining the sample. At site s_1 is it estimated that there is a 50% chance the sample is readily accessible and can be obtained using a negligible amount of battery power. There is also a 50% chance that the sample requires using a high-powered laser to uncover the sample, consuming 10 units of battery power. Alternatively at site s_2 there is an 80% chance that the sample can be obtained using a simple low-powered drill, which consumes 5 units of battery power, but a 20% chance that it will require using the high-powered laser. Based on the distances between sites, the rover knows it will cost approximately 1 unit of battery to travel to s_1, 2 units to travel to s_2, and 2 units to travel between site s_1 and s_2.

Figure 1(a) is an example of how an expected cost solution differs from a risk-aware solution. The minimum expected cost solution is to travel from o to s_1 to s_2 resulting in an expected cost of 5. The expected cost can be calculated as follows. Travel from o to s_1 costs 1. At site s_1 there is a 0.5 probability that the item can be obtained for 0 cost, but there is also a 0.5 probability that the cost is 10. If the cost is 10, then it is better to spend 2 budget to travel to site s_2 where there is an expected cost of 6. Thus the total expected cost of path $\langle o, s_1, s_2 \rangle$ is $1 + 0.5 \cdot 0 + 0.5 \cdot (\min(10, 2 + 6)) = 5$. However, the expected cost solution assumes that an average case analysis is desired and that there is no limit on budget.



(a)



(b)

FIGURE 1. (a) An example stochastic physical search problem, where visiting s_1 minimizes expected cost, but visiting s_2 maximizes the probability of success (minimizes risk of failure). (b) When infinite costs are possible, expected cost solutions are undefined.

If, on the other hand, there is a limited budget of 7, and the search will only be performed once, the path $\langle o, s_1, s_2 \rangle$ has a probability of success of only 0.5 since there is a 0.5 chance of getting the item at s_1 for a cost of 0 and after spending 3 budget to travel to s_2 the agent will not have enough to purchase the item at s_2 for any of the possible prices. The optimal risk-aware solution is to first visit s_2 and then visit s_1. The path $\langle o, s_2, s_1 \rangle$ results in a 0.9 probability of successfully obtaining the item. To calculate the probability of success of this path it is easier to calculate one minus the probability of failure at every site along the path. Thus, traveling to s_2 spends 2 budget on travel, resulting in a probability of failure of 0.2 since the agent only has 5 budget available for obtaining the item. However, if the item is not available for a cost of 5, the agent can travel next to s_1 which uses 2 budget leaving a remainder of 3. The agent has a probability of failure at s_2 of 0.5 and the total probability of success along path $\langle o, s_2, s_1 \rangle$ is $1 - 0.2 \cdot 0.5 = 0.9$.

The preceding example shows the difference between an expected cost solution and a solution that is risk-aware; the expected cost solution will minimize cost over the long run, while a risk-aware solution is concerned with the actual probability of success for a single execution. Additionally, if, as shown in Figure 1(b), the cost of 10 is replaced by ∞ (representing a possibility that the sample cannot be obtained), then an expected cost is not defined—further motivating our focus on risk-aware solutions that take into account the probability of failure and limited budget.

Our work extends recent work by Hazon et al. (2013), that analyzed and solved Max-Probability and Min-Budget problems where sites are located along a path. We provide algorithms for stochastic physical search that take into account risk and the probability

of failure and work on any complete metric graph. Additionally, our work complements the work of Kang and Ouyang (2011) by providing risk-aware (as opposed to minimum expected costs) solutions. Some of the ideas contained in this work were initially published in a preliminary workshop paper (Brown et al., 2015). This work significantly extends, synthesizes, and enhances our previous work.

Our main contributions are: (1) A mixed-integer linear programming formulation of the Min-Budget and Max-Probability RA-SPS problems on complete metric graphs, (2) Provably correct Branch-and-bound algorithms that find exact solutions much faster than state-of-the-art MILP solvers, (3) An empirical investigation of the hardness landscape of RA-SPS problems for a large number of synthetic problems of varying size and complexity, and (4) An analysis of both a greedy and randomized local search heuristic which enable solutions that are close to optimal when exact solutions become intractable.

The remainder of the paper is as follows. In the next section, we formalize the RA-SPS problem and discuss some interesting properties of its solutions. Section 3 presents a review of related literature. Section 4 presents a MILP formulation of the RA-SPS problem. Section 5 describes our customized branch-and-bound solution. Section 6 presents a series of numerical experiments and sensitivity analysis for solving the problem using an off-the-shelf MILP solver and our customized branch-and-bound solver. Section 7 proposes several simple heuristics for solving RA-SPS and empirically studies their performance. We end with a discussion, conclusions, and suggestions for future work.

2. PROBLEM FORMULATION

A Risk-Aware Stochastic Physical Search problem is defined by a graph $G(S^+, E)$ with a set of sites $S^+ = S \cup \{o\}$ where $S = \{s_1, ..., s_m\}$ is the set of m sites where the single item of interest may possibly be obtained, o is the origin location, and $E \subseteq S^+ \times S^+$ is the set of edges. We define $adj(i) = \{j : (i, j) \in E\}$. We assume that each $(i, j) \in E$ has a non-negative cost of travel t_{ij} that is deterministic and known. An agent must start at origin site o, and visit a subset of points in S to obtain the item. We assume that our graphs are complete metric graphs, thus an optimal solution needs to only visit each site at most once.

The cost of obtaining the item at each site $s_i \in S$ is a random variable C_i with an associated probability mass function $P_i(c)$, which gives the probability that the item will cost c at site s_i. We assume that the actual cost is not revealed until the agent visits the site and that the cost remains fixed thereafter. We further assume that there are a finite number of possible costs in the support of $P_i(c)$, $\forall i \in S$.

We examine two different objectives for the RA-SPS problem

- **Min-Budget**: given a required probability of success p^*_{succ}, minimize the budget necessary to ensure the item can be obtained with probability at least p^*_{succ}.
- **Max-Probability**: maximize the probability of obtaining the item while ensuring the sum of travel and item costs do not exceed an initial budget, B^*.

In both cases, given a path $\langle v_0, v_1, \ldots, v_k \rangle$ with $v_i \in S^+$ and $v_0 = o$, the probability of success is $p_{succ} = 1 - \prod_i Pr(\text{failure given budget } b_i \text{ at } v_i)$ and $b_{i+1} = b_i - t_{v_i, v_{i+1}}$ $(i = 0, \ldots, k-1)$ with b_0 equal to the starting budget B^*.

Both Min-Budget and Max-Probability have an associated family of optimal solutions parameterized by p^*_{succ} and B^*, respectively. Figure 2 shows a simple example of an SPS problem instance that illustrates this property. As the starting budget is increased the optimal sites to visit (starting at s) changes. Similarly, changing the required probability of success changes the optimal path. The box in the upper right shows the full Pareto front formed for this problem.

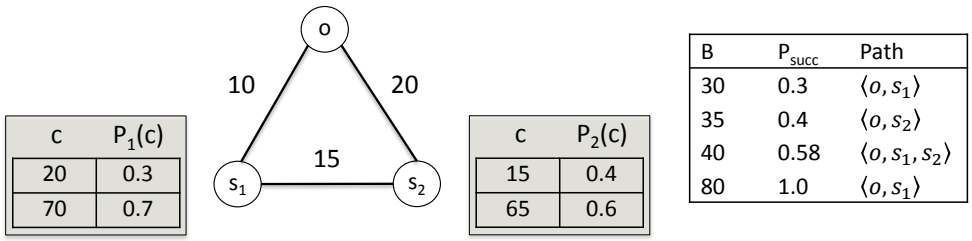


FIGURE 2. A simple instance of a 2 site SPS problem with the agent starting at location s. For each site, s_i, there is a probability $P_i(c)$ of purchasing the item at cost c. The table (upper right) indicates the budget thresholds at which the agent can achieve the corresponding probability of success, p_{succ}, via the path.

3. RELATED WORK

Our work is related to the areas of economic search theory, traveling salesman-like problems, path planning, and planning and scheduling. We briefly highlight the relevant work in each of these areas and how it relates to RA-SPS. We finish this section by describing recent work on the specific problem of stochastic physical search. We note that much of this related work is based on the literature review in (Hazon et al., 2013).

3.1. Economic Search Theory

There is a significant and varied body of work relating to search theory (Aumann and Hart, 1994) and sequential exploration (Lippman and McCall, 1976) in the operations research and economic communities, including the famous secretary problem (Ferguson, 1989) and Pandora's problem (Weitzman, 1979). In many cases there is an optimal stopping criterion for search based on some reservation price (Rochlin and Sarne, 2013). However, research and algorithms for economic-based search assume that the cost associated with observing a given opportunity does not change along the search process. This work does not capture critical aspects of physical search, where physically changing an agents position changes the travel distance (and corresponding travel cost) to explore other sites, and where the probability to obtain the item is dependent on the search path taken (e.g. a longer search path uses more budget which reduces the budget available to obtain the item).

3.2. TSP extensions

Search over physical spaces has been extensively studied in the fields of computational intelligence and operations research in the contexts of several extensions of the well-known Traveling Salesman Problem (TSP) (Gutin and Punnen, 2002; Laporte and Martín, 2007).

The Traveling Salesman Problem with Profits (Jozefowiez et al., 2008) is a generalization of the TSP where a profit is obtained when a site is visited. Performance is generally measured in terms of two competing objectives: (1) minimize tour length and (2) maximize the total profit. These objectives have been combined in several ways, resulting in several different types of problems. The Profitable Tour Problem (Dell'Amico et al., 1995) seeks to maximize the collected profit minus the tour length. The Orienteering Problem (Vansteenwegen et al., 2011) seeks to maximize total profit collected while keeping the tour length under some threshold. This problem has also been referred to as the selective traveling salesman problem (Laporte and Martello, 1990) and the maximum collection problem (Kataoka,

1988). Finally, the Quota Traveling Salesman Problem (Awerbuch et al., 1998; Ausiello et al., 2004) seeks to minimize the tour length while guaranteeing a minimum profit. This problem is also called the prize-collection traveling salesman (Balas, 1989; Ausiello et al., 2008). In each of the above variants, travel budget and prizes are distinct, with currencies that are not interchangeable. Therefore, the budget for travel does not affect the prize collected at a node. In contrast, in our RA-SPS problem formulation, exploration and fulfilling an objective use the same resource, and it is possible for prolonged exploration to make it impossible to achieve the objective due to exhausting the budget.

The Canadian Traveller Problem (CTP) is a generalization of the shortest path problem to problems where the graph is only partially observable and needs to be explored (Papadimitriou and Yannakakis, 1989; Nikolova and Karger, 2008). This problem has also been studied in Operations Research as the Shortest Path Problem with Recourse where each edge is associated with a probability of being in the graph (Polychronopoulos and Tsitsiklis, 1996). In the CTP the emphasis is on minimizing the competitive ratio in the worst case and in the SSPPR the emphasis is on minimizing the expected cost. Rather than having uncertainty on the edges of the graph, our problem has a known graph structure but unknown costs at each of the nodes of the graph. Also our problem takes both a limited resource and the actual probability of success into account, while both the CTP and the SSPPR have only one objective, minimize the competitive ratio or expected cost, respectively.

3.3. Path Planning

Related to graph search, path planning is concerned with finding paths through the graph that minimize cost. One form of uncertainty is restricted knowledge of which paths exist and their cost. (Nikolova et al., 2006) A common and well-studied heuristic for planning paths is the A* algorithm, and its associated dynamic variants, D* and D* lite (Stentz, 1995; Koenig and Likhachev, 2005). While the dynamic versions allow the goal location to change, because they do not consider uncertain values or costs at the possible goal locations, these algorithms cannot be used to solve our problem. Stern et al. (2014) investigate extensions of A* to solve bounded-cost search problems which are similar to the Max-Probability variant in that they try to find a solution given a limited starting budget.

3.4. Planning and Scheduling

Our work is similar to several problems in planning and scheduling. One similar problem is resource constrained planning, where an agent has actions that consume resources and the agent is required to reach the goal using some initial amount of resources (Nakhost et al., 2010). Researchers have also studied the closely related problem of probabilistic planning with no observability (Kushmerick et al., 1995). In this setting, there is an initial belief state, in the form of a probability distribution, over the set of world states, as well as sets of probabilistic actions and goals. In particular, our work is very similar to the research on conformant probabilistic planning (CPP) problems (Domshlak and Hoffmann, 2007; Brafman and Taig, 2011; Taig and I Brafman, 2013) which can contain failure states. CPP is the task of generating a sequence of actions to achieve a goal without sensing. Thus, the goal is to find a sequence of actions in the belief state space that results from the uncertainty of initial states and action effects and can be formulated in two different ways: (1) find a plan achieving the goal with probability that exceeds a threshold, or (2) fix the time horizon and ask for a plan with maximum probability (Lee et al., 2014; Taig and Brafman, 2014). Conformant probabilistic planning allows for stochastic actions, but much work focuses on deterministic actions (Taig and Brafman, 2014). (Taig and I Brafman, 2013)

reduce probabilities to action costs and reduce CPP into cost bounded planning where the probability of failure is the cost.

The conformant probabilistic planner COMPLAN (Huang, 2006) uses a similar solution approach to our Max-Probability algorithm: depth-first branch and bound with pruning of search nodes if an upper bound on the probability of success of the best plans below a node is not greater than the best plan found so far. The main difference is that COMPLAN uses a fixed horizon length to determine the number of actions allowed, whereas we specify a budget that must be used to move along edges with varying costs and also must be used to obtain the item. We also have a known starting state.

A key difference between our work and previous work on CPP is the interplay between travel costs and item costs. We assume that traveling and purchasing use the same resource. While it may be possible to fold the uncertainty into stochastic actions, the probabilities would need to be dependent on either the time horizon (assuming uniform travel costs) or on the sum of the previous actions in order to account for the fact that using budget to travel decreases the probability that an item can be purchased. Future work should consider whether our problem formulation can be transformed into a CPP problem as this would open up a large number of other solution methods.

In the planning domain, a Markov Decision Process (MDP) is often used to formulate the set of actions at each state and their resulting effects (Boutilier et al., 2011; Little et al., 2005). However, these solutions typically focus only on average case analysis and seek to maximize the expected long term reward. Risk-aware planning for MDPs was proposed as an alternative to optimizing expected value by Moldovan and Abbeel (2012). They use a risk parameter to interpolate between extreme risk aversion and risk ignorance and show that a spread of different policies are possible, but does not consider a budget used to travel and accomplish tasks. Benazera et al. (2005) models a rover planning problem as a hybrid-MDP (with both continuous and discrete variables). They consider bounded resources where the choice of action may depend on current available resources.

There are some similarities to the scheduling problem, or the allocation of resources over time, in the presence of uncertainty. (Werner, 2002; Herroelen and Leus, 2005) In these problems, the underlying deterministic planning problem is inherently NP-hard, and the goal is to find ways of dealing with additional uncertainty. In this work, if all costs are known a priori it is not difficult to find an optimal solution and a trivial solution exists. In scheduling problems, the plan does not have influence on how the uncertain aspects of the problem are determined. In our proposed work, the uncertainty is a function of the strategy, and the order of sites visited has a direct effect on how uncertainty is dispelled over time.

3.5. Previous work on Stochastic Physical Search

This work extends the preliminary analysis of the RA-SPS problem and the mixed-integer linear program (MILP) and branch-and-bound solutions that was first presented in (Brown et al., 2015). Other researchers have looked at the Stochastic Physical Search problem in several different contexts. We summarize the main results below and highlight how our approach extends or complements existing research.

While physical search has long been of interest in the fields of Artificial Intelligence and Operations Research, the concept of a stochastic physical search problem and the Min-Expected-Cost, Min-Budget, and Max-Probability variants were introduced only recently (Aumann et al., 2008; Hazon et al., 2009, 2013). Hazon et al. show that the *Min-Expected-Cost* variant is NP-hard in general metric spaces and that the Min-Budget and Max-Probability variants are NP-Complete on general metric spaces and on trees. They also provide polynomial-time algorithms for solving 1-dimensional variants where the sites are located along a path for both the single and multi-agent cases, but never evaluate these algorithms on any actual

problem instances. Brown et al. (2015) examine the problem of determining how many agents are required for multi-agent stochastic physical search problems. Our work extends and enhances previous work by providing exact solvers for the single-agent Min-Budget and Max-Probability variants on complete metric graphs. We further provide the first empirical insights into the hardness landscapes of the Min-Budget and Max-Probability problem instances.

The Min-Expected-Cost problem variant has recently been solved for complete metric graphs (Kang and Ouyang, 2011). Kang et al. formulate this problem as a Traveling Purchaser Problem (Pearn and Chien, 1998; Laporte and Riera-ledesma, 2002; Teeninga and Volgenant, 2004) with stochastic prices, where an agent must find a minimum-expected-cost path between markets to purchase a set of required commodities. They formulate the problem as a MILP and solve it using dynamic programming. However, this work does not consider the problem of using a limited resource to both travel and complete tasks, assumes a potentially unbounded available budget, and does not consider problems that have a risk of not obtaining the item and thus has no notion of the probability of success. While it seems that adding some probability of infinite costs would provide a way to model failure, it is unclear how to compute expected cost in this case and the existing algorithms will simply return an expected cost of infinity. Our work complements the work of Kang et al. by providing solvers for the Min-Budget and Max-Probability problems which implicitly take into account probabilities of failure and risk.

A special case of the Stochastic Physical Search problem, called the Binary Stochastic Physical Search problem, involves the search for an item which is either available for free (cost of 0) or unavailable (cost of ∞) at each site. Hudack et al. (2015) model an intelligence collection activity as multiobjective Binary SPS problem and provide exact and approximation algorithms for generating nondominated solution sets that can be used by a human decision maker. Our work extends this model to include cost distributions with any number of real-valued costs.

4. MILP FORMULATION

To formulate the problem as an MILP we add a virtual destination node d with no outgoing edges, but which is reachable from every site in S^+ with a travel cost of zero. The destination node is necessary to enforce our flow constraints on the solution path and acts simply as a marker for the end of the search. Let the possible values of the item's cost at site $i \in S$ be $\{c_{i,1}, c_{i,2}, \dots c_{i,\kappa}\}$, given in increasing order. This induces a set of exclusive cost intervals at i, $R_i = \{[c_{i,y}, c_{i,y+1}) \mid y = 0, 1, \dots \kappa\}$, where $c_{i,0} = 0,\ c_{i,\kappa+1} = \infty$. We assume that all sites have κ cost values; if any site i has fewer cost values then it can be augmented with arbitrary dummy cost values c with $P_i(c) = 0$. Specifically, for $i = o, d$, $P_i(c < \infty) = 0$, $P_i(\infty) = 1$, i.e., the item cannot be obtained at o, d.

We formulate both the Min-Budget and Max-Probability problems as mixed-integer linear programming problems. We define x_{ij} as a binary decision variable where $x_{ij} = 1$ means that edge (i, j) is part of the optimal solution, and $x_{ij} = 0$ otherwise. We define two continuous variables b_i, and lp_i for each $i \in S^+$. The variable b_i represents the budget available upon arriving at site i and the variable ℓp_i represents the log probability of failing to obtain the item of interest at site i with budget b_i. Lastly, for each site $i \in S^+$, and for each possible cost interval $r \in R_i$, we define two binary indicator variables α^1_{ir} and α^2_{ir}. Suppose $r = [c_{i,y}, c_{i,y+1})$, for some $0 \leqslant y \leqslant \kappa$. Then $\alpha^1_{ir} = 1$ iff $b_i < c_{i,y}$, and $\alpha^2_{ir} = 1$ iff $b_i \geqslant c_{i,y+1}$. In other words, $\alpha^1_{ir} = \alpha^2_{ir} = 0$ iff $b_i \in [c_{i,y}, c_{i,y+1})$.

If $b_i \in [c_{i,0}, c_{i,1})$, then the agent visits site i but does not have enough to purchase the item at any possible revealed cost. While having a budget in this interval makes the

C	P(C)
3	0.7
10	0.3

(a) Cost profile

Budget interval	$\log(p_{failure})$
$[0, 3)$	0
$[3, 10)$	$\log(0.3)$
$[10, \infty)$	$-\infty$.

(b) Corresponding budget partition

FIGURE 3. A budget interval with with n costs partitions the interval $[0, \infty)$ into $n + 1$ possible budget intervals with an associated log probability of failure.

probability of failure at site i equal to 1, having $b_i \in [c_{i,0}, c_{i,1})$ can be beneficial as it allows an agent to traverse through site i without purchasing the item and then possibly purchase the item for a low cost at a subsequent site.

For each site $i \in S$ and each cost interval $r = [c_{i,y}, c_{i,y+1}) \in R_i$ we define constants p_{ir} as the log probability of failing to obtain the item at site i given that the available budget $b_i \in [c_{i,y}, c_{i,y+1})$. Thus, p_{ir} are computed from the input as $p_{ir} = \log(\sum_{y<x\leqslant\kappa} P_i(c_{i,x}))$. For example, the cost profile shown in Figure 3 partitions the budget space into three intervals with the corresponding log probabilities of failure. In this example there are three possibilities when the rover arrives at the site: (1) the rover's budget (battery power) is in the interval $[0, 3)$ and it cannot obtain the item, (2) the rover's budget is in the interval $[3, 10)$ and it has enough battery power left to obtain the item at the lower cost, but will fail to obtain the item with probability 0.3, and (3) the rover has sufficient battery power to obtain the item for any of the possible costs.

4.1. Min-Budget

The formulation of the Min-Budget problem as a MILP is shown in Figure 4. The objective function (1) is to minimize the starting budget at the origin. Constraints (2) and (3) are the linkage constraints that ensure that every site in S that is entered must be exited and that each site can be visited at most once. Constraints (4) and (5) make sure that the solution path starts at the origin and ends at the destination. Constraint (6) elminates cycles in the path—this constraint is necessary to avoid disconnected path fragments, but results in an exponential number of constraints. Constraints (7) ensure that if the edge (i, j) is in the solution, then the budget at site j is equal to the budget at site i minus the cost to travel from i to j, where M is a sufficiently large constant value. Constraints (8) and (9) ensure that b_i and lp_i can be nonzero only when site i is on the solution path. Constraint (10) ensures that the solution will successfully obtain the item with probability p_{succ}.

The most difficult part of formulating the RA-SPS problem as a MILP is enforcing the conditional constraint

$$\big(c_{i,y} \leqslant b_i < c_{i,y+1}\big) \Rightarrow \big(lp_i = p_{ir}\big)$$

$\forall i \in S, r = [c_{i,y}, c_{i,y+1}) \in R_i$. This constraint ensures that the log probability of failure, lp_i, at site i is set to the correct value based on the available budget b_i. Constraints (11) represent this conditional constraint. To see this, first note that the conditional expression $A \Rightarrow B$ is logically equivalent to $\neg A \vee B$. This gives us the equivalent expression

$$(b_i < c_{i,y}) \vee (c_{i,y+1} \leqslant b_i) \vee (lp_i = p_{ir}). \tag{16}$$

We use the binary variables α^1_{ir} and α^2_{ir} to indicate the truth of the first two clauses, respectively. Figure 5 shows the results corresponding to the possible values of α^1_{ir} and α^2_{ir}. The first four constraints in 11 ensure that α^1_{ir} and α^2_{ir} indicate the correct propositions. The value δ is a small constant required because of the non-inclusive upper bound, $c_{i,y+1}$, in the

$$\textbf{min} \quad b_o \tag{1}$$

subject to

$$\sum_{i \in adj(j)} x_{ij} \; - \sum_{k \in adj(j)} x_{jk} = 0, \;\; \forall j \in S, \tag{2}$$

$$\sum_{i \in adj(j)} x_{ij} \leqslant 1, \;\; \forall j \in S, \tag{3}$$

$$\sum_{i \in adj(o)} x_{oi} = 1, \quad \sum_{i \in adj(d)} x_{id} = 1, \tag{4}$$

$$x_{io} = 0, \quad x_{di} = 0, \;\; \forall i \in S^+, \tag{5}$$

$$\sum_{(i,j) \in W \times W} x_{ij} \leqslant |W| - 1, \;\; \forall W \subseteq S, \tag{6}$$

$$\left.\begin{array}{l} b_i - x_{ij} \cdot t_{ij} \geqslant b_j - (1 - x_{ij}) \cdot M, \\ b_i - x_{ij} \cdot t_{ij} \leqslant b_j + (1 - x_{ij}) \cdot M, \end{array}\right\} \forall i, j \in S^+, \tag{7}$$

$$b_i \leqslant \sum_{j \in adj(i)} x_{ij} \cdot M, \;\; \forall i \in S, \tag{8}$$

$$lp_i \geqslant -\sum_{j \in adj(i)} x_{ij} \cdot M, \;\; \forall i \in S \tag{9}$$

$$\sum_{j \in S} lp_j \leqslant \log(1 - p^*_{succ}) \tag{10}$$

$$\left.\begin{array}{l} b_i - c_{i,y} + M \cdot \alpha^1_{ir} \geqslant 0, \\ b_i - c_{i,y} + \delta - M(1 - \alpha^1_{ir}) \leqslant 0, \\ b_i - c_{i,y+1} + \delta - M \cdot \alpha^2_{ir} \leqslant 0, \\ b_i - c_{i,y+1} + M(1 - \alpha^2_{ir}) \geqslant 0, \\ lp_i - p_{ir} + M \cdot \alpha^2_{ir} \geqslant 0, \\ lp_i - p_{ir} - M \cdot \alpha^1_{ir} \leqslant 0, \end{array}\right\} \begin{array}{l} \forall i \in S, \\ \forall r = [c_{i,y}, c_{i,y+1}) \in R_i, \end{array} \tag{11}$$

$$b_i \geqslant 0, \;\; \forall i \in S^+, \tag{12}$$

$$lp_i \leqslant 0, \;\; \forall i \in S, \tag{13}$$

$$x_{ij} \in \{0, 1\}, \;\; \forall i, j \in S^+, \tag{14}$$

$$\alpha^1_{ir}, \alpha^2_{ir} \in \{0, 1\}, \;\; \forall i \in S, r \in R_i \tag{15}$$

FIGURE 4. MILP formulation of the Min-Budget problem.

cost interval $[c_{i,y}, c_{i,y+1})$. The final two constraints in (11) ensure that the results shown in Figure 5 hold. Constraints (12) – (15) ensure that all decision variables are in the correct ranges.

α^1_{ir}	α^2_{ir}	desired result
0	0	$lp_i = p_{ir}$
1	0	$lp_i \geqslant p_{ir}$
0	1	$lp_i \leqslant p_{ir}$
1	1	NA

FIGURE 5. Desired results for all possible values of α^1_{ir} and α^2_{ir}. Because α^1 and α^2 are mutually exclusive, the last line can never be true.

4.2. Max-Probability

To obtain the Max-Probability version we replace the objective (1) with

$$\min \sum_{j \in N} lp_j \tag{1*}$$

and replace constraint (10) with the budget constraint

$$b_o = B^* \tag{10*}$$

where B^* is the starting budget. The remaining constraints are unchanged.

The MILP formulations provide a formalism that can be solved using off-the-shelf solvers to provide baseline solutions. However, it is often the case that custom branch-and-bound techniques can use problem specific heuristics to out-perform generic MILP solvers. In the next section we give the details of custom branch-and-bound algorithms for Min-Budget and Max-Probability.

5. BRANCH-AND-BOUND FORMULATIONS

We next detail the state representation, the initial state, successor function, and bounding criteria, for each algorithm. Each algorithm performs a standard depth first branch-and-bound search on the state space, bounding and pruning sections of the search space as explained later.

To simplify notation we again use the failure function $f_i(b)$ that gives the probability of failure when budget b is presented at site i. This is a decreasing function given by

$$f_i(b) = \sum_{x : c_{i,x} > b} P_i(c_{i,x}).$$

5.1. Min-Budget

A solution to the Min-Budget problem is to find a path π and budget B such that the probability of successfully obtaining the item along π is at least p^*_{succ} and $B^* = b_o$ is minimal. The algorithm explores all possible available budget intervals and prunes branches of the search space if the current budget interval requires more budget than the best solution found so far or precludes reaching the required probability of success given the remaining unvisited sites.

5.1.1. *State representation.* Because there are an infinite number of possible initial budgets, B^*, we cannot search over single values of B^*. Instead, we search over a finite number of possible budget ranges for B^*. When the search reaches site i with a budget range $[l_i, u_i)$, the branch-and-bound state is represented as:

$$\langle \pi, [l_i, u_i), p \rangle$$

where $\pi = \langle o, \ldots, i \rangle$ is the path (sequence of sites) followed in G to reach site i starting at the origin o and ending at site i, and p is the probability of failure accumulated along the path π in the branch-and-bound tree. The initial state is then $\langle \langle o \rangle, [0, \infty), 1.0 \rangle$.

5.1.2. *Successor function.* Suppose S_π is the set of all sites on a path π. Given a state $\langle \pi, [l_i, u_i), p \rangle$ we create its potential successor states as follows: for each site $j \in adj(i) \setminus S_\pi$, and for each cost interval $[c_{j,x}, c_{j,x+1})$ at site j, we create a potential successor state $\langle \pi', [l_j, u_j), p' \rangle$, where $\pi' = \langle o, \ldots, i, j \rangle$ is the concatenation of j to π, $[l_j, u_j)$ is a projection of the available budget $[l_i, u_i)$ onto j's cost interval $[c_{j,x}, c_{j,x+1})$ after traversing the edge (i, j), given by

$$[l_j, u_j) = [\max(l_i - t_{ij}, c_{j,x}), \min(u_i - t_{ij}, c_{j,x+1})) \tag{17}$$

and $p' = p \cdot f_j(l_j)$. If $[l_j, u_j)$ is empty, then the corresponding successor can be discarded. Equation (17) is a direct result of combining two intervals. Given one available budget interval $[a, b)$ that is to be projected onto a second budget interval $[c, d)$, the resulting budget interval is $[\max(a, c), \min(b, d))$. Returning to Equation (17), if we start at site i with budget in the range $[l_i, u_i)$ and travel to site j we incur travel cost t_{ij} reducing our budget available for obtain to the interval $[l_i - t_{ij}, u_i - t_{ij})$. Projecting this onto the new budget interval $[l_j, u_j)$ results in Equation (17). This process is depicted in Figure 6.

When a terminal state (one with no successor) $\langle \pi, [l_i, u_i), p \rangle$ is reached, it is a *feasible solution* if $p \leqslant 1 - p^*_{succ}$, otherwise it can be discarded. If it is a feasible solution, then the minimum initial budget required to travel along π and reach i with a budget in the range $[l_i, u_i)$ can be computed as

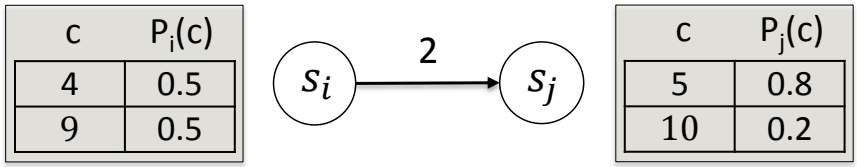
$$\sigma_i = l_i + \sum_{h=0}^{|\pi|-2} t_{\pi_h, \pi_{h+1}} \tag{18}$$

5.1.3. *Bounding criteria.* A state $\langle \pi, [l_i, u_i), p \rangle$ and the entire branch-and-bound subtree rooted at this state can be pruned if either:

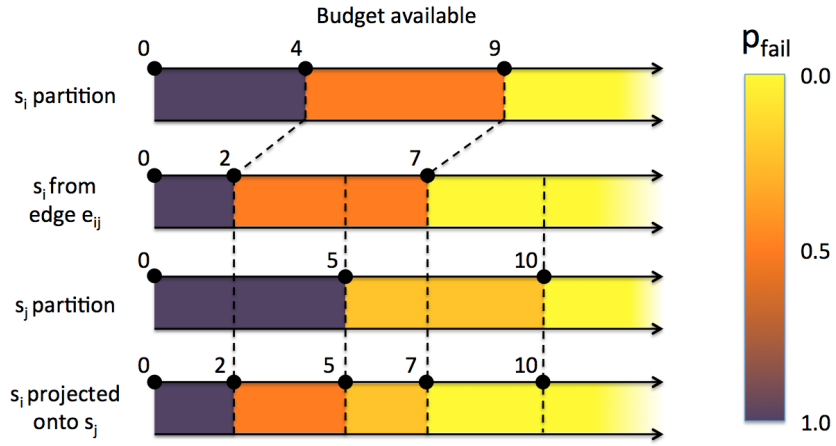
(1) $\sigma_i \geqslant B^*$, where B^* is the best min budget solution found so far that achieves p^*_{succ};
(2) or the following holds:

$$p \cdot \prod_{j \in S \setminus S_\pi} f_j \left(u_i - \min_{k \in (S \setminus S_\pi) \cup \{i\}} t_{kj} \right) > 1 - p^*_{succ}.$$

5.1.4. *Example.* Returning to our example in Figure 2 (reproduced for convenience in Figure 7), assume $p^*_{succ} = 0.95$. Upon expanding our search from o to s_1, we have three budget possibilities: (1) the available budget at s_1 is in the range $[0, 20)$ so we cannot obtain the item at s_1, (2) we have budget in the range $[20, 70)$, resulting in a cumulative probability of failure of 0.7, or (3) we have budget in the range $[70, \infty)$ in which case we succeed with probability 1, and have a feasible solution with total budget $B^* = 70 + 10 = 80$. Returning to possibility (2), we can now expand the interval $[20, 70)$ to the successor s_2 to get two non-empty possible budget intervals: $[5, 15)$ and $[15, 55)$, with corresponding cumulative probabilities of failure 0.7 and 0.28, respectively. Neither of these are feasible. In fact, we should never have considered possibility (2), since a budget upper bound of 70 at s_1 cannot result in a feasible solution. This is because if our available budget at s_1 is less than 70, we will have less than 55 budget available at s_2, resulting in a cumulative probability of failure $p_{fail} = f_{s_1}(70) \cdot f_{s_2}(70 - 15) = 0.7 \cdot 0.6 = 0.42$, which is not less than the maximum allowed probability of failure of 0.05. Possibility (3) can be pruned in the same way since



(a)



(b)

FIGURE 6. Graphical depiction of projecting from one set of cost partitions to another (a) Path goes from s_i to s_j (b) depiction of budget intervals using Equation (17). Budget interval $[0, 4)$ is projected onto the interval $[0, 2)$, $[4, 9)$ is projected onto the intervals $[2, 5)$ and $[5, 7)$, and $[9, \infty)$ is projected onto the intervals $[7, 10)$ and $[10, \infty)$. This results in five possible budget partitions $[0, 2)$, $[2, 5)$, $[5, 7)$, $[7, 10)$, and $[10, \infty)$.

a maximum available budget of 20 can never achieve the minimum required probability of success.

The search process now backs up the search tree and tries expanding the search from o to s_2. This induces three possible state partitions: $\langle\langle o, s_2\rangle, [0, 15), 1\rangle$, $\langle\langle o, s_2\rangle, [15, 65), 0.6\rangle$, and $\langle\langle o, s_2\rangle, [65, \infty), 0\rangle$. The branch along the partition $[0, 15)$ can be pruned by condition (2) since the minimum probability of failure is 1, which is greater than $1 - p^*_{succ}$. The second state along partition $[15, 65)$ has a minimum probability of failure of $0.6 \cdot f_{s_1}(50) = 0.42$ which is greater than 0.05 and is pruned by bounding criterion (2). The final state is pruned by bounding criterion (1) because it requires at least 85 budget which is higher than the best solution found so far, that required a budget of only 80. Thus, the optimal solution path for $p^*_{succ} = 0.95$ is $\langle o, s_1, s_2\rangle$ with a budget of 80.

Generalizing this kind of reasoning, we obtain two bounding criteria. The first is to prune any search branch that requires more budget than the best solution found so far. The second heuristic is the *look-ahead* heuristic, that takes the current available budget and forms an optimistic estimate of the probability of success of this search branch. To compute a lower bound on the probabilty of success we make the false, but optimistic assumption that our available budget at each site j is our current maximum available budget minus the travel

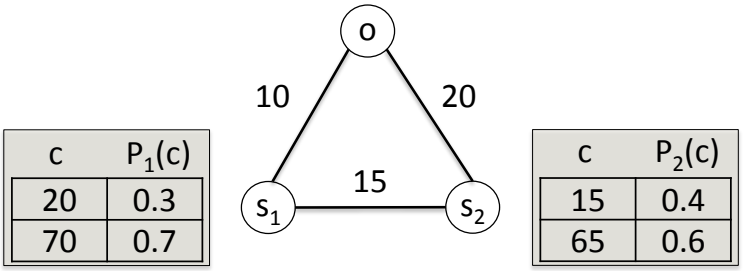


FIGURE 7. A simple instance of a 2 site SPS problem with the agent starting at location s. For each site, s_i, there is a probability $P_i(c)$ of purchasing the item at cost c.

cost along the least cost edge leading from either the current site or an unvisited site into j. Multiplying all of these probabilities of failure gives us a lower bound on the best probability of failure possible along this branch. If the resulting probability of success is less than the required probability of success we can safely prune this branch.

Proposition 1: For any state $\langle \pi, [l_i, u_i), p \rangle$ and any of its successors $\langle \pi', [l_j, u_j), p' \rangle$, $\sigma_i \leqslant \sigma_j$.

Proof. We have the following:

$$\begin{aligned}
\sigma_i &= l_i + \sum_{h=0}^{|\pi|-2} t_{\pi_h, \pi_{h+1}} \\
&= (l_i - t_{ij}) + \sum_{h=0}^{|\pi|-2} t_{\pi_h, \pi_{h+1}} + t_{ij} \\
&= l_i - t_{ij} + \sum_{h=0}^{|\pi'|-2} t_{\pi'_h, \pi'_{h+1}}, \ \ \because \pi \equiv \pi' \text{ up to } i \\
&\leqslant l_j + \sum_{h=0}^{|\pi'|-2} t_{\pi'_h, \pi'_{h+1}}, \text{ by Equation 17} \\
&= \sigma_j.
\end{aligned}$$

□

In words, the above proposition shows that the minimum initial budget does not decrease along any search path, hence the first bounding criterion is correct. The following proposition establishes the correctness of the second bounding criterion.

Proposition 2: For any state $\langle \pi, [l_i, u_i), p \rangle$, if the condition in the second bounding criterion is satisfied then there can be no feasible solution in the branch-and-bound sub-tree rooted at this state.

Proof. For the purposes of contradiction assume that the condition of the second bounding criterion is satisfied *and* there exists a feasible solution in the branch-and-bound sub-tree. Let $\langle \pi', [l_z, u_z), p' \rangle$ be a feasible solution (i.e., terminal state with $p' \leqslant 1 - p^*_{succ}$) in the

branch-and-bound sub-tree rooted at $\langle \pi, [l_i, u_i), p \rangle$. Consider any index k on the path π' beyond π, i.e., $|\pi| \leqslant k \leqslant |\pi'| - 1$. By Equation 17, the upper end of the budget range in the corresponding state is

$$\begin{aligned}
u_{\pi'_k} &\leqslant u_{\pi'_{k-1}} - t_{\pi'_{k-1},\pi'_k} \\
&\leqslant \dots \\
&\leqslant u_{\pi'_{|\pi|-1}} - \sum_{h=|\pi|}^{h=k} t_{\pi'_{h-1},\pi'_h} \\
&= u_i - \sum_{h=|\pi|}^{h=k} t_{\pi'_{h-1},\pi'_h} \\
&\leqslant u_i - t_{\pi'_{k-1},\pi'_k}, \text{ since edge costs are non-negative} \\
&\leqslant u_i - \min_{h \in (S \setminus S_\pi) \cup \{i\}} t_{h,\pi'_k}
\end{aligned}$$

As a result, $f_{\pi'_k}(l_{\pi'_k}) \geqslant f_{\pi'_k}(u_{\pi'_k}) \geqslant f_{\pi'_k}(u_i - \min_{h \in (S \setminus S_\pi) \cup \{i\}} t_{h,\pi'_k})$, since f is a decreasing function. Now,

$$\begin{aligned}
p' &= p \cdot \prod_{k=|\pi|}^{|\pi'|-1} f_{\pi'_k}(l_{\pi'_k}) \\
&\geqslant p \cdot \prod_{k=|\pi|}^{|\pi'|-1} f_{\pi'_k}(u_i - \min_{h \in (S \setminus S_\pi) \cup \{i\}} t_{h,\pi'_k}) \\
&\geqslant p \cdot \prod_{j \in S \setminus S_\pi} f_j(u_i - \min_{h \in (S \setminus S_\pi) \cup \{i\}} t_{h,j}) \\
&> 1 - p^*_{succ}, \text{ by the 2nd bounding criterion,}
\end{aligned}$$

where the penultimate step results from the fact that $f_j(\cdot) \leqslant 1$ and that the product considers all sites $j \in S \setminus S_\pi$ regardless of whether j is on the path π' or not. The above shows that $p' > 1 - p^*_{succ}$ which contradicts of our assumption that $\langle \pi', [l_z, u_z), p' \rangle$ is a feasible solution. □

5.1.5. *Polynomial special case.* Before we explain the branch-and-bound algorithm for Max-Probability we note that while Min-Budget is NP-Complete, a polynomial time solution exists for the special case of $p^*_{succ} = 1$.

Proposition 3: When $p^*_{succ} = 1$, Min-Budget can be solved in $O(|E| + |S| \log |S|)$ time.

Proof. To achieve $p^*_{succ} = 1$, the optimal path π^* must reach at least one site, $i^* \in S$, with budget $b = c_{i^*,\rho}$, to ensure $f_{i^*}(b) = 0$. Any path to i^* other than the least cost path will require more budget, and thus be suboptimal. Thus, once we have found the least cost path, $\hat{\pi}^i$, from o to site i, $\forall i \in S$, we have

$$B^* = \min_{i \in \mathcal{S}} \left(\sum_{h=0}^{|\hat{\pi}^i|-2} t_{\hat{\pi}^i_h, \hat{\pi}^i_{h+1}} + c_{i,\rho} \right). \tag{19}$$

The least cost path from o to every site in S can be calculated in time $O(|E| + |S| \log |S|)$, using Dijkstra's algorithm with a Fibonacci heap (Fredman and Tarjan, 1987). □

5.2. Max-Probability

The formulation for Max-Probability is similar, albeit simpler, with a state representation of $\langle \pi, b_i, p \rangle$ where b_i is the actual budget brought to site i. π and p are as in the ***Min-Budget*** formulation. The initial state is $\langle \langle o \rangle, B^*, 1.0 \rangle$, and the successor function is defined as follows: given a state $\langle \pi, b_i, p \rangle$, for each site $j \in adj(i) \setminus S_\pi$ we create a potential successor state as $\langle \pi', b_i - t_{ij}, p' \rangle$ where π' is as defined in the ***Min-Budget*** formulation and $p' = p \cdot f_j(b_i - t_{ij})$. The successor is discarded if $t_{ij} \geqslant b_i$. The surviving successors can be sorted in increasing order of p', for efficiency. A state $\langle \pi, b_i, p \rangle$ is terminal if it has no successor, or if $p = 0$ (i.e., the item will be obtained for sure). If $p = 0$ we can end the search since no further improvement of $p_{succ} = 1 - p$ is possible. If the best max probability (of success) solution found so far is p^*_{succ}, then a state $\langle \pi, b_i, p \rangle$ can be pruned if

$$p \cdot \prod_{j \in S \setminus S_\pi} f_j \left(b_i - \min_{k \in (S \setminus S_\pi) \cup \{i\}} t_{kj} \right) > 1 - p^*_{succ}. \tag{20}$$

The correctness of the above bounding criterion can be established in a way similar to Proposition 2.

6. EXPERIMENTAL RESULTS AND ANALYSIS

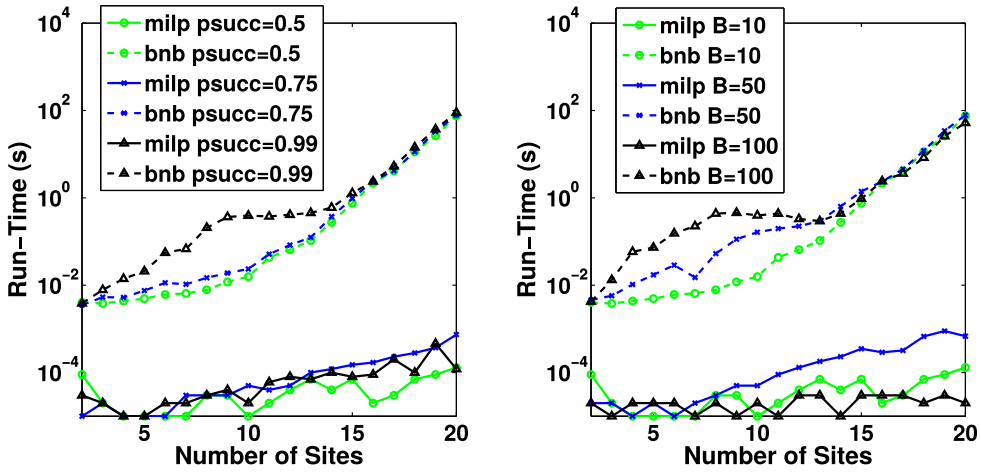
Previously, we proposed both an MILP formulation of the RA-SPS problem as well as a customized branch-and-bound algorithm. In this section we show that our customized branch-and-bound solver results in a substantial decrease in run-time when compared with solutions to our MILP formulation using an off-the-shelf solver.

We then investigate the hardness landscape of the Min-Budget and Max-Probability problems to provide insights into where the most difficult problems lie. For our following analysis we generated random problem instances with various numbers of sites and costs. Costs were chosen uniform randomly from the interval $[1, 100]$. Unless otherwise specified the probability for each cost c_i was chosen by randomly picking a value from the interval $(0, 1)$ for each cost and then normalizing these values to sum to one. Edge costs between each pair of sites were chosen uniform randomly from the interval $[1, 100]$, except for edge costs to the destination site d, which were set to 0. For simplicity, the results presented in sections 6.1–6.4 use only two costs. We explore different numbers of costs in section 6.5 and explore problems with a probability of infinite cost in section 6.6.

6.1. MILP vs. Branch-and-Bound

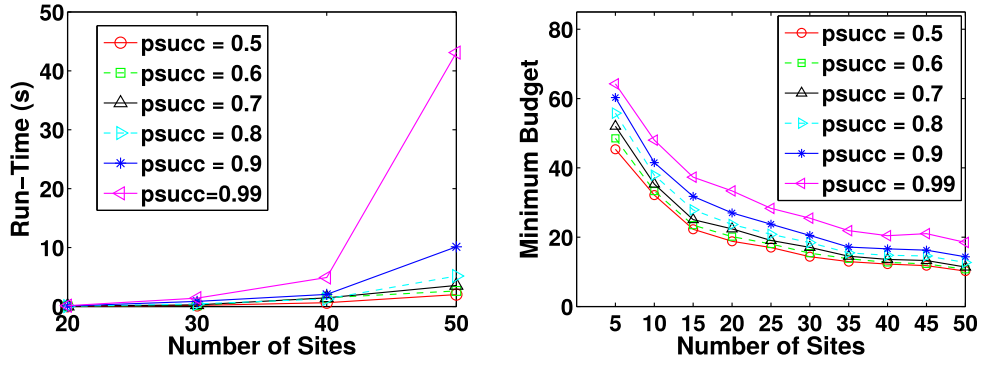
We compare the run-time complexity of solving the MILP formulation using an off-the-shelf solver versus solving using our custom branch-and-bound algorithms. We solved the Min-Budget and *Max-Probability* MILP problems using the CPLEX based SCIP solver available on NEOS (Czyzyk et al., 1998; Dolan, 2001; Gropp and Moré, 1997). We generated SPS problems with the number of sites $|S|$ ranging from 2 to 9, and generated 20 random problem instances for each value of $|S|$. We then computed the average run-time for the different solution methods on these instances. We used $M = 500000$, $\delta = 0.001$, and $\log(0) = -100000$. Infinite travel costs were set to 100000 and infinite purchase costs were set to 500. The results for solving Min-Budget with $P_{succ} = 0.75$ and Max-Probability with $B^* = 50$ are shown in Figure 8 (note the log scale of the y-axis).

As anticipated, average run-time for the MILP is drastically longer than the run-time for the branch-and-bound solutions over a variety of parameters. Based on these results we chose to restrict our remaining analysis to the branch-and-bound algorithms.



(a) (b)

FIGURE 8. (a) Average run-time for the Min-Budget branch-and-bound algorithm. (b) Average minimum budget results from the Min-Budget branch-and-bound algorithm.



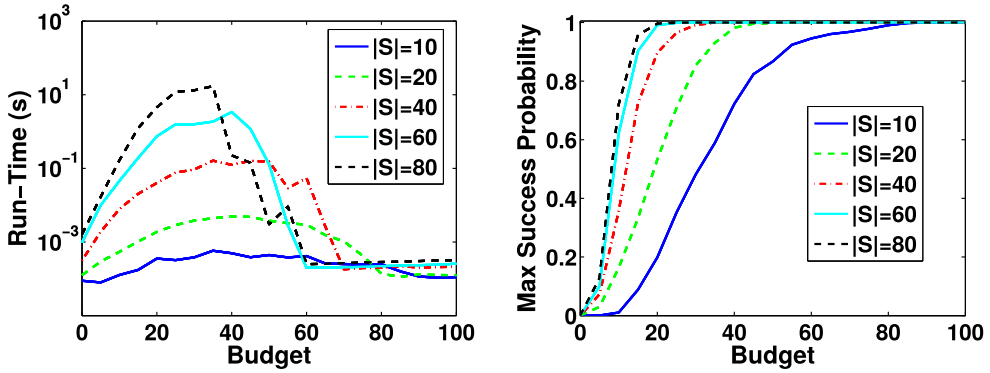
(a) (b)

FIGURE 9. (a) Average run-time for the Min-Budget MILP and branch-and-bound algorithm. (b) Average run-time for the Max-Probability MILP and branch-and-bound algorithm.

6.2. Branch-and-Bound Analysis for Min-Budget

To study the properties of the Min-Budget problem we generated a set of random problem instances, with the number of sites varying between 5 and 50. For each problem size, we generated 100 random graphs with edge costs and item costs randomly chosen between 1 and 100 and evaluated our algorithm on these 100 replicates. Figure 9(a) shows that the run-time of the Min-Budget branch-and-bound solver increases exponentially as expected as the number of sites increases. We also see that requiring a higher probability of success requires significantly more computational time as the number of sites increases.

Figure 9(b) shows that as the number of sites increases, the optimal minimum budget steadily decreases. These results make sense given that a larger number of sites means a higher chance of being able to obtain the item at a low cost. Additionally, as expected we see that higher required probabilities of success require higher budgets.



(a) (b)

FIGURE 10. (a) Average run-time for the Max-Probability branch-and-bound algorithm. (b) Average maximum p_{succ} results from the Max-Probability branch-and-bound algorithm.

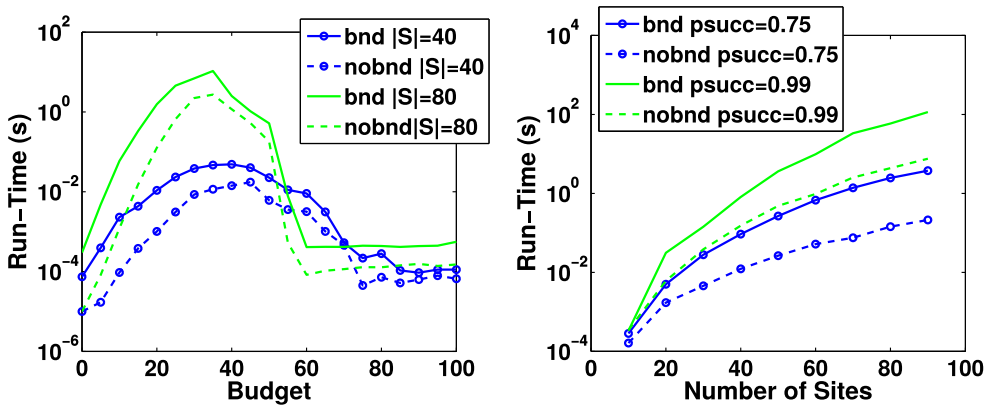
6.3. Branch-and-Bound Analysis for Max-Probability

We also investigated the Max-Probability branch-and-bound algorithm using the same random problem generation approach. Figure 10(a) shows the average run-time results over 100 randomly generated complete graphs with different numbers of sites. We see an interesting trend as the available starting budget is increased from 0 to 100. When the starting budget is between 20 and 40 we see an exponential increase in run-time (note the log scale of the y-axis) as $|S|$ increases to between 30 and 40 and then decreases as the budget approaches 100. This peak in the run-time occurs because problems with very small budgets cannot explore much of the search space before running out of budget. Additionally, because we terminate search if we ever find a path with $p_{succ} = 1.0$, problems with large available starting budget allow the algorithm to quickly find these "perfect" solution paths and quickly terminate search. Similar to binary constraint satisfaction problems, the Max-Probability problem is easy to solve if there is a very large or very small budget: either visiting any site guarantees obtaining the item or the number of reachable sites is so small the search process can easily be solved with minimal branching. The real difficulty in both problems comes when the starting budget or number of constraints is somewhere between these two extremes. There are now many possible solutions that must be explored to determine which is optimal. This phase transition is reminiscent of the phase transition found when solving binary constraint satisfaction problems (Prosser, 1996): when there are few constraints, problems are easy to solve and when there are many constraints it is easy to show that the problem is infeasible.

Figure 10(b) shows a similar result, but in terms of the maximum probability of success of the optimal path found by the Max-Probability branch-and-bound algorithm. We see a sigmoidal phase transition over the average probability of success as the available budget is increased for different numbers of sites.

6.4. Bounding Criteria Sensitivity Analysis

The look-ahead bounding criterion (criterion (2)) used by both Min-Budget and Max-Probability is computationally intensive, $O(n^2)$. To determine whether it is worthwhile to use this criterion we ran an experiment using the same parameters as the previous section: $p^*_{succ} = 0.9$ and $B^* = 40$ for 40 sites. We evaluated 100 random problems. For Min-Budget and Max-Probability we compared the performance with and without this criterion. Figure 11 shows that in fact the pruning capability of the second bounding criterion is outweighed by its computational cost.



(a) (b)

FIGURE 11. Average run-times for the (a) Max-Probability and (b) Min-Budget branch-and-bound algorithms with and without the look-ahead bounding criterion. Note the different scales.

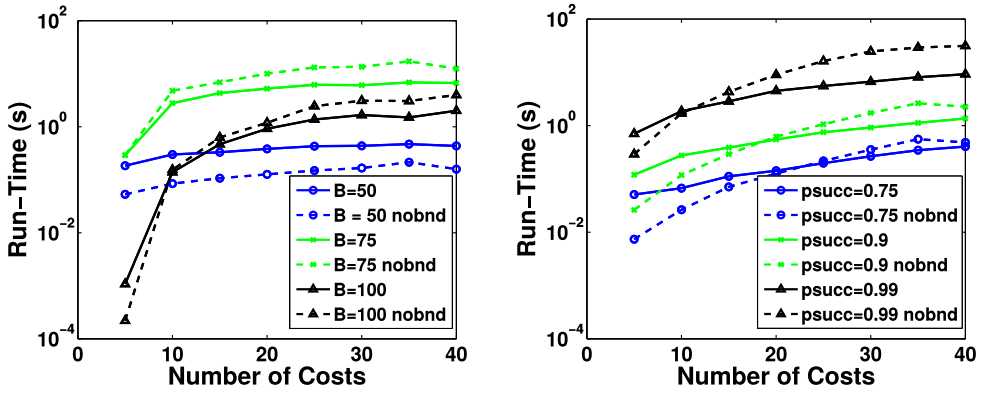
6.5. Multiple Costs

The previous analysis has focused on the simplified case of two finite costs possible at every site (other than the starting site). We now consider how the problem difficulty changes as the number of possible costs at each site increases. Figure 12 shows how the run time changes for both Min-Budget and Max-Probability as the number of possible costs increases. Results are shown for 30 sites where Min-Budget was solved with p^*_{succ} = 0.75, 0.9 and 0.99 and Max-Probability was solved with B^* = 50, 75, and 100. We see that Min-Budget's run time increases as the number of costs increase, and that the same is true for Max-Probability, except the increase is more gradual. We also found that eliminating the look-ahead bounding criterion actually increases run-time for larger budgets, larger required probabilities of success, and for larger numbers of costs. For Min-Budget, as more costs are added, this increases the branching factor. Thus, being able to prune more branches is eventually worth the extra computation. For Max-Probability, with 2 costs (see Figure 11(a)) there is a smaller gap between including and not including the bounding criterion. As the number of costs increases, the compuation to evaluate the probability of success at each node increases, thus for budgets that require evaluating many branches, there comes a point where the extra pruning is worth the extra cost.

6.6. Unbounded RA-SPS

The previous analysis focused on problems with random uniform edge costs and item costs. However, when probabilities on item costs are uniformly distributed, often a solution can be found easily by simply moving directly to a site with a very high probability of success. Additionally, if all costs are finite, there exists a budget threshold at which the item can be obtained with certainty at a site, terminating search.

To make the problems much more difficult we add a probability of failure at every site by ensuring that $P_i(\infty) > 0 \; \forall i \in S$, referred to hereafter as the *Unbounded SPS problem*. This removes the ability of Max-Probability to exit the search early. It also means that finding a plan that can achieve $p^*_{succ} = 1$ is impossible, and thus precludes the polynomial time solution for Min-Budget discussed in Proposition 3. To generate Unbounded SPS problem instances we generate the travel costs uniformly from the interval [1,100]. We then generate cost profiles for each site by picking a single finite cost c_i in the interval [1,100] with



(a) (b)

FIGURE 12. Average run-time for (a) Max-Probability branch-and-bound algorithm and (b) Min-Budget, with and without the look-ahead bounding criterion over different numbers of costs at each site.

$P_i(c_i) \sim U(0, 0.5)$ and $P_i(\infty) = 1 - P_i(c_i)\ \forall i \in S$, where $U(\cdot, \cdot)$ is the continuous uniform distribution.

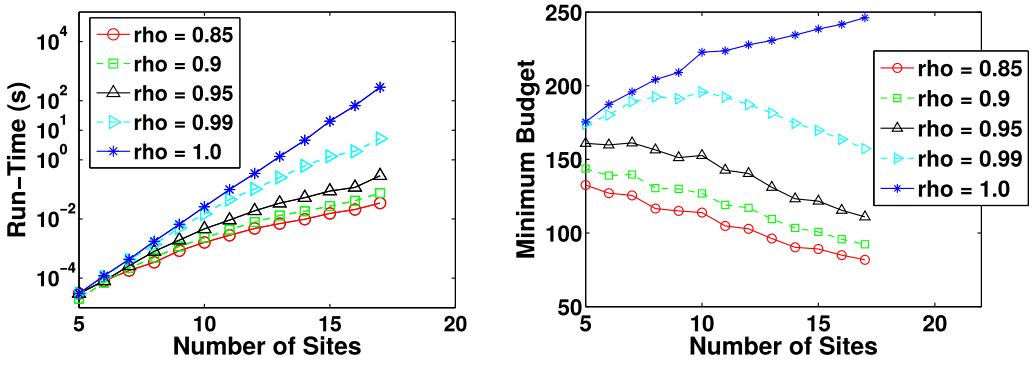
6.6.1. *Min-Budget for Unbounded SPS.* Including a probability that the item cannot be obtained at each site makes some values of p^*_{succ} infeasible, e.g., finding a solution that guarantees $p^*_{succ} = 1$. For the infinite cost case we avoid the problem of potentially infeasible problems by first computing the maximum probability of success as

$$p^{max}_{succ} = 1 - \prod_{i \in S} P_i(\infty). \tag{21}$$

Solving Min-Budget with $p^*_{succ} = p^{max}_{succ}$ is feasible and requires visiting every site. Because of the exponential complexity involved in examining every possible tour of the sites, finding an optimal solution using our branch and bound technique will only work for very small instances. We consider the case where $p^*_{succ} = \rho \cdot p^{max}_{succ}$ for different values of ρ. The results in Figure 13 shows a much more dramatic increase in complexity. When $\rho = 1$ we are forced to examine solutions that visit every site, resulting in dramatic increases in run-times and larger required budgets as the number of sites increases.

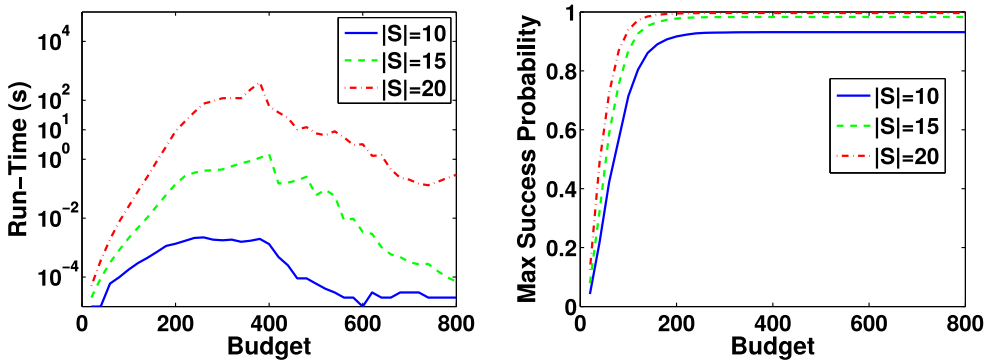
6.6.2. *Max-Probability for Unbounded SPS.* When solving *Max-Probability* for Unbounded SPS, we can no longer exit the search process by finding a perfect solution that gives $p^*_{succ} = 1$. However, given a graph we can easily calculate p^{max}_{succ}. If we ever achieve a probability of success equal to p^{max}_{succ} we can exit the search since we know that finding another path with $p^*_{succ} > p^{max}_{succ}$ is impossible.

Figure 14 shows that Max-Probability on the Unbounded SPS problem once again exhibits a bump in run-time complexity, but for larger budgets. Because the probability of success at each site is bounded above by 0.5, we require a larger amount of budget than for the uniform probability case. The maximum probability of success now depends on the number of sites available, as shown in Figure 14(b). In Figure 10(b) the maximum probability of success was always 1, since all costs were finite. In the unbounded case, this is not always true, but the maximum possible success probability approaches 1 as the number of sites increases.



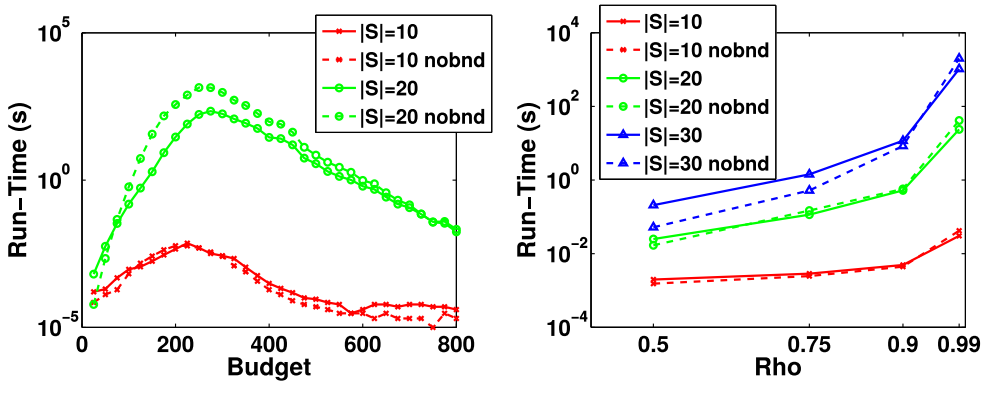
(a) (b)

FIGURE 13. Results for Min-Budget on the Unbounded SPS problem: (a) average run-time, (b) average minimum budget required to achieve p^*_{succ}.



(a) (b)

FIGURE 14. Results for Max-Probability on the Unbounded RA-SPS problem: (a) Average run-time over different starting budgets, (b) Average maximum p_{succ} given a certain starting budget.



(a) (b)

FIGURE 15. Results for (a) Max-Probability and (b) Min-Budget on the Unbounded RA-SPS problem with and without the look-ahead bounding criterion.

6.6.3. *Effect of Bounding Criteria.* We also ran an experiment to test the effects of removing the look-ahead bounding criterion. The results are shown in Figure 15. The results show that removing the bounding criterion improves run-time for low ρ and low and high initial budgets; however, for $\rho = 0.99$ and for budgets between 200 and 400, removing the look-ahead bounding criterion actually increases the run-time. These regions are the most difficult, and as such they require searching the most branches of the search space. Removing the bounding criterion saves some computation time and improves the overall run-time for problems instances that require searching less of the solution space; however, the harder instances (middling budget and high ρ) benefit from the extra pruning since the search space is much larger.

7. HEURISTICS

The previous sections developed and analyzed both a MILP formulation of the RA-SPS and a customized branch-and-bound algorithm for solving RA-SPS problems. Both of these approaches provide optimal solutions but require exponential run-time in the worst-case. In this section we examine the performance of several heuristics that allow us to solve large problem instances efficiently. We examine the performance of both a greedy and a randomized local search heuristic. Before discussing the details of these heuristics, we first note that a solution to both the Min-Budget and Max-Probability problems can be represented by an ordering of the sites in S. Given any ordering of the sites we can perform our branch and bound search along the fixed path determined by the ordering. Thus, we can generate potential solutions by generating permutations of the sites and solving the Min-Budget and Max-Probability problems where the sites are visited in the order specified by the permutation. This reduces the problem to finding a solution along a fixed path and, as we show next, this restriction enables efficient heuristics for the Min-Budget and Max-Probability solutions.

7.1. Solutions along a fixed path

We now show that solving Min-Budget and Max-Probability along a fixed path takes polynomial time.

Lemma 1: Consider traveling from site i to site j with available budget in the interval $[l_i, u_i)$. If site j contains k costs that fall within a budget interval $[l_i - t_{ij}, u_i - t_{ij})$ then projecting $[l_i, u_i)$ across edge t_{ij} onto all budget intervals of site j results in at most $k + 1$ partitions.

Proof. Let the k costs be $c_{j,y}, c_{j,y+1}, \ldots, c_{j,y+k-1}$. This induces $k + 1$ partitions that have non-empty intersections with $[l_i - t_{ij}, u_i - t_{ij})$:

$$[c_{j,y-1}, c_{j,y}), [c_{j,y}, c_{j,y+1}), \ldots, [c_{j,y+k-2}, c_{j,y+k-1}). \tag{22}$$

Using Equation (17) to project $[l, u)$ onto each of these we have the following $k + 1$ budget partitions: $[l_i, c_{j,y})$, $[c_{j,y}, c_{j,y+1})$, $\ldots$, $[c_{j,y+k-2}, c_{j,y+k-1})$, $[c_{j,y+k-1}, u_i)$. □

Lemma 2: Consider a path π and two adjacent sites i and j along π, i.e., $\pi_m = i$ and $\pi_{m+1} = j$ for some $0 \leqslant m < |\pi|$. If there are at most κ prices at a site j, then the number of budget partitions evaluated at site j by the Min-Budget branch and bound algorithm along π is at most $\kappa + |R_i|$.

Proof. Consider each partition $[l_i, u_i) \in R_i$. Let the number of costs available at site j that are contained in $[l_i - t_{ij}, u_i - t_{ij})$ be equal to κ_i. Then by Lemma 1 the number of partitions

that need to be considered at site j is at most

$$\sum_{i=1}^{|R_i|}(\kappa_i + 1) = \kappa + |R_i| \tag{23}$$

□

Figure 6 shows an example of Lemma 2. Three partitions $[0, 4), [4, 9)$, and $[7, \infty)$ are projected across an edge of cost 2 onto three partitions $[0, 5), [5, 10)$, and $[10, \infty)$, resulting in $5 < \kappa + |R_i| = 6$ partitions.

Theorem 1: Given a fixed path π of length n where each site along π has no more than κ possible prices, *Min-Budget* can be solved in time $O(\kappa n^2)$.

Proof. At the starting site we only have one partition $[0, \infty)$, so by Lemma 2 the number of partitions evaluated by Min-Budget at the mth site along π is bounded above by

$$\mathcal{P}_m = 1 + (m - 1) * \kappa \tag{24}$$

Thus the upper bound on the total number of branch and bound states evaluated along a path π of length n is

$$\sum_{m=1}^{n} \mathcal{P}_m = \frac{n(2 + (n - 1)\kappa)}{2} \tag{25}$$

Thus the number of states considered is linear in κ and quadratic in the length of the path n resulting in $O(\kappa n^2)$ complexity. □

Theorem 2: Given a fixed path π of length n Max-Probability can be solved in time $O(n)$.

Proof. Unlike the Min-Budget problem, in the Max-Probability problem each site has only one successor in the branch and bound algorithm, thus the number of branch and bound states is equal to the size of the path. □

7.2. Algorithms

The previous section showed that if we are given a single permutation π of the sites in S, we can solve the Max-Probability or Min-Budget problems in polynomial time on the path defined by the permutation by simply applying the successor functions for the branch-and-bound algorithms described previously. Thus, heuristic methods can include any way of searching over permutations of the sites in S. We investigate a simple greedy heuristic and an iterative randomized local search heuristic.

7.2.1. *Greedy.* Our greedy heuristic for Min-Budget starts at o and picks the next unvisited site j such that

$$j = \underset{j \in unvisited}{\operatorname{argmin}} \min_{r=1,\ldots\kappa} \frac{t_{ij} + c_{j,r}}{1 - f_j(c_{j,r})}. \tag{26}$$

In other words, when starting at site i we pick the next site that has the smallest cost to success ratio over all possible successors and possible costs.

For Max-Probability we have a given starting budget and must choose where to move to maximize our probability of success. Starting at o our Max-Probability greedy heuristic picks the next unvisited site j such that

$$j = \underset{j \in unvisited}{\operatorname{argmin}} f_j(b_i - t_{ij}) \tag{27}$$

where b_i is the budget available at current site i.

TABLE 1. Average % from optimal budget for *Min-Budget* and *Max-Probability* for travel and item costs between 1 and 100 and random uniform probabilities. Results are for $p^*_{succ} = 0.99$ and $B^* = 30$, respectively. Best results in each row are bolded.

sites	Greedy	RLS	RLS-G
minb_20	44.3	42.3	**28.2**
minb_50	83.1	86.1	**51.2**
minb_100	84.1	84.8	**56.1**
maxp_20	31.2	33.8	**29.3**
maxp_50	21.2	27.0	**19.5**
maxp_100	10.2	16.5	**8.7**

TABLE 2. Heuristic performance for Min-Budget on the unbounded RA-SPS problem. Results for show *% improvement* over the best of $|S|$ random permutations. Best results in each row are bolded.

ρ	sites	Greedy	RLS	RLS-G
0.75	100	63.23	55.72	**70.95**
	500	77.94	69.04	**82.47**
	1000	81.94	72.54	**84.79**
	5000	87.30	75.98	**88.73**
0.9	100	68.85	65.07	**75.45**
	500	82.86	75.07	**85.82**
	1000	85.83	78.13	**88.39**
	5000	90.88	80.33	**91.80**
0.99	100	74.62	77.03	**81.40**
	500	87.16	83.74	**89.70**
	1000	90.26	86.21	**92.09**
	5000	94.05	88.72	**94.82**

7.2.2. *Randomized Local Search.* For our randomized local search heuristic we use an approach that has shown success on other combinatorial optimization problems (Polyakovskiy et al., 2014). We start with an initial permutation of the sites π and repeatedly try to hill-climb in permutation space by swapping the ordering of two of the sites. To evaluate the solution quality of any permutation we compute the minimum budget required to meet p^*_{succ} or the maximum probability of success given initial budget B^*. We initialized the local search with either with a random permutation (RLS) or with the solution obtained using the greedy heuristic (RLS-G). We terminated the randomized local search after $X = |S| \cdot (|S| - 1)/2$ swaps that did not result in an improvement, at which point we assume that we have found a local optima.

7.3. Results

We first compare the performance of the heuristics on RA-SPS problems with only finite costs. We generated problem instances with travel and item costs chosen uniform random between 1 and 100, and probabilities chosen uniform randomly and normalized to be between 0 and 1. We compare the heuristic solutions to the optimal. We used values of $p^*_{succ} = 0.99$, $B^* = 30$ for Min-Budget and Max-Probability, respectively. These values were chosen to lie in the difficult regions of the problem space, as shown in Figures 9 and

TABLE 3. Heuristic performance for Max-Probability on the unbounded RA-SPS problem. Results for show *% improvement* over the best of $|S|$ random permutations. Best results in each row are bolded.

B^*	sites	Greedy	RLS	RLS-G
100	100	18.22	**24.89**	20.24
	500	16.41	**21.42**	18.50
	1000	14.85	**18.57**	16.38
	5000	12.14	**14.15**	12.78
300	100	3.51	**4.56**	4.37
	500	2.29	**2.68**	2.61
	1000	1.80	**2.12**	2.09
	5000	1.29	**1.36**	1.35
600	100	0.36	**0.41**	**0.41**
	500	0.18	**0.20**	**0.20**
	1000	0.14	**0.15**	**0.15**
	5000	**0.08**	**0.08**	**0.08**

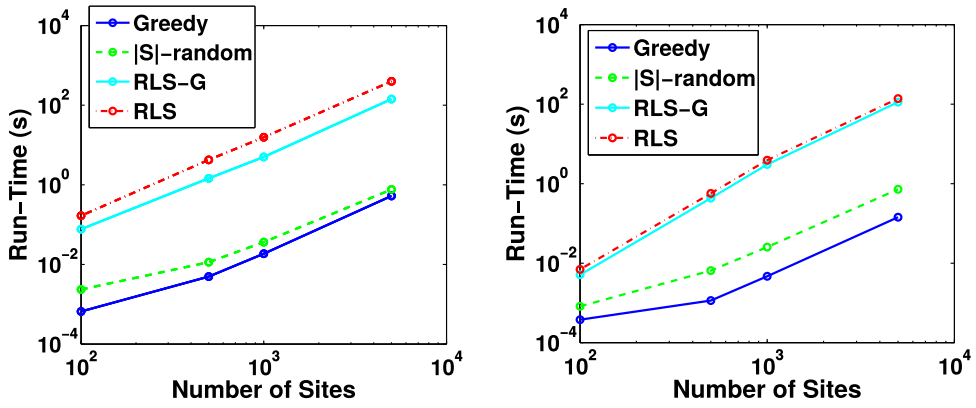


FIGURE 16. Average run-times for (a) Min-Budget and (b) Max-Probability for Greedy, $|S|$-random, RLS, and RLS-G heuristics for large numbers of sites. Results are for $\rho = 0.99$ and $B^* = 300$, respectively.

10. The resulting average performances for each of the heuristics over 100 random instances are reported in Table 1.

We also compared the performance of the heuristics on the Unbounded RA-SPS problem, generating problems as described in the previous section on infinite item costs. We examined results for several values of ρ and B^* on large numbers of sites where optimal solutions are not feasible. We compare the results of the heuristics to the best of $|S|$ randomly generated permutations. The average results over 100 random instances for Min-Budget and Max-Probability are shown in Tables 2 and 3, respectively.

For Min-Budget (see Table 2) we see that often the Greedy heuristic performs better than the RLS heuristic; however, RLS-G performs the best since it is initialized with the greedy solution and then hill climbs from there. As ρ increases, all the heuristic solutions perform increasingly better than random. This can be explained by noting that when $\rho = 1$ the Min-Budget problem is equivalent to the TSP, thus higher values of ρ make finding a good solution more difficult. Because the number of possible random permutations grows

exponentially, the chance of randomly picking a good solution decreases for larger numbers of sites, as shown by the increasing performance over random for larger instances.

For Max-Probability (see Table 3) we see that in all but the final row, Greedy performs the worse than using RLS. If we initialize RLS with the Greedy solution, we see that the RLS-G heuristic also tends to perform worse than RLS for smaller starting budgets of 100 and 300 and performs no better than RLS for a starting budget of 600. Thus, seeding the RLS algorithm with the Greedy solution appears to negatively bias the randomized local search and is not able to find its way to good local optima. Note that the trend for Max-Probability is opposite that of Min-Budget. As the number of sites increases, the percent improvement over random decreases. This is because adding more sites increases the probability that a random path will achieve a high probability of success given the initial budget. A similar trend can be seen with the size of the initial budget. When the initial budget increases, it becomes increasingly easy to find a good solution that achieves a high probability of success. For a starting budget of 600 and 5000 sites all heuristics have the same average performance which is very close to the performance of a simple random heuristic. Note that in this case the random policy has an average probability of success of 0.9991.

8. DISCUSSION

Before concluding we discuss the relevance of some of our assumptions. We assume that each $(i, j) \in E$ has a non-negative cost of travel t_{ij} that is deterministic and known. These travel costs could be obtained by calculating actual distances from road maps or satellite imagery. We also assume a finite number of potential costs at each site. We believe this is a reasonable assumption, given that our probabilistic prior knowledge comes from some sort of estimation method which will likely have only a finite number of possibilities (e.g. the Mars rover with a finite number of methods for obtaining a rock sample). Even if the cost distributions are continuous, they may be discretized to a fixed number of costs, and our algorithms will find solutions within some ϵ of optimal depending on the granularity of the discretization.

In some physical search problems there may be a required physical destination (e.g. a UAV that must land on a run-way when it has finished). Our previous algorithms and MILP formulation can easily be extended to this case. We first find the shortest cost path to reach the destination from each site. This "return cost" is then simply added to each cost at each site to obtain a new cost $\tilde{c}_{i,\kappa} = c_{i,\kappa} + t_{id}$. This ensures that successfully obtaining an item means that the agent has enough budget to not only obtain the item, but to also to travel to the destination site. Using $\tilde{c}$ instead of c allows us to use the identical MILP formulation for solving problems with a required physical destination site. To adapt the branch-and-bound approach we can simply add shortest path costs to the possible item costs and then only keep solutions that end at the destination node. The heuristics can be adjusted similarly to account for a physical destination.

9. CONCLUSIONS AND FUTURE WORK

Many real-world search problems involve searching for an item in a physical environment where there is a single budget that must be used to both explore and obtain the item, and where the exact cost to obtain the item is not fully known in advance. Some examples include patrol or surveillance, a rover seeking to mine a specific mineral, or even a shopper looking for an affordable souvenir among many stores. Despite the applicability of this type of model, previous work in the field of Artificial Intelligence has almost exclusively focused on solutions in simple one-dimensional spaces (Hazon et al., 2013). Researchers

in Operations Research have examined stochastic physical search problems, but only find minimum expected cost solutions (Kang and Ouyang, 2011). Because minimum expected cost solutions assume an average case analysis, these solutions do not provide risk-aware solutions for applications where the law of large numbers does not apply, such as when the solution is executed only once. Our work extends previous work by providing algorithms that find risk-aware solutions for stochastic physical search problems on complete metric graphs that intelligently balance the cost of the solution along with the actual probability of success.

To the best of our knowledge, this work provides the first exact and heuristic algorithmic solutions to Min-Budget and Max-Probability Risk-Aware Stochastic Physical Search (RA-SPS) on complete metric graphs. We first formulated these problems as a mixed-integer linear program. This provides a theoretical formalism and bench-mark from which we developed custom branch-and-bound algorithms that take advantage of the problem structure allowing for much faster execution times. We also generated empirical insights into the hardness landscape of the RA-SPS problem. Based on these results, we see that while similar, the Min-Budget and Max-Probability problems have interesting characteristics that differentiate them. Both exhibit worst case exponential complexity; however, we see that the size of problem, the starting budget, and required probability of success have a large impact on tractability. We examined several simple heuristics for solving both Min-Budget and Max-Probability. Based on our results we see that simple greedy solutions perform poorly when compared to local search. We hypothesize that evolutionary search methods will retain the benefits of our local search method while enabling an exploration of more local optima.

For future work, there are still many interesting open problems. Extending our algorithms to allow for multiple agents searching for multiple items on general graphs is of great interest. Including multiple intelligent agents allows for many interesting extensions, including heterogeneous agents with different capabilities and different available budgets, agents that can communicate and possibly collaborate to obtain an item or accomplish a task, and possibly self-interested or adversarial agents. Additionally, developing polynomial-time approximation schemes for both the single and multi-agent cases on general graphs is still an open problem.

Our work has examined stochastic physical search problems where there is probabilistic knowledge about the item costs, but we have assumed that the travel costs are known. Once interesting avenue of future research is to allow stochastic travel costs. We are currently investigating the effects of possible failures while traveling along edges for multi-agent planning problems. Another interesting extension would be to use more complex distributions to reflect the uncertainty. Specifically, having dependent random variables would allow the probability of acquiring the item increase (or decrease) at some sites if it is not easily acquired at the current location.

Finally, even though our discussions in this paper have focused on budgets that represent battery power or fuel, the costs in the problems may also represent time and an agent may be given only a finite amount of time to complete a task. In future work we hope to explore this idea more fully, including the interesting extensions to multiple agents searching in parallel.

ACKNOWLEDGMENT

We thank the anonymous reviewers for helpful comments and suggestions. The formulation of the MILP and the formulation and theoretical analysis of the bounding criteria were supported by an AFOSR Faculty Fellowship awarded to Bikramjit Banerjee in summer 2014. This work was also supported by an AFOSR Laboratory Research grant. The views expressed in this article are those of the authors and do not necessarily reflect the official

policy or position of the Air Force Research Laboratory, Department of Defense, nor the U.S. Government.

REFERENCES

AUMANN, ROBERT J, and SERGIU HART. 1994. Handbook of game theory with economic applications, Volume 2. Elsevier.

AUMANN, YONATAN, NOAM HAZON, SARIT KRAUS, and DAVID SARNE. 2008. Physical search problems applying economic search models. *In* AAAI, pp. 9–16.

AUSIELLO, GIORGIO, VINCENZO BONIFACI, and LUIGI LAURA. 2008. The online prize-collecting traveling salesman problem. Information Processing Letters, **107**(6):199–204.

AUSIELLO, GIORGIO, MARC DEMANGE, LUIGI LAURA, and VANGELIS PASCHOS. 2004. Algorithms for the on-line quota traveling salesman problem. *In* Computing and Combinatorics. Springer, pp. 290–299.

AWERBUCH, BARUCH, YOSSI AZAR, AVRIM BLUM, and SANTOSH VEMPALA. 1998. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. SIAM Journal on Computing, **28**(1):254–262.

BALAS, EGON. 1989. The prize collecting traveling salesman problem. Networks, **19**(6):621–636.

BENAZERA, EMMANUEL, RONEN BRAFMAN, NICOLAS MEULEAU, ERIC A HANSEN, and OTHERS. 2005. Planning with continuous resources in stochastic domains. *In* IJCAI, pp. 1244–1251.

BOUTILIER, CRAIG, THOMAS DEAN, and STEVE HANKS. 2011. Decision-theoretic planning: Structural assumptions and computational leverage. *In* arXiv preprint arXiv:1105.5460.

BRAFMAN, RONEN I, and RAN TAIG. 2011. A translation based approach to probabilistic conformant planning. *In* Algorithmic Decision Theory. Springer, pp. 16–27.

BROWN, DANIEL S., JEFFREY HUDACK, and BIKRAMJIT BANERJEE. 2015. Algorithms for stochastic physical search on general graphs. *In* AAAI Workshop on Planning Search and Optimization.

BROWN, DANIEL S, STEVEN LOSCALZO, and NATHANIEL GEMELLI. 2015. k-agent sufficiency for multiagent stochastic physical search problems. *In* Algorithmic Decision Theory. Springer, pp. 171–186.

CZYZYK, JOSEPH, MICHAEL P MESNIER, and JORGE J MORÉ. 1998. The neos server. Computing in Science and Engineering, **5**(3):68–75.

DELL'AMICO, MAURO, FRANCESCO MAFFIOLI, and PETER VÄRBRAND. 1995. On prize-collecting tours and the asymmetric travelling salesman problem. International Transactions in Operational Research, **2**(3):297–308.

DOLAN, ELIZABETH D. 2001. Neos server 4.0 administrative guide. *In* arXiv preprint cs/0107034.

DOMSHLAK, CARMEL, and JÖRG HOFFMANN. 2007. Probabilistic planning via heuristic forward search and weighted model counting. J. Artif. Intell. Res.(JAIR), **30**:565–620.

FERGUSON, THOMAS S. 1989. Who solved the secretary problem? *In* Statistical science, pp. 282–289.

FREDMAN, MICHAEL L, and ROBERT ENDRE TARJAN. 1987. Fibonacci heaps and their uses in improved network optimization algorithms. Journal of the ACM (JACM), **34**(3):596–615.

GROPP, WILLIAM, and JORGE MORÉ. 1997. Optimization environments and the neos server. *In* Approximation theory and optimization, pp. 167–182.

GUTIN, GREGORY, and ABRAHAM P PUNNEN. 2002. The traveling salesman problem and its variations, Volume 12. Springer.

HAZON, NOAM, YONATAN AUMANN, and SARIT KRAUS. 2009. Collaborative multi agent physical search with probabilistic knowledge. *In* Twenty-First International Joint Conference on Artificial Intelligence.

HAZON, NOAM, YONATAN AUMANN, SARIT KRAUS, and DAVID SARNE. 2013. Physical search problems with probabilistic knowledge. Artificial Intelligence, **196**:26–52. ISSN 00043702. . http://linkinghub.elsevier.com/retrieve/pii/S0004370213000027.

HERROELEN, WILLY, and ROEL LEUS. 2005. Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research, **165**(2):289–306. ISSN 03772217. . http://linkinghub.elsevier.com/retrieve/pii/S0377221704002401.

HUANG, JINBO. 2006. Complan: A conformant probabilistic planner. *In* ICAPS 2006, pp. 63.

HUDACK, JEFFREY, NATHANIEL GEMELLI, DANIEL S. BROWN, STEVEN LOSCALZO, and JAE OH. 2015. Multiobjective optimization for the stochastic physical search problem. *In* The 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems.

JOZEFOWIEZ, NICOLAS, FRED GLOVER, and MANUEL LAGUNA. 2008. Multi-objective meta-heuristics for the traveling salesman problem with profits. Journal of Mathematical Modelling and Algorithms, **7**(2):177–195.

KANG, SEUNGMO, and YANFENG OUYANG. 2011. The traveling purchaser problem with stochastic prices: Exact and approximate algorithms. European Journal of Operational Research, **209**(3):265–272. ISSN 03772217. . http://linkinghub.elsevier.com/retrieve/pii/S0377221710006089.

KATAOKA, SEIJI. 1988. An algorithm for single constraint maximum collection problem. J. OPER. RES. SOC. JAPAN., **31**(4):515–530.

KOENIG, SVEN, and MAXIM LIKHACHEV. 2005. Fast replanning for navigation in unknown terrain. Robotics, IEEE Transactions on, **21**(3):354–363.

KUSHMERICK, NICHOLAS, STEVE HANKS, and DANIEL S WELD. 1995. An algorithm for probabilistic planning. Artificial Intelligence, **76**(1):239–286.

LAPORTE, GILBERT, and SILVANO MARTELLO. 1990. The selective travelling salesman problem. Discrete applied mathematics, **26**(2):193–207.

LAPORTE, GILBERT, and INMACULADA RODRÍGUEZ MARTÍN. 2007. Locating a cycle in a transportation or a telecommunications network. Networks, **50**(1):92–108.

LAPORTE, GILBERT, and JORGE RIERA-LEDESMA. 2002. A BRANCH-AND-CUT ALGORITHM FOR THE UNDIRECTED TRAVELING PURCHASER PROBLEM. pp. 940–951.

LEE, JUNKYU, RADU MARINESCU, and RINA DECHTER. 2014. Applying marginal map search to probabilistic conformant planning: Initial results. *In* Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence.

LIPPMAN, STEVEN A, and JOHN MCCALL. 1976. The economics of job search: A survey. Economic inquiry, **14**(2):155–189.

LITTLE, IAIN, DOUGLAS ABERDEEN, and S THIÉBAUX. 2005. Prottle: A probabilistic temporal planner. *In* AAAI. http://www.aaai.org/Papers/AAAI/2005/AAAI05-187.pdf.

MOLDOVAN, TEODOR MIHAI, and PIETER ABBEEL. 2012. Risk aversion in markov decision processes via near optimal chernoff bounds. *In* NIPS, pp. 3140–3148.

NAKHOST, HOOTAN, JÖRG HOFFMANN, and MARTIN MÜLLER. 2010. Improving local search for resource-constrained planning. *In* Third Annual Symposium on Combinatorial Search.

NIKOLOVA, E, MATTHEW BRAND, and DR KARGER. 2006. Optimal Route Planning under Uncertainty. *In* ICAPS, pp. 131–140. http://www.aaai.org/Papers/ICAPS/2006/ICAPS06-014.pdf.

NIKOLOVA, EVDOKIA, and DAVID R KARGER. 2008. Route planning under uncertainty: The canadian traveller problem. *In* AAAI, pp. 969–974.

PAPADIMITRIOU, CHRISTOS H, and MIHALIS YANNAKAKIS. 1989. Shortest paths without a map. *In* Automata, Languages and Programming. Springer, pp. 610–620.

PEARN, W.L., and R.C. CHIEN. 1998. Improved solutions for the traveling purchaser problem. Computers & Operations Research, **25**(11):879–885. ISSN 03050548. . http://linkinghub.elsevier.com/retrieve/pii/S030505489800032X.

POLYAKOVSKIY, SERGEY, MOHAMMAD REZA, MARKUS WAGNER, ZBIGNIEW MICHALEWICZ, and FRANK NEUMANN. 2014. A comprehensive benchmark set and heuristics for the traveling thief problem. *In* Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Vancouver, Canada.

POLYCHRONOPOULOS, GEORGE H, and JOHN N TSITSIKLIS. 1996. Stochastic shortest path problems with recourse. Networks, **27**(2):133–143.

PROSSER, PATRICK. 1996. An empirical study of phase transitions in binary constraint satisfaction problems. Artificial Intelligence, **81**(1):81–109.

ROCHLIN, IGOR, and DAVID SARNE. 2013. Information sharing under costly communication in joint exploration. *In* Proceedings of AAAI, pp. 847–853. http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/viewPDFInterstitial/6224/7281.

STENTZ, ANTHONY. 1995. The focussed D* algorithm for real-time replanning. *In* IJCAI, (August). http://www.cs.cmu.edu/ motionplanning/papers/sbp_papers/s/stentz_D2.pdf.

STERN, RONI, ARIEL FELNER, JUR VAN DEN BERG, RAMI PUZIS, RAJAT SHAH, and KEN GOLDBERG. 2014. Potential-based bounded-cost search and anytime non-parametric a. Artificial Intelligence, **214**:1–25.

TAIG, RAN, and RONEN I BRAFMAN. 2014. A relevance-based compilation method for conformant probabilistic planning. *In* Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 2374–2380.

TAIG, RAN, and RONEN I BRAFMAN. 2013. Compiling conformant probabilistic planning problems into classical planning. *In* Twenty-Third International Conference on Automated Planning and Scheduling.

TEENINGA, A., and A. VOLGENANT. 2004. Improved heuristics for the traveling purchaser problem. Computers & Operations Research, **31**(1):139–150. ISSN 03050548. . http://linkinghub.elsevier.com/retrieve/pii/S0305054802001934.

VANSTEENWEGEN, PIETER, WOUTER SOUFFRIAU, and DIRK VAN OUDHEUSDEN. 2011. The orienteering problem: A survey. European Journal of Operational Research, **209**(1):1–10.

WAGSTAFF, KL, DR THOMPSON, W ABBEY, A ALLWOOD, DL BEKKER, NA CABROL, T FUCHS, and K ORTEGA. 2013. Smart, texture-sensitive instrument classification for in situ rock and layer analysis. Geophysical Research Letters, **40**(16):4188–4193.

WEITZMAN, MARTIN L. 1979. Optimal search for the best alternative. *In* Econometrica: Journal of the Econometric Society, pp. 641–654.

WERNER, FRANK. 2002. Scheduling under Uncertainty. *In* econ.kuleuven.be, pp. 1–4. http://www.econ.kuleuven.be/eng/tew/academic/prodbel/PMS2012/pdf file talk werner[1].pdf.

WETTERGREEN, DAVID, GREYDON FOIL, MICHAEL FURLONG, and DAVID R THOMPSON. 2014. Science autonomy for rover subsurface exploration of the atacama desert. AI Magazine, **35**(4):47–60.