Repeated-Task Canadian Traveler Problem

Zahy Bnaya and Ariel Felner
Information Systems Engineering
Deutsche Telekom Labs
Ben-Gurion University
Be'er-Sheva, Israel
{zahy,felner}@bgu.ac.il

Solomon Eyal Shimony, Dror Fried, Olga Maksin
Computer Science,
Ben-Gurion University
Be'er-Sheva, Isrel
{shimony,dfried,omaksin}@cs.bgu.ac.il

Abstract

In the Canadian Traveler Problem (CTP) a traveling agent is given a graph, where some of the edges may be blocked, with a known probability. A solution for CTP is a policy, that has the smallest expected traversal cost. CTP is intracable. Previous work has focused on the case of a single agent. We generalize CTP to a repeated task version where a number of agents need to travel to the same goal, minimizing their combined travel cost. We provide optimal algorithms for the special case of disjoint path graphs. Based on a previous UCT-based approach for the single agent case, a framework is developed for the multi-agent case and four variants are given - two of which are based on the results for disjoint-path graphs. Empirical results show the benefits of the suggested framework and the resulting heuristics. For small graphs where we could compare to optimal policies, our approach achieves near-optimal results at only a fraction of the computation cost.

Introduction

The Canadian Traveler Problem (CTP) (Nikolova and Karger 2008; Papadimitriou and Yannakakis 1991; Bar-Noy and Schieber 1991) is a navigation problem where traveling agent receives a graph as input and needs to travel from its initial location to a given goal location. The complication in CTP is that some edges of the graph may be blocked with a known probability. The basic action in the CTP is a *move* action along an edge of the graph. Moves incur costs. After arriving at a node, the agent can sense its incident edges at no cost. The task is to minimize the travel cost of reaching the goal. Since some of the graph's edges may be blocked, a simple search for a path does not work; a solution is a contingent plan, or policy, that has the smallest expected traversal cost.

In the basic version of the CTP, a single agent is assumed. However, in many realistic settings the problem needs to be solved for a group of agents, requiring minimization of the combined travel cost of all agents. For example, think of an owner of a fleet of trucks, who has to move the trucks from a single source to a single destination. In this paper, we generalize CTP to its multi-agent repeated-task setting, that assumes that the agents traverse the graph sequentially. We also briefly examine the interleaved version, where agents are allowed to traverse the graph concurrently. Single-agent CTP is intractable (Papadimitriou and Yannakakis 1991). Having multiple agents further increases the size of the state-space, which makes finding an optimal solution even harder and is also intractable. Therefore, we cannot find optimal solutions efficiently in general. However, we provide efficient optimal solutions for special case graphs which are then used to generate heuristics for the general case. These heuristics are tested empirically against baselines developed for single and multi-agent. Experimental results show that our heuristics appear to improve the expected travel cost, and are not too far from the optimal for small graphs.

Variants of the Canadian Traveler Problem

In the *single agent* version of CTP, a traveling agent is given a connected weighted graph $G = (V, E)$, a source vertex ($s \in V$), and a target vertex ($t \in V$). The input graph G may undergo changes, that are not known to the agent, *before* the agent begins to act, but remains fixed subsequently. In particular, some of the edges in E may become *blocked* and thus untraversable. Each edge $e \in E$ has a weight, $w(e)$, and is *blocked* with a known *blocking probability* $p(e)$, or *traversable* with probability $q(e) = 1 - p(e)$. The agent can perform *move* actions along an unblocked edge which incurs a *travel cost* equal to $w(e)$. The status of an edge (*blocked* or *traversable*) is revealed to the agent, only after the agent reaches a vertex incident to that edge.

The task of the agent is to travel from s to t with minimal total travel cost. As the exact travel cost is uncertain until the end, a solution to CTP is a traveling strategy (policy) which yields a small (ideally optimal) *expected* travel cost.

To illustrate CTP, consider figure 1 with unknown edges $e_{0,1}$ and $e_{1,1}$. If $e_{0,1}$ is traversable, the cheapest path is (s, v_0, t) with cost 1.5, but if $e_{0,1}$ is blocked, and $e_{1,1}$ is traversable, the cheapest path is (s, v_1, t) with cost 2.5. Taking into account the blocking events ($e_{0,1}$ is likely to be blocked, and its status can only be observed from v_0), the optimal policy is to try v_1 first, then if $e_{1,1}$ is blocked go to v_0; finally if $e_{0,1}$ is also blocked, reach t though path I_2.

Multi-agent CTP

In this paper, we generalize CTP to its multi-agent case. where several (n) agents operate in the given graph. We

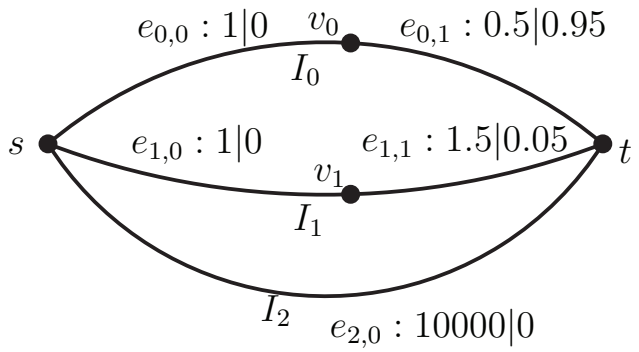

Figure 1: CTP example. Edge label $w|p$ denotes edges cost w, blocking probability p.

assume that the agents are fully cooperative and aim to minimize their total travel cost. In addition, we assume a communication paradigm of *full knowledge sharing*. That is, any new information discovered by an agent (e.g., whether an edge is blocked or traversable) is immediately made known to all other agents. This assumption is equivalent to having a centralized control of all agents. In our trucks example, the trucks have full communication with the owner, who, on his part, is interested in minimizing the expected total cost of the entire fleet.[1]

Repeated task CTP In the *Repeated task* multi-agent CTP (called CTP-REP(n) for short) there is only one *active agent* at a time. All other agents are inactive until the currently active agent reaches t. An agent that reaches t becomes inactive again (is "out of the game"), and can make no additional actions or observations. The goal is a natural extension of single-agent CTP: all n agents begin at s, and must reach t. We need to find a policy for the agents that minimizes the expected total travel cost of reaching this goal. Equivalently, this can be seen as repeating the single-agent CTP task n times, but in each case based on observations gathered in all previous traversals. Similar combinations of multi-agent and repeated task navigation were presented by (Meshulam, Felner, and Kraus 2005) but for for the case of completely unknown graphs (as opposed to the CTP framework here).

Interleaved repeated-task CTP In the more general *interleaved action multi-agent CTP* (called CTP-MULTI(n) for short), the agents make their moves in a round-robin fashion. No-op actions, where an agent does not move in a specific turn, with a cost of zero, are permitted. Note that one could consider an alternate definition of the problem, where agents move in parallel. However, in the latter setting, we can prove the following property: *there exists an optimal policy where all agents but one perform a "no-op" at every step*, since no cost is incurred by doing nothing (proof by policy simulation). Thus, there is no need to examine the parallel action case.

[1]The treatment of other types of agents (e.g., selfish or adversarial agents) and other communications paradigms (e.g., message exchanging) is beyond the scope of this paper.

Complexity of Single and Multi-Agent CTP

Define a *weather* as a given assignment of {traversable,blocked} status to an edge. There are $2^{|E|}$ different possible weathers. Therefore, the size of the *state space* of CTP is $V \times 2^{|E|}$ because a state also includes the specific location of the agent. Partially Observable Markov Decision Processes (POMDP) offer a model for cost optimization under uncertainty, and CTP can be modeled as an indefinite horizon POMDP (Hansen 2007). POMDPs use the notion of a *belief state* to indicate knowledge about the state. In CTP, the belief state is the location of the agent coupled with the belief status about the edges (the hidden variables). One could specify edge variables as ternary variables with domain: {*traversable, blocked, unknown*}, thus the belief space size is $O(|V|3^{|E|})$ (where is the agent, what do we know about the status of each edge). An optimal policy for this POMDP, is a mapping from the belief space into the set of actions, that minimizes the expected travel cost.

Solving POMDPs is PSPACE-hard (in the number of states) in the general case. However, because the only source of uncertainty in CTP is in the initial belief state and not in the outcome of the actions, CTP is actually a special case of the simpler *deterministic* POMDP (Littman 1996; Bonet 2009), which can be solved in time polynomial in the size of the state space. Unfortunately these results do not help solving CTP, because the size of the CTP state space ($O(|V|2^{|E|})$) is exponential in the size of the input graph. In fact CTP is known to be #P-hard (Papadimitriou and Yannakakis 1991), and some variants are known to be PSPACE-complete. Multi-agent CTP is potentially harder. The size of the state space is now $O(|V|^n 2^{|E|})$ for CTP-MULTI(n) as we need to specify the locations of n agents for each state, and $O(n|V|2^{|E|})$ for CTP-REP(n) because we have n phases, one per agent.

Theoretical Results on Disjoint Paths

We restrict the theoretical analysis in this paper to disjoint-path graphs (CTP-DISJ) such as that shown in figure 1 because every departure from this topology makes the analysis prohibitively complex, even for single-agent CTP. For example, adding even one edge to an instance of CTP-DISJ, either between nodes in different paths or from such a node directly to t, violates the conditions under which *committing policies* (see below) are optimal. The situation is made even more complex in multi-agent CTP. Nevertheless, our theoretical results on this restricted version provide insight for the general case, as shown below.

A CTP-DISJ graph has $k \geq 2$ paths, denoted by $I_0, \cdots, I_{k-1}$. We assume w.l.o.g. that at least one path is known to be traversable. Otherwise, we can add a traversable path consisting of one edge with a finite, but very large cost, between s and t. If the agent reveals that all other paths are blocked, it can backtrack and follow this additional edge, which can be thought of as a rescue choice, such as *call an helicopter*, that the agent must take if no regular path to the goal exists.

The length r_i of each path I_i, is the number of the edges of I_i. The edges of path I_i starting from s are denoted by $e_{i,j}$

for $0 \leq j < r_i$, (see Fig. 1). For a path I_i, and an edge $e_{i,j}$, let $W_{i,j} = \sum_{l<j} w(e_{i,l})$ be the cost of the path I_i up to edge $e_{i,j}$ without including $e_{i,j}$, and let $W_i = W_{i,r_i}$ be the cost of the entire path I_i. We also define Q_i to be the probability of path I_i being unblocked; thus $Q_i = \prod_{l<r_i} q(e_{i,l})$.

Single-Agent CTP on disjoint paths

The analysis for single-agent CTP-DISJ is from (Bnaya, Felner, and Shimony 2009; Nikolova and Karger 2008) and repeated here as the basis for the multi-agent case. *Reasonable* policies are policies that do not involve the agent walking around without making headway and without obtaining new information. An optimal policy must also be reasonable. Two "macro actions" are defined in terms of which all reasonable policies on disjoint-paths graph can be specified. Both macro actions are defined for an agent situated at s.

Definition 1 *For path I_i, macro action $TRY(i)$ is to move forward along path I_i until reaching t; if a blocked edge is encountered, the agent returns along the path and stops at s. An agent performing a TRY action on a path is said to be* trying *the path.*

Definition 2 *For path I_i, and an edge $e_{i,j}$, macro action $INV(i,j)$ (meaning* investigate*) is to move forward along path I_i until reaching (without crossing) $e_{i,j}$, or a blocked edge, whichever occurs first. In either case the agent returns to s.*

For path I_i, denote by $BC(I_i)$ the random variable representing the backtracking cost of path I_i: the cost of traveling path I_i, finding I_i blocked, and returning to s. Since the path can be blocked anywhere, and the cost to reach edge $e_{i,j}$ is $W_{i,j}$, the expected backtracking cost is:

$$E[BC(I_i)] = 2 \sum_{j<r_i} W_{i,j} p(e_{i,j}) \prod_{l<j} q(e_{i,l}) \qquad (1)$$

Denote the expected cost of $TRY(I_i)$ by $E[TRY(I_i)]$:

$$E[TRY(I_i)] = Q_i W_i + E[BC(I_i)] \qquad (2)$$

A policy which consists only of TRY actions, but never uses INV actions, (that is, only backtracks if a blocked edge is revealed), is called *committing*. Since a TRY macro action either reaches t or finds a path blocked, it never makes sense to try the same path more than once, and thus all such committing policies can be represented by an order of paths to be tried. Let M be an instance of CTP-DISJ, and χ^*_M be a committing policy for M in which the agent tries the paths in a non decreasing order of $\frac{E[TRY(I_i)]}{Q_i}$. Assuming without loss of generality that $\frac{E[TRY(I_i)]}{Q_i}$ are all different, and thus χ^*_M is unique. Then the following theorem (Bnaya, Felner, and Shimony 2009) holds:

Theorem 1 *χ^*_M is an optimal policy for M.* □

Repeated CTP in disjoint-path graphs

We now adapt these results to Repeated CTP with n agents on disjoint paths graph (CTP-DISJ-REP(n)). Let M be an instance of CTP-DISJ-REP(n) with k paths. Note that any *reasonable* policy in M can be represented using only TRY and INV macro actions as follows. Let $TRY(l,i)$ be the action in which agent A_l tries path I_i, and let $INV(l,i,j)$ be the action in which agent A_l performs $INV(i,j)$. A policy for an agent A_i is *committing* if A_i executes only TRY actions. Likewise, a policy (for a set of agents) is *committing* if it consists only of TRY actions for all agents. It is non-trivial to show that in repeated CTP, TRY actions suffice for optimality – this requires definition of the restricted followers-commiting policies, discussed next.

Let π be a policy for M, where whenever A_0 reaches t through path I_i, for some $i < k$, the agents $A_1, \cdots A_{n-1}$ traverse I_i as well. A policy π with this property is called a *followers-committing* policy, and the agents $A_1, \cdots A_{n-1}$ are said to *follow* A_0 in π.

Denote by $TRY_n(I_i)$ the action $TRY(0,i)$ executed in a followers-committing policy in CTP-DISJ-REP(n). The results of such actions are either that A_0 traverses I_i to t, and the other agents follow A_0 on I_i, or that A_0 finds I_i blocked and backtracks to s (other agents staying idle). Let $E[TRY_n(I_i)]$ be the expected cost of $TRY_n(I_i)$. We get:

$$E[TRY_n(I_i)] = nQ_iW_i + E[BC(I_i)] \qquad (3)$$

Let $<_*$ be the following order on the k paths: $l <_* j$ if and only if $\frac{E[TRY_n(I_i)]}{Q_i} < \frac{E[TRY_n(I_j)]}{Q_j}$, assuming again w.l.o.g. that all of these fractions are different. Let σ_* be the permutation on $\{0, \cdots, k-1\}$ which is induced by $<_*$.

Definition 3 *Let π^*_M be the followers-committing policy where A_0 executes the committing policy of trying the paths by increasing order of $<_*$, and $A_1, \cdots, A_{n-1}$ follow A_0.*

We show that π^*_M is optimal, but first need to show that π^*_M is optimal among all followers-committing policies. Denote the expected cost of policy π by $c(\pi)$.

Lemma 1 *Let π be a followers-committing policy for M. Then $c(\pi^*_M) \leq c(\pi)$.*

Proof outline: Note that any followers-committing policy π for an instance M of CTP-DISJ-REP(n) can be re-cast as an equivalent CTP-DISJ problem instance M', where an extra cost of $(n-1)W_i$ is incurred once that agent reaches t through path I_i, as follows. M' is an extension of M, where we add at the end of each path I_i, an additional traversable edge e_{i,r_i} incident on t, with a cost of $(n-1)W_i$. Since, in a followers-committing policy for M, all agents follow the first agent, thus each agent incurs a cost of W_i, there is a trivial bijection F between followers-committing policies in M, and committing policies in M', that preserves expected costs, such that $F(\pi^*_M) = \chi^*_{M'}$. Now suppose that π is a followers-committing policy for M. Then by theorem 1, $c(F(\pi^*_M)) \leq c(F(\pi))$, hence $c(\pi^*_M) \leq c(\pi)$.

Theorem 2 *π^*_M is an optimal policy for M.*

Proof outline: By induction on n. For $n = 1$, we have an instance of CTP-DISJ-REP(1), M', which is also an instance of CTP-DISJ. Hence, by theorem 1, $\pi^*_{M'}$ is optimal.

We now assume inductively that π^*_M is an optimal policy for any instance of CTP-DISJ-REP($n-1$), and show that this property holds for all instances of CTP-DISJ-REP(n) as well. Let M be an instance of CTP-DISJ-REP(n). Given a committing policy $\overline{\pi}$ for M, denote by $\sigma_{\overline{\pi}}$ the permutation on $\{0, \cdots, k-1\}$ which is induced by the order in

which A_0 tries the paths in $\overline{\pi}$. For σ, and ν, permutations on $\{0, \cdots, k-1\}$, let $d(\sigma, \nu)$ be the "Euclidean" distance between σ, and ν (the Euclidean distance is known as $d(\sigma, \nu) \stackrel{\text{def}}{=} \sqrt{\sum_{i<k}(\sigma(i) - \nu(i))^2}$).

A CTP instance M can be represented as a directed action/outcome AND/OR tree T_M, and a policy π can be represented by an AND-subtree T_π of T_M. For a general policy π', let an INV*-edge* be any action-edge in $T_{\pi'}$ where the action taken by the agent is INV, and let $INV(T_{\pi'})$ be the number of INV-edges in $T_{\pi'}$. Since π^*_M is committing, then $INV(T_{\pi^*_M}) = 0$.

Let π be an optimal policy for M, with $INV(T_\pi)$ as small as possible, and such that if $INV(T_\pi) = 0$, then $d(\sigma_\pi, \sigma_*)$ is minimal. There are two cases:

(1:) π **is committing.** Then $INV(T_\pi) = 0$. We may assume that A_0 tries the paths in π in the order of $\{I_0, I_1, \cdots I_{k-1}\}$. By the induction assumption we have that $A_2, \cdots A_{n-1}$ follow A_1 in π. If A_1 also follows A_0, then π is a followers-committing policy, and by lemma 1, $c(\pi^*_M) \leq c(\pi)$, hence π^*_M is an optimal policy for M. If A_1 does not follow A_0, we can show that there is a path $u < k-1$ such that $u+1 <_* u$. We then define a policy π' which is the same as π, except that I_{u+1} is tried right before I_u. As π' is committing as well, $INV(T_{\pi'}) = INV(T_\pi) = 0$. Then we can show that π' is an optimal policy for M, and $d(\sigma_{\pi'}, \sigma_*) < d(\sigma_\pi, \sigma_*)$, contradicting the minimality of $d(\sigma_\pi, \sigma_*)$ among the committing optimal policies of M.

(2:) π **is not committing**. Then $INV(T_\pi) > 0$. We can then show that T_π contains a subtree, T, with only one INV-edge, and define a policy π' which is obtained from π by replacing T with another tree, T', which has no INV-edges at all. We then show that π' is optimal and $INV(T_{\pi'}) < INV(T_\pi)$, contradicting the minimality of $INV(T_\pi)$ among the optimal policies of M. □

Example 1 *Consider Fig. 1. We have* $\frac{E[TRY_1(I_0)]}{Q_0} = 39.5$*, and* $\frac{E[TRY_1(I_1)]}{Q_1} = 2.6$*. Hence by theorem 2, the optimal single agent policy is committing to try path* I_1 *before* I_0*. However,* $\frac{E[TRY_{38}(I_0)]}{Q_0} = 95$*, and* $\frac{E[TRY_{38}(I_1)]}{Q_1} = 95.1$*, hence for* $n \geq 38$ *agents, the optimal policy is for the first agent, to try path* I_0 *before* I_1*, and for the other agents to follow the first agent's path to* t*.*

Interleaved-action CTP in disjoint-path graphs We briefly consider interleaved action CTP in disjoint-path graphs (CTP-DISJ-MULTI(n)). Since agents can start moving before the first active agent has reached t, it is by no means clear that the optimal policy can be described using only TRY and INV macro actions. In fact, it is easy to see that for more general graphs, the optimal policy requires interleaved actions. For example, adding a (certainly traversible) path that costs 100 from v_1 to t in Fig. 1, the optimal 2-agent policy is to send the first agent to v_1, and if $e_{1,1}$ is blocked, send the second agent to v_0 to check $e_{0,1}$, while the first agent waits at v_1.

Since for disjoint paths this type of scenario cannot occur, we are led to suspect that there is no advantage to having more than one active agent at a time in this topology. We have checked this empirically by generating the optimal policies (using value iteration) with and without interleaved actions for small randomly generated problem instances. From hundreds of such non disjoint-path instances, more than 10% of the cases required interleaved actions to achieve the optimal policy. Conversely, in all of over a thousand such disjoint-path graph instances, the optimal policy for CTP-DISJ-REP(n) was also optimal for CTP–DISJ-MULTI(n). Hence we state the following:

Conjecture: Any optimal policy for CTP-DISJ-REP(n) is also optimal for CTP-DISJ-MULTI(n).

Heuristics for repeated-task CTP

For general graphs, the size of the tree of the optimal policy, and the time to compute it, are exponential in the number of unknown edges even for single-agent CTP. Thus, such policies can be computed and described only for very small graphs. We thus examine non-optimal policies suggested for CTP in the literature. The *optimistic* policy (OPT) (Bnaya, Felner, and Shimony 2009; Eyerich, Keller, and Helmert 2010), is based on the "free-space assumption" (commonly used in robotics) which assumes that all *unknown edges* are traversable. The agent computes the shortest path under this optimistic assumption, and begins to traverse it. The shortest path from the current location to the goal is re-computed every time an edge on the currently traveresed path is found to be blocked. OPT is fast, but far from optimal in expected cost.

(Eyerich, Keller, and Helmert 2010) performed systematic comparison between sampling-based schemes. Their best sampling method (denoted as CTP-UCT) was based on the UCT scheme (Kocsis and Szepesvári 2006). They showed that CTP-UCT converges in the limit to the optimal behavior and outperforms other techniques for single agent-CTP. We revisit single-agent CTP-UCT and then generalize it to the multi-agent case.

UCT for Single-Agent CTP

CTP-UCT works by performing "rollouts" as follows. For each rollout a *weather* (defined above) is randomized. Status of *unknown edges* is revealed only when the search reaches the appropriate node. Starting with the initial belief state b_0, choose an action a_1 to explore. The choice is made according to the CTP-UCT formula defined below. This results in a new belief state b_1, depending on b_0, the current weather, and the action a_1. Then, a new action is chosen, resulting in a new belief state b_2. This is repeated until we reach a belief state b_m where the agent is located at t. This is a single rollout, involving a sequence of belief states $< b_0, b_1 .... b_m >$.

Let $\sigma =< b_0, b_1 .... b_i >$ be a partial sequence of belief states generated in the (k+1)-th rollout where the agent location in b_i is not t. Define: **(1:)** $R^k(\sigma)$ to be the number of rollouts among the first k rollouts that start with σ **(2:)** $C^k(\sigma)$ to be the average cost incurred on these rollouts from the end of σ to the goal.

There are several possible successor belief states $b'_1, b'_2 ...$, one for each possible action $a_1, a_2 ....$ Each successor b'_i, when appended to σ, results in an extended sequence of belief states denoted by σ_i. Based on (Kocsis and Szepesvári 2006), CTP-UCT picks the action whose successor state maximizes:

$$UCT(\sigma_i) = B\sqrt{\frac{logR^k(\sigma)}{R^k(\sigma_i)}} - cost(a_i) - C^k(\sigma_i)$$

where $B > 0$ is a parameter, and $cost(a_i)$ is the cost of the edge traversed by action a_i. The idea here is to allow "exploration" initially (i.e. to sample previously little-visited actions), due to first term in the equation. Eventually, when many rollouts are performed the equation gives more weight to minimizing the expected travel cost as expressed by the last 2 terms in the equation. In order to direct the sampling better towards the goal, the value $C^k(\sigma_i)$ is initially based on the optimistic cost, based on the free-space assumption.

After completing a pre-determined number of rollouts, an agent performs the action with the minimal average cost to reach the goal, where averaging is computed based on the rollouts. This results in a new belief state actually reached by the agent, and the process (of CTP-UCT) is repeated until the agent reached the goal.

Generalization of UCT to Repeated-Task CTP

In CTP-REP, knowledge revealed to an agent is immediately shared with all other agents. All agents share a *known graph* which includes all edges known to be traversable. There are 4 different ways to generalize UCT to work for repeated CTP depending on the following attributes.

1: behavior of consecutive agents. After the first agent activates CTP-UCT there are two possible behaviors of the remaining agents. They can be *reasoning agents*, that is, they also active CTP-UCT. Or, they can be *followers*: Since the first agent has arrived at the goal, the known graph includes at least one path to the goal. The shortest such known path is *followed* by the remaining $n-1$ agents.

2: considering consecutive agents. An agent can be *inconsiderate*, ignore the remaining agents, and use the CTP-UCT formula of the single agent case. By contrast, it can be *considerate*. In this case, the agent *considers* the fact that the remaining $n-1$ agents are expected to travel to the goal too and therefore their expected travel cost should be taken into account. One way to do it is by assuming that consecutive agents will follow the shortest known path. This idea is based on the theoretical results for disjoint graphs, where this policy is actually optimal. While no longer necessarily optimal for general graphs, it is used here as a heuristic. In order to implement this scheme, we modify the CTP-UCT formula by adding an additional F^k term (stands for "future path"), which is the cost of the shortest path we expect to discover at the end of the rollout. This is multiplied by $n-1$, since $n-1$ agents are expected to follow that shortest path as well. We therefore use the following modified UCT rule:

$$UCTR(\sigma_i) = UCT(\sigma_i) - (n-1)F^k(\sigma_i)$$

where $F^k(\sigma_i)$ is the average over k rollouts of the shortest path from s to t among edges known to be unblocked in belief states at the end of the rollout starting from σ_i. As this quantity is not well defined initially, it is initialized based on the value of the shortest path from s to t under the free-space assumption.

Example 1 demonstrates the difference between a considerate and an inconsiderate agent, as an inconsiderate first agent always chooses to try I_1 before I_0 no matter how many agents follow it. There are four possible combinations, which lead to four possible variants that generalize CTP-UCT as follows.

(1:) Inconsiderate, followers. In this variant, only the first agent activates CTP-UCT for choosing its actions. The first agent *inconsiderately* ignores the existence of the other agents and activates CTP-UCT for choosing its actions. The other agents are *followers*. This variant is labeled UCTR1.

(2:) Considerate, followers.. The first agent activates UCTR. The other agents are *followers*. This variant is labeled UCTR2.

(3:) Inconsiderate, reasoning. Here, *all* agents are *reasoning* agents but they are all *inconsiderate*, i.e., they all activate CTP-UCT. This variant is labeled UCTR3.

(4:) Considerate, reasoning. *All* agents are *reasoning* agents and they all activate UCTR. This variant is labeled UCTR4.

Empirical Evaluation

Our experiments included instances with small number of unknown edges. This enables the evaluation of a given policy under all possible weathers and measuring the policy performance accurately as all cases have been evaluated. We experimented on problem instances based on the graph depicted in figure 2. In the figure, the source and target vertices are labeled s and t, respectively. Each edge is labeled with the edge weight.

We generated two sets of instances where we randomly selected 4 and 8 unknown edges (not adjacent to s) respectively. We also uniformly select blocking-probabilities for each of the unknown edges in the range of $(0.01, 0.9)$. Similar trends were observed and we only report results on the graph with 8 unknown edges.

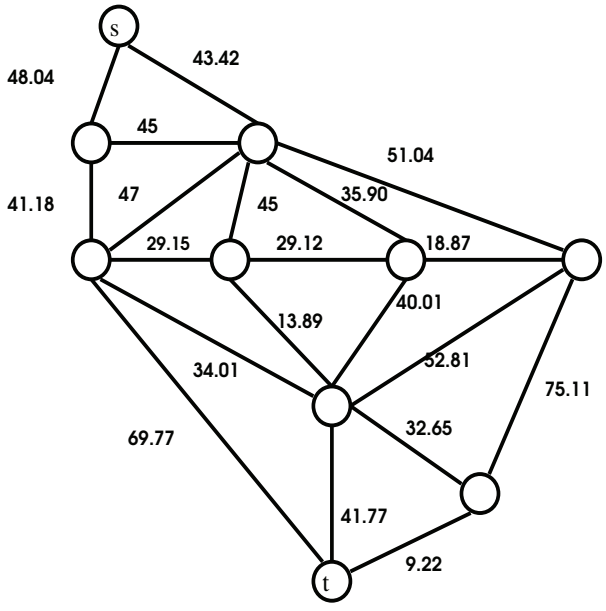

Figure 2: Our example graph.

We implemented six policies for CTP-REP. First, we computed the optimal policy using value iteration. This serves as a lower bound on the travel cost. Second, we implemented a bound we call *cautious-blind* (C-Blind in the tables). The cautious-blind uses only the known edges of

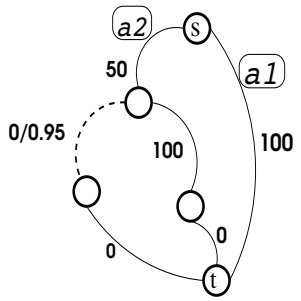

Figure 3: Explicit graph example.

the graph to find a shortest path from source to target and then follows it. If any unknown edges are revealed to the agent while traversing, the agent ignores this information. Cautious-blind serves only to indicate a type of "worst-case" and thus we define a range of costs that makes sense for each problem instance. Finally, the expected costs for our four UCTR variants described above are achieved by averaging simulations of the policy under each possible weather of an instance. The travel cost incurred in each simulation is weighted according to the probability of the weather. Therefore, this gives the expected cost of the policy on a given instance. UCTR variants were tried with 10000 to 200000 rollouts.

agents	n=2	n=3	n=10
Optimal	15.45	35.37	361.00
UCTR1	1.72	1.73	1.73
UCTR2	1.77	1.84	1.94
UCTR3	3.15	4.23	6.23
UCTR4	3.25	4.54	6.34

Table 1: Execution times in minutes

Table 1 presents the execution times in minutes required to generate the polices on a typical instance with 8 unknown edges. Clearly, as the number of agents, n, grows, computing the optimal policy consumes significantly more time than any other policy. UCTR1, and UCTR2 are both policies that are based on our "followers" principle. Thus, the execution of UCT is only performed once, for the first agent. As a result, the time it takes to calculate these policies is significantly smaller than UCTR3 and UCTR4.

We have performed a number of experiments and did not see evidence that the *reasoning* policies UCTR3 and UCTR4 show significant improvements over the *followers* polices UCTR1 and UCTR2. Since the time spent to compute the *reasoning* policies is much larger than the *followers* polices, we conclude that these versions are not recommended. This bolsters our theoretical results, which show that following is optimal on disjoint paths: while not optimal in general, it is a strong heuristic. We thus only report the total cost of the *followers* polices UCTR1 and UCTR2.

Table 2 presents the average costs incurred on 10 instances of our graph (with 8 unknown edges) by the two benchmark algorithms and by the followers policies UCTR1 and UCTR2 when the number of agents n was varied. Table 3 presents results on two representative instances. It is important to observe from both tables that both UCTR1 and UCTR2 are very close to the optimal policy, and are far better than the cautious-blind bound. This strengthen our claim that the optimal behavior for the disjoint-path graphs is a strong heuristic for general graphs.

	n=1	n=2	n=3	n=10
Optimal	154.87	307.09	459.32	1524.89
UCTR1	155.02	309.75	464.48	1547.59
UCTR2	155.02	307.38	464.24	1548.75
C-Blind	161.18	322.36	485.54	1611.8

Table 2: Expected cost averages

	Instance 1			Instance 2		
	n=1	n=3	n=10	n=1	n=3	n=10
Optimal	164.32	487.03	1615.52	150.14	450.09	1499.60
UCTR1	164.32	487.36	1618.00	150.14	450.25	1500.64
UCTR2	164.32	487.13	1619.32	150.14	450.13	1500.06
C-Blind	178.79	536.37	1787.90	164.10	492.30	1641.00

Table 3: Results for the set of 20 edges with 8 unknown

When the number of agents n is small, UCTR2 slightly outperforms UCTR1. When n is larger (10 agents), results are inconclusive. As indicated by varying the number of rollouts (not shown), the reason is that with many agents the "considerate" assumption and the large extra F^k term in the UCTR formula (especially as it is multiplied by the number of remaining agents) amplifies the sampling noise. This noise is hard to overcome unless the number of rollouts is increased drastically.

Discussion

UCTR2 outperforms UCTR1 if some policy Π_c exists (c for *considerate*) such that using Π_c on CTP-REP has lower expected cost than taking the "selfish" policy Π_s where each agent tries to minimize it's own cost function. We expect UCTR2 to resolve with Π_c and UCTR1 to resolve with Π_s.

Denote Π^1 as the policy that the first agent takes and Π^F as the "followers" policy, used by the rest of the agents. In general, $E[\Pi^1_s] \leq E[\Pi^1_c]$ since Π^1_s tries to minimize the cost function without the burden of the follower agents.

Despite this fact, UCTR2 will outperforms UCTR1 when:

$$E[\Pi^1_c] + (n-1) \times E[\Pi^F_c] < E[\Pi^1_s] + (n-1) \times E[\Pi^F_s].$$

The graph in figure 3 demonstrates such a case, and indeed UCTR2 outperforms UCTR1 when the number of agents is large enough. In this instance, all edges are known except the dashed edge which has blocking probability of 0.95.

On the initial state s, $\Pi^1(s)$ chosen by UCTR1 results in taking action a_1 (traverse a_1) with cost 100 since taking the alternative action a_2 evaluates to $0.05 \times 50 + 0.95 \times 150 = 145$. The follower policy $\Pi^F(s)$ also follows a_1 and evaluates to 100.

Now, let's consider UCTR2. If $\Pi^1(s)$ chooses to take a_1 then the follower policy, $\Pi^F(s)$ evaluates to 100, as in UCTR1. However if $\Pi^1(s)$ takes a_2 then the value of $\Pi^F(s)$ will be $0.95 \times 100 + 0.05 \times 50 = 97.5$ which is slightly better

for the followers. It is beneficial for UCT2 to choose $\Pi^1(s)$ to be a_2 when the extra cost incurred by $\Pi^1(s)$ is XXX by the advantage gained by the followers. Therefore, it is beneficial to take a_2 only when $(n-1)\times(100-97.5) > 145-100$ or $n > 19$. Table 4 presents the weighted cost of the two versions as a function of the number of agents.

n	UCTR1	$\Pi^1(s) = a_1$	$\Pi^1(s) = a_2$	UCTR2
18	1800	1800	1802.5	1800
19	1900	1900	1900	1900
20	2000	2000	1997.5	**1997.5**
21	2100	2100	2095	**2095**

Table 4: Expected cost averages

We also experimented with instances that have large number of unknown edges (up to 20). In this case, evaluating the policy against all weathers is not practical therefor only a subset of weathers is used. In these experiments we get inconclusive results - on some cases UCTR-1 seems to outperform UCTR-2 probably. In such cases, although the above condition holds, experimenting with only a subset of weathers does not necessarily demonstrate the benefit of using UCTR2 because of the negative skewness of $E[\Pi_c^2]$.

Conclusions

Repeated task CTP is introduced, and an optimal *followers* policy for the special case of disjoint-path graphs is defined and proved. This *followers* policy forms the basis of heuristics for reasonable non-optimal policies for general graphs, as suggested by an empirical evaluation on using a suite of sampling algorithms based on UCT. While optimal for disjoint-path graphs, the *followers* policy is very strong and is near optimal for general graphs.

The more general interleaved-agent CTP was also introduced. However, this problem is more complicated due to the much larger state and action spaces. For disjoint-path graphs, empirical results indicate that the *followers* policy is optimal, and that there is nothing to gain by allowing interleaved actions. Proving (or disproving) this conjecture would be a good way to continue research on this CTP variant. Experimental results demonstrate cases where UCTR1 outperformed UCTR2. Future research will focus further in the interleave version of the problem. Similarly, experiments on different types of graphs will better reveal which policy performs best under what circumstances.

References

Bar-Noy, A., and Schieber, B. 1991. The Canadian Traveller Problem. In *SODA*, 261–270.

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

Bonet, B. 2009. Deterministic POMDPs Revisited. In *(UAI). Montreal, Canada*.

Eyerich, P.; Keller, T.; and Helmert, M. 2010. High-Quality Policies for the Canadian Traveler's Problem. In *AAAI 2010*.

Hansen, E. A. 2007. Indefinite-horizon POMDPs with action-based termination. In *AAAI*, 1237–1242.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *ECML*, 282–293.

Littman, M. 1996. *Algorithms for Sequnetial Decision Making*. Ph.D. Dissertation, Brown University.

Meshulam, R.; Felner, A.; and Kraus, S. 2005. Utility-based multi-agent system for performing repeated navigation tasks. In *AAMAS*, 887–894.

Nikolova, E., and Karger, D. R. 2008. Route Planning under Uncertainty: The Canadian Traveller Problem. *AAAI* 969–974.

Papadimitriou, C. H., and Yannakakis, M. 1991. Shortest Paths Without a Map. *Theor. Comput. Sci.* 84(1):127–150.