Basic Static Assignment to Transportation Networks

Uncongested Networks

Introduction

Assignment to uncongested networks is based on the assumption that costs do not depend on flows. In other words, path flows, and thus link flows, are obtained from path choice probabilities that are themselves computed from flow-independent link performance attributes and costs. Uncongested assignment models are used for the analysis of relatively uncongested road transportation systems (generally, link cost functions are almost flat with respect to flows for flow-capacity ratios up to values around (0.50–0.70). They are also often used for analyzing public transport systems, for which costs may be assumed independent of link passenger flows if the available capacity is sufficient. Furthermore, uncongested network assignment models are a key component of congested network assignment models, which are described in the following sections.

UNcongested network (UN) assignment models are defined by the demand model, expressing path flows as a function of path costs and demand flows:

$$h_{UN,od} = h_{UN,od}(g_{od}; d_{od}) = d_{od} P_{od}(-g_{od}) \forall od$$

$$h_{UN} = h_{UN}(g; d) = P(-g)d \qquad 1$$

The path costs g can be obtained from the link costs c with, and the link flows f corresponding to the path flows h is given. Figure 1 depicts these relationships graphically, applying the framework in Fig.1 to the case of uncongested network assignment.

General uncongested network assignment models can also be expressed in terms of link variables by combining (1). The result is called the uncongested network assignment map, which associates a link flow vector with each demand flow vector and link cost vector, and can be expressed in an aggregate or disaggregate way as:

$$f_{UN} = f_{UN}(c; d) = \sum_{od} d_{od} \Delta_{od}(-\Delta_{od}^T c - g_{od}^{NA}) \ \forall c$$

$$f_{UN} = f_{UN}(c; d) = \Delta P(-\Delta^T c - g^{NA})d \quad \forall c \qquad 2$$

Note that link flows depend nonlinearly on the link costs, but linearly on the demand flows so that the effect of each O-D pair can be evaluated separately. In the next sections, probabilistic and deterministic path choice models, which lead respectively to stochastic and deterministic uncongested network assignment models and algorithms, are considered in turn.

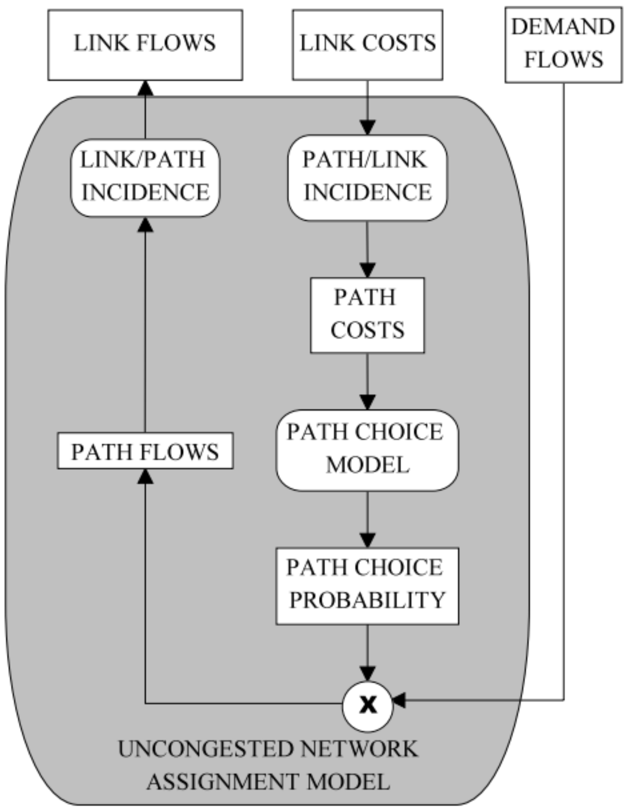

Fig. 1 Schematic representation of uncongested network assignment models.

The path costs g can be obtained from the link costs c with (1), and the link flows f corresponding to the path flows h. Figure 1 depicts these relationships graphically.

General uncongested network assignment models can also be expressed in terms of link variables by combining (1) with (1) in (lecture, 5) and (3). The result is called the uncongested network assignment map, which associates a link flow vector with each demand flow vector and link cost vector, and can be expressed in an aggregate or disaggregate way as:

$$f_{UN} = f_{UN}(c; d) = \sum_{od} d_{od} \Delta_{od} p_{od}(-\Delta_{od}^{cT} - g_{od}^{NA}) \qquad \forall c$$

$$f_{UN} = f_{UN}(c; d) = \Delta P(-\Delta^{T} c - g^{NA}) d \qquad \forall c \qquad (2)$$

Note that link flows depend nonlinearly on the link costs, but linearly on the demand flows so that the effect of each O-D pair can be evaluated separately. In the next sections, probabilistic and deterministic path choice models, which lead respectively to stochastic and deterministic uncongested network assignment models and algorithms, are considered in turn.

Models for Stochastic Assignment

If path choice behavior is simulated through a probabilistic random utility model, the resulting assignment model is known as a Stochastic UNcongested network (SUN) assignment. In this case, the resulting link or path flows correspond to a situation in which, for each O-D pair, the perceived cost of the used paths is less than or equal to the cost of every other path; this can be viewed as a generalization of Wardrop's first principle. Using the probabilistic path choice models, recall that each vector of link and path costs determines a unique choice probability vector. Hence the uncongested assignment map is given by the stochastic uncongested assignment function, f SUN(c; d). This function is a one-to-one correspondence that, for a given vector of link costs c, outputs a vector of link flows f belonging to the nonempty, compact, and convex set of feasible link flows (Fig. 2):

$$f_{SUN} = f_{SUN}(c,d) = \sum_{od} d_{od} \Delta_{od} p_{od}(-\Delta_{od}^{T} c - f_{od}^{NA}) \in S_f \qquad \forall c \qquad (3)$$

Apart from the demand vector, the parameters of the stochastic uncongested assignment function include those of the path choice model (such as the coefficients of the systematic utility and the variance of the random residuals), and those of the supply model (such as travel times and generalized costs, together with the graph topology). Under certain assumptions on the path choice function, function (3) has features that will be useful in the analysis of stochastic equilibrium assignment models.

Variance and covariance of link and path flow, are considered random variables. Assuming probabilistic path choice behavior (with known demand flows d_{od}) and independent user choices, the path flows h_{od} can be considered as realizations of multinomial random variables H_{od}. The values h_{od} calculated with the stochastic uncongested network assignment model represents the means of H_{od}, for the most general case of demand models involving all choice dimensions. Therefore, the mean, variance, and covariance of the elements of the path flow random vector H can be expressed as:

$$E[H_k] = h_{SUN,k} = d_{od} p_{od,k} \quad \forall od, k$$

$$Var[H_k] = d_{od} p_{od,k}\left(1 - p_{od,k}\right) \; \forall od, k$$

$$Cov[H_k, H_j] \begin{cases} -d_{od} p_{od,k} P_{od,j} & k, j \in K_{od} \\ o & otherwise \end{cases} \qquad \forall od, k, j$$

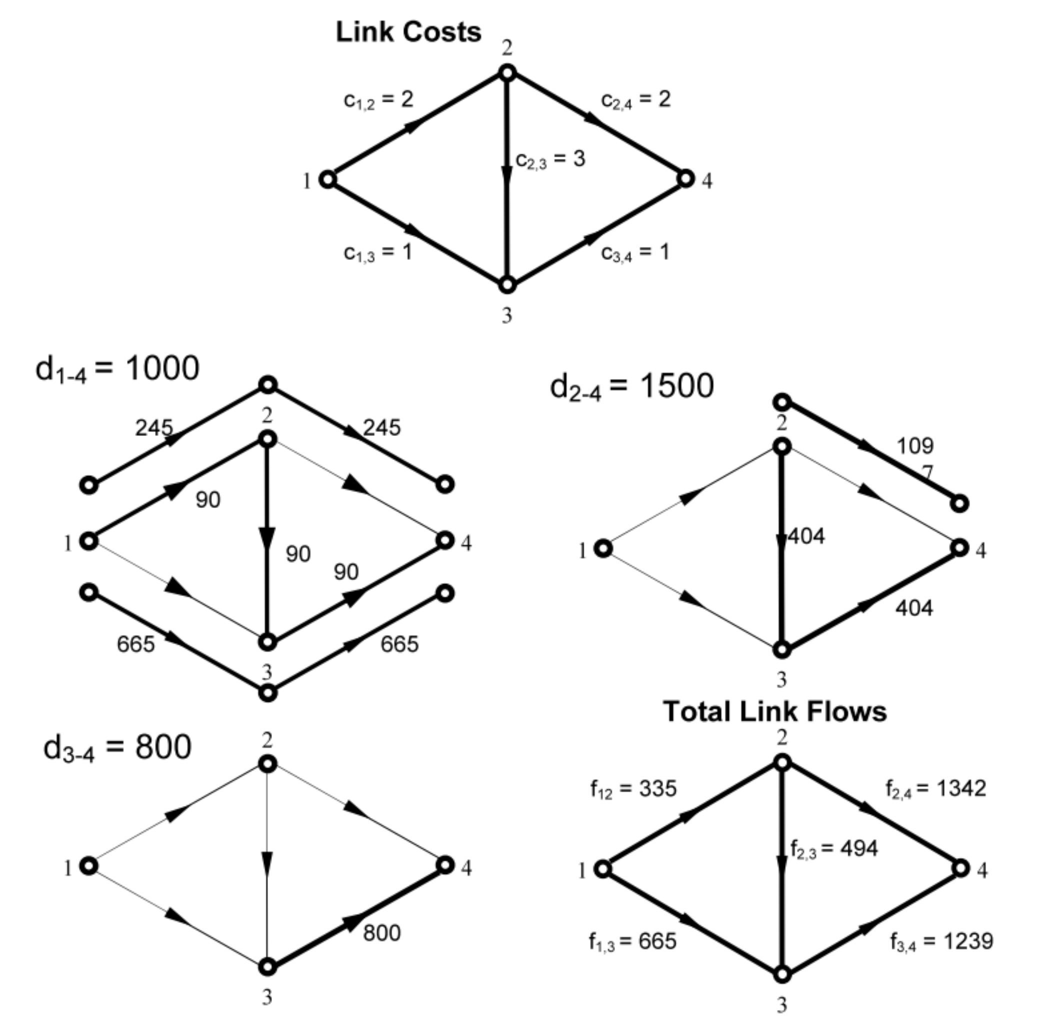

Fig. 2 Stochastic UNcongested network (SUN) assignment with the path choice model.

The first equation expresses the elements of the mean vector $h_{SUN} = E[H]$ of random vector H , and the last two equations give the elements of its variance-covariance matrix ΣH . If the path flow vector $h = [h_{od}]_{od}$ is considered to be a realization of the random vector H , then the link flow vector $f = \Delta h$, obtained from h by a linear transformation, is a realization of a link flow random vector F. Thus the mean vector and variance-covariance matrix of random vector F can be expressed in terms of the corresponding values of the path flow random variable, $h_{SUN} = E[H]$ and ΣH . In fact $E[F] = \Delta E[H] = \Delta h_{SUN} = f_{SUN}$ and $\Sigma F = \Delta T \ \Sigma H \Delta$.

Assignment function computation. The link flow vector defined by the stochastic uncongested assignment function for a given link cost vector can easily be calculated when explicit path enumeration can be carried out.

When paths are explicitly enumerated, path costs can be easily computed from link costs by applying the link–path incidence relationship (1) in (lecture, 5). Nonadditive costs can be easily handled. Similarly, path flows can be obtained by applying the demand model and its extensions, and link flows can be computed from path flows using the congruence relationship. Eventually, EMPU, given by

$S_{od} = S_{od}\,(\Delta_{od}^{T} c - g_{od}^{NA})$,

This is related to the path choice alternatives available for O-D pair *od*, which can also be readily calculated.

It should be recalled that, for probit path choice models, it is not possible analytically to calculate choice probabilities or to evaluate the demand model. Nonetheless, unbiased estimates of path choice probabilities and of the corresponding path flows can be obtained in the probit case by applying a Monte Carlo sampling technique. The method generates a random vector realization, where each component of the vector is considered the perceived cost random residual of an O-D path. The corresponding path perceived cost is computed by adding the path systematic cost to the residual. The perceived costs of all O-D paths are computed in this way. For each O-D pair, the demand flow is assigned to the path with the minimum perceived cost. These steps are repeated for each of sample of *m* random vector realizations, and the resulting path flows are averaged. These averages are unbiased estimates of the stochastic uncongested network path flows:

$$\bar{h}^{m} = \sum_{j=1,m} h^{j/m}$$

Where:

$h^{j} = h_{SPA}(g + \varepsilon^{j})$ is the vector of path flows obtained by assigning the demand flow of each O-D pair to the shortest path w.r.t. the perceived path costs $g + \varepsilon^{j}$

g is the vector of systematic path costs

$\varepsilon\, j \leftarrow MVN(0,\Sigma)$ is the *j* th (in a sample of *m*) perceived path cost random residual vector; in probit path choice, ε^{j} is obtained as a realization of a multivariate normal random variable with zero mean and variance-covariance matrix Σ

h^{m} is an unbiased estimate of the SUN assignment path flow vector, obtained from a sample of *m* perceived path cost vectors.

Moreover, the average perceived shortest path cost, computed with respect to the paths that connect an O-D pair, is an unbiased estimate of EMPU associated with the O-D pair path choice alternatives.
In practice, the path flow estimate $\bar{h}^{m}$ can be obtained by evaluating the following recursive equations up to j = m, starting with j = 0 and $\bar{h}^{0} = 0$:

$$j = j + 1$$
$$\varepsilon^{j} \leftarrow MVN(0, \Sigma)$$
$$\bar{h}^{j} = \left((j-1)\bar{h}^{j-1} + h^{j}\right)/j$$

In applications, direct use of this approach can be computationally burdensome because of the need to generate multiple realizations of a multivariate normal random variable with nonzero covariance, $\varepsilon^j \leftarrow MVN(0, \Sigma)$. On the other hand, the method allows arbitrary covariance structures (due, e.g., to positive or negative correlations between the perceived cost random residuals of different links). When this generality is not required, it is convenient to generate perceived path costs from link costs.

Models for Deterministic Assignment

Under the assumption of deterministic path choice behavior, the demand flow of each O-D pair is assigned to the minimum cost path(s) (i.e., paths with maximum systematic utility), whereas no flow is assigned to other paths. For this reason, the Deterministic UNcongested network (DUN) assignment is also known as an all-or-nothing assignment. In general, as has already been noted, multiple path choice probability vectors may correspond to a single vector of link and path costs. It follows that the general uncongested network assignment relationship (2) must be specified as the deterministic uncongested network assignment map:

$h_{DUN} = h_{DUN}(g; d) \quad \in S_h,$

Which is a one-to-many (or point-to-set) map between path costs and flows. In other words, because there may be several alternative minimum cost paths connecting an origin to a destination, a given path and link cost vector may correspond to multiple vectors of deterministic uncongested network paths and link flows. Consequently, the study of the properties of deterministic network loading frequently uses indirect formulations, equivalent to (2), based on the formulation of the deterministic demand model as a system of inequalities (3b) in (lecture 6). Summing the inequalities overall O-D pairs yields expression (4):

$$g^T(h - h_{DUN}) \geq 0 \; \forall h \in S_h \qquad 4$$

The resultant path (or link) flows satisfy Wardrop's first principle. Figure 3 presents an example of the deterministic uncongested network assignment model.

If nonadditive path costs are zero, $g^{NA} = 0$, total path costs coincide with additive costs $g^T = (g^{ADD})^T = c^T \Delta$, and it is easy to verify that (4) is equivalent to:

$$c^T(f - f_{DUN}) \geq 0 \quad \forall f \in S_f \qquad 5$$

On the other hand, when there is nonadditive path costs expression (4) is equivalent to:

$$c^T(f - f_{DUN}) + (g^{NA})^T(h - h_{DUN}) \geq 0 \; \forall f = \Delta h, \; \forall h \in S_h \qquad 6$$

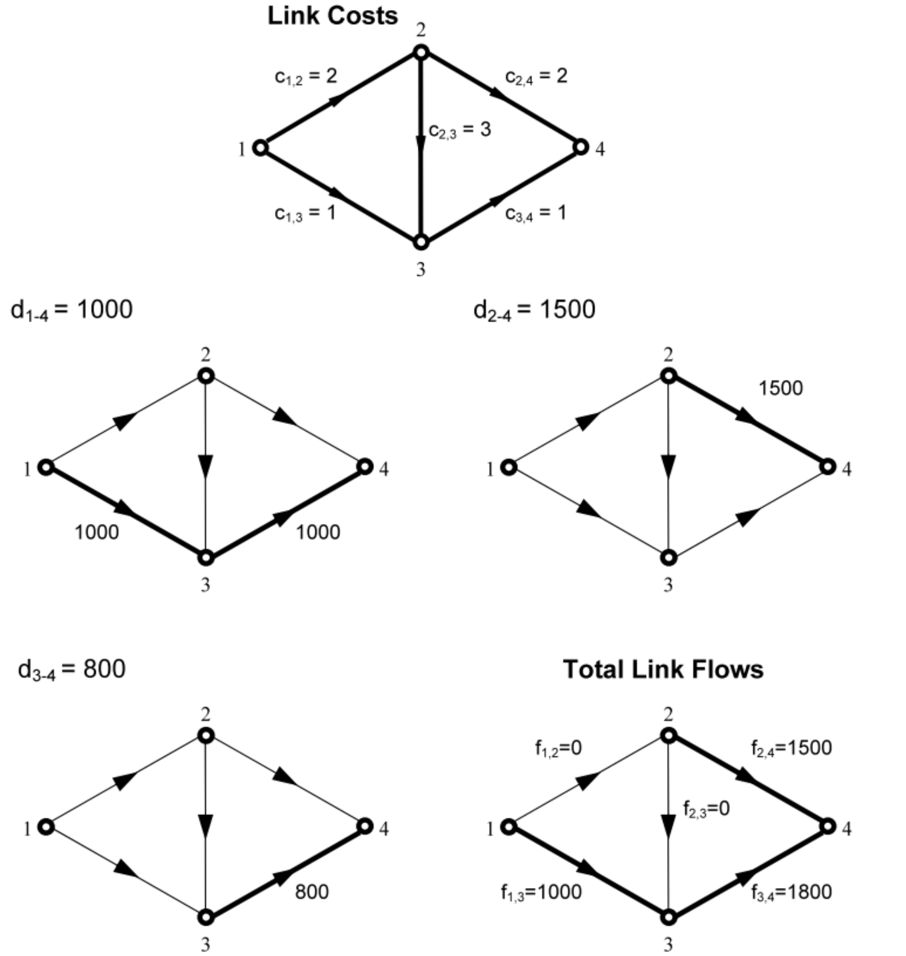

Fig. 3 Deterministic UNcongested network (DUN) assignment, with the path choice model.

In order to facilitate the analysis and solution of model (6), it can be reformulated without any explicit reference to path flows. Let:

$G^{NA} = (g^{NA})^T h$ be the total nonadditive cost corresponding to a feasible path flow vector h

$G^{NA}_{DUN} = (g^{NA})^T h_{DUN}$ be the total nonadditive cost of the deterministic uncongested assignment of path flow vector h_{DUN}

The following relationship, involving link flows f_{DUN} and total nonadditive cost G^{NA}_{DUN}, holds for deterministic uncongested network assignment.

$$c^T(f - f_{DUN}) + 1(G^{NA} - G^{NA}_{DUN}) \geq 0$$

$$\forall f = \Delta h, \ \forall G^{NA} = (g^{NA})^T h \ \forall h \in S_h$$

Model (7) can be made formally similar to model (6) by considering an additional pseudo link *a*, with which is associated an additional row within matrix Δ, with "flow" G^{NA} and cost 1. The existence of solutions of any of the inequality systems (4) and (6) is assured because they are defined over compact feasible sets. Demand flows affect the solution because they appear in the definition of the feasible sets over which the problems are defined:

Formulation with optimization models. Deterministic uncongested network assignments can also be formulated with an optimization model, more precisely, with a linear programming model. It is easy to verify that, if the nonadditive path costs are zero, the inequality system (5) is equivalent to an optimization model with linear objective function and a set of linear equality and inequality constraints as given below.

$$f_{DUN}(c;d) = argmin\ c^T f$$

$$f \in S_f(d) \qquad 7$$

Where the notation $S_f(d)$ highlights the role of the demand flow vector in the definition of the feasible link flow set. If there are nonadditive path costs, the relation (7) becomes:

$$(f_{DUN}(c;d), G^{NA}_{DUN}) = arg \min_{f, G_{NA}} c^T f + 1. G^{NA}$$

$$f = \Delta h, \quad G^{NA} = (g^{NA})^T h, \;\; h \in S_h \qquad 8$$

These formulations are most easily understood by considering that the assignment of each demand flow to a minimum cost path corresponds to the case where the cost for each user and the total network cost is both minimum (the link costs being independent of flows).

Regardless of the model adopted, the link flow vector (or rather one of the vectors) resulting from deterministic uncongested network assignment can easily be calculated when using path choice models based on explicit path enumeration. When nonadditive path costs are equal to zero, a link flow vector can easily be obtained without explicit path enumeration using procedures based on algorithms for the calculation of minimum cost paths, or by directly solving optimization models (7) and (8).

Shortest Path Algorithms

Modeling of path choice behavior in assignment algorithms frequently involves identification of the shortest paths between pairs of nodes. In particular, assignment algorithms that incorporate deterministic path choice assumptions require the identification of the shortest path (or paths) between each pair of nodes, whereas stochastic uncongested network assignment algorithms that incorporate probabilistic path choice models sometimes compute shortest paths as a step in the processing. Furthermore, models that construct a relevant path set by applying a selective approach and explicitly enumerating paths generally involve the solution of the shortest path problem. For example, the relevant path set could be specified as the set of paths that minimize different link attributes such as distance, monetary cost, and travel time; alternatively, they might be identified as the first *k* shortest paths with respect to some link attribute.

If only *elementary paths* (those without loops) are relevant, there are a finite number of them and in principle, they could be enumerated for each pair of origin and destination nodes. The shortest path could then be identified by inspection. When explicit enumeration of all paths is not feasible due to their large number, as is often the case, algorithms that avoid explicit enumeration must be adopted. These are described here.

Applications in transportation network assignment typically do not require the determination of the shortest path between all possible pairs of nodes, but only between pairs of origin and destination nodes (O-D pair) relative to centroids. It should be remembered that each centroid is represented network model by two unconnected nodes: an origin node, with only exiting links, and a destination node, with only entering links

Nonetheless, rather than computing the shortest path for each individual O-D pair, in turn, it is often easier to compute the set of shortest paths between an origin (or destination) node and all other network nodes (including the possible destination nodes), looping over the origins (or destinations) until shortest paths for all O-D pairs have been found. This approach is usually more computationally efficient than determining all O-D paths one at a time and corresponds more closely to the typical processing logic of assignment algorithms (which generally treat all flows from an origin or to a destination in one step). This section, therefore, describes the basic structure of algorithms for computing shortest paths from an origin node o to all network nodes (forward shortest paths), or from all network nodes to a destination node d (backward shortest paths).

For simplicity, the performance variable associated with each link is referred to as cost, in as much as in practice it often represents a generalized transportation cost. However, it could just as well be any other performance measure (distance, travel time, etc.). Only link-additive performance measures are considered unless otherwise noted. Moreover, the link performance variable is assumed to be nonnegative. Let:

$c_a = c_{ij} \geq 0$ be the cost on link $a = (i, j)$

$Z_{i,j} \geq 0$ be the cost of the shortest path between any pair of nodes *i* and *j* ; note that in general, it may happen that $Z_{i,j} \neq Z_{j,i}$ (due, e.g., to one-way streets, slopes, etc.).

The shortest path costs satisfy the triangle inequality:

$$Z_{i,j} + Z_{j,k} \geq Z_{i,k} \qquad \forall i, j, k$$

This can be seen by noting that if, for a pair of nodes *i* and *k*, there were a node *j* for which

$Z_{i,j} + Z_{j,k} < Z_{i,k}$, then the cost of the path from *i* to *k* through node *j* would be less than $Z_{i,k}$, contradicting the definition of $Z_{i,k}$ as the cost of the shortest path from *i* to *k*. Because *i* and *k* are arbitrary, this relationship holds in particular for origin and destination nodes, and the shortest paths between them.

The triangle inequality implies that link costs and shortest path costs satisfy the *Bellman principle*, which states that the shortest path is itself made up of shortest paths:

If link (i, j) belongs to the shortest path between *o* and *j*

Then $Z_{o,i} + c_{ij} = Z_{o,j}$ otherwise $Z_{o,i} + c_{ij} \geq Z_{o,j}$

More generally:

If link *(i, j)* belongs to the shortest path between *o* and *d*

Then $Z_{o,i} + c_{ij} + Z_{j,d} = Z_{o,d}$ otherwise $Z_{o,i} + c_{ij} + Z_{j,d} \geq Z_{o,d}$

If there is only one shortest path between each pair of nodes in a network, the second assertion of each of the above two formulations of the *Bellman principle* holds as a strict inequality. It can easily be seen that, for an uncongested network, the *Bellman principle* is equivalent to the first *Wardrop* principle discussed previously.

It should be recognized that if there is only one shortest path between each pair of nodes (or, when there are several shortest paths if only one is considered), the set of shortest paths from an origin node o to the other network nodes forms a forward tree T (*o*) rooted at node *o*. Any forward tree can be described by specifying, for each node *j*, the unique link that enters it (or equivalently by specifying the initial node of this entering link). Similarly, the set of shortest paths from all network nodes to a destination node *d* forms a backward tree T (*d*) rooted at node *d*. Any backward tree can be described by specifying the unique link that exits from each node *i* (or equivalently by specifying the final node of this exiting link). The use of the same notation for forwarding trees from an origin *o* and for backward trees towards a destination *d* is not ambiguous, because we only consider trees rooted at the origin or destination nodes: in this case, the type of root (origin or destination) defines the type of tree (forward or backward).

Given any forward tree T (*o*) from origin node *o*, let:

$X_{T(o),i} \geq 0$ be the cost along the unique path from node *o* to node *i* in tree T(*o*) It follows that

$$X_{T(0),i} + c_{ij} = X_{T(0),j} \qquad \forall (i,j) \in T(0)$$

A tree T (*o*) from the origin node *o* is the shortest path tree (or is one such tree when there are multiple shortest paths between some pairs of nodes) if and only if the following condition, is deduced from the *Bellman principle*, is verified.

$$X_{T(0),i} + c_{ij} \geq X_{T(0),j} \qquad \forall (i,j) \notin T(0) \qquad 9$$

In this case, the values $X_{T(0),i}$ are the shortest path costs $Z_{o,i}$.

Similarly, given a backward tree T (*d*) towards destination node *d*, let:

$X_{i,T(d)} \geq 0$ be the cost along the unique path from node *i* to destination *d* in tree T (d)

It follows that

$c_{ij} + X_{j,T(d)} \geq X_{i,T(d)} \qquad \forall (i,j) \in T(d)$

In this case, a tree T (d) to destination node d is the shortest path tree (or is one such tree when there are multiple shortest paths between some pairs of nodes) if and only if the following condition is verified.

$c_{ij} + X_{j,T(d)} \geq X_{i,T(d)} \qquad \forall (i,j) \notin T(d)$ 10

In this case the values $X_{i,T(d)}$ are again the shortest path costs $Z_{i,d}$.

The algorithms commonly used to compute forward (resp., backward) shortest-path trees are based on the iterative updating of the values $X_{T(0),i}$ (resp., $X_{j,T(d)}$), called the node labels. In each iteration, a node is chosen, and the labels of immediately downstream (resp., upstream) nodes are examined and updated as required. Iterations continue until condition (9) (resp., (10)) holds everywhere, at which point the minimum path costs have been found. Bookkeeping operations carried out along with the label updates enable the specific minimum path to (resp., from) each node to be traced.

The number of steps that an algorithm requires to compute the minimum path tree depends on its strategy for choosing, in each iteration, the node at which to verify whether further updating steps are needed.

Examples of updates for a forward tree from origin o, and for a backward tree towards destination d, are shown in Figs. 4a and 4b.

When there are multiple shortest paths between a particular O-D pair, the set of shortest paths from an origin (or towards a destination) is no longer a tree. The algorithms presented above will determine only one of the shortest paths; the particular one identified depends on the order in which the nodes are examined. The algorithms can easily be modified to compute all possible shortest paths, although in practice this is rarely done.

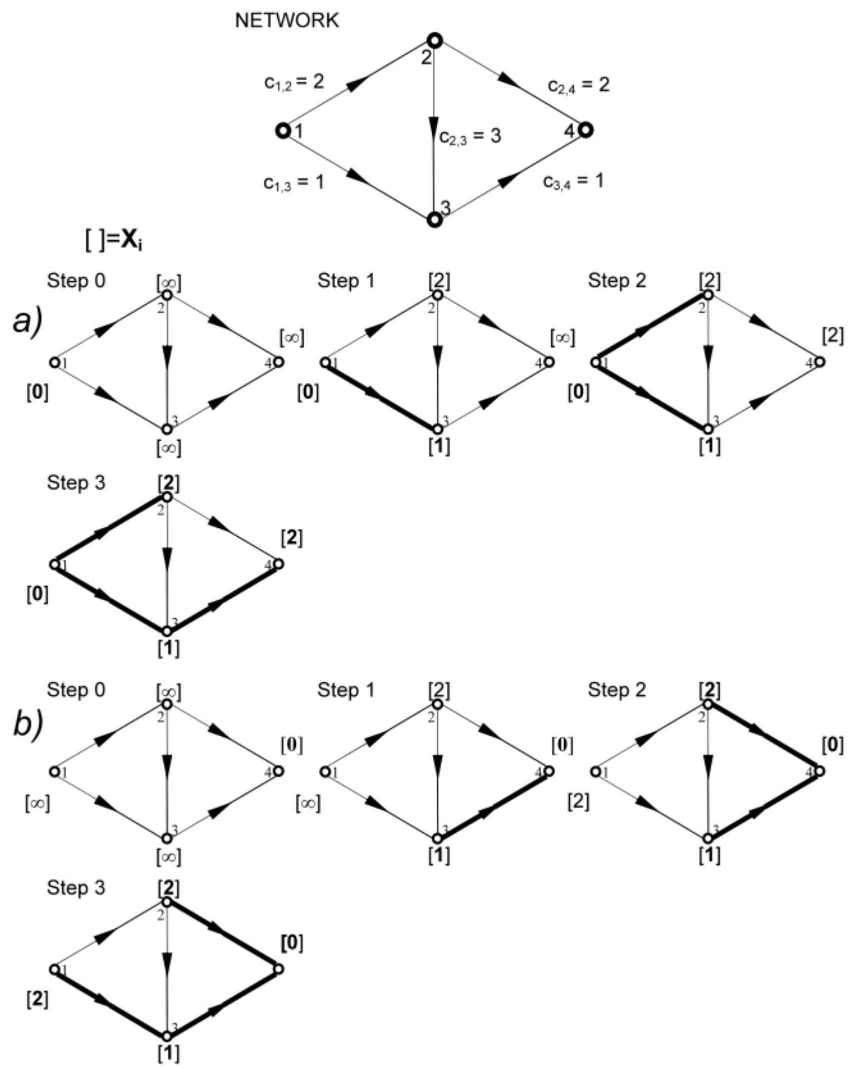

Fig. 4 Example of forward (a) and backward (b) label-setting shortest path algorithms.

Algorithms for Uncongested Network Deterministic Assignment

Under the assumption of deterministic path choice behavior, all users traveling from an origin to a destination choose the shortest path between them; this leads to deterministic uncongested network assignment. Algorithms for DUN assignment are known as all-or-nothing assignment algorithms.

As observed above, if multiple shortest paths connect an O-D pair, then path flows, and therefore link flows, are not uniquely defined. However, shortest path algorithms usually compute a single path between each O-D pair. The specific path identified depends on the implementation details of the algorithm and in particular on the ordering of the nodes.

Link flows can therefore be calculated by assigning all the flow of each O-D pair to the links of the shortest O-D path, and nothing to the links of other paths. In practice, all-or-nothing algorithms generally process the entire tree of shortest paths from an origin or to a destination, rather than individual shortest O-D paths. They can be implemented with two different approaches. Both start with an empty network.

In the *sequential approach*, once the shortest path tree from an origin *o* has been calculated, the O-D demand d_{od} from the origin towards each destination *d* is added to the flows on all the links on the path from *o* to *d*. The DUN link flows result when all O-D specific flows have been accumulated on each link in this way. An example of the sequential algorithm is given in Fig. 4. The procedure is analogous if the shortest path tree towards each destination *d* is calculated.

In contrast to the sequential approach, other DUN assignment algorithms follow a *simultaneous* approach. Simultaneous algorithms are computationally more efficient and can be extended to DUN assignment models for transit networks (shortest hyperpaths). These algorithms are particularly efficient if each shortest path tree designates the nodes in order of increasing minimum cost from the origin (or to the destination). As discussed above, such an order is automatically obtained from label-setting shortest-path algorithms.

Simultaneous algorithms from an origin are based on the calculation of the flow entering each node, defined as the sum of the flows on the links incident to the node. Considering one origin *o* at a time, each destination node *d* is initially assigned the corresponding demand flow d_{od} as its entering flow; all other nodes are tentatively assigned a zero entering flow. Once the tree of shortest paths from origin o has been calculated, the algorithm examines each node *i* in decreasing order of minimum cost, starting with the node farthest from origin *o* (i.e., the node *i* with the highest value Z_{oi}), and working backward until *o* is reached. The flow entering node *i* is assigned to the unique previous link in the shortest path tree and added to the flow entering the initial node of this link. The order adopted is such that, when node *i* is examined, all nodes farther from the origin have already been examined. Consequently, there cannot be any node still to be examined from which the flow could contribute to the flow entering node *i*.

For each O-D pair *od*, the EMPU associated with deterministic path choice is given by the cost on the shortest path, $S_{od} = Z_{od}$. An example of the application of a simultaneous algorithm is given in Fig. 5. The procedure is analogous if the shortest path trees towards each destination *d* are calculated.

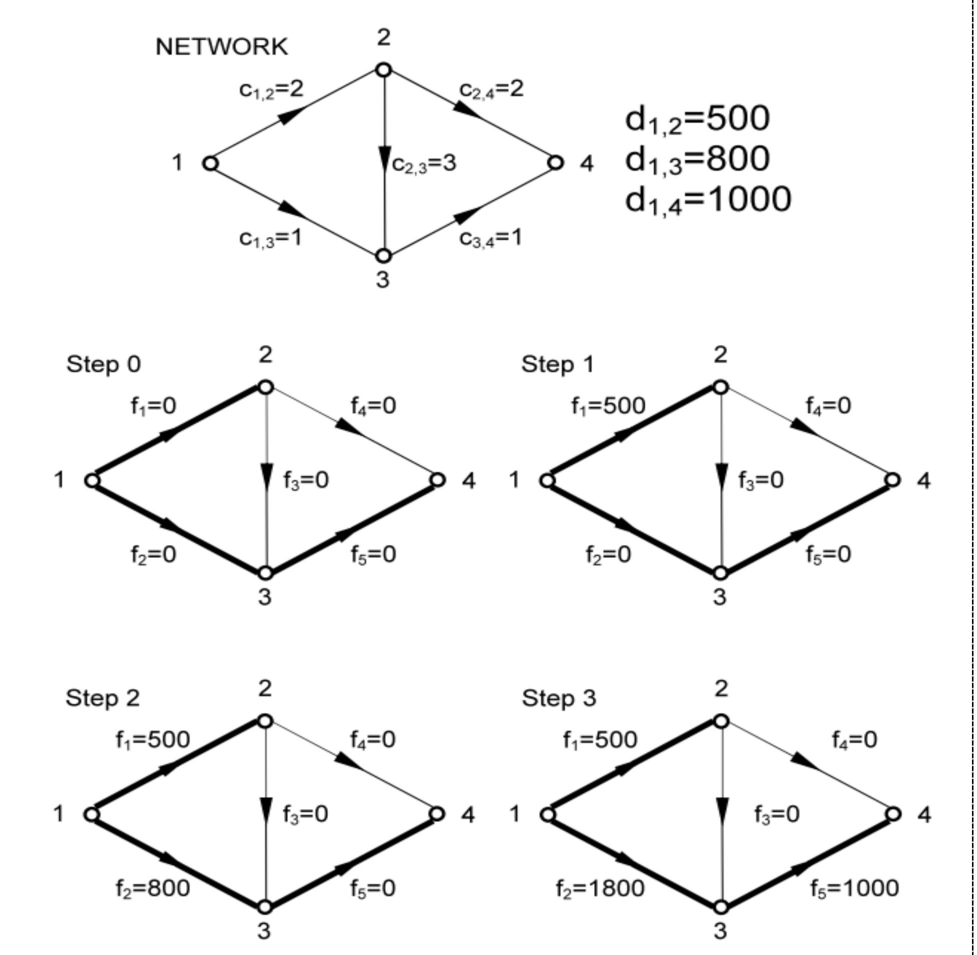

Fig. 5 Example of sequential forward algorithm for DUN assignment.