Robust path choice in networks with failures

Michael C. Ferris

Computer Sciences Department, University of Wisconsin, Madison, Wisconsin 53706 [1]

Andrzej Ruszczyński

Department of Industrial Engineering, University of Wisconsin, Madison, Wisconsin 53706 [2]

Abstract

The problem of adaptive routing in a network with failures is considered. The network may be in one of finitely many states characterized by different travel times along the arcs, and transitions between the states occur according to a continuous-time Markov chain. The objective is to develop a routing strategy that minimizes the total expected travel time. Dynamic programming models and flow-oriented models are developed and analyzed in the uncapacitated and the capacitated case. It is shown that the robust plan can be found from a special two-stage stochastic programming problem in which the second stage models the re-routing problem after the state transition in the network. The models are illustrated on an example of the Sioux Falls transportation network. The computational results reveal striking properties of different routing policies and show that substantial improvements in both duration and size of jams can be achieved by employing robust strategies.

1 Introduction

The notion of a traffic jam is familiar to almost everyone who has driven a car. Reducing the frequency and size of such jams is a key issue that traffic planners and automatic guidance systems have to deal with almost continually [15,20]. One way to reduce jams is to increase the capacity or the number of the arcs in the transportation network. This is typically very expensive to perform and in some extreme cases can be shown to have a detrimental effect [7,14].

[1] The research of this author was supported in part by the National Science Foundation under grant CCR-9157632 and the Air Force Office of Scientific Research under grant F49620-98-1-0417.

[2] Present address: Rutgers University, Department of Management Science and Information Systems, Piscataway, NJ 08854.

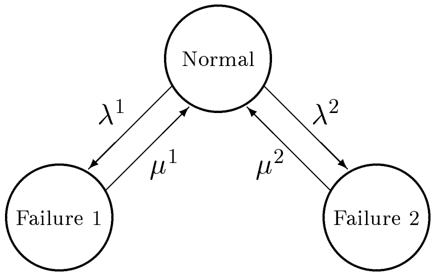

Fig. 1.1. Transition diagram of network's states.

Our approach is somewhat complementary. It has become apparent that although there is much research being carried out on providing timely and accurate information to drivers [1,4,21], there is much work still to be performed regarding the generation and use of complex dynamical models to improve the congestion levels in large urban areas. We are going to develop a new modeling paradigm, namely robust guidance strategies. We shall introduce a number of models for networks subject to failures, discuss robust guidance strategies and illustrate the results on an example of a real-world network. Since the net effect of the robust strategies that we propose here is to reduce the size of jams when failures occur, the modeling format proposed herein can also be used for more effective design of fault tolerant communications networks. In this case, the network is typically designed to have enough residual capacity to safely reroute flow when a failure occurs by generating at least two paths through the network for each demand pair. If robust strategies are used for path choice, we believe that significantly smaller capacities need to be provided since flows can be more effectively rerouted whenever a failure occurs.

Our assumptions are rather abstract and the models that we develop apply to other types of networks, most notably—telecommunication networks. In fact, we believe that there is much potential in applying some our ideas in telecommunication. We use the transportation terminology mainly for illustrative purposes and to guide intuition.

We consider a network with node set $\mathcal{N}$ and arc set $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$, and random non-negative arc costs (travel times) $\tilde{c}_{ij}$ and capacities $\tilde{u}_{ij}$, $(i, j) \in \mathcal{A}$. The travel times and capacities are determined by the *state* of the network $\ell \in \{0, \ldots, L\}$: for each ℓ the travel times are equal to c^{ℓ}_{ij} and capacities to u^{ℓ}_{ij}. The states evolve in time at random, according to a continuous-time Markov chain, and the current state is known to the vehicles in the network.

We shall at first focus our attention on the simple case when the only possible transitions are between state 0 (representing the *normal* operation mode) and states $\ell \neq 0$ (representing *failure* modes). The rate of transition from 0 to $\ell \neq 0$ will be denoted by λ^{ℓ}, and the transition rate back by μ^{ℓ}, see Figure 1.1. It will become clear in the sequel how more general transition diagrams can be treated.

The problem is as follows. At each node $n \in \mathcal{N}$ there is a constant demand flow s_n that must be moved through the network to some destination node D at the minimum expected travel time. Related problem formulations have been studied by many authors. In [5,6,16] *stochastic shortest path* problems are analyzed; they correspond to our setting when no arc capacity constraints are present. In [3] related *two-stage* problems are considered in the case when some arc lengths may become infinite. Problems with capacity constraints are discussed in [12] in the case of a dynamic (layered) network with one source (see also [17] and references therein).

To facilitate the analysis and to provide ground for more general cases we make the following simplifying assumptions.

(A1) If the state of the system changes from k to ℓ when a vehicle is on arc (i,j) the travel time on (i,j) remains equal to c_{ij}^k for this vehicle; it experiences new travel times only after hitting j.
(A2) The products $\lambda^\ell c_{ij}^0$ and $\mu^\ell c_{ij}^\ell$ are much smaller than one for each ℓ and each $(i,j) \in \mathcal{A}$.

Condition (A1) amounts to assuming that failures occur at the initial sections of the arcs and do not affect those who have passed them. It is equally simple to consider other cases, except the notation then becomes more involved.

Condition (A2) implies that the failures are rare and the probability of more than one state transition during the travel time of a vehicle on an arc (i,j) is negligibly small. In many real–world transportation networks this assumption can be legitimately criticised, but in the area of telecommunication networks it is more likely to be satisfied. Nevertheless, we shall see that in the presence of capacity constraints even rare failures lead to new and rather involved models.

In section 2 we consider the uncapacitated version of the problem and show a simple dynamic programming solution. Sections 3 and 4 develop a flow-oriented model in the two-stage case, where only one state transition in a vehicle's journey is allowed; sections 5 and 6 illustrate how this model can be extended to the multistage case. In sections 7–9 we analyze the capacitated problem, in which the main issue is the interaction between vehicles that started at different times but reach a node at the same time, thus leading to jams. We analyze models that guide vehicles in a way that minimizes the cost of jams and facilitate the clearing of jams in a user-specified time frame. Finally, in section 10 we consider a real-world network of Sioux Falls to illustrate the models and solutions.

2 Single vehicle approach and dynamic programming

Let us at first consider the uncapacitated problem: $u_{ij}^\ell = +\infty$ for $(i,j) \in \mathcal{A}$ and $\ell = 0,\dots,L$. Owing to the Markov property of state transitions and to the linearity of the costs, the problem can be solved by dynamic programming methods as the *stochastic* *shortest path problem* [5]. With each node $i \in \mathcal{N}$ and each state $\ell \in \{0, 1, \ldots, L\}$ we associate the *cost-to-go* v_i^ℓ: the least expected travel time from i to D when the initial network state is ℓ. They satisfy the following Bellman's equation:

$$v_i^\ell = \min_{(i,j)\in\mathcal{A}} \mathbb{E}\{c_{ij}^\ell + v_j^{\ell'}\}, \quad i \in \mathcal{N} \setminus \{D\}, \quad \ell = 0, \ldots, L, \tag{2.1}$$

where ℓ' is the state of the network when the vehicle reaches node j. We set $v_D^\ell = 0$ for all ℓ.

Assume that the initial state of the network is 0. If a vehicle enters arc (i, j), the state of the network may change during its travel time c_{ij}^0. Owing to (A2), the probability of more than one state transition in time c_{ij}^0 is negligible, and the travel time itself remains unchanged, by virtue of (A1). Thus the probabilities $p_{0\ell}$ of transition from 0 to ℓ in this time are given by

$$p_{00} \approx \Big(1 - \sum_{\ell=1}^{L} \lambda^\ell\Big) c_{ij}^0, \quad p_{0\ell} \approx \lambda^\ell c_{ij}^0, \quad \ell = 1, \ldots, L.$$

If (A1) or (A2) is not satisfied, we can still calculate these transition probabilities by methods of Markov process theory. [1]

Equation (2.1) for $\ell = 0$ reads

$$v_i^0 = \min_{(i,j)\in\mathcal{A}} \Big\{c_{ij}^0 + \Big(1 - c_{ij}^0 \sum_{\ell=1}^{L} \lambda^\ell\Big) v_j^0 + c_{ij}^0 \sum_{\ell=1}^{L} \lambda^\ell v_j^\ell\Big\}, \quad i \in \mathcal{N} \setminus \{D\}. \tag{2.2}$$

Similarly, for $\ell = 1, \ldots, L$ we obtain

$$v_i^\ell = \min_{(i,j)\in\mathcal{A}} \Big\{c_{ij}^\ell + \Big(1 - c_{ij}^\ell \mu^\ell\Big) v_j^\ell + c_{ij}^\ell \mu^\ell v_j^0\Big\}, \quad i \in \mathcal{N} \setminus \{D\}. \tag{2.3}$$

To simplify the exposition, let us temporarily make a further (over-)simplification: assume that it is reasonable to neglect cases when more than one state transition occurs in the time of vehicle's entire journey, so that the transition diagram is given in Figure 2.1. Then,

[1] If more than one transition is possible, the probabilities $p_{0\ell}$ are given by the first row of the matrix $P(t) = \exp(Gt)$, with $t = c_{ij}^0$ and the *generator* G having nonzero entries: $g_{00} = -\sum_{\ell=1}^{L} \lambda^\ell$, $g_{0\ell} = \lambda^\ell$, $g_{\ell 0} = \mu^\ell$, $g_{\ell\ell} = -\mu^\ell$, $\ell = 1, \ldots, L$ (see [9]). If the travel time on (i, j) changes instantaneously when the state changes (due to the change in speed), the calculation becomes more involved, because the next state ℓ is not the state of the network after a fixed time c_{ij}^0 but after a random *stopping time* $\tilde{c}_{ij}$. Again, the evaluation of the transition probabilities is possible by using the location of the vehicle on this arc instead of time, and the corresponding transition rates. We shall not pursue this interesting avenue here; our assumptions allow the approximation $P(t) = \exp(Gt) \approx I + Gt$, because Gt is small.

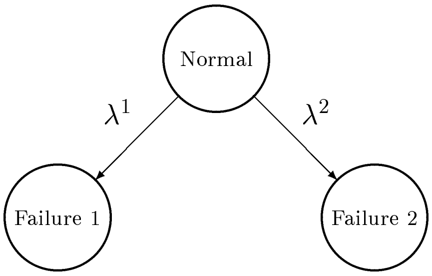

Fig. 2.1. Transition diagram for a one-step lookahead policy.

for the purpose of designing the route for a vehicle starting at state 0, we may set $\mu^\ell = 0$ in (2.3) so that it becomes the following system:

$$v_i^\ell = \min_{(i,j)\in\mathcal{A}} \{c_{ij}^\ell + v_j^\ell\}, \quad i \in \mathcal{N} \setminus \{D\}, \quad \ell = 1, \ldots, L. \tag{2.4}$$

These equations can be solved for each $\ell = 1, \ldots, L$ independently, and their solutions substituted into (2.2). The expected total travel cost is equal to $\sum_{n\in\mathcal{N}\setminus\{D\}} s_n v_n^0$. We shall call the solution thus obtained a *one-step lookahead policy* (see [5]).

If the initial state of the network is $\ell \neq 0$ when a vehicle starts its journey, the one step lookahead policy is given by a similar approximation: we set all $\lambda^\ell = 0$, $\ell \neq 0$, in (2.2), but consider (2.3) in its full form.

Our considerations make it clear how to proceed when more that one transition has a probability that cannot be neglected. We shall return to this question in sections 5 and 6, where more general n-step lookahead policies are considered.

3 The flow formulation

The purpose of this section is to develop a flow model of the uncapacitated problem discussed in section 2. This model will be more difficult to analyse and to solve than the dynamic programming equations. However, it will be a more convenient starting point for the development of models for the capacitated case, where the applicability of dynamic programming is rather limited.

Before proceeding to the details, let us stress again that in the case when failures may occur and when the information about that can be used for re-routing, the initial choice of the path should accomodate this possibility. Our flow-based models will help us to formalize this fundamental issue.

Suppose that the state of the network at time $t = 0$ is 0. Consider the vehicles starting in a time interval $[0, \varDelta]$ with some small $\varDelta > 0$ and assume that vehicles starting from each node follow some pre-specified path. Let us denote by $f_{ij}(\varDelta)$ the expected number of these vehicles which reach arc (i,j) at state 0 (at any time $t \geq 0$), and by $g^{\ell}_{ij}(\varDelta)$, $\ell \neq 0$, the expected number of these vehicles that enter (i,j) at state $\ell \neq 0$. We shall develop the flow balance of these vehicles at node i. Observe that the vehicles traversing an arc (k,i) experience with probability $c^0_{ki}\sum_{\ell=1}^{L}\lambda^l$ a state transition before reaching i, thus disappearing from our balance of vehicles at state 0 (we use (A2) to neglect the chance of two or more transitions). On the other hand, vehicles that entered (k,i) at state $\ell \neq 0$ have the chance of $\mu^l c^{\ell}_{ki}$ of getting to i at state 0 and creating an extra inflow at i. The expected source inflow at i at network's state 0 will be $s_i\varDelta - o_i(\varDelta)$, where $o_i(\varDelta)$ represents the expected number of vehicles that start in the interval $[0, \varDelta]$ at network's state $\ell \neq 0$. Clearly, $o_i(\varDelta)/\varDelta \to 0$ as $\varDelta \to 0$. Thus

$$\sum_{(i,j)\in\mathcal{A}} f_{ij}(\varDelta) - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c^0_{ki}\sum_{\ell=1}^{L}\lambda^{\ell}\Big) f_{ki}(\varDelta) - \sum_{\ell=1}^{L}\sum_{(k,i)\in\mathcal{A}} c^{\ell}_{ki}\mu^{\ell} g^{\ell}_{ki}(\varDelta) = s_i\varDelta - o_i(\varDelta). \quad (3.1)$$

Let us denote

$$x_{ij} = \lim_{\varDelta\to 0} f_{ij}(\varDelta)/\varDelta, \quad y^{\ell}_{ij} = \lim_{\varDelta\to 0} g^{\ell}_{ij}(\varDelta)/\varDelta, \quad \ell = 1,\ldots,L.$$

The variables x_{ij} and y^{ℓ}_{ij} can be interpreted as expected marginal flows at states 0 and ℓ. These limits exist, because if all vehicles starting from some node follow the same path, the number of those which reach a given arc at a certain state of the network is proportional to the number of vehicles that started.

The optimal values of the expected 'marginal flows' x_{ij} and y^{ℓ}_{ij} solve the following linear programming problem:

$$\min\Big\{\sum_{(i,j)\in\mathcal{A}} c^0_{ij}x_{ij} + \sum_{\ell=1}^{L}\sum_{(i,j)\in\mathcal{A}} c^{\ell}_{ij}y^{\ell}_{ij}\Big\} \quad (3.2)$$

$$\sum_{(i,j)\in\mathcal{A}} x_{ij} - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c^0_{ki}\sum_{\ell=1}^{L}\lambda^{\ell}\Big) x_{ki} - \sum_{\ell=1}^{L}\sum_{(k,i)\in\mathcal{A}} c^{\ell}_{ki}\mu^{\ell} y^{\ell}_{ki} = s_i, \quad i \in \mathcal{N}\setminus\{D\}, \quad (3.3)$$

$$-\lambda^{\ell}\sum_{(k,i)\in\mathcal{A}} c^0_{ki}x_{ki} + \sum_{(i,j)\in\mathcal{A}} y^{\ell}_{ij} - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c^{\ell}_{ki}\mu^{\ell}\Big) y^{\ell}_{ki} = 0, \quad i \in \mathcal{N}\setminus\{D\},$$

$$\ell = 1,\ldots,L, \quad (3.4)$$

$$x_{ij} \geq 0, \quad y^{\ell}_{ij} \geq 0, \quad (i,j) \in \mathcal{A}, \quad \ell = 0,\ldots,L. \quad (3.5)$$

Equation (3.3), which results from dividing (3.1) by $\varDelta$ and passing to the limit with $\varDelta \to 0$, expresses the balance of expected marginal flows at node i in state 0, while (3.4) (by a similar argument) is the flow balance in state $\ell \neq 0$.

Let us stress that we consider in these balances only vehicles that started at state 0. We do not need to include vehicles that started after the state transition, because in the uncapacitated case they do not interact with the vehicles which are already on the way. In the capacitated case, which will be discussed in sections 7–9, we shall revise this model and consider all vehicles.

The objective is the total expected travel time of vehicles that start in the normal state.

We shall show that at a certain optimal basic solution to (3.2)–(3.5) the dual variables are the optimal costs-to-go satisfying the dynamic programming equations (2.2)–(2.3).

Let us define a *stable* optimal basis of (3.2)–(3.5) as a basis which remains optimal when the demands s_i in (3.3) are replaced by $s_i + \varepsilon$, and the zero demands in (3.4) by ε, for all $i \in \mathcal{N} \setminus \{D\}$ and all sufficiently small $\varepsilon > 0$. We shall show that one of optimal bases must be stable.

Let $\varepsilon_k \downarrow 0$, and let B_k be an optimal basis for the problem with demands in (3.3) and (3.4) equal to $s_i + \varepsilon_k$ and ε_k, respectively. At least one of these bases must appear at infinitely many k; this is our stable basis. Indeed, passing with k to ∞ we see that it is optimal for the original problem. Moreover, at each node i at least one of the variables x_{ij} corresponding to the outgoing arcs $(i,j) \in \mathcal{A}$ must be basic. The last observation follows from the fact that this basis defines a feasible basic solution for the demand $s_i + \varepsilon_k > 0$, and the only positive coefficients in (3.3) are those at x_{ij}. The same applies to the variables y_{ij}^{ℓ} in the row (3.4) with demand $\varepsilon_k > 0$, for all $\ell \neq 0$. It follows that every node $i \in \mathcal{N} \setminus \{D\}$ has at least $L+1$ basic variables associated with it: one for each $\ell = 0, \dots, L$. Since the number of constraints (3.3)–(3.4) is equal to the number of nodes in $\mathcal{N} \setminus \{D\}$ multiplied by $L+1$, each node i has exactly one basic variable x_{ij} associated with state 0, and exactly one basic variable y_{ij}^{ℓ} for each state $\ell \neq 0$. [2]

The dual variables v_i^{ℓ} associated with a stable optimal basis yield non-negative reduced costs

$$c_{ij}^0 - v_i^0 + \Big(1 - c_{ij}^0 \sum_{\ell=1}^{L} \lambda^{\ell}\Big) v_j^0 + c_{ij}^0 \sum_{\ell=1}^{L} \lambda^{\ell} v_j^{\ell} \geq 0, \quad i \in \mathcal{N} \setminus \{D\},$$

$$c_{ij}^{\ell} - v_i^{\ell} + \Big(1 - c_{ij}^{\ell} \mu^{\ell}\Big) v_j^{\ell} + c_{ij}^{\ell} \mu^{\ell} v_j^0 \geq 0, \quad i \in \mathcal{N} \setminus \{D\}, \quad \ell = 1, \dots, L.$$

[2] In contrast to the deterministic shortest path problem, the basic arcs do not have to form a spanning tree. Indeed, in the absence of arc capacities it may be profitable to traverse a loop until the state of the network changes. Consider the example with two nodes: 1 and 2 (destination node) and arcs $(1,1)$ and $(1,2)$. If $c_{1,2}^0 = 1$, $c_{1,2}^1 = 100$ and $\mu^1 = 0.1$, in state 1 it is optimal to choose arc $(1,1)$ (wait for the return to the normal state).

For each i and each ℓ the inequality for the basic variable becomes an equation, so (2.2)–(2.3) is satisfied.

If we neglect the possibility of more than one transition per journey, the flow formulation simplifies:

$$\min\Big\{\sum_{(i,j)\in\mathcal{A}} c_{ij}^0 x_{ij} + \sum_{\ell=1}^{L}\sum_{(i,j)\in\mathcal{A}} c_{ij}^\ell y_{ij}^\ell\Big\} \tag{3.6}$$

$$\sum_{(i,j)\in\mathcal{A}} x_{ij} - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c_{ki}^0 \sum_{\ell=1}^{L}\lambda^\ell\Big) x_{ki} = s_i, \quad i\in\mathcal{N}\setminus\{D\}, \tag{3.7}$$

$$-\lambda^\ell \sum_{(k,i)\in\mathcal{A}} c_{ki}^0 x_{ki} + \sum_{(i,j)\in\mathcal{A}} y_{ij}^\ell - \sum_{(k,i)\in\mathcal{A}} y_{ki}^\ell = 0, \quad i\in\mathcal{N}\setminus\{D\}, \quad \ell=1,\dots,L, \tag{3.8}$$

$$x_{ij}\ge 0, \quad y_{ij}^\ell \ge 0, \quad (i,j)\in\mathcal{A}, \quad \ell = 0,\dots,L. \tag{3.9}$$

Again, by an identical argument, the dual variables associated with (3.7)–(3.8) at a stable optimal basic solution satisfy the simplified dynamic programming equations (2.2)–(2.4).

We emphasize the fact that the variables in (3.6)–(3.9) are expected flows rather than flow realizations under different scenarios. To obtain the flow realization X in the normal operation mode (the *plan*) we find the basic variable x_{ij_B} for each $i\in\mathcal{N}\setminus\{D\}$; there is only one for each i at a stable basic solution. The arc (i, j_B) is the one to choose under the normal operation mode, that is

$$X_{ij_B} = s_i + \sum_{(k,i)\in\mathcal{A}} X_{ki}, \quad i\in\mathcal{N}\setminus\{D\}.$$

Scenarios 'normal-failure ℓ' in our problem are actually collections of many events, so the 'flow' in this case is still a random object. But again, for each $\ell\neq 0$ there is only one basic variable associated with node i and it indicates which arc should be chosen as a detour from this node in case the state changes to ℓ.

Problem (3.6)–(3.9) is similar to two-stage stochastic programming problems with x playing the role of first stage variables, and y^ℓ being the second stage variables in scenario $\ell = 1,\dots,L$. However, we should keep in mind that our scenarios are aggregates of very large collections of events, and both x and y represent expected values over these collections. Such aggregate representation is possible owing to the linearity of the cost function.

4 The equivalent first stage problem

Supposing that the expected flow of the plan x is fixed, the re-routing flows y^ℓ, $\ell = 1, \dots, L$, in (3.6)–(3.9) can be calculated from the second stage problems:

$$\min \sum_{(i,j)\in\mathcal{A}} c_{ij}^\ell y_{ij}^\ell \tag{4.1}$$

$$\sum_{(i,j)\in\mathcal{A}} y_{ij}^\ell - \sum_{(k,i)\in\mathcal{A}} y_{ki}^\ell = \lambda^\ell \sum_{(k,i)\in\mathcal{A}} c_{ki}^0 x_{ki}, \quad i \in \mathcal{N} \setminus \{D\}, \tag{4.2}$$

$$y_{ij}^\ell \geq 0, \quad (i,j) \in \mathcal{A}. \tag{4.3}$$

The right hand side of (4.2)—the supply—is the expected marginal flow of vehicles that experienced state transition from 0 to ℓ while passing arcs ending at node i. We notice the relation of this supply vector to the cost terms associated with x in (3.6).

The optimal value of (4.1)–(4.3) is a function of the plan x. Let us denote it by $q^\ell(x)$. Then we can compactly rewrite (3.6)–(3.9) as

$$\min \Big\{ \sum_{(i,j)\in\mathcal{A}} c_{ij}^0 x_{ij} + \sum_{\ell=1}^{L} q^\ell(x) \Big\}, \tag{4.4}$$

subject to (3.7) and the nonnegativity constraint on x.

In general, such two-stage problems need to be solved iteratively. For example, given a plan x, the second stage problems (4.1)–(4.3) provide some information about the functions $q^\ell(x)$. This information can be used to revise the plan and the iteration can be repeated. In our case, however, a one pass approach is sufficient, because the functions $q^\ell(\cdot)$ are linear. We have

$$q^\ell(x) = \sum_{(i,j)\in\mathcal{A}} v_j^\ell c_{ij}^0 x_{ij}, \tag{4.5}$$

where v_i^ℓ are dual variables (node potentials) associated with the constraints (4.2) at an optimal basic solution to (4.1)–(4.3). Indeed, by the duality theory in linear programming the optimal value in (4.1)–(4.3), if it is solvable, is equal to the optimal value of its dual, and the right hand side of (4.5) multiplied by λ^ℓ is the objective of the dual. Moreover, as known from general theory of network programming the optimal basis corresponds to a spanning tree, and the values of dual variables can be calculated from (2.4), independently of the supply; they depend only on c^ℓ.

Substitution into (4.4) yields the compact first stage problem:

$$\min \sum_{(i,j)\in\mathcal{A}} \Big(1+\sum_{\ell=1}^{L}\lambda^\ell v_j^\ell\Big) c_{ij}^0 x_{ij} \tag{4.6}$$

$$\sum_{(i,j)\in\mathcal{A}} x_{ij} - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c_{ki}^0 \sum_{\ell=1}^{L}\lambda^\ell\Big) x_{ki} = s_i, \quad i \in \mathcal{N}\setminus\{D\}, \tag{4.7}$$

$$x_{ij} \geq 0, \quad (i,j)\in\mathcal{A}. \tag{4.8}$$

As a conclusion from this simple analysis we observe that the robust paths from each node to D can be found by solving an aggregate generalized network flow problem (4.6)–(4.8) (see [2]). Its cost coefficients are not actual travel times; they are expected travel times obtained by

$$\widehat{c}_{ij} = \Big(1+\sum_{\ell=1}^{L}\lambda^l v_j^\ell\Big) c_{ij}^0.$$

It is interesting to observe that the modified arc costs $\widehat{c}_{ij}$ are obtained by a multiplicative modification of the original costs, with the multipliers independent of the travel times c_{ij}^0.

It should be stressed that paths from i to D that have equal expected travel times may have different actual travel times. This is different from the deterministic formulation where such an equality is the fundamental property of all optimal paths. Here, exchange of travel time for security is possible.

5 Scenarios 'failure–normal' and 'normal–failure–normal'

It is clear how to treat the case when the vehicles *start* at network's state ℓ. A simple example of the underlying Markov chain is given in Figure 5.1.

The travel plan ξ^ℓ for these vehicles is given by the problem

$$\min \sum_{(i,j)\in\mathcal{A}} \Big(1+\mu^\ell w_j^0\Big) c_{ij}^\ell \xi_{ij}^\ell \tag{5.1}$$

$$\sum_{(i,j)\in\mathcal{A}} \xi_{ij}^\ell - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c_{ki}^\ell \mu^\ell\Big) \xi_{ki}^\ell = s_i, \quad i \in \mathcal{N}\setminus\{D\}, \tag{5.2}$$

$$\xi_{ij}^\ell \geq 0, \quad (i,j)\in\mathcal{A}, \tag{5.3}$$

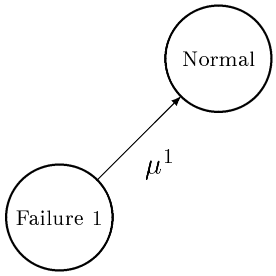

Fig. 5.1. Transition diagram for model starting in failure mode.

in which w_j^0 represents the shortest time from j to D under the normal operation mode 0. Problem (5.1)–(5.3) is identical to (4.6)–(4.8), only the role of state 0 is played by ℓ, and 0 is its only successor. Note that we use other node potentials w_j^0 than those associated with (4.7), because we neglect the possibility of another failure, once the network returns to state 0.

The analysis of scenarios 'normal-failure-normal' can be built on the top of that. Problem (5.1)–(5.3) replaces (4.1)–(4.3) to calculate the re-routing cost:

$$\min \sum_{(i,j)\in\mathcal{A}} \left(1 + \mu^\ell w_j^0\right) c_{ij}^\ell y_{ij}^\ell \tag{5.4}$$

$$\sum_{(i,j)\in\mathcal{A}} y_{ij}^\ell - \sum_{(k,i)\in\mathcal{A}} \left(1 - c_{ki}^\ell \mu^\ell\right) y_{ki}^\ell = \lambda^\ell \sum_{(k,i)\in\mathcal{A}} c_{ki}^0 x_{ki}, \quad i \in \mathcal{N} \setminus \{D\}, \tag{5.5}$$

$$y_{ij}^\ell \geq 0, \quad (i,j) \in \mathcal{A}. \tag{5.6}$$

The optimal Lagrange multipliers (node potentials) v_j^ℓ associated with the constraints (5.5) enter the first stage problem (4.6)–(4.8). They will no longer be shortest times but shortest virtual times.

6 General state transitions and variable-depth lookahead policies

The above analysis also suggests the way to develop lookahead policies for a more general case of an arbitrary Markov chain of network states.

Let $\mathcal{L}$ denote the finite collection of possible states of the network and let c_{ij}^ℓ the travel time on arc (i,j) associated with state $\ell \in \mathcal{L}$. We assume that the states evolve according to a Markov chain with transition rates μ_{kl}, $k, l \in \mathcal{L}$. The initial state is 0.

The problem is as follows. Given demands s_i, $i \in \mathcal{N} \setminus \{D\}$, where D is a fixed destination

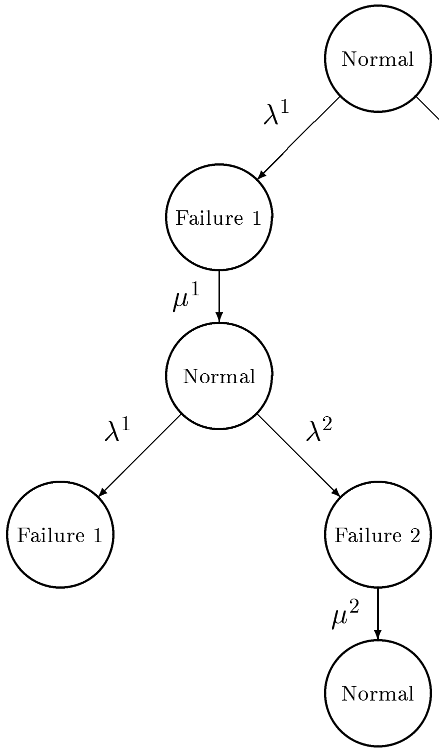

Fig. 6.1. Scenario tree for a variable-depth lookahead policy.

node, find the policy that minimizes the sum of expected travel times.

Let us develop a multistage linear programming model for this problem. At first, we select a collection of *scenarios*: sequences of states that may occur with a non-negligible probability during a journey of a vehicle. For the problem addressed in sections 1–4, these scenarios may represent sequences of failures, or different stages of a failure, which we want to distinguish because of their different effect on travel times.

It should be kept in mind that our 'scenarios' cover large collections of events, because we do not specify the times at which the transitions occur, but only the order of states visited. The scenarios form a tree $\mathcal{T}$ with the initial state at the root node and the paths from the root to the leaves identical to selected paths in the chain's graph. For example, for the state transition diagram of Figure 1.1 we may wish to consider the scenario tree of Figure 6.1.

Observe that the same state may occur at different locations in the scenario tree, and usually different characteristics (node potentials) will be associated with it. We denote by $\ell(m)$ the Markov chain's state corresponding to node $m \in \mathcal{T}$, by $\mathcal{S}(m)$ the set of successors of node m in the scenario tree and by $f(m)$ the predecessor of m in the tree. Each node $m \in \mathcal{T}$ has an associated path $\pi(m)$ from the root: it defines a unique sequence of network states that leads to $\ell(m)$.

The transition rate in $\mathcal{T}$ from node m to its successor $n \in \mathcal{S}(m)$ is $\lambda_{mn} = \mu_{\ell(m),\ell(n)}$; only transitions in the direction from the root to the leaves may occur.

Let us denote (with a slight abuse of notation) by $x_{ij}^m \Delta$ the expected number of vehicles that started at the normal state in the time interval $[0, \Delta]$ and entered arc (i, j) after experiencing state transitions of the path $\pi(m)$ (from our earlier considerations it is clear that it is for small Δ proportional to the length of this interval). The problem can be formulated as follows.

$$\min \sum_{m\in\mathcal{T}} \sum_{(i,j)\in\mathcal{A}} c_{ij}^{\ell(m)} x_{ij}^m \tag{6.1}$$

$$\sum_{(i,j)\in\mathcal{A}} x_{ij}^0 - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c_{ki}^0 \sum_{n\in\mathcal{S}(0)} \lambda_{0n}\Big) x_{ki}^0 = s_i, \quad i \in \mathcal{N} \setminus \{D\}, \tag{6.2}$$

$$-\lambda_{f(m),m} \sum_{(k,i)\in\mathcal{A}} c_{ki}^{\ell(f(m))} x_{ki}^{f(m)} + \sum_{(i,j)\in\mathcal{A}} x_{ij}^m - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c_{ki}^{\ell(m)} \sum_{n\in\mathcal{S}(m)} \lambda_{mn}\Big) x_{ki}^m = 0, \quad i \in \mathcal{N} \setminus \{D\}, \quad m \in \mathcal{T} \setminus \{0\}. \tag{6.3}$$

$$x_{ij}^m \ge 0, \quad (i,j) \in \mathcal{A}, \quad m \in \mathcal{T}. \tag{6.4}$$

In this problem $c_{ij}^{\ell(m)}$ denotes the travel time along the arc (i, j) in the state $\ell(m)$ associated with the node m of the scenario tree; similarly, $c_{ki}^{\ell(f(m))}$ is the travel time in the state associated with the parent node.

To solve (6.1)–(6.4), we associate with each node m of the scenario tree $\mathcal{T}$ the following linear program

$$\min \sum_{(i,j)\in\mathcal{A}} \Big(1 + \sum_{n\in\mathcal{S}(m)} \lambda_{mn} v_j^n\Big) c_{ij}^{\ell(m)} x_{ij}^m \tag{6.5}$$

$$\sum_{(i,j)\in\mathcal{A}} x_{ij}^m - \sum_{(k,i)\in\mathcal{A}} \Big(1 - c_{ki}^{\ell(m)} \sum_{n\in\mathcal{S}(m)} \lambda_{mn}\Big) x_{ki}^m = s_i^m, \quad i \in \mathcal{N} \setminus \{D\}, \tag{6.6}$$

$$x_{ij}^m \ge 0, \quad (i,j) \in \mathcal{A}. \tag{6.7}$$

In this problem the demand vector s^m (if $m \neq 0$) is the expected marginal flow of vehicles to be re-routed inherited from the predecessor $f(m)$:

$$s_i^m = \lambda_{f(m),m} \sum_{(k,i)\in\mathcal{A}} c_{ki}^{\ell(f(m))} x_{ki}^{f(m)}, \quad i \in \mathcal{N} \setminus \{D\}.$$

At the root node $s^0 = s$. The constants v_j^n appearing in the objective are potentials of the network's nodes that correspond to the constraints (6.6) at the optimal basic solutions of successor nodes $n \in \mathcal{S}(m)$. Again, analogously to section 4, the optimal value of (6.5)–(6.7) is linear with respect to the demand s_i^m. Consequently, all these problems can be solved in a one-pass approach by traversing the scenario tree from the leaves to the root.

We shall omit the straightforward manipulations which show that such a tree of problems indeed solves the problem of finding the best route, when only the scenarios included into the tree may occur. It almost exactly repeats our argument for the 'normal–failure' and 'normal–failure–normal' scenarios considered earlier, except that the notation would obscure the otherwise clear idea.

7 Introduction to the capacitated problem

Let us now consider a more general case with arc capacities u_{ij}^ℓ, associated with states $\ell = 0, \dots, L$. The main difficulty, as compared to the uncapacitated network, is that we cannot work with expected values of flows, as in section 3. Actual flow realizations under different events must be considered. We also cannot ignore the interactions between different flow subvectors, if they share the same arc at the same time. So, events and time must be present in our model.

We shall again restrict our attention to the basic case of state 0 representing the normal operation mode, and states $\ell = 1, \dots, L$ representing failure modes, as described in section 1. In the case of capacitated arcs the dynamic programming approach becomes extremely difficult, because the current location of all vehicles must be incorporated into the state vector. We shall follow the flow modeling approach described in sections 3–5 to develop an appropriate one-step lookahead model.

Let us consider the model (3.6)–(3.9), and define $Y^\ell = y^\ell/\lambda^\ell$, $\ell = 1, \dots, L$. Further, let X_{ij} be the flow of vehicles that enter arc (i,j) in the normal operation mode. Assuming $1 - c_{ki}^0 \sum_{\ell=1}^L \lambda^\ell \approx 1$ in (3.7), we may use the approximation $X_{ij} \approx x_{ij}$. Dividing (3.8) by λ^ℓ we obtain the problem:

$$\min\Big\{ \sum_{(i,j)\in\mathcal{A}} c_{ij}^0 X_{ij} + \sum_{\ell=1}^{L} \lambda^\ell \sum_{(i,j)\in\mathcal{A}} c_{ij}^\ell Y_{ij}^\ell \Big\} \tag{7.1}$$

$$\sum_{(i,j)\in\mathcal{A}} X_{ij} - \sum_{(k,i)\in\mathcal{A}} X_{ki} = s_i, \quad i \in \mathcal{N} \setminus \{D\}, \tag{7.2}$$

$$-\sum_{(k,i)\in\mathcal{A}} c_{ki}^0 X_{ki} + \sum_{(i,j)\in\mathcal{A}} Y_{ij}^\ell - \sum_{(k,i)\in\mathcal{A}} Y_{ki}^\ell = 0, \quad i \in \mathcal{N} \setminus \{D\}, \quad \ell = 1, \dots, L, \tag{7.3}$$

$$X_{ij} \ge 0, \quad Y_{ij}^\ell \ge 0, \quad (i,j) \in \mathcal{A}, \quad \ell = 0, \dots, L. \tag{7.4}$$

The vector X in the above problem plays the role of the *plan*—the flow to be used in the normal state—while the vectors Y^ℓ are the flows of re-routed vehicles in states $\ell \neq 1$ (that is, vehicles that experienced the state transition from 0 to ℓ during their journey). Equation (7.2) is the flow conservation constraint in the normal operation mode. Equation (7.3) is the flow conservation constraint for vehicles that started in the normal operation mode but experienced transition to state ℓ during their journey.

The subtle difference between (7.1)–(7.4) and (3.6)–(3.9) is that here we ignore the fact that the arcs that lie further from the sources on the plan's routes have a slightly smaller chance of being used, because state transition may occur before a vehicle reaches them. We keep, though, the terms with λ^l in the objective function (7.1), because they may contain large re-routing costs $c_{ij}^\ell Y_{ij}^\ell$. Apart from this small inaccuracy, the model (7.1)–(7.4) allows us to take into account the arc capacities for the plan. However, we still cannot formulate capacity constraints for the re-routed flows, because we do not know at which time they will need particular arcs, and what will be their interaction with the vehicles that start *after* the transition occurred. In other words, we need a non-stationary model of the transient period immediately after the state transition.

8 The nonstationary re-routing problem

Let us assume that all travel times are integer and let M be an upper bound on all of them. Suppose that a transition from state 0 to state ℓ takes place, and let $t = 0$ denote the time of this transition. Let $Y_{ij}^\ell(t)$ be the flow of re-routed vehicles entering arc (i, j) at time t. They satisfy the flow conservation equations

$$\sum_{(i,j)\in\mathcal{A}} Y_{ij}^\ell(t) - \sum_{\substack{(k,i)\in\mathcal{A}\\ c_{ki}^\ell \le t}} Y_{ki}^\ell(t - c_{ki}^\ell) = \sigma_i(t), \quad i \in \mathcal{N} \setminus \{D\}, \quad t = 0, 1, 2, \ldots, \tag{8.1}$$

where $\sigma_i(t)$ is the inflow into i of the vehicles that experienced the state transition while traveling along the arcs leading to i:

$$\sigma_i(t) = \sum_{\substack{(k,i)\in\mathcal{A}\\ c_{ki}^0 > t}} X_{ki}. \tag{8.2}$$

Since the supply (8.2) vanishes after a finite time (for which an upper bound M is known), we know that the flows Y^ℓ will vanish after a finite time, too, although this time may be much larger than M.

Let $X_{ij}^\ell(t)$ denote the flow of vehicles that started when network's state was ℓ and that enter arc (i, j) at time t. Since we have many sources, and the network is not layered, we cannot ignore the interactions of the rescheduled flow Y^ℓ with the flow $X^\ell(t)$. We make a simplifying assumption that further state transitions do not occur during the time that we are calculating X^ℓ. Even with this assumption, we cannot avoid modeling the initial non-stationary phase, when the re-routed flow $Y^\ell(t)$ and the new flow $X^\ell(t)$ interact. The policy that we develop under this assumption is termed a one-step lookahead policy.

Denoting by T the optimization horizon and by $Z^\ell(t) = Y^\ell(t) + X^\ell(t)$ the effective flow after the state transition, we obtain the problem

$$\min \sum_{t=0}^{T} \sum_{(i,j)\in\mathcal{A}} c_{ij}^\ell Z_{ij}^\ell(t) \tag{8.3}$$

$$\sum_{(i,j)\in\mathcal{A}} Z_{ij}^\ell(t) - \sum_{\substack{(k,i)\in\mathcal{A}\\ c_{ki}^\ell \le t}} Z_{ki}^\ell(t - c_{ki}^\ell) = s_i + \sigma_i(t), \quad i \in \mathcal{N} \setminus \{D\}, \quad t = 0, 1, \ldots, T, \tag{8.4}$$

$$0 \le Z_{ij}^\ell(t) \le u_{ij}^\ell, \quad (i,j) \in \mathcal{A}, \quad t = 0, 1, \ldots, T, \tag{8.5}$$

where the additional supply $\sigma_i(t)$ is given by (8.2). Let us note the fundamental difference between (8.4) and (3.4): in the capacitated case *all* vehicles entering arc (i,j) must be considered.

The optimal value $Q^\ell(X)$ of (8.3)–(8.5) is the rescheduling cost for the plan X, when transition to state ℓ occurs.

Our further considerations are based on the assumption that the value T does not matter for determining the robust plan X, provided T is large enough, and that the solution to (8.3)–(8.5) becomes for large t equal to a solution of the 'steady-state' problem associated with state ℓ:

$$\min \sum_{(i,j)\in\mathcal{A}} c_{ij}^\ell \overline{X}_{ij}^\ell$$

$$\sum_{(i,j)\in\mathcal{A}} \overline{X}_{ij}^\ell - \sum_{(k,i)\in\mathcal{A}} \overline{X}_{ki}^\ell = s_i, \quad i \in \mathcal{N} \setminus \{D\},$$

$$0 \le \overline{X}_{ij}^\ell \le u_{ij}^\ell, \quad (i,j) \in \mathcal{A}.$$

The detailed analysis of this issue exceeds the scope of the present note. From now on we shall simply assume that T is large enough and fixed, e.g., at the double of the maximum travel time. We also assume that a transition from one plan to another is feasible. This can be guaranteed by the existence of uncapacitated arcs (i,i) of length one for each node i which effectively model 'waiting' at node i.

To avoid some terminal effects associated with the fact that the vehicles that start late cannot make it to the destination anyway, and therefore choose short arcs, we may augment

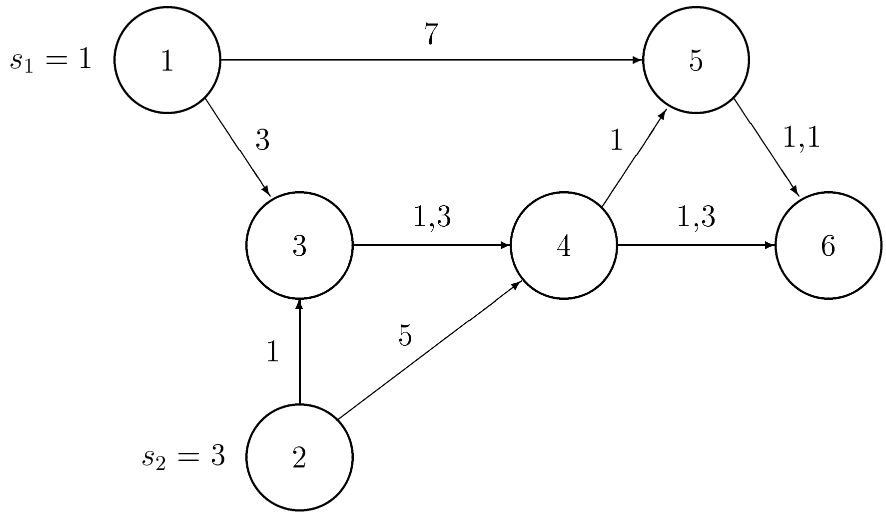

Fig. 8.1. Counter-example for steady-state attainment. The numbers at the arcs are travel times, or pairs (travel time, capacity) if capacity is finite.

(8.3)–(8.5) with terminal conditions:

$$Z_{ij}^{\ell}(t) = \overline{X}_{ij}^{\ell}, \quad t = T - \tau, T - \tau + 1, \ldots, T - 1, T, \quad (i, j) \in \mathcal{A}, \tag{8.6}$$

where τ is some constant (for example, the maximum travel time on the arcs). In fact, by choosing T (or τ) one may change the allowed length of the transient period, before the flow settles on the new steady-state solution.

If there is no possibility of waiting (no uncapacitated arcs (i, i) of length one) it is not guaranteed that the flow must settle on the 'steady-state' solution $\overline{X}^{\ell}$ at all, as the example shown in Figure 8.1 demonstrates. The example has 6 nodes, the destination node being $D = 6$ and apart from the capacities noted in the figure, all other arcs are uncapacitated. There is only one failure state, in which the travel time on the arc $(2, 3)$ increases to $c_{23}^1 = 10$; other times and capacities remain unchanged. It is clear that the optimal solution in state 0 is to send the flow of 3 units along 2-3-4-6, and 1 unit along 1-5-6. After the failure on arc (2,3), the flow from node 2 switches to the path 2-4-6, but there is no way to switch the flow from node 1 to the now empty and shorter path 1-3-4-5-6. Indeed, assume that the failure occurs at time $t = t_0$, and that $Z_{13}(t) = 1$ for $t \geq t_0$. Consider the flow

$\xi_4(t)$ coming to node 4 at time t. It is not difficult to see that

$$\xi_4(t) = \begin{cases} 3 & \text{for } t < t_0 + 2, \\ 0 & \text{for } t \in [t_0 + 2, t_0 + 4), \\ 1 & \text{for } t \in [t_0 + 4, t_0 + 5), \\ 4 & \text{for } t \geq t_0 + 5. \end{cases}$$

By the capacity constraint on $(4, 6)$, it follows that $Z_{45}(t) = 1$ for $t \geq t_0 + 5$. But then the flow $\xi_5(t)$ coming to node 5 equals 2 for $t \in [t_0 + 6, t_0 + 7]$, because there are still vehicles on the arc $(1, 5)$ that were sent *before* the failure occurred: at $t \in [t_0 - 1, t_0)$. This violates the capacity constraint on the arc $(5, 6)$ in this time interval.

9 The robust capacitated problem

We are now ready to formulate the robust planning problem in the capacitated case:

$$\min\Big\{ \sum_{(i,j)\in\mathcal{A}} c^0_{ij} X_{ij} + \sum_{\ell=1}^{L} \lambda^\ell Q^\ell(X) \Big\} \tag{9.1}$$

$$\sum_{(i,j)\in\mathcal{A}} X_{ij} - \sum_{(k,i)\in\mathcal{A}} X_{ki} = s_i, \quad i \in \mathcal{N} \setminus \{D\}, \tag{9.2}$$

$$0 \leq X_{ij} \leq u^0_{ij}, \quad (i, j) \in \mathcal{A}. \tag{9.3}$$

The functions $Q^\ell(X)$ are the optimal values of the re-routing problems in scenarios $\ell = 1, \ldots, L$.

Problem (9.1)–(9.3) is similar to two-stage stochastic programming problems (see [13,18,22] and the references therein). Much is known about these problems, and efficient solution techniques exist that exploit the structure of the model in question (see [22] and the references therein). We summarize below the facts that are relevant for our case.

The simplest approach is to include the linear programs defining $Q^\ell(X)$ into (9.1)–(9.3) and construct a giant linear programming problem with a dual block angular structure:

$$\min \sum_{(i,j)\in\mathcal{A}} \Big(c^0_{ij} X_{ij} + \sum_{\ell=1}^{L} \lambda^\ell \sum_{t=0}^{T} c^\ell_{ij} Z^\ell_{ij}(t) \Big) \tag{9.4}$$

subject to (9.2)–(9.3) and (8.4)–(8.5). This problem can be solved by standard linear programming techniques, such as the simplex method or interior point methods.

This large scale linear program can also be used to derive the optimality conditions that provide some additional insight into our model. Let $w_j^\ell(t)$ be the optimal values of Lagrange multipliers associated with (8.4) for $j \in \mathcal{N} \setminus \{D\}$ and $\ell = 1, \ldots, L$. Adding the constraints (8.4) multiplied by $w_j^\ell(t)$ to the objective (9.4) we observe that the optimal plan X solves the following network programming problem:

$$\min \sum_{(i,j)\in\mathcal{A}} \Big(c_{ij}^0 + \sum_{\ell=1}^{L} \lambda^\ell \sum_{t=0}^{c_{ij}^0 - 1} w_j^\ell(t) \Big) X_{ij}$$

subject to (9.2)–(9.3). It is similar to the deterministic flow problem, only the costs are modified in a manner analogous to (4.6)–(4.8).

Another solution approach, especially useful when the full linear programming formulation is very large, is to apply special decomposition methods developed for two-stage stochastic programming problems [19]. These methods work with the compact formulation (9.1)–(9.3) and use objective and feasibility cuts for Q^ℓ, $\ell = 1, \ldots, L$, to construct sufficiently good models of (9.1). In this way problems with very many scenarios can be solved, because derivation of objective and feasibility cuts can be carried out within independent subproblems for $\ell = 1, \ldots, L$.

10 Computational Results

We have investigated the effects of the modeling formats described in the previous sections of this paper on a simple example using the Sioux Falls network with data that is given in [11]. In this example, the network has 76 arcs and 24 nodes with symmetric traveling times given in Table 10. Node 24 was taken as the destination node with the demands for trips to this node being given by

$$0.11 \times [1, 0, 0, 2, 0, 1, 1, 2, 2, 8, 6, 5, 8, 4, 4, 3, 3, 0, 1, 4, 5, 11, 7]$$

The factor 0.11 was used only to conform with the data given in [11]. Replacing the factor by 1 and modifying the arc capacities conformally leads only to changes in the sizes of the flows given in Figures 10.7 and 10.9.

Our computations were carried out using the modeling language GAMS [8], calling the CPLEX simplex code (with default options) to solve the resulting linear programs.

We considered first the uncapacitated case as a means of illuminating the model formats described in Sections 3 and 5. Related computational work for Section 4 can be found in [10]. We initially solved (3.2)–(3.5) with $\lambda^\ell = \mu^\ell = 0$ to find the deterministic solution in this simple case. This essentially amounts to finding shortest paths in the network. The

Table 10.1
Traveling times for Sioux Falls network.

i,j	c^0_{ij}	i,j	c^0_{ij}	i,j	c^0_{ij}	i,j	c^0_{ij}
1 , 2	6	1 , 3	4	2 , 6	5	3 , 4	4
4 , 5	2	5 , 6	4	5 , 9	5	6 , 8	2
9 , 8	1	8 , 7	3	3 , 12	4	4 , 11	6
9 , 10	3	8 , 16	5	7 , 18	2	12 , 11	6
11 , 12	5	10 , 16	5	16 , 18	3	10 , 17	8
16 , 17	2	11 , 14	4	10 , 15	6	17 , 19	2
14 , 15	5	15 , 19	4	14 , 23	4	15 , 22	4
23 , 22	4	12 , 13	3	23 , 24	2	22 , 21	2
19 , 20	4	22 , 20	5	18 , 20	4	13 , 24	4
24 , 21	3	21 , 20	6				

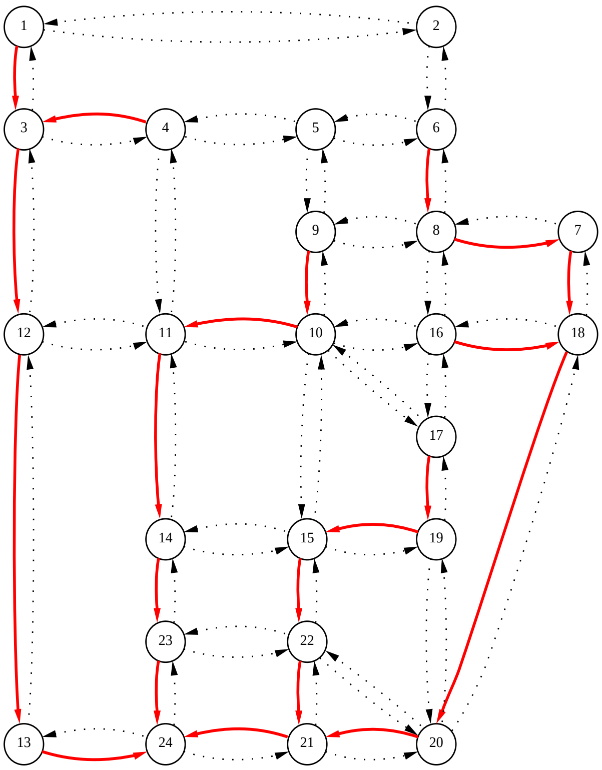

Fig. 10.1. Deterministic solution for Sioux Falls network.

form of the solution is depicted in Figure 10.1. Since the form of the solution is more illuminating than the values of the flows in the plan, we only depict the paths on which there is nonzero flow in all our figures.

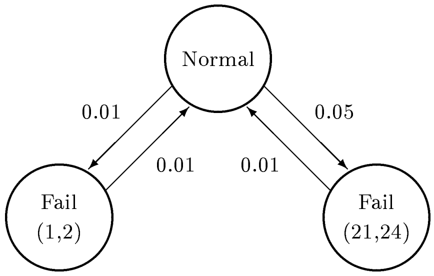

Fig. 10.2. Transition diagram for full model.

We then investigated the full effect of our Markov modeling paradigm in the uncapacitated case by allowing certain failures to occur. Thus starting in the normal state, we allow for the possibility of failure on two arcs, namely arc (1,2) and arc (21,24). These were chosen for illustration purposes due to the fact that (1,2) has no flow in the base case, and (21,24) is the heaviest used arc. By failure, we mean that the cost of the arc rises from 6 or 2 respectively to 100. The failure rates of these arcs were 0.01 and 0.05 respectively. (While these costs and rates do not satisfy assumption (A2), they were chosen to emphasize the effects of a failure. Smaller values for arc failure costs and/or failure rates change the numerical values in the solutions, but the conclusions drawn from the model are essentially unchanged.) Furthermore, we allow the possibility of these arcs being fixed and the corresponding transition rates were chosen as 0.01. The resulting transition diagram is shown in Figure 10.2. The robust plan arising from solving (3.2)–(3.5) with the above inputs is shown in Figure 10.3. Note the only change in the form of this plan over the deterministic plan is that flow from node 6 is sent along the path 6-5-4-3-12-13-24, instead of along 6-8-7-18-20-21-24. It is interesting to observe that only flow paths from nodes that are remote from the destination are changed. It is intuitively clear that if we wish to make the plan more robust to possible failures on the two given arcs at minimal cost, then we should only penalize the nodes that really have some ability to change. In the example, the largest v^0 is found at node 6, changing the effective costs of the paths and resulting in the new path flow noted above. The actual costs of the two paths are 21 and 20 respectively.

The rerouting flows are also found by analyzing the solution of (3.2)–(3.5). If arc (21,24) fails, the only rerouting changes to the flow pattern are shown in Figure 10.4. Note that we have already redirected the flow from 6 so that it does not use arc (21,24). The changes that are indicated are essentially unavoidable because in the model we have postulated, large amounts of flow arrive at nodes 20 and 21 before we have time to react to the information that a failure has occurred.

As we would expect, there is no rerouting performed due to a failure on arc (1,2). However, at this point, we note that is is important to use the full generality of the model format to generate realistic scenarios of events. If we were to use the over simplified Markov model shown in Figure 10.5, along with the problem (3.6)–(3.9) then rerouting can occur when arc (1,2) fails. The deficiency in this over simplified model is that once a failure has

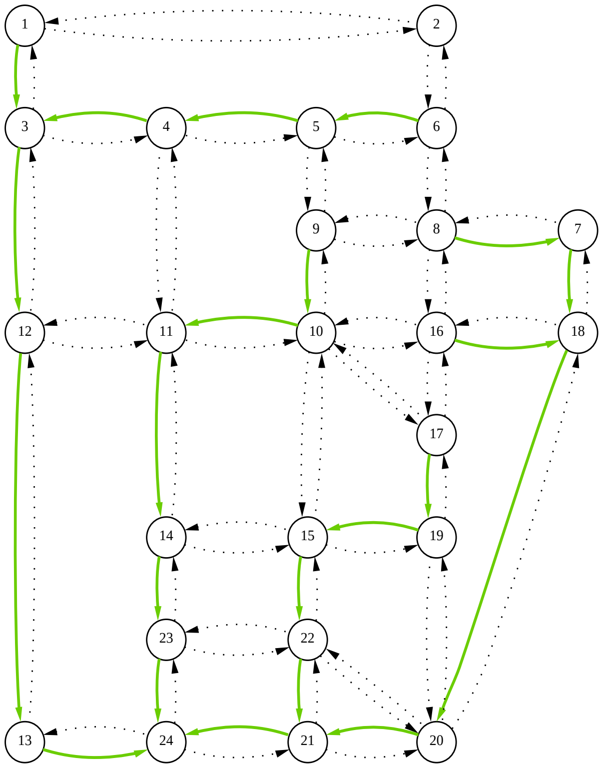

Fig. 10.3. Robust solution for Sioux Falls network.

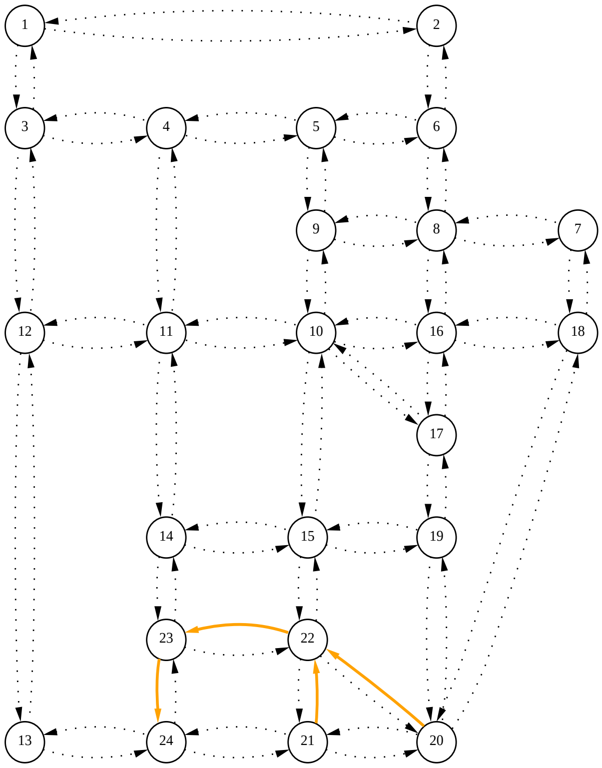

Fig. 10.4. Rerouting under failure of (21,24).

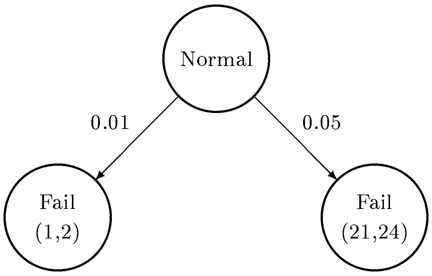

Fig. 10.5. Transition diagram for simple model.

occurred in arc (1,2), the model assumes there can be no more failures at all. Since the path 19-15-22-21-24 has the same length as 19-20-21-24 the rerouting arbitrarily chooses to send the flow along the latter path under the new certainty that arc (21,24) will not fail now.

In the realistic model postulated by Figure 10.2, the very small probability (10^{-6}) of a failure in (1,2), followed by a fix of (1,2) then a failure of (21,24) forces the flows to remain on the more robust path 19-15-22-21-24.

We note also that our robust plan is cheaper than the naive approach of following the deterministic plan and re-routing if a failure occurs. To show this in our example, we first solved for the deterministic plan as outlined above, and then fixed all the x_{ij} variables to zero that were not used in this plan. The rerouting paths were then calculated by solving (3.2)–(3.5) with the extra constraints added to fix the corresponding x_{ij} variables. The resulting plan obviously has the same structure as the deterministic plan shown in Figure 10.1. The rerouting plans in this case are identical to those given in the paragraph above. However, the cost of following this approach is about 0.1% worse than that of following our robust approach. It can be made correspondingly worse by increasing the demand of trips from node 6 to node 24 from 1 to 100, for example. In this case, the robust plan is 2.7% better than the naive plan in objective function value. A similar improvement in the robust model over the naive one can be generated by increasing the failure rate on arc (21,24).

We then turned to the capacited model and tested the formulations given in Sections 8 and 9 as follows. We incorporated capacities of 0.5 on arcs (15,14) and (22,23) so that when a failure of arc (21,24) occurs, all the flow could not be rerouted through these arcs. The resulting robust solution plan obtained from minimizing (9.4) subject to (9.2)–(9.3), (8.4)–(8.5) and (8.6) is depicted in Figure 10.6. An interesting paradox can be observed. Arcs (15,14) and (22,21) that are not used at all in the uncapacitated robust plan have saturating flow sent across them in the robust plan with capacities. This paradox can be explained by the fact that the arcs are heavily used in the uncapacitated case for rerouting. Thus, to avoid the major expense of rerouting large amounts of flow through arc (10,11) in the event that (21,24) fails, it is better to send as much flow as possible away from

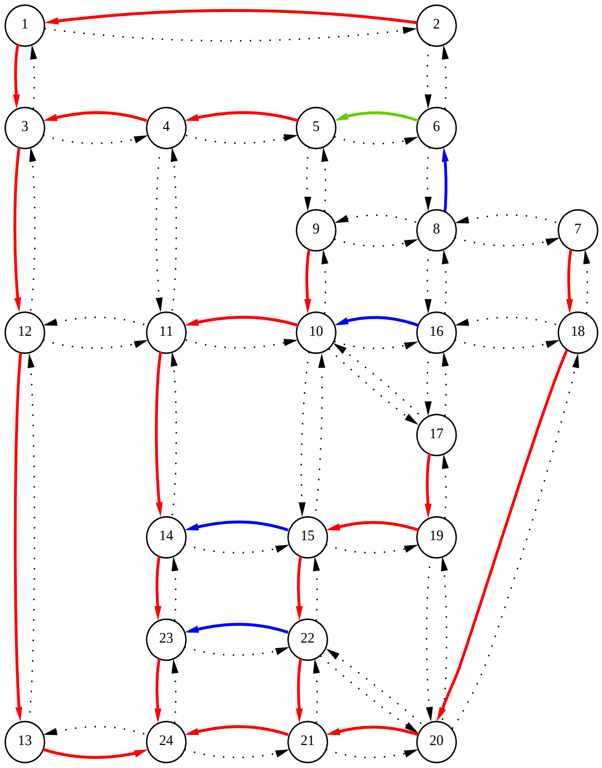

Fig. 10.6. Robust solution in capacitated case ($T = 40$).

potential bottlenecks. Also, in the capacitated robust plan, flow is sent from node 8 to node 6, which is in direct contrast to the deterministic solution depicted in Figure 10.1 which sends flow from 6 to 8.

In our GAMS implementation, we allow the modeler to specify a value T_p for the length of the transient period. This is implemented by taking the value for T as $T_p + \max c_{ij}^0$ in (8.4)–(8.5) and using a value of $\tau = \max c_{ij}^0$ in (8.6). In order to demonstrate the effect of our robust plan on the transient behavior of the jam, we show the flows on two representative arcs. These rerouting flows are calculated under two different plans. The charts on the top of Figure 10.7 depict the transient behavior of the flows on the arcs (15,10) and (10,11) under the naive plan that chooses shortest paths initially, and then reroutes the flow when a failure occurs.

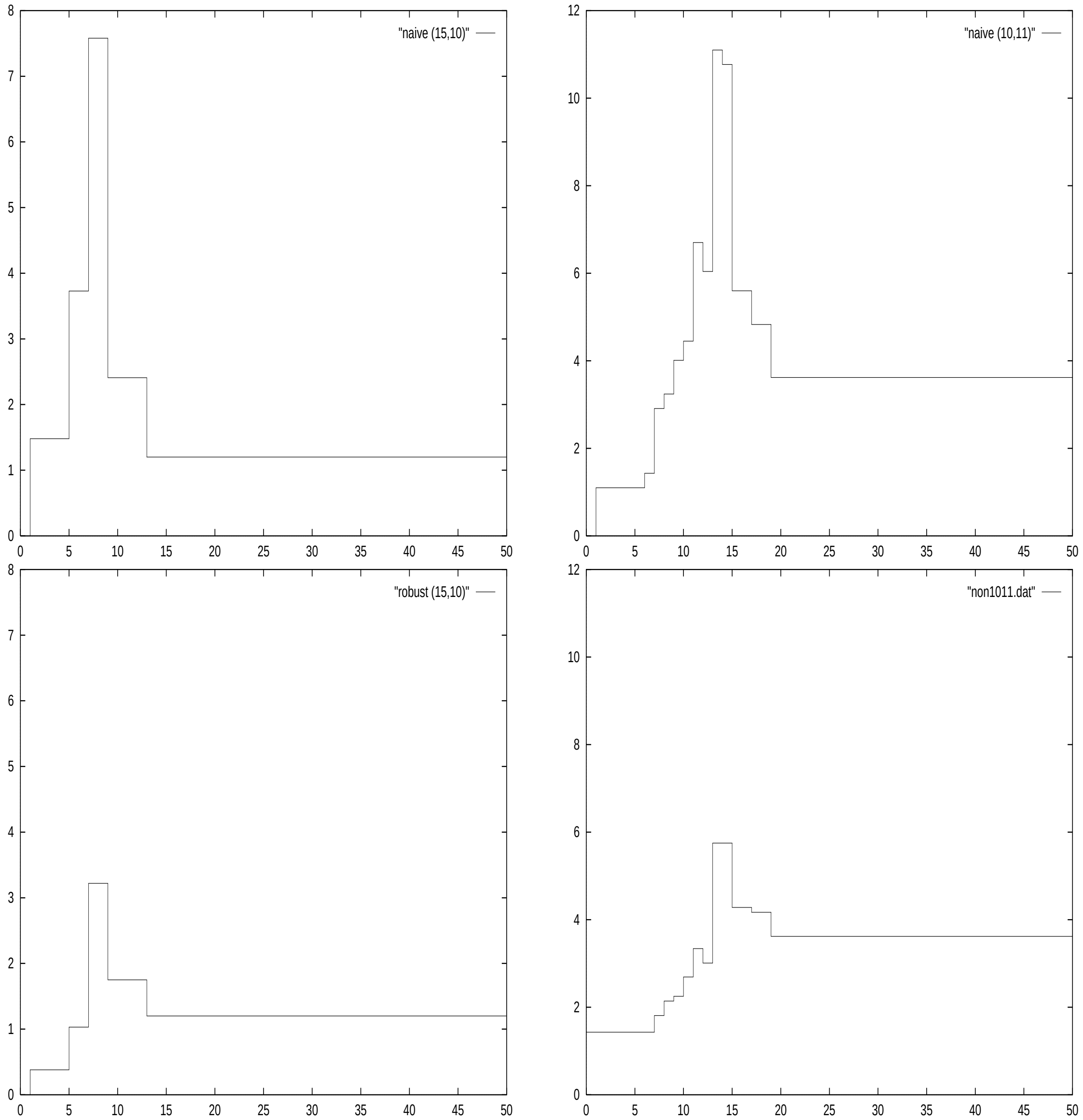

Fig. 10.7. Rerouted flow on (15,10) and (10,11) using naive plan (top) or robust plan (bottom). Failure occurs at $t = 1$.

The charts on the bottom depict the rerouting flows that occur when we follow the robust plan; both of these rerouting procedures allow a period $T_p = 40$ to attain the steady state solution. Note that on both of these arcs, the amount of flow that has to be rerouted in the robust case is less than half that needed to be rerouted when the naive plan is followed.

The model format also allows the ability to try to reach the steady state solutions in a user-specified time. Instead of allowing a time horizon of $T_p = 40$, we attempted to solve the same robust planning problem as outlined above with a horizon of $T_p = 20$. Since we are forcing the jams to disappear very quickly, it is necessary to modify our plan to allow the flows to be rerouted if necessary. The new base plan is shown in Figure 10.8; this figure should be contrasted with Figure 10.6. In particular, note that the new robust plan sends

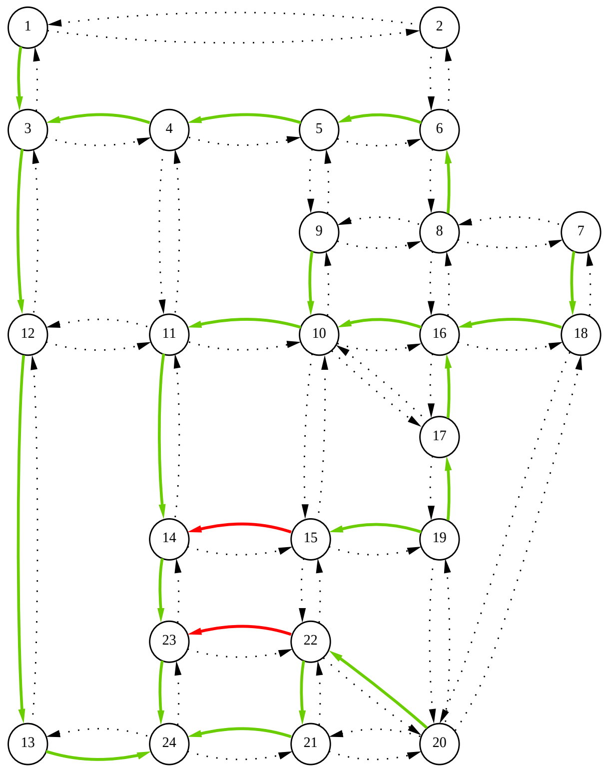

Fig. 10.8. Robust solution in capacitated case ($T_p = 20$).

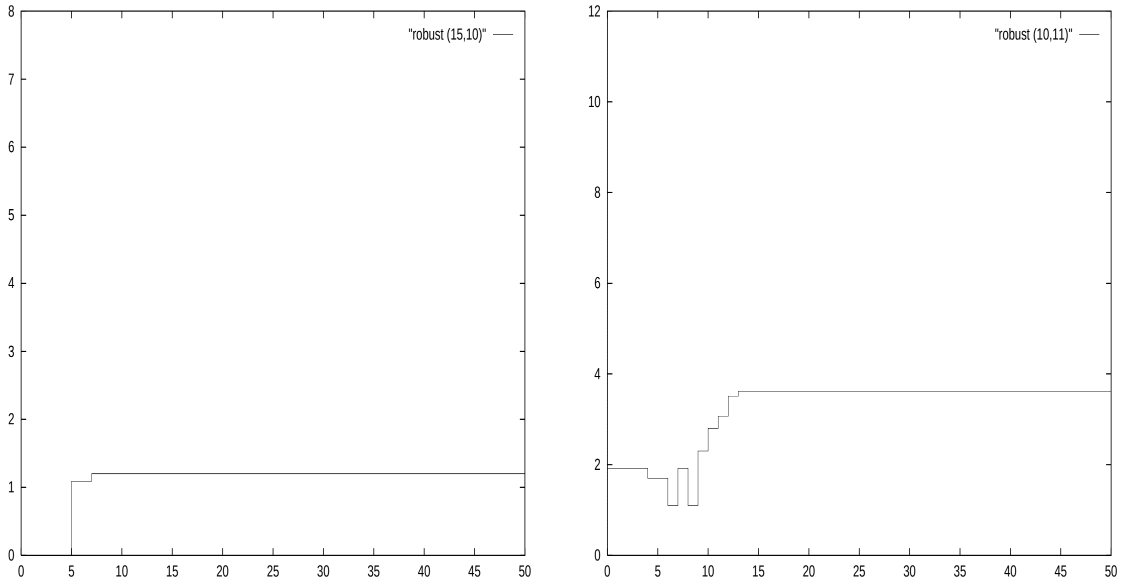

Fig. 10.9. Rerouted flow on (15,10) and (10,11) using robust base plan with $T_p = 20$.

flow from 19 to 17 (in direct contrast to the previous robust plan). The new robust plan has several features that are similar to the rerouting plan under failure of (21,24). The amounts of flow needing to be redirected are therefore smaller in this case, as can be seen in Figure 10.9. The robust plan reroutes less flow due to the fact that it has less time to perform the rerouting. In fact, some flow purposely takes longer paths to ensure that it can be rerouted quickly enough to attain the steady-state in the period $T_p = 20$ (see for example, flow on 20-22-21, and no flow on (20,21)).

This has clear ramifications in the design of fault-tolerant communication networks. When a robust plan is used for normal flow scheduling, much less redundant capacity needs to be built into the system to withstand failures. This undoubtedly leads to significant monetary savings. Furthermore, networks can be designed that allow jams to be cleared in a user-specified time frame.

Acknowledgement

The authors are indebted to three anonymous referees whose insightful remarks helped to improve the formulation and interpretation of the models described in this paper.

References

[1] R. Arnott, A. de Palma, and R. Lindsey, "Does providing information to drivers reduce traffic congestion?", *Transportation Research A*, 25 (1991) 309–318.

[2] R.K. Ahuja, T.L. Magnanti and J.B Orlin, *Network Flows: Theory, Algorithms, and Applications* (Prentice-Hall, Englewood Cliffs NJ, 1993).

[3] G. Andreatta and L. Romeo, "Stochastic shortest paths with recourse", *Networks* 18 (1988) 193–204.

[4] M. E. Ben-Akiva, A. de Palma and I. Kaysi, "Dynamic network models and driver information systems", *Transportation Research A*, 25 (1992) 251–266.

[5] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, (Athena Scientific, Belmont MA, 1995).

[6] D.P. Bertsekas and J.N. Tsitsiklis, "An analysis of stochastic shortest path problems," *Mathematics of Operations Research* 16 (1991) 580–595.

[7] D. Braess, "Über ein Paradoxon aus der Verkehrsplanung", *Unternehmensforschung* 12 (1968) 258–268.

[8] A. Brooke, D. Kendrick and A. Meeraus, *GAMS: A User's Guide*, (The Scientific Press, South San Francisco, CA, 1988).

[9] K.L. Chung, *Markov Chains with Stationary Transition Probabilities*, (Springer-Verlag, Berlin, 1960).

[10] S.P. Dirkse and M.C. Ferris, "Traffic Modeling and Variational Inequalities using GAMS", in: Ph.L. Toint, M. Labbe, K. Tanczos and G. Laporte (Eds.), *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, (NATO ASI Series F, Springer-Verlag, Berlin 1997).

[11] T. L. Friesz, D. Bernstein, N. J. Mehta, R. L. Tobin and S. Ganjalizadeh, "Day-to-day dynamic network disequilibria and idealized traveller information systems", *Operations Research* 42 (1994) 1120–1136.

[12] G.D. Glockner and G.L. Nemhauser, "Dynamic network flow with uncertain arc capacities: formulation and problem structure", report LEC-96-08, School of Industrial and Systems Engineering, Georgia Institute of Technology, June 1996.

[13] P. Kall and S.W. Wallace, *Stochastic Programming* (John Wiley & Sons, Chichester, 1994).

[14] J. D. Murchland, "Braess' paradox of traffic flow", *Transportation Research*, 4 (1970) 391–394.

[15] M. Patriksson, *The Traffic Assignment Problem, Models and Methods.* (VSP, Utrecht, NL, 1994).

[16] G.H. Polychronopoulos and J.N. Tsitsiklis, "Stochastic shortest path problems with recourse", *Networks* 27 (1996) 133–143.

[17] W.B. Powell, P. Jaillet and A. Odoni, "Stochastic and dynamic networks and routing", in: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser (Eds.), *Network Routing* (North-Holland, Amsterdam 1995), pp. 141–295.

[18] A. Prékopa, *Stochastic Programming* (Kluwer, Dordrecht, 1995).

[19] A. Ruszczyński, "Decomposition methods in stochastic programming," *Mathematical Programming*, 79 (1997) 333–353.

[20] Y. Sheffi. *Urban Transportation Networks.* (Prentice-Hall, Englewood Cliffs, USA, 1985).

[21] D. P. Watling and T. van Vuren, "The modelling of dynamic route guidance", *Transportation Research, Part C*, 1 (1993).

[22] R.J.-B. Wets, "Large scale linear programming techniques in stochastic programming," in: Yu.M. Ermoliev and R.J.-B. Wets, eds., *Numerical Techniques for Stochastic Optimization* (Springer–Verlag, Berlin, 1988), pp. 65–93.