

**A STRATEGIC MODEL FOR
DYNAMIC TRAFFIC ASSIGNMENT***

by

Younes Hamdouch¹
Patrice Marcotte¹
Sang Nguyen¹

23 octobre 2002

*This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR).

¹Département d'informatique et de recherche opérationnelle et Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, CANADA H3C 3J7

ABSTRACT

We propose a model of dynamic traffic assignment where strategic choices are an integral part of user behaviour. The model provides a discrete-time description of flow variation through a road network involving arcs with rigid capacities. A driver's strategy assigns, to each node of the network, a set of arcs in the forward star of that node, sorted according to a preference order. The main element of the model is a 'within-day' submodel where strategic volumes are loaded onto the network in accordance with the first-in first-out discipline and user preferences. An equilibrium assignment is achieved when expected delays are minimal, for every origin-destination pair. We prove the existence of such an assignment and provide numerical results on test networks.

Keywords: dynamic traffic assignment, strategy, hyperpath, capacities.

RÉSUMÉ

Cet article propose un modèle dynamique discret pour le trafic routier où le comportement des usagers est dicté par des stratégies. Le modèle permet d'analyser la variation des flots de véhicules dans les réseaux dont les arcs sont munis de capacités rigides. Chaque stratégie associe à tout nœud du réseau un ensemble d'arcs incidents, ordonné en ordre de préférence décroissante. L'élément central du modèle est un sous-modèle où les volumes stratégiques sont chargés sur le réseau en respectant les priorités temporelles (premier arrivé premier servi) ainsi que les préférences. Un équilibre est atteint lorsque tous les automobilistes sont affectés à des plus courts chemins reliant leur origine et leur destination. Nous démontrons l'existence d'une affectation respectant ce principe d'équilibre et présentons des résultats numériques sur des réseaux test.

Mots-Clés: trafic routier, affectation d'équilibre dynamique, stratégie, hyperchemin, capacités.

1 Introduction

Traffic assignment has been the topic of many studies in the last four decades, and it can be safely argued that the static case is settled by now. However, extending the results from a static to a dynamic environment is far from trivial and, as no agreement on an ideal model has been reached within the scientific community, dynamic traffic assignment remains a topic of active research.

Macroscopic dynamic models belong to one of three classes: models using exit functions (Merchant and Nemhauser [10], Carey [2], Drissi-Kaïtouni and Gendreau [4]), models based on space-time networks (Drissi and Hamada-Benchekroun [3], Zawack and Thompson [15]) and models using arc travel delays (Friesz and al. [5], Asatarita [1]). In all these models, users travel along paths of the underlying network, which path cannot be modified en route. If congestion levels become very high, however, users may choose to switch path rather than experience unduly delays on a given route. Our proposal for dealing with this situation is to assume that users behave strategically and modify their route choice according to congestion conditions. More precisely, a strategy associates with each node of the network an ordered set of successor nodes, whose access is controlled by three factors: the arrival time at the node, the capacity of the corresponding arc, the number of users with overlapping strategies.

This model adapts the strategic approach of Marcotte, Nguyen and Schoeb [9] and Hamdouch [6] to a time-varying environment. As in most dynamic models, the main challenge consists in performing efficiently the ‘loading’ operation, whereby strategic volumes are converted into path and arc volumes. It is a key feature of our approach that the loading preserves the order of arrival of users at the nodes of the network, and thus automatically fulfills the FIFO (First-In First Out) rule.

The remainder of this paper is organized as follows. In the next section, we introduce the strategic concepts underlying the dynamic assignment model and formulate the equilibrium problem as a variational inequality. Based on these results, we develop an algorithmic framework and provide numerical results. Throughout the paper, we assume that time-dependent demand and departure times are known, and that congestion is the sole consequence of waiting time at the tails of congested nodes.

2 A strategic dynamic assignment model

Let $G = (N, A)$ be a network with node set N , arc set A and time-varying demand d_{qr}^t ($t = 0, 1, \dots, T' \leq T$), for users departing origin node q at t and bound for destination node r . The objective of our research is to determine strategic volumes and travel times over a fixed time interval T whose expected delays are consistent with the travel costs c_a and the arc capacities u_a . We assume that the time interval is sufficiently large to allow all vehicles to reach their destination by T and that, whenever the number of drivers that wish to access an arc exceeds its capacity, a proportion of those commuters will either travel on an alternative (unsaturated) arc, or wait until an arc become available later in time. All operations take place on a

time-space network $R = (V, E)$ which is constructed as follows:

- each node $i \in N$ is expanded into $T + 1$ nodes i_t , $t = 0, 1, \dots, T$:

$$V = \{i_t : i \in N, 0 \leq t \leq T\}$$

- each link $a \in A$ is expanded to $T + 1 - c_a$ links (i_t, j_{t+c_a}) such that $t + c_a \leq T$; we refer to these links as **travel links**. The waiting time at the nodes is represented by means of **waiting arcs** of infinite capacity denoted (i_t, i_{t+1}) where $t = 0, 1, \dots, T - 1$.

The set of arcs is partitioned into the set E_1 of access arcs and its complement E_2 , the set of waiting arcs:

$$\begin{aligned} E_1 &= \{(i_t, j_{t+c_{ij}}) : (i, j) \in A, 0 \leq t \leq T - c_{ij}\} \\ E_2 &= \{(i_t, i_{t+1}) : i \in N, 0 \leq t \leq T - 1\} \\ E &= E_1 \cup E_2 \end{aligned}$$

As an example, the time-space network R of 5 periods corresponding to the example of Figure 1 is illustrated in Figure 2. Note that R is a standard acyclic graph with $|V| = n(T + 1)$ and $|E| \leq (m + n)T$ where $n = |N|$ and $m = |A|$.

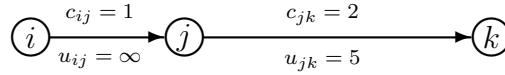


Figure 1: A simple network

2.1 Strategies

A **strategy** s is associated with an origin-destination pair $(q, r)(t)$ and maps every node j_t of the time-space network into an ordered set of successor nodes that embodies the arrival instant at j_t . More precisely, let E_j^t denote the set of all subsets of the forward star j_t^+ associated with node j_t . A strategy s assigns to each node $j_t \in V$ an element $E_j^{stt'}$ ($t' \leq t$) in E_j^t . The set $E_j^{stt'}$ is the preference order at node j_t for users adopting strategy s and having reached node j at instant t' . Three strategies for the simple network of Figure 2 are listed below:

preference order:	E_i^{s00}	E_i^{s11}	E_i^{s22}	E_j^{s11}	E_j^{s21}	E_j^{s22}	E_j^{s33}	E_k^{s55}
s_1 :	$[j_1]$	$[]$	$[]$	$[k_3, j_2]$	$[k_4]$	$[k_4, j_3]$	$[]$	$[]$
s_2 :	$[]$	$[j_2]$	$[]$	$[]$	$[k_4, j_3]$	$[k_4]$	$[k_5]$	$[]$
s_3 :	$[]$	$[]$	$[j_3]$	$[]$	$[]$	$[]$	$[k_5]$	$[]$

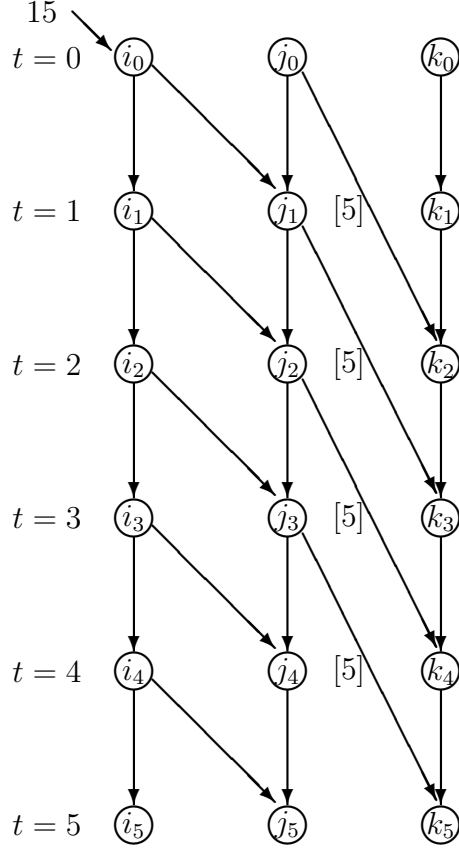


Figure 2: Expansion of the simple network

With any given strategy s , we associate the **strategic graph** $H(s) = (V^s, E^s)$, where E^s is the union of the set E_1^s of travel arcs $\{(j_t, k_{t+c_{jk}}) \in E_1 \mid k_{t+c_{jk}} \in E_j^{stt'}, t' \leq t\}$ and the set E_2^s of waiting arcs $\{(j_t, j_{t+1}) \in E_2 \mid j_{t+1} \in E_j^{stt'}, t' \leq t\}$. Every node $j_t \in V^s$ is connected to the destination $r(s)$. The strategic graph induced by strategy s_1 is shown in Figure 3.

The set of all strategies associated with OD pair $(q, r)(t)$ is denoted by S_{qr}^t and we denote by

$$S = \bigcup_{0 \leq t \leq T} S^t = \bigcup_{0 \leq t \leq T} \bigcup_{(q, r, t)} S_{qr}^t,$$

the set of all strategies.

2.2 Strategic volumes and strategic costs

Let x_s^t represent the number of drivers using strategy s and leaving their origin $q(s)$ at instant t . We denote by $x = (x_s^t)_{s \in S}$ the vector of **strategic volumes** and by X the set of all demand-feasible such vectors, i.e.,:

$$X = \{x : \sum_{s \in S_{qr}^t} x_s^t = d_{qr}^t \quad \forall (q, r), \forall t \in T\} \quad (1)$$

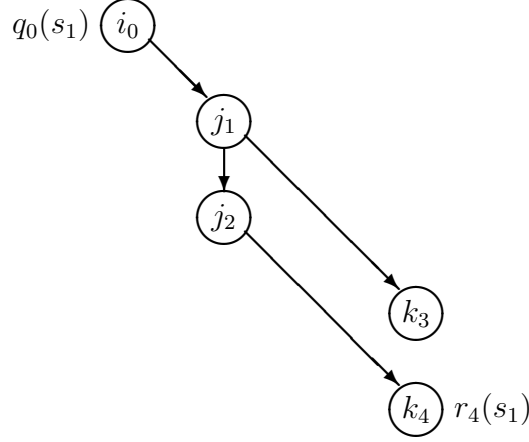


Figure 3: The strategic graph of s_1 on simple network

The cost of strategy s depends directly on the **arc access probabilities** $\pi_{jk}^{st'}(x)$ and $\pi_j^{st'}(x)$, the latter corresponding to a waiting arc; $\pi_{jk}^{st'}(x)$ (respectively $\pi_j^{st'}(x)$) is the probability that drivers adopting strategy s access node $k_{t'+c_{jk}}$ (respectively $j_{t'+1}$) from node $j_{t'}$. These access probabilities are computed by a dynamic **loading process** that will be the focus of Section 3. The arc access probabilities induce **node access probabilities**, computed recursively as follows:

$$\tau_j^{st'}(x) = \begin{cases} 0 & \text{if } j_{t'} \notin V^s \\ 1 & \text{if } j_{t'} = q_t(s) \\ \tau_j^{s(t'-1)} \pi_j^{s(t'-1)} + \sum_{k_{t_c} \in j_{t'}^-} \tau_k^{st_c} \pi_{kj}^{st_c} & \text{if } j_{t'} \in V^s \setminus \{q_t(s)\}, \end{cases} \quad (2)$$

where $t_c = t' - c_{kj}$ is the instant when a user of strategy s leaves node k through arc (k, j) .

In a natural fashion, we define the **strategic cost** of a strategy s with departure time t and destination $q(s)$ as the weighted sum:

$$C_s^t(x) = \sum_{t' \geq t} \left\{ \sum_{(j_{t'}, k_{t'+c_{jk}}) \in E_1^s} \tau_j^{st'}(x) \pi_{jk}^{st'}(x) c_{jk} + \sum_{(j_{t'}, j_{t'+1}) \in E_2^s} \tau_j^{st'}(x) \pi_j^{st'}(x) \right\}. \quad (3)$$

An equilibrium strategic flow x^* is then readily characterized as a solution of the variational inequality

$$\langle C(x^*), x^* - x \rangle \leq 0 \quad \forall x \in X. \quad (4)$$

If the cost function C were continuous, the existence of an equilibrium solution would follow directly from classical fixed point theorems, such as Brouwer's or Kakutani's. A technical difficulty arises in the degenerate situation where a null strategic volume wants to access an arc with null residual capacity. This can be resolved exactly as in Hamdouch [7], i.e., by showing that defining the access probability to zero in the degenerate case yields a lower semi-continuous cost mapping C . The proof will not be repeated here.

2.3 The FIFO rule and flow priority

A main feature of our model is that the FIFO rule is strictly enforced. At a node $j_{t'} \in V$, this is achieved dividing each group of users $z_j^{st'}$ into subgroups according to their arrival instant at node j . We denote by $z_j^{st't''}$ the strategic volume at node $j_{t'}$ adopting strategy s and having reached node j at instant $t'' \leq t'$ and regroup all active strategies into a class $\bar{S}^{t't''}$ restricted to users having reached node j at instant t'' :

$$\bar{S}^{t't''} = \{s \in S : E_j^{st't''} \neq \emptyset, z_j^{st't''} > 0\}. \quad (5)$$

Based on the access probabilities $\pi_{jk}^{st'}(x)$ and $\pi_j^{st'}(x)$, the volume $z_j^{st't''}$ is then computed according to the recursion

$$z_j^{st't''} = \begin{cases} 0 & \text{if } j_{t'} \notin V^s \\ x_s^t & \text{if } j_{t'} = q_t(s) \text{ } (t'' = t') \\ \pi_j^{s(t'-1)} z_j^{s(t'-1)t''} & \text{if } j_{t'} \in V^s \setminus \{q_t(s)\} \text{ and } t'' \leq t' - 1 \\ \sum_{k_{t_c} \in j_{t'}^-} \pi_{kj}^{st_c} z_k^{st_c} & \text{if } j_{t'} \in V^s \setminus \{q_t(s)\} \text{ and } t'' = t' \text{ } (t_c = (t' - c_{jk})). \end{cases} \quad (6)$$

In order to control access priorities, we assign the highest priority to users of the first class $\bar{S}^{t'0}$. The strategic volume of the remaining classes are then loaded in increasing order according to their arrival instant at node j . This ordering is an integral part of the loading process described in the next section.

3 The dynamic loading process

In static models, the arc-path incidence matrix is flow independent. As a consequence, link information can be readily recovered from path flow information. In dynamic models, the situation is different and calls for a mechanism that maps path (or strategic) flows into link flows. In our model, this operation is performed by a loading procedure that assigns users to their preferred available arc, until their first choices becomes saturated. Saturated arcs are then removed from the preference sets and the procedure is iterated until all users are assigned to an outgoing arc.

This procedure will be explained in detail on the example corresponding to figure 4, where capacities of the physical arcs (j, k) and (j, l) are set to 10 and 15, respectively. We focus on the loading process at node j_4 , i.e., the physical node j at instant 4. A total of 20 users are already in the waiting queue at j ; among these, 15 users have been in the queue for two periods, and the remaining 5 for one period. They are joined in the queue by 20 users that access node j from node i . All users adopt one of either two strategies: s_1 or s_2 , which are specified by the preference sets $E_j^{stt'}$ displayed in the figured. Note that the preference set of a strategy can be time dependent, which it is in our example. Next to each arc is shown its volume, together with the repartition among the two classes.

We initiate the process by loading the strategic flow associated with class \bar{S}^{42} , i.e., flow that arrived at node j at instant 2. Since capacity is not exceeded, 3 users from

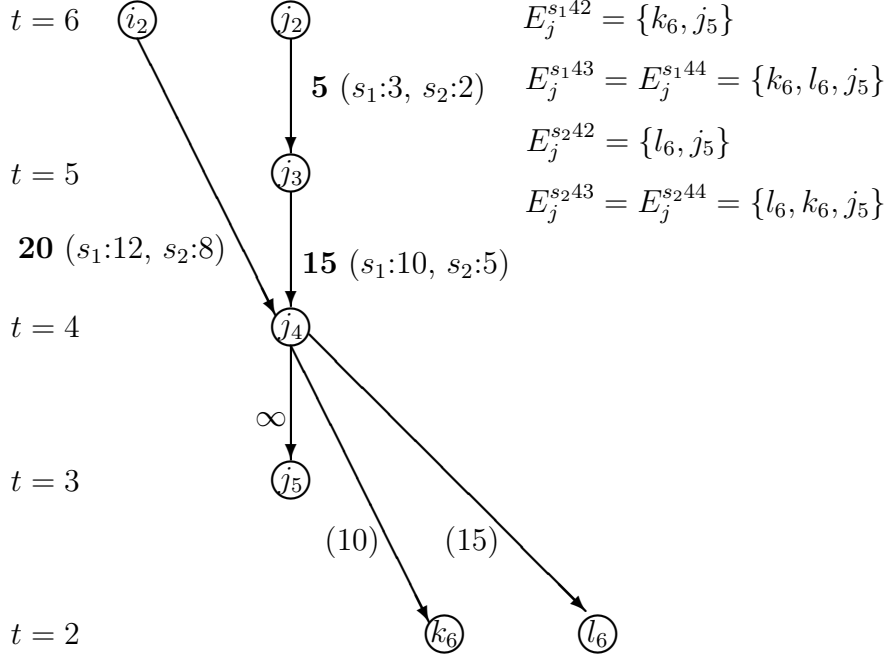


Figure 4: An example of strategic dynamic loading

s_1 and 2 from s_2 head for nodes k_6 and l_6 , respectively. Next, residual capacities are updated and the loading ends for drivers of class. At the second iteration, strategic volumes z_j^{s143} and z_j^{s243} of the second class \bar{S}^{43} are assigned to their respective preferred node at a uniform rate. Since the ratio $7/10$ is less than $13/5$, arc (j_4, k_6) gets saturated first. At that instant, $\frac{7}{10}$ of s_1 and s_2 flows are assigned, i.e., 7 users from s_1 reach node k_6 and 3.5 from s_2 reach l_6 . At iteration 3, the saturation of (j_4, k_6) forces users of strategy s_1 to adopt their second best choice l_6 ; 4.5 units of volume (3 from s_1 and 1.5 from s_2) vie for arc (j_4, l_6) whose residual capacity is 9.5. Since capacity exceeds demand, the loading of class \bar{S}^{43} terminates. At the fourth iteration, 12 units of the strategic volume z_j^{s144} and 8 of z_j^{s244} compete for the residual capacity (5) of arc (j_4, l_6) . According to the ratio 1 to 4, 3 units from s_1 and 2 from s_2 reach node l_6 . A fifth iteration is required to load the unassigned flow to the waiting arc (j_4, j_5) . The arc probabilities are then obtained by dividing the amount of strategic volume having accessed an arc by the total corresponding strategic volume. More details are provided in Table 1.

A pseudocode of the algorithm, which extends that presented in Hamdouch et al [7] for static networks with priority rules, is given in the appendix. It provides a formal description of the loading of a strategic flow vector $x = (x_s^t)_{s \in W^t \subset S^t, t \leq T}$ with respect to a subset of strategies W^t **working set**. The working set includes at least one strategy for every OD pair $(q, r)(t)$.

	arc:	(j_4, k_6)	(j_4, l_6)	(j_4, j_5)
iter. 1	res. cap. flow	10 $3(z_j^{s_1 42})$	15 $2(z_j^{s_2 42})$	∞ 0
iter. 2	res. cap. flow	7 $3(z_j^{s_1 42}) + 7(z_j^{s_1 43})$	13 $2(z_j^{s_2 42}) + 3.5(z_j^{s_2 43})$	∞ 0
iter. 3	res. cap. flow	0 $3(z_j^{s_1 42}) + 7(z_j^{s_1 43})$	9.5 $2(z_j^{s_2 42}) + 3.5(z_j^{s_2 43})$ $+ 3(z_j^{s_1 43}) + 1.5(z_j^{s_2 43})$	∞ 0
iter. 4	res. cap. flow	0 $3(z_j^{s_1 42}) + 7(z_j^{s_1 43})$	5 $2(z_j^{s_2 42}) + 5(z_j^{s_2 43})$ $+ 3(z_j^{s_1 43}) + 3(z_j^{s_1 44})$ $+ 2(z_j^{s_2 44})$	∞ 0
iter. 5	res. cap. flow	0 $3(z_j^{s_1 42}) + 7(z_j^{s_1 43})$	0 $2(z_j^{s_2 42}) + 5(z_j^{s_2 43}) + 9(z_j^{s_1 44})$ $+ 3(z_j^{s_1 43}) + 3(z_j^{s_1 44}) + 6(z_j^{s_2 44})$ $+ 2(z_j^{s_2 44})$	∞
access probabilities $\pi_{jk}^{s_1 4} = \frac{10}{25}$ $\pi_{jl}^{s_1 4} = \frac{6}{25}$ $\pi_j^{s_1 4} = \frac{9}{25}$ $\pi_{jl}^{s_2 4} = \frac{9}{15}$ $\pi_j^{s_2 4} = \frac{6}{15}$				

Table 1: Dynamic loading process at node j_4

4 Computation of an optimal strategy

The algorithms that we implemented are extensions to the dynamic case of algorithms presented in [6]. At each iteration, one moves from the current strategic vector toward another vector that incorporates a ‘best’ response to current traffic conditions, for each origin-destination pair. More precisely we construct, for every OD pair $(q, r)(t)$ and instant t , a strategy $s_{qrt}^*(x)$ such that:

$$C_{s^*}^t(x) = \min_{s \in S_{qr}^t} \{C_s^t(x)\}.$$

The construction of the solution s^* to the above problem is achieved by an algorithm that plays the role of the shortest path algorithm in standard static traffic assignment. It is based on dynamic programming and uses the information (strategic volumes, access probabilities) generated by the loading procedure. One conceptual difficulty is that access probabilities associated with strategies not yet in the working set are unknown. However, a **micro loading** procedure can be devised to perform that very operation.

The knowledge of the strategic vector z allows us to start the micro-loading at node $j_{t'}$ which must comply with the FIFO rule. At each instant p ($t \leq p \leq t'$), the optimal preference set $E_j^{s^*t'p}$ is constructed by running the micro-loading over the sets $\bar{S}^{t't}, \bar{S}^{t't+1}, \dots, \bar{S}^{t'p-1}$ and $\bar{S}^{t'p} \cup \{s^*\}$ given that the users of the optimal (unknown) strategy s^* have reached node j at instant p . The dependence of the preference set on the arrival instant at node j is key to enforcing the FIFO condition at node $j_{t'}$. The recursive process of Nguyen and Pallottino [11] for computing a shortest hyperpath can then be applied to the space-time network R . The expected travel cost from node $j_{t'}$ to the destination r_T , denoted $\omega_{pj}^{s^*t'}$, is obtained from Bellman’s generalized Bellman:

$$\omega_{pj}^{s^*t'} = \begin{cases} \infty & \text{if } j \neq r, t = T \\ 0 & \text{if } j = r \\ \sum_{k_{t''} \in E_j^{s^*t'p}} \pi_{jk}^{s^*t'} (c_{jk} + \omega_{t''k}^{s^*t''}) + \pi_j^{s^*t'} (1 + \omega_{pj}^{s^*t'+1}) & \text{if } j \neq r, t < T, \end{cases} \quad (7)$$

where $t'' = t' + c_{jk}$ is the arrival instant at node $k \in j^-$, and where the optimal preference order $E_j^{s^*t'p}$ is constructed efficiently by sorting the labels $c_{jk} + \omega_k^{s^*t''}$ in increasing order, with

$$\omega_k^{s^*t''} = \begin{cases} \omega_{pk}^{s^*t''} & \text{if } k_{t''} = k_{t'+c_{jk}} = j_{t'+1} \\ \omega_{t''k}^{s^*t''} & \text{otherwise.} \end{cases}$$

An optimal solution s^* , together with its cost vector ω^* , is then obtained by solving Bellman’s equations in reverse topological order, i.e., starting at instant T .

We close this section with an illustration of one iteration of the algorithm within the space-time network. Let us consider the situation depicted in Figure 5, where the relevant data comprises the arrival instants, the arc flows, their associated strategies, the preference orders associated with strategies s_1 and s_2 , delays, capacities (in

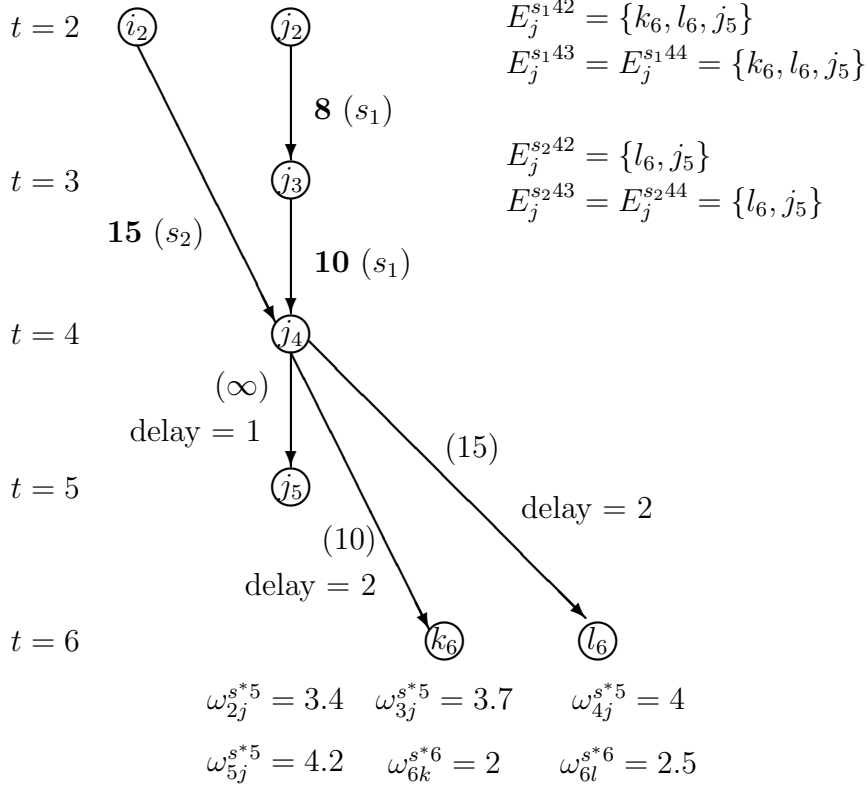


Figure 5: Computing an optimal strategy at node j_4

parenthesis) and cost-to-go (bottom of picture) that are available at nodes j_5, k_6 and l_6 , since their topological index is higher than that of node j . We proceed to compute the best strategic response, starting at the physical node j .

The optimal preference orders are obtained by minimizing the sum of arc delays and the corresponding cost-to-go. Accordingly, the inequalities

$$c_{jk} + \omega_{6k}^{s^* 6} = 2 + 2 < 1 + \omega_{2j}^{s^* 5} = 1 + 3.4 < c_{jl} + \omega_{6l}^{s^* 6} = 2 + 2.5$$

yield $E_j^{s^* 42} = [k_6, j_5, l_6]$. In a similar fashion, we derive $E_j^{s^* 43} = [k_6, l_6, j_5]$ from

$$c_{jk} + \omega_{6k}^{s^* 6} = 4 < c_{jl} + \omega_{6l}^{s^* 6} = 4.5 < 1 + \omega_{3j}^{s^* 5} = 1 + 3.7$$

and $E_j^{s^* 44} = [k_6, l_6, j_5]$ from the inequalities

$$4 < 4.5 < 1 + \omega_{4j}^{s^* 5} = 1 + 4.$$

The algorithm must now determine the values of $\omega_{2j}^{s^* 4}$, $\omega_{3j}^{s^* 4}$ and $\omega_{4j}^{s^* 4}$ at node j_4 . We start with instant 2 and the strategy set $\bar{S}^{42} \cup \{s^*\} = \{s_1, s^*\}$. Since flow does not exceed available capacity, all users from s_1 and s^* head for node k_6 . This yields the access probabilities $\pi_{jk}^{s^* 4} = 1$, $\pi_j^{s^* 4} = \pi_{jl}^{s^* 4} = 0$ and the cost

$$\begin{aligned} \omega_{2j}^{s^* 4} &= \pi_{jk}^{s^* 4}(2 + \omega_{6k}^{s^* 6}) + \pi_j^{s^* 4}(1 + \omega_{2j}^{s^* 6}) + \pi_{jl}^{s^* 4}(2 + \omega_{6l}^{s^* 6}) \\ &= 1(4) + 0(4.4) + 0(4.5) = 4. \end{aligned}$$

We now repeat the loading operation for users that arrived at instant 3. We first perform the assignment of the 8 users that arrived at instant 2 to their preferred node k_6 . Next, the residual capacity of arc (j, k) is updated (it becomes $10-8=2$) and the loading ends for drivers of strategies \bar{S}^{42} . Next, the strategic volumes $z_j^{s_1 43}$ and $z_j^{s^*}$ associated with the strategies in \bar{S}^{43} are assigned to their respective preferred node at the uniform rate $2/10$. At that instant, $\frac{1}{5}$ of s_1 and s^* flows have reached node k_6 , and the arc (j_4, k_6) is saturated. This forces users of strategy s_1 and s^* to adopt their second best choice l_6 ; 8 units of volume from s_1 and s^* access arc (j_4, l_6) , whose residual capacity (15) exceeds demand (8). This yields the access probabilities $\pi_{jk}^{s^* 4} = 1/5$, $\pi_{jl}^{s^* 4} = 4/5$ and $\pi_j^{s^* 4} = 0$, from which we derive the cost-to-go

$$\begin{aligned}\omega_{3j}^{s^* 4} &= \pi_{jk}^{s^* 4}(2 + \omega_{6k}^{s^* 6}) + \pi_{jl}^{s^* 4}(2 + \omega_{6l}^{s^* 6}) + \pi_j^{s^* 4}(1 + \omega_{3j}^{s^* 6}) \\ &= \frac{1}{5}(4) + \frac{4}{5}(4.5) + 0(4.7) = \frac{22}{5} = 4.4.\end{aligned}$$

Then, we proceed with users that reached node j at instant 4. This involves loading the flows associated with strategy sets \bar{S}^{42} , \bar{S}^{43} and $\bar{S}^{44} \cup \{s^*\}$. This yields (we skip details) $\pi_{jk}^{s^* 4} = 0$, $\pi_{jl}^{s^* 4} = 7/15$, $\pi_j^{s^* 4} = 8/15$ and $\omega_{4j}^{s^* 4}$ and

$$\begin{aligned}\omega_{4j}^{s^* 4} &= \pi_{jk}^{s^* 4}(2 + \omega_{6k}^{s^* 6}) + \pi_{jl}^{s^* 4}(2 + \omega_{6l}^{s^* 6}) + \pi_j^{s^* 4}(1 + \omega_{4j}^{s^* 6}) \\ &= \frac{7}{15}(4.5) + \frac{8}{15}(5) = \frac{71.5}{15} = 4.76.\end{aligned}$$

5 Numerical results

Numerical results have been obtained using algorithm **DSTRATEQ**, which mimics the method of successive averages (MSA) for solving monotone variational inequalities. Since the cost mapping C may fail to be monotone, this method is heuristic in our context. At each iteration, we set

$$x^{k+1} = (1 - \theta^k)x^k + \theta^k \bar{x}(x^k), \quad k = 1, 2, \dots$$

where $\theta^k \in (0, 1)$ and

$$\bar{x}_s^t(x^k) = \begin{cases} d_{qr}^t & \text{if } s = s^* \\ 0 & \text{otherwise} \end{cases}, \quad \forall (q, r) \quad \forall s \in W_{qr}^t, \quad \forall t \leq T.$$

For improved convergence, the stepsize was chosen adaptively and independently for each OD pair. More precisely:

$$\theta_s = 1 - \frac{C^{s^*}}{C_t^s}.$$

The algorithm was halted as soon as the relative gap function became smaller than some predetermined tolerance. We report numerical results on one small and one medium-sized network.

5.1 Small network

Our first test problem is based on the bidirectional network illustrated in Figure 6. The number of periods T is set to 65 and the latest departure instant T' is set to 9. Additional data for the reference scenario is shown in Tables 2 and 3. A second scenario was constructed by decreasing the capacities of arcs (2,3) and (3,2) to 5 units, and simultaneously multiplying all demand by the factor $3/2$. Key statistics are displayed in Tables 4, 5, 6 and 7, namely: number of active strategies, gap function, CPU time and minimum strategic costs.

In both scenarios, **DSTRATEQ** converged rapidly to a unique equilibrium solution, from any starting point. The relative gap function decreases to 1% after roughly 20 iterations for the both networks. Also, as expected, the increase pressure on available capacity in the second scenario yields an increase in the number of active strategies, and to increased CPU time. It can also be readily observed from the output data that FIFO is satisfied, i.e., later departures at the origin node result in later arrivals at the destination. Due to the demand patterns and the system's dynamics, travel delays increase sharply as one approaches the peak period, around instants 6 and 7.

5.2 Sioux Falls network

This classical network (see [13]) illustrated in Figure 7 comprises 24 nodes, 76 arcs and 8 *OD* pairs. The number of periods T and the latest departure time T' are set to 100 and 4, respectively.

Numerical tests have been performed on three scenarios. Tables 8 and 9 contain the relevant input data for the base scenario **A**. In scenario **B**, some arc capacities are modified (see Table 10) and demand is multiplied by a factor 2 throughout. The modifications relevant to scenario **C** are displayed in Table 11. For each scenario, the control parameters ϵ_1 and ϵ_2 have been set to 0.1.

Tables 12, 13 and 14 show that convergence occurs at a reasonable rate, with a relative gap inferior to 1% reached after less than 20 iterations. As was the case on the small network, the algorithm converged toward a unique equilibrium solution. Tables 15, 16 and 17 give the profiles of minimum travel delays for different departure times. While the nature of the solution changes, both qualitatively and quantitatively, in Scenarios **B** and **C** (note the sharp increase of the size of the working set for Scenario **B**), the rate of convergence was not greatly influenced by the increased congestion. In particular, the running time of each major iteration remained stable, at around one minute per iteration.

5.3 Variation of travel delays

In the strategic model, and equilibrium is reached when *average* delays are equalized. However, faced with two strategies with equal mean delays, a “risk-averse” user might prefer the one with the smallest standard deviation σ_s^t , defined as

$$\sigma_s^t(x) = \left[\sum_{p(t) \in P^s} \kappa_{p(t)}^s(x) (C^{p(t)})^2 - [C_s^t(x)]^2 \right]^{\frac{1}{2}},$$

where $\kappa_{p(t)}^s(x)$ is the probability of using path $p(t)$ and $C^{p(t)}$ is the actual delay along path $p(t)$.

The inclusion of a variation criterion within the model would make it intractable from a computational point of view. However, we computed, for Sioux Falls' scenario **B** the standard deviation of strategic costs for OD pair (7,20), and found a value of 0.5 . This small value is most likely to be perceived as negligible by the users, and validates *a posteriori* the choice of a single criterion in the equilibrium model.

6 Conclusion

In this paper, we have shown that the concept of strategy, previously used in the static transit assignment problem, can be adapted to the realm of dynamic traffic assignment on capacitated networks. This yields a model where the FIFO rule is automatically satisfied. The nonvacuity of the solution set was proved and the implementation showed the applicability of the concept to small networks. Moreover, the numerical results provided interpretable insights on the strategic behavior of users.

This opens up several avenues for further research. One is the extension of the work of [7] on static capacitated transit systems to a dynamic environment. Another topic of research concerns the efficiency of the loading algorithm. While its high running time follows from the number of *required* elementary operations, which can be evaluated fairly accurately, it is due primarily to the fact that all physical arcs of our test networks are capacitated. Future research will focus on taking advantage of the small number of congested arcs in real networks. Finally, the strategic approach is well suited at addressing situations where random accidents or incidents occur, and where user behaviour is dictated by the appraisal of current traffic conditions, based on past (experience) or present (on-board vehicle) information.

References

- [1] V. Astarita (1996), *A continuous time link model for dynamic network loading based on travel time function*, Proc. 13th International Symposium on Transportation and Traffic Theory (Tarrytown, N.Y.) (J.B. Lesort, ed.), Pergamon-Elsevier, pp. 79-102.
- [2] M. Carey (1992), *Nonconvexity of the dynamic traffic assignment problem*, Transportation Research B26, pp. 127-133.
- [3] O. Drissi-Kaitouni, A. Hamada-Benchekroun (1992), *A dynamic traffic assignment model and a solution algorithm*, Transportation Science 26, pp. 119-128.

- [4] O. Drissi-Kaitouni, M. Gendreau (1992), *A new dynamic traffic assignment model*, Publication CRT-854, Centre de recherche sur les transports, Université de Montréal.
- [5] T.L. Friesz, D. Bernstein, T.E. Smith, R. Tobin et B.W. Wie (1993), *A variational inequality formulation on the dynamic network user equilibrium problem*, Operations Research 41, pp. 179-191.
- [6] Y. Hamdouch (2002), *Affectation statique et dynamique des usagers dans un réseau de transport avec capacités rigides*, Ph.D dissertation, Département d'informatique et de recherche opérationnelle, Université de Montréal.
- [7] Y. Hamdouch, P. Marcotte and S. Nguyen (2002), *Capacitated traffic assignment with priority rules*, preprint.
- [8] P. Marcotte and S. Nguyen (1998), *Hyperpath formulations of traffic assignment problems*, Equilibrium and Advanced Transportation Modelling (P. Marcotte and S. Nguyen, eds.), Kluwer Academic Publisher, pp. 175-199.
- [9] P. Marcotte, S. Nguyen and A. Schoeb (2000), *A strategic flow model of traffic assignment in capacitated networks*, Publication 2000-10, Centre de recherche sur les transports, Université de Montréal.
- [10] D. Merchant et G. Nemhauser (1978), *A model and an algorithm for the dynamic traffic assignment problems*, Transportation Science 12, pp. 183-199.
- [11] S. Nguyen and S. Pallottino (1989), *Hyperpaths and shortest hyperpaths*, In B. Simeone, editor, Combinatorial Optimization, volume 1403 of Lecture Notes in Mathematics, pp. 258-271, Springer-Verlag, Berlin.
- [12] S. Nguyen, S. Pallottino et P. Malucelli (2001), *A modeling framework for the passenger assignment on a transport network with time-tables*, Transportation Science 55B, pp. 238-249.
- [13] C. Suwansirikul, T.L. Friesz and R.L. Tobin (1987), *Equilibrium decomposition optimization: a heuristic for the continuous equilibrium network design problem*, Transportation science 21, pp. 254-263.
- [14] J.G. Wardrop (1952), *Some theoretical aspects of road traffic research*, Proceedings of the Institution of Civil Engineers, Part II, pp. 325-378.
- [15] D. Zawack et G. Thompson (1987), *A dynamic space-time network flow model for city traffic congestion*, Transportation Science 21, pp. 153-162.

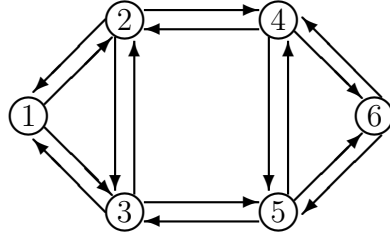


Figure 6: Small network ($T = 65$)

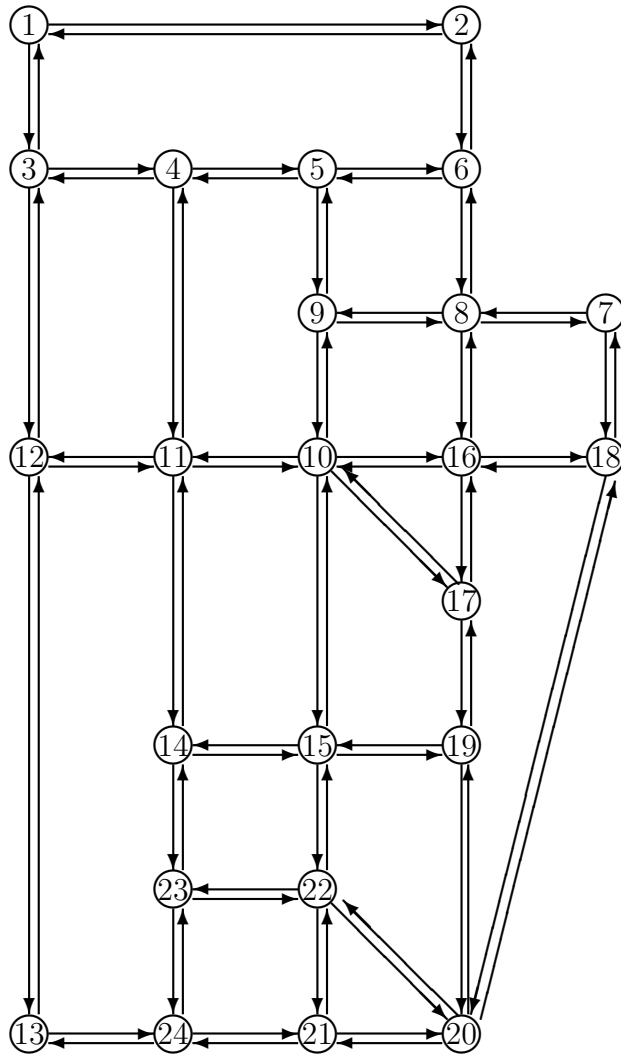


Figure 7: Sioux Falls network

Arc	Cost	Capacity	Arc	Cost	Capacity
(1,2)	3	20	(3,5)	4	10
(2,1)	3	20	(5,3)	4	10
(1,3)	2	20	(4,5)	4	10
(3,1)	2	20	(5,4)	4	10
(2,3)	1	10	(4,6)	5	25
(3,2)	1	10	(6,4)	5	25
(2,4)	2	10	(5,6)	6	25
(4,2)	2	10	(6,5)	6	25

Table 2: Small network: arc data

$OD(t)$	Demand	$OD(t)$	Demand
(1,6,0)	5	(6,1,0)	10
(1,6,1)	10	(6,1,1)	15
(1,6,2)	25	(6,1,2)	30
(1,6,3)	18	(6,1,3)	23
(1,6,4)	15	(6,1,4)	20
(1,6,5)	17	(6,1,5)	22
(1,6,6)	30	(6,1,6)	35
(1,6,7)	45	(6,1,7)	50
(1,6,8)	15	(6,1,8)	20
(1,6,9)	12	(6,1,9)	17

Table 3: Small network: OD information

# iter.	W	Gap (%)	Cpu (sec)
0	20	26.4923	3.52
1	34	11.3380	8.47
2	34	5.5682	13.37
5	36	2.8952	28.36
10	40	1.5934	54.81
20	40	0.8830	111.27
30	40	0.5842	164.44
50	40	0.3081	275.80
100	40	0.0817	555.09

Table 4: Small network: results of first scenario

$OD(t)$	Minimum cost	$OD(t)$	Minimum cost
(1,6,0)	10.000	(6,1,0)	10.000
(1,6,1)	10.000	(6,1,1)	10.333
(1,6,2)	10.799	(6,1,2)	11.333
(1,6,3)	11.888	(6,1,3)	12.156
(1,6,4)	12.000	(6,1,4)	12.189
(1,6,5)	12.000	(6,1,5)	12.263
(1,6,6)	12.209	(6,1,6)	12.617
(1,6,7)	13.075	(6,1,7)	13.831
(1,6,8)	13.360	(6,1,8)	14.536
(1,6,9)	13.018	(6,1,9)	14.555

Table 5: Small network: minimum costs after 100 iterations (first scenario)

# iter.	$ W $	Gap (%)	Cpu(sec)
0	20	40.8394	4.12
1	36	13.3624	9.61
2	36	6.1367	15.00
5	40	3.8735	32.16
10	47	2.1099	64.32
20	53	1.0616	135.83
30	55	0.7803	210.38
50	57	0.4901	367.39
100	57	0.2222	765.75

Table 6: Small network: results of second scenario

$OD(t)$	Minimum cost	$OD(t)$	Minimum cost
(1,6,0)	10.000	(6,1,0)	10.333
(1,6,1)	10.333	(6,1,1)	11.111
(1,6,2)	11.528	(6,1,2)	12.499
(1,6,3)	12.347	(6,1,3)	13.492
(1,6,4)	12.508	(6,1,4)	14.133
(1,6,5)	12.674	(6,1,5)	14.641
(1,6,6)	13.501	(6,1,6)	15.808
(1,6,7)	15.289	(6,1,7)	18.134
(1,6,8)	16.613	(6,1,8)	19.669
(1,6,9)	16.188	(6,1,9)	19.992

Table 7: Small network: minimum costs after 100 iterations (second scenario)

Arc	Cost	Capacity	Arc	Cost	Capacity
(1,2)	6	60	(13,24)	4	60
(1,3)	2	60	(14,11)	3	15
(2,1)	6	60	(14,15)	4	60
(2,6)	2	60	(14,23)	2	10
(3,1)	2	60	(15,10)	4	20
(3,4)	2	60	(15,14)	4	60
(3,12)	4	60	(15,19)	2	60
(4,3)	2	60	(15,22)	2	60
(4,5)	2	60	(16,8)	2	40
(4,11)	3	10	(16,10)	2	60
(5,4)	2	60	(16,17)	2	40
(5,6)	10	60	(16,18)	2	60
(5,9)	2	20	(17,10)	6	60
(6,2)	2	60	(17,16)	2	40
(6,5)	10	60	(17,19)	6	60
(6,8)	2	60	(18,7)	4	60
(7,8)	2	60	(18,16)	2	60
(7,18)	4	60	(18,20)	16	60
(8,6)	2	60	(19,15)	2	60
(8,7)	2	60	(19,17)	6	60
(8,9)	4	60	(19,20)	4	30
(8,16)	2	40	(20,18)	16	60
(9,5)	2	20	(20,19)	4	30
(9,8)	4	60	(20,21)	4	60
(9,10)	2	20	(20,22)	6	10
(10,9)	2	20	(21,20)	4	60
(10,11)	3	10	(21,22)	8	60
(10,15)	4	20	(21,24)	4	60
(10,16)	2	60	(22,15)	2	60
(10,17)	6	60	(22,20)	6	10
(11,4)	3	10	(22,21)	8	60
(11,10)	3	10	(22,23)	2	10
(11,12)	4	60	(23,14)	2	10
(11,14)	3	15	(23,22)	2	10
(12,3)	4	60	(23,24)	2	60
(12,11)	4	60	(24,13)	4	60
(12,13)	14	60	(24,21)	4	60
(13,12)	14	60	(24,23)	2	60

Table 8: Sioux Falls: arc data

$OD(t)$	Demand	$OD(t)$	Demand
(1,20,0)	25	(20,1,0)	25
(1,20,1)	40	(20,1,1)	40
(1,20,2)	55	(20,1,2)	55
(1,20,3)	30	(20,1,3)	30
(1,20,4)	20	(20,1,4)	20
(1,24,0)	20	(24,1,0)	20
(1,24,1)	35	(24,1,1)	35
(1,24,2)	50	(24,1,2)	50
(1,24,3)	25	(24,1,3)	25
(1,24,4)	15	(24,1,4)	15
(7,20,0)	15	(20,7,0)	15
(7,20,1)	30	(20,7,1)	30
(7,20,2)	45	(20,7,2)	45
(7,20,3)	20	(20,7,3)	20
(7,20,4)	10	(20,7,4)	10
(7,24,0)	15	(24,7,0)	15
(7,24,1)	30	(24,7,1)	30
(7,24,2)	45	(24,7,2)	45
(7,24,3)	20	(24,7,3)	20
(7,24,4)	10	(24,7,4)	10

Table 9: Sioux Falls: OD information (scenario **A**)

Arc	Capacity	Arc	Capacity
(8,16)	40	(16,8)	40
(16,17)	40	(17,16)	40
(19,20)	30	(20,19)	30

Table 10: Sioux Falls: arc capacities for scenario **B**

Pair $OD(t)$	Demand
(7,20,0)	60
(7,20,1)	120
(7,20,2)	180
(7,20,3)	80
(7,20,4)	40

Table 11: Sioux Falls: demand for OD pair (7,20) (scenario **C**)

# iter.	$ W $	Gap (%)	Cpu (sec)
0	40	6.3853	49.81
1	64	5.7933	118.56
2	67	4.9266	188.85
5	70	2.9340	406.69
10	77	1.5637	788.17
20	78	0.7630	1595.60
30	79	0.4816	2426.13
50	80	0.3279	4157.09

Table 12: Sioux Falls results (scenario **A**)

# iter.	$ W $	Gap (%)	Cpu (sec)
0	40	32.5518	55.39
1	75	20.2456	139.39
2	92	14.4513	236.32
5	123	7.0109	577.66
10	149	3.5233	1250.41
20	168	1.4144	2811.61
30	171	0.9505	4479.48
50	172	0.5053	7894.81

Table 13: Sioux Falls results (scenario **B**)

# iter.	$ W $	Gap (%)	Cpu (sec)
0	40	38.6861	56.49
1	76	24.6751	142.80
2	93	16.0157	240.55
5	130	7.5075	593.06
10	159	3.6671	1301.51
20	186	1.5922	2970.06
30	193	0.9470	4763.01
50	192	0.6316	8447.17

Table 14: Sioux Falls results (scenario **C**)

$OD(t)$	Minimum cost	$OD(t)$	Minimum cost
(1,20,0)	20.692	(20,1,0)	20.253
(1,20,1)	21.563	(20,1,1)	21.009
(1,20,2)	22.013	(20,1,2)	22.646
(1,20,3)	23.342	(20,1,3)	23.270
(1,20,4)	23.401	(20,1,4)	23.374
(1,24,0)	14.914	(24,1,0)	14.956
(1,24,1)	17.550	(24,1,1)	17.157
(1,24,2)	20.783	(24,1,2)	20.839
(1,24,3)	23.439	(24,1,3)	23.031
(1,24,4)	23.419	(24,1,4)	23.487
(7,20,0)	16.000	(20,7,0)	16.250
(7,20,1)	16.333	(20,7,1)	16.997
(7,20,2)	17.212	(20,7,2)	18.559
(7,20,3)	17.461	(20,7,3)	19.239
(7,20,4)	17.447	(20,7,4)	19.373
(7,24,0)	16.087	(24,0)	16.087
(7,24,1)	16.910	(24,1)	17.117
(7,24,2)	19.283	(24,2)	19.683
(7,24,3)	20.659	(24,3)	21.546
(7,24,4)	21.416	(24,4)	21.549

Table 15: Sioux Falls: minimum costs after 50 iterations (scenario **A**)

$OD(t)$	Minimum cost	$OD(t)$	Minimum cost
(1,20,0)	24.806	(20,1,0)	22.078
(1,20,1)	27.135	(20,1,1)	25.708
(1,20,2)	29.592	(20,1,2)	29.207
(1,20,3)	31.631	(20,1,3)	30.134
(1,20,4)	31.910	(20,1,4)	30.124
(1,24,0)	18.752	(24,1,0)	16.348
(1,24,1)	23.608	(24,1,1)	21.325
(1,24,2)	25.624	(24,1,2)	24.833
(1,24,3)	26.465	(24,1,3)	25.430
(1,24,4)	26.589	(24,1,4)	25.009
(7,20,0)	16.750	(20,7,0)	17.920
(7,20,1)	19.263	(20,7,1)	19.892
(7,20,2)	20.586	(20,7,2)	20.350
(7,20,3)	20.989	(20,7,3)	20.709
(7,20,4)	20.114	(20,7,4)	20.467
(7,24,0)	16.882	(24,0)	16.873
(7,24,1)	20.448	(24,1)	20.944
(7,24,2)	26.144	(24,2)	26.343
(7,24,3)	28.737	(24,3)	27.814
(7,24,4)	28.108	(24,4)	26.843

Table 16: Sioux Falls: minimum costs after 50 iterations (scenario **B**)

$OD(t)$	Minimum cost	$OD(t)$	Minimum costs
(1,20,0)	25.357	(20,1,0)	22.078
(1,20,1)	27.843	(20,1,1)	25.714
(1,20,2)	30.383	(20,1,2)	29.081
(1,20,3)	32.000	(20,1,3)	30.228
(1,20,4)	31.835	(20,1,4)	30.204
(1,24,0)	18.600	(24,1,0)	16.348
(1,24,1)	23.785	(24,1,1)	21.323
(1,24,2)	25.707	(24,1,2)	24.835
(1,24,3)	26.205	(24,1,3)	25.437
(1,24,4)	26.224	(24,1,4)	25.019
(7,20,0)	17.686	(20,7,0)	17.920
(7,20,1)	20.312	(20,7,1)	19.892
(7,20,2)	21.872	(20,7,2)	20.363
(7,20,3)	22.567	(20,7,3)	20.681
(7,20,4)	22.138	(20,7,4)	20.462
(7,24,0)	17.461	(24,0)	16.873
(7,24,1)	21.144	(24,1)	20.940
(7,24,2)	26.823	(24,2)	26.318
(7,24,3)	29.808	(24,3)	27.819
(7,24,4)	29.658	(24,4)	26.880

Table 17: Sioux Falls: minimum costs after 50 iterations (scenario **C**)

APPENDIX

ALGORITHM DCAPLOAD (x)

input: $x = \{x_s^t\}_{s \in W^t, t \leq T}$	<i>[strategic volume vector]</i>
output: $\{\pi_{jk}^{st}(x)\}_{(j,k) \in A, s \in W^t}$ $C = \{C_s^t(x)\}_{s \in W^t, t \leq T}$	<i>[conditional arc probabilities]</i> <i>[vector of strategic costs]</i>

INITIALIZATION

for $t = 0$ to T do for $s \in W^t$ do $C_s^t := 0$ $z_{q(s)}^{st} := x_s^t$ $\tau_{q(s)}^{st} := 1$ for $t' = t$ to T do for $j_{t'} \in V$ ($j_{t'} \neq q_t(s)$) do for $t'' = t$ to t' do $z_j^{st't''} := 0$ endfor $\tau_j^{st'} := 0$ for $k_{t'+c_{jk}} \in E_j^{st'}$ do $v_{jk}^{st'} := 0$ $\pi_{jk}^{st'} := 0$ endfor endfor endfor endfor	<i>[cost of strategy s]</i> <i>[volume at origine node $q_t(s)$]</i> <i>[probability of accessing $q_t(s)$]</i> <i>[strategic volume at node $j_{t'}$ if j has been reached at instant t'']</i> <i>[conditional probability of accessing node $j_{t'}$ using strategy s]</i> <i>[strategic volume on arc $(j_{t'}, k_{t'+c_{jk}})$]</i> <i>[conditional probability of accessing arc $(j_{t'}, k_{t'+c_{jk}})$ using strategy s]</i>
--	--

LOADING PHASE

for $t = 0$ to T do for $j_t \in V$ do for $k_{t+c_{jk}} \in j_t^+$ do $\bar{u}_{jk} := u_{jk}$ endfor $\bar{u}_j := \infty$ for $t' = 0$ to t do $z_j^{st} := z_j^{st} + z_j^{st't'}$ $\bar{S}^{tt'} := \{s \in W : z_j^{st't'} > 0\}$ endfor for $t' = 0$ to t do for $s \in \bar{S}^{tt'}$ do	<i>[time topological order]</i> <i>[residual capacity of arc $(j_t, k_{t+c_{jk}})$]</i> <i>[(j_t, j_{t+1}) is uncapacitated]</i> <i>[total strategic volume z_j^{st}]</i> <i>[set of active strategies at j_t according to instant of accessing node j]</i>
--	---

$\overline{E}_j^{stt'} := E_j^{stt'}$	[residual preference order]
$\overline{z}_j^{stt'} := z_j^{stt'}$	[residual strategic volume]
endfor	
while $\overline{S}^{tt'} \neq \emptyset$ do	[begin of loading for $\overline{S}^{tt'}$]
$K := \emptyset$	[set of residual first choices]
for $k_{t+c_{jk}} \in j_t^+$ do	[$k_{t+c_{jk}} = j_{t+1}$ if waiting arc]
$d_k := 0$	[demand for node $k_{t+c_{jk}}$]
$W_k := \emptyset$	[set of strategies having node
endfor	$k_{t+c_{jk}}$ as residual first choice]
for $s \in \overline{S}^{tt'}$ do	
$k_{t+c_{jk}} := \overline{E}_j^{stt'}(1)$	[first residual choice of strategy s]
$K := K \cup \{k\}$	[construction of K
	$k = j$ if $(j_t, \overline{E}_j^{stt'}(1))$ is waiting arc]
$W_k := W_k \cup \{s\}$	[construction of W_k]
$d_k := d_k + \overline{z}_j^{stt'}$	[construction of d_k]
endfor	
$\mu := \max\{d_k / \overline{u}_{jk} : k \in K\}$	
$\beta^{tt'} := \min\{1/\mu, 1\}$	[proportion of affected residual volume]
for $k \in K$ do	
for $s \in W_k$ do	
$t'' := t + c_{jk}$	[instant of accessing node k]
if $k = j$ then	
$z_j^{st+1t'} := z_j^{st+1t'} + \beta^{tt'} \overline{z}_j^{stt'}$	[updating volume $z_j^{st+1t'}$]
else	
$z_k^{st''t''} := z_k^{st''t''} + \beta^{tt'} \overline{z}_j^{stt'}$	[updating volume $z_k^{st''t''}$]
endif	
$v_{jk}^{st} := v_{jk}^{st} + \beta^{tt'} \overline{z}_j^{stt'}$	[updating arc strategic volume v_{jk}^{st}]
$\overline{z}_j^{stt'} := (1 - \beta^{tt'}) \overline{z}_j^{stt'}$	[updating residual volume $\overline{z}_j^{stt'}$]
$\pi_{jk}^{st} := v_{jk}^{st} / z_j^{st}$	[updating probability π_{jk}^{st}]
endfor	
$\overline{u}_{jk} := \overline{u}_{jk} - \beta^{tt'} d_k$	[updating residual capacity \overline{u}_{jk}]
if $\overline{u}_{jk} = 0$ then	
for $t'' = 0$ to t do	
for $s \in \overline{S}^{tt''}$ do	
$\overline{E}_j^{stt''} := \overline{E}_j^{stt''} - \{k_{t+c_{jk}}\}$	[updating residual preference set]
endfor	
endfor	
endif	
endfor	
if $\beta^{tt'} = 1$ then	
$\overline{S}^{tt'} = \emptyset$	[end of loading for $\overline{S}^{tt'}$]
endif	
endwhile	

```

    endfor
  endfor

  for  $s \in W^t$ 
    for  $t' = t$  to  $T$ 
      for  $j_{t'} \in V$ 
        for  $k_{t'+c_{jk}} \in j_{t'}^+$  do
           $t'' = t' + c_{jk}$  [period of accessing node k]
           $C_s^{t'} := C_s^t + c_{jk} \tau_j^{st'} \pi_{jk}^{st'}$  [update of cost  $C_s^t$ ]
           $\tau_k^{st''} := \tau_k^{st'} + \tau_j^{st'} \pi_{jk}^{st'}$  [update of probability  $\tau_k^{st''}$ ]
        endfor
         $C_s^{t'} := C_s^t + \tau_j^{st'} \pi_j^{st'}$  [update of cost  $C_s^t$ ]
         $\tau_j^{s(t'+1)} := \tau_j^{s(t'+1)} + \tau_j^{st'} \pi_j^{st'}$  [update of probability  $\tau_j^{s(t'+1)}$ ]
      endfor
    endfor
  endfor
endfor

```