Stochastic Quasi-gradient Algorithm for the Off-line Stochastic Dynamic Traffic Assignment Problem

Srinivas Peeta
School of Civil Engineering
Purdue University
West Lafayette, IN 47907-1284

and

Chao Zhou
Sabre, Inc.
1 East Kirkwood Blvd.
Southlake, Texas 76092

Abstract

This paper proposes a stochastic quasi-gradient (SQG) based algorithm to solve the off-line stochastic dynamic traffic assignment (DTA) problem that explicitly incorporates randomness in O-D demand, as part of a hybrid DTA deployment framework for real-time operations. The problem is formulated as a stochastic programming DTA model with multiple user classes. Due to the complexities introduced by real-time traffic dynamics and system characteristics, well-behaved properties cannot be guaranteed for the resulting formulation and analytical functional forms that adequately capture traffic realism typically do not exist for the associated objective functions. Hence, a simulation-based SQG method that is applicable for a generalized differentiable (locally Lipschitz) non-convex objective function and non-convex constraint set is proposed to solve the problem. Simulation is used to estimate quasi-gradients that are stochastic to incorporate demand randomness. The solution approach is a generalization of the deterministic DTA solution methodology; under it, deterministic DTA models are special cases. Of practical significance, it provides a robust solution for the field deployment of DTA, or an initial solution for hybrid real-time strategies. The solution algorithm searches a larger feasible domain of the solution space, leading to a potentially more robust and computationally more efficient solution than its deterministic counterparts. These advantages are highlighted through simulation experiments.

Keywords

Deployable dynamic traffic assignment; stochastic optimization; stochastic quasi-gradient methods; simulation-based optimization.

1. Introduction

The vast majority of dynamic traffic assignment (DTA) models in the literature are deterministic as they assume the origin-destination (O-D) demands and supply conditions to be known *a priori* for the entire planning horizon of interest. These assumptions aid the solution process and highlight a fundamentally complex characteristic of the DTA problem; the dependence of the current solution on the future traffic conditions. Typically, DTA solutions are specified in terms of link flows, path assignments or path assignment proportions. Peeta and Ziliaskopoulos (2001) provide a comprehensive review of DTA models in terms of their characteristics, formulation approaches and solution methodologies. They can be broadly classified into two groups: analytical and simulation-based models. Due to the complexities introduced by real-time traffic dynamics and system characteristics, analytical functional forms that adequately capture traffic realism typically do not exist for the associated objective functions and/or constraints. Hence, the resulting formulations generally lack well-behaved properties, precluding the guarantee of global optimality and desirable mathematical properties for the solution. Approaches that generate exact analytical solutions compromise on key characteristics of the problem, primarily the traffic flow modeling aspects, raising issues of realism. Simulation-based models can capture the traffic dynamics robustly, but mathematical properties cannot be derived for them. Due to the unrealistic assumption of complete *a priori* knowledge on demand and supply conditions for the entire planning horizon, deterministic DTA models are not robust for real-time operations. Also, their solution procedures are computationally intensive, precluding real-time deployment. Hence, they are primarily viewed as off-line models and serve as useful benchmarks for real-time DTA deployment strategies.

In the context of DTA, the sources of randomness inherent to general traffic networks equipped with advanced information systems are O-D demands, incidents and other supply conditions, driver response to supplied information, and data measurement errors. If one or more of these sources of stochasticity are explicitly modeled, it leads to a stochastic DTA problem. If a stochastic DTA problem is addressed off-line, that is, ignoring on-line issues, an off-line stochastic DTA problem is obtained. Typically, the primary focus of an off-line stochastic DTA problem is on generating an effective off-line solution. When on-line issues that arise in the context of DTA deployment are addressed, the resulting formulation is labeled an on-line stochastic DTA problem. These issues include solution robustness, solution stability, real-time computational tractability, and consistency with the actual system. Hence, an on-line stochastic DTA model should address the randomness in O-D demands and network supply, measurement errors, and on-line driver behavior. However, due to the complexity involved, it is practical to address these issues by deploying a combination of on-line DTA solution strategies and consistency-checking procedures (Peeta and Bulusu, 1999). The consistency-checking procedure seeks to bridge the potential gap between the predicted system state due to the on-line DTA solution strategy and the actual system unfolding on-line. Hence, it acts as a corrective/feedback mechanism after the actual conditions evolve in real-time. It addresses several factors such as incorrect predictions, unpredicted variations, incorrect modeling, and measurement errors. The on-line DTA solution strategies focus on solution robustness and computational tractability. Solution robustness issues arise due to demand and supply (primarily due to incidents) randomness in real-time traffic systems. Computational tractability implies the need to solve the problem in less than real-time to provide drivers timely routing information or to enable real-time control strategies. The on-line solution strategies can be classified into reactive, predictive/iterative, and hybrid approaches. Reactive approaches (Hawas and Mahmassani, 1995; Pavlis and Papageorgiou, 1999) propose solutions based only on current traffic conditions and avoid prediction. Predictive approaches (Peeta and Mahmassani, 1995a) use historical data and predictive components (in terms of demand and/or supply conditions) in addition to the current traffic conditions to determine the path assignment proportions. Hybrid approaches (Peeta and Zhou, 2002) combine elements of predictive and reactive approaches to generate deployable solutions.

Peeta and Mahmassani (1995a) propose a stage-based rolling horizon framework for real-time DTA deployment. It is an iterative approach in which accurate predictions of future O-D demands and network supply conditions are assumed available for the next stage. Since it is stage-based, the rolling horizon approach ensures that unpredicted variations in on-line traffic conditions can be adequately accounted for in subsequent stages. However, if the actual O-D desires and network conditions in a stage are significantly different from the forecasts, the solution is sub-optimal. Also, despite solving DTA problems for truncated horizons represented by the stages, the rolling horizon approach can entail substantial computational burden in centralized architectures.

Hawas and Mahmassani (1995) propose a reactive decentralized DTA strategy to address the on-line computational burden. The traffic network is divided into small regions in which local controllers use currently available information and heuristics to make local path assignment decisions. An attractive feature of this approach is the flexibility in defining the territorial size of each controller based on its processing capabilities, thereby circumventing issues of computational burden. Also, it does not require O-D demand predictions. However, since controllers act independently based only on local rules, there is no coordination between them. If the proportion of inter-territory vehicles is large, the associated solution deviates substantially from the optimal solution. Hawas (1995) extends the above approach to develop a cooperative scheme that enables exchanging information between neighboring controllers. While the decentralized architecture is more robust under incident situations because the local rule heuristics are more responsive to current network conditions, there can be substantial degradation in performance under non-incident conditions compared to the benchmark centralized deterministic DTA solution. Also, this approach does not exploit historical data on time-dependent O-D demand and incidents.

Pavlis and Papageorgiou (1999) propose a computationally efficient reactive decentralized feedback strategy that avoids predictions and reacts to real-time measurements based on simple control rules to establish equal instantaneous travel times on the used routes for an O-D pair. However, akin to Hawas and Mahmassani(1995), historical data is not exploited. Also, the underlying driver behavior processes are not explicitly incorporated, limiting its robustness to specific network topologies.

The three deployable DTA approaches discussed heretofore address the on-line DTA problem, but do not explicitly incorporate the various sources of randomness. Peeta and Zhou (2002) develop a hybrid DTA deployment approach consisting of off-line and on-line components for the on-line stochastic DTA problem for traffic networks with advanced information systems. It aims to exploit historical traffic data while ensuring real-time solution tractability. Based on the *a priori* optimization concept (Jaillet, 1988), the computationally intensive off-line component is used to generate a robust initial solution, called the *a priori* solution, for real-time operations. The on-line component updates the *a priori* solution in real-time based on the unfolding demand and supply conditions for that specific day. The off-line component solves deterministic DTA problems for the entire planning horizon for likely O-D demand realizations generated from historical probability distributions. The associated solutions are combined to determine the *a priori* solution. Hence, the off-line component solves the off-line stochastic DTA problem in which demand randomness is explicitly considered. Following the *a priori* optimization concept, the on-line component for a given day consists of efficiently updating the off-line solution for each assignment interval using a dynamic solution update heuristic. Supply randomness, primarily manifesting as incidents, is addressed on-line (Peeta and Zhou, 2002) since incident characteristics can be incorporated seamlessly with minor time lags. The dynamic update heuristic under incidents uses the deterministic DTA solutions generated off-line for several likely incident scenarios, in addition to the *a priori* solution. Thereby, the off-line component exploits historical data and the on-line component circumvents demand/supply predictions and the need to solve a DTA problem in real-time.

A key focus of the *a priori* optimization technique is the robustness of the *a priori* solution. Here, robustness implies that the solution should minimize the expected system travel time averaged over several likely O-D demand realizations. That is, across the range of likely O-D demand realizations, it should have the minimal average deviation from the corresponding deterministic DTA solutions. Two characteristics are desirable for the *a priori* solution: (i) it should be robust vis-à-vis the optimal solution to the particular realization, on average, and (ii) it should be robust enough so that the update solution on-line for each realization can be obtained efficiently.

Peeta and Zhou (1999) propose a heuristic procedure to obtain the *a priori* solution for the off-line stochastic DTA problem using the Monte Carlo simulation approach. The deterministic DTA solutions to several likely O-D demand realizations generated from the probability distributions are averaged assuming equal likelihood of the generated realizations. The averaged path assignment proportions are the initial solution for the on-line component. The study illustrates that the *a priori* solution is substantially more robust than the benchmark mean O-D demand based deterministic DTA solution, even under incidents, for real-time operations. This is because the *a priori* solution generates a larger relevant path set by considering several realizations compared to the single realization represented by the mean O-D demand pattern. Peeta and Zhou (1999) also discuss the characteristics of the O-D demand distributions and the mechanism to determine the number of O-D demand realizations L required to limit the tolerable error in average system travel time. While the heuristic procedure to determine the *a priori* solution is practically convenient, it lacks theoretical insights.

This paper develops a theoretical framework to generalize the deterministic DTA problem and the associated solution methodology to incorporate stochasticity. It addresses the off-line stochastic DTA problem that incorporates randomness in O-D demand. Formulated as a stochastic programming model, the problem is solved using a stochastic quasi-gradient (SQG) based algorithm to generate a robust initial off-line solution for real-time deployment, called the off-line *a priori* solution. It is a conceptually elegant stochastic extension and the generalization of the methodology (Peeta, 1994; Peeta and Mahmassani, 1995a) to solve the multiple user classes deterministic DTA problem in which O-D demands are assumed known *a priori* for the entire planning horizon. Deterministic DTA models are special cases under this general framework. The approach illustrates theoretical insights and focuses on the inherent characteristics of the DTA problem. Given the lack of analytical functional forms to represent the typical DTA objective functions, which involve traffic flow modeling, simulation is used to estimate quasi-gradients in the associated solution algorithm. This is because gradients or sub-gradients cannot be determined for the general DTA problem unless compromises in traffic realism are made. The estimated quasi-gradients are stochastic to incorporate demand randomness. The SQG-based algorithm is iterative and updates the current solution using the quasi-gradients determined in the current iteration. Hence, the updated solution in each iteration consists of a single vector of path assignment proportions determined based on the expected system travel time across all O-D demand realizations. Since the *a priori* solution is an off-line entity, computational tractability issues do not arise.

The paper is divided into six sections. Section 2 formulates the off-line stochastic DTA problem. Section 3 discusses the theoretical background for solving the problem. Section 4 describes the SQG-based solution algorithm and discusses various associated theoretical aspects. Section 5 analyzes the SQG-based algorithm through simulation experiments. Concluding comments are provided in Section 6.

2. Formulation of the off-line stochastic DTA problem

2.1 Definition of variables for the formulation

The following notation is used to represent variables in the formulation:

i = origin node, $i \in I$

j = destination node, $j \in J$

n = node in the network, $n \in N$

a = link in the network, $a \in A$

τ = departure (or assignment) time interval, τ = 1,......, T

t = current time interval, t = 1,......, T

Δ = length of a time interval

u = user class, $u \in U$

K_{ij}^{u} = number of paths for user class u between OD pair (i, j)

$k(u)$ = path in the network, $k(u) \in K_{ij}^{u}$, $i \in I$, $j \in J$, and $u \in U$

Λ = set of feasible time-dependent O-D demand realizations

λ = indicator for a feasible O-D demand realization, $\lambda \in \Lambda$

p_{λ} = probability of realization λ

$r_{ij}^{\tau u}$ = number of vehicles of user class u departing from i to j in period τ

$r_{ij}^{\tau u,\lambda}$ = number of vehicles of user class u departing from i to j in period τ in realization λ

$f_{ijk(u)}^{\tau u}$ = proportion of vehicles departing from *i* to *j* assigned to path *k(u)* in period τ

$r_{ijk(u)}^{\tau u,\lambda}$ = number of vehicles of class *u* departing from *i* to *j* in interval τ assigned to path *k(u)* in realization λ

$\delta_{ijk(u)}^{\tau tau,\lambda}$ = time-dependent link path incidence indicator:

$$\delta_{ijk(u)}^{\tau tau,\lambda} \begin{cases} = 1, \text{ if } r_{ijk(u)}^{\tau u,\lambda} \text{ is on link } a \text{ in interval } t \\ = 0, \text{ if link } a \text{ does not belong to path } k(u) \\ = 0, \text{ if } \tau > t \\ = 0, \text{ if } r_{ijk(u)}^{\tau u,\lambda} \text{ is not on link } a \text{ in interval } t \end{cases}$$

$T_{ijk(u)}^{\tau u,\lambda}$ = experienced path travel time for vehicles of user class *u* going from *i* to *j* that are assigned path *k(u)* in interval τ for realization λ

$\widetilde{T}_{ijk(u)}^{\tau u,\lambda}$ = path marginal travel time for vehicles of user class *u* going from *i* to *j* that are assigned path *k(u)* in interval τ for realization λ

$w_{ijk(u)}^{\tau tau,\lambda}$ = number of vehicles of user class *u* going from *i* to *j* assigned to path *k(u)* in interval τ that are on link *a* at the beginning of interval *t* for realization λ

$e_{ijk(u)}^{\tau tau,\lambda}$ = number of vehicles of user class *u* going from *i* to *j* assigned to path *k(u)* in interval τ that enter link *a* in interval *t* for realization λ

$m_{ijk(u)}^{\tau tau,\lambda}$ = number of vehicles of user class *u* going from *i* to *j* assigned to path *k(u)* in interval τ with exit link *a* in interval *t* for realization λ

$\theta_{ij}^{\tau,\lambda}$ = minimum experienced travel time from *i* to *j* for vehicles assigned in interval τ for realization λ

$\tilde{\theta}_{ij}^{\tau,\lambda}$ = minimum marginal travel time from *i* to *j* for vehicles assigned in interval τ for realization λ

$w^{ta,\lambda}$ = number of vehicles on link *a* at the beginning of interval *t* for realization λ

$e^{ta,\lambda}$ = number of vehicles which enter link *a* in interval *t* for realization λ

$m^{ta,\lambda}$ = number of vehicles which exit link *a* in interval *t* for realization λ

$I_n^{t,\lambda}$ = number of vehicles generated at node *n* in interval *t* for realization λ

$O_n^{t,\lambda}$ = number of vehicles which exit the network at node *n* in interval *t* for realization λ

$\hat{T}_{vjk(u)}^{\tau u,\lambda}$ = instantaneous path travel time for vehicles of user class *u* going from node *v* to node *j* that are assigned to path *k(u)* in interval τ for realization λ

$\hat{T}_{vjk(u)*}^{\tau u,\lambda}$ = minimum instantaneous path travel time for vehicles of user class *u* going from node *v* to node *j* in interval τ for realization λ

κ = user belonging to class 4

$\zeta\kappa$ = relative indifference threshold for user κ

ε_κ = the absolute minimum travel time improvement needed for a path switch for user κ

$\eta_\kappa(v)$ = path switching indicator variable; $\eta_\kappa(v)$ = 1 if user κ switches from current path to the best alternate path from node *v* to the destination, and 0 if the current path is maintained

C(n) = set of links incident to node *n*

B(n) = set of links incident from node *n*

$R_{ij}^{\tau u}(\mu, \sigma)$ = historical distribution of O-D desires for O-D pair (*i*, *j*) for user class *u* in interval τ with mean $\mu_{ij}^{\tau u}$ and standard deviation $\sigma_{ij}^{\tau u}$.

2.2 Problem description

In the context of advanced information systems, the ideal situation from a controller's perspective is one in which all users are equipped to receive information, and follow route guidance instructions based on a system optimal strategy. However, it is unrealistic to assume complete market penetration. Also, different users may have different capabilities to access information. Additionally, equipped users may have different preferences, and may behave differently even when supplied the same information. Therefore, it is necessary to classify network users into different classes based on information accessibility, information supply strategy, and their behavior under the supplied information. In this study, the network users are categorized into four user classes for generality. They are: (i) non-equipped drivers who follow pre-specified paths which may be historically known or solved for exogenously (PS or $u = 1$), (ii) equipped drivers who follow prescribed system optimal paths (SO or $u = 2$), (iii) equipped drivers who follow user optimal paths (UE or $u = 3$), and (iv) equipped drivers who follow a boundedly-rational switching rule in response to descriptive information on prevailing conditions (BR or $u = 4$). The boundedly rational path switching rule states that users switch from the current path at a decision point (typically a node) if travel time savings based on current traffic conditions on an alternative path exceed a certain threshold. If necessary, these four classes can be further divided into sub-classes.

The off-line stochastic DTA problem in this study is defined as follows. Consider a traffic network represented by a directed graph $G(N, A)$, where N is the set of nodes and A the set of directed arcs. A node can represent a trip origin and/or a destination and/or a junction of physical links. A network with multiple origins $i \in I$ and destinations $j \in J$ is considered for generality. The analysis period of interest, taken here as the planning horizon, is discretized into small equal intervals $t = 1,\ldots, T$. Given the vector of time-dependent O-D demand distributions $R_{ij}^{\tau u}(\mu, \sigma)$ for the planning horizon, $\forall\ i \in I, j \in J, u \in U$ and $\tau = 1,\ldots, T$, determine time-dependent path assignment proportions so as to minimize the expected total system travel time over all O-D demand scenarios while satisfying certain conditions for the user classes. Hence the objective is to find, across all feasible (Peeta and Zhou, 1999) O-D demand scenarios, the optimal time-dependent path assignment proportions $f_{ijk(u)}^{\tau u}$ to be assigned to paths $k(u) = 1,..., K_{ij}^{u}$, $\forall\, i \in I, j \in J, u \in U$ and $\tau = 1,\ldots, T$.

2.3 Problem formulation

Given:

$R_{ij}^{\tau u}(\mu, \sigma)$, $\forall\, i \in I, j \in J, u \in U$ and $\tau = 1,\ldots, T$.

Objective function:

$$\text{Min} \quad \sum_{i}\sum_{j}\sum_{k(u)}\sum_{\tau}\sum_{u}\sum_{\lambda\in\Lambda}[p_{\lambda} r_{ij}^{\tau u,\lambda} f_{ijk(u)}^{\tau u} T_{ijk(u)}^{\tau u,\lambda}] \tag{4a}$$

Subject to:

$$\sum_{k(u)} f_{ijk(u)}^{\tau u} = 1, \qquad \forall i, j, u, \tau \tag{4b}$$

$$\sum_{b} e^{tb,\lambda} = \sum_{c} m^{tc,\lambda} + I_{n}^{t,\lambda} - O_{n}^{t,\lambda}, \qquad \forall t, n, \lambda, b \in B(n), c \in C(n) \tag{4c}$$

$$w^{ta,\lambda} = w^{(t-1)a,\lambda} + e^{(t-1)a,\lambda} - m^{(t-1)a,\lambda}, \qquad \forall t, a, \lambda \tag{4d}$$

$$w^{ta,\lambda} = \sum_{u}\sum_{k(u)}\sum_{\tau}\sum_{i}\sum_{j}[r_{ij}^{\tau u,\lambda} \cdot f_{ijk(u)}^{\tau u} \cdot \delta_{ijk(u)}^{\tau tau,\lambda}], \qquad \forall t, a, \lambda \tag{4e}$$

$$T_{ijk(u)}^{\tau u,\lambda} = \sum_{t=1}^{T}\sum_{a}[\delta_{ijk(u)}^{\tau tau,\lambda} \cdot \Delta], \qquad \forall i, j, \tau, u, k(u), \lambda \tag{4f}$$

$$\delta_{ijk(u)}^{\tau tau,\lambda} = H[(r_{ij}^{\tau u,\lambda} \cdot f_{ijk(u)}^{\tau u}), \forall i, j, u, k(u), \tau, \lambda], \qquad \forall i, j, \tau, t, a, u, k(u), \lambda \tag{4g}$$

$$e^{ta,\lambda} = \sum_{u}\sum_{k(u)}\sum_{\tau}\sum_{i}\sum_{j} e_{ijk(u)}^{\tau tau,\lambda}, \qquad \forall t, a, \lambda \tag{4h}$$

$$m^{ta,\lambda} = \sum_{u} \sum_{k(u)} \sum_{\tau} \sum_{i} \sum_{j} m_{ijk(u)}^{\tau au,\lambda}, \quad \forall t, a, \lambda \tag{4i}$$

$$I_n^{t,\lambda} = \sum_{j} \sum_{u} r_{nj}^{tu,\lambda}, \qquad \forall t, \lambda, n \in I \tag{4j}$$

$$O_n^{t,\lambda} = \sum_{u} \sum_{k(u)} \sum_{\tau} \sum_{i} \sum_{c} m_{ink(u)}^{\tau cu,\lambda}, \qquad \forall t, \lambda, n \in J, c \in C(n) \tag{4k}$$

$$\tau \leq t \ , \tag{4l}$$

$$\delta_{ijk(u)}^{\tau au,\lambda} = 0 \text{ or } 1, \qquad \forall i,\ j,\ \tau, u, k(u), t, a, \lambda \tag{4m}$$

$$[\widetilde{T}_{ijk(2)}^{\tau 2,\lambda} - \widetilde{\theta}_{ij}^{\tau,\lambda}] \cdot r_{ij}^{\tau 2,\lambda} f_{ijk(2)}^{\tau 2} = 0, \quad \forall i,\ j,\ \tau, k(2), \lambda \tag{4n}$$

$$[\widetilde{T}_{ijk(2)}^{\tau 2,\lambda} - \widetilde{\theta}_{ij}^{\tau,\lambda}] \geq 0, \qquad \forall i,\ j,\ \tau, k(2), \lambda \tag{4o}$$

$$[T_{ijk(3)}^{\tau 3,\lambda} - \theta_{ij}^{\tau,\lambda}] \cdot r_{ij}^{\tau 3,\lambda} f_{ijk(3)}^{\tau 3} = 0, \quad \forall i,\ j,\ \tau, k(3), \lambda \tag{4p}$$

$$[T_{ijk(3)}^{\tau 3,\lambda} - \theta_{ij}^{\tau,\lambda}] \geq 0, \qquad \forall i,\ j,\ \tau, k(3), \lambda \tag{4q}$$

$$\eta_{\kappa}(v) = \begin{cases} 1, & if \ \ \hat{T}_{vjk(4)}^{\tau 4,\lambda} - \hat{T}_{vjk(4)*}^{\tau 4,\lambda} > \max(\zeta_{\kappa} \cdot \hat{T}_{vjk(4)}^{\tau 4,\lambda}, \varepsilon_{\kappa}), \\ 0, o.w. \end{cases}$$

$$\forall v \in N, j,\ \tau, k(4), \kappa, \lambda \tag{4r}$$

$$\text{All variables} \geq 0 \tag{4s}$$

$f_{ijk(u)}^{\tau u}$, the path assignment proportions for vehicles of class u departing from i to j in time interval τ, are the decision variables in the formulation. The number of vehicles assigned to a path for an O-D demand realization λ is computed as $r_{ijk(u)}^{\tau u,\lambda} = r_{ij}^{\tau u,\lambda} \cdot f_{ijk(u)}^{\tau u}$. The objective function (4a) states that the goal is to minimize the expected system travel time of the assigned vehicles in the system given the set of feasible O-D demand realizations. It is a non-linear expression due to the interactions between $f^{\tau u}_{ijk(u)}$ and $T^{\tau u,\lambda}_{ijk(u)}$. Feasible O-D demand realizations are those that satisfy the O-D demand characteristics of the particular network of interest (Peeta and Zhou, 1999). Constraints (4f) and (4g) illustrate that the $T^{\tau u,\lambda}_{ijk(u)}$ are themselves a complicated non-explicit function of the assignment decisions $f^{\tau u}_{ijk(u)}$. The intractability arises due to (4g) which represent the traffic flow in the network and capture the complex non-linear time-dependent interactions among vehicles. There exists no known analytical function $H(r^{\tau u,\lambda}_{ij} \cdot f^{\tau u}_{ijk(u)})$ that can adequately capture these interactions. Hence, a common method to determine $H(.)$ is through simulation. Also, the complexity of dynamic traffic phenomena precludes the guarantee of properties such as continuity and differentiability for the objective function.

Constraints (4b) state that the sum of the proportions of vehicles to be assigned to the various paths between an O-D pair is equal to one. They represent the conservation constraints of O-D desires (vehicles) at the origin nodes $i \in I$. Constraints (4c) denote the conservation of vehicles at nodes. They imply that vehicles cannot be stored at nodes, and state that at any time t on a node n the number of vehicles entering all links incident from the node should equal the sum of the number of vehicles exiting links incident to that node and the net generation. Constraints (4d) represent the conservation of vehicles on links, and state that the number of vehicles on any link a at the beginning of time interval t is the net algebraic sum of the number of vehicles on the link at the beginning of the previous time interval (t-1), vehicles entering the link in the interval (t-1), and vehicles exiting the link in the interval (t-1).

Constraints (4e), (4f), (4g) incorporate the 0-1 time-dependent link-path variables $\delta^{\tau a u,\lambda}_{ijk(u)}$. Constraints (4e) relate the number of vehicles on a link $w^{ta,\lambda}$ to the path assignments $r^{\tau u,\lambda}_{ij} \cdot f^{\tau u}_{ijk(u)}$.

They are non-linear due to the dependence of the link-path incidence variables on the path assignments, as expressed in (4g). Constraints (4f) define the path travel times using the incidence variables. The number of time steps in which $\delta_{ijk(u)}^{\tau au,\lambda}$ (for given i, j, u, k, λ, and τ) takes a value 1 implies the number of discrete time steps that the corresponding "packet" of vehicles $r_{ijk(u)}^{\tau u,\lambda} = r_{ij}^{\tau u,\lambda} \cdot f_{ijk(u)}^{\tau u}$ spend in the system, and multiplying with Δ gives the actual (or experienced) travel time for a vehicle in that packet. Constraints (4f) have several implications for the formulation of dynamic traffic assignment models in general. First, the use of time-dependent link-path incidence variables $\delta_{ijk(u)}^{\tau au,\lambda}$ provides a capability for computing the actual travel time of vehicles. This circumvents the use of analytical link performance functions and/or simplistic queuing models to compute link travel times to determine the path travel times. Second, they illustrate the difference between formulations based on instantaneous and experienced travel times. Third, and of fundamental significance to the characterization of dynamic assignment problems, they recognize the dependence of current assignment decisions on future traffic conditions.

Constraints (4h) and (4i) are definitional constraints for the number of vehicles entering and exiting links, respectively, in time interval t. Constraints (4j) and (4k) are definitional constraints for the number of vehicles entering and exiting the network, respectively, at node n in time interval t. Constraints (4l) are the temporal correctness constraints that restrict the start (or departure) time interval τ of assigned vehicles to be at most the current time interval t. Constraints (4m) restrict the time-dependent incidence variables to take the values 0 or 1. Constraints (4n) and (4o) represent the conditions that must be satisfied by the SO class vehicles (user class 2). The SO conditions are satisfied in an expected sense; positive path assignment proportions exist for paths that satisfy the condition that the expected marginal travel time on that path across all O-D demand realizations is equal to the expected minimum marginal travel time for the associated O-D pair across all O-D demand realizations. These conditions reduce to constraints (4n) and 4(o). Similarly, constraints (4p) and (4q) represent the conditions that must be satisfied by the UE class vehicles (user class 3). Constraints (4r) represent the conditions that must be satisfied by the group of users that follow a boundedly-rational switching rule (user class 4). Constraints (4s) represent the non-negativity constraints.

Note that time discretization of the formulation requires that the time intervals considered be sufficiently small to ensure that assigned users belonging to a group ($\tau, i, j, k\ (u), \lambda$) are all located on the same link at any time t. This can be seamlessly ensured through a simulation-based approach with a fine resolution for the simulation time interval (t).

Equations (4a)-(4s) represent the generalized formulation of the stochastic dynamic traffic assignment problem under O-D demand randomness. The commonly formulated deterministic DTA model, which assumes *a priori* knowledge of the time-dependent O-D demands and/or network supply conditions, is a special case of (4a)-(4s).

3. Theoretical background

3.1 Stochastic quasi-gradient methods

Stochastic quasi-gradient methods (Ermoliev, 1983; Ermoliev and Wets, 1988) are numerical techniques for solving stochastic optimization problems with a complex nature of objective functions and constraints. They generalize the stochastic approximation method for unconstrained optimization of the expectation of a random function to problems involving general constraints and non-differentiable functions. They are random search techniques that use asymptotically consistent estimates for the values of the functions and their derivatives, rather than precise values, in the course of searching for the optimal solution. Consider the minimization problem:

$$\text{Minimize} \qquad F^0(x) \tag{5}$$

Subject to:

$$F^i(x) \le 0, \qquad i = 1,......, m, \tag{6}$$

$$x \in X \subseteq R^n \tag{7}$$

If the functions $F^v(x)$, $v = 0,\ldots, m$ are convex, then for every x we have the inequality:

$$F^v(z) - F^v(x) \ge \langle F^v_x(x), z\text{-}x\rangle, \quad \forall z \in X \tag{8}$$

where F^v_x is a generalized gradient at x and $\langle\cdot,\cdot\rangle$ is the Euclidean inner product $\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i = x^T y$.

In SQG methods, the sequence of approximations at iteration s, x^s, s=0, 1, 2,… is constructed using statistical estimates of $F^v(x^s)$ and $F^v_x(x^s)$ rather than their precise values. These estimates are denoted by random numbers $\phi^v(s)$ and vectors $\xi^v(s)$ which on average are close to $F^v(x^s)$ and $F^v_x(x^s)$, respectively. $\phi^v(s)$ and $\xi^v(s)$ are constructed using information on the past history of the optimization process, generated by the path (x^0, …, x^s). For example,

$$E\{\phi^v(s) \mid x^0, \ldots, x^s\} = F^v(x^s) + a_v(s) \tag{9}$$

$$E\{\xi^v(s) \mid x^0, \ldots, x^s\} = F^v_x(x^s) + b^v(s) \tag{10}$$

where the numbers $a_v(s)$ and the vectors $b^v(s)$ may depend on (x^0, …, x^s). For convergence to an optimal solution, as $s \to \infty$, we must have:

$$a_v(s) \to 0, \text{ and } \|b^v(s)\| \to 0 \tag{11}$$

directly or in such a way that,

$$F^{\nu}(x^*) - F^{\nu}(x^s) \geq \langle E\{\xi^{\nu}(s) \mid x^0, \ldots, x^s\}, x^*\text{-}x\rangle + \gamma_{\nu}(s) \qquad (12)$$

where x^* is an optimal solution and $\gamma_{\nu}(s) \to 0$ as $s \to \infty$. The vector $\xi^{\nu}(s)$ is called the stochastic quasi-gradient when $b^{\nu}(s) \not\equiv 0$, or the stochastic generalized gradient when $b^{\nu}(s) \equiv 0$.

For stochastic programming problems with the following general form:

$$F^{\nu}(x) = \mathrm{E}[\psi^{\nu}(x,\omega)],\ \nu = 0, \ldots, m \qquad (13)$$

$\xi^{\nu}(s)$ can be set equal to a sub-gradient of $\psi^{\nu}(.,\omega)$ at x^s,

$$\xi^{\nu}(s) = \psi_x^{\nu}(x^s, \omega^s) \qquad (14)$$

where ω is a random variable belonging to the appropriate probability space, and ω^s is an observation of ω at iteration s. More generally,

$$\xi^{\nu}(s) = \frac{1}{N_s} \sum_{k=1}^{N_s} \psi_x^{\nu}(x^s, \omega^{sk}) \qquad (15)$$

where ω^{sk}, $k = 1,\ldots.., N_s$, $N_s > 0$, is a collection of independent samples.

Suppose we need to minimize a continuous convex function F^0(x) in $x \in X \subseteq R^n$, where X is a closed convex set. If a projection on X, $\pi_X(y) = \text{argmin } \{\|y\text{-}x\|^2: x \in X\}$, can be easily calculated, then the projection method can be applied. The method is defined by the relations:

$$x^{s+1} = \pi_X[x^s - \rho_s\, \xi^0(s)],\ s=0, 1, \ldots \qquad (16)$$

$$F^0(x^*) - F^0(x) \geq \langle E\{\xi^0(s) \mid x^0, \ldots, x^s\}, x^*\text{-}x\rangle + \gamma_0(s) \qquad (17)$$

where ρ_s is the step size, $\gamma^0(s)$ depends on ($x^0, \ldots, x^s$), and $x^* \in X^*$, X^* is the optimal solution set. Ermoliev and Wets (1988) show that the sequence defined by (16) converges to the optimal solution of the original problem with probability 1 under some natural assumptions of interest in practice. They indicate that if $F^0(x)$ is a convex continuous function, X is a convex compact set, and the parameters ρ_s and $\gamma^0(s)$ satisfy with probability 1 the conditions:

$$\rho_s \geq 0,\ \sum_{s=0}^{\infty} \rho_s = \infty,\ \sum E\{\rho_s \,|\, \gamma_0(s) \,| + \rho_s^2 \,\| \xi^0(s) \|^2 \} < \infty \tag{18}$$

then $\lim_{s \to \infty} x^s \in X^*$ with probability 1. Ermoliev (1976) also shows that if $\gamma_0(s) \equiv 0$ and if instead of (18), only the conditions $\rho_s \downarrow 0,\ \sum_{s=0}^{\infty} \rho_s = \infty$ hold, then:

$$\inf_{x^*} E \| x^* - x^s \|^2 \to 0 \tag{19}$$

For non-convex differentiable functions, if the value of the function $F^v(x)$, $v = 0,...,m$ can be calculated easily, then a finite difference approximation method can be used to approximate the gradients $F_x^v(x^s)$ at current point x^s:

$$F_x^v(x^s) \sim \sum_{j=1}^{n} \frac{F^v(x^s + \Delta_s e^j) - F^v(x^s)}{\Delta_s} e^j , \tag{20}$$

where e^j is the unit vector on the j-th axis and $\Delta_s > 0$. However if the functions are non-differentiable, the finite difference approximation does not guarantee the convergence of the optimization procedures. In this case, a modified version of the finite difference approximation scheme through a slight randomization can be used:

$$F_x^v(x^s) \sim \xi^v(s) = \sum_{j=1}^{n} \frac{F^v(\bar{x}^s + \Delta_s e^j) - F^v(\bar{x}^s)}{\Delta_s} e^j \tag{21}$$

where $F_x^v(x^s)$ is a sub-gradient, $\bar{x}^s = (x_1^s + h_1^s, ..., x_j^s + h_j^s, ..., x_n^s + h_n^s)$, and h_j^s are independent random quantities uniformly distributed on the interval $[-\frac{\Delta_s}{2}, \frac{\Delta_s}{2}]$. The convergence of corresponding optimization procedures can be proved (Ermoliev and Wets, 1988) as $\Delta_s \rightarrow 0$ if $F_x^v(x^s)$ are local Lipschitz functions. The vectors $\xi^v(s)$ defined by (21) are statistical estimates of the sub-gradients $F_x^v(x^s)$, satisfying the general requirements (9), (10), and (11). For problems of the form $F^v(x) = E[\psi^v(x,\omega)]$, $v = 0,...,m$, the following analogue of (21) can be used:

$$F_x^v(x^s) \sim \xi^v(s) = \sum_{j=1}^{n} \frac{\psi^v(\bar{x}^s + \Delta_s e^j, \omega^{sj}) - \psi^v(\bar{x}^s, \omega^{s0})}{\Delta_s} e^j . \quad (22)$$

3.2 System optimal and user equilibrium class conditions

Of the four user classes specified in the formulation (4a)-(4s), the PS class users are assumed to have no access to information and can be viewed as background network users. The BR class users decide their paths using the boundedly-rational path switching rule based on the current network conditions. Therefore, the traffic controller provides route guidance to the SO and UE class users only, implying the need to solve for paths that satisfy the SO and UE objectives. The SO paths are based on the controller's objective to minimize the total system travel time while the UE paths are based on users' objectives of minimizing their individual travel times. In the iterative solution algorithm for formulation (4a)-(4s), discussed in Section 4, these objectives are used to determine the search directions for the decision variables. By contrast, the contributions of the PS and BR classes to determining the search directions in the solution algorithm are indirect; they are a factor in the determination of the path travel times. This section discusses marginal travel times and their significance to the determination of the SO paths for the multiple user classes off-line stochastic DTA problem being addressed here. The discussion will illustrate that the interpretation of the SO class conditions is not straightforward here unlike in the single user class problem with only SO users. The conditions to be satisfied by the UE class users are also discussed.

Peeta (1994) and Peeta and Mahmassani (1995b) discuss single user class marginal travel times for the deterministic dynamic traffic assignment problem. They define the time-dependent path marginal travel time for the deterministic DTA case as the effect of an additional vehicle on path k (from i to j) in time interval τ on the system travel time. We extend these concepts to the multiple user classes stochastic case under O-D demand randomness, where the system travel time $z(f, \lambda)$ for realization λ is given by:

$$z(f,\lambda)=\sum_{i}\sum_{j}\sum_{k(u)}\sum_{\tau}\sum_{u}[r_{ij}^{\tau u,\lambda} f_{ijk(u)}^{\tau u} T_{ijk(u)}^{\tau u,\lambda}] \tag{23}$$

The time-dependent marginal travel time for realization λ in the multiple user classes stochastic case, $\widetilde{T}_{ijk(2)}^{\tau 2,\lambda}$, is defined as the effect that the addition of a vehicle in time interval τ to path k of user class 2 (SO user class) from i to j has on the system travel time for realization λ.

$$\widetilde{T}_{ijk(2)}^{\tau 2,\lambda} = \frac{dz(f,\lambda)}{d(\sum_{u} r_{ij}^{\tau u,\lambda} f_{ijk(2)}^{\tau u})} \tag{24}$$

Note that (24) implies that the marginal travel time computation is relevant only for the SO class paths, though these paths may have vehicles of other user classes. This has key implications for the interpretation of the SO user class in a multiple user classes DTA problem with a system optimal objective, as in (4a)-(4s). It states that the marginal travel times depend on all user classes though they are used to determine only the SO class paths. Therefore, the SO class paths are based on the system optimal objective function constrained by the conditions that need to be satisfied by the other user classes. Hence, the SO user class path travel time, on average, may not be the least compared to the average travel times of other user classes. For example, simulation experiments in Peeta and Zhou (1999) indicate that the UE class path travel time is, on average, less than the SO class path travel time when the four user classes discussed in Section 2 are considered. By contrast, in the single user class context, the average SO path travel time is the least.

The expected marginal travel time across O-D demand realizations can then be written as:

$$\widetilde{T}_{ijk(2)}^{\tau 2} = \sum_{\lambda} p_{\lambda} \widetilde{T}_{ijk(2)}^{\tau 2,\lambda} = \sum_{\lambda} p_{\lambda} \frac{dz(f,\lambda)}{d(\sum_{u} r_{ij}^{\tau u,\lambda} f_{ijk(2)}^{\tau u})} \tag{25}$$

By using the Lagrange Multiplier approach and extending the derivation of the SO class conditions for the single user class deterministic DTA case (Peeta and Mahmassani, 1995b), the conditions to be satisfied by the SO class in the multiple user classes stochastic case can be derived (see Zhou, 2002):

$$[\widetilde{T}_{ijk(2)}^{\tau 2} - \widetilde{\theta}_{ij}^{\tau}] \cdot r_{ij}^{\tau 2,\lambda} f_{ijk(2)}^{\tau 2} = 0, \quad \forall i,\ j,\ \tau,\ k(2),\ \lambda \tag{26}$$

$$[\widetilde{T}_{ijk(2)}^{\tau 2} - \widetilde{\theta}_{ij}^{\tau}] \geq 0, \qquad \forall i,\ j,\ \tau,\ k(2),\ \lambda \tag{27}$$

where $\widetilde{\theta}_{ij}^{\tau}$ is the expected minimum marginal travel time from i to j for vehicles assigned in interval τ. Noting that $r_{ij}^{\tau 2,\lambda}$ are constant values for the specific realizations λ, (26) and (27) imply that $f_{ijk(2)}^{\tau 2}$ can be positive for a path $k(2)$ only if the expected marginal travel time across O-D demand realizations is equal to the expected minimum marginal travel time from i to j for vehicles assigned in interval τ. Conversely, if $\widetilde{T}_{ijk(2)}^{\tau 2}$ is greater than $\widetilde{\theta}_{ij}^{\tau}$, $f_{ijk(2)}^{\tau 2}$ is zero for that path.

Using (25) and the definition $\widetilde{\theta}_{ij}^{\tau} = \sum_{\lambda} p_{\lambda} \widetilde{\theta}_{ij}^{\tau,\lambda}$, (26) and (27) can be re-written as:

$$[\sum_{\lambda} p_{\lambda} \widetilde{T}_{ijk(2)}^{\tau 2,\lambda} - \sum_{\lambda} p_{\lambda} \widetilde{\theta}_{ij}^{\tau,\lambda}] \cdot r_{ij}^{\tau 2,\lambda} f_{ijk(2)}^{\tau 2} = 0, \quad \forall i,\ j,\ \tau,\ k(2),\ \lambda \tag{28}$$

$$[\sum_{\lambda} p_{\lambda}\widetilde{T}_{ijk(2)}^{\tau 2,\lambda} - \sum_{\lambda} p_{\lambda}\widetilde{\theta}_{ij}^{\tau,\lambda}] \geq 0, \qquad \forall i,\ j,\ \tau, k(2), \lambda \tag{29}$$

Re-arranging the terms of (28) and (29), we obtain:

$$[\sum_{\lambda} p_{\lambda}(\widetilde{T}_{ijk(2)}^{\tau 2,\lambda} - \widetilde{\theta}_{ij}^{\tau,\lambda})] \cdot r_{ij}^{\tau 2,\lambda} f_{ijk(2)}^{\tau 2} = 0, \quad \forall i,\ j,\ \tau, k(2), \lambda \tag{30}$$

$$[\sum_{\lambda} p_{\lambda}(\widetilde{T}_{ijk(2)}^{\tau 2,\lambda} - \widetilde{\theta}_{ij}^{\tau,\lambda})] \geq 0, \qquad \forall i,\ j,\ \tau, k(2), \lambda \tag{31}$$

Since $p_{\lambda} > 0$ for every realization λ, it follows that ($\widetilde{T}_{ijk(2)}^{\tau 2,\lambda}$-$\widetilde{\theta}_{ij}^{\tau,\lambda}$) should be equal to zero for every λ if $f_{ijk(2)}^{\tau 2}$ is to be positive. Hence, (30) and (31) reduce to:

$$[\widetilde{T}_{ijk(2)}^{\tau 2,\lambda} - \widetilde{\theta}_{ij}^{\tau,\lambda}] \cdot r_{ij}^{\tau 2,\lambda} f_{ijk(2)}^{\tau 2} = 0, \quad \forall i,\ j,\ \tau, k(2), \lambda \tag{32}$$

$$[\widetilde{T}_{ijk(2)}^{\tau 2,\lambda} - \widetilde{\theta}_{ij}^{\tau,\lambda}] \geq 0, \qquad \forall i,\ j,\ \tau, k(2), \lambda \tag{33}$$

which are the same as (4n) and (4o) respectively. (32) and (33) state that satisfying (26) and (27) is equivalent to the SO class conditions being satisfied individually for every realization λ. This is the natural extension of the single user class deterministic DTA SO conditions to the multiple user classes stochastic case under O-D demand randomness.

Akin to (26) and (27) for the SO class, the UE class conditions should also be satisfied in an expected sense:

$$[\overline{T}_{ijk(3)}^{\tau 3} - \theta_{ij}^{\tau}] \cdot r_{ij}^{\tau 3,\lambda} f_{ijk(3)}^{\tau 3} = 0, \quad \forall i,\ j,\ \tau, k(3), \lambda \tag{34}$$

$$[\overline{T}_{ijk(3)}^{\tau 3} - \theta_{ij}^{\tau}] \geq 0, \qquad \forall i,\ j,\ \tau, k(3), \lambda \tag{35}$$

where θ_{ij}^{τ} is the expected minimum experienced travel time from *i* to *j* for vehicles assigned in interval τ, and $\overline{T}_{ijk(3)}^{\tau 3}$ is the expected experienced travel time across O-D demand realizations for path *k(*3). Following a similar logic, (34) and (35) reduce to:

$$[T_{ijk(3)}^{\tau 3,\lambda} - \theta_{ij}^{\tau,\lambda}] \cdot r_{ij}^{\tau 3,\lambda} f_{ijk(3)}^{\tau 3} = 0, \quad \forall i,\ j,\ \tau,\ k(3),\ \lambda \tag{36}$$

$$[T_{ijk(3)}^{\tau 3,\lambda} - \theta_{ij}^{\tau,\lambda}] \geq 0, \qquad \forall i,\ j,\ \tau,\ k(3),\ \lambda \tag{37}$$

which are the same as (4p) and (4q) respectively. (36) and (37) state that satisfying (34) and (35) is equivalent to the UE class conditions being satisfied individually for every realization λ. This is the natural extension of the single user class deterministic DTA UE conditions to the multiple user classes stochastic case under O-D demand randomness.

4 Solution methodology

4.1 Solution framework

Conceptual description

The off-line stochastic DTA problem is solved using an iterative SQG-based solution algorithm. The solution algorithm uses a simulation-based multiple user classes deterministic DTA algorithm (Peeta and Mahmassani, 1995a) and a SQG method to compute stochastic quasi-gradients. Under this framework, first *L* feasible time-dependent O-D demand realizations are generated. At iteration *s*, the current solution represented by the vector of path assignment proportions f^{s} is used by the simulation-based deterministic DTA algorithm to generate an auxiliary solution for each O-D demand realization. Hence, *L* auxiliary solutions are generated for the SO and UE classes. The auxiliary solutions for a realization and the current solution f^{s} are used to generate move directions for that realization. Due to the complexities inherent to the dynamic traffic assignment problem and the lack of explicit analytical functional forms to realistically compute the objective function value, simulation is used here to estimate the move directions. The move directions for each realization are used to compute the vector of stochastic quasi-gradients ξ^s for that iteration. The move size is based on the method of successive averages (Blum, 1954; Powell and Sheffi, 1982). The updated solution f^{s+1} is computed for the SO and UE classes. The paths of the BR class vehicles are obtained from the traffic simulator while those of PS class users remain unchanged. The iteration counter is updated to s+1, and the procedure is repeated until the pre-specified convergence criteria are satisfied.

Stochastic quasi-gradient based algorithm

The SQG-based algorithm used here is based on the stochastic generalized gradient method with projection onto a non-convex constraint set (Ermoliev and Norkin, 1998). As discussed in Section 2, the $\psi^0(x,\omega)$ in (13) may not have an explicit analytical structure. Let $\psi^0(x,\omega)$ be generalized differentiable (Ermoliev and Norkin, 1998), that is, it is locally Lipschitz but generally not directionally differentiable. Therefore, the stochastic objective function $F^0(x)$ = E[$\psi^0(x,\omega)$] is generalized differentiable. Let the set X = $\{x \mid \varphi(x) \le 0\}$ be defined by a generalized differentiable function $\varphi(x)$ that satisfies a regularity condition (Ermoliev and Norkin, 1998). Also, let X^* denote the problem solution set and F^* the set of optimal values, $F^* = \{F(x) \mid x \in X^*\}$. Then, for the non-convex expectation objective function $F^0(x)$ and a non-convex constraint set, the stochastic generalized gradient method is as follows:

$$x^0 \in X \tag{38}$$

$$x^{s+1}(\omega) = \Pi_X[\, x^s(\omega) - \rho_s \, \xi^s \,], \qquad s = 0, 1, 2, \ldots\ldots \tag{39}$$

$$\xi^s = \frac{1}{N_s} \sum_{k=1}^{N_s} \psi_x^0 (x^s, \omega^{sk}) \tag{40}$$

where $x^s(\omega)$ is the current approximation to the optimal solution for realization ω, Π_X is a projection operator which ensures that decision variables satisfy the constraint set, N_s is the number of realizations sampled, ξ^s is the vector of stochastic quasi-gradients for iteration s, and ρ_s is the monotonously decreasing move size that satisfies:

$$\rho_s \geq 0, \ \sum_{s=0}^{\infty} \rho_s = \infty, \ \sum_{s=0}^{\infty} \rho_s^2 < \infty. \tag{41}$$

The method (38)-(40) combines concepts from the projection stochastic quasi-gradient method (Ermoliev, 1976) and the average stochastic gradient method for non-convex functions (Mikhalevich, Gupal, and Norkin, 1987). Ermoliev and Norkin (1998) prove a local convergence result by showing that if the sequence $\{ x^s(\omega) \}$ is generated by method (38)-(41), then its minimum limit points in terms of $F^0(x)$ are almost surely contained in the solution set X^*. In the formulation of the off-line stochastic DTA problem in Section 2, $\omega \equiv \lambda$, $N_s \equiv L$, and $x \equiv f$.

4.2 Theoretical aspects of the solution algorithm

Choice of move direction

As discussed earlier, objective functions for DTA problems that incorporate traffic realism adequately typically lack explicit analytical functional forms and properties such as convexity or differentiability. In (4a)-(4s), since the analytical form of the objective function is not known: (i) simulation is used to compute the objective function value, and (ii) the gradients or sub-gradients cannot be computed in a straightforward manner. Instead, quasi-gradients, which are statistical estimates of the gradients, are obtained by using simulation and the conditions in Section 3.2 to be satisfied by the SO and UE user classes. Since (4a)-(4s) addresses the randomness in O-D demands, stochastic quasi-gradients are computed by taking the expectation of the quasi-gradients across the *L* O-D demand realizations.

A standard approach to compute the quasi-gradients is by using the finite difference approximations discussed in Section 3.1. This method entails projecting the change in the objective function due to a small perturbation to each decision variable while keeping the other decision variables unchanged. However, since simulation is used to compute the objective function value for each realization, this implies one simulation for determining the move direction for a single decision variable in each search iteration for a realization. Due to the large number of decision variables involved, this is computationally challenging even in an off-line context as it implies a large number of simulations per iteration for each realization. The SQG-based solution algorithm circumvents this computational intractability by using only one simulation per realization per iteration to estimate the quasi-gradients for each realization.

Given the complexities arising from traffic flow dynamics and the interactions among multiple user classes with different characteristics and objectives, a descent direction cannot be guaranteed for any iteration in a realization. While this issue exists even for the single user class context, it is further compounded in the multiple user classes context by the fact that while the move directions are based on the SO and UE objectives, the PS and BR class vehicles also influence system performance. For time interval τ in realization λ, the time-dependent least marginal travel time paths and shortest travel time paths are obtained using the link flows and travel times generated by simulating the path assignment proportions of the current solution. As discussed in Section 3.2, the least marginal travel time paths are relevant to the SO class and the shortest travel time paths to the UE class. They are used to determine an auxiliary feasible solution in terms of path assignment proportions, $y_{ijk(u)}^{\tau u,\lambda,s}$, u = 2, 3, using an all-or-nothing assignment. The direction from the current solution $f_{ijk(u)}^{\tau u,s}$ to the auxiliary solution $y_{ijk(u)}^{\tau u,\lambda,s}$, u = 2, 3, provides an estimate of the gradient direction, and is labeled the quasi-gradient $d_{ijk(u)}^{\tau u,\lambda,s}$:

$$d_{ijk(u)}^{\tau u,\lambda,s} = y_{ijk(u)}^{\tau u,\lambda,s} - f_{ijk(u)}^{\tau u,s}, \quad \forall\ i, j, \tau, \lambda, k(u), \text{and } u = 2, 3 \tag{42}$$

Assuming that $p_\lambda = \frac{1}{L}$ (Peeta and Zhou, 1999) given the large number of potential realizations, the stochastic quasi-gradients $\xi_{ijk(u)}^{\tau u,s}$ are obtained by averaging the quasi-gradients (or move directions) over all realizations:

$$\xi_{ijk(u)}^{\tau u,s} = \frac{1}{L}\sum_{\lambda=1}^{L} d_{ijk(u)}^{\tau u,\lambda,s}, \quad \forall\ i, j, \tau, k(u), \text{and } u = 2, 3 \tag{43}$$

The auxiliary solutions in (42) are random variables as they are estimated for specific realizations. Hence, $d_{ijk(u)}^{\tau u,\lambda,s}$ and $\xi_{ijk(u)}^{\tau u,s}$ are random variables implying a random move direction vector. Several previous studies involving deterministic DTA solutions for various networks (for example, Peeta and Zhou, 1999), and the SQG algorithm for the current stochastic DTA problem show smooth convergence of the objective function, and/or a descent direction on average. This suggests that the lack of a guarantee of a descent direction may not be a significant practical issue. Also, as discussed in Section 4.1, when the relatively broad and less restrictive conditions for the SQG method (38)-(41) are satisfied, local convergence is likely.

Choice of move size

Pflug (1988) discusses various move size methods for SQG algorithms. Since there is no guarantee of a unique solution for the problem being addressed here, the practical focus of the solution algorithm is to reach some neighborhood of the solution rather than to find the precise value of the solution itself. Hence, interactive or automatic adaptive move sizes (Gaivoronski, 1988) that use information accumulated during the solution process may be attractive options. However, they entail additional computational costs leading to the usual trade-offs between computation and accuracy. Another approach is a pre-determined move size that satisfies (41). A simple choice satisfying (41) is $\rho_s = \frac{1}{s}$, which in many cases provides the best possible asymptotic rate of convergence (Fabian, 1967). This property does not necessarily imply an advantage in stochastic approximation type procedures where the focus is on reaching some neighborhood of the solution. However, when we factor in the notion that explicit analytical functional forms are lacking for the problem objective function, estimated here through simulation, a pre-determined move size can be an attractive option. Therefore, the method of successive averages (MSA), based on $\rho_s = \frac{1}{s}$, is used here to update the solution. In general, MSA is less restrictive in the sense that even if a random move direction vector is used, convergence entails a descent direction only on average. In our problem context, the solution for the next iteration, $f_{ijk(u)}^{\tau u,s+1}$, is obtained using MSA:

$$f_{ijk(u)}^{\tau u,s+1} = f_{ijk(u)}^{\tau u,s} + \frac{1}{s}\xi_{ijk(u)}^{\tau u,s}, \forall\ i, j, \tau, k(u), \text{ and } u = 2, 3 \tag{44}$$

Projection operator

The projection operator is required to ensure the updated solution is in the feasible region. In our algorithm, the solution $f_{ijk(u)}^{\tau u,s+1}$ for u = 2, 3, obtained using (44) automatically satisfies these constraints as illustrated hereafter. Based on (42), (43), and (44):

$$\sum_{k(u)} f_{ijk(u)}^{\tau u,s+1} = \sum_{k(u)} \left(f_{ijk(u)}^{\tau u,s} + \frac{1}{s}\xi_{ijk(u)}^{\tau u,s} \right)$$

$$= 1 + \frac{1}{s}\sum_{k(u)} \xi_{ijk(u)}^{\tau u,s}$$

$$= 1 + \frac{1}{s \cdot L} \sum_{k(u)} \sum_{\lambda} (y_{ijk(u)}^{\tau u,\lambda,s} - f_{ijk(u)}^{\tau u,s})$$

$$= 1 + \frac{1}{s \cdot L} \sum_{\lambda} \left(\sum_{k(u)} y_{ijk(u)}^{\tau u,\lambda,s} - \sum_{k(u)} f_{ijk(u)}^{\tau u,s} \right) \tag{45}$$

Since $y_{ijk(u)}^{\tau u,\lambda,s}$ are based on all-or-nothing assignment, and $f_{ijk(u)}^{\tau u,s}$ are the current feasible solutions, for any O-D pair (i, j) and u = 2, 3, we have $\sum_{k(u)} y_{ijk(u)}^{\tau u,\lambda,s} = 1$ and $\sum_{k(u)} f_{ijk(u)}^{\tau u,s} = 1$. Therefore, the second term of the right hand side of equations (45) is equal to zero. Hence, (45) becomes $\sum_{k(u)} f_{ijk(u)}^{\tau u,s+1} = 1$, $\forall$ $i, j, \tau, k(u)$, and u = 2, 3. Therefore, the conservation constraints are satisfied for u = 2, 3. The conservation constraints for u = 1 are trivially satisfied since the associated path assignment proportions are pre-determined and fixed. The conservation constraints for u = 4 are implicitly ensured in the simulation since the paths for the BR class vehicles are determined based on the current traffic conditions. Also, they are trivially satisfied when averaged across all O-D demand realizations. Hence, the conservation constraints are satisfied for all user classes.

Further, from (44):

$$f_{ijk(u)}^{\tau u,s+1} = f_{ijk(u)}^{\tau u,s} + \frac{1}{s} \xi_{ijk(u)}^{\tau u,s}$$

$$= f_{ijk(u)}^{\tau u,s} + \frac{1}{s} \left(\frac{1}{L} \sum_{\lambda} d_{ijk(u)}^{\tau u,\lambda,s} \right)$$

$$= f_{ijk(u)}^{\tau u,s} + \frac{1}{s \cdot L} \sum_{\lambda} y_{ijk(u)}^{\tau u,\lambda,s} - \frac{1}{s \cdot L} \sum_{\lambda} f_{ijk(u)}^{\tau u,s}$$

$$= f_{ijk(u)}^{\tau u,s} (1 - \frac{1}{s}) + \frac{1}{s \cdot L} \sum_{\lambda} y_{ijk(u)}^{\tau u,\lambda,s} \tag{46}$$

Since $f_{ijk(u)}^{\tau u,s}$ and $y_{ijk(u)}^{\tau u,\lambda,s}$ are path assignment proportions, $f_{ijk(u)}^{\tau u,s} \geq 0$ and $y_{ijk(u)}^{\tau u,\lambda,s} \geq 0$. Also, as $(1 - \frac{1}{s}) \geq 0$, therefore $f_{ijk(u)}^{\tau u,s+1} \geq 0$, $\forall$ $i, j, \tau, k(u), u = 2, 3$. Following the earlier logic for user classes 1 and 4, the non-negativity constraints are satisfied for all user classes.

Stopping criteria

There are several potential stopping criteria for the proposed SQG-based algorithm. One criterion is to stop when the improvement in the objective function value is less than a small pre-specified value. However, this criterion may not be robust in situations where the objective function does not show smooth convergence or has oscillatory tendencies. Another criterion is to stop when the step size is smaller than a pre-specified value. When using MSA, this criterion is equivalent to reaching the pre-specified upper bound on the number of iterations. A third criterion is to terminate the algorithm if the difference in the decision variables in successive iterations does not exceed a small pre-specified constant. In an actual implementation, careful testing can identify the best criterion among these for that traffic system. To the extent that our objectives are exploratory and developmental, in the experiments in Section 5 we track the change in the objective function value over several iterations to trigger termination of the algorithm. So, an extension of the first stopping criterion is used here.

4.3 The SQG-based solution algorithm

Fig. 1 illustrates the SQG-based solution algorithm for the off-line stochastic DTA problem represented by (4). It is a generalization of the simulation-based multiple user classes deterministic DTA solution algorithm, called the MUCTDTA algorithm (Peeta and Mahmassani, 1995a), to the stochastic case that incorporates randomness in O-D demands. The algorithm is as follows:

Step 0: Generate L feasible time-dependent O-D demand realizations using historical distributions (Peeta and Zhou, 1999). Set the iteration counter $s = 0$.

Step 1: For each generated realization, assign the O-D desires of the equipped user classes (u = 2, 3, 4) to an initial feasible path set. The paths of unequipped vehicles, represented by u = 1, are known *a priori* and remain unchanged throughout the iterative search process. After computing the initial paths for all realizations, obtain the vector of the path assignment proportions averaged across realizations $f_{ijk(u)}^{\tau u,0}$, $\forall$ $i, j, \tau, u = 1,\ldots 4$, and $k(u)$. This represents the initial solution. Set the realization counter $\lambda = 1$.

Step 2: For realization λ, obtain the path assignments $r_{ijk(u)}^{\tau u,\lambda,s} = r_{ij}^{\tau u,\lambda,s} f_{ijk(u)}^{\tau u,s}$, $\forall$ $i, j, \tau, u = 2, 3, 4$, $k(u)$, and $r_{ijk(1)}^{\tau 1,\lambda,s} = r_{ijk(1)}^{\tau 1,\lambda,0}$, $\forall$ $i, j, \tau, k(1)$. The set of path assignments for the entire horizon of interest are simulated using the traffic simulator DYNASMART (Jayakrishnan et al., 1995). The simulation output includes link level and aggregate performance measures. One key measure is the system travel time for realization λ.

Step 3: Compute the link marginal travel times (Zhou, 2002; Peeta and Mahmassani, 1995a) for SO users using the time-dependent experienced link travel times and the number of vehicles on links obtained as post-simulation data from Step 2.

Step 4: Compute the time-dependent least marginal travel time paths and shortest travel time paths.

Step 5: Perform an all-or-nothing assignment of all SO and UE O-D demand desires by assigning the SO users to the least marginal travel time paths and the UE users to the shortest average travel time paths. The result is a set of auxiliary path assignment proportions $y_{ijk(u)}^{\tau u,\lambda,s}$, $\forall$ $i, j, \tau, k(u), u$ = 2, 3. Obtain the move directions $d_{ijk(u)}^{\tau u,\lambda,s}$ for realization λ using equation (42).

Step 6: If $\lambda = L$, go to Step 7; otherwise set $\lambda = \lambda + 1$, and go to Step 2.

Step 7: Compute the average system travel time over all O-D demand realizations. This represents the objective function value in iteration s. Check for convergence using the stopping criterion. If the criterion is satisfied, stop the computation and specify $f_{ijk(u)}^{\tau u,s}$, $\forall$ i, j, τ, u, and $k(u)$, as the solution. Otherwise, go to Step 8.

Step 8: Calculate the stochastic quasi-gradients $\xi_{ijk(u)}^{\tau u,s}$, $\forall$ $i, j, \tau, k(u)$, and $u = 2, 3$, using (43). Update the proportions of vehicles to be assigned to paths for the SO and UE class users $f_{ijk(u)}^{\tau u,s+1}$ using the MSA as shown in (44). Update the path assignment proportions for the BR user class by averaging across O-D demand realizations. Set $s = s+1$, $\lambda = 1$, and go to Step 2.

5. Simulation experiments

5.1 Set-up

Fig. 2 illustrates the test network. It consists of 50 nodes, 168 links, and 320 O-D pairs. All links are 0.4 kilometers long and have two lanes except freeway ramps, which are single-laned. The freeway links have a mean free speed of 88 km/h and other links have a mean free speed of 48 km/h. In terms of signal control, 26 nodes in the network have pre-timed signalization, 8 have actuated signal control, and the rest have no signal control. The planning horizon is 35 minutes long and is divided into seven 5-minute assignment intervals, with the first 5-minute interval representing the start-up time for the network to be reasonably occupied. An average of 21856 vehicles is generated across O-D demand realizations over the 35 minutes, representing a medium congestion level for this network. The O-D demand distributions for an O-D pair are assumed constant within an assignment interval, but not across assignment intervals. Two types of O-D demand distributions are considered in the experiments.

The first is a truncated normal distribution with upper and lower bounds. The second consists of positively and negatively skewed distributions with upper and lower bounds, as shown in Fig. 3.

5.2 Objectives and significance

The deterministic DTA solution obtained by assuming complete *a priori* knowledge of the time-dependent O-D demands for the planning horizon of interest generates the best possible system performance for that O-D demand pattern (or realization). Hence, it serves as an ideal benchmark to compare the performance of a stochastic DTA algorithm under O-D demand randomness. We use the solution of the MUCTDTA algorithm to compare the performance of the proposed SQG-based stochastic DTA solution algorithm. Since O-D demand stochasticity entails the consideration of several potential O-D demand realizations, an appropriate comparison of the MUCTDTA and SQG algorithms requires a consistent definition of the MUCTDTA solution. In this context, the MUCTDTA solution is obtained by using the MUCTDTA algorithm to solve each generated time-dependent O-D demand realization individually and then averaging the associated system travel times across all O-D demand realizations. The experiments seek to: (i) analyze some characteristics of the SQG-based solution algorithm, and (ii) compare the stochastic SQG-based DTA solution with the commonly used deterministic DTA solution under several realistic scenarios.

A fundamental difference between the MUCTDTA and SQG solutions in a DTA deployment context is that the SQG solution is a unique set of path assignment proportions and is *independent* of the actual O-D demand pattern realized. This is synergistic with actual field scenarios where the O-D demands are not known *a priori* for the entire planning horizon on any given day. By contrast, the MUCTDTA solution presumes that the O-D demands are known *a priori* for the entire horizon before solving for the associated path assignment proportions. Hence, the MUCTDTA solution is not a field deployable DTA solution. Hence, if the experiments show that the SQG solution is comparable in performance to the deterministic MUCTDTA solution, the SQG solution would represent an effective deployable DTA solution.

The MUCTDTA and SQG algorithms are essentially off-line entities given the computational burden involved. However, since the SQG solution is a unique set of path assignment proportions, it can also serve as an initial solution for hybrid deployable DTA approaches (Peeta and Zhou, 2002) that additionally incorporate on-line solution update procedures to respond to the actual traffic conditions unfolding on a given day.

5.3 Test scenarios

Five scenarios are considered, as described hereafter:

Scenario 1: In this scenario, 41 time-dependent O-D demand realizations are generated using the truncated normal distributions. Each user class constitutes about 25% of the vehicles generated. The generated realizations are solved individually using the MUCTDTA algorithm. The SQG algorithm is used to obtain a single set of time-dependent path-assignment proportions across the 41 realizations. However, at each SQG solution iteration, the same 41 O-D demand realizations are used to calculate the quasi-gradients for solution update. This scenario directly compares the performance of the MUCTDTA and SQG solutions.

Scenario 2: In this scenario, at each SQG solution iteration, a new set of 41 time-dependent O-D demand realizations from the same distributions used in Scenario 1 are used to calculate the quasi-gradients. As in Scenario 1, each user class constitutes 25% of the vehicles generated. This scenario analyzes whether using the same set of O-D demand realizations in each iteration of the SQG-based algorithm is sufficient to obtain accurate quasi-gradient estimates compared to using different O-D demand realization sets across iterations. That is, does solution robustness increase through the consideration of a wider set of realizations.

Scenario 3: In this scenario, user class fractions in the traffic stream are assumed to follow normal distributions rather than the constant values of Scenario 1. From a real-world perspective, this scenario is more realistic as user class proportions may change with time. The same 41 time-dependent O-D demand realizations generated in Scenario 1 are used here. This scenario examines the effectiveness of the SQG-based algorithm when randomness in user class fractions is considered.

Scenario 4: In this scenario, one set of 41 time-dependent O-D demand realizations is generated for each skewed O-D demand distribution shown in Fig. 3. The skewed distributions have identical means and bounds as the normal distributions assumed in Scenario 1. This scenario examines the effectiveness of the SQG-based algorithm when the actual O-D demand distributions (skewed) are different from the assumed ones (normal distributions).

Scenario 5: This scenario explores the effectiveness of the SQG-based algorithm under incidents. Two severe incidents starting at time 5 minutes, are present for 30 and 35 minute durations respectively, and are assumed to block ninety-five and eighty-five percent of the associated link capacities, respectively. The incident locations are shown in Fig. 2.

5.4 Results

Fig. 4 shows the results of Scenario 1. The SQG average system travel time obtained in the fourth iteration is reached only after nine iterations by the MUCTDTA algorithm, indicating that the SQG solution converges substantially faster than the MUCTDTA solution. This can be explained by the simultaneous consideration of several feasible O-D demand realizations in each iterative step of the SQG-based algorithm, which enables the generation of a large vector of "good" feasible paths. It is important to note here that a larger path set does not necessarily imply a better solution, whether in the single O-D demand context or the stochastic DTA context. The relevance of the path set and the appropriate proportioning of vehicular assignments across the path set represent the critical factors. The SQG-based algorithm synergistically incorporates “good” feasible paths generated from several O-D demand patterns into the search process to determine one vector of path assignment proportions in each iterative step. By contrast, the path assignment vectors generated in each iterative step of deterministic DTA algorithms are based on a single O-D demand pattern, typically the mean O-D demand pattern. Of practical significance, the SQG-based algorithm provides a unique vector of path assignment proportions for all realizations while the MUCTDTA algorithm has a different vector for each realization.

Fig. 5 compares the SQG solution results of Scenario 1 (called single-SQG) and Scenario 2 (called multi-SQG). It indicates that there is no perceptible difference between them, suggesting that the stochastic quasi-gradients obtained using a single set of O-D demand realizations are sufficiently accurate. Since the use of new O-D demand realization sets to calculate quasi-gradients in each iteration requires substantially more resources compared to using just one set, this result has important practical implications.

The results of Scenario 3 are highlighted in Fig. 6. They mirror the conclusions from Scenario 1, and further indicate that robustness is conserved even when user class fractions vary across realizations. User class fractions are random variables. If their randomness can be captured and represented through distributions, it can be incorporated into the SQG solution. If it is difficult to obtain these fractions on-line, then their mean values can be used to obtain the solution, as in Scenario 1. The conservation of system performance under random user class fractions is an important property vis-à-vis realistic on-line implementation of the SQG solution.

Figs. 7 and 8 illustrate the performance of the SQG-based algorithm under the positive and negative skewed distributions, respectively. The robustness of the SQG solution is conserved even when the actual distributions are different from the assumed ones. This feature is also synergistic for the on-line implementation of the SQG solution. Also, akin to the previous results, the SQG-based algorithm performs better than the deterministic MUCTDTA algorithm.

Fig. 9 shows the results of Scenario 5. It should be noted that since the state-of-the-art of incident prediction cannot provide accurate incident occurrence predictions, Scenario 5 is not realistic in terms of real-world applicability. It is used only to examine the effectiveness of the SQG-based algorithm under incident situations. As before, it performs better than the MUCTDTA algorithm.

6. Concluding comments

This paper proposes an SQG-based algorithm to solve the DTA problem that explicitly incorporates the randomness in O-D demands. The off-line stochastic DTA problem is formulated and the associated theoretical aspects are illustrated. The solution algorithm is a generalization and a conceptually elegant extension of deterministic DTA solution methods that assume O-D demands to be known *a priori* for the entire planning horizon. Due to the lack of a guarantee of differentiability for the DTA objective function, quasi-gradients are estimated that are stochastic to incorporate O-D demand randomness. Due to the lack of explicit analytical forms for the objective function and some constraints because of traffic realism issues, simulation is used to estimate the quasi-gradients.

The SQG-based solution framework does not require O-D demand forecasts for a given day, although it is easy to incorporate such information if available. The approach is an off-line procedure due to the significant computational burden associated with the use of an iterative simulation-based solution method. However, experiments illustrate that the SQG solution performs as well as the deterministic DTA solution, on average. This is highly significant from a deployment standpoint as the SQG solution is independent of the O-D demand pattern on any given day. By contrast, deterministic

DTA solutions are not deployable as they require the O-D demand pattern for the entire planning horizon to be known before the time-dependent path assignment proportions for a given day are determined. Thereby, the SQG solution can be deployed without real-time update on any day on which the O-D demand pattern conforms to the historical demand distributions and significant network supply changes do not occur. However, incidents can significantly affect network supply. Peeta and Zhou (2002) propose a hybrid DTA deployment framework that incorporates a computationally efficient on-line component that responds to the unfolding demand and supply conditions on a given day. In this context, the SQG solution plays the role of a robust initial *a priori* solution for the on-line component.

The experiments suggest that the SQG solution is comparable to the MUCTDTA solution under various real-world scenarios. Also, the SQG-based algorithm converges faster than the equivalent MUCTDTA algorithm. The typical deterministic DTA solution proposed for deployment is based on the mean O-D demand pattern. Peeta and Zhou (2002) show that the SQG solution performs significantly better than the mean O-D demand based deterministic solution. They also show that it performs better than even the rolling horizon deployment framework in the absence of incidents.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 9702612. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Blum, J.R., 1954. Multidimensional stochastic approximation methods. Annals of Mathematical Statistics 25, 737-744.

Ermoliev, Y., 1976. Stochastic programming methods. (in Russian) Nauka, Moscow.

Ermoliev, Y., 1983. Stochastic quasigradient methods and their application to systems optimization. Stochastics 4, 1-37.

Ermoliev, Y., Norkin, V.I., 1998. Stochastic generalized gradient method for non-convex non-smooth stochastic optimization. Cybernetics and Systems Analysis 34 (2), 50-71.

Ermoliev, Y., Wets, R.J.B., (eds.), 1988. Numerical techniques for stochastic optimization. Springer-Verlag, Berlin.

Fabian, V., 1967. Stochastic approximation of minima with improved asymptotic speed. Annals of Mathematical Statistics 38, 191-200.

Gaivoronski, A., 1988. Stochastic quasigradient methods and their implementation. Chapter 16 in Numerical Techniques for Stochastic Optimization, Ermoliev, Y., Wets, R.J.B., (eds.), 313-352.

Hawas, Y.E., 1995. A decentralized architecture and local search procedures for real-time route guidance in congested vehicular traffic networks. Doctoral dissertation, The University of Texas at Austin.

Hawas, Y.E., Mahmassani, H.S., 1995. A decentralized scheme for real-time route guidance in vehicular traffic networks. presented at Second World Congress on Intelligent Transport Systems, Yokohama, Japan.

Jaillet, P., 1988. *A priori* solution of a traveling salesman problem in which a random subset of the customers are visited. Operations Research 36 (6), 929-936.

Jayakrishnan, R., Mahmassani, H.S., Hu, T-Y., 1995. An evaluation tool for advanced traffic information and management systems in urban networks. Transportation Research C 2 (2), 129-147.

Mikhalevich, V.S., Gupal, A.M., Norkin, V.I., 1987. Methods of non-convex optimization. (in Russian) Nauka, Moscow.

Pavlis, Y., Papageorgiou, M., 1999. Simple decentralized feedback strategies for route guidance in traffic networks. Transportation Science 33, 264-278.

Peeta, S., 1994. System optimal dynamic traffic assignment in congested networks with advanced information systems. Doctoral dissertation, The University of Texas at Austin.

Peeta, S., Bulusu, S., 1999. A generalized singular value decomposition approach for consistent on-line dynamic traffic assignment. Transportation Research Record 1667, 77-87.

Peeta, S., Mahmassani, H.S., 1995a. Multiple user classes real-time traffic assignment for on-line operations: a rolling horizon solution framework. Transportation Research C 3 (2), 83-98.

Peeta, S., Mahmassani, H. S., 1995b. System optimal and user equilibrium time-dependent traffic assignment in congested networks. Annals of Operations Research 60, 81-113.

Peeta, S., Zhou, C., 1999. Robustness of the offline *a priori* stochastic dynamic traffic assignment solution for online operations. Transportation Research C 7 (5), 281-303.

Peeta, S., Zhou, C., 2002. A hybrid deployable dynamic traffic assignment framework for robust online route guidance. Networks and Spatial Economics, 2 (3).

Peeta, S., Ziliaskopoulos, A., 2001. Foundations of dynamic traffic assignment: the past, the present and the future. Networks and Spatial Economics 1 (3/4), 233-266.

Pflug, G.Ch., 1988. Stepsize rules, stopping times, and their implementation in stochastic quasigradient algorithms. Chapter 17 in Numerical Techniques for Stochastic Optimization, Ermoliev, Y., Wets, R.J.B., (eds.), 353-372.

Powell, W.B., Sheffi, Y. 1982. The convergence of equilibrium algorithms with predetermined step size. Transportation Science 16 (l), 45-55.

Zhou, C., 2002. Stochastic dynamic traffic assignment for robust on-line operations under real-time information systems. Doctoral dissertation, Purdue University.

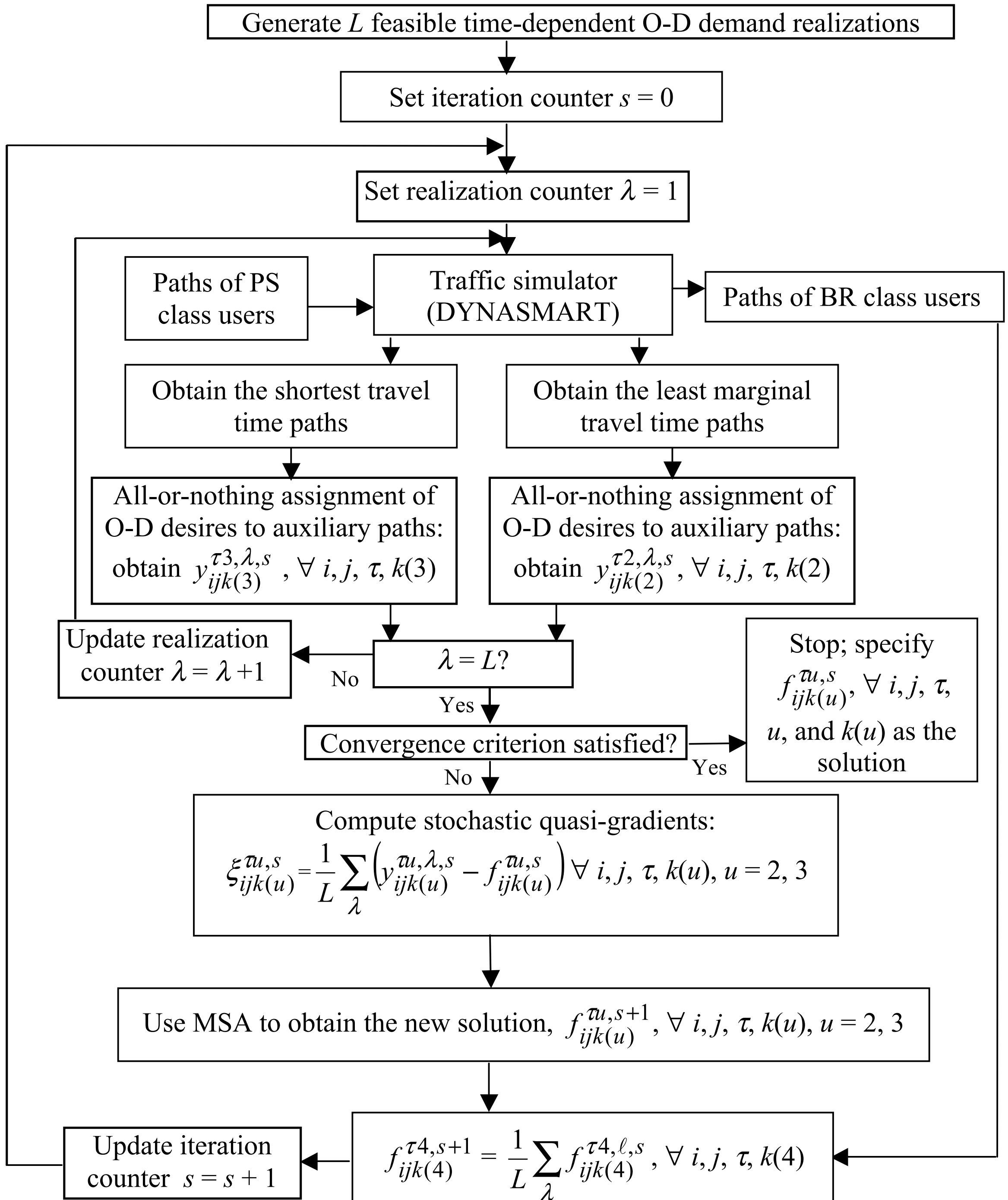


Fig. 1. The SQG-based solution algorithm

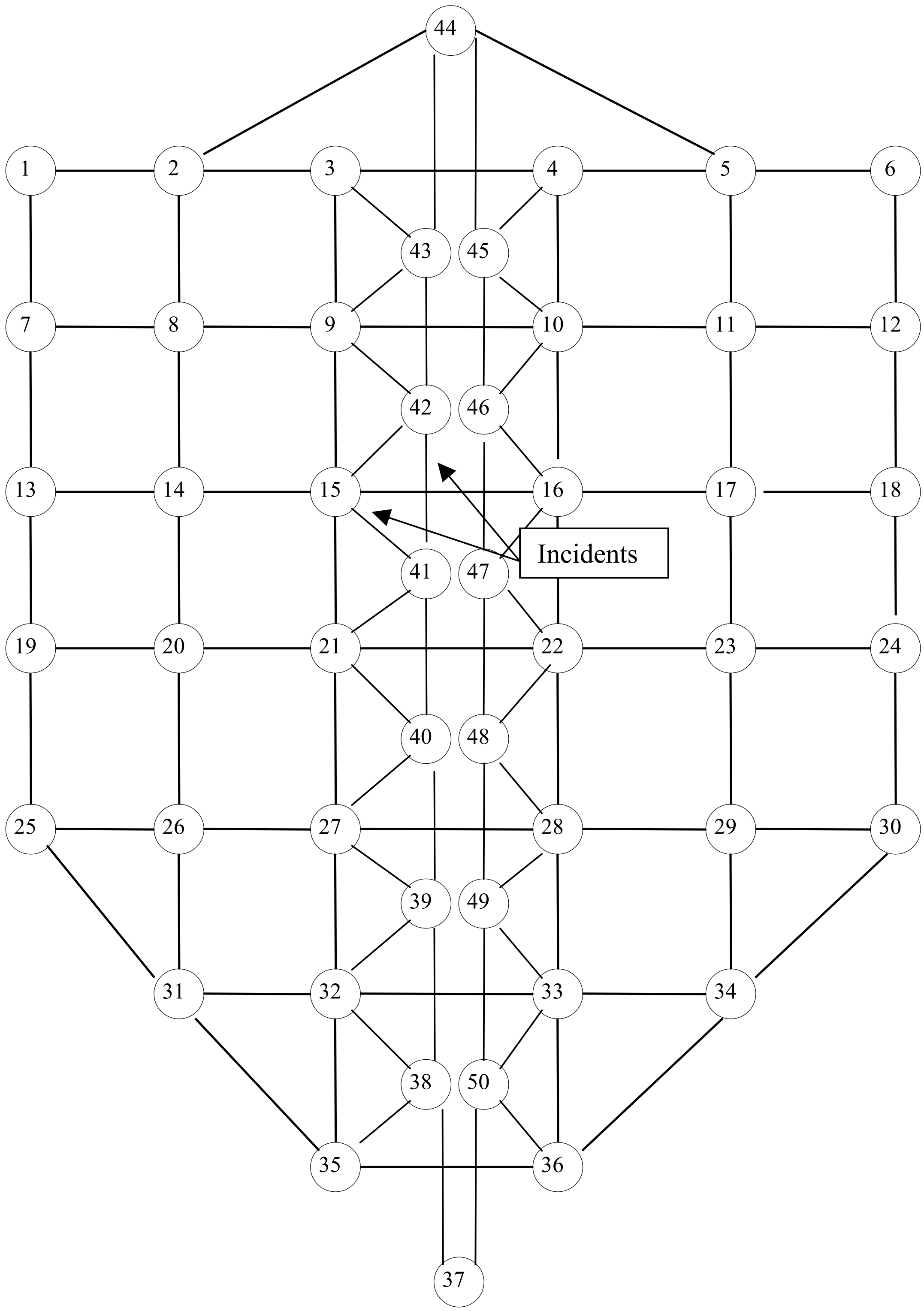
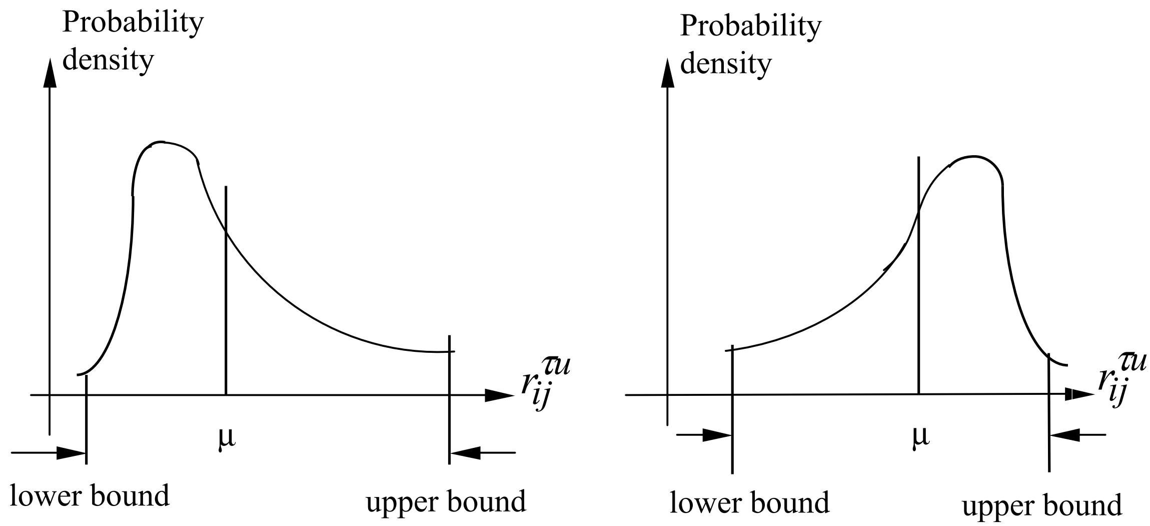


Fig. 2. Network structure



(a) Positive skewed distribution

(b) Negative skewed distribution

Fig. 3. Skewed O-D demand distributions

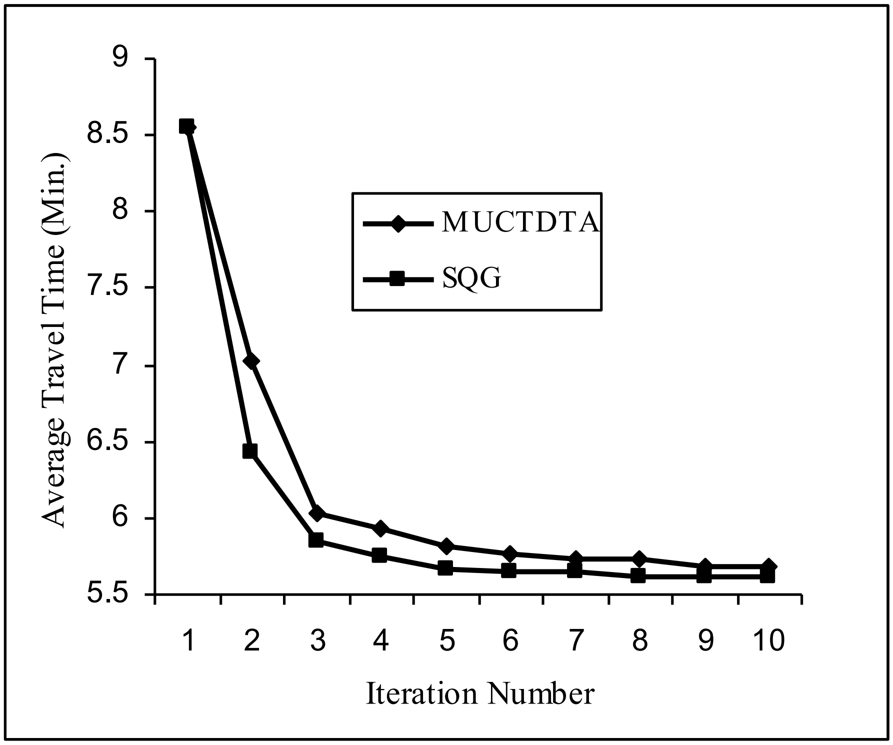


Fig. 4. Solution convergence

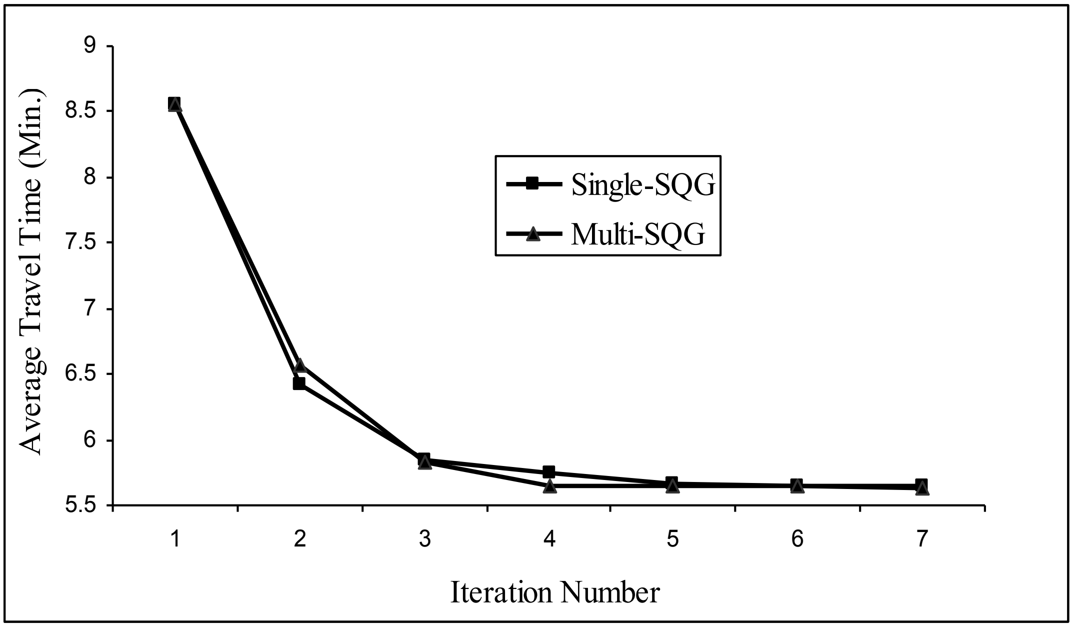


Fig. 5. SQG convergence under single and multiple sets of O-D demand realizations

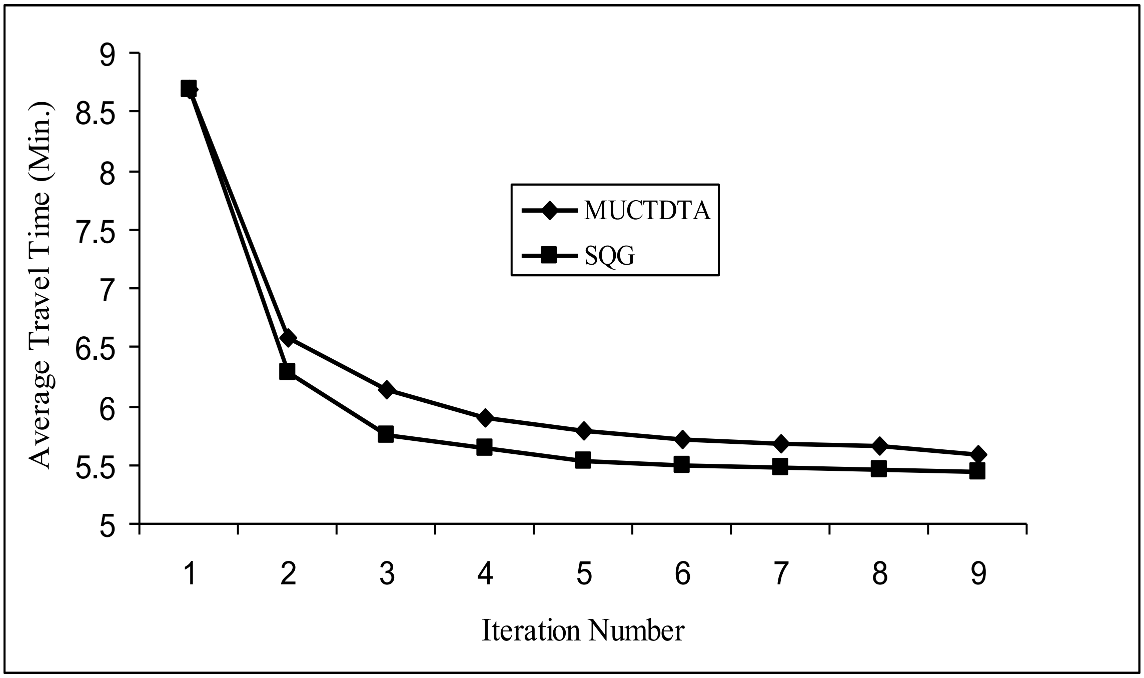


Fig. 6. Solution convergence under random user class fractions

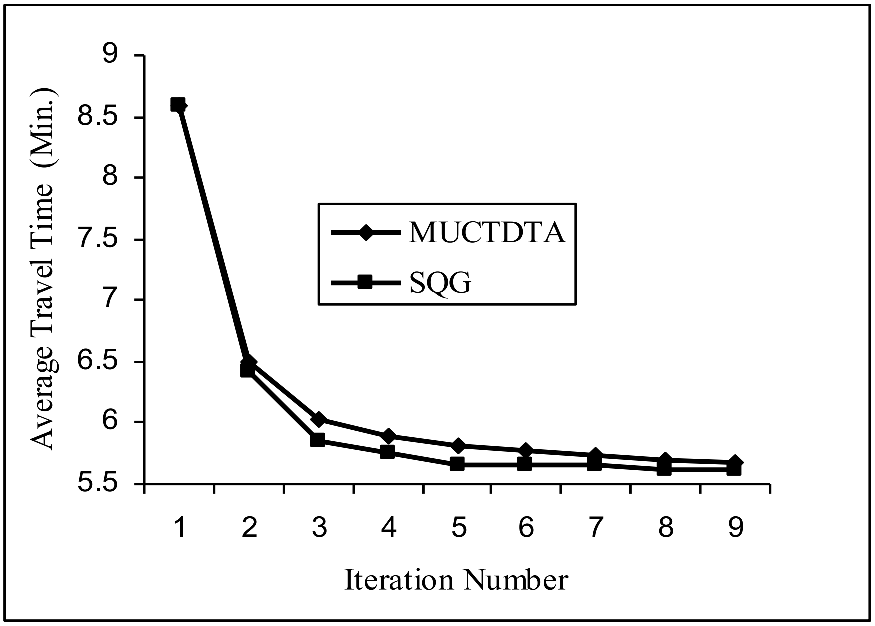


Fig. 7. Solution convergence under positive skewed distributions

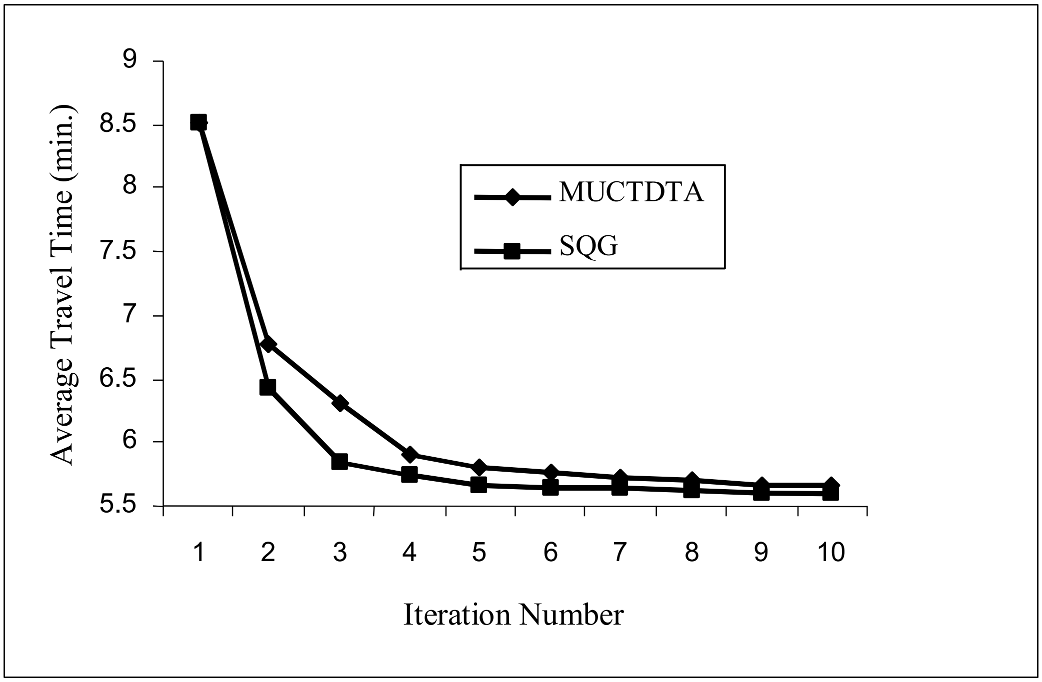


Fig. 8. Solution convergence under negative skewed distributions

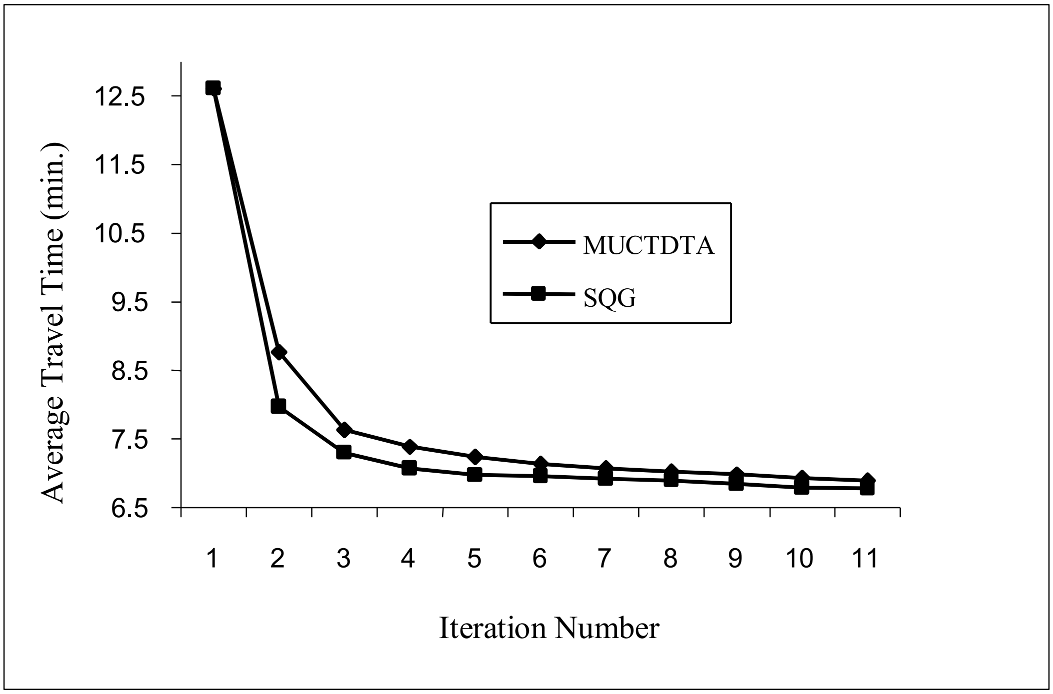


Fig. 9. Solution convergence under incidents