AN INTERSECTION ORIGIN-DESTINATION FLOW OPTIMIZATION PROBLEM FOR EVACUATION NETWORK DESIGN

Dr. Chi Xie, Associate Professor S. Travis Waller, and Professor Kara M. Kockelman
Center for Transportation Research
Department of Civil, Architectural and Environmental Engineering
The University of Texas at Austin
1 University Station, C1761
Austin, TX 78712
Phone: (512) 471-4622
Fax: (512) 475-8744
E-mail: {chi.xie, stw, kkockelm}@mail.utexas.edu

The following paper is a pre-print and the final publication can be found in
Transportation Research Record No. 2234: 105-115, 2011.
Presented at the 90th Transportation Research Board Annual Meeting, January 2011

ABSTRACT

A lane-based evacuation network that incorporates the lane reversal and crossing elimination strategies can be virtually decomposed as a set of roadway subnetworks and intersection subnetworks. To facilitate the algorithmic advantage from this network decomposition mechanism, this paper considers an intersection origin-destination flow optimization problem arising from the evacuation network design that integrates these two capacity- and connectivity-reallocation strategies. This work presents a sufficient condition of network flows for the problem existence and validness and develops an efficient simplex-based method for problem solutions. Numerical examples are provided to illustrate the effectiveness of the method.

***Keywords*:** Intersection subnetwork, evacuation planning, lane reversal, crossing elimination, simplex method, the Hitchcock-Koopmans transportation problem

INTRODUCTION

In seeking the most effective ways to minimize the traffic congestion and disaster threat over an urban or regional evacuation network, many models aim at optimizing system performance by diversely routing evacuees to evade traffic bottlenecks, controlling their departure times to avoid jam creation, or manipulating network configurations to increase throughout capacity . Among optimization-oriented evacuation planning models, those based on physical network reconfigurations are typically formulated as optimal network design or redesign problems. Here, redesign means that most evacuation-network design problems involve only short-term, tactical-level (or operational-level) network reconfigurations based on existing network capacity and topology, rather than strategic, permanent alterations to network infrastructure.

Two types of network reconfiguration strategies, namely lane reversal along roadways and crossing elimination at intersections, have been introduced into evacuation network design in recent years. These two strategies supplement one another by increasing capacity in specific traffic directions and creating an interruption-free traffic environment throughout the evacuation network. While both have proven reasonably effective as capacity-increasing measures (for accelerating the evacuation process), their combination can enhance performance further.

Solving this design problem with lane reversal and crossing elimination, however, poses a great challenge, because of the complex mutual connectivity requirement imposed by the two lane-based network design strategies and the large number of solution spaces generated (by the lane-based network setup, in which each intersection must be explicitly modeled as a subnetwork). One way to reduce the problem's complexity is to relax the crossing elimination constraints and redefine the relaxed problem, on the standard node-arc network. Such simplification can be characterized by Lagrangian relaxation. To compensate for relaxation of the crossing elimination constraints, one needs to evaluate the number of traffic crossing points at each intersection and substitute the evaluation result back to the solution algorithm to guide its search itinerary. This evaluation requirement results in an intersection origin-destination (O-D) flow optimization problem. As will be shown later, under this specific modeling context, the O-D flow optimization problem's objective is to find an intersection O-D flow pattern that minimizes the number of traffic crossing points without altering the entire network's flow pattern. It is clear that this problem differs from more traditional work in intersection O-D flow estimation, which generally seeks to replicate a most-likely O-D flow pattern (see, for example, *1*-*6*).

The remainder of this paper is organized into five sections. The evacuation network design problem is introduced with lane reversal and crossing elimination. Then, the intersection subnetwork O-D flow optimization problem is formulated, and conditions for problem existence and validity are discussed. The major contribution of this work is the development of a simplex-based solution method for the proposed O-D flow optimization problem. Sections 4 and 5 elaborate on and illustrate the method. The paper's final section concludes with some modeling extensions.

EVACUATION NETWORK OPTIMIZATION WITH LANE REVERSAL AND CROSSING ELIMINATION

Evacuation planning with lane reversal and crossing elimination has been formulated as a lane-based network design problem (see *7*-*11*). These two lane-based capacity- and connectivity-reallocation settings alter the network's capacity and connectivity properties along roadway sections and at intersections, respectively.

Lane reversal is not a new concept. The use of lane reversal results in traffic "contraflow" or "counterflow" operation, which is often used to better accommodate traffic demand imbalances across the two opposed driving directions of a single congested roadway section. A number of early studies concerning the design, efficiency, feasibility and safety issues of lane reversal can be seen in, for example, MacDorman (*12*), Glickman (*13*), Hemphill and Surti (*14*), and Caudill and Kuo (*15*). An update on the development of lane reversal techniques and applications as well as its current state of planning and engineering practices was recently provided by Wolshon and Lambert (*16*). In evacuation cases, the traffic direction of inbound lanes along some designated roadways may be reversed to better accommodate outbound traffic. This lane-reallocation strategy has been used extensively by several U.S. states along the Atlantic and Gulf coasts for hurricane evacuations (since its initial implementation in Georgia during the period of

Hurricane Floyd in 1999) (*17*). In state and regional evacuations, lane reversal is typically applied to the major arteries (i.e., interstate and state highways).

Given the discrete nature of lane-based configurations, seeking an optimal allocation of lane reversals across an evacuation network typically poses an NP-hard combinatorial optimization problem. The problem is intractable for networks of realistic size. So a number of researchers, including Hamza-Lup et al. (*18*, *19*), Tuydes and Ziliaskopoulos (*20*), Kim et al. (*21*) and Meng et al. (*22*), have developed a set of heuristic and metaheuristic methods (including tabu search, simulated annealing, and genetic algorithms) to approximate network solutions for the lane reversal optimization problem.

Crossing elimination at intersections has attracted relatively little attention in evacuation planning and management. Cova and Johnson (*23*) suggested using this measure as a lane-based routing strategy for emergency evacuations to reduce traffic control delays at intersections (e.g., delays due to traffic signals and stop signs). The basic rationale for applying a policy of crossing elimination during evacuation events is to convert an intersection with interrupted-flow situations into an uninterrupted-flow facility by prohibiting some turning movements (through blocking lane entries and limiting flow directions). By eliminating stop-and-go traffic control devices, intersection capacity for permitted traffic movements is significantly augmented.

A few benefits from implementing the crossing elimination strategy are evident. First, evacuation planners seek to increase throughout capacity at intersections for outbound directions. Second, this strategy channels traffic flow along certain movements and reduces the possibility of potential traffic conflicts, thereby potentially improving traffic safety at intersections. This feature is particularly useful during an emergency evacuation, when the driving population may confront chaos and panic, and be forced to drive in a more aggressive manner. Third, during a post-disaster evacuation, traffic signals and communication systems may be malfunctioning due to widespread power outages and other issues. Such failure often occur following no-notice disasters (as in the aftermath of Mexico City's 1985 earthquake) (*24*).

Ideally, crossing elimination and lane reversal should be jointly used, to improve evacuation-network clearance times and overall performance. Tuydes and Ziliaskopoulos (*20*) have noted that the traffic control configurations at intersections and interchanges should be reset to maximize the efficiency of traffic movements for lane reversal operations. Xie (*7*), Kalafaras and Peeta (*8*), Xie and Turnquist (*9*, *10*), and Xie et al. (*11*) suggested simultaneous consideration of lane reversal and crossing elimination in evacuation network optimization and presented a class of integrated network design models, separately targeting static and dynamic network settings. These studies help justify that the combination of both strategies.

For illustration, Figure 1 depicts a few examples of their joint use. It is readily seen that these strategies create a set of mutual network connectivity requirements at intersections. For this reason, this paper emphasizes an explicit model for network connectivity at intersections and the resulting network design model is based on an expanded network representation (shown in Figure 2).

Such an expanded network contains a number of intersection subnetworks and roadway subnetworks. As shown in Figure 2, if all intersections are four-way with two-way legs and all through and turning movements are permitted, each intersection subnetwork consists of 8 nodes and 12 links. The roadway-section subnetwork between adjacent intersections includes 6 nodes and 4 links (where each of the lane directions is represented by a pair of consecutive directional links and for each traffic direction there is one upstream node, one downstream node and one intermediate node). The upstream and downstream nodes (e.g., nodes 2, 10, 9 and 3 in Figure 2) provide connections between the roadway section and its adjacent intersections. The intermediate nodes (e.g., nodes 11 and 12) are labeled traffic source nodes. As a common network setting in transportation planning practice, it is assumed that all the traffic collected by a roadway section from its adjacent traffic analysis zone(s) originates virtually, from the zone centroid, and connect with that section's intermediate node.

It should be noted that, in such an expanded network, arcs in an intersection subnetwork have different properties from those in a roadway subnetwork: a roadway arc is treated as an ordinary graphical arc, associated with capacity, cost, and other travel supply-demand attributes, while an intersection arc is an impedance-free arc and only functions with providing the network connectivity. This is consistent with the standard network setting of the graphical node-arc representation of a traffic network: all delays occur along arcs while nodes only provide network connectivity (with nodes representing intersections, origins, and destinations).

While the expanded representation depicts traffic networks at a finer level, it significantly increases the network's size and hence the computational cost of evaluating flows. This is especially undesirable when solving a network design problem because the network typically needs to be evaluated repeatedly; so an expanded network implies much more computation than its standard counterpart.

However, given the modeling fact that intersection arcs do not incur any cost or delay (this is especially true when crossing elimination is implemented so that no signal or stopping delay occurs at intersections), one can evaluate flows on the standard network and the expanded network equivalently (i.e., their network performance matrices [e.g., network-wide cost or congestion level] are equivalent), except that the standard network's evaluation ignores intersection crossings (since the standard network simply models intersections as nodes). In other words, by using the standard network, one ignores the set of constraints related to the crossing elimination operation in the network design model.

Thanks to such network reduction (from the expanded network to the standard network), model complexity is accordingly reduced following use of Lagrangian relaxation. This emerges by relaxing the set of crossing elimination constraints and inserting these into the objective function as a penalty term. This network reduction or Lagrangian relaxation results in two algorithmic advantages: first, as discussed above, one can evaluate the network flow pattern on the standard network, which is much more computationally efficient than the network evaluation on the expanded network; second, it simplifies the model structure and accelerates the solution search process by avoiding direct manipulation of the set of complex crossing elimination constraints in the original solution space defined on the basis of the expanded network.

Under the network reduction or Lagrangian relaxation framework, the added computational task is to evaluate the penalty term. The value of the penalty term will be used to guide the solution search process to converge to the optimal solution of the original network design problem. The value of the penalty term for each intersection, i.e., number of crossing points at an intersection, however, is not fully determined by the network flow pattern obtained from the standard network. In fact, the representation of an intersection as a node in the standard network merely treats the intersection as a "black box". One way to evaluate the penalty term is to determine the minimum number of traffic crossing points given the intersection's incoming and outgoing flows. In the intersection subnetwork, if we look at each incoming flow from an "origin" node and each outgoing flow to a "destination" node, this problem can be defined as an intersection O-D flow optimization problem as follows.

AN INTERSECTION O-D FLOW OPTIMIZATION PROBLEM

For a four-leg intersection, the intersection O-D flow optimization problem may be briefly described as follows: given all the inbound traffic flow rates (from origin nodes) and outbound traffic flow rates (to destination nodes) of the intersection subnetwork, the problem is to find an O-D flow pattern that minimizes the number of traffic crossing points between the traffic movements with a positive O-D flow rate.

Let us use an example to illustrate the problem configuration. As shown in Figure 3, a typical four-leg intersection is represented by a network with 8 nodes and 12 arcs. Each node represents either a traffic supply point (i.e., origin node) or a traffic demand point (i.e., destination node). In Figure 3, nodes 1, 3, 5 and 7 are origin nodes and nodes 2, 4, 6 and 8 are destination nodes. Each arc connecting an origin node and a destination node represents a feasible traffic movement. For example, in Figure 3, arc 1→2 emanates from node 1 (origin node) to node 2 (destination node), which means that a positive traffic flow rate is allowed from node 1 to node 2. It is readily seen that in the four-leg intersection case, there are three outgoing arcs for each origin node while there are three incoming arcs for each destination node.

Many traffic movements potentially cross each other in the intersection. Arc 1→2, for example, which is a left-turn movement, potentially crosses arcs 3→6, 7→8, 3→4, and 5→8, if all these traffic movements are allowed. As is well known, a right-turn movement does not cause any crossing point, e.g., arc 1→6. The objective of this intersection O-D flow optimization problem is to find an optimal traffic movement configuration that minimizes the number of crossing points caused by left-turn and through movements, subject to the traffic supply and demand requirements at origins and destinations, respectively. By using the notation shown in Figure 3, the problem formulation can be written as:

$$\min \quad z(\mathbf{y}) = \sum_{ij,mn} \left(y_{ij} + y_{mn} - 1\right)^{+} \tag{1.1}$$

where $\left(y_{ij} + y_{mn} - 1\right)^{+} = \max\left(0, y_{ij} + y_{mn} - 1\right)$

$$\text{subject to} \quad y_{ij}, y_{mn} \in \{0, 1\} \qquad \forall i \to j, m \to n \tag{1.2}$$

$$x_{ij} \le u_{ij} x_{ij}, x_{mn} \le u_{mn} x_{mn} \qquad \forall i \to j, m \to n \tag{1.3}$$

$$x_{ij}, x_{mn} \geq 0 \qquad \forall i \to j, m \to n \tag{1.4}$$

$$\sum_{i \in S_i} x_{ij} - b_j = 0 \qquad \forall j \tag{1.5}$$

$$\sum_{n \in R_m} x_{mn} - b_m = 0 \qquad \forall m \tag{1.6}$$

In the above mixed linear integer programming model, there are two sets of decision variables, the arc variables, y_{ij} (or y_{mn}), indicating the connectivity between a supply node i (or m) and a demand node j (or n) in the intersection subnetwork, and the flow variables, x_{ij} (or x_{mn}), represents the traffic flow rate on arc $i \to j$ (or $m \to n$). In the capacity constraint (i.e, constraint (1.3)), the "capacity" u_{ij} (or u_{mn}) does not impose an upper bound on x_{ij} (or x_{mn}) indeed, but appears merely as a sufficiently large number so as to represent the following arc-flow relationship: if $y_{ij} = 1$ (or $y_{mn} = 1$), $x_{ij} \geq 0$ (or $x_{mn} \geq 0$); if $y_{ij} = 0$ (or $y_{mn} = 0$), $x_{ij} = 0$ (or $x_{mn} = 0$). In the flow conservation constraints (i.e., constraints (1.5) and (1.6)), b_j and b_m are the input of the model, and S_j and T_m respectively represent the set containing the origin nodes of all the intersection arcs pointing to destination node j and the set containing the destination nodes of all the arcs emanating from origin node m, e.g., in Figure 3, $S_2 = \{1, 7, 5\}$ and $T_3 = \{4, 6, 8\}$.

Solving the intersection O-D optimization subproblem is indeed a local network design problem and a traffic reassignment process for the intersection subnetwork. Such a subnetwork change is certainly a change to the expanded network. This change, however, will not cause a change of the traffic flow pattern obtained from a traffic assignment process on the standard network in our case. In other words, the traffic flow pattern obtained from the standard network can still be maintained in the expanded network with the intersection crossing reduction/optimization. This conclusion holds subject to a *homogeneous flow* requirement that is satisfied by two modeling settings defined in our evacuation network design problem. This requirement is a sufficient (but perhaps not necessary) condition to the conclusion.

The first setting is that the underlying traffic assignment algorithm used for generating the traffic flow pattern implies the Markovian routing behavior that any individual would choose his or her remaining route to the destination without considering the route he or she has experienced between the origin and his or her current location. The resulting traffic flow pattern possesses the property that the traffic flow arriving at any intermediate node in a network is assigned as if this node is a destination. Many traffic assignment algorithms imply this Markovian routing property, including the classic all-or-nothing method for uncongested networks, the Frank-Wolfe algorithm (*24*) for deterministic user-equilibrium networks, Dial's algorithm (*25*) and Akamatsu's algorithm (*26*) for logit-based stochastic user equilibrium networks, and so on. In Xie and Turnquist (*9*, *10*), an analytical network loading algorithm based on Clark's approximation (*27*, *28*) is employed to approximate the probit-based stochastic user-equilibrium traffic flow pattern. The underlying individual route choice behavior within this approximation procedure also possesses the Markovian routing feature, which virtually assures the traffic flow merging at any intermediate node is *homogeneous by origin*.

The second setting is the one-destination network representation, which has been widely used by many researchers to model the integrated route and destination choice behavior in evacuation networks (see, for example, *29*-*32*, *7*). An immediate result from this setting is that all individuals departing from or arriving at any single source or intermediate node in the network go to the same destination. From a modeling perspective, this result guarantees that all individuals going through a node are in a homogeneous population with a single route choice function (that implies an identical route choice probability distribution with each individual at any intersection). Note that in a general multi-commodity network (i.e., a network with multiple origins and destinations), the intersection O-D flow optimization process may change the paths of traffic flows going through the intersection, and so possibly the destinations of these path flows. The occurrence of a destination change would possibly result in an infeasible traffic flow pattern. (By infeasible, we mean that the resulting traffic flow pattern caused by the intersection O-D flow optimization process may not satisfy the flow conservation constraints.) However, this phenomenon will not occur in a network with the one-destination setting; or, in other words, the traffic flow diverging at any intermediate node is *homogeneous by destination*.

As a result, from the two settings, we can conclude that the traffic flow between any intermediate node and the destination mode (in the standard network) can be regarded as a homogeneous flow pattern as if it is assigned between these two nodes. As long as the (arc-based) traffic flow pattern holds, any individual's Markovian route choice behavior would not be changed.

This intersection O-D flow optimization problem could be efficiently solved using some traditional integer programming methods, such as the branch-and-bound algorithm, due to its relatively small solution space. In the case of a four-leg intersection subnetwork, it has only 8 binary integer variables and 8 real variables with 8 capacity constraints and 8 flow conservation constraints. This algorithm embeds a vertex-and-branch tree structure in its search process, where the linear relaxation subproblem at each vertex is used to establish the lower bound for the feasible region corresponding to the vertex. Two simple algorithmic choices may be applied to accelerate the branch-and-bound search for this mixed integer program. To see these, once again, let us refer to Figure 3. We can observe that, for example, first, if $b_2 = 0$, we immediately have $x_{12} = 0$, $x_{52} = 0$ and $x_{72} = 0$, and $y_{12} = 0$, $y_{52} = 0$ and $y_{72} = 0$; second, if $x_{12} > 0$, assign as much flow to x_{52} as possible, where x_{52} is the flow rate on the right-turn arc 5→2 arriving at node 2, since a right turn would not cause any crossing conflict. Application of these simple rules at the beginning of a branch-and-bound search can effectively reduce the remaining search space.

However, we consider a more efficient solution algorithm here, whose search process mimics the pivot move of the classic simplex method for linear programming problems. To a single intersection O-D flow optimization problem presented here, the computational cost saving from the application of this simplex-based algorithm (compared to the branch-and-bound method) is quite trivial, due to the small size of the problem. However, considering this problem is a subproblem of the evacuation network design problem with lane reversal and crossing elimination and needs to be solved repeatedly during the solution search process (i.e., there are a large number of intersections in a network and the objective functions of the relaxed Lagrangian problem needs to be evaluated in a large number of times), the algorithm efficiency has a major impact on the computational cost of the overall network optimization process.

A SIMPLEX-BASED SOLUTION METHOD

Note that the flow conservation constraints of this problem have a special structure analogous to the Hitchcock-Koopmans transportation problem (see *33*). Specifically, given a set of supply nodes and demand nodes, the problem is to find a feasible "transportation flow pattern" between the supply and demand nodes, satisfying all the supply and demand requirements. This connection can be seen by setting origin nodes 1, 3, 5 and 7 as the supply nodes and destination nodes 2, 4, 6 and 8 as the demand nodes as well as constraint (1.5) as a demand constraint and constraint (1.6) as a supply constraint. Also, we can conveniently represent the supply and demand constraints into the so-called "transportation tableau", as shown in Figure 3, in which rows represent the supply nodes 1, 3, 5 and 7, columns represent the demand nodes 2, 4, 6 and 8, and the cell in row 1 and column 2, for example, represents flow variable x_{12}. If no flow is allowed between a supply node and a demand node, the cell in the corresponding row and column is illustrated as a shaded block. Moreover, for each supply node, the supply flow rate is indicated on the right of the corresponding row; for each demand node, the demand flow rate is indicated on the bottom of the corresponding column. The difference from the intersection O-D flow optimization problem to the transportation problem is also obvious: the intersection optimization model has its extra integer requirement and its objective function is nonlinear and integral.

It is well known that the transportation problem can be efficiently solved by the simplex method, which starts from a basic feasible solution and iteratively improve its objective function value by updating the current solution from one basic feasible point to another until the optimal solution is found. A basic feasible solution of the transportation problem can be conveniently represented by a rooted spanning tree in its transportation tableau (see Figure 4), which contains exactly $s + d - 1$ basic variables, where s and d are respectively the numbers of supply and demand nodes.

Despite the added complexity from our intersection subnetwork optimization problem, its structural similarity to the transportation problem inspired us to devise an efficient simplex-based iterative solution procedure, which can guarantee the optimality for the intersection optimization problem after a limited number of steps. The rationale behind this simplex-based algorithm emerges from the facts listed below.

For the sake of discussion convenience, we define the following terms in describing the intersection optimization problem. Given $\mathbf{x} = (\cdots, x_{ij}, \cdots)$ and $\mathbf{y} = (\cdots, y_{ij}, \cdots)$, we call a solution $(\mathbf{x}, \mathbf{y})$ a basic feasible solution to the defined problem if $\mathbf{x}$ is a basic feasible solution in the feasible region for the arc flows (i.e., constraints (1.4)-(1.6)) and $\mathbf{y}$ is feasible. The set of all basic variables in a basic feasible solution is called the basis. Given a basic feasible solution, another basic feasible solution is called its neighbor if it can be reached by exchanging a pair of basic variables between the two solutions. All such neighboring solutions to this solution constitute its neighborhood. We also define $N(\mathbf{x})$ as the number of nonzero flow variables in solution $(\mathbf{x}, \mathbf{y})$. It is obvious that $N(\mathbf{x}) \le s + d - 1$ if $(\mathbf{x}, \mathbf{y})$ is a basic feasible solution of the defined problem, where $s = 4$ and $d = 4$.

Lemma 1. If a solution $(\mathbf{x}^*, \mathbf{y}^*)$ to the defined intersection optimization problem is optimal, it is a basic feasible solution; otherwise, an alternative basic feasible optimal solution exists.

Proof. Let us assume that $(\mathbf{x}^*, \mathbf{y}^*)$ is not a basic feasible solution. By definition, this means either $\mathbf{y}^*$ is not feasible, $\mathbf{x}^*$ is not feasible, or $\mathbf{x}^*$ is not basic. It is manifest that either the condition that $\mathbf{x}^*$ or $\mathbf{y}^*$ is not feasible contradicts the assumption given by the lemma, therefore, $\mathbf{x}^*$ and $\mathbf{y}^*$ must be feasible.

If $\mathbf{x}^*$ is not basic while $\mathbf{x}^*$ and $\mathbf{y}^*$ are both feasible, it implies that $N(\mathbf{x}) > s + d - 1$. It reflects in the tableau that there is at least one cycle on which all the corner cells are with positive flow variables. We may adjust the flow values in these corner cells while maintaining the flow reservation feasibility until one (or more) variable, say x_{ij}, reaches its lower bound (i.e., $x_{ij} = 0$). The flow values in other cells of the tableau are not changed. Apparently, this procedure breaks a cycle in the tableau and produces an updated solution $(\mathbf{x}', \mathbf{y}^*)$ with fewer positive flow variables, i.e., $N(\mathbf{x}) < N(\mathbf{x}^*)$. Following this flow adjustment $\mathbf{x}^* \rightarrow \mathbf{x}'$, we can make an adjustment $\mathbf{y}^* \rightarrow \mathbf{y}'$ so as to obtain a new feasible solution $(\mathbf{x}', \mathbf{y}')$ without violating the problem feasibility by setting y_{ij} from 1 to 0 since $x_{ij} = 0$.

We can do all such adjustments until $N(\mathbf{x}) \leq s + d - 1$ and $z(\mathbf{y}')$ becomes a basic feasible solution. The immediate result from this adjustment is an improvement of the objective function value, i.e., $z(\mathbf{y}^*) \rightarrow z(\mathbf{y}')$, where $z(\mathbf{y}') \leq z(\mathbf{y}^*)$. If $z(\mathbf{y}') < z(\mathbf{y}^*)$, it contradicts the assumption in the lemma that $(\mathbf{x}^*, \mathbf{y}^*)$ is an optimal solution; if $z(\mathbf{y}') = z(\mathbf{y}^*)$, then we have that $(\mathbf{x}', \mathbf{y}')$ is also optimal. Therefore, we can conclude that either $(\mathbf{x}^*, \mathbf{y}^*)$ is a basic feasible solution or $(\mathbf{x}', \mathbf{y}')$ that is basic feasible is an alternative optimal solution. ■

This conclusion provides us with a theoretical foundation to devise a method that searches for the optimal solution of the intersection optimization problem along an itinerary consisting of only its basic feasible points. The iteration between two consecutive basic feasible solutions can be realized by a pivot-move neighborhood search. To guarantee the optimality of a basic feasible solution obtained by pivot moves, we need to investigate whether a local optimal solution to its neighborhood is globally optimal. A common way to carry out this investigation is convex analysis.

We rewrite the formulated mixed linear integer programming problem into an alternative formulation as follows:

$$\min \quad z(\mathbf{x}) = \left\{ \sum_{ij,mn} \left(y_{ij} + y_{mn} - 1 \right)^+ : y_{ij}, y_{mn} \in \{0,1\}, x_{ij} \leq u_{ij} y_{ij}, x_{mn} \leq u_{mn} y_{mn}, \forall i \rightarrow j, m \rightarrow n \right\} \tag{2.1}$$

$$\text{subject to} \quad x_{ij}, x_{mn} \geq 0 \qquad \forall i \rightarrow j, m \rightarrow n \tag{2.2}$$

$$\sum_{i \in S_i} x_{ij} - b_j = 0 \qquad \forall j \tag{2.3}$$

$$\sum_{n \in R_m} x_{mn} - b_m = 0 \qquad \forall m \tag{2.4}$$

where arcs $i \to j$ and $m \to n$ are a pair of arcs which geometrically cross each other if their traffic flow rates are both positive.

This new problem formulation has the same structure as the transportation problem except for the objective function. It is readily known that the feasible region of this problem is a bounded polyhedral set. The remaining problem is the convexity property of the objective function $z(\mathbf{x})$. Let us consider $f(\lambda) = z(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)$ and $g(\lambda) = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$, given that $\mathbf{x}_1$ and $\mathbf{x}_2$ are any two feasible solutions and $0 < \lambda < 1$. It is easy to know that both $f(\lambda)$ and $g(\lambda)$ can be expressed as the sum of the following terms, respectively:

$$\begin{aligned} f(\lambda) &= z(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \\ &= \textstyle\sum_{ij,mn} \left(y'_{ij} + y'_{mn} - 1\right)^+ \end{aligned}$$

where $y_{ij}, y_{mn} \in \{0,1\}, \lambda x_{ij}^1 + (1-\lambda)x_{ij}^2 \le u_{ij} y'_{ij}, \lambda x_{mn}^1 + (1-\lambda)x_{mn}^2 \le u_{mn} y'_{mn}, \forall i \to j, m \to n$

and

$$\begin{aligned} g(\lambda) &= \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \\ &= \textstyle\sum_{ij,mn} \left[\lambda\left(y_{ij}^1 + y_{mn}^1 - 1\right)^+ + (1-\lambda)\left(y_{ij}^2 + y_{mn}^2 - 1\right)^+\right] \end{aligned}$$

where $y_{ij}^1, y_{mn}^1 \in \{0,1\}, x_{ij}^1 \le u_{ij} y_{ij}^1, x_{mn}^1 \le u_{mn} y_{mn}^1, \forall i \to j, m \to n$, and
$y_{ij}^2, y_{mn}^2 \in \{0,1\}, x_{ij}^2 \le u_{ij} y_{ij}^2, x_{mn}^2 \le u_{mn} y_{mn}^2, \forall i \to j, m \to n$

To compare the values of $f(\lambda)$ and $g(\lambda)$, consider the following four conditions: if given $x_{ij}^1 x_{mn}^1 = 0$ (i.e., either $x_{ij}^1 = 0$ or $x_{mn}^1 = 0$) and $x_{ij}^2 x_{mn}^2 = 0$ (i.e., either $x_{ij}^2 = 0$ or $x_{mn}^2 = 0$), $\left(y'_{ij} + y'_{mn} - 1\right)^+ = 0$ and $\lambda\left(y_{ij}^1 + y_{mn}^1 - 1\right)^+ + (1-\lambda)\left(y_{ij}^2 + y_{mn}^2 - 1\right)^+ = 0$; if $x_{ij}^1 x_{mn}^1 > 0$ and $x_{ij}^2 x_{mn}^2 = 0$, we obtain $\left(y'_{ij} + y'_{mn} - 1\right)^+ = 1$ and $\lambda\left(y_{ij}^1 + y_{mn}^1 - 1\right)^+ + (1-\lambda)\left(y_{ij}^2 + y_{mn}^2 - 1\right)^+ = \lambda$; if $x_{ij}^1 x_{mn}^1 = 0$ and $x_{ij}^2 x_{mn}^2 > 0$, we obtain $\left(y'_{ij} + y'_{mn} - 1\right)^+ = 1$ and $\left(y_{ij}^1 + y_{mn}^1 - 1\right)^+ + (1-\lambda)\left(y_{ij}^2 + y_{mn}^2 - 1\right)^+ = 1 - \lambda$; if $x_{ij}^1 x_{mn}^1 > 0$ and $x_{ij}^2 x_{mn}^2 > 0$, $\left(y'_{ij} + y'_{mn} - 1\right)^+ = 1$ and $\left(y_{ij}^1 + y_{mn}^1 - 1\right)^+ + (1-\lambda)\left(y_{ij}^2 + y_{mn}^2 - 1\right)^+ = \lambda + (1-\lambda) = 1$. Combining all these conditions, we know that $f(\lambda) \geq g(\lambda)$ holds for any $0 < \lambda < 1$. Therefore, $z(\mathbf{x})$ is a concave function (where, more specifically, given the integer characteristic, we know that $z(\mathbf{x})$ is a stepwise concave function).

Given that the feasible region is a convex set but the objective function is a concave function, we cannot in general guarantee the global optimality of a local optimum. However, for the defined intersection subnetwork optimization problem with its special structure, we can show that no local optimum can be actually held by a simplex-based procedure.

Lemma 2. If a basic feasible solution to the defined intersection subnetwork optimization problem is a local optimal solution to its neighborhood, it is also a global optimal solution.

Proof. We can distinguish flow variables in two types: 1) "right-turn" flow variables, which do not impose any traffic crossing points; and 2) "left-turn" and "through" flow variables, which would potentially cause crossing points. The value of the objective function is determined by the values of the "left-turn" and "through" flow variables. Suppose that x_{ij} and x_{mn} are two variables of the second type and their corresponding arcs may have a potential crossing point. The distribution of values of $\left(y_{ij} + y_{mn} - 1\right)^+$ is shown in Figure 5, in which the feasible region for x_{ij} and x_{mn} are the projection of the whole feasible region of $\mathbf{x}$ on the plane of x_{ij} and x_{mn}. Needless to say, $\left(y_{ij} + y_{mn} - 1\right)^+$ has two possible values: when either $x_{ij} = 0$ or $x_{mn} = 0$, $\left(y_{ij} + y_{mn} - 1\right)^+ = 0$, and when both $x_{ij} > 0$ and $x_{mn} > 0$, $\left(y_{ij} + y_{mn} - 1\right)^+ = 1$.

Note that for any pair of x_{ij} and x_{mn} with a potential crossing point between their arcs, its feasible region subject to constraints (1.4)-(1.6) and the corresponding value distribution of $\left(y_{ij} + y_{mn} - 1\right)^+$ can be represented by one of the conditions in Figure 5. If a local optimal solution that is not globally optimal exists, there is at least one pair of x_{ij} and x_{mn} such that there are two separate subregions both with $\left(y_{ij} + y_{mn} - 1\right)^+ = 0$ in its feasible region. However, none of the feasible regions presented above includes such a case. Therefore, a local optimal solution will not be blocked from other optimal solutions by simplex-based pivot moves and it is actually a global optimal solution. ■

The conclusion given above assures the global optimality of a simplex-based search; it, however, does not guarantee the optimality uniqueness. In fact, it is possible to have multiple optimal solutions to the defined intersection subnetwork optimization problem, in which some solutions are basic feasible solutions and others are not. But we know that at least one of the optimal solutions is a basic feasible solution.

Now we have all the required theoretical elements to guarantee the correctness of the proposed algorithm. The algorithmic procedure of the resulting simplex-based pivot-move method can be sketched as follows:

Step 1. Obtain a starting basic feasible solution as the current solution and compute its objective function value z^*. This can be accompolished by applying the northwest corner rule in the tableau (see *33*);

Step 2. Conduct all the candidate pivot moves by entering each nonbasic variable into the basis and compute the updated objective function value with each candidate move. Choose the best move with the lowest objective function value z';

Step 3. Compare the objective function value with the best move, z', and the current objective function value, z^*. If $z' \geq z^*$, stop the iteration and we have the optimal solution z^* at hand; if $z' < z^*$, implement the best move to obtain the updated basic feasible solution and assign $z^* = z'$, and then go to step 2.

NUMERICAL EXAMPLES

For the illustration purpose, we present a couple of numerical examples of the algorithm application in this section.

The first example problem with its network and tableau representations is given in Figure 6(a). The initial basic feasible solution derived by the northwest corner rule is shown in Figure 6(b), in which the basis consists of variables x_{12}, x_{14}, x_{16}, x_{34}, x_{56}, x_{58}, and x_{74}, and the objective function value with this solution is 5. Starting from this initial solution, it is found that by examining all the nonbasic variables that a pivot move that the nonbasic variable x_{52} enters the basis and the basic variable x_{56} leaves the basis yields a best move (i.e., the lowest objective function value). By implementing this move, we get an updated basic feasible solution, the basis of which includes variables x_{12}, x_{14}, x_{16}, x_{34}, x_{52}, x_{58} and x_{74}, and the objective function value of which is 3. This updated solution is illustrated in Figure 6(c). The same examination and pivot procedure is then applied to proceed with the search for improved solutions. Next, we obtain the basic feasible solution at iteration 2 by entering x_{38} into the basis and getting rid of x_{12} from the basis, as shown in Figure 6(d), whose objective function value is 1. Since this solution cannot be improved by a single pivot move, we can conclude that it is the optimal solution to the problem.

The second example is a copy of the first one except that the values of b_2 and b_4 are swapped. The initial solution obtained by applying the northwest corner rule is shown in Figure 7(a), in which the basis consists of x_{12}, x_{34}, x_{36}, x_{38}, x_{52}, x_{58} and x_{72} and the objective function value with this solution is 7. At the first iteration, it is found that two pivot moves yields the same best objective function value (i.e., the value is 3). These two moves are respectively that x_{16} enters the basis and x_{58} leaves the basis, and x_{74} enters the basis and x_{58} leaves. Since the two pivot moves improves the objective function value by the same quantity, we can implement either of them to obtain the next basic feasible solution. For completeness, we present the basic feasible solutions resulted from both the moves respectively in Figure 7(b) and Figure 7(c). Further examinations on these two solutions conclude that both of the solutions are optimal to the problem since no pivot move that improves the objective function value can be found. This example demonstrates a case that more than one optimal solution exist at the same time.

CONCLUDING REMARKS

This text presents an intersection O-D flow optimization problem, which arises as a subproblem in the evacuation network design with lane reversal and crossing elimination. The problem's special structure analogous to the Hitchcock-Koopmans transportation problem leads to the creation of a simplex-based solution method. The method proves to be a very efficient procedure compared to traditional integer programming methods, in that in most cases it can find the optimal solution in just a few iterations. The model can be readily modified to accommodate other types of intersections and interchanges. But the solution method may deserve an in-depth reinvestigation on its applicability to other intersection/interchange cases. This remains a piece of future work to the authors. In this study, we simply use the number of traffic crossing points as the objective function of the O-D flow optimization problem and as the penalty term of the relaxed evacuation network design problem. Other network status information, for example, traffic flow rates through intersections, may be used to interpret the violation conditions of the crossing elimination constraints more accurately and favor a faster Lagrangian relaxation-based solution method.

REFERENCES

[1] Bell, M.G.H. (1985). “Variances and covariances for origin-destination flows when estimated by log-linear models.” Transportation Research, 19B(6), 497-507.

[2] Cremer, M. and Keller, H. (1987). “A new class of dynamic methods for the identification of origin-destination flows.” Transportation Research, 21B(2), 117-132.

[3] Nihan, N.L. and Davis, G.A. (1989). “Application of prediction-error minimization and maximum likelihood to estimate intersection O-D matrices from traffic counts.” Transportation Science, 23(2), 77-99.

[4] Hamed, M.M. and Nihan, N.L. (1996). “Estimating intersection O-D matrices from observed time-series of traffic counts: The Zellner estimator.” Journal of Advanced Transportation, 30(3), 57-78.

[5] Li, B. and De Moor, B. (1999). “Recursive estimation based on the equality-constrained optimization for intersection origin-destination matrices.” Transportation Research, 33B(3), 203-214.

[6] Lou, Y. and Yin, Y. (2006). “Real-time estimation of origin-destination flows for actuation-controlled intersections.” Proceedings of the IEEE ITSC 2006, IEEE Intelligent Transportation Systems Conference, Toronto, Canada, September 17-20, 2006.

[7] Xie, C. (2008). Evacuation network optimization: Models, solution methods and applications, Ph.D. Thesis, School of Civil and Environmental Engineering, Cornell University, Ithaca, NY.

[8] Kalafaras, G. and Peeta, S. (2009). “Planning for evacuation: Insights from an efficient network design model.” Journal of Infrastructure Systems, 15(1), 21-30.

[9] Xie, C. and Turnquist, M.A. (2009). “Integrated evacuation network optimization and emergency vehicle assignment.” Transportation Research Record, 2091, 79-90.

[10] Xie, C. and Turnquist, M.A. (2010). “Lane-based evacuation network optimization: An integrated Lagrangian relaxation and tabu search approach.” Transportation Research. (In press)

[11] Xie, C., Lin, D.Y. and Waller, S.T. (2010). “A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies.” Transportation Research, 46E(3), 295-316.

[12] MacDorman, L.C. (1965). “Case study in sensitivity of highway economic factors.” Highway Research Record, 100, 2-19.

[13] Glickman, T.S. (1970). “Optimal periodic control of reversible traffic operations.” Ph.D. Thesis, Department of Civil Engineering, Johns Hopkins University, Baltimore, MD.

[14] Hemphill, J. and Surti, V.H. (1974). “A feasibility study of a reversible-lane facility for a Denver street corridor.” Transportation Research Record, 1974, 29-32.

[15] Caudill, R.J. and Kuo, N.M. (1983). "Development of an interactive planning model for contraflow lane evaluation." Transportation Research Record, 906, 47-54.

[16] Wolshon, P.B. and Lambert, L. (2004). Convertible Roadways and Lanes: A Synthesis of Highway Practice. NCHRP Synthesis 340, National Cooperative Highway Research Program, Transportation Research Board, Washington, DC.

[17] Urbina, E. and Wolshon, B. (2003). "National review of hurricane evacuation plans and policies: A comparison and contrast of state practices." Transportation Research, 37A(3), 257-275.

[18] Hamza-Lup, G.L., Hua, K.A., Le, M. and Peng, R. (2004). "Enhancing intelligent transportation systems to improve and support homeland security." Proceedings of the 7th IEEE Intelligent Transportation Systems Conference, Washington, DC, October 3-6, 2004.

[19] Hamza-Lup, G.L., Hua, K.A. and Peng, R. (2007). "Leveraging e-transportation in real-time traffic evacuation management." Electronic Commerce Research and Applications, 6(4), 413-424.

[20] Tuydes, H. and Ziliaskopoulos, A.K. (2006). "Tabu-based heuristic approach for optimization of network evacuation contraflow." Transportation Research Record, 1964, 157-168.

[21] Kim, S., Shekhar, S. and Min, M. (2008). "Contraflow transportation network reconfiguration for evacuation route planning." Transactions on Knowledge and Data Engineering, 20(8), 1115-1129.

[22] Meng, Q., Khoo, H.L. and Cheu, R.L. (2008). "Microscopic traffic simulation model-based optimization approach for the contraflow lane configuration problem." Journal of Transportation Engineering, 134(1), 41-49.

[23] Cova, T.J. and Johnson, J.P. (2003). "A network flow model for lane-based evacuation routing," Transportation Research, 37A(7), 579-604.

[24] LeBlanc, L.J., Morlok, E.K. and Pierskalla, W.P. (1975). "An efficient approach to solving the road network equilibrium traffic assignment problem." Transportation Research, 9(5), 309-318.

[25] Dial, R.B. (1971). "A probabilistic multipath traffic assignment problem which obviates path enumeration." Transportation Research, 5(2), 83-111.

[26] Akamatsu, T. (1996). "Cyclic flows, Markov process and stochastic traffic assignment." Transportation Research, 30B(5), 369-386.

[27] Maher, M.J. (1992). "SAM—A stochastic assignment model." Mathematics in Transport Planning and Control, J.D. Griffiths, ed., Oxford University Press, Oxford, England, UK.

[28] Maher, M.J. and Hughes, P.C. (1997). "A probit-based stochastic user equilibrium assignment model," Transportation Research, 31B(4), 341-355.

[29] Sherali, H.D., Carter, T.B. and Hobeika, A.G. (1991). "A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions." Transportation Research, 25B(6), 439-452.

[30] Yuan, F., Han, L.D., Chin, S.M. and Hwang, H. (2006). "Proposed framework for simultaneous optimization of evacuation traffic destination and route assignment." Transportation Research Record, 1964, 50-58.

[31] Chiu, Y.C. and Zheng, H. (2007). "Real-time mobilization decisions for multi-priority emergency response resources and evacuation groups: Model formulation and solution." Transportation Research, 43E(6), 710-736.

[32] Yao, T., Mandala, S.R. and Chung, B.D. (2009). "Evacuation transportation planning under uncertainty: A robust optimization approach." Networks and Spatial Economics, 9(2), 171-189.

[33] Bazaraa, M.S., Jarvis, J.J. and Sherali, H.D. (1990). Linear Programming and Network Flows. John Wiley & Sons, New York, NY.

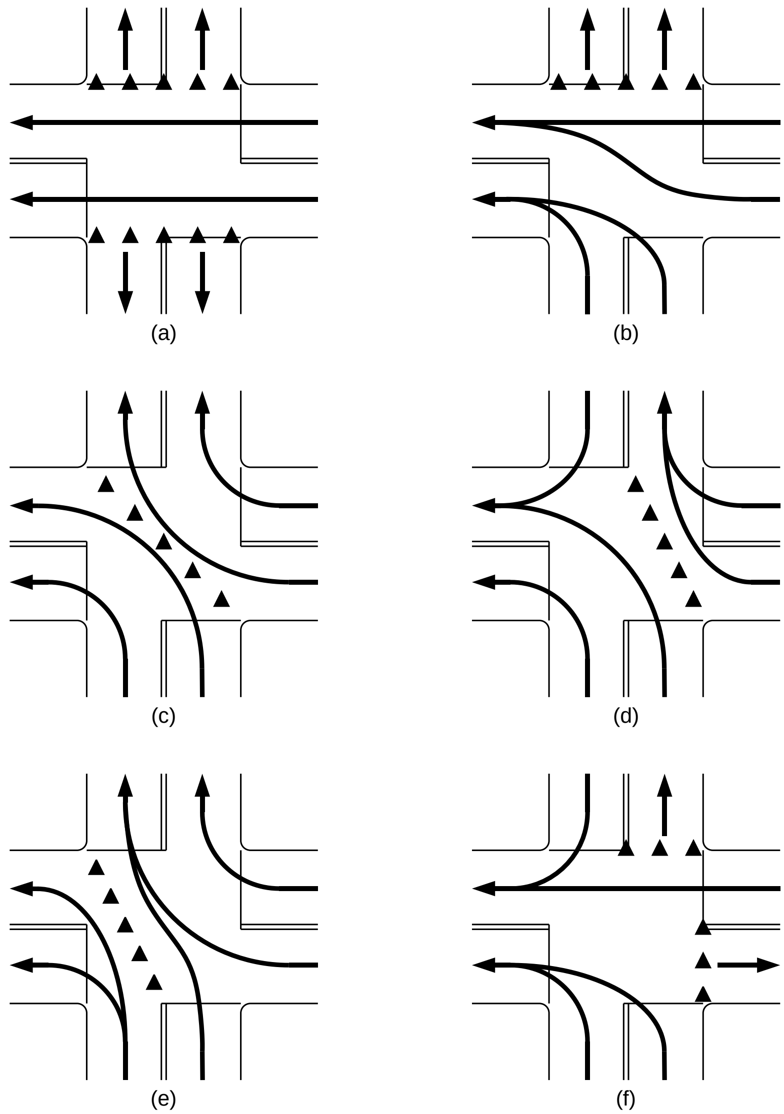
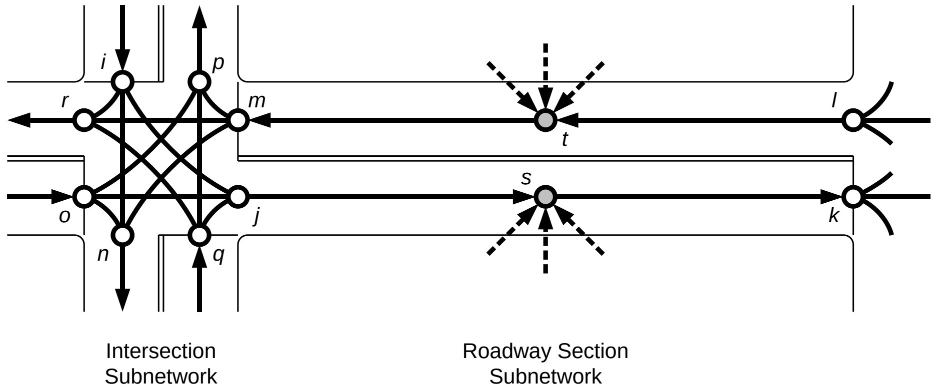
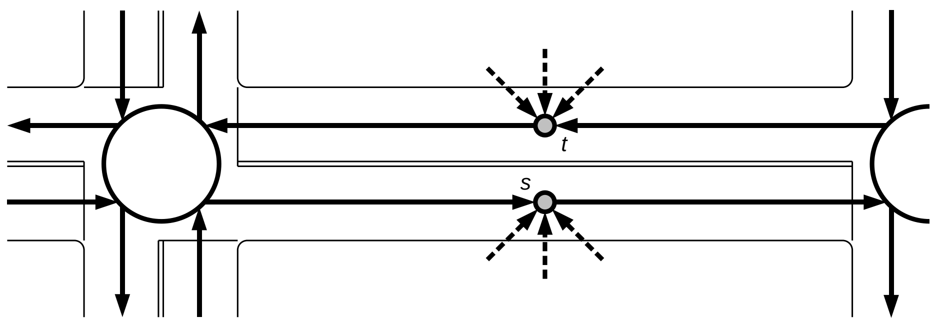


Figure 1 Examples of the joint use of lane reversal and crossing elimination

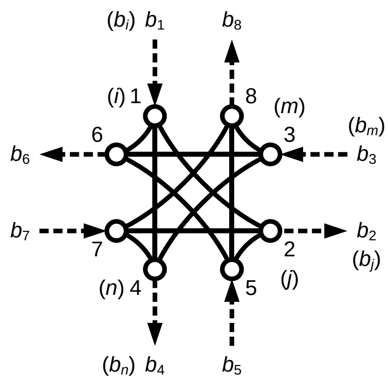


(a) The expanded network representation



(b) The standard network representation

Figure 2 Node-arc network representations

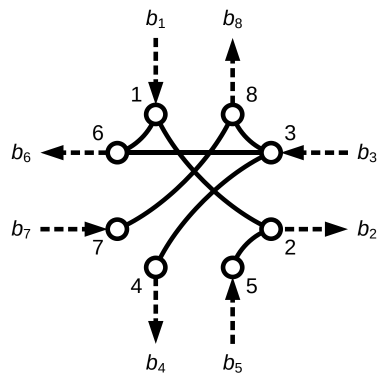


(a) A four-leg intersection subnetwork

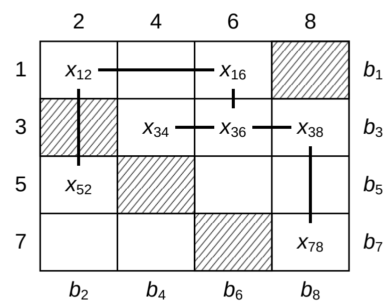
	2	4	6	8	
1	x_{12}	x_{14}	x_{16}		b_1
3		x_{34}	x_{36}	x_{38}	b_3
5	x_{52}		x_{56}	x_{58}	b_5
7	x_{72}	x_{74}		x_{78}	b_7
	b_2	b_4	b_6	b_8	

(b) The tableau representation of the four-leg intersection subnetwork

Figure 3 A four-leg intersection subnetwork and its corresponding transportation tableau

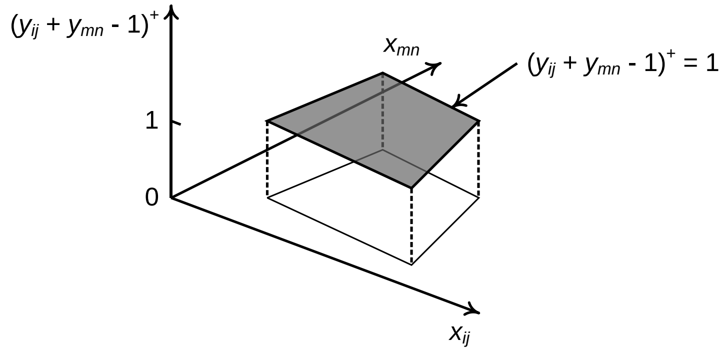


(a) The network representation of a basic feasible solution

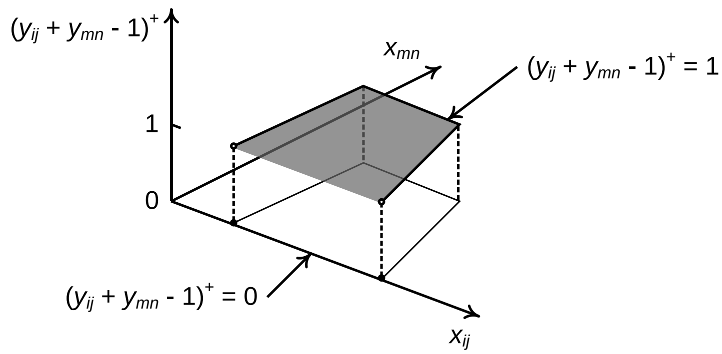


(b) The tableau representation of a basic feasible solution

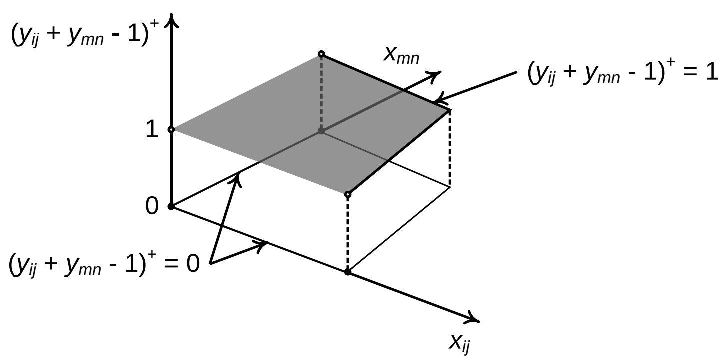
Figure 4 Representation of a basic feasible solution in the network and the tableau



(a) $x_{ij} > 0$ and $x_{mn} > 0$

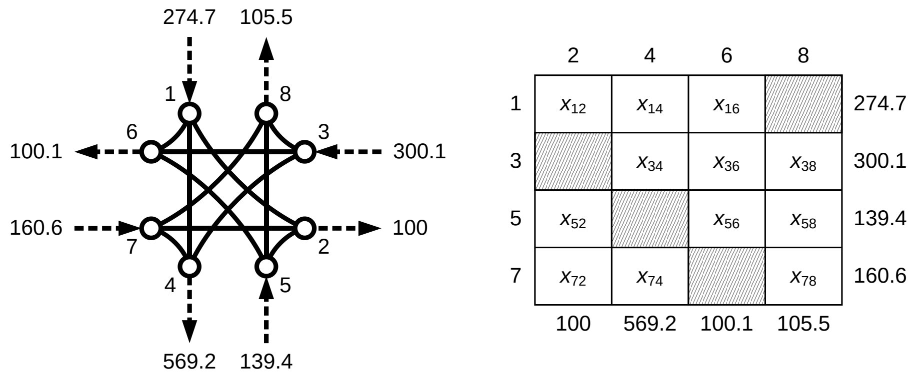


(b) $x_{ij} > 0$ and $x_{mn} \geq 0$

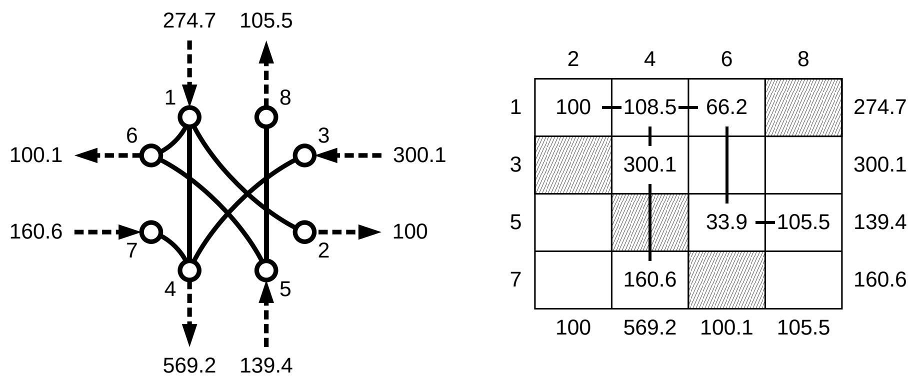


(c) $x_{ij} \geq 0$ and $x_{mn} \geq 0$

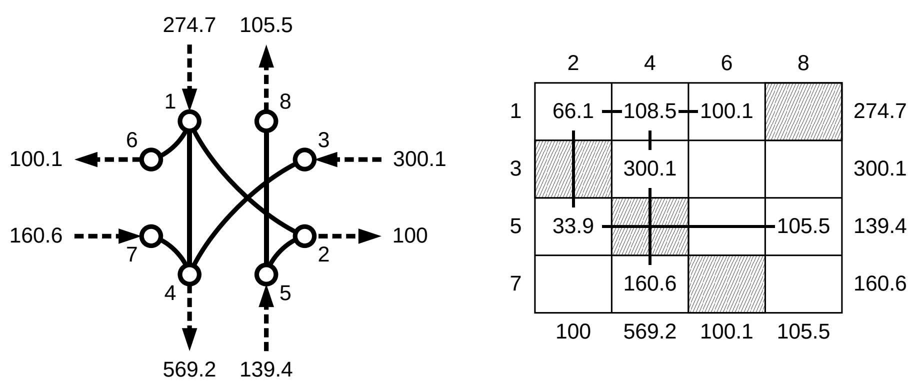
Figure 5 Feasible region of a pair of flow variables x_{ij} and x_{mn} with a potential crossing point and the corresponding $\left(y_{ij} + y_{mn} - 1\right)^+$ value



(a) The network and tableau representations of the problem

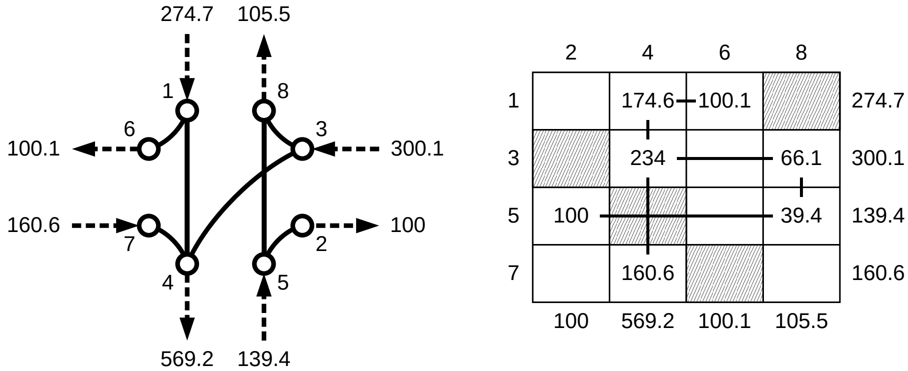


(b) Iteration 0 (Objective function value: 5)



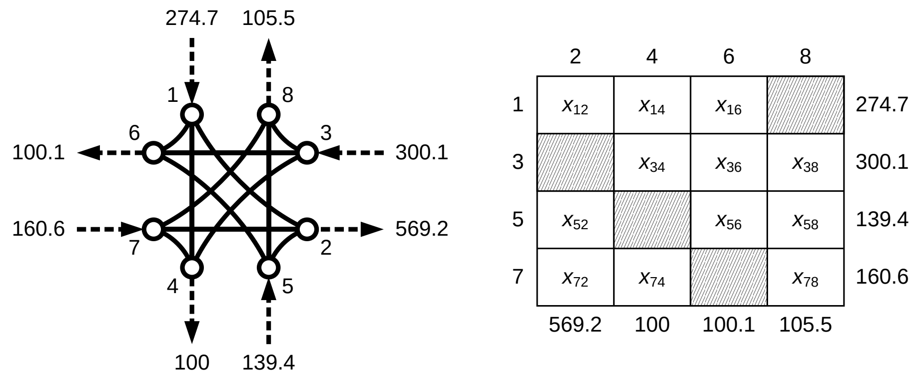
(c) Iteration 1 (Objective function value: 3)

Figure 6 The first numerical example and its solutions by the simplex-based method

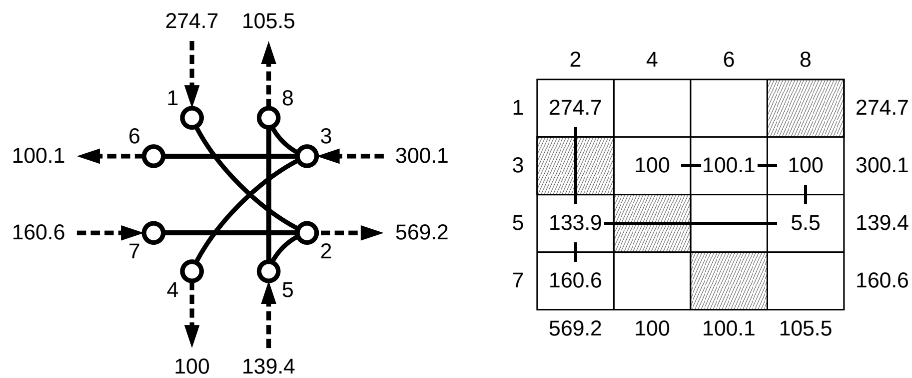


(d) Iteration 2 (Objective function value: 1)

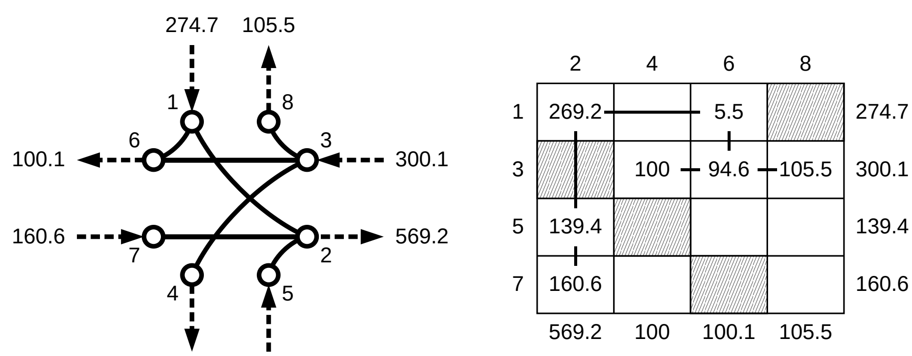
Figure 6 (Continued)



(a) The network and tableau representations of the problem

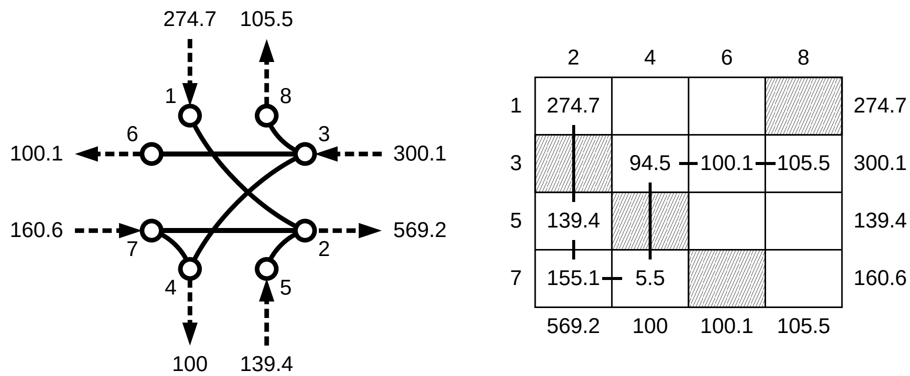


(b) Iteration 0 (Objective function value: 7)



(c) Iteration 1 (Objective function value: 3)

Figure 7 The second numerical example and its solutions by the simplex-based method



(d) Iteration 1 (Objective function value: 3)

Figure 7 (Continued)