Algorithms for Stochastic Physical Search on General Graphs

Daniel S. Brown and **Jeffrey Hudack**
AFRL Information Directorate
Rome, NY 13441
{daniel.brown.81, jeffrey.hudack}@us.af.mil

Bikramjit Banerjee
University of Southern Mississippi
Hattiesburg, MS 39402
bikramjit.banerjee@usm.edu

Abstract

Stochastic Physical Search (SPS) refers to the search for an item in a physical environment where the item's price is stochastic, and where the cost to obtain the item includes both travel and purchase costs. This type of problem models task planning scenarios where the cost of completing an objective at a location is drawn from a probability distribution, reflecting the influence of unknown factors. Prior work on this domain has focused on solutions where the *expected* cost is minimized. Recently, SPS problems with other objectives have been proposed and theoretically analyzed, in particular when either the budget or the desired probability of success is fixed. However, general optimal solvers for these new variants do not yet exist. We present algorithms for optimal solution of these variants on general graphs. We formulate them as mixed integer linear programming problems, and solve them using an off-the-shelf MILP solver. We then develop custom branch and bound algorithms which result in a dramatic reduction in computation speed. Using these algorithms, we generate empirical insights into the hardness landscape of the fixed budget and fixed probability of success SPS variants.

Introduction

For agents operating in both real-world and virtual environments, completing a set of objectives may require costly exploration to dispel uncertainty before selecting from available options. These types of problems appear in many different domains. Shoppers may need to pay for gas or public transportation in order to visit several stores to find the best deal on a commodity. Utility companies searching for damaged equipment when service is lost will need to generate a search path that minimizes fuel costs and considers the uncertain amount of equipment and time needed for repairs. A search-and-rescue plane looking for a missing airliner at various locations (with varying likelihoods of success) in an ocean will want to optimize its chance of success while minimizing cost (fuel, time) and risk (local weather).

We refer to scenarios such as these as Stochastic Physical Search (SPS). In these situations, a priori information can provide estimates of the expected cost of achieving an objective at a given location. However, the actual cost of completing an objective at a location is drawn from a probability distribution, reflecting the influence of unknown factors. Additionally, this uncertainty can only be removed through actually visiting locations, thus incurring an exploration or travel cost. This work provides solution methods with increased flexibility and robustness to deal with uncertainty, which is an inevitable feature of operating in the real world.

Previous work on physical search with uncertainty has developed exact and heuristic solutions that minimize *expected cost* (Kang and Ouyang 2011). This works well for environments where we seek to meet objectives on a recurring basis. However, for mission-critical applications where we only get one chance to succeed, it is important to generate solutions that accommodate for limited resources and an acceptable risk of failure. This viewpoint is shared by Hazon, et al. in their recent work on physical search with probabilistic knowledge (Hazon et al. 2013), where they provide a thorough examination of the complexity and methodology for generating solution of this form on a *path graph*. Applying their problem formulation to find optimal solutions on general graphs is the focus of our research.

There are two key components of a solution to an SPS instance: budget and probability of success. 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 we seek a *Min-Budget* path that minimizes the budget required to meet that threshold. We can also combine both of these objectives into the overall goal of finding a *Min-Expected-Cost* path.

The contributions of this work are: 1) A generalization of Stochastic Physical Search to general graphs, 2) A mixed-integer linear programming formulation of the *Min-Budget* and *Max-Probability* SPS variants on complete graphs, 3) branch and bound algorithms that find exact solutions for these variants, and 4) An empirical investigation of the hardness landscape of these problem variants on random graphs.

Problem formulation

A stochastic physical search problem is defined by a graph $G(S^+, E)$ with a set of sites $S^+ = S \cup \{o, d\}$ where $S = \{s_1, ..., s_m\}$ is the set of m sites offering an item of interest, o and d are the origin and destination locations, and $E \subseteq$

$S^+ \times S^+$ is the set of edges. We define $adj(i) = \{j : (i, j) \in E\}$. 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, visit a subset of points in S to purchase a item, and then end at the destination point d. It is assumed that item cannot be obtained at the origin and destination sites.

The cost of purchasing 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 is a finite number of possible costs in the support of $P_i(c), \forall i \in S$.

The goal is to find an acyclic path that meets one of three objective functions (Hazon et al. 2013)

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

Where for a path $< v_1, v_2, \ldots, v_k >$ with $v_i \in S^+$ and $v_1 = o$, $v_k = d$, the probability of success is $p_{succ} = 1 - \prod_i Pr(\text{failure given budget } b_i \text{ at } v_i)$ and $b_{i+1} = b_i - t_{v_i, v_{i+1}}$ $(i = 1, \ldots, k-1)$.

We note that in the *Min-Budget* and *Max-Probability* variants it is assumed that the agent will purchase the item at the first available opportunity (i.e. when its budget at a site is $\geq$ the revealed price at that site) and terminate search. In these cases the destination site acts simply as a marker for the end of search and we assign a zero cost edge leading from each site offering the item to the destination site.

Intuitively, it may seem that a solution that minimizes expected cost could also be used to provide optimal solutions to the fixed budget and fixed probability of success formulations for SPS. We provide a simple counterexample in Figure 1, where s_1 is the start site and we seek to determine a strategy that meets the desired objectives. Each site has a probability distribution over the cost of the item at that site and a fixed travel cost exists between all sites in the graph. We assume the destination node, d, is connected to s_2, s_3, s_4 with zero cost edges.

We iterate over all possible paths that visit all sites in the network, and show the values for each formulation in Table 1. As a baseline, we compute cost of each path, as would be used in a TSP (open-tour) solution. Even though all sites have the same expected cost, the *shortest path* cost is clearly not equivalent to the *Min-Expected-Cost* results. To further motivate our focus on the bounded cases of *Min-Budget* and *Max-Probability*, we also show that the minimum expected cost values do not provide the respective minimum and maximum values for these formulations. This example highlights the differences between these problems, and justifies new solution concepts for cases when the objective is to maximize the probability of success with a limited budget, or to minimize budget with a fixed probability of success, and when an average or expected cost approach is unwanted, given the non-repetitive nature of the task.

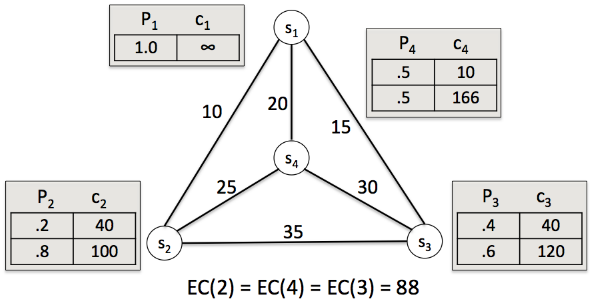
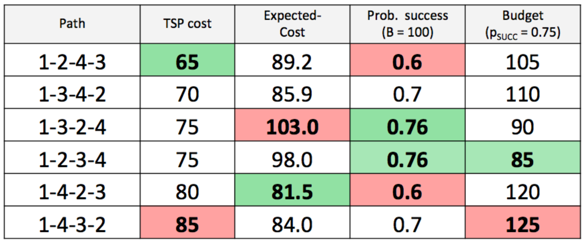


Figure 1: A simple SPS problem with 4 sites. For each site i, there is a set of costs C_i, and their associated probability P_i. The agent starts at site s_1 and must determine a path through the graph that meets a given objective function. All sites have the same expected cost of 88.



Path	TSP cost	Expected-Cost	Prob. success (B = 100)	Budget (p_{succ} = 0.75)
1-2-4-3	**65**	89.2	**0.6**	105
1-3-4-2	70	85.9	0.7	110
1-3-2-4	75	**103.0**	**0.76**	90
1-2-3-4	75	98.0	**0.76**	**85**
1-4-2-3	80	**81.5**	**0.6**	120
1-4-3-2	**85**	84.0	0.7	**125**

Table 1: The resulting values for different problem formulations based on the example in Figure 1. We consider TSP minimum length path, the expected cost, the probability of success given a maximum budget of 100, and the budget required to achieve a probability of a success of at least 0.75. Each of these formulations returns different best (green) and worst (red) values, showing they are not equivalent.

Related Work

There is a significant and varied body of work relating to search problems, beginning with the "secretary problem" (Ferguson 1989) and the well-known Traveling Salesman Problem (TSP); see (Gutin and Punnen 2002) for a comprehensive review. There are many variants of the TSP, such as the Traveling Repairman Problem (Afrati et al. 1986), the k-TSP (Arora and Karakostas 2000), the Generalized TSP (GTSP) (Snyder and Daskin 2006), and the Traveling Purchaser Problem (TPP) (Ramesh 1981; Singh and van Oudheusden 1997; Laporte, Riera-Ledesma, and Salazar-González 2003), all of which have attracted researchers in both operations research and artificial intelligence. However, the majority of this work assumes the costs involved are fixed and known.

A number of related works have introduced stochastic or unknown travel and item costs, an early example being Pandora's Problem (Weitzman 1979), which solves for the threshold on cost for choosing to open boxes with stochastic rewards. The Stochastic TSP, which has stochastic travel costs, has been solved for maximizing probability on instances with fixed time windows (Carraway, Morin, and Moskowitz 1989). Finally, the Orienteering Problem with Stochastic Profits (OPSP) introduces a distribution over the profits at each site and seeks to meet a profit threshold within a given time limit (Tang and Miller-Hooks 2005; Campbell, Gendreau, and Thomas 2011).

The *Min-Expected-Cost* problem variant has recently been solved for general graphs (Kang and Ouyang 2011). Kang et al. formulate this problem as a Traveling Purchaser Problem with stochastic prices, where an agent must find a minimum-expected-cost path between markets to purchase a set of required commodities. This work does not consider risk or the probability of success and assumes an unbounded budget. Our work complements the work of Kang et al. by providing solvers for the *Min-Budget* and *Max-Probability* problem variants on general graphs.

The concept of a physical search problem and the *Min-Expected-Cost*, *Min-Budget*, and *Max-Probability* variants were introduced by Aumann et al. and Hazon et al. (Aumann et al. 2008; Hazon, Aumann, and Kraus 2009; Hazon et al. 2013). They 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 these variants when 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. Our work extends this work by providing, to our knowledge, the first exact solvers for the *Min-Budget* and *Max-Probability* variants on general graphs. We further provide the first empirical insights into the hardness landscapes of the *Min-Budget* and *Max-Probability* problem instances.

MILP Formulation

Let the possible values of the item's cost at site $i \in S$ be $\{c_{i,1}, c_{i,2}, \dots c_{i,\rho}\}$, given in increasing order. This induces a set of exclusive cost intervals at i, $R_i = \{[c_{i,y}, c_{i,y+1}) \mid y = 0, 1, \dots \rho\}$, where $c_{i,0} = 0,\ c_{i,\rho+1} = \infty$. We assume that all sites have ρ 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 purchased 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 $0 \leq y \leq \rho$. Then $\alpha^1_{ir} = 1$ iff $b_i < c_{i,y}$, and $\alpha^2_{ir} = 1$ iff $b_i \geq c_{i,y+1}$. In other words, $\alpha^1_{ir} = \alpha^2_{ir} = 0$ iff $b_i \in [c_{i,y}, c_{i,y+1})$.

Due to the requirement of linearity, we use log probabilities to enable summations rather than products. 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 $p_{ir} = \log(\sum_{y<x\leq\rho} P_i(c_{i,x}))$.

Min-Budget

The formulation of the min budget problem is as follows:

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

subject to

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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. 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}. Constraints (11) represent the conditional constraint

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

$\forall i \in S, r = [c_{i,y}, c_{i,y+1}) \in R_i$. These constraints ensure that the log probability of failure at site i is set to the correct value based on the available budget b_i. The value δ is a small constant required because of the non-inclusive upper bound, $c_{i,y+1}$, in the cost interval $[c_{i,y}, c_{i,y+1})$. Finally, constraints (12) – (15) ensure that all decision variables are in the correct ranges.

Max-Probability

To obtain the *Max-Probability* version we can simply 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 nice formalism and can be solved using off-the-shelf solvers to provide baseline solutions. However, it is often the case that custom branch-and-bound techniques 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*.

Branch-and-Bound Formulations

Due to space constraints, we detail only 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.

We define a 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

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

Min-Budget

State representation 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 $\langle \langle o \rangle, [0, \infty), 1.0 \rangle$.

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

$$[l_j, u_j) = [\max(l_i - t_{ij}, c_{j,x}), \min(u_i - t_{ij}, c_{j,x+1})) \tag{16}$$

and $p' = p \cdot f_j(l_j)$. If $[l_j, u_j)$ is empty, then the corresponding successor can be discarded.

When a terminal state (one with no successor) $\langle \pi, [l_i, u_i), p \rangle$ is reached, it is a *feasible solution* if $p \le 1 - p^*_{succ}$, 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

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

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

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

(2) or the following holds:

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

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

Proof: We have the following:

$$\begin{aligned}
\sigma_i &= l_i + \sum_{h=0}^{|\pi|-2} t_{\pi_h, \pi_{h+1}} \\
&= (l_i - t_{ij}) + \sum_{h=0}^{|\pi|-2} t_{\pi_h, \pi_{h+1}} + t_{ij} \\
&= l_i - t_{ij} + \sum_{h=0}^{|\pi'|-2} t_{\pi'_h, \pi'_{h+1}}, \quad \because \pi \equiv \pi' \text{ up to } i \\
&\le l_j + \sum_{h=0}^{|\pi'|-2} t_{\pi'_h, \pi'_{h+1}}, \text{ by equation 16} \\
&= \sigma_j. \qquad \square
\end{aligned}$$

In words, the above proposition shows that the minimum initial budget increases monotonically 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: (by contradiction) If possible, let $\langle \pi', [l_z, u_z), p' \rangle$ be a feasible solution (i.e., terminal state with $p' \leq 1 - p^*_{succ}$) 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., $|\pi| \leq k \leq |\pi'| - 1$. By equation 16, the upper end of the budget range in the corresponding state is

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

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

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

where the penultimate step considers all sites $j \in S \setminus S_\pi$ regardless of whether j is on the path π' or not, since $f_j(\cdot) \leq 1$. The above is a contradiction. □

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 time $O(|E| + |S| \log |S|)$.*

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

$$B^* = \min_{i \in \mathcal{S}} \left(\sum_{h=0}^{|\hat{\pi}^i|-2} t_{\hat{\pi}^i_h, \hat{\pi}^i_{h+1}} + c_{i,\rho} \right). \qquad (17)$$

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). □

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 $j \in adj(i) \setminus N_\pi$ we create a potential successor state as $\langle \pi', b_i - t_{ij}, p' \rangle$ where π' is as defined in the *Min-Budge*t formulation and $p' = p \cdot f_j(b_i - t_{ij})$. The successor is discarded if $t_{ij} \geq b_i$. 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 purchased 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

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

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

Experimental Results and Analysis

To investigate the hardness landscape of the *Min-Budget* and *Max-Probability* problems we generated random problem instances with various numbers of sites each having two possible costs. Both costs were chosen random uniformly from the interval $[1, 100]$. The probability of the lower cost p was chosen randomly and the probability of the higher cost was then assigned to $1 - p$. Symmetric edge costs between every two sites were chosen random uniformly from the interval $[1, 100]$, except for edge costs to the destination site d, which were set to 0.

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, Mesnier, and Moré 1998; Dolan 2001; Gropp and Moré 1997). We generated SPS problems with the number of site $|S|$ ranging from 2 to 9 sites 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. The results for solving *Min-Budget* with $P_{succ} = 0.75$ and *Max-Probability* with $B^* = 50$ are shown in Figure 2 (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. Based on these results we chose to restrict our remaining analysis to the branch-and-bound algorithms.

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

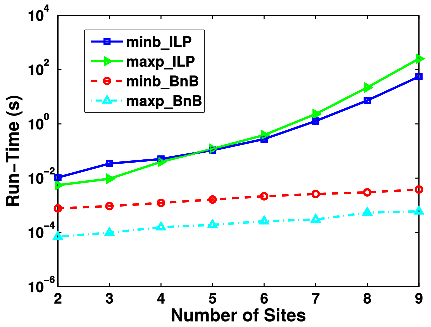


Figure 2: Comparison of average run-times for the MILP and branch-and-bound algorithms.

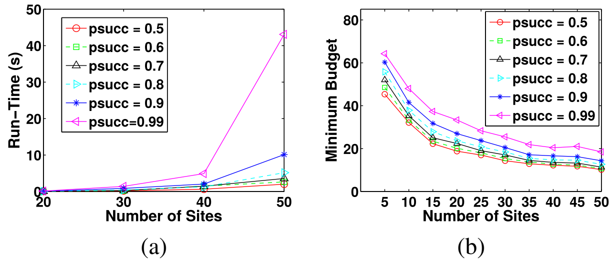


Figure 3: (a) Average run-time for the *Min-Budget* branch-and-bound algorithm. (b) Average minimum budget results from the *Min-Budget* branch-and-bound algorithm.

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 3(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 3(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 purchasing the item at a low price. Additionally, as expected we see that higher required probabilities of success require higher budgets.

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 4(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 terminate the search before exploring all branches.

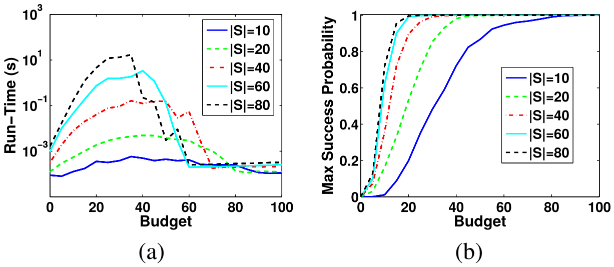


Figure 4: (a) Average run-time for the *Max-Probability* branch-and-bound algorithm. (b) Average maximum p_{succ} results from the *Max-Probability* branch-and-bound algorithm.

Figure 4(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.

Conclusions

We provided a generalization of Stochastic Physical Search to general graphs. To the best of our knowledge, this work provides the first exact algorithmic solutions to the *Min-Budget* and *Max-Probability* variants on general graphs. We first formulated the problem as a mixed-integer linear program. This provides a theoretical formalism and bench-mark from which we developed custom branch-and-bound algorithms to find the exact solutions for the *Min-Budget* and *Max-Probability* formulations. The results for the branch-and-bound algorithm show that taking advantage of the problem structure and knowledge of the cost profiles of unvisited sites allow much faster execution times.

We also generated empirical insights into the hardness landscape of the variants. Based on these results, we see that while similar, the *Min-Budget* and *Max-Probability* problems have interesting characteristics. Both exhibit worst case exponential complexity; however, we see that the size of problem and the starting budget and required probability of success have a large impact on tractability. In problems where a high probability of success is required, but a perfect solution cannot be found, both problems become intractable for large numbers of sites. Thus, while we have developed the first ever exact algorithms for the *Min-Budget* and *Max-Probability* SPS variants, heuristic and approximate solutions will be needed to deal with these difficult problem instances. Future work also includes extending this work to the multi-item and multi-agent cases on general graphs, investigating other methods of problem generation, and exploring real world data sets.

References

Afrati, F.; Cosmadakis, C.; Papadimitriou, G.; Papageorgiou, N.; and Papakostantinou. 1986. The complexity of the traveling repairman problem. *Theoret. Inform. Appl. 20* 79–87.

Arora, S., and Karakostas, G. 2000. A 2+ ε approximation algorithm for the k-mst problem. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, 754–759. Society for Industrial and Applied Mathematics.

Aumann, Y.; Hazon, N.; Kraus, S.; and Sarne, D. 2008. Physical search problems applying economic search models. In *AAAI*, 9–16.

Campbell, A. M.; Gendreau, M.; and Thomas, B. W. 2011. The orienteering problem with stochastic travel and service times. *Annals of Operations Research* 186(1):61–81.

Carraway, R. L.; Morin, T. L.; and Moskowitz, H. 1989. Generalized dynamic programming for stochastic combinatorial optimization. *Operations Research* 37(5):819–829.

Czyzyk, J.; Mesnier, M. P.; and Moré, J. J. 1998. The neos server. *Computing in Science and Engineering* 5(3):68–75.

Dolan, E. D. 2001. Neos server 4.0 administrative guide. *arXiv preprint cs/0107034*.

Ferguson, T. S. 1989. Who solved the secretary problem? *Statistical science* 282–289.

Fredman, M. L., and Tarjan, R. E. 1987. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)* 34(3):596–615.

Gropp, W., and Moré, J. 1997. Optimization environments and the neos server. *Approximation theory and optimization* 167–182.

Gutin, G., and Punnen, A. P. 2002. *The traveling salesman problem and its variations*, volume 12. Springer.

Hazon, N.; Aumann, Y.; and Kraus, S. 2009. Collaborative multi agent physical search with probabilistic knowledge. In *Twenty-First International Joint Conference on Artificial Intelligence*.

Hazon, N.; Aumann, Y.; Kraus, S.; and Sarne, D. 2013. Physical search problems with probabilistic knowledge. *Artificial Intelligence* 196:26–52.

Kang, S., and Ouyang, Y. 2011. The traveling purchaser problem with stochastic prices: Exact and approximate algorithms. *European Journal of Operational Research* 209(3):265–272.

Laporte, G.; Riera-Ledesma, J.; and Salazar-González, J.-J. 2003. A branch-and-cut algorithm for the undirected traveling purchaser problem. *Operations Research* 51(6):940–951.

Ramesh, T. 1981. Traveling purchaser problem. *Opsearch* 18:78–91.

Singh, K. N., and van Oudheusden, D. L. 1997. A branch and bound algorithm for the traveling purchaser problem. *European Journal of Operational Research* 97(3):571–579.

Snyder, L. V., and Daskin, M. S. 2006. A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research* 174(1):38–53.

Tang, H., and Miller-Hooks, E. 2005. A tabu search heuristic for the team orienteering problem. *Computers & Operations Research* 32(6):1379–1407.

Weitzman, M. L. 1979. Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society* 641–654.