Netw Spat Econ
DOI 10.1007/s11067-009-9108-9

Optimal Information Location for Adaptive Routing

Stephen D. Boyles · S. Travis Waller

Abstract One strategy for addressing uncertain roadway conditions and travel times is to provide real-time travel information to drivers through variable message signs, highway advisory radio, or other means. However, providing such information is often costly, and decisions must be made about the most useful places to inform drivers about local conditions. This paper addresses this question, building on adaptive routing algorithms describing optimal traveler behavior in stochastic networks with *en route* information. Three specific problem contexts are formulated: routing of a single vehicle, assignment of multiple vehicles in an uncongested network, and adaptive equilibrium with congestion. A network contraction procedure is described which makes an enumerative algorithm computationally feasible for small-to-medium sized roadway networks, along with heuristics which can be applied for large-scale networks. These algorithms are demonstrated on three networks of varying size.

Keywords Intelligent transportation systems · Adaptive routing · Advanced traveler information systems · User equilibrium

1 Introduction

Transportation systems are inherently uncertain. Events such as incidents, poor weather, variations in travel demand, and the chaotic nature of congested

S. D. Boyles (✉)
Dept. 3295, University of Wyoming, 1000 E University Ave, Laramie, WY 82071, USA
e-mail: sboyles@gmail.com

S. T. Waller
The University of Texas at Austin, 1 University Station C1761, Austin, TX 78712, USA

Published online: 10 September 2009

vehicle flow, make it impossible for drivers to predict the travel time on any given route with certainty. A significant amount of travel delay, if not the majority, can be attributed to nonrecurring causes (Lindley 1987), and both researchers and practitioners have long recognized the importance that this uncertainty plays in describing traveler behavior. One common strategy for mitigating uncertainty is information provision through advanced traveler information systems (ATIS), such as variable message signs (VMSs), highway advisory radio (HAR), or many other technologies. These devices often provide information to drivers *en route*, so while drivers may anticipate receiving information at certain locations, they cannot anticipate the specific message they will receive. Thus, adaptive routing algorithms are needed to describe how drivers respond to this type of information.

Within this context, public agencies must make decisions about where to locate devices such as VMSs or HARs. Installing these devices is costly, and a limited budget is available—for instance, an agency may only have sufficient funds for placing three VMS signs in a certain city, and must decide how to locate them to maximize the benefit to drivers.

Alternately, the information location problems can also be used to provide adaptive driving directions for individuals. Many services are available which provide a route connecting a given origin and a given destination; however, in congested regions, the expected travel time can be reduced by providing several alternatives which can be used depending on observed traffic conditions. Current online shortest path algorithms can provide some insight on this problem, but their practical application is limited to real-time devices (such as in-vehicle navigation systems) because these typically assume a re-routing decision can be made at *every* node, and there is no easy way to convey this to drivers through printable directions or other format given *a priori*. On the other hand, by restricting re-routing decisions to a small number of nodes, one can simply report several complete paths to drivers, which is far more easily understood—the problem becomes one of deciding where to allow this re-routing, which is identical to the VMS location problem faced by a public agency. In this case, it may not be necessary to assume an external information provision device, but base online decisions on qualitative observations made by the driver: "If the freeway is congested, exit onto this arterial."

As mentioned above, considerable research has been performed on adaptive routing algorithms. Andreatta and Romeo (1988) considered a routing problem where arcs may fail, in which case a traveler may need to follow a predetermined "recourse path" to the destination. For acyclic networks, Psaraftis and Tsitsiklis (1993) describe an algorithm which can determine an optimal routing policy. When cycles are permitted, Miller-Hooks (2001) presents a polynomial-time algorithm for independent arc costs, while Waller and Ziliaskopoulos (2002) and Provan (2003) present pseudopolynomial algorithms for several arc dependence scenarios. The case of general dependency is more difficult, and is NP-complete, unless one takes the "reset assumption," in which arc costs can vary on successive visits; Polychronopoulos and Tsitsiklis

(1996) and Provan (2003) develop algorithms for this case. Pretolani (2000) and Miller-Hooks (2001) also address adaptive routing when costs are time-dependent as well as stochastic. Gao (2005) presents an algorithm when arc costs are time-dependent, stochastic, and have general dependency. These algorithms typically assume users wish to minimize their expected travel time; other behaviors have also been considered, such as minimizing schedule delay (Gao and Chabini 2006) or maximizing the probability of on-time arrival (Nie and Fan 2006). Any of these algorithms can be used to help determine the best locations to provide information, depending on the assumptions one makes about the underlying network.

Several researchers have also conducted studies regarding optimal locations for providing information. Abbas and McCoy (1999) applied a genetic algorithm to place VMSs at locations that maximize the number of vehicles which observe these signs, but did not consider adaptive behavior in response to this information. Chiu et al. (2001) and Chiu and Huynh (2007) combine a mesoscopic dynamic traffic assignment simulation with a tabu search heuristic to optimally locate VMSs. Incidents were randomly generated using a Monte Carlo scheme, and some drivers would switch routes if their path encounters an incident and a VMS sign; based on the resulting flow patterns, a set of VMS locations was determined to optimize some measure of effectiveness. Huynh et al. (2003) uses a similar analysis framework to find the optimal locations of portable VMSs in a real-time framework, using the G-D heuristic. Although the simulation approach allows a rich set of traffic and behavioral impacts to be modeled, the computational burden associated with many simulation runs on a large network can be troublesome.

This limitation was realized by Henderson (2004), who adopted a static equilibrium framework for VMS location, together with a discrete choice model to determine the proportion of drivers who switch routes in response to learning of an incident. Several heuristic techniques are developed and compared, including a genetic algorithm and a greedy approach based on sequential location. While computationally faster, this approach implicitly assumes that drivers do not anticipate receiving information; that is, their initial route choice is not affected by the VMS locations, so links with a VMS do not "attract" drivers who anticipate benefitting from that information, for instance. Although this distinction may seem subtle, this anticipation effect can lead to radically different route choices for rational drivers, even from the origin (Boyles 2006).

The research presented here complements these works by providing analytical network algorithms for locating information, where users both anticipate receiving information and adjust their routes adaptively. The remainder of this paper is organized as follows. Section 2 describes the problem context formally, along with rigorous definitions of three information location problems addressed in the paper. Section 3 describes a network contraction procedure which allows candidate solutions to be evaluated extremely rapidly. Section 4 describes exact algorithms and heuristics for solving these three problems, which are then demonstrated in Section 5. Finally, Section 6 concludes the paper by summarizing the key contributions and pointing to future research directions.

2 Problem definitions

Consider a directed graph $G = (N, A, Z, \mathbf{D})$ where $n = |N|$ and $m = |A|$ denote the number of nodes and arcs, respectively, $Z \subset N$ represents the zones where trips begin and end, and $\mathbf{D} \in \mathbb{R}_+^{|Z| \times |Z|}$ is a matrix whose elements d_{uv} represent travel demand between zones u and v. Let $\Gamma(i)$ denote the set of nodes adjacent to node i, and $\Gamma^{-1}(i)$ the set of nodes to which i is adjacent. For each arc $a \in A$, let $\delta(a)$ represent its downstream node, that is, if $a = (i, j)$, then $\delta(a) = j$. Each arc (i, j) can exist in one of a discrete set of states $s_{ij} \in S_{ij}$ representing, for instance, normal operating conditions, a mild incident, and a severe incident. Each state s_{ij} occurs with probability p_{ij}^s, and is associated with a cost function $c_{ij}^s(x_{ij}^s)$ mapping the demand for travel x_{ij}^s on this arc to the corresponding cost.

The state of an arc is independently determined at each traversal (that is, we take the "reset assumption")—for example, if a driver's route traverses the same arc more than once, a different state may be experienced on both visits; and different drivers traversing the same arc may each experience different travel times. This assumption is motivated by the observation that the events causing this uncertainty (such as incidents) typically do not last for the entire study period. For instance, if an incident blocks a freeway lane for half an hour out of a three-hour study period, only a sixth of drivers will see the incident, assuming uniform arrivals.[1] Thus, the probabilities p_{ij}^s should be interpreted as the chance that any given driver traveling on arc (i, j) sees state s, not the probability that the arc is in state s for the entire study period.

Drivers receive travel information at a set of *information nodes* $R \subset N$ by learning the current states of the adjacent arcs. Note that information is only given for adjacent arcs: we assume that either no information is provided on more distant arcs (for instance, due to space limitations on a VMS), or that any information is outdated by the time a driver would reach that arc. Mathematically, a driver arriving at a node $i \in R$ receives a message $\theta \in \Theta_i = \times_{(i,j) \in A} S_{ij}$; with the independence assumption, the probability of receiving message θ is $\rho_i^\theta = \prod_{(i,j) \in A} p_{ij}^{s(\theta)}$ where $s(\theta)$ is the state conveyed by message θ. At non-information nodes, drivers have no information; symbolically, we write this as receiving a message $\theta = \emptyset$ with probability 1.

For instance, consider the network shown in Fig. 1, where the arc labels represent costs. Arcs (A, B), (C, E), and (D, E) have deterministic cost, while the costs of (A, E), (B, C) and (B, D) take on one of two values with equal

[1]This assumption is also valid under Poisson arrivals, in terms of the expected number of drivers seeing a given state.

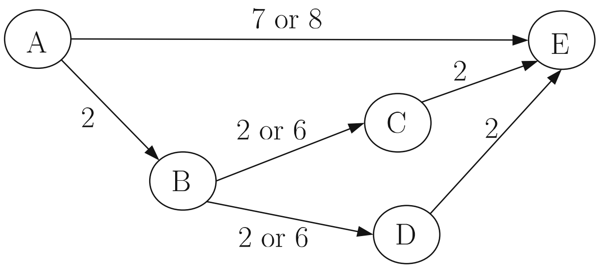


Fig. 1 Example network to demonstrate notation and concepts

probability. Assume that there is one traveler departing node A and destined for node E, and that the costs are fixed and independent of this traveler's decision. If node A is an information node, the traveler learns the cost on (A, E) and (A, B), so the potential messages are $(7, 2)$ and $(8, 2)$ and $\Theta_A = \{(7, 2), (8, 2)\}$. On the other hand, if A was not an information node, $\Theta_A = \{\emptyset\}$, and the driver must choose a path without knowing the exact costs on these arcs. Likewise, if node B were an information node, the messages would indicate the costs on (B, C) and (B, D), with $\Theta_B = \{(2, 2), (2, 6), (6, 2), (6, 6)\}$.

A flexible way of describing adaptive routing is to use routing *policies*. A *node-state* is defined as a pair (i, θ) representing arrival at node i and receiving message θ, with $\Phi = \{(i, \theta) : i \in N, \theta \in \Theta_i\}$ the set of all node-states. A routing policy can be formally described as a function $\pi : \Phi \to A$ indicating the arc chosen by a driver arriving at node i and learning information θ, where $\delta(\pi(i, \theta)) \in \Gamma(i)$ for all $(i, \theta) \in \Phi$. With a slight abuse of notation, let $E[\pi]$ be the expected cost experienced by a driver who follows policy π.

Note that policies only describe the next arc taken in the path; upon arrival at the downstream end of that arc, the traveler will experience another node state (possibly corresponding to no information $\emptyset$), choose the next arc according to the policy, and so on. At first glance, this approach might appear limited or myopic. In fact, the opposite is true, and a policy is a more general way to specify user behavior than a path. If a policy prescribes the same choice of outgoing arc regardless of the information received, the traveler will follow a classical path, indicating that the set of paths is a subset of the set of policies. Drivers are assumed to follow the routing policy that minimizes their *expected* travel cost, a straightforward analogue of the assumption that drivers follow shortest paths in deterministic networks. Least expected-cost policies can be determined by applying the TD-OSP algorithm of Waller and Ziliaskopoulos (2002) when the cost functions $c_{ij}(\cdot)$ are constant. When they are flow-dependent, an optimal assignment of users to least expected-cost policies can be accomplished using the UER2 algorithm, based on Unnikrishnan (2008). Additional details on these algorithms can be found in the Appendix.

Again referring to the example in Fig. 1, consider the case where $R = \{B\}$, that is, B is the only information node. The set of node-states is shown in Table 1, along with the optimal routing policy. Since node A is not an information node, the driver will always choose to travel to node B, at which

Table 1 Node states and optimal policy for the example network with $R = \{B\}$

Node state	Chosen arc
$(A, \emptyset)$	(A, B)
$(B, (2, 2))$	(B, C)
$(B, (2, 6))$	(B, C)
$(B, (6, 2))$	(B, D)
$(B, (6, 6))$	(B, C)
$(C, \emptyset)$	(C, E)
$(D, \emptyset)$	(D, E)

point the costs of (B, C) and (B, D) will be revealed. If either of these arcs is in the "low" state (cost 2), it will be chosen by the driver, who will then continue on to node E and experience a total trip cost of 6 units. The only way an arc will be traversed in the "high"state (cost 6) is if both (B, C) and (B, D) have high cost, which occurs with probability 1/4 and results in a total trip cost of 10 units. Thus, the expected cost of this policy is $6 \times 3/4 + 10 \times 1/4 = 7$ units.

Note that the driver exhibits anticipatory behavior: the only reason for traveling to node B is because information will be revealed at that point. Without information and adaptive routing, the least expected-cost path is simply to follow arc (A, E) directly to the destination, with expected cost 7.5; this demonstrates that the driver's route choice at the origin can be affected by information provided at a later time.

Table 2 shows the set of node states and optimal policy if A was the only information node, rather than B. In this case, the optimal strategy is to always choose arc (A, E), with an expected cost of 7.5. Therefore, in this example, it is better to provide information at node B rather than node A, because the resulting optimal policy has lower expected cost.

Let the cost of providing information at node i be given by C_i, and assume that a given budget B is available for this purpose. These costs can either be monetary (as with a public agency seeking to install VMS signs) or abstract (as with driving directions, where one can use unit cost for C_i and set B to the maximum number of information nodes). Within these assumptions, we consider three different information location problems. In each case, the goal is to find a set of information nodes $R^* \in \overline{R}$ optimizing a particular objective, where $\overline{R}$ represents the set of feasible information node sets (that is, the information node sets whose cost does not exceed the available budget).

Individual Information Provision (IIP) In this problem, we are only concerned with optimizing a single traveler's expected travel time, so only one element of $\mathbf{D}$ is nonzero, and the cost functions c_{ij}^s are constant, because an atomic individual's travel decision will not affect the costs they experience.

Table 2 Node states and optimal policy for the example network with $R = \{A\}$

Node state	Chosen arc
$(A, (7, 2))$	(A, E)
$(A, (8, 2))$	(A, E)
$(B, \emptyset)$	(B, C)
$(C, \emptyset)$	(C, E)
$(D, \emptyset)$	(D, E)

Table 3 Overview of problems IIP, UIP, and CIP

	IIP	UIP	CIP
OD pairs	One	Many	Many
Link costs	Constant	Constant	Flow-dependent
Objective function	$E[\pi^*]$	$TSTT$	$TSTT$
Key algorithm	TD-OSP	TD-OSP	UER

This problem is appropriate for providing adaptive driving directions for an individual with a private service.

Uncongested Information Provision (UIP) In this problem, we are concerned with minimizing the total system travel time of a large number of travelers, where congestion effects are ignored:

$$TSTT = \sum_{(i,j)\in A} \sum_{s\in S_{ij}} x_{ij}^s c_{ij}^s$$

That is, **D** may take on general values, but the cost functions c_{ij}^s are still constant. This problem is appropriate for representing information provision on large networks with minimal congestion, such as freight routes in rural areas where weather closures may require re-routing.

Congested Information Provision (CIP) In this case, we are again concerned with minimizing the total system travel time, but here congestion effects must be considered, so the costs c_{ij}^s will depend on the flows x_{ij}^s. This is appropriate for representing urban areas where incidents cause significant reliability issues.

Clearly, IIP is a special case of UIP, and both of these are special cases of CIP. Table 3 briefly summarizes the differences between these problems.

Finally, as a practical note, it is well-known that not all drivers will switch routes in response to information received *en route*. For the purposes of this paper, such users can be ignored as long as the number of such drivers is known, by incorporating their presence into the cost functions as "background" traffic. Behavioral models where switching occurs only under certain circumstances (trip purpose, degree of time savings, freeway vs. arterial) are not considered in the present work.

3 Network contraction

It is not trivial to evaluate a given set of information nodes R. The most straightforward approach is to apply an online routing or equilibrium algorithm to the network with information nodes R. For IIP, this consists of a single application of TD-OSP to determine the expected travel cost from the origin to the destination with R the information nodes. For UIP, because TD-OSP calculates an "all-to-one" optimal policy tree, one can calculate the total system travel time by applying TD-OSP n times, once for each possible destination, multiplying the expected travel cost from each origin by the travel demand, and summing over all origins and destinations. For CIP, the UER algorithm must be run to convergence.

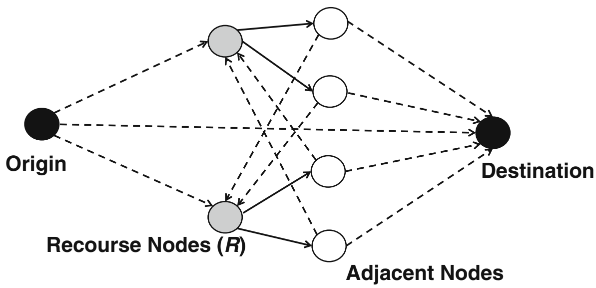
This direct approach is undesirable for two reasons. First, applying these algorithms requires some computation time, and any conceivable solution algorithm requires evaluation of a large number of potential information node sets. Second, the computation time required for each of these algorithms grows with network size: TD-OSP requires $O(n^2 m S \log(nS))$ time, where $S = \max_{ij} |S_{ij}|$, and UER2, which involves repeated solution of TD-OSP, exhibits comparable growth in run time.

The good news is that a faster approach for evaluating information nodes is available for IIP and UIP, allowing TD-OSP to be applied to a much smaller network. For simplicity, we first describe this procedure for IIP, then show how it is adapted for UIP.

Because drivers can only make a recourse decision at an information node, their routes they travel are deterministic except at such nodes, simply because they do not receive any information which would cause them to switch paths. Furthermore, at information nodes, drivers only learn information about adjacent arcs. Upon arriving at the downstream end of these arcs, they will continue to follow a deterministic path until encountering another information node or the destination.

This can be represented by constructing a contracted network $G'(R) = (N'(R), A'(R))$, where the contracted node set $N'(R)$ consists of the origin, the destination, the information nodes R, and the nodes adjacent to information nodes, and where the contracted arc set $A'(R)$ connects the origin to each information node and the destination, each information node to its adjacent nodes, and every adjacent node to each information node and the destination. Figure 2 shows a sample contracted network for two information nodes (marked in grey). In this figure, solid lines represent arcs which also exist in the original network G ("direct arcs"), while dashed lines represent a deterministic path connecting its tail and head nodes in G ("path arcs"). The only direct arcs are those connecting recourse nodes to their adjacent nodes; all of the other contracted arcs represent paths in G. We denote the set of direct and path arcs as $A'_D(R)$ and $A'_P(R)$, respectively. Note that $N'(R)$ contains at most $2 + |R| + \sum_{i \in R} |\Gamma(i)|$ nodes and $A'(R)$ contains at most $(1 + \sum_{i \in R} |\Gamma(i)|)(|R| + 1) + \sum_{i \in R} |\Gamma(i)|$ arcs.

Fig. 2 Example contracted network with $|R| = 2$



To demonstrate this concept using a larger network, Fig. 3 shows how a contracted network is created on the well-known Sioux Falls network. The black nodes denote the origin and the destination, while the grey nodes indicate the information nodes. Of the two costs shown in the original network, the lower cost occurs with probability 0.9, while the higher cost occurs with probability 0.1.

Note that the only arcs in the contracted graph with uncertain costs are those adjacent to information nodes, since these are the only locations where an adaptive decision can be made. The remaining nodes are connected by arcs with deterministic cost, representing the cost of the least expected-cost path between these. (The justification for choosing these costs is given in Theorem 1.) Although this network is only slightly smaller than the original network, the contracted network would be nearly the same size regardless of the number of nodes and arcs in the original graph, assuming the node connectivity is comparable.

In particular, by choosing the path arcs to represent least expected-cost paths between their tail and head nodes, and by setting the arc's cost to the expected cost of this path, the optimal policy π' on the contracted graph has the same expected cost as the optimal policy π^* on the original graph, as shown below.

Theorem 1 $E[\pi'] = E[\pi^*]$

Proof We first show that $E[\pi'] \leq E[\pi^*]$. Consider the following procedure CONTRACT, applied to a node i where $\Theta_i = \{\emptyset\}$: eliminate i from the graph, along

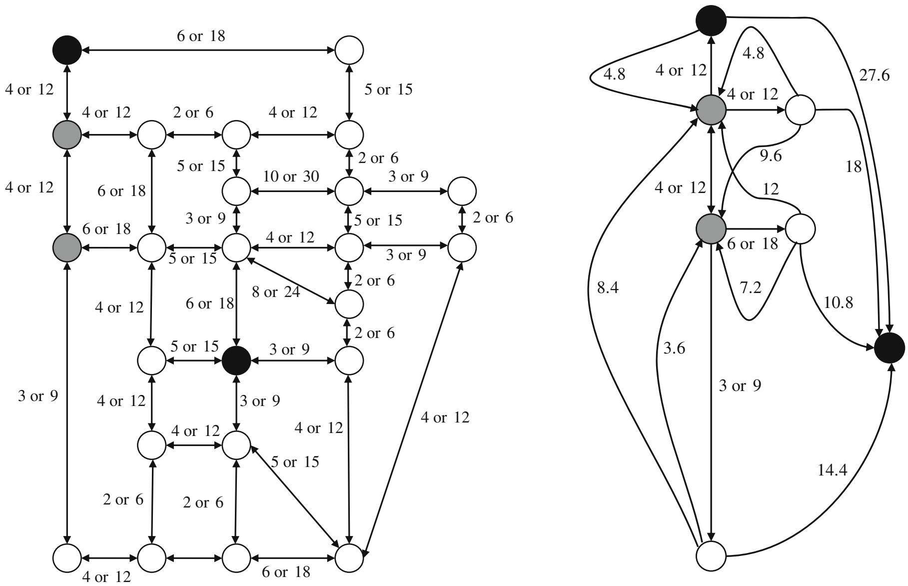


Fig. 3 Sioux Falls network and a contracted graph for two information nodes

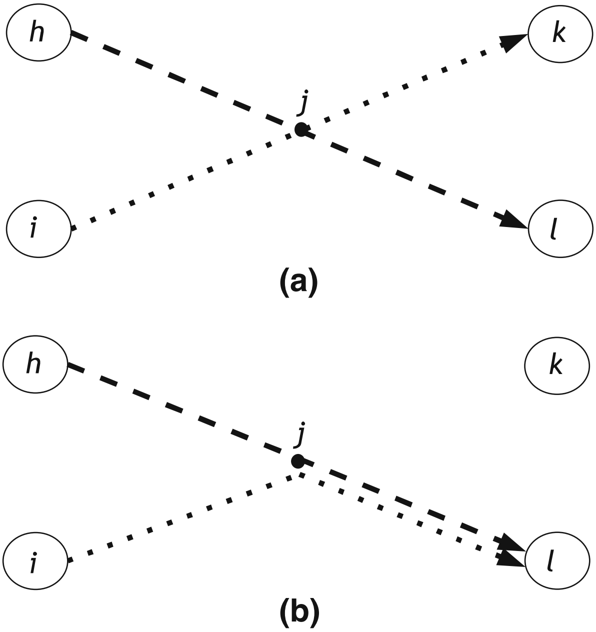


Fig. 4 Potential conflict with expanding policies on the contracted graph, and resolution procedure (**a**, **b**)

with all arcs adjacent to i. For each arc incident to i, replace that arc's head node with $\pi^*(i, \emptyset)$, and add the expected cost of arc $(i, \pi^*(i, \emptyset))$ to the cost of each of its states. Note that the cost of π^* is unaffected by this procedure (in fact, the policy itself is essentially unaffected, aside from the trivial removal of node-state $(i, \emptyset)$). Returning to graph G, iteratively apply CONTRACT, each time choosing a node i which is neither an information node, nor immediately adjacent to an information node. Each step does not affect the cost of the optimal policy, and the resulting graph is a subgraph of G' (since clearly the deterministic components of π^* must represent least expected-cost paths), implying $E[\pi'] \leq E[\pi^*]$.

Similarly, we can show that $E[\pi^*] \leq E[\pi']$, which is enough to prove the result. Since the arcs A'_P represent least expected-cost paths in G, a policy in G with equal expected cost can be trivially constructed by expanding the policy π' using these paths, unless there exists a node $j \in N$ which is part of two such shortest paths to different nodes k and l (see Fig. 4(a)). Thus, assume that such a node exists.[2] Let L_k and L_l be labels representing the expected travel cost from k to the destination v; since arc costs are independent, these labels do not depend on the path taken to reach these nodes. Since π' is optimal, $L_k \leq L_l$, because otherwise the path segment $j - k$ could be replaced by $j - l$. By the same argument, $L_l \leq L_k$ and thus $L_l = L_k$. Thus, when constructing a

[2]Essentially, at non-information nodes, a traveler following a policy in G must make the same decision regardless of their past travel history, while a traveler following a path arc has an additional piece of information—the tail and head nodes of that path. We must show that this additional information cannot improve the expected cost of the optimal policy.

policy in G from π', altering one of the expanded paths from a path arc to be consistent with the expanded path from another (Fig. 4(b)) does not change the expected cost of the policy. □

The contracted graph is extremely useful for solving IIP and UIP because it allows the value of a set of information nodes to be evaluated by applying TD-OSP to a much smaller graph. In particular, note that the size of the contracted graph does not depend on the size of the original graph. Since $|R| \ll n$ in most cases, this leads to an enormous reduction in the time needed for evaluation. (Of course, the number of feasible sets of information nodes still grows with network size.)

To evaluate a set of information nodes for UIP, one might imagine that a contracted graph should be constructed for each OD pair in Z^2. However, a more efficient approach is possible. Because TD-OSP is an "all-to-one" label correcting algorithm, it suffices to construct a single contracted graph for each destination, provided that every origin node is included as well; that is, taking the union of all of the contracted graphs corresponding to a single destination. The contracted graphs formed in this manner will contain at most $1 + |Z| + |R| + \sum_{i \in R} |\Gamma(i)|$ nodes and $(|Z| + \sum_{i \in R} |\Gamma(i)|)(|R| + 1) + \sum_{i \in R} |\Gamma(i)|$ arcs.

One might object that performing TD-OSP $|Z|$ times to the slightly larger destination-based networks is worse than $|Z|^2$ applications on the smaller single origin-destination networks, because TD-OSP grows faster than linearly in network size. However, since the origin nodes have no reverse star, their addition involves very little increase in the run time, certainly much less than the worst-case bound. A better comparison is the number of node labels which must be calculated; with the given graph sizes, using the destination-based networks requires the calculation of roughly $(|Z|^2 - |Z|)(1 + \sum_{i \in R} |\Gamma(i)|)$ fewer labels than the use of the single origin-destination networks, a savings which is substantial in large networks where many network contractions need to be performed.

One might also wonder why the nodes adjacent to information nodes are retained in the contracted networks, because no information is received there and a deterministic decision is made. By including these nodes, the cost of the path arcs are obtained from a simple lookup from an all-pairs shortest path computation performed upon initialization. If the adjacent nodes are contracted, the path arcs can exist in multiple states, and costs must be calculated for each state. This introduces additional computational requirements which negate the savings from a slightly smaller network, and complicates the implementation.

Unfortunately, this contraction procedure is not useful for CIP, because the link costs are flow-dependent, implying that multiple paths will be used by each OD pair in general, and thus generating the appropriate cost function for the path arcs is difficult. Furthermore, the cost of a path arc can depend on the flow on a separate arc, and equilibrium models with adaptive routing are not yet available for networks with asymmetric cost functions.

4 Solution methods

All three of the information location problems described above are difficult to solve exactly, as IIP, UIP, and CIP are essentially a facility location or network design problem, where the solution cost is determined by the travel cost experienced by the driver(s). Such problems are notoriously difficult to solve due to their nonlinearity and discrete nature, and enumerative techniques are often required to find the exact optimal solution. It is not difficult to show that IIP is NP-hard: consider the 0-1 knapsack problem $\max_{\mathbf{x}} \mathbf{v} \cdot \mathbf{x}$ among n_K objects such that $\mathbf{w} \cdot \mathbf{x} \leq 1$ and $\mathbf{x} \in \{0, 1\}^{n_K}$. Construct a graph $G_K = (N_K, A_K)$ with $N_K = \{1, 2, \ldots, n_K, n_K + 1\} \cup \{1', 2', \ldots, n'_K\}$ and $A_K = \{(1, 2), (2, 3), \ldots, (n_K, n_K + 1)\} \cup \{(1, 1'), (2, 2'), \ldots, (n_K, n'_K)\} \cup \{(1', 2), (2', 3), \ldots, (n'_K, n_K + 1)\}$ (see Fig. 5). Each arc $(i, i + 1) \in \{(1, 2), (2, 3), \ldots, (n_K, n_K + 1)\}$ exists in one of two states with equal probability; these states have cost $-2v_i$ and ∞, respectively. All other arcs have cost zero deterministically, and define $C_i = w_i$ for each node, along with $B = 1$. Consider solving IIP on G_K: if a node $i \in N_K$ is an information node, the optimal policy is clearly to follow $(i, i + 1)$ if that arc has cost $-2v_i$, and to follow (i, i') otherwise. For non-information nodes j, the optimal policy is to always follow (j, j'), and the expected cost of any such policy is the negative of the knapsack objective when the objects corresponding to information nodes are selected. As this knapsack problem is well-known to be NP-hard, IIP must be NP-hard as well. Furthermore, as IIP is a special case of UIP and CIP, the NP-hardness of these problems follows immediately.

Thus no efficient, exact solution algorithms can be provided for these problems at present. Still, one way to determine the optimal set R^* is to simply calculate the total travel time resulting from each set in $\overline{R}$ being chosen as information nodes, and identifying the best such set. This is clearly inefficient, but network contraction makes enumeration computationally feasible for solving IIP or UIP on small- to medium-sized networks. That is, the contracted graph corresponding to each feasible set of information nodes is constructed, TD-OSP applied for each destination,

If $|R| \leq R_{max}$ for all feasible information sets R, then $O(n^{R_{max}})$ sets must be examined. Although this growth is polynomial in network size (assuming fixed R_{max}), a large planning network (such as those used to model Chicago, IL or Philadelphia, PA) can easily include over 10,000 nodes, and locating even three information nodes via enumeration would require more than a trillion

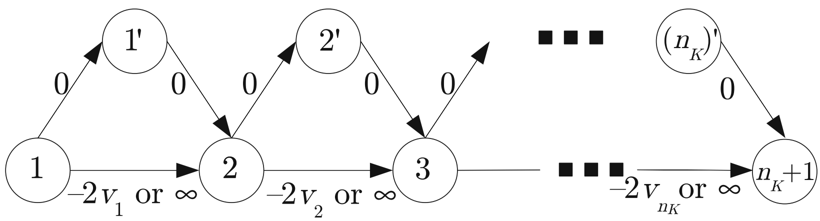


Fig. 5 Reduction from the 0-1 knapsack problem

iterations of network contraction and TD-OSP. Thus, it is still necessary to develop heuristic solution procedures.

Many heuristics employ the notion of a neighborhood to specify which feasible solutions are considered "adjacent" in a search procedure. In this paper, the neighborhood $\mathcal{N}(R)$ of a set of information nodes R is defined as the set of feasible information node sets which differ from R by exactly one node. Returning to the example in Fig. 3, where $R = \{5, 9\}$, $\mathcal{N}(R)$ is the union of the sets, $\{5\}$, $\{9\}$, $\{(5, i) : i \in N - \{5, 9\}\}$, $\{(i, 9) : i \in N - \{5, 9\}\}$, and $\{(5, 9, i) : i \in N - \{5, 9\}\}$, intersected with $\overline{R}$. In general this set is of size $O(nR_{max})$.

This suggests a local search heuristic, where one starts with an initial feasible set of information nodes, and considers each neighboring set. If any of them has a lesser cost, the one with minimal cost is chosen as the new incumbent solution, and the search repeated with the new neighborhood. If none has a lower cost, the current incumbent is declared a local optimum and the search halted. An initial feasible solution must be generated in some way; three approaches considered in this paper are:

1. Labels representing the expected cost from each origin to each destination are calculated for the "full-information" and "no information" cases (that is, where adaptive routing is allowed at each node, and where drivers must choose their route *a priori* using expected costs). The difference between these is defined as the *benefit* of information at node for that origin-destination pair; the total benefit is calculated by multiplying the benefit to each OD pair by its demand value, and summing. Proceeding in a greedy manner, construct the initial set R by repeatedly adding the nodes with the highest benefit-cost ratio, until doing so is no longer feasible.
2. Instead of choosing all of the nodes with highest benefit-cost ratio at the same time, proceed iteratively: after selecting the node i with the highest total benefit, re-calculate the "no information" labels by allowing information at node i along with the updated benefits, select the node with the highest total benefit which can feasibly be added to the initial set, and so on.
3. A purely random selection of nodes can be made for the initial solution.

Local search with this neighborhood definition is not guaranteed to find the optimal solution to IIP, as shown by the network in Fig. 6. The only nodes where information can provide any benefit to travelers are nodes 2, 3, 5, and 6. By inspection, the optimal set is $R = \{2, 3\}$, with an expected travel cost of 48. However, if the incumbent set is $R = \{5, 6\}$ (as would occur as the initial set under the first two decision rules), none of the neighboring information sets ($\{2, 5\}$,$\{2, 6\}$,$\{3, 5\}$,$\{3, 6\}$) reduce the expected cost below its current value of 51. Thus, with this neighborhood definition, there can exist sets which are locally optimal, but not globally so.

Finally, one can apply a purely greedy approach: consider all feasible information sets of size one, and select the set R_1 providing the greatest reduction in the objective function per unit of cost, relative to the case where

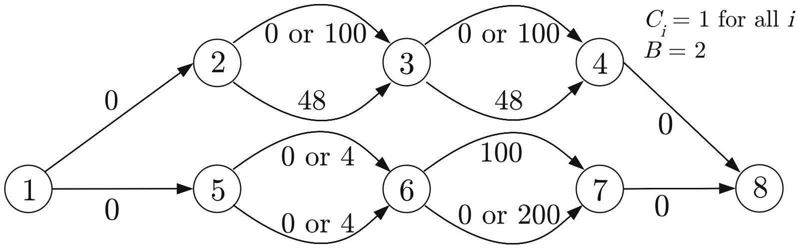


Fig. 6 Network demonstrating how local search and the greedy heuristics can fail

$R = \emptyset$. Next, consider all nodes which can be feasibly be added to R_1, and choose the set R_2 with providing the greatest objective function per unit of cost, relative to R_1. This procedure is repeated until no additional information nodes can feasibly be added. Thus, each iteration involves examining $O(n)$ new solutions. As with local search, this procedure need not produce the optimal solution, even when all nodes have equal cost, as seen by the network in Fig. 6. If information can only be provided at one node, the best location is node 6, reducing the optimal expected travel time from 96 to 52. Given that information is provided at node 6, the best node to choose second is node 5, reducing travel cost to 51; however, the optimal set of size two is {2, 3}, with expected cost 48.

From a practical standpoint, significant gains in computation time can often be obtained by judicious choice of the feasible sets $\overline{R}$, as influenced through the node costs C_i. For instance, there is no benefit to providing information at a node with only one exiting arc, because drivers at this node must choose the same arc regardless of any information received, and because any such information is only valid locally. Such nodes commonly exist where freeway onramps merge, and at certain intersections involving one-way streets or turn restrictions. In the Chicago Regional network (described more fully in the following section), roughly five percent of the nodes can be excluded by this criterion. Since the number of feasible sets can grow exponentially with respect to the network size this saving can be significant: if $\overline{R} = N^3$, for instance, a time savings of nearly fifteen percent can be seen in an enumerative search. Separately, one may also be able to restrict attention *a priori* to a small subset of nodes, such as those adjoining freeway links and major arterials, leading to an even greater reduction in the size of the feasible set.

One should also note that all of these methods are highly parallelizable, which will decrease computation times substantially if available.

5 Demonstration

These algorithms were tested on four standard transportation networks of varying sizes, obtained from Bar-Gera (2009). Table 4 shows properties of these networks, as well as the problems which will be studied for each network

Table 4 Characteristics of the test networks

	Sioux Falls	Anaheim	Barcelona	Chicago regional
n	24	416	1020	12,982
m	76	914	2522	39,018
z	24	38	110	1790
SP time (s)	0.00	1.14	17.3	34500
Problems	IIP,UIP,CIP	IIP,UIP,CIP	IIP,UIP	IIP

and the time required to find the shortest paths between each pair of nodes using the Floyd-Warshall algorithm. While IIP can be studied on all four networks, memory and time considerations preclude analyzing UIP on the Chicago Regional network or analyzing CIP on the Barcelona or Chicago Regional networks.

Each of the solution methods described in the previous section is implemented and tested. Local search is applied using each of the three rules for generating an initial information set; when the initial configuration is random, the search is repeated five times and the best solution chosen. Additionally, for comparison with a standard metaheuristic, simulated annealing is used to generate an information set, using the same neighborhood definition as the local search. The cooling schedule and other parameters are determined separately for each test network, adapting the procedure in Chiang and Russell (1996) to ensure that the initial probability of accepting a disimproving move is five percent, and that the number of iterations between cooling is equal to half of the neighborhood size. Computation times are reported for a 3.4 GHz Pentium 4 machine using Windows XP with 2 GB RAM, and all algorithms are terminated after one hour of running time.

For each test case, the cost of providing information at each node is one cost unit, and the cases $B = 2$ and $B = 3$ are considered. That is, two feasible sets $\overline{R}$ are considered: R^2 (all sets of two information nodes) and R^3 (all sets of three information nodes). For IIP and UIP, arc costs are assumed to equal the free-flow travel time with probability 0.9, and three times the free-flow travel time with probability 0.1; for CIP, the free-flow travel times vary in the same manner, with the capacity constant; the well-known BPR relation is used to relate link flows to travel times, with shape parameters $\alpha = 0.15$ and $\beta = 4$. Travel demand for UIP and CIP is the same as the standard network files; for IIP, the origin and destination are the two nodes farthest apart, in terms of shortest free-flow travel time.

Results from solving IIP on the four networks are shown in Table 5, showing the sets of information nodes found by the algorithms, the computation time needed to find these (in seconds), and the amount of benefits provided by information, relative to the benefits attainable by providing information everywhere. (That is, the difference between the expected travel cost with that information and the "no-information" expected travel cost, divided by the difference between the "full-information" and "no-information" expected travel costs.) The time required for finding the shortest path between each pair of nodes is reported in Table 4 and is not included in the computation

Table 5 Individual information provision (IIP) on test networks

	$\|R\| = 2$			$\|R\| = 3$		
	Nodes	Time (s)	Benefit	Nodes	Time (s)	Benefit
(a) Sioux Falls						
Enumeration	3,12	0.02	56.5%	3,11,12	0.32	72.2%
Local search 1	3,12	0.01	56.5%	3,11,12	0.02	72.2%
Local search 2	3,12	0.01	56.5%	3,11,12	0.02	72.2%
Local search 3	1,12	0.05	50.5%	1,11,12	0.10	69.3%
Greedy	3,12	0.00	56.5%	1,11,12	0.00	69.3%
Simulated annealing	3,12	0.01	56.5%	3,11,12	0.32	72.2%
(b) Anaheim						
Enumeration	404,405	4.41	42.7%	305,404,405	957.93	56.0%
Local search 1	404,405	0.35	42.7%	305,404,405	0.69	56.0%
Local search 2	404,405	0.52	42.7%	305,404,405	0.84	56.0%
Local search 3	180,404	0.28	22.9%	136,371,404	0.83	22.9%
Greedy	404,405	0.04	42.7%	305,404,405	0.21	56.0%
Simulated annealing	404,405	0.20	42.7%	201,404,405	0.25	42.7%
(c) Barcelona						
Enumeration	249,1009	116	47.8%	1,351,783	3600[a]	47.8%
Local search 1	249,1009	2.95	47.8%	249,306,1009	5.22	57.8%
Local search 2	249,1009	5.23	47.8%	249,306,1009	9.62	57.8%
Local search 3	550,1009	12.8	33.6%	909,921,1009	37.6	33.6%
Greedy	249,1009	0.27	47.8%	249,306,1009	0.57	57.8%
Simulated annealing	963,1009	1.97	41.5%	249,826,1009	12.3	51.2%
(d) Chicago regional						
Enumeration	2184,9883	3600[b]	18.0%	1,2184,9883	3600[c]	18.0%
Local search 1	9446,9447	29.5	19.9%	2755,9476,12299	35.0	0.00%
Local search 2	9446,9447	59.0	19.9%	2755,9476,12299	94.4	0.00%
Local search 3	7051,9883	15.1	14.9%	1896,3358,9883	60.5	14.9%
Greedy	6826,9883	3.04	19.9%	6826,8625,9883	6.57	23.5%
Simulated annealing	6822,9883	31.5	18.5%	2184,8625,9883	34.8	21.6%

[a] 1.56% of feasible space explored in time limit

[b] 66.6% of feasible space explored in time limit

[c] 0.01% of feasible space explored in time limit

times recorded here, in order to more clearly differentiate the impact of the algorithms which have a common initialization.

Several results are apparent. First and most notably, the greedy heuristic always found the best known solution, in substantially less time; this suggests that pitfalls such as those in Fig. 6 may be relatively rare in transportation networks, and that the sets of information nodes tend to "nest" in that optimal sets of one size are subsets of optimal sets of a larger size. This may be reasonable, in that the best places to locate information in large networks are geographically disparate, and providing information at one node may only have a limited impact on the benefits of providing information at another, distant node. On the other hand, he frequent failure of Local Search 3 (initialized randomly) to find the optimal information node sets, even with five restarts, suggests that local search quite often leads to non-globally optimal solutions if not initialized carefully.

Interestingly, the first two rules for determining the initial candidate set for a local search always produced identical sets of information nodes, and found the global optimum solutions, although rule one requires less computation time. This occurs because rule one only requires one application of TD-OSP on the whole network, while rule two requires one application per information node; while the benefits of iteratively updating cost labels are not apparent in these networks. Unsurprisingly, enumeration quickly grows intractable; at the observed pace for the first hour of computation, identifying the optimal set of three information nodes on the Chicago Regional network would require more than a year.

Similar results are seen when solving UIP on the three smallest networks (Table 6). Note the substantial increase in computation time, since optimal policies must be found for each destination in the network, not just one. The comments which applied to IIP are mainly applicable here as well. Although the first two decision rules for initializing the local search seem to produce different results for locating three information nodes in the Barcelona network, this is an artifact introduced by the one-hour time limit and the greater time needed to initialize rule two. Given more time to proceed, Local Search 2 would have followed the same search trajectory as Local Search 1 in this network.

Table 7 shows the results from solving CIP on the Anaheim and Sioux Falls networks. Interestingly, for the Anaheim network, using a random seed for the local search yielded a better two-information node solution than was

Table 6 Uncongested information provision (UIP) on test networks

	$\|R\| = 2$			$\|R\| = 3$		
	Nodes	Time (s)	Benefit	Nodes	Time (s)	Benefit
(a) Sioux Falls						
Enumeration	10,16	1.41	29.0%	10,15,16	18.57	38.3%
Local search 1	10,16	0.87	29.0%	10,15,16	3.31	38.3%
Local search 2	10,16	0.97	29.0%	10,15,16	3.60	38.3%
Local search 3	10,16	2.27	29.0%	10,15,16	5.71	38.3%
Greedy	10,16	0.20	29.0%	6,10,14	0.48	26.8%
Simulated annealing	10,16	0.75	29.0%	10,15,16	3.66	38.3%
(b) Anaheim						
Enumeration	91,232	648.3	18.8%	2,91,232	3600[a]	25.6%
Local search 1	91,232	20.8	18.8%	91,232,236	57.8	25.6%
Local search 2	91,232	24.0	18.8%	91,232,236	65.7	25.6%
Local search 3	227,232	55.9	12.5%	91,95,232	129.5	19.2%
Greedy	91,232	4.4	18.8%	91,232,236	8.1	25.6%
Simulated annealing	91,232	9.2	18.8%	91,232,236	27.6	25.6%
(c) Barcelona						
Enumeration	1,766	3600[b]	5.06%	1,8,766	3600[c]	5.14%
Local search 1	555,766	1733	11.7%	555,673,766	3600	20.6%
Local search 2	555,766	2425	11.7%	72,766,887	3600	11.8%
Local search 3	306,766	3000	6.32%	210,366,762	3600	8.86%
Greedy	555,766	303	11.7%	555,673,766	623	20.6%
Simulated annealing	682,762	888	9.24%	555,676,816	1610	15.3%

[a] 3.97% of feasible space explored in time limit

[b] 3.90% of feasible space explored in time limit

[c] 0.01% of feasible space explored in time limit

Table 7 Congested information provision (CIP) on test networks

	$\|R\| = 2$			$\|R\| = 3$		
	Nodes	Time (s)	Benefit	Nodes	Time (s)	Benefit
(a) Sioux Falls						
Enumeration	10,15	40.4	35.4%	10,11,15	310	45.1%
Local search 1	10,15	19.0	35.4%	10,11,15	55.8	45.1%
Local search 2	10,15	13.0	35.4%	10,11,15	31.6	45.1%
Local search 3	10,15	64.3	35.4%	10,15,24	105	43.3%
Greedy	10,15	6.0	35.4%	10,11,15	9.5	45.1%
Simulated annealing	10,15	23.8	35.4%	10,11,15	47.0	45.1%
(b) Anaheim						
Enumeration	21,319	3600[a]	11.0%	1,21,319	3600[b]	11.0%
Local search 1	319,355	463	13.6%	319,355,407	1030	17.0%
Local search 2	319,355	467	13.6%	319,355,407	1033	17.0%
Local search 3	388,389	1331	15.8%	86,302,319	2403	11.0%
Greedy	319,355	147	13.6%	319,355,407	236	17.0%
Simulated annealing	319,355	169	13.6%	268,319,355	554	11.0%

[a]20.9% of feasible space explored in time limit

[b]0.15% of feasible space explored in time limit

found by any of the other heuristics, the only time that this heuristic found a better solution than the others. Comparing UIP and CIP, one sees the benefits attainable from only two or three information nodes are higher in the Sioux Falls network when congestion effects are present, but lower in the Anaheim network. This may be due to differences in the congestion level on these networks: the average volume-to-capacity ratios for the no-information equilibrium assignment in these networks are 1.48 for Sioux Falls, and 0.32 for Anaheim. Again note the significant increase in computation time needed to solve this problem, as evaluating any feasible solution involves an equilibration, and no network contraction is available to speed the process.

In all cases, note that a sizable portion of the total possible benefits from information provision can be achieved even when only providing information at two or three nodes.

6 Conclusion

This paper addressed the problem of choosing the optimal locations to provide real-time traffic information, in three different forms: routing of an individual vehicle, routing of multiple vehicles in an uncongested system, and multiple-vehicle equilibrium in a congested network. As even the simplest of these problems is NP-hard, heuristics were developed to solve each of these problems. For the two simplest cases, a network contraction procedure allows rapid evaluation of candidate solutions. These heuristics were then tested in networks of varying sizes, showing that a substantial portion of the benefits of information are available even when providing information only at two or three nodes.

Although providing initial insight into these problems, there are several valuable and interesting extensions of this work which can enhance the realism of these models. Generalizing the information beyond a single step is possible within this framework, although the complexity of the algorithms presented here grows substantially, and a more efficient way to handle general information would be welcome. Likewise, relaxing independence of arc states could enhance the realism of our model. Perhaps the most significant assumption made here is the static nature of congestion; although a dynamic version of these problems would seem to require greatly more computation time, such a model could model reliability and congestion with much greater fidelity.

Acknowledgement The authors would like to thank Nezamuddin for useful comments made on an earlier draft of this paper, and for several fruitful discussions regarding heuristic approaches.

Appendix

Algorithms and Pseudocode

This appendix provides a brief overview of the TD-OSP and UER2 algorithms used in this paper. TD-OSP is taken directly from Waller and Ziliaskopoulos (2002), and determines the expected cost of an optimal adaptive routing policy through a label-correcting technique involving a scan-eligible list SEL. The presentation of the algorithm below is adapted to the notation in this paper,

Algorithm 1 TD-OSP($\mathbf{C}$)

{Argument: matrix $\mathbf{C}$ of (fixed) arc costs in each state}
$L_v \leftarrow 0$
$L_i \leftarrow \infty \qquad \forall i \in N - v$
$\pi(i,\theta) \leftarrow \emptyset \qquad \forall i \in N, \theta \in \Theta_i$
$SEL \leftarrow \Gamma^{-1}(v)$
while $SEL \neq \emptyset$ **do**
    Remove a node i from SEL
    $tempL \leftarrow 0$
    for all $\theta \in \Theta_i$ **do** {Note that $|\Theta_i| > 1$ only if $i \in R$}
        $tempL \leftarrow tempL + \rho_i^\theta \min_{(i,j)\in A}\{c_{ij}^{s(\theta)} + L(j)\}$
        $\pi(i,\theta) \leftarrow \arg\min_{(i,j)\in A}\{c_{ij}^{s(\theta)} + L(j)\}$
    end for
    if $tempL < L_i$ **then**
        $L_i \leftarrow tempL$
        $SEL \leftarrow SEL \cup \Gamma^{-1}(i)$
    end if
end while
return π

and to allow both information nodes and non-information nodes. Input to TD-OSP is a destination v, and the output is the optimal policy π^* with respect to the (fixed) arc costs by state, provided as input.

The version shown here is used for CIP, when the entire routing policy is needed. For UIP and IIP, all that is needed is the expected cost of the optimal routing policy, a variation for which Waller and Ziliaskopoulos (2002) provides a pseudopolynomial algorithm involving an inner reduction step of the expected cost vector.

The UER2 algorithm is used to evaluate feasible solutions for the CIP problem, finding an equilibrium among *policies* rather than fixed paths as in the traditional deterministic user equilibrium problem. The problem setting in this paper matches Model B of Unnikrishnan and Waller (2009), who shows that the equilibrium state-dependent link flow matrix $\mathbf{X} = [x_{ij}^s]$ solves the optimization problem

$$\min \sum_{(i,j)\in A} \sum_{s \in S_{ij}} \int_0^{x_{ij}^s} c_{ij}^s(x)dx$$

where the x_{ij}^s are generated by a feasible policy assignment with respect to the demand matrix $\mathbf{D}$. Unnikrishnan (2008) uses an incidence matrix to map each policy to the proportion of its flow which uses arc (i, j) in state s, and then applies the Frank-Wolfe algorithm to solve this program (Algorithm 2).

In this paper, we adopt a different approach for mapping policies to arc flows, applying a "policy loading" algorithm (Algorithm 3) which assigns flow from all origins to a given destination v in each stage. Each node i is associated with a label T representing the number of vehicles at node i which have yet to reach the destination. Initially, T_i is equal to the demand from i to v. Nodes

Algorithm 2 UER2

```
X_0 ← 0
for all z ∈ Z do
    π*_z ← TD-OSP(c^s_ij(0), z)
    X_0 ← X_0 + LoadPolicy(π*_z, z, D_z)
end for
Calculate relative gap γ from flows X_0
while γ < γ_0 do {Convergence criterion}
    X ← 0
    for all z ∈ Z do
        π*_z ← TD-OSP(c^s_ij(0), z)
        X ← X + LoadPolicy(π*_z, z, D_z)
    end for
    α ← arg min_{α∈[0,1]} Σ_{(i,j)∈A} Σ_{s∈S_ij} ∫_0^{α(x_0)^s_ij + (1−α)x^s_ij} c^s_ij(x)dx
    X_0 ← αX_0 + (1 − α)X
    Calculate relative gap γ from flows X_0
end while
```

Algorithm 3 LOADPOLICY(π, v, D)

```
{Arguments: policy π, destination v, demand vector D for v)}
X ← 0
SEL ← ∅ {Binary max-heap used to identify nodes to scan}
for all i ∈ N do
    if i ∈ Z and D_i > 0 then
        T_i ← D_i
        SEL ← SEL ∪ i
    else
        T_i ← 0
    end if
end for
while SEL ≠ ∅ do
    Remove a node i from SEL with maximum L_i
    if T_i > T_min then {Number of remaining vehicles is sufficiently large}
        for all θ ∈ Θ_i do
            x_π(i,θ) ← x_π(i,θ) + ρ_i^θ T_i
            if δ(π(i,θ)) ≠ v then
                T_δ(π(i,θ)) ← T_δ(π(i,θ)) + ρ_i^θ T_i
                SEL ← SEL ∪ π(i,θ)
            end if
        end for
    else {Terminate cycling by shifting flow to adjacent node with lowest L}
        j_min ← arg min_{j∈Γ(i)} L_j
        for all s ∈ S(i, j_min) do
            x_(i,j_min),s ← x_(i,j_min),s + p^s_{ij_m in} T_i
        end for
        T_j_min ← T_j_min + T_i
        SEL ← SEL ∪ j_min
    end if
    T_i ← 0
end while
return X
```

with positive T_i are scanned, and these vehicles assigned to adjacent nodes according to the routing policy and the probability of each message being received, thus increasing T for the adjacent nodes, and reducing T_i to zero. Since vehicles typically move from nodes with higher expected cost labels L to nodes with lower expected cost labels, a binary heap is used to keep track of the nodes with positive T, identifying the highest-cost node at each iteration. This alteration we name UER2.

The prime advantages of UER2 are (i) the ability to handle networks with cycles, through an iterative approach and a tolerance $T_{min} \ll ||\mathbf{D}||$ to terminate long cycles; and (ii) a substantial (approximately $O(m)$) reduction in the computation time needed, since all origins corresponding to the same destination are processed simultaneously and redundant flow shifts are eliminated. The implications of this change, and proofs of convergence and correctness, are discussed more fully in Boyles (2009) and Boyles and Waller (2009).

References

Abbas M, McCoy P (1999) Optimizing variable message sign locations on freeways using genetic algorithms. In: Presented at the 78th annual meeting of the transportation research board. Washington, DC

Andreatta G, Romeo L (1988) Stochastic shortest paths with recourse. Networks 18:193–204

Bar-Gera H (2009) Transportation test problems. Website: http://www.bgu.ac.il/~bargera/tntp/. Accessed 28 April 2009

Boyles SD (2006) Reliable routing with recourse in stochastic, time-dependent transportation networks. Master's thesis, The University of Texas at Austin

Boyles SD (2009) Operational, supply-side uncertainty in transportation networks: causes, effects, and mitigation strategies. Ph.D. thesis, The University of Texas at Austin

Boyles SD, Waller ST (2009) Online routing and equilibrium with nonlinear objective functions. Working paper

Chiang W-C, Russell RA (1996) Simulated annealing metaheuristisc for the vehicle routing problem with time windows. Ann Oper Res 63:3–27

Chiu Y-C, Huynh N (2007) Location configuration design for dynamic message signs under stochastic incident scenarios. Transp Res Part C 15(1):333–50

Chiu Y-C, Huynh N, Mahmassani H (2001) Determining optimal locations for VMS's under stochastic incident scenarios. In: Presented at the 80th annual meeting of the transportation research board. Washington, DC

Gao S (2005) Optimal adaptive routing and traffic assignment in stochastic time-dependent networks. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA

Gao S, Chabini I (2006) Optimal routing policy problems in stochastic time-dependent networks. Transp Res Part B 40(2):93–122

Henderson JM (2004) A planning model for optimizing locations of changeable message signs. Master's thesis, University of Waterloo

Huynh N, Chiu Y-C, Mahmassani HS (2003) Finding near-optimal locations for variable message signs for real-time network traffic management. Transp Res Rec 1856:34–53

Lindley J (1987) Urban freeway congestion: quantification of the problem and effectiveness of potential solutions. ITE Journal 57:27–32

Miller-Hooks ED (2001) Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. Networks 37(1):35–52

Nie Y, Fan Y (2006) Arriving-on-time problem: discrete algorithm that ensures convergence. Transp Res Rec 1964:193–200

Polychronopoulos GH, Tsitsiklis JN (1996) Stochastic shortest path problems with recourse. Networks 27(2):133–143

Pretolani D (2000) A directed hypergraph model for random time dependent shortest paths. Eur J Oper Res 123:315–324

Provan JS (2003) A polynomial-time algorithm to find shortest paths with recourse. Networks 41(2):115–125

Psaraftis HN, Tsitsiklis JN (1993) Dynamic shortest paths in acyclic networks with Markovian arc costs. Oper Res 41(1):91–101

Unnikrishnan A (2008) Equilibrium models accounting for uncertainty and information provision in transportation networks. Ph.D. thesis, The University of Texas at Austin

Unnikrishnan A, Waller ST (2009) User equilibrium with recourse. Networks and Spatial Economics. Accepted for publication

Waller ST, Ziliaskopoulos AK (2002) On the online shortest path problem with limited arc cost dependencies. Networks 40(4):216–227