A Polynomial Time Optimal Algorithm for Robot-Human Search under Uncertainty

Shaofei Chen[1], Tim Baarslag[2], Dengji Zhao[2], Jing Chen[1], Lincheng Shen[1]
[1] College of Mechatronics and Automation, National University of Defense Technology, China.
[2] Electronics and Computer Science, University of Southampton, UK.

Abstract

This paper studies a search problem involving a robot that is searching for a certain item in an uncertain environment (e.g., searching minerals on Moon) that allows only limited interaction with humans. The uncertainty of the environment comes from the rewards of undiscovered items and the availability of costly human help. The goal of the robot is to maximize the reward of the items found while minimising the search costs. We show that this search problem is polynomially solvable with a novel integration of the human help, which has not been studied in the literature before. Furthermore, we empirically evaluate our solution with simulations and show that it significantly outperforms several benchmark approaches.

1 Introduction

Robots are increasingly used to perform search tasks, especially under the extreme environments that are hard to approach. Furthermore, there is often uncertainty about the nature of the environment which is another barrier stopping humans directly access the environment [Liu and Nejat, 2013].

For example, a rover may search valuable minerals underneath the ground of a planet and the quality of each mineral is uncertain before revealing it. In a disaster response scenario, an unmanned ground vehicle (UGV) may look for an important piece of plane debris in a flight crash region. In such scenarios, it is costly for the robot (e.g., the energy/time to reveal a mineral or the risk to visit a location) to become aware of the exact reward of a search option (e.g., the quality of a mineral or whether the expected piece of plane debris can be found at a certain location). Crucially, humans may help robots to reduce the uncertainty of the rewards of each search option (e.g., provide some information based on snapshots and crowdsourced reports about the environments). However, humans may not always available and asking humans for help might be costly as well [Fogarty *et al.*, 2005; Schmidt-Rohr *et al.*, 2008].

Given the uncertainties of both the environment and the availability of human help, the challenges in planning the actions of the robot are twofold. Firstly, at each time step, the robot needs to decide whether to search to acquire more knowledge about the environment or to select one of the known options to complete the search. Secondly, to reveal the value of a search option/action, the robot needs to decide to whether ask human for help or to discover it by itself. Therefore, for optimal decision making, the robot should balance the values of different actions.

Previous work in human-robot community has mainly considered the architecture mechanisms of interactions between humans and robots [Murphy, 2004; Nourbakhsh *et al.*, 2005; Goodrich and Schultz, 2007; Gao *et al.*, 2014]. Recently, state-of-the-art planning technologies, both under certainty and uncertainty, have been adapted to decision making for human and robot teams [Bresina and Morris, 2007; Talamadupula *et al.*, 2010; Rosenthal and Veloso, 2011; 2012; Rosenfeld *et al.*, 2015]. However, algorithms for these general planning models cannot be effectively applied to our search problem which has many (possibly infinite number of) possible values on the reward of each action.

To combat this challenge, we focus on a class of the robot search problem where costs of visiting potential locations are independent with each other, and expect to find polynomial time optimal solutions. Compared with general models in [Rosenthal and Veloso, 2011; 2012], this independence assumption narrows the generality of our model, but it still covers an important part of problems regularly faced in robot search. For example, an autonomous rover explores among items that are not far from each other where the energy/time cost of physical movements is much less than that of revealing minerals, or an UGV looks for an important piece of plane debris where the more important cost is the risk of visiting the search area and these risks at different locations can be regarded independent with each other.

Against this background, we propose a new model of *robot-human search* (RHS), and propose a novel optimal algorithm to solve it. Specifically, similar with other search or planning approaches [Hazon *et al.*, 2013; Kang and Ouyang, 2011; Rosenthal and Veloso, 2011; 2012], our robot's decision making problem can also be formulated as a dynamic programming instance. Instead of using their typical approximation algorithms as a solution concept, we design an index based search algorithm. In more detail, we define indices for each possible action and design a search rule that always executes the action with the highest index. In particular, we show that, given the assumption of independence of costs, our index based search algorithm is polynomial time complex and provably optimal. This paper advances the state-of-the-art in the following ways:

- We propose a new formal model of a robot searches an item with uncertain knowledge of the environment and costly human help. The model not only accounts for searching with proactively asking human help, but also explicitly captures the uncertainties of both the rewards of items and the availability of human.
- We design a polynomial time algorithm to solve the search problem and theoretically prove its optimality. Compared with the algorithms proposed in [Rosenthal and Veloso, 2011; 2012] for more general settings, our approach improves the computation significantly.
- Furthermore, we empirically evaluate our search solution in simulations and show that it significantly outperforms several benchmark approaches.

2 Formal Model

In this section, we first present the model of *robot-human search* (RHS) and then give an example to explain it.

We consider a robot searching for an item among n designated locations in an environment with human help and the interactions among the robot, the human and the environment are shown in Figure 1 (a). For each location i, $1 \leq i \leq n$, there is an item with a potential reward x_i with probability distribution function $F_i(x_i)$, and the rewards of these items are independent with each other. The reward of each item is uncertain (a priori), and the robot can observe the true reward by revealing the item or asking humans to check it. The revealing cost is denoted by c_i^{reveal}, while asking humans for help takes the robot a cost c_i^{ask}. We denote availability p as the probability that a human may provide the expected information when the robot is asking for help. The robot keeps on revealing these items with the help from the human, and then selects one of the revealed items to collect. Therefore, the goal of the robot is to maximize the reward of the obtained item while minimizing the sum of search costs.

Specifically, we define the states of an item, which are illustrated in Figure 1 (b). All possible states of an item are denoted by (Unknown, Unavailable), (Known, Unavailable) and (Known, Available), where Known/Unknown indicates that the reward of an item is Known/Unknown by the robot, and Available/Unavailable is that the item could be collected or not yet. Initially, an item is at the state of (Unknown, Unavailable), which means its reward is unknown and the item can not be obtained unless some physical barriers are removed.[1] First, it costs the robot c_i^{reveal} to *reveal* an item and make this item known and available for the robot, i.e., (Unknown, Unavailable) → (Known, Available). Then, the robot may select one of known items to *collect*, which also means that the search task is ended. Moreover, the human may provide the robot some knowledge of the reward of an item to the robot. In practice, the human is not always available or interruptible [Fogarty *et al.*, 2005; Rosenthal *et al.*, 2012] and can not always respond. It takes the robot a cost c^{ask} to *ask* the human for help. For example, it takes 2.7 seconds for transmitting a signal from Moon to Earth and 14 minutes from Mars to Earth. If the human is available, she then *checks* the reward of the item, i.e., (Unknown, Unavailable) → (Known, Unavailable); otherwise, the state does not change. In addition, for a checked item at location i, the robot still have to pay the cost c_i^{reveal} to make it available to collect, i.e., (Known, Unavailable) → (Known, Available).

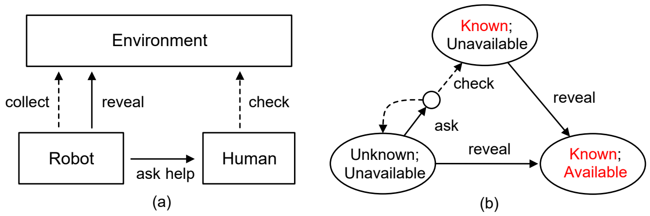

Figure 1: (a) The interactions among the robot, the human and the environment. (b) The transition states of an item.

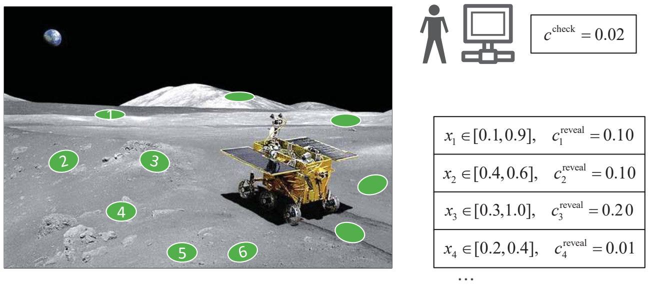
1
2
3
4
5
6
$c^{\text{check}} = 0.02$
$x_1 \in [0.1, 0.9], \quad c_1^{\text{reveal}} = 0.10$
$x_2 \in [0.4, 0.6], \quad c_2^{\text{reveal}} = 0.10$
$x_3 \in [0.3, 1.0], \quad c_3^{\text{reveal}} = 0.20$
$x_4 \in [0.2, 0.4], \quad c_4^{\text{reveal}} = 0.01$
...

Figure 2: A rover searches an mineral on Moon with human help, where green spots are the search locations.

We now provide a simple example to explain how the robot would operate in this scenario. As shown in Figure 2, an autonomous rover searches to collect a mineral on the land of Moon with the help from a person who is stayed on Earth. In order to assess the amount of reward x_i at a specific location i, the robot is able to ask help from the person with a cost c^{ask} or physically excavate the covered soil and measure the item directly by itself with a cost c_i^{reveal}. The parameters of the minerals at different locations are listed in Figure 2. By analysing this, we can find that different actions with different minerals should be handled differently. For example:

- The expected utilities of mineral 1 and mineral 2 are the same. For instance, if the robot reveals x_1 and it turns out to be 0.9 (higher than the supremum of x_2), there is no need to visit mineral 2 anymore. However, this does not hold for the action of revealing mineral 2. Therefore, even for two items with equal expected utilities, different sequences of revealing them may have different results.
- Compared with mineral 2, mineral 3 associates with a higher expected reward and a higher reveal cost. If the robot asks the person to check x_3 and it turns out to be 0.6, the robot should give up mineral 3 (because it will

[1] Note that, in this paper, "available" of an item indicates that the item is ready and a robot can collect it at any time, while "available" of a human means that the human can help a robot at a time step.

be $x_3 - c_3^{\text{reveal}} \leq x_2 - c_2^{\text{reveal}}$, i.e., $\forall x_2 \in [0.4, 0.6], 0.6 - 0.2 \leq x_2 - 0.1$). Thus, a cost 0.2 of revealing mineral 3 has been avoided. Therefore, checking an item before revealing it may reduce the total search cost.

- For a mineral with a reveal cost that less than $c^{\text{ask}} = 0.02$, such as mineral 4 with $c_4^{\text{reveal}} = 0.01$, the action of the robot revealing x_4 is prior to asking the person to help to check x_4. Therefore, it is unhelpful to check an item with a less reveal cost.

The robot's utility is thus subject to not only the rewards of items but also the costs of actions. Consequently, a robot's strategy should maximize the overall benefit resulting from the search process, defined as the value of the option eventually collected, minus the costs accumulated along the process, rather than merely finding the best valued option.

Having defined RHS, we now need an approach for the robot to plan its search actions based on the current states and knowledge of items. Hence, in what follows, we propose a dynamic programming formulation for the robot's decision making and then design algorithms to solve it.

3 Robot's Dynamic Programming Model

Given the proposed formal model, in this section, we formulate the robot's decision making in RHS as a dynamic programming problem.

We denote the collection of the n items by $I = \{1, 2, \ldots, n\}$ and partition I into two sets: a growing set of known items $S \subseteq I$ and its complement $\bar{S}$ of unknown items.

At each time, the robot may choose whether to *ask* the human to check an unknown item from $\bar{S}$, to autonomously *reveal* an unknown item from $\bar{S}$, or to stop the search and select one of the known items from S to *collect*.

First, we can denote the *collect reward* of collecting a known item $i \in S$ as follows:

$$r_i = \begin{cases} x_i & \text{if } i \text{ was revealed by robot,} \\ -c_i^{\text{reveal}} + x_i & \text{if } i \text{ was checked by human.} \end{cases} \tag{1}$$

When the robot decides to collect an item, its optimal strategy is simply to select the item with the highest current known collect reward:

$$y = \max_{i \in S} r_i. \tag{2}$$

We then denote the state at any time by $(\bar{S}, y)$ and define $\Psi(\bar{S}, y)$ as the expected present value of following an optimal policy from this time on when the set of unknown locations is $\bar{S}$ and the maximum known reward is y. Note that, we do not explicitly incorporate the availability of the human into the state. Instead, we consider the human's uncertain availability in the recursive relation of valuation function of each state.

For each subset $\bar{S}$ and y, the valuation function must satisfy the fundamental recursive relation:

$$\Psi(\bar{S}, y) = \max\left\{y, \Psi^{\text{reveal}}(\bar{S}, y), \Psi^{\text{ask}}(\bar{S}, y)\right\}, \tag{3}$$

where

$$\begin{aligned} \Psi^{\text{reveal}}(\bar{S}, y) = \max_{i \in \bar{S}} & \left\{ -c_i^{\text{reveal}} + \Psi(\bar{S} - \{i\}, y) \int_{-\infty}^{y} \mathrm{d}F_i(x) \right. \\ & \left. + \int_{y}^{\infty} \Psi(\bar{S} - \{i\}, x) \mathrm{d}F_i(x) \right\}, \end{aligned}$$

$$\begin{aligned} \Psi^{\text{ask}}(\bar{S}, y) &= p\hat{\Psi}^{\text{ask}}(\bar{S}, y) + (1-p)\left(\Psi^{\text{ask}}(\bar{S}, y) - c^{\text{ask}}\right) \\ &= \hat{\Psi}^{\text{ask}}(\bar{S}, y) - \frac{(1-p)c^{\text{ask}}}{p} \\ &= \max_{i \in \bar{S}} \left\{ -\frac{c^{\text{ask}}}{p} + \Psi(\bar{S} - \{i\}, y) \int_{-\infty}^{y} \mathrm{d}F_i^{\text{ask}}(x) \right. \\ & \quad \left. + \int_{y}^{\infty} \Psi(\bar{S} - \{i\}, x) \mathrm{d}F_i^{\text{ask}}(x) \right\}, \end{aligned}$$

where $\Psi^{\text{reveal}}(\bar{S}, y)$ and $\Psi^{\text{ask}}(\bar{S}, y)$ are the values of reveal and ask respectively, and $F_i^{\text{ask}}(x) = F_i(x + c_i^{\text{reveal}})$ is the cumulative distribution function for $x = x_i - c_i^{\text{reveal}}$, which stands for the reward that the robot obtains if it collects the item at location i. In more detail, for each state $(\bar{S}, y)$, the robot should compare the values of different actions.[2] For revealing an item, we should take the following into account:

- If reward $x \leq y$, y does not change and we have expected utility $-c_i^{\text{reveal}} + \Psi(\bar{S} - \{i\}, y)$;
- Otherwise, y is updated to x and the expected utility is $-c_i^{\text{reveal}} + \Psi(\bar{S} - \{i\}, x)$.

When asking the human to check an item, we should take the human availability into account as follows:

- If no help is available (with a probability $1 - p$), nothing changes and we have expected utility $-c_i^{\text{ask}} + \Psi(\bar{S}, y)$;
- Otherwise, we denote the value when the human is available as $\hat{\Psi}^{\text{ask}}(\bar{S}, y)$, whose recursive relation is similar with $\Psi^{\text{reveal}}(\bar{S}, y)$ as analysed above.

Thus, we can sum up the optimal solution for RHS as follows: for current state $(\bar{S}, y)$ of RHS, an optimal solution maximizes the value $\Psi(\bar{S}, y)$ that computed by Equation (3).

We have thus formulated the robot's decision making as a dynamic programming. However, in this form, the recursive value function is computationally intractable for problems with large n. Specifically, the computation time and storage requirement of this dynamic programming are the same as those for traditional travelling salesman problems with n visiting nodes, i.e., of complexity $O(n^2 2^n)$ [Bellman, 1962]. Therefore, we design an index-based policy that can optimally solve the problem in polynomial time in the next section.

4 Search Strategy

Inspired by Pandora's rule [Weitzman, 1979] for elicitation problems [Baarslag and Gerding, 2015], we define indices for each reveal and ask action. Specifically, for location i, we define two indices with respect to each reveal and ask action, which are denoted by *reveal index* z_i^{reveal} and *ask index* z_i^{ask} and computed by:

$$z_i^{\text{reveal}} = -c_i^{\text{reveal}} + z_i^{\text{reveal}} \int_{-\infty}^{z_i^{\text{reveal}}} \mathrm{d}F_i(x) + \int_{z_i^{\text{reveal}}}^{\infty} x \mathrm{d}F_i(x), \tag{4}$$

$$z_i^{\text{ask}} = -\frac{c^{\text{ask}}}{p} + z_i^{\text{ask}} \int_{-\infty}^{z_i^{\text{ask}}} \mathrm{d}F_i^{\text{ask}}(x) + \int_{z_i^{\text{ask}}}^{\infty} x \mathrm{d}F_i^{\text{ask}}(x). \tag{5}$$

[2] To note, in our model, c_i^{reveal} is considered as part of the action cost when the robot takes the reveal action whereas c^{ask} is the action cost when the robot asks the human. In the latter case, c_i^{reveal} has been put in as part of the collect reward function (Equation (1)).

Given state $(\bar{S}, y)$ and the set of indices $\{z_i^{\text{reveal}}, z_i^{\text{ask}} | i \in \bar{S}\}$, we design the following simple (but optimal) policy, called *Search Rule* as follows:

> **Search Rule**:
>
> ASK/REVEAL RULE: If a location is to be asked for help (or revealed by the robot), it should be an unknown location with the highest ask index (or reveal index) . Whether to ask the human to check or to reveal a location is determined by the overall highest index.
>
> COLLECT RULE: Terminate search whenever the maximum known collect reward exceeds both the ask index and reveal index of every unknown location. Then select the item with the highest known collect reward to collect.

Given the above search strategy, we next analyse its computational complexity and prove its optimality.

Theorem 1. *The complexity of the* Search Rule *is* $O(n \log n)$.

Proof. In Algorithm 1, as the robot chooses actions based on the orders of index values of all items and this order does not need to be updated during the search process, the complexity of our strategy depends on computing the order of these indices and it then is $O(n \log n)$. □

Theorem 2. *The* Search Rule *is an optimal strategy for RHS.*

Proof. To prove our *Search Rule* optimal for RHS, we follow the proof of [Baarslag and Gerding, 2015] to show that RHS can be mapped to Pandora's problem [Weitzman, 1979], which is an economic-based search model about opening boxes. In Pandora's problem, each closed box contains a potential reward with a probability distribution function and features a cost to open it and learn its contents. First, each known location with its item in S can be seen as an opened box with a reward r_i. Then, as shown in Figure 3, we can view each unknown location i holding two independent copies of boxes i^{reveal} and i^{ask}: i^{reveal} contains a potential reward x_i with the probability distribution function $F_i(x_i)$ and its opening cost is c_i^{reveal}; i^{ask} contains $x_i - c_i^{\text{reveal}}$ with the probability distribution function $F_i(x_i)$ and its opening cost is $\frac{c^{\text{ask}}}{p}$. Once a box is opened, it moves from the set of closed boxes $\bar{S}$ to the set of opened boxes S, and the other box at the same location is deleted from the world. The index based policy is as follows: (1) if a box is to be opened, it should be that closed box with highest index; (2) terminate search whenever the maximum sampled reward exceeds the index of any closed box. It has been proven in [Weitzman, 1979] that this type of strategy is optimal in terms of expected reward. Therefore, RHS is mapped to a Pandora's problem with boxes $\{i^{\text{reveal}}, i^{\text{ask}} \mid i \in I\}$ and our *Search Rule* is optimal in terms of expected reward (Equation (3)). □

Next, we design the algorithm (see Algorithm 1) of executing the search strategy. First, compute the highest collect reward of known items (line 2-3) and compute the reveal indices and ask indices of unknown items (line 4-7). Then, compare these indices (line 8) and select an item to collect (line 9-10), reveal (line 11-12) or ask for help (line 13-14). When asking for help, if the human is not available, it is not feasible to repeatedly execute ask actions until response is received at a time step because the availability always can not change immediately. Therefore, if the human is not available at a time step, the robot will then select a known item to collect or select an unknown one to reveal.

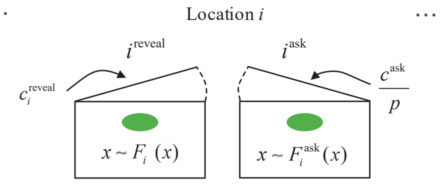

Figure 3: Mapping from a reveal or ask action to opening a box in Pandora's problem.

Algorithm 1 Executing search policy

```
 1: procedure SEARCH($S, \bar{S}$)
        ▷ Find the highest collect reward of known items.
 2:     $y = \max_{i \in S} r_i$
 3:     $\hat{i} = \arg\max_{i \in S} r_i$
        ▷ Calculate the reveal indices and ask indices.
 4:     for $i \in \bar{S}$ do
 5:         $z_i^{\text{reveal}} \leftarrow$ Solve $c_i^{\text{reveal}} = \int_{z_i^{\text{reveal}}}^{\infty} (x - z_i^{\text{reveal}}) \mathrm{d}F_i(x)$
 6:         $z_i^{\text{ask}} \leftarrow$ Solve $\frac{c^{\text{ask}}}{p} = \int_{z_i^{\text{ask}}}^{\infty} (x - z_i^{\text{ask}}) \mathrm{d}F_i^{\text{ask}}(x)$
 7:     end for
        ▷ Compare all indices and perform the optimal action.
 8:     $i^* = \arg\max_{i \in S} \max\{z_i^{\text{reveal}}, z_i^{\text{ask}}\}$
 9:     if $y \geq \max\{z_{i^*}^{\text{reveal}}, z_{i^*}^{\text{ask}}\}$ then
10:         return COLLECT($S, \bar{S}, \hat{i}$)
11:     else if $z_{i^*}^{\text{reveal}} \geq z_{i^*}^{\text{ask}}$ then
12:         return REVEAL($S, \bar{S}, i^*$)
13:     else
14:         return ASK($S, \bar{S}, i^*$)
15:     end if
16: end procedure
```

Given our designed *Search Rule* and Algorithm 1, we can derive some desirable properties of the optimal solution. Firstly, if the human with a higher availability p, the action of asking the human to check an item's reward then associates a higher index.

Property 1. *For any unknown item $i \in \bar{S}$, its ask index z_i^{ask} increases with the human availability p, while its reveal index z_i^{reveal} is independent of p.*

Secondly, we can derive that the reveal index and ask index satisfies a property as follows:

Property 2. *For any unknown item $i \in \bar{S}$, if $c_i^{reveal} < \frac{c^{ask}}{p}$, then $z_i^{reveal} > z_i^{ask}$, i.e., the action of asking the human to check i is dominated by the action of revealing it.*

Proof. We prove this property by reductio ad absurdum, i.e., if $z_i^{\text{reveal}} \leq z_i^{\text{ask}}$, then $c_i^{\text{reveal}} \geq \frac{c^{\text{ask}}}{p}$. Specifically, first, for any item with a potential reward x with probability distribution function $F(x)$, the function $\int_z^{\infty} (x - z) \mathrm{d}F(x)$ decreases with z (because its derivative function is $-(x - z)\dot{F}(x) \leq 0$, where $x \in \{z, \infty\}$). Next, if $z_i^{\text{reveal}} \leq z_i^{\text{ask}}$, then $z_i^{\text{reveal}} \leq$

$z_i^{\text{ask}} + c_i^{\text{reveal}}$. Then $\int_{z_i^{\text{reveal}}}^{\infty}(x - z_i^{\text{reveal}})\mathrm{d}F_i(x) - \int_{z_i^{\text{ask}}}^{\infty}(x - z_i^{\text{ask}})\mathrm{d}F_i(x + c_i^{\text{reveal}}) \geq 0$. Given this, from Equation (4) and (5), we get that $c_i^{\text{reveal}} \geq \frac{c^{\text{ask}}}{p}$. Thus, we derive that if $c_i^{\text{reveal}} < \frac{c^{\text{ask}}}{p}$, then $z_i^{\text{reveal}} > z_i^{\text{ask}}$. □

5 Experiments

To evaluate the performance of our *Search Rule* (called "Optimal" for short in this section) for RHS, we design five baseline benchmark strategies for comparison in terms of average utilities and interactions of robot, human and environment.

5.1 Setup and Benchmarks

We define the statistics as follows: (1) *Average utilities:* Given the objective defined in Section 2, the utility of an simulation is the reward of the obtained item minus the accumulated costs; (2) n_Ask: Average times that the robot tries to ask for help; (3) n_Check: Average times that the robot receives help from the human; (4) n_Reveal: Average number of items that revealed by the robot; (5) n_Known: Average number of known items, which are either revealed by the robot or checked by the human. In particular, we use average utilities generated by different algorithms to evaluate their performance and use other four statistics to analyse the interactions during search processes.

We then design two experiments as follows:

Experiment A: We first design scenarios of searching among 4 items based on the example defined in Section 2, and construct these scenarios by using their parameters. We vary the human availability p as a parameter of the experiments, choosing values between 0 and 1 with 0.05 increments.

Experiment B: We next design more general scenarios to evaluate the performance of different algorithms with varying reveal costs. In particular, we construct 400 scenarios with 10 items. In each scenario, the reward probability function of each item is setted a uniform distribution $U(a, b)$, with $a < b$ uniformly sampled from $U(0, 1)$. Other parameters are setted as: $p = 0.75$, $c^{\text{ask}} = 0.02$. We vary the reveal costs c_i^{reveal} as a parameter of the experiments, choosing values between 0 and 0.2 with 0.02 increments.

We compare our optimal search strategy with five benchmark strategies that are listed as follows:

Random: The robot randomly selects an item to reveal, check or collect at every time step.

All: The robot asks the human to check or reveals all of the items before collecting any of them. For each unknown item i, the robot decides to reveal it if $c_i^{\text{reveal}} \leq c^{\text{ask}}$, otherwise asks the human to check it. Once all items are known, select the one with highest reward to collect. This algorithm is worthwhile for low ask or reveal costs.

Highest expected value: Instead of using index to evaluate an unknown item, a reasonable strategy may uses the policy of selecting an item with the highest expected value (i.e., $\max_{i \in \bar{S}}\{\mathbb{E}(x_i) - c_i^{\text{reveal}}\}$) to reveal.

Optimal without Human: Without considering human help, the robot searches by itself using our optimal policy that deletes the actions of asking the human for help from the action space. On one hand, this strategy can be used to compare the optimality with "Highest expected value", both of which do not consider human help. On the other hand, by comparing this strategy with "Optimal", we can evaluate the improvements by introducing the mechanism of human-robot interaction to our robot search problems.

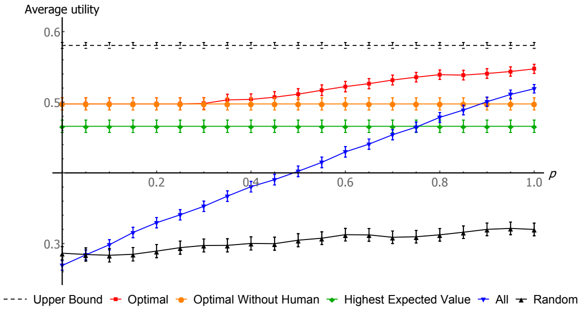

Figure 4: Average utilities for different human availabilities in Experiment A.

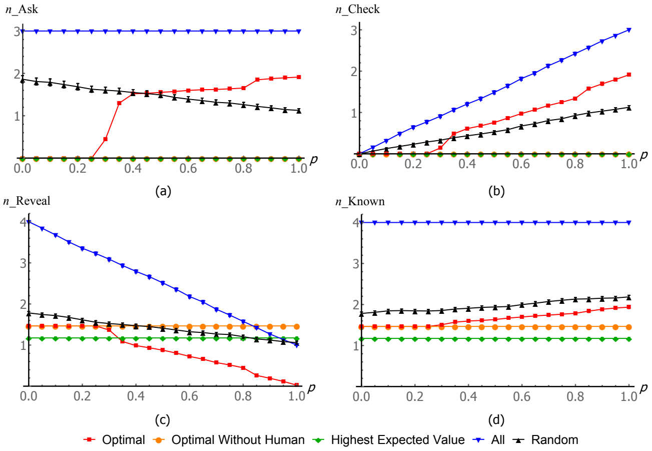

Figure 5: Average (a) asked times, (b) checked times, (c) revealed times and (d) final number of known items for different human availabilities in Experiment A.

Upper bound: An upper bound represents a solution for situations where we assume that the robot is aware of the true information about the human and all the items, including whether the human is available or not at each time step, and the true utility of each item. Note that, this is not a tight upper bound of our solution as the robot is assumed to have more information beyond the basic model.

5.2 Results and Discussion

In Experiment A, for each human availability, we make 1000 simulations. The average utilities obtained by different search strategies are shown in Figure 4 and Figure 5 shows the statistics of n_Ask, n_Check, n_Reveal and n_Known. In these figures, the error bars depict the 95% confidence intervals around the means and non-overlapping error bars invalidate the null hypothesis with $\alpha = 0.05$. As we can see from Figure 4, our optimal search policy ("Optimal") significantly outperforms all others. Specifically, the average utilities obtained by

"Optimal" and "All" gradually increase with human availability p and approach the result of "Upper Bound" when $p = 1$, while "Optimal" performs much better than "All" for low human availabilities. This is because that, for a higher human availability, the robot has a higher probability to receive help from the human when asking the human with a cost c^{ask} (as shown in Figure 5 (a) and (b)). Furthermore, for high availabilities ($p > 0.6$), both "Optimal" and "All" perform better than other algorithms as for some items with high reveal costs they may ask human to check the exact rewards of these items (as we can see in Figure 5 (b) and (c) that for "Optimal" and "All", n_Check increases and n_Reveal decreases with human availability). Moreover, for the algorithms without considering human help, our "Optimal without human" performs better than "Highest expected value" as our search rule considers the expected reward of whole search process, instead of only the expected reward of each action. In addition, we can see from Figure 5 (d) that "Optimal" stops search at a proper number of known items, which is more than "Optimal without human" and "Highest expected value" but less than "All" and "Random", i.e., "Optimal" performs well on "effectively using resources to search key items". Finally, "Random" performs much worse than all other strategies, which is because that any search action is costly and some items may have a bad reward and "Random" not only performs bad on average utilities but also may get a randomly bad result.

In Experiment B, for each reveal cost, we make 1000 simulations. The average utilities obtained by different search strategies are shown in Figure 6 and Figure 7 shows the statistics of n_Ask, n_Check, n_Reveal and n_Known. Although the average utilities obtained by all the strategies decease with the reveal cost, our "Optimal" strategy outperforms all others and is quite close to the "Upper bound". Specifically, if reveal costs are zero, the results of "Optimal", "Optimal without human" and "All" approach that of "Upper Bound", which is because that all the three strategies may reveal all unknown items with zero costs and then select the one with highest reward to collect (as shown in Figure 6). However, "Highest expected value" stops the search if there exists a known item whose collect reward is higher than all the expected values of unknown items (some of them may hold a higher exact reward). Moreover, as shown in Figure 7 (c), n_Reveal of "Highest expected value" and "Upper Bound" are independent with reveal costs $c_i^{\text{reveal}}, \forall i \in I$, and the average utilities generated by "Highest expected value" and "Upper Bound" linearly decease with reveal costs $c_i^{\text{reveal}}, \forall i \in I$ (as shown in Figure 6), which can be easily deduced from their options of actions (i.e., based on $\mathbb{E}(x_i) - c_i^{\text{reveal}}$ and $x_i - c_i^{\text{reveal}}$ respectively). In addition, the generated utilities of "Optimal" and "Optimal without human" decease with reveal costs more rapidly than "Highest expected value". This is because that, as reveal costs increase, the former two strategies will reveal less items and the difference between the sequences of solutions of these strategies deceases.

Given these results, we conclude that our strategy "Optimal" significantly outperforms the benchmarks and the performance of the robot is improved by using the strategies with human-robot interaction.

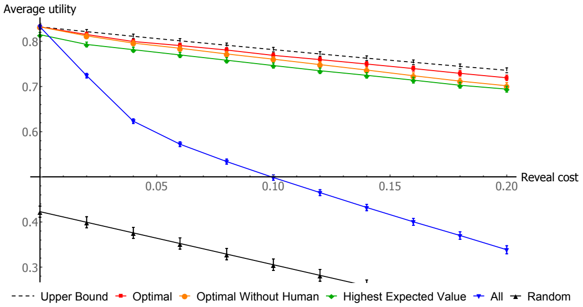

Figure 6: Average utilities for different reveal costs in Experiment B.

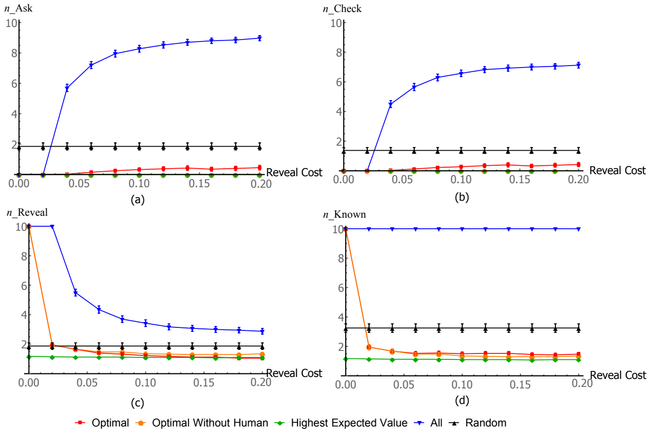

Figure 7: Average (a) asked times, (b) checked times, (c) revealed times and (d) final number of known items for different reveal costs in Experiment B.

6 Conclusion

In this paper, we formulate a new model for robot search tasks with some prior knowledge and human help. In particular, we consider the situation in which a robot searches an item to collect, while the item's utility is unknown until the robot reveals it or asks a human to perform further checks on it. We propose a polynomial optimal strategy and empirically show that our approach is efficient and significantly outperforms other baseline strategies. Moreover, instead of aiming to obtain the best item, our approach can be extended to a more general formulation in which the objective is a function of all found items. Specifically, we can extend our search rule based on the technique in [Olszewski and Weber, 2015] in which Pandora's rule is extended to solve more general problems. Fortunately, after the extension, the complexity and optimality will remain the same. Future work will consider how to deal with mixed types of costs that the robot can incur (e.g., time, distance, etc). For example, in some cases, a human operator is available, but the time to wait for the answer can be too long. Therefore, besides the cost of asking a person, the wait-cost for a possibly-delayed response should also be considered. Moreover, in some environments, items with high reward may be impossible to pick up, and the cost of picking them up should be separated from their rewards.

Acknowledgments

Shaofei Chen is supported in part by Hunan Provincial Innovation Foundation for Postgraduate (No.CX2013B013), National University of Defense Technology for Outstanding Graduate Innovation Fund (No.B130302) and National Natural Science Foundation of China (No. 61403411).

References

[Baarslag and Gerding, 2015] Tim Baarslag and Enrico H. Gerding. Optimal incremental preference elicitation during negotiation. In *Proceedings of the Twenty-fourth International Joint Conference on Artificial Intelligence*, IJCAI'15, pages 3–9. AAAI Press, 2015.

[Bellman, 1962] R Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of ACM*, 9(1):61–63, 1962.

[Bresina and Morris, 2007] John L Bresina and Paul H Morris. Mixed-initiative planning in space mission operations. *AI magazine*, 28(2):75, 2007.

[Fogarty *et al.*, 2005] James Fogarty, Scott E Hudson, Christopher G Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C Lee, and Jie Yang. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):119–146, 2005.

[Gao *et al.*, 2014] Fei Gao, M.L. Cummings, and E.T. Solovey. Modeling teamwork in supervisory control of multiple robots. *Human-Machine Systems, IEEE Transactions on*, 44(4):441–453, Aug 2014.

[Goodrich and Schultz, 2007] Michael A Goodrich and Alan C Schultz. Human-robot interaction: a survey. *Foundations and trends in human-computer interaction*, 1(3):203–275, 2007.

[Hazon *et al.*, 2013] Noam Hazon, Yonatan Aumann, Sarit Kraus, and David Sarne. Physical search problems with probabilistic knowledge. *Artificial Intelligence*, 196:26–52, 2013.

[Kang and Ouyang, 2011] Seungmo Kang and Yanfeng Ouyang. The traveling purchaser problem with stochastic prices: Exact and approximate algorithms. *European Journal of Operational Research*, pages 265–272, 2011.

[Liu and Nejat, 2013] Yugang Liu and Goldie Nejat. Robotic urban search and rescue: A survey from the control perspective. *Journal of Intelligent & Robotic Systems*, 72(2):147–165, 2013.

[Murphy, 2004] Robin Roberson Murphy. Human-robot interaction in rescue robotics. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):138–153, 2004.

[Nourbakhsh *et al.*, 2005] Illah R Nourbakhsh, Katia Sycara, Mary Koes, Mark Yong, Michael Lewis, and Steve Burion. Human-robot teaming for search and rescue. *Pervasive Computing, IEEE*, 4(1):72–79, 2005.

[Olszewski and Weber, 2015] Wojciech Olszewski and Richard Weber. A more general pandora rule? *Journal of Economic Theory*, 160:429 – 437, 2015.

[Rosenfeld *et al.*, 2015] Ariel Rosenfeld, Noa Agmon, Oleg Maksimov, Amos Azaria, and Sarit Kraus. Intelligent agent supporting human-multi-robot team collaboration. In *IJCAI*, pages 1902–1908. AAAI Press, 2015.

[Rosenthal and Veloso, 2011] Stephanie Rosenthal and Manuela Veloso. Modeling humans as observation providers using pomdps. In *RO-MAN, 2011 IEEE*, pages 53–58. IEEE, 2011.

[Rosenthal and Veloso, 2012] Stephanie Rosenthal and Manuela M Veloso. Mobile robot planning to seek help with spatially-situated tasks. In *AAAI*, volume 4, page 1, 2012.

[Rosenthal *et al.*, 2012] Stephanie Rosenthal, Manuela Veloso, and Anind K Dey. Is someone in this office available to help me? *Journal of Intelligent & Robotic Systems*, 66(1-2):205–221, 2012.

[Schmidt-Rohr *et al.*, 2008] Sven R Schmidt-Rohr, Steffen Knoop, M Losch, and Rüdiger Dillmann. Reasoning for a multi-modal service robot considering uncertainty in human-robot interaction. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*, pages 249–254. IEEE, 2008.

[Talamadupula *et al.*, 2010] Kartik Talamadupula, J Benton, Subbarao Kambhampati, Paul Schermerhorn, and Matthias Scheutz. Planning for human-robot teaming in open worlds. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 1(2):14, 2010.

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