High-Quality Policies for the Canadian Traveler's Problem

Patrick Eyerich and **Thomas Keller** and **Malte Helmert**
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Georges-Köhler-Allee 52
79110 Freiburg, Germany
{eyerich,tkeller,helmert}@informatik.uni-freiburg.de

Abstract

We consider the stochastic variant of the Canadian Traveler's Problem, a path planning problem where adverse weather can cause some roads to be untraversable. The agent does not initially know which roads can be used. However, it knows a probability distribution for the weather, and it can observe the status of roads incident to its location. The objective is to find a policy with low expected travel cost.

We introduce and compare several algorithms for the stochastic CTP. Unlike the optimistic approach most commonly considered in the literature, the new approaches we propose take uncertainty into account explicitly. We show that this property enables them to generate policies of much higher quality than the optimistic one, both theoretically and experimentally.

Introduction

The Canadian Traveler's Problem (CTP) was introduced by Papadimitriou and Yannakakis (1991) as a path planning problem with imperfect information about the roadmap. It has drawn considerable attention from researchers in AI search (e. g., Nikolova and Karger 2008; Bnaya, Felner, and Shimony 2009) and is closely related to navigation tasks in uncertain terrain considered in the robotics literature (e. g., Koenig and Likhachev 2002; Ferguson, Stentz, and Thrun 2004). One practical application is outdoor navigation of autonomous robots with the help of a rough map based on a satellite image or a map constructed from previous scans of the environment (Likhachev and Stentz 2006).

Informally, the task is to travel from the initial location to some goal location on a road network given as an undirected weighted graph. This is complicated by the fact that certain roads may be covered by snow and hence be impassable, and the traversability of a road can only be observed from the two incident locations. The weather remains static during the agent's traversal of the graph, so once a road has been observed, its status is known with certainty. Hence, the problem is fully deterministic apart from the initial state uncertainty about which roads are usable.

Many variants of the CTP have been suggested. Papadimitriou and Yannakakis (1991) describe an adversarial setting and a stochastic setting. In the adversarial setting, the objective is to find a policy that minimizes the worst-case ratio between the actual travel cost and the optimal travel cost under perfect information. In the stochastic setting, the status of each road is determined by an independent random choice according to a known probability distribution, and the objective is to minimize expected travel cost (with some subtleties discussed in the next section).

This paper deals with the stochastic CTP, which is the most frequently considered version of the problem and has itself spawned further variants. For example, Nikolova and Karger (2008) describe an optimal algorithm for the stochastic CTP on *disjoint-path* graphs and a recent paper by Bnaya, Felner, and Shimony (2009) studies a variation of the CTP where the status of a road may be sensed remotely, at a cost, and the objective is to minimize the sum of travel cost and sensing cost. A very similar problem to the stochastic CTP where blocking probabilities are associated with graph vertices rather than edges is discussed in the robot path planning community (e. g., Ferguson, Stentz, and Thrun 2004; Likhachev and Stentz 2006).

Although many papers discuss the stochastic CTP, we are not aware of any work that makes a significant attempt at reasoning about the uncertainty that is an integral part of the problem except for studies of special-case graphs (Nikolova and Karger 2008) or instances with very low amounts of uncertainty (Ferguson, Stentz, and Thrun 2004; Likhachev and Stentz 2006). The predominant approach for the general stochastic CTP and related problems is the optimistic policy that always follows the shortest path that *might* be traversable under the agent's current information, no matter how likely it is for this path to be blocked at some point.

Our main contribution is that we show that taking uncertainty into account in the CTP leads to significant improvements over the optimistic policy. On the theoretical side, we show that while the optimistic policy can be arbitrarily worse than the optimal solution, probabilistic policies based on the UCT algorithm (Kocsis and Szepesvári 2006) converge to the global optimum. On the empirical side, we show the advantages of probabilistic approaches over greedy optimism on a range of benchmark instances.

In the following section, we formalize the problem and discuss some basic properties. We then present four algorithms for the CTP, including the common optimistic approach as well as more sophisticated techniques that take uncertainty into account. This is followed by a theoretical comparison of the approaches and an empirical evaluation, after which we conclude.

The Canadian Traveler's Problem

An instance of the CTP is a 6-tuple $\mathcal{I} = \langle V, E, p, c, v_0, v_\star \rangle$, where

- $\langle V, E \rangle$ is a connected undirected graph (*roadmap*) with vertex set V (*locations*) and edge set E (*roads*),
- $p : E \to [0, 1)$ defines the *blocking probabilities* of roads,
- $c : E \to \mathbb{N}_0$ defines the *travel costs* of roads, and
- $v_0, v_\star \in V$ are the *initial* and *goal* locations.

Roads with blocking probability 0 are called *guaranteed*. (We do not allow blocking probabilities of 1 because they cause technical complications in several places, but they can be equivalently modeled by omitting the respective roads.)

A *weather* for a CTP instance with roads E is a subset $W \subseteq E$ representing the roads that are traversable (not blocked by snow) in that weather. Weather W is called *good* if v_0 and $v_\star$ remain connected when only using roads in W. Otherwise, W is called *bad*.

The algorithmic problem considered in this paper is that of computing a good *policy* for a CTP instance. As usual for problems of acting under uncertainty, policies can be represented as mappings from belief states to actions. It is important to note that while the agent interacts with the environment, its knowledge about road traversability grows monotonically because the weather does not change dynamically. Hence, CTP instances are *deterministic POMDPs*, i. e., POMDPs where the only source of uncertainty is incomplete information about the initial state (Littman 1996; Bonet 2009). Deterministic POMDPs are less complex than general POMDPs in that they always have a finite set of reachable belief states.

In the case of the CTP, a belief state can be represented by the agent's current location on the roadmap and a partition of the roads into three disjoint sets: the roads known to be traversable, the roads known to be blocked, and the *unknown* roads (those for which the agent does not have any information). Hence, the number of belief states for a CTP instance with roadmap $\langle V, E \rangle$ is bounded by $|V| \cdot 3^{|E|}$.

To illustrate the random choices of the environment and decision steps of the agent that define the belief space of the problem, the following description shows how a particular *run* (a single interaction of the agent with the environment) on instance $\mathcal{I}$ under policy π proceeds:

- Initially, the environment randomly chooses a weather W by independently marking each road e as blocked with probability $p(e)$ and as traversable otherwise. The problem instance is revealed to the agent, but the randomly chosen weather is not. The agent is initially located at v_0.
- At every decision step, all weather information for the agent's current location v is revealed, i. e., the agent observes which of the roads incident to v are blocked.
- If the current agent location v is the goal location, the run is finished. Otherwise, the agent moves to a new location according to its policy. It may only move to locations that are connected to v by a road e which is traversable under the weather W. This incurs a cost of $c(e)$.
- The cost of the run, denoted by $\mathit{cost}(\mathcal{I}, W, \pi)$, is the sum over all costs incurred by the agent's movements.

We are interested in policies of *expected low cost*, i. e., policies that tend to incur a low cost on a typical run. It is tempting to define the cost of a policy simply as the expected value for $\mathit{cost}(\mathcal{I}, W, \pi)$, where the expectation is with respect to the random choice of weather (and possibly further randomization performed by the policy). However, observe that in case of bad weather it is not possible to complete a run, which is most naturally modeled as infinite cost for that run. This implies that if there is a nonzero chance of bad weather (which is the case iff there exists no path from v_0 to $v_\star$ consisting only of guaranteed roads) the expected cost of all policies would be infinite under this definition.

Fortunately, this problem is easy to avoid by instead defining the cost of the policy as the expected cost for all runs with *good weather*, replacing the prior probabilities for the weather by the posterior probabilities under the condition that the weather is good. It is not hard to prove that changing the probabilities in this fashion does not affect a rational agent's decisions. (Put shortly, the important argument is that it is *always* rational for the agent to assume that the weather is good, because the cost of a run in bad weather is infinite in any case, regardless of the agent's behavior.)

We thus define the cost of a policy π for instance $\mathcal{I}$ as

$$\mathit{cost}(\mathcal{I}, \pi) = \sum_{W \subseteq E} P(W) \cdot \mathit{cost}(\mathcal{I}, W, \pi), \qquad (1)$$

where $P(W)$ is the conditional probability that weather W is chosen given that some good weather is chosen.

Due to the exponential number of possible weathers, it is usually impractical to compute the cost of a given policy π according to Eq. 1. In our empirical experiments we will estimate $\mathit{cost}(\mathcal{I}, \pi)$ by sampling.

Reasonable Policies and Upper Bound. Finding optimal policies for the CTP is difficult. Papadimitriou and Yannakakis (1991) showed that the problem is contained in PSPACE and #P-hard, and so far, optimal solutions could only be generated for instances of trivial size. For example, Zeisberger (2005) describes one optimal approach using a dedicated solver and one optimal approach using a generic POMDP solver, neither of which scales to instances with more than 15 unknown roads.

However, it is not difficult to provide *upper bounds* on the optimal cost, and to find policies that meet these upper bounds. Let N be the number of locations of a given instance. We can divide each run into *phases* where a new phase begins whenever the agent visits some previously unvisited location for the first time. With N locations, there can be at most $N - 1$ such phases in a run. *Within a phase* that starts at location v and ends at location v', the agent only traverses roads on the *known subgraph*, i. e., the graph consisting of only those roads the agent knows to be traversable, by the definition of phases. (New information can only be obtained when reaching a previously unvisited location, ending the phase.)

We can then demand that movements within a phase are performed on *shortest paths* of the known subgraph. We call policies that satisfy this requirement *reasonable*. At the start of the n-th phase, n distinct location have been visited, and hence at most n roads can be traversed by a reasonable policy until a new location is reached, ending the phase. We can thus bound the total number of movements in the run by $\sum_{i=1}^{N-1} i = \frac{1}{2}(N-1)N$, so that the cost of a reasonable policy in good weather is bounded by $\frac{1}{2}(N-1)NC$, where C is the maximal cost of all roads.

Policies for the CTP

We describe four algorithms to compute policies for the CTP. One of them ignores the blocking probabilities in its movement decisions, while the other three take them into account. We will use the same names to refer to the algorithms that compute the policies and the policies themselves. For example, applying the *optimistic algorithm* to a CTP instance results in the *optimistic policy* for the given instance.

All policies π computed by our algorithms can be described in terms of greedy choices with respect to a *cost function* C_π for belief states. (We avoid the term *value function* commonly used in the MDP literature because values are typically maximized, while costs are minimized. Of course, minimizing C_π is equivalent to maximizing $-C_\pi$.) When queried for the next move in belief state b, policy π considers the costs $C_\pi(b')$ for all *successor* belief states b' of b and returns the movements that lead to a successor minimizing the sum of $C_\pi(b')$ and the travel cost from b to b'. To enforce reasonable policies, we define successors of b as those belief states which can be reached through a shortest path in the known subgraph that either ends at the goal or at a location where the agent obtains new information. Once a policy has committed to a movement sequence, no new cost values are computed until the sequence has been completed.

The last approach we consider, UCT, does not actually involve separate computations of $C_\pi(b')$ for each successor. Instead, it only computes $C_\pi(b)$, i. e., performs a computation for the *current* belief state, which produces cost estimates for all successors as a side effect. We abstract from this detail in the following discussion.

It is desirable for cost functions to accurately reflect the actual expected cost to goal. In particular, a policy based on the *optimal cost function* C^* produces optimal behavior. Therefore, we will theoretically compare policies in terms of how accurately their cost functions approximate C^*.

Optimism

We begin with the simplest approach, the *optimistic policy* (OMT). Optimism is a very common approach to the CTP (e. g., Bnaya, Felner, and Shimony 2009) and to robotic motion planning in uncertain environments, where many papers focus on efficient implementations of the optimistic policy (e. g., Stentz 1994; Koenig and Likhachev 2002).

The optimistic policy is based on what is called the *free space assumption* in the robotics literature: as long as it is *possible* that a given road is traversable, we assume that it is traversable.

Formally, the optimistic cost function in belief state b, $C_{\text{OMT}}(b)$, is the distance from the agent location to the goal in the *optimistic roadmap* for b, which is the graph that includes all roads that are known to be traversable in b or unknown in b. Finding shortest paths in the optimistic roadmap is a standard shortest path problem without uncertainty, and hence $C_{\text{OMT}}(b)$ can be efficiently computed.

A sophisticated implementation of the optimistic policy might use algorithms like D* Lite (Koenig and Likhachev 2002) to speed up distance computations, exploiting that over the course of a run, an agent solves a sequence of *similar* path planning problems, allowing reuse of information. Since the focus of this work is on the *quality* of the policy, which is not affected by how C_{OMT} is computed, our implementation simply uses Dijkstra's algorithm.

Hindsight Optimization

The optimistic policy is indeed exceedingly optimistic: its cost estimates are based on the minimum cost to goal in the *best possible weather* given the agent's knowledge. An alternative approach that is less optimistic but still allows us to reduce cost estimation to (a series of) shortest path computations in regular graphs is *hindsight optimization* (HOP).

At each belief state, the hindsight optimization approach performs a sequence of iterations called *rollouts*. The number of rollouts N is a parameter of the algorithm: more rollouts require more time, but tend to produce more stable cost estimates. In each rollout, we first randomly generate a weather according to the blocking probabilities of the CTP instance that is consistent with the agent's knowledge in the given belief state b. In other words, we randomly determine the status of unknown roads using the correct probabilities. If the resulting weather W is bad, the rollout counts as failed. Otherwise, the rollout counts as successful and we compute the distance from the agent's location to the goal in the subgraph of the roadmap that is traversable in W. The hindsight optimization cost estimate $C_{\text{HOP}}^N(b)$ for N rollouts is the average of the computed distances over all successful rollouts.

An alternative and fairly descriptive name for hindsight optimization is *averaging over clairvoyance* (Russell and Norvig 1995). For each weather we consider, we assume that the agent is "clairvoyant", i. e., knows ahead of time which roads are traversable and hence follows the shortest goal path. Since we do not know the actual weather, we *average* over several weathers through stochastic sampling.

Hindsight optimization has recently attracted considerable interest in the stochastic planning community (e. g., Yoon et al. 2008), where it has served as the basis of some highly efficient planning systems. It has also been successfully used for dealing with hidden information in card games, including the one-player game Klondike Solitaire (Bjarnason, Fern, and Tadepalli 2009) and the two-party games bridge (Ginsberg 1999) and Skat (Buro et al. 2009).

Despite these successes, the approach has well-known theoretical weaknesses: it often converges to a suboptimal policy as the number of rollouts approaches infinity. Frank and Basin (2001) give an example of this for the game of bridge, and Russell and Norvig (1995) describe a very simple MDP where HOP fails. In the next section, we give an example of the suboptimality of the HOP policy for the CTP.

Optimistic Rollout

The assumption of clairvoyance is the Achilles heel of the hindsight optimization approach. Our next algorithm, *optimistic rollout* (ORO), addresses this issue by modifying how each rollout is performed. The optimistic rollout approach computes its cost function C^N_{ORO} in the same way as hindsight optimization, by performing a sequence of N rollouts and averaging over cost estimates for successful rollouts.

The difference between the two algorithms is in how the cost estimates of a rollout are computed: in a successful rollout with weather W, rather than using the clairvoyant goal distance, ORO *simulates the optimistic policy* on W and uses the cost of the resulting run as the rollout cost. Hence, in each rollout the agent follows a shortest path in the optimistic graph until it reaches the goal or a road which is blocked in W. In the latter case, it recomputes the optimistic distances based on the new information and follows a new path, iterating in this fashion until it reaches the goal. The total distance traveled then serves as the rollout cost.

Clearly, optimistic rollout is only one representative of a family of *policy rollout* algorithms, as *any* policy could be used in place of the optimistic policy OMT. We choose OMT because it offers a good trade-off between speed and quality.

UCT

The final approach we consider is the *UCT* algorithm (Kocsis and Szepesvári 2006). UCT, which stands for *upper confidence bounds applied to trees*, is a state-of-the-art algorithm for many problems of acting under uncertainty, including playing Klondike solitaire (Bjarnason, Fern, and Tadepalli 2009), which like the CTP is a single-agent problem where the only source of uncertainty is incomplete information about the probabilistically selected initial state.

Similar to the previous algorithms, UCT performs N rollouts, where N is a parameter. As in the ORO algorithm, each UCT rollout computes an actual run from the agent location to the goal for the given weather, without using information that is hidden to the agent, and uses the average cost of successful rollouts as the overall cost estimate $C^N_{\text{UCT}}(b)$. The difference between UCT and ORO is in how the agent's movements during each rollout are determined. While each rollout is independent in ORO, this is not the case in UCT.

Throughout the following description, let b be the belief state on which UCT is queried. A *belief sequence* $\sigma = \langle b, b_1, \dots, b_i \rangle$ is a sequence of belief states that describes a possible partial rollout starting from b. We define

- $R^k(\sigma)$: the *number of rollouts* among the first k rollouts for belief state b that start with sequence σ, and
- $C^k(\sigma)$: the *average travel cost* to complete these $R^k(\sigma)$ rollouts from σ, i. e., the average cost that is incurred on these rollouts from the end of σ to the goal.

Each UCT rollout starts from belief sequence $\langle b \rangle$ and iteratively adds successor belief states until the goal is reached. Let ρ be an unfinished belief sequence for the $(k+1)$-th rollout which ends in belief state b_i. We must describe how UCT picks the next belief state among the successors $b'_1, \dots, b'_m$ of b_i. Let ρ_i be the sequence $\langle \rho; b'_i \rangle$, i. e., ρ extended with b'_i. UCT favors successors that led to *low cost* in previous rollouts (where $C^k(\rho_i)$ is low) and have been *rarely tried* in previous rollouts (where $R^k(\rho_i)$ is low). To balance these criteria, which is the classical trade-off between exploitation and exploration, it picks a candidate ρ_i maximizing the *UCT formula* $B\sqrt{\frac{\log R^k(\rho)}{R^k(\rho_i)}} - \mathit{cost}(\rho, \rho_i) - C^k(\rho_i)$, where $\mathit{cost}(\rho, \rho_i)$ is the travel cost from ρ to ρ_i and $B > 0$ is a *bias* parameter of which more will be said shortly. If $R^k(\rho_i) = 0$, the value of the formula is considered to be ∞, so that the first m rollouts starting with ρ visit each successor once. The UCT formula is designed to select each successor arbitrarily often given sufficiently many visits of ρ, yet successors that have been unpromising in the past are chosen increasingly more rarely over time.

Blind vs. Optimistic UCT. Note that the UCT algorithm as described so far does not take into account any problem-specific information that would bias the rollouts towards the goal. We call the resulting approach *blind UCT* (UCTB). Our experimental results will show that UCTB does not perform very well on the CTP; it would require a prohibitively large number of rollouts to converge to a good policy. However, it is possible to slightly modify the basic UCT algorithm to provide it with some guidance towards the goal. Specifically, we implemented the following two modifications that result in the *optimistic UCT approach* (UCTO):

- When extending a partial rollout ρ which has several unvisited successors, break ties in favor of successors with low C_{OMT} value.
- When evaluating the UCT formula, define $R^k(\sigma)$ and $C^k(\sigma)$ as if there had been M additional rollouts for each successor ρ_i, each with cost $C_{\text{OMT}}(b'_i)$, where M is another algorithm parameter.

These modifications guide early rollouts towards promising parts of the belief space while not affecting the behavior in the limit. Similar extensions to UCT have shown great success in the game of Go (Gelly and Silver 2007).

In our experiments, we used a value of $M = 20$, which was determined empirically. We obtained comparable results for other values in the range 5–80, but significantly worse performance for $M = 0$ or $M = 1$.

The cost function used for the additional "virtual" rollouts (in our case C_{OMT}) is somewhat reminiscent of heuristic functions for deterministic search problems, and the M parameter plays a somewhat similar role to the weight parameter in the weighted A* search algorithm (Pearl 1984) in the sense that it balances to what extent the algorithm relies on heuristic information rather than information obtained by search. However, unlike the weight parameter in weighted A*, our parameter M does not have a clear cut influence on solution quality. As in deterministic search, it is an interesting question how to trade off between the accuracy and computation speed of the cost function used for the virtual rollouts. For example, one might use the cost functions of the HOP or ORO approaches instead of the optimistic cost.

Bias Parameter. To complete our discussion of UCT, we describe how we choose the bias parameter B which balances exploration and exploitation. The analysis in the UCT convergence proof by Kocsis and Szepesvári (2006) suggests that B should be chosen in such a way that it grows linearly with the optimal cost $C^*(b)$. This is also desirable because it means that the policy remains invariant when applying a scaling constant to the travel costs. As the optimal cost is of course unavailable, we estimate it for the $(k+1)$-th rollout by the average cost of the previous k rollouts. (This is undefined for $k = 0$, but B does not affect the choices of the first rollout anyway.) For the UCTO variant, we additionally divide the bias by 10 to further encourage exploitation.

Theoretical Evaluation

We have introduced four different approaches for the CTP. (We treat UCTB and UCTO as a single approach in this section, as all results apply equally to both). What are their strengths and weaknesses? How accurately do their cost functions approximate the true cost C^*? Here we present some formal answers to these questions. For space reasons, we only provide proof sketches. We begin with a basic result:

Theorem 1 *As the number of rollouts N approaches ∞, the HOP, ORO and UCT cost functions converge in probability.*

Proof sketch: Individual HOP or ORO rollout costs are independent and identically-distributed bounded random variables, so the strong law of large numbers applies. (Boundedness follows from our discussion of reasonable policies.)

The UCT result is covered by the proof of Theorem 2. ■

Convergence of cost functions in probability implies convergence of the induced policies with probability 1 for those belief states which have a unique successor that minimizes the cost function in the limit. If the minimizing successor is not unique, the policy in the limit will randomly choose one of the minimizers.

In the rest of this section, we denote the cost functions to which the N-rollout cost functions converge with C^∞_{HOP}, C^∞_{ORO} and C^∞_{UCT} and consider the policies in the limit rather than policies based on a finite number of rollouts. Theorem 1 ensures that these notions are well-defined.

To motivate the ideas underlying our main result, the example instance in Fig. 1 illustrates the different pitfalls that OMT, HOP and ORO fall prey to. We assume that ϵ is very small and limit attention to runs where all roads with blocking probability ϵ are traversable and the road with blocking probability $1-\epsilon$ is blocked.

The optimistic policy is led astray by the cheap but very unlikely path that reaches $v_\star$ via v_6. It would follow the path v_0–v_5–v_6–v_5–$v_\star$, for a total cost of 170.

Hindsight optimization chooses wrongly because there is a high probability of a cheap goal path via v_1 and any of the locations v_2/v_3/v_4, but it is not clear *which* of these three locations to enter. It would assign a cost of 100 to the v_0–$v_\star$ choice, a cost close to 90 to the v_0–v_5 choice (due to path v_0–v_5–$v_\star$) and a cost close to 75 ($= 10 + (\frac{7}{8} \cdot 60 + \frac{1}{8} \cdot 100)$) to the v_0–v_1 choice, hence moving to v_1 first. At v_1 it would realize the suboptimality of its choice and ultimately reach the goal via path v_0–v_1–v_0–v_5–$v_\star$ at cost 110.

Optimistic rollout is fooled by the fact that OMT acts suboptimally in v_5, giving rise to an exaggerated cost estimate for v_5. It would follow the path v_0–$v_\star$ at cost 100.

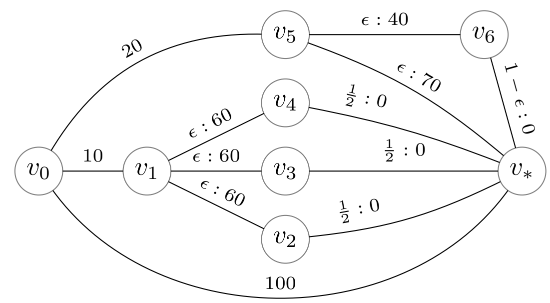

Figure 1: Example with pitfalls for OMT, HOP and ORO. Edge labels $p : w$ denote blocking probability p (omitted for guaranteed roads, i. e., when $p = 0$) and travel cost w.

Finally, UCT converges to the optimal policy, following the path v_0–v_5–$v_\star$ at cost 90. This is a consequence of our main result, which we now present.

Theorem 2 *For all CTP instances $\mathcal{I}$ and belief states b:*

$$C_{OMT}(b) \leq C^\infty_{HOP}(b) \leq C^\infty_{UCT}(b) = C^*(b) \leq C^\infty_{ORO}(b),$$

where UCT *refers to both policy variants. Moreover, there are instances where all inequalities are strict and the ratio between any two different cost functions is arbitrarily large.*

Proof sketch: For UCTB, convergence to the optimal cost function (and hence also to the optimal policy) follows from a slight generalization of Theorem 6 of Kocsis and Szepesvári (2006). The modifications to UCTB that give rise to UCTO do not affect behavior in the limit as the algorithm will eventually explore all branches an unbounded number of times, independently of any initialization to the R^k and C^k values, and the contribution of the initial R^k and C^k values to the UCT formula converges to zero over time.

$C^*(b) \leq C^\infty_{\text{ORO}}(b)$ holds because each ORO rollout corresponds to an *actual* run of the CTP instance under some policy (namely, the optimistic one), which cannot have a lower expected cost than the optimal cost C^*.

To prove $C_{\text{OMT}}(b) \leq C^\infty_{\text{HOP}}(b) \leq C^*(b)$, let $\mathcal{I}$ be the given instance with road set R and let Π be the set of all policies for $\mathcal{I}$. We can show that for the initial belief state b_0:

$$C_{\text{OMT}}(b_0) = \min_{\pi\in\Pi} \min_{W\subseteq R} \textit{cost}(\mathcal{I}, W, \pi)$$
$$C^\infty_{\text{HOP}}(b_0) = E[\min_{\pi\in\Pi} \textit{cost}(\mathcal{I}, W, \pi)]$$
$$C^*(b_0) = \min_{\pi\in\Pi} E[\textit{cost}(\mathcal{I}, W, \pi)]$$

where expected values are w.r.t. the random choice of (good) weather W. The result for b_0 follows from this by simple arithmetic and readily generalizes to all belief states.

To show arbitrary separation between C_{OMT}, C^∞_{HOP}, C^* and C^∞_{ORO}, we use augmented versions of the "pitfalls" for the respective algorithms exemplified in Fig. 1. ■

Experimental Evaluation

To evaluate the algorithms empirically, we performed experiments on Delaunay graphs, following the example of

	OMT	**HOP**	**ORO**	**UCTB**	**UCTO**
20-1	205.9 ± 7	171.6 ± 6	176.3 ± 5	210.7 ± 7	$\mathbf{169.0} \pm 6$
20-2	187.0 ± 5	155.8 ± 3	150.3 ± 3	176.4 ± 4	$\mathbf{148.9} \pm 3$
20-3	139.5 ± 6	138.7 ± 6	134.2 ± 6	150.7 ± 7	$\mathbf{132.5} \pm 6$
20-4	266.2 ± 8	286.8 ± 8	264.2 ± 7	264.8 ± 9	$\mathbf{235.2} \pm 7$
20-5	163.1 ± 7	113.3 ± 5	113.0 ± 6	123.2 ± 7	$\mathbf{111.3} \pm 5$
20-6	180.2 ± 6	142.0 ± 4	134.4 ± 4	165.4 ± 6	$\mathbf{133.1} \pm 3$
20-7	172.2 ± 5	150.2 ± 4	168.8 ± 4	191.6 ± 6	$\mathbf{148.2} \pm 4$
20-8	150.1 ± 6	$\mathbf{133.6} \pm 5$	137.7 ± 5	160.1 ± 7	134.5 ± 5
20-9	222.0 ± 5	177.1 ± 4	176.4 ± 4	235.2 ± 6	$\mathbf{173.9} \pm 4$
20-10	178.2 ± 6	188.1 ± 6	$\mathbf{166.3} \pm 5$	180.8 ± 7	167.0 ± 5
$\varnothing\, C$	186.5 ± 2	165.7 ± 2	162.2 ± 2	185.9 ± 2	$\mathbf{155.4} \pm 2$
$\varnothing\, T_{\mathrm{run}}$	0.00 s	0.73 s	2.12 s	2.37 s	1.57 s
$\varnothing\, T_{\mathrm{dec}}$	0.00 s	0.04 s	0.14 s	0.13 s	0.10 s
50-1	255.5 ± 10	250.6 ± 9	214.3 ± 7	229.4 ± 12	$\mathbf{186.1} \pm 7$
50-2	467.1 ± 11	375.4 ± 7	406.1 ± 8	918.0 ± 16	$\mathbf{365.5} \pm 7$
50-3	281.5 ± 9	294.5 ± 7	268.5 ± 7	382.1 ± 15	$\mathbf{255.6} \pm 7$
50-4	289.8 ± 9	263.9 ± 7	241.6 ± 7	296.6 ± 12	$\mathbf{230.5} \pm 7$
50-5	285.5 ± 10	239.5 ± 8	229.5 ± 7	290.8 ± 11	$\mathbf{225.4} \pm 7$
50-6	251.3 ± 10	253.2 ± 9	238.3 ± 9	405.2 ± 21	$\mathbf{236.3} \pm 8$
50-7	242.2 ± 9	221.9 ± 7	209.3 ± 7	250.5 ± 11	$\mathbf{206.3} \pm 7$
50-8	355.1 ± 11	302.2 ± 9	300.4 ± 8	462.6 ± 15	$\mathbf{277.6} \pm 8$
50-9	327.4 ± 13	281.8 ± 11	238.1 ± 9	295.2 ± 18	$\mathbf{222.5} \pm 9$
50-10	281.6 ± 8	271.2 ± 7	249.0 ± 6	390.8 ± 15	$\mathbf{240.8} \pm 6$
$\varnothing\, C$	303.7 ± 3	275.4 ± 3	259.5 ± 3	392.1 ± 6	$\mathbf{244.7} \pm 2$
$\varnothing\, T_{\mathrm{run}}$	0.00 s	6.36 s	28.02 s	40.48 s	13.99 s
$\varnothing\, T_{\mathrm{dec}}$	0.00 s	0.15 s	0.85 s	0.96 s	0.44 s

Table 1: Average travel costs with 95% confidence intervals for 1000 runs on roadmaps with 20 (top) and 50 (bottom) locations. Best results on each graph are highlighted in bold. In each block, the three last rows show average travel cost ($\varnothing\, C$), average runtime per run ($\varnothing\, T_{\mathrm{run}}$), and average runtime per decision ($\varnothing\, T_{\mathrm{dec}}$) for the ten graphs in that block.

Bnaya, Felner, and Shimony (2009). For each algorithm and benchmark graph, we performed 1000 runs to estimate the true policy cost as defined in Eq. 1 with sufficient accuracy.

Main experiment. In our main experiment, we generated random Delaunay graphs with 20–50 locations. The smaller graphs with 20 locations have 49–52 roads; the larger ones with 50 locations have 133–139 roads. All roads could potentially be blocked, with blocking probabilities chosen uniformly in the range $[0, 1)$. Travel costs were chosen uniformly from $\{1, \dots, 50\}$. Initial and goal locations were set to be at "opposite ends" of the graph.

We evaluated all algorithms on these 20 benchmarks, using 10000 rollouts for the probabilistic algorithms. Table 1 shows the outcome of the experiment. The optimistic UCT algorithm dominates, always providing the cheapest policies except for two cases where the difference between UCTO and the best performance is not statistically significant. In addition to UCTO, the HOP and ORO algorithms also significantly outperform the optimistic approach, clearly demonstrating the benefit of taking uncertainty into account for the CTP. These overall results nicely complement our theoretical results. We conjecture that for some of the graphs where the UCTO policy significantly outperforms the other policies, it reaches a solution quality that is unobtainable for HOP and ORO in the limit.

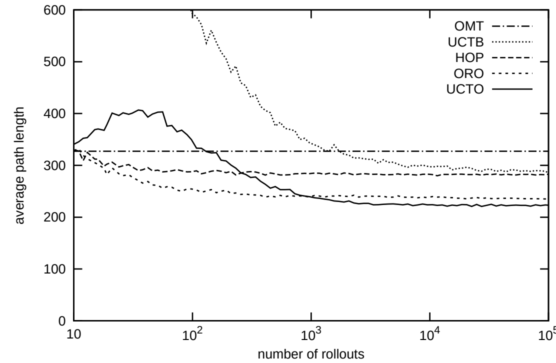

Figure 2: Average travel cost as a function of rollout number for benchmark instance 50-9.

On average, the UCTO algorithm reduces the costs compared to the optimistic policy by 16.7% on the smaller graphs and by 19.4% on the larger graphs, a huge improvement. The blind UCT algorithm does not fare well, converging too slowly – a not unexpected result, as the initial rollouts of UCTB have to reach the goal through random walks. The poor performance of UCTB underlines that these benchmarks are far from trivial.

While we want to emphasize policy quality here, not runtime, the table also provides some runtime results. They show that in our implementation, UCTO is about twice as slow as HOP and twice as fast as ORO on the larger graphs. In absolute terms, UCTO on average requires 0.10 seconds per decision on the smaller graphs and 0.44 seconds on the larger graphs. The perhaps surprising speed advantage of UCTO over ORO can be explained by the fact that (as indicated at the beginning of the section on policies for the CTP) UCTO only performs 10000 total rollouts per query, while HOP and ORO perform 10000 rollouts for each successor belief state. UCTB is the slowest algorithm by far because its rollouts tend to require more steps to reach the goal and do not revisit previously explored belief sequences to the same extent as UCTO.

Rollouts and Scalability. To analyze the speed of convergence and scalability of the probabilistic algorithms, we performed additional experiments on individual benchmarks where we varied the rollout number in the range 10–100000. Figure 2 shows the outcome for benchmark graph 50-9. We see that apart from UCTB, the probabilistic algorithms already obtain a better quality than the optimistic policy with only about 100 rollouts, which require very little computation. ORO and HOP begin to level off after about 1000 rollouts, where UCTO still continues to improve.

To further evaluate the scalability of the algorithms, we have also performed limited experiments on benchmarks with up to 500 locations, which show that the advantage of UCTO over the optimistic policy tends to increase on larger instances, while runtime per decision grows slightly more than linearly in the problem size.

Remote Sensing. In our last experiment, we evaluated the performance of our stochastic algorithms on the benchmark

Algorithm	$p = 0.1$	$p = 0.3$	$p = 0.5$	$p = 0.6$
Always	$+30.27\%$	$+32.88\%$	$+39.15\%$	$+30.05\%$
Exp	$+0.43\%$	-0.39%	-6.84%	$+3.30\%$
VOI	$+0.64\%$	$\mathbf{-4.70\%}$	-2.16%	-6.06%
ORO	$\mathbf{-0.12\%}$	-2.12%	-5.76%	-5.11%
UCTO	$+0.13\%$	-2.64%	$\mathbf{-6.95\%}$	$\mathbf{-7.13\%}$

Table 2: Results for CTP with remote sensing (sensing cost 5), reported as average cost differences compared to OMT (called "Never" by Bnaya et al.) for the four different graph classes in the Bnaya et al. benchmark set. In column "$p = x$", all roads are blocked with a probability of x. Negative numbers indicate improvements. Best performances in bold.

instances of Bnaya et al. These are benchmarks for a different problem, a CTP variant where agents may sense the status of roads from a distance, at a cost of 5. The policies "Always", "Exp", and "VOI" suggested by Bnaya et al. make use of these capabilities. To these policies we compare the solution qualities obtained by our policies when treating the same benchmarks as *regular* CTP instances. Thus, we compare policies that attempt to make use of sensing capabilities intelligently to ones that *never perform remote sensing*. The experimental results (Table 2) show that these never-sensing policies are competitive with the best policies of Bnaya et al.

Conclusion

We investigated the problem of finding high-quality policies for the stochastic version of the Canadian Traveler's problem. In addition to the optimistic approach commonly considered in the CTP literature, we discussed three algorithms which take into account blocking probabilities in their decision-making process.

We studied the convergence properties of these algorithms and proved a clear ordering between the underlying cost functions. Experimentally, we showed that the new algorithms, in particular our adaptation of UCT, offer significant improvements over the optimistic approach. These improvements are large enough to offer competitive performance to state-of-the art approaches for the CTP with remote sensing even when performing no sensing at all.

In the future, we want to examine if better initialization procedures can further improve the convergence behavior of our UCT-based algorithm. Furthermore, we intend to adapt our algorithms to related problems such as the CTP with remote sensing and to more general problems such as probabilistic planning.

Acknowledgments

We thank Zahy Bnaya for his help with the experiments on the CTP with remote sensing. Some of the experiments reported in this paper were conducted with computing resources provided by the Black Forest Grid initiative.

This work was supported by the European Union as part of the Integrated Project "CogX – Cognitive Systems that Self-Understand and Self-Extend" (FP7-ICT-2xo15181-CogX). For more information, see `http://cogx.eu/`. It was also supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14 AVACS). For more information, see `http://www.avacs.org/`.

References

Bjarnason, R.; Fern, A.; and Tadepalli, P. 2009. Lower bounding Klondike solitaire with Monte-Carlo planning. In *Proc. ICAPS 2009*, 26–33.

Bnaya, Z.; Felner, A.; and Shimony, S. E. 2009. Canadian traveler problem with remote sensing. In *Proc. IJCAI 2009*, 437–442.

Bonet, B. 2009. Deterministic POMDPs revisited. In *Proc. UAI 2009*, 59–66.

Buro, M.; Long, J. R.; Furtak, T.; and Sturtevant, N. 2009. Improving state evaluation, inference, and search in trick-based card games. In *Proc. IJCAI 2009*, 1407–1413.

Ferguson, D.; Stentz, A.; and Thrun, S. 2004. PAO* for planning with hidden state. In *Proc. ICRA 2004*, 2840–2847.

Frank, I., and Basin, D. A. 2001. A theoretical and empirical investigation of search in imperfect information games. *Theoretical Computer Science* 252(1–2):217–256.

Gelly, S., and Silver, D. 2007. Combining online and offline knowledge in UCT. In *Proc. ICML 2007*, 273–280.

Ginsberg, M. L. 1999. GIB: Steps toward an expert-level bridge-playing program. In *Proc. IJCAI 1999*, 584–593.

Kocsis, L., and Szepesvári, C. 2006. Bandit based Monte-Carlo planning. In *Proc. ECML 2006*, 282–293.

Koenig, S., and Likhachev, M. 2002. D* Lite. In *Proc. AAAI 2002*, 476–483.

Likhachev, M., and Stentz, A. 2006. PPCP: Efficient probabilistic planning with clear preferences in partially-known environments. In *Proc. AAAI 2006*, 860–867.

Littman, M. L. 1996. *Algorithms for Sequential Decision Making*. Ph.D. Dissertation, Brown University, Providence, Rhode Island.

Nikolova, E., and Karger, D. R. 2008. Route planning under uncertainty: The Canadian traveller problem. In *Proc. AAAI 2008*, 969–974.

Papadimitriou, C. H., and Yannakakis, M. 1991. Shortest paths without a map. *Theoretical Computer Science* 84(1):127–150.

Pearl, J. 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence — A Modern Approach*. Prentice Hall.

Stentz, A. 1994. Optimal and efficient path planning for partially-known environments. In *Proc. ICRA 1994*, 3310–3317.

Yoon, S.; Fern, A.; Givan, R.; and Kambhampati, S. 2008. Probabilistic planning via determinization in hindsight. In *Proc. AAAI 2008*, 1010–1016.

Zeisberger, U. 2005. Pfadplanung unter Unsicherheit. Diplomarbeit, Albert-Ludwigs-Universität Freiburg.