Maximizing the Probability of Arriving on Time

Ananya Das and Joao Cassamano

Math & Computer Science Department,
Lake Forest College, Lake Forest IL
{adas,cassamanojc}@lakeforest.edu

Abstract. *We study the problem of maximizing the probability of arriving on time in a stochastic network. Nodes and links in the network may be congested or uncongested, and their states change over time and are based on states of adjacent nodes. Given a source, destination, and time limit, the goal is to adaptively choose the next node to visit to maximize the probability of arriving to the destination on time. We present a dynamic programming solution to solve this problem. We also consider a variation of this problem where the traveler is allowed the option to wait at a node rather than visit the next node. For this setting, we identify properties of networks for which the optimal solution does not require revisiting nodes.*

Keywords: stochastic networks, routing, dynamic programming

1 Introduction

Transportation networks are inherently uncertain with random events such as accidents, vehicle failure, inclement weather, and road closures often causing congestion and delays. These events often affect a group of locations and roads in a region of the network, rather than just one location. Congestion at one location is likely to cause congestion at nearby locations and roads. Fortunately, new technologies make it increasingly easy for travelers to obtain real-time information about traffic conditions, allowing them to make potentially better route decisions. This setting can be modeled with a stochastic network where a node can be in one of two states: congested or uncongested. Congestion at one node is likely to cause congestion at nearby nodes and congestion at a node causes congestion at its incident edges. Traversing an edge in the congested state requires more time than traversing the edge when it is uncongested. Since the state of a node may change over time and cannot be determined until the node is reached, there is uncertainty with both node and link states.

For this setting, we consider the problem of maximizing the probability of arriving at a specified destination node within a given amount of time. Since the network is stochastic, we cannot solve this problem *a-priori*. Instead, we solve the problem using a step-by-step approach where for each step, we make a decision based on the state of the current node, the remaining time, and the conditional congestion probabilities of adjacent nodes.

Specifically, given source and destination locations, and a maximum time limit, we propose an efficient dynamic programming solution to find a path from the source to the destination that maximizes the probability of arriving by the time limit. We find that some optimal routes require revisiting nodes, which may not be reflective of realistic networks. Therefore, we also consider a variation of this problem where the traveler has the option to wait at a node rather than visit another node. We identify properties of networks for which the optimal solution requires waiting at a node rather than revisiting nodes.

2 Related Work

The problem of finding an optimal path in a stochastic network has been studied extensively and numerous variations of the problem have been considered. Some early examples include [5] and [12] which consider networks where link costs are random variables following known probability distributions. The work in [5] studies the problem of finding the path that maximizes the probability of arriving by a predefined time whereas [12] finds the path that has the highest probability of being the shortest path. The authors of [11] assume a network where link travel times evolve based on an independent Markov process. They give solutions for finding a path with minimal expected travel time. The authors of [6] consider a setting where link travel times depend on the time of day and propose heuristics to estimate the mean and variance of arrival times for a source-destination node pair. In [13], the authors study a network setting in which the state of each link is dependent on the predecessor link, and is independent of the states of nodes. They develop heuristics to determine the sequence of links to traverse such that the expected travel time is minimized. For different settings, several works consider the problem of maximizing the expected value of various utility functions ([2], [8], [9]). However, the authors of [8] show that such utility-based models can be solved efficiently only if an affine or exponential utility function is employed.

More recently, the authors of [3] considered a network where link costs are based on conditional probabilities of adjacent nodes. They consider the problem of finding the path with the minimum expected travel time. The authors of [4] consider a network where link costs follow independent probability distributions. They give an approximate solution to the problem of finding the path that maximizes the probability of arriving to a destination by a predetermined time. The authors of [10] also assume link travel times are defined by a probability distribution and propose an algorithm to address a similar problem: finding the shortest paths to guarantee a given probability of arriving on-time. Recently, the authors of [7] consider a network setting where the traveler has information about several link travel times (not just adjacent links) and give heuristics for maximizing a general utility function.

In this work we consider the stochastic setting studied in [3] where the congestion probability for nodes and links depend on nearby nodes. Whereas they focus on minimizing expected travel time, we focus on a different goal: given a source, destination, and desired arrival time, find the sequence of nodes to visit that maximizes the probability of arriving on time. Whereas previous studies proposed approximate solutions to routing problems in stochastic networks ([4]), we provide a dynamic programming solution that, given a maximum time limit, yields an exact solution to the problem. We also consider the cycling policy, described in [13], where the traveler has the option to revisit locations; and the waiting policy, described in [1], where the traveler has the option to wait at a location. Both policies may help to improve the objective function. We compare these policies by identifying networks for which the waiting policy will always be more beneficial than the cycling policy.

3 Problem Statement

We consider the Arriving on Time Problem. The input is a network with n nodes, a source node s, a destination node d, and a time limit t. The goal is to find the path that maximizes the probability of arriving to d from s within time t. Since the states of nodes can change, we cannot determine the optimal path *a-priori*. Instead, once we arrive at a node, we must adaptively determine the next node to visit that maximizes the probability of reaching d given the remaining time. We assume that there is a time limit T_{max} such that for every pair of vertices i and j, t cannot exceed T_{max}.

For adjacent nodes i and j, let $P_c(j|i)$ denote the conditional probability that j is congested given that i is congested and let $P_u(j|i)$ denote the conditional probability that j is uncongested given that i is uncongested. Then if i is congested, j is uncongested with probability $1 - P_c(j|i)$ and if i is uncongested then j is congested with probability $1 - P_u(j|i)$. To reach node j directly from i, we must traverse the edge (i, j). It takes $t_c(i, j) > 0$ time units to traverse (i, j) if i is congested and $t_u(i, j) > 0$ time units if i is uncongested. We assume the uncongested travel time is no more than the congested travel time so $t_u(i, j) \leq t_c(i, j)$. For simplicity, we assume that if i is congested, we can reach an adjacent node j within t time units with probability 0 if $t < t_c(i, j)$ and with probability 1 if $t \geq t_c(i, j)$. Similarly if i is uncongested, we can reach an adjacent node j within t time units with probability 0 if $t < t_u(i, j)$ and with probability 1 if $t \geq t_u(i, j)$. Note that extending this model so that these probabilities are instead determined by a distribution is straightforward. Let $N_c^t(i)$ denote the set of nodes, j, such that j is adjacent to i and $t \geq t_c(i, j)$ and let $N_u^t(i)$ denote the set of nodes, j, such that j is adjacent to i and $t \geq t_u(i, j)$. Then if node i is congested or uncongested, we consider all nodes in $N_c^t(i)$ or $N_u^t(i)$, respectively, for the next possible node to visit.

In Fig. 1(a), suppose we determine that j is the optimal node to visit from i. Since i is congested, traversing link (i, j) will take time $t_c(i, j)$ and in Fig. 1(b), since i is uncongested, traversing link (i, j) will take time $t_u(i, j)$. In either case, once we arrive at node j, if j is congested then traversing any link from j will take time equal to the congested time of the link. If j is uncongested then traversing any link from j will take time equal to the uncongested time of the link. Once we arrive at j we find the optimal next node to visit as we did for node i. Specifically, we find the node that yields the maximum on-time arrival probability from j given the amount of time remaining.

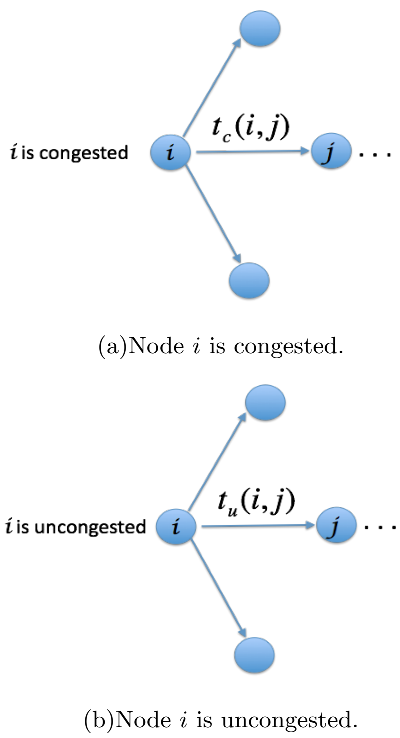


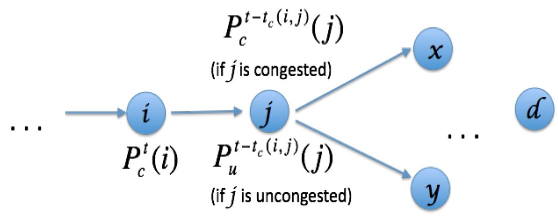
Fig. 1. Traversing link (i, j). (a) Since i is congested, traversing link (i, j) takes time $t_c(i, j)$. (b) Since i is uncongested, traversing link (i, j) takes time $t_u(i, j)$. When we arrive at j, it will be either congested or uncongested. In either case we must find the next node to visit that maximizes the on-time arrival probability from j.

For all nodes i and some destination, let $P_c^t(i)$ denote the maximum probability of arriving to the destination on time given t time units when i is congested; and let $P_u^t(i)$ denote the maximum on-time arrival probability given t time units when i is uncongested. Suppose node i is currently congested. If j is the optimal next node to visit, then when we move from i to j, j will either be congested (with probability $P_c(j|i)$) or uncongested (with probability $1 - P_c(j|i)$). In either case, we will have $t - t_c(i, j)$ time units remaining and we would like to maximize the probability of arriving on time from j to the destination with the remaining time. In Fig. 2(a), if j is congested, the probability of arriving to the destination on time from j is $P_c^{t-t_c(i,j)}(j)$. If j is uncongested, the probability is $P_u^{t-t_c(i,j)}(j)$. If instead, i was uncongested (see Fig. 2(b)), then the maximum on-time arrival probability from j is $P_c^{t-t_u(i,j)}(j)$ (if j is congested) or $P_u^{t-t_u(i,j)}(j)$ (if j is uncongested).

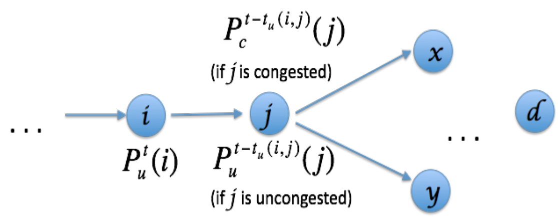
In this manner, we can determine the optimal sequence of nodes to visit from the source to the destination and the maximum probability of arriving on time. Notice that we cannot determine the optimal node to visit from j until we arrive at j and observe its state. For example, if j is congested, the optimal next node may be x and if j is uncongested the optimal next node may be y.

Figure 3 shows a simple example of the problem. Assume the source is s and the destination is d. For the conditional probabilities, assume $P_c(a|s) = .9$, $P_c(b|s) = .6$, $P_u(a|s) = .7$, and $P_u(b|s) = .6$. For the travel times, assume $t_c(s, a) = 6$, $t_c(a, d) = 7$, $t_c(s, b) = 5$, $t_c(b, d) = 9$, $t_u(s, a) = 5$, $t_u(a, d) = 4$, $t_u(s, b) = 4$, and $t_u(b, d) = 3$. The optimal path depends on both the state of s and the time limit. If s is congested, then with a time limit of $t = 10$ units, the optimal path is s-b-d with probability $P_c^{10}(s) = .4$, whereas path s-a-d yields probability .1. However, if s is uncongested, then the optimal path is s-a-d with probability $P_u^{10}(s) = .7$, whereas path s-b-d yields probability .6. If we have a time limit of $t = 7$ and s is uncongested, then the optimal path is s-b-d with probability $P_c^7(s) = .6$, whereas s-a-d yields probability 0. If s is congested, then both paths yield probability 0.

The general problem can be solved with the following dynamic programming formulation:



(a)Node i is congested.



(b)Node i is uncongested.

Fig. 2. Traversing a graph for the Arriving on Time Problem. (a) Since i is congested, link (i, j) costs $t_c(i, j)$ so we have $t - t_c(i, j)$ time units remaining when we arrive at j. (b) Since i is uncongested, link (i, j) costs $t_u(i, j)$ so we have $t - t_u(i, j)$ time units remaining. When we arrive at j, it will be either congested or uncongested. In either case we must find the next node to visit that maximizes the on-time arrival probability from j.

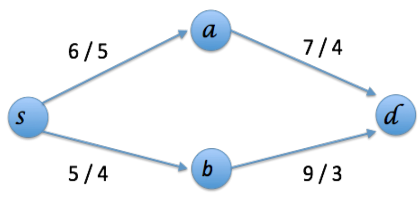


Fig. 3. Example of the Arriving On-Time Problem. Congested/uncongested travel times are shown beside each edge. In this example, the optimal path depends on the state of the source node and the time limit.

$$P_c^0(i) = 0 \text{ if } i \text{ is not the destination} \tag{1}$$
$$P_u^0(i) = 0 \text{ if } i \text{ is not the destination} \tag{2}$$
$$P_c^t(i) = 1 \text{ for all } t \geq 0 \text{ if } i \text{ is the destination} \tag{3}$$
$$P_u^t(i) = 1 \text{ for all } t \geq 0 \text{ if } i \text{ is the destination} \tag{4}$$

$$P_c^t(i) = 0 \text{ if } i \text{ is not the destination and } t < t_c(i,j) \text{ for all } j \text{ adjacent to } i$$
$$\max_{\forall j \in N_c^t(i)} \{P_c(j|i) P_c^{t-t_c(i,j)}(j) + (1 - P_c(j|i)) P_u^{t-t_c(i,j)}(j)\} \text{ otherwise}$$

$$P_u^t(i) = 0 \text{ if } i \text{ is not the destination and } t < t_u(i,j) \text{ for all } j \text{ adjacent to } i$$
$$\max_{\forall j \in N_u^t(i)} \{P_u(j|i) P_u^{t-t_u(i,j)}(j) + (1 - P_u(j|i)) P_c^{t-t_u(i,j)}(j)\} \text{ otherwise}$$

The algorithm can be implemented in a manner similar to Dijkstra's shortest path algorithm. For a graph with n nodes and maximum time limit T_{max}, it requires $O(n^2 T_{max})$ space and time.

Proposition 1. *The quantities $P_c^t(i)$ and $P_u^t(i)$ are non-decreasing in t.*

Proof. We will prove that $P_c^t(i) \geq P_c^{t-1}(i)$ for all $t \geq 1$. The proof also holds for $P_u^t(i)$.

Base Case: If i is the destination then for all $t \geq 1$, $P_c^t(i) = 1$ so $P_c^t(i) = P_c^{t-1}(i)$. If i is not the destination, then for $t = 1$, $P_c^{t-1}(i) = P_c^0(i) = 0$, so $P_c^t(i) = P_c^1 \geq P_c^0(i)$.

Inductive Hypothesis. Assume $P_c^t(i) \geq P_c^{t-1}(i)$ and $P_u^t(i) \geq P_u^{t-1}(i)$.

We will show $P_c^{t+1}(i) \geq P_c^t(i)$.

$$P_c^{t+1}(i) = \max_{\forall j \in N_c^t(i)} \{P_c(j|i) P_c^{t+1-t_c(i,j)}(j) + (1 - P_c(j|i)) P_u^{t+1-t_c(i,j)}(j)\}$$

By the inductive hypothesis, we know $P_c^{t+1-t_c(i,j)}(j) \geq P_c^{t-t_c(i,j)}(j)$ and $P_u^{t+1-t_c(i,j)}(j) \geq P_u^{t-t_c(i,j)}(j)$
.

$$\begin{aligned} P_c^{t+1}(i) &\geq \max_{\forall j \in N_c^t(i)} \{P_c(j|i) P_c^{t-t_c(i,j)}(j) + (1 - P_c(j|i)) P_u^{t-t_c(i,j)}(j)\} \\ &= P_c^t(i) \end{aligned}$$

Similarly, $P_u^{t+1}(i) \geq P_u^t(i)$.

3.1 Node Revisiting

In the current problem formulation, the optimal route from the source to the destination may include revisiting one or more nodes. For example, suppose we arrive at a congested node i and there is a short path from i to the destination that yields a low on-time arrival probability if taken when i is congested (see Fig. 4). Suppose from i, we can also traverse a cycle such that the probability of returning to i in the congested state is small. Then, the optimal solution may be to traverse this cycle if doing so yields a higher on-time arrival probability than directly heading towards the destination from i in the congested state. The subgraph in Fig. 4 shows an example where continuously revisiting a node improves the probability of arriving on time. Suppose we would like to arrive at d from s in the congested state. Assume $P_c(i|j) = P_c(j|i) = .6$, and $P_c(i|s) = P_u(i|j) = P_u(j|i) = .9$. Assume $t_c(i,d) = 10$ and $t_u(i,d) = 1$. Assume that both the congested and uncongested travel times of all other edges are one time unit, so $t_c(x,y) = t_u(x,y) = 1$ for all edges (x,y) except (i,d). In this example, traveling directly from s to d without traversing the cycle containing nodes i and j yields probability $P_c^2(s) = 0.1$. However, with two additional time units, traversing the cycle containing i and j, specifically with the path s–i–j–i–d, yields

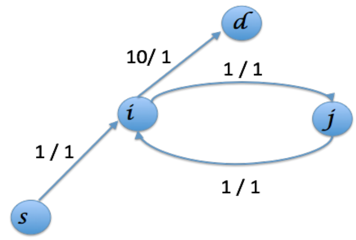


Fig. 4. Example where revisiting node i improves the on-time arrival probability. Congested/uncongested travel times are shown beside each edge.

probability $P_c^4(s) = .625$. Similarly, traversing the cycle twice and thrice yields probabilities $P_c^6(s) = .756$ and $P_c^8(s) = .789$, respectively. The example shows that revisiting a node may help to improve the probability of arriving to the destination within the time limit.

However, this scenario does not reflect realistic situations. Specifically, if we are at a congested location, traversing a cycle to revisit the location would never yield the highest probability of arriving on time. If heading directly towards the destination from a congested location is unlikely to get us to the destination on time, then a better option would be to simply *wait* at the current location (for a short time) for the congestion to clear up.

It is possible that after waiting for some time, the location will become uncongested and therefore yield a higher probability of arriving on time. When the location becomes uncongested, we can then decide which route will yield the highest on-time arrival probability. In the following section we consider a variation of the problem that models this more realistic setting.

4 Waiting Option

We now present a version of the Arriving on Time Problem where the traveler has the option to wait at a congested node (in the hopes that the node will become uncongested). As described above, in real-world situations, we may decide to wait at a congested location for traffic to clear up and the location to become uncongested. However, we would find no reason to wait at an uncongested location for it to become congested. Therefore, we assume that the option to wait only applies to a node in the congested state.

As before, $P_c(j|i)$ denotes the conditional probability that j is congested given that i is congested, $P_u(j|i)$ denotes the conditional probability that j is uncongested given that i is uncongested, $t_c(i,j)$ and $t_u(i,j)$ denote the times to traverse edge (i,j) when i is congested and uncongested, respectively. We also define a new term $P_c^*(i,w)$, which is the conditional probability that i is congested after waiting at congested i for w time units. As in the original problem, we limit t and w to T_{max}.

As before, $P_c^t(i)$ denotes the maximum probability of arriving to the destination from i in the congested state, within at most t time units and $P_u^t(i)$ denotes the probability of arriving to the destination from i in the uncongested state within at most t time units. Given a source s and a destination, our goal is find the find the sequence of nodes to visit that maximizes $P_c^t(s)$ and $P_u^t(s)$.

Let us consider the previous example in Fig. 3. If s was congested and the time limit was $t = 10$, then paths s-b-d and s-a-d yielded on-time arrival probabilities of .4 and .1, respectively. Now, suppose we have the option to wait at a node. Assume $P_c^*(a,1) = .1$, so with probability .1, node a stays congested if we wait at a for one time unit. For simplicity, assume this probability is zero for all other nodes. Then if s is congested, the optimal solution is to wait at a for one time unit, i.e. by using path s-a-a-d, with probability $P_c^{10}(s) = .91$.

This problem can be solved with the following dynamic programming formulation:

$$P_c^0(i) = 0 \text{ if } i \text{ is not the destination} \tag{5}$$
$$P_u^0(i) = 0 \text{ if } i \text{ is not the destination} \tag{6}$$
$$P_c^t(i) = 1 \text{ for all } t \geq 0 \text{ if } i \text{ is the destination} \tag{7}$$
$$P_u^t(i) = 1 \text{ for all } t \geq 0 \text{ if } i \text{ is the destination} \tag{8}$$

$$P_c^t(i) = 0 \text{ if } i \text{ is not the destination and } t < t_c(i,j) \text{ for all } j \text{ adjacent to } i \tag{9}$$

$$\max \begin{cases} \max_{\forall j \in N_c^t(i)} \{P_c(j|i) P_c^{t-t_c(i,j)}(j) & \text{(Not waiting at } i) \\ \qquad +(1 - P_c(j|i)) P_u^{t-t_c(i,j)}(j)\} & \\ & \\ \max_{\forall w \leq T_{max}} \{P_c^*(i,w) P_c^{t-w}(i) & \\ \qquad +(1 - P_c^*(i,w)) P_u^{t-w}(i)\} & \text{(Waiting at } i) \end{cases} \tag{10}$$

$$P_u^t(i) = 0 \text{ if } i \text{ is not the destination and } t < t_c(i,j) \text{ for all } j \text{ adjacent to } i \tag{11}$$
$$\max_{\forall j \in N_u^t(i)} \{P_u(j|i) P_u^{t-t_u(i,j)}(j) + (1 - P_u(j|i)) P_c^{t-t_u(i,j)}(j)\} \tag{12}$$

Proposition 2. *The quantities $P_c^t(i)$ and $P_u^t(i)$ are non-decreasing in t.*

Proof. We first prove the proposition for $P_c^t(i)$. We will prove that $P_c^t(i) \geq P_c^{t-1}(i)$ for all $t \geq 1$.

Base Case: If i is the destination then for all $t \geq 1$, $P_c^t(i) = 1$ so $P_c^t(i) = P_c^{t-1}(i)$. If i is not the destination, then for $t = 1$, $P_c^{t-1}(i) = P_c^0(i) = 0$, so $P_c^t(i) = P_c^1(i) \geq P_c^0(i)$.

Inductive Hypothesis. Assume $P_c^t(i) \geq P_c^{t-1}(i)$ and $P_u^t(i) \geq P_u^{t-1}(i)$.

We will show $P_c^{t+1}(i) \geq P_c^t(i)$. From 10, we have:

$$P_c^{t+1}(i) = \max \begin{cases} \max_{\forall j \in N_c^t(i)} \{P_c(j|i) P_c^{t+1-t_c(i,j)}(j) & \\ \qquad +(1 - P_c(j|i)) P_u^{t+1-t_c(i,j)}(j)\} & \text{(Not waiting at } i) \\ & \\ \max_{\forall w \leq T_{max}} \{P_c^*(i,w) P_c^{t+1-w}(i) & \\ \qquad +(1 - P_c^*(i,w)) P_u^{t+1-w}(i)\} & \text{(Waiting at } i) \end{cases}$$

$$\geq \max \begin{cases} \max_{\forall j \in N_c^t(i)} \{P_c(j|i) P_c^{t-t_c(i,j)}(j) & \\ \qquad +(1 - P_c(j|i)) P_u^{t-t_c(i,j)}(j)\} & \\ & \text{(By Ind. Hyp.)} \\ \max_{\forall w \leq T_{max}} \{P_c^*(i,w) P_c^{t-w}(i) & \\ \qquad +(1 - P_c^*(i,w)) P_u^{t-w}(i)\} & \end{cases}$$
$$= P_c^t(i)$$

Proof. We now prove that $P_u^t(i)$ is non-decreasing in t. We show that $P_u^t(i) \geq P_u^{t-1}(i)$ for all $t \geq 1$.

Base Case: If i is the destination then for all $t \geq 1$, $P_u^t(i) = 1$ so $P_u^t(i) = P_u^{t-1}(i)$. If i is not the destination, then for $t = 1$, $P_u^{t-1}(i) = P_u^0(i) = 0$, so $P_u^t(i) = P_u^1 \geq P_u^0(i)$.

Inductive Hypothesis. Assume $P_c^t(i) \geq P_c^{t-1}(i)$ and $P_u^t(i) \geq P_u^{t-1}(i)$

We will show $P_u^{t+1}(i) \geq P_u^t(i)$. From 12, we have:

$$
\begin{aligned}
P_u^{t+1}(i) &= \max_{\forall j \in N_u^t(i)} \{P_u(j|i) P_u^{t+1-t_u(i,j)}(j) + (1 - P_u(j|i)) P_c^{t+1-t_u(i,j)}(j)\} \\
&\geq \max_{\forall j \in N_u^t(i)} \{P_u(j|i) P_u^{t-t_u(i,j)}(j) + (1 - P_u(j|i)) P_c^{t-t_u(i,j)}(j)\} \quad \text{(By Ind. Hyp.)} \\
&= P_u^t(i)
\end{aligned}
$$

4.1 Revisiting vs. Waiting

Recall that in the first version of the problem, where there is no option to wait at a node, revisiting a node may improve the probability of arriving on time. In this section we focus on the new version of the problem where the traveler has the option to wait at a node. We identify a class of networks for which revisiting nodes cannot improve the on-time arrival probability.

We will show that in this new model, revisiting a node cannot improve the on-time arrival probability for networks that exhibit both of the following two properties:

Property 1. Given some destination, for any node i and some fixed t, $P_u^t(i) > P_c^t(i)$. In other words, there is a higher chance of reaching the destination on time from location i if i is uncongested rather than congested. This property reflects realistic settings since the presence of traffic at the current location does not improve the probability of arriving to the destination on time.

Property 2. For any two neighbors i and j, and any w, $1 - P_c^*(i, w) > P_u(i|j)$ and $1 - P_c^*(i, w) > 1 - P_c(i|j)$. In other words, it is more probable for location i to become uncongested after waiting at i for w time units than by visiting i from a neighboring node j.

We now prove that for networks that exhibit Properties 1 and 2, revisiting a node cannot improve the on-time arrival probability. Specifically, we prove that for all nodes i, the arrival probability achieved by *revisiting* i after k time units will be no more than the on-time arrival probability achieved by *waiting* at a node i for k time units.

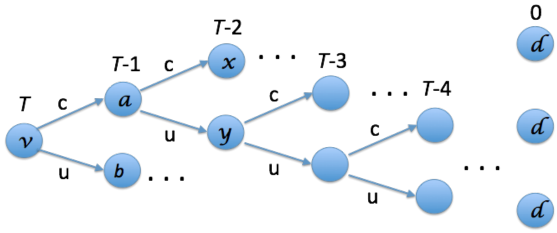
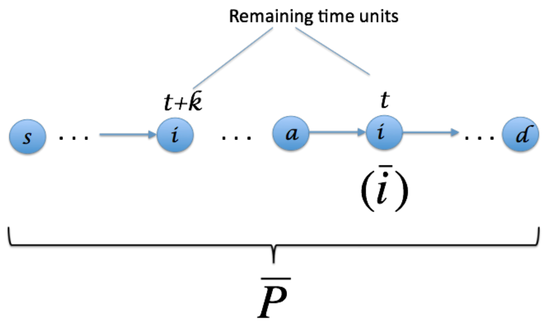


Fig. 5. $Tree_{v,T}$.

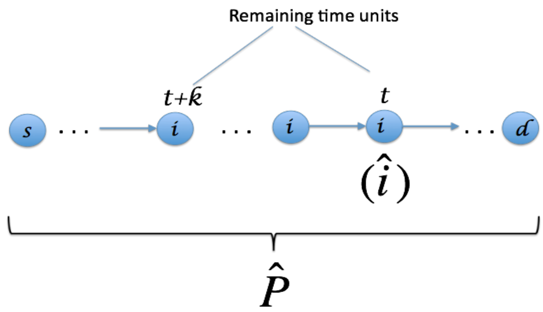
For this proof, we use the following notion of a *Tree of Paths*.

Definition 1. *Tree of Paths: Given a destination d, for every node v and all time limits $T > 0$, we can construct a binary tree of paths, $Tree_{v,T}$ that has its root labeled as v and all leaves labeled as d (see Fig. 5). $Tree_{v,T}$ is built as follows. Suppose node a is the optimal node to visit from the root v when v is congested and b is the optimal node to visit from v when v is uncongested. Then in $Tree_{v,T}$, the children of v will be a and b. Similarly, if x and y are the optimal nodes to visit from a when a is congested and uncongested, respectively, then the children of a will be x and y. If some node i has a child who is also i, this means the optimal decision at the parent i is to wait at i for one time unit. Every path in $Tree_{v,T}$ is the optimal path to visit from v, given a sequence of node states and using at most T time units.*

Consider $Tree_{s,T}$ for some source s, destination d, and time limit T. There are two ways that a path from s to d on $Tree_{s,T}$ may contain a repeating node i (see Fig. 6). We now describe these two types



(a) Path $\bar{P}$ revisits i after k time units.



(b) Path $\hat{P}$ waits at i for k time units.

Fig. 6. Two paths with repeating node i.

of paths. Type 1: For some $k > 0$, i first appears when there are $t + k$ time units remaining, reappears when there are t time units remaining, and each node between the first and second appearance is not i. Paths of Type 1 indicate that the optimal solution requires visiting i when there will be $t + k$ time units remaining and then *revisiting* i when there will be t time units remaining (see Fig. 6(a)). Type 2: For some $k > 0$, i appears in a consecutive sequence for k time units. Paths of Type 2 indicate that the optimal solution requires *waiting* at i for k time units (see Fig. 6(b)). We will show that for networks that exhibit Properties 1 and 2, only paths of Type 2 exist in $Tree_{s,T}$. In other words, a repeating node, i, in an optimal path indicates the decision to wait at i, rather than revisit i.

Theorem 1. *For any network that exhibits Properties 1 and 2, the on-time arrival probability achieved from a path that requires waiting at a node i will be greater than the on-time arrival probability achieved from a path that requires revisiting i.*

Proof. Consider a path P from $Tree_{s,T}$. P represents the optimal path from s to the destination given T time units and a sequence of node states. There are four possible cases where a path P may contain a repeating node i: Case 1) i is uncongested in the first appearance and uncongested in the last appearance; Case 2) i is uncongested in the first appearance and congested in the last appearance; Case 3) i is congested in the first appearance and uncongested in the last appearance; and Case 4) i is congested in the first appearance and congested in the last appearance. We will first show that if P is an optimal path then Cases 1) and 2) cannot occur. We will then show that Cases 3) and 4) can occur only if P is a path of Type 2. In other words, the repeating i must occur from the decision to wait at i rather than revisit i.

Let $t + k$ denote the number of time units remaining after the first occurrence of i and let t denote the number of time units remaining after the last occurrence of i (see Fig. 7).

Case 1: i is uncongested when there are $t + k$ time units remaining and uncongested when there are t time units remaining. From Proposition 2, we know $P_u^{t+k}(i) \geq P_u^t(i)$. Since visiting i the first time (when there will be $t + k$ time units remaining) yields a higher probability than visiting i the last time (when there will be t time units remaining), there is no need to revisit i, therefore Case 1 cannot occur.

Case 2: i is uncongested when there are $t + k$ time units remaining and congested when there are t time units remaining. By Property 1 we have:

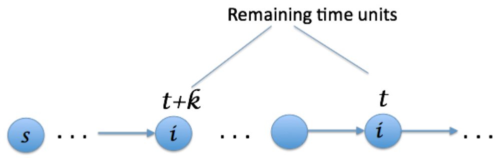


Fig. 7. A path with repeating node i.

$$P_u^{t+k}(i) > P_c^{t+k}(i) \tag{13}$$

From Proposition 2 we have:

$$P_c^{t+k}(i) > P_c^{t}(i) \tag{14}$$

Therefore $P_u^{t+k}(i) > P_c^t(i)$. As in Case 1, since visiting i the first time yields a higher probability than visiting i the last time, there is no need to revisit i, therefore Case 2 cannot occur.

Cases 3 and 4: i is congested when there are $t+k$ time units remaining. The following is true if i is in either state (congested or uncongested) when there are t time units remaining. We will prove by contradiction that both cases can occur only with paths of Type 2.

Consider path $\bar{P}$ in Fig. 6(a) and path $\hat{P}$ in Fig. 6(b). Both paths start at source s and end at destination d. Assume that $\bar{P}$ and (to the contrary) $\hat{P}$ are the optimal paths given a time limit and sequence of node states. Suppose that $\bar{P}$ and $\hat{P}$ are identical up to the first appearance of node i. In path $\bar{P}$, we first visit i when there will be $t+k$ time units remaining, we then visit a sequence of nodes (none of which are i), and finally, revisit i when there will be t time units remaining. In other words, path $\bar{P}$ contains a cycle starting and ending at i (see Fig. 6(a)). In path $\hat{P}$, we wait at i for k time units, starting when there will be $t+k$ time units remaining and ending when there will be t time units remaining (see Fig. 6(b)). Let $\bar{i_t}$ denote the last occurrence of i on path $\bar{P}$ and let $\hat{i_t}$ denote the last occurrence of i on path $\hat{P}$. We will show that the on-time arrival probability from $\hat{i_t}$ will always be more than that of $\bar{i_t}$. Since replacing the subpath from i to $\bar{i_t}$ in $\bar{P}$ with the subpath from i to $\hat{i_t}$ in $\hat{P}$ will always yield a higher on-time arrival probability, $\bar{P}$ cannot be optimal. In other words, waiting at a node will always yield a higher on-time arrival probability than revisiting a node.

We first consider $\bar{P}$ (i.e. paths of Type 1) and the on-time probability achieved from $\bar{i}$. Recall that we reach $\bar{i}$ after revisiting i after k time units. Let a denote the node immediately before $\bar{i_t}$ in $\bar{P}$ (see Fig. 6(a)).

Sub-Case 1: a is congested. We first consider the case where a is congested. Node i will be congested with probability $P_c(i|a)$ and uncongested with probability $1 - P_c(i|a)$. If i is congested, then with probability $P_c^t(i)$, we will reach d on time; if node i is uncongested, then with probability $P_u^t(i)$ we will reach d on time. Therefore, we can express the on-time arrival probability from $\bar{i_t}$ given that the previously visited node was a in the congested state as follows:

$$P_c(i|a)P_c^t(i) + (1 - P_c(i|a))P_u^t(i) \tag{15}$$

We now consider $\hat{P}$ (i.e. paths of Type 2) and the on-time probability achieved from $\hat{i}$. Recall that we reach $\hat{i}$ after waiting at i for k time units. Node i will be congested with probability $P_c^*(i,k)$ and uncongested with probability $1 - P_c^*(i,k)$. If there are t time units remaining and i is congested, then with probability $P_c^t(i)$, we will reach d on time; if node i is uncongested, then with probability $P_u^t(i)$ we will reach d on time. Note that since $\hat{i}$ represents node i during the *last* time unit of waiting at i, the node following $\hat{i}$ cannot be i. Therefore the arrival probability from $\hat{i}$ is:

$$P_c^*(i,k)P_c^t(i) + (1 - P_c^*(i,k))P_u^t(i) \tag{16}$$

We will prove by contradiction that for networks exhibiting Properties 1 and 2, (15) always yields a lower probability than (waiteq).

For networks exhibiting Properties 1 and 2, assume the contrary:

$$\begin{gathered}
P_c(i|a)P_c^t(i) + (1 - P_c(i|a))P_u^t(i) > P_c^*(i,k)P_c^t(i) + (1 - P_c^*(i,k))P_u^t(i) \\
P_c(i|a)P_c^t(i) + P_u^t(i) - P_c(i|a)P_u^t(i) > P_c^*(i,k)P_c^t(i) + P_u^t(i) - P_c^*(i,k)P_u^t(i) \\
P_c(i|a)[P_c^t(i) - P_u^t(i)] > P_c^*(i,k)[P_c^t(i) - P_u^t(i)]
\end{gathered}$$

By Property 1 we know $P_c^t(i) < P_u^t(i)$, so we have:

$$P_c(i|a) < P_c^*(i,k) \tag{17}$$

which violates Property 2 (i.e. $1 - P_c(i|a) < 1 - P_c^*(i,k)$), which contradicts our assumption. Therefore, if a is congested, we achieve a higher arrival probability from waiting at i for k time units than by revisiting i after a.

<u>*Sub-Case 2*</u>*: a is uncongested* We now consider the case where a is uncongested. Then the arrival probability from $\bar{i_t}$ is:

$$(1 - P_u(i|a))P_c^t(i) + P_u(i|a)P_u^t(i) \tag{18}$$

Again, the arrival probability achieved from waiting at i for k time units is:

$$P_c^*(i,k)P_c^t(i) + (1 - P_c^*(i,k))P_u^t(i) \tag{19}$$

Again, we will prove by contradiction that for networks exhibiting Properties 1 and 2, (18) always yields a lower probability than that of (19).

For networks exhibiting Properties 1 and 2, assume the contrary:

$$\begin{gathered}
P_c^*(i,k)P_c^t(i) + (1 - P_c^*(i,k))P_u^t(i) < (1 - P_u(i|a))P_c^t(i) + P_u(i|a)P_u^t(i) \\
P_c^*(i,k)P_c^t(i) + P_u^t(i) - P_c^*(i,k)P_u^t(i) < P_c^t(i) - P_u(i|a)P_c^t(i) + P_u(i|a)P_u^t(i) \\
P_c^*(i,k)[P_c^t(i) - P_u^t(i)] + P_u^t(i) < P_u(i|a)[P_u^t(i) - P_c^t(i)] + P_c^t(i) \\
P_c^*(i,k)[P_c^t(i) - P_u^t(i)] + P_u^t(i) - P_c^t(i) < P_u(i|a)[P_u^t(i) - P_c^t(i)] \\
1 - P_c^*(i,k) < P_u(i|a)
\end{gathered}$$

which violates Property 2, which contradicts our assumption. Therefore, if a is uncongested, we achieve a higher arrival probability by waiting at i for k time units than by revisiting i k time units after a.

We have shown that regardless of the state of a, waiting at node i will yield a higher on-time probability than revisiting i.

We have shown that Cases 3 and 4 are the only situations where an optimal path may contain a reappearing node. We have also shown that in both of these cases, for networks that exhibit Properties 1 and 2, only paths of Type 2 may occur (i.e. paths where the reappearing node occurs from the decision to wait at the node) and not paths of Type 1 (i.e. paths where the reappearing node occurs from the decision to revisit the node). This affirms our claim that for these networks, the on-time arrival probability achieved from waiting at a node will be greater than the on-time arrival probability achieved from revisiting a node.

5 Conclusion

We study the problem of finding the path that maximizes the probability of arriving to a destination from a source within a specified time limit. We present an efficient dynamic programming solution that solves this adaptive routing problem. We prove that as the time limit increases, the maximum probability also increases. We also show that for some networks, revisiting a node is required to achieve the optimal probability. Since this is contrary to most realistic settings, we consider a variation of the problem where the traveler is always given the option to wait at a node rather than visit another node. We identify two important properties of networks in this setting: (1) The on-time arrival probability from any node is always higher if the node is uncongested rather than congested; and (2) a node is more likely to become uncongested after waiting at the node for any amount of time than from visiting it from an adjacent node. For networks with these properties, we prove that the optimal solution will not require revisiting nodes, which is reflective of most realistic settings.

References

1. B.C. Dean. Algorithms for Minimum-Cost Path in Time-Dependent Networks with Waiting Policies. *Networks*, vol. 44, pp. 41-46, 2004.
2. A. Eiger, P. B. Mirchandani , and H. Soroush. Path Preferences and Optimal Paths in Probabilistic Networks. *Transportation Science*, vol. 19, pp. 75-84, 1985.
3. Y. Fan, R. Kalaba, and J. Moore. Shortest Paths in Stochastic Networks with Correlated Link Costs. *Computers and Mathematics with Applications*, vol. 49, pp. 1549-1564, 2005.
4. Y. Fan, R. Kalaba, and J. Moore. Arriving on Time. *Journal of Optimization Theory and Applications*, vol. 127, no. 3, pp. 497-513, Dec. 2005.
5. H. Frank. Shortest Paths in Probabilistic Graphs. *Operations Research*, vol. 17, pp. 583-599, 1969.
6. L. Fu and L. Rillet. Expected Shortest Paths in Dynamic and Stochastic Traffic Networks. *Transportation Research*, vol. 32, no. 7, pp. 499-516, Sep. 1998.
7. S. Gao and H. Huang. Real-Time Traveler Information for Optimal Adaptive Routing in Stochastic Time-Dependent Networks. *Transportation Research Part C*, vol. 21, pp. 196-213, 2011.
8. R. P. Loui. Optimal Paths in Graphs with Stochastic or Multidimensional Weights. *Communications of the ACM*, vol. 26, pp. 670-676, 1983.
9. I. Murthy and S. Sarkar. Stochastic Shortest Path Problems with Piecewise Linear Concave Utility Functions, *Management Science*, vol. 44, pp.125-136, 1998.
10. Y. Nie and X. Wu. Shortest Path Problem Considering On-Time Arrival Probability. *Transportation Research Part B*, vol. 43, pp. 597-613, 2009.
11. G. H. Polychronopoulos and J. N. Tsitsiklis. Stochastic Shortest Path Problems with Recourse. *Operations Research Letters*, vol. 10, pp. 329- 334, August 1991.
12. E. H. Sigal, A.A.B. Pritsker, and J.J. Solberg. The Stochastic Shortest Route Problem. *Operations Research*, vol. 28, pp. 1122-1129, 1980.
13. S. Waller and A. Ziliaskopoulos. On the Online Shortest Path Problem with Limited Arc Cost Dependencies. *Networks*, vol, 40, no. 4, pp. 216 - 227, Oct. 2002.