Robust dynamic traffic assignment for single destination networks under demand and capacity uncertainty

(Article begins on next page)

Robust Dynamic Traffic Assignment for Single Destination Networks under Demand and Capacity Uncertainty

Giuseppe C. Calafiore[a,c], Marco Ghirardi[b] and Alessandro Rizzo[a]

[a]Department of Electronics and Telecommunications (DET), Politecnico di Torino, Italy; [b]Department of Management and Production Engineering (DIGEP), Politecnico di Torino, Italy; [c]IEIIT-CNR, Italy

ARTICLE HISTORY
Compiled June 21, 2019

ABSTRACT
In this paper we discuss the system-optimum dynamic traffic assignment (SO-DTA) problem in the presence of time-dependent uncertainties on both traffic demands and road link capacities. Building on an earlier formulation of the problem based on the cell transmission model (CTM), the SO-DTA problem is robustly solved, in a probabilistic sense, within the framework of random convex programs (RCPs). Differently from traditional robust optimization schemes, which find a solution that is valid for all the values of the uncertain parameters, in the RCP approach we use a fixed number of random realizations of the uncertainty, and we are able to guarantee a priori a desired upper bound on the probability that a new, unseen realization of the uncertainty would make the computed solution unfeasible. The particular problem structure and the introduction of an effective domination criterion for discarding a large number of generated samples enables the computation of a robust solution for medium- to large-scale networks, with low desired violation probability, with a moderate computational effort. The proposed approach is quite general and applicable to any problem that can be formulated through a linear programming model, where the stochastic parameters appear in the constraint constant terms only. Simulation results corroborate the effectiveness of our approach.

KEYWORDS
Traffic assignment; Cell transmission model; Robust optimization; Scenario optimization; Robustness analysis

1. Introduction

Traffic assignment problems deal with the management of vehicle flows in road networks and aim at determining flows according to some suitable optimality criterion, while satisfying practical constraints, such as capacity and sustainable traffic limitations on the network links (Bell & Iida, 1997; Friesz, Kwon, & Bernstein, 2007; Peeta, 1994; Peeta & Ziliaskopoulos, 2001). Notably, dynamic traffic assignment (DTA) aims at allocating traffic flows based on a time-varying origin-destination (OD) demand to a set of paths established on a network of roads. This problem, originally formulated in the 1950s, has been tackled under two different optimality perspectives. The user equilibrium, or Wardrop equilibrium allocation (UE-DTA), where the objective is a

CORRESPONDING AUTHOR. A. Rizzo. Email: alessandro.rizzo@polito.it

selfish optimization of the individual travel time (Wardrop, 1952), and the system-optimal allocation (SO-DTA), which pursues a global optimization of the travel times, toward the satisfaction of a social optimality objective, possibly at the expenses of the individual optimality (Como, Lovisari, & Savla, 2016; Merchant & Nemhauser, 1978a, 1978b; Samaranayake et al., 2015).

The cell transmission model (CTM) has been one of the most used models to formalize and solve the SO-DTA problem (Daganzo, 1994, 1995). In particular, the formulation made by Ziliaskopoulos (Ziliaskopoulos, 2000), which uses some relaxations of the original constraints to yield a linear program, is the one adopted in this paper to tackle the SO-DTA problem in the presence of uncertainty both in the traffic demand and in the road link capacities. Indeed, a critical issue in mathematical programming formulations of traffic assignment problems relates to the fact that the problem input data, such as OD demand levels or effective capacities of links, are seldom known in advance with precision. If nominal or expected values of such parameters are used in an optimization model, the resulting optimal solution is likely to critically depend on these values and to potentially yield poor system performance when the actual system parameters deviate significantly from the nominal ones (Rasouli & Timmermans, 2012; Zhao & Kockelman, 2002).

The most commonly studied sources of uncertainties in DTA are variations in travel demand (Garrett & Wachs, 1996; Oppenheim et al., 1995; Yang, Chen, Xu, & Wong, 2013) and in road capacities (Chung, Yao, Xie, & Thorsen, 2011; Sun, Gao, & Zhao, 2014). Several researchers agree on the fact that characterizing those uncertainties is a key point in modeling DTA (Knoop, Hoogendoorn, & van Zuylen, 2008; A. Yazici, Kamga, & Ozbay, 2015; M. Yazici & Ozbay, 2007). For instance, in (Smith, Qin, & Venkatanarayana, 2003), a data analysis campaign leads to the observation that a good fit for capacity reduction due to the occurrence of incidents is a Beta distribution. Statistical characterization of uncertainties is a research topic on its own, and is out of the scope of this paper. However, as it will be shown later, one of the strengths of our approach is the capability to treat any statistical distribution that is assumed on the parameter variations.

Standard approaches for dealing with uncertainty include numerically-intensive techniques that solve a large number of problem instances for randomly generated parameter patterns, followed by adjustments of the obtained solution on the basis of on-line data, when available, see, e.g., (Peeta & Zhou, 1999). In (Ukkusuri, Mathew, & Waller, 2007), a static assignment problem with uncertain OD demands is formulated and solved by means of genetic programming. The approach provides near-optimal solutions to the network design problem. However, genetic algorithms are notoriously computationally intensive, and in the mentioned application parallelization and approximation techniques are put forward to deal with large scale networks. In (Waller & Ziliaskopoulos, 2006), the dynamic SO-DTA problem is tackled in the case of time-dependent demands, modeled as random variables with known probability distributions. However, the problem is solved only on a nominal scenario. In (Kachroo & Ozbay, 2006), H_∞ feedback theory is leveraged to solve a real-time robust SO-DTA problem. However, the method is applied to a relatively small network and its application to realistic, large scale networks may entail a heavy computational burden. In (Florian, Mahut, & Tremblay, 2008), the problem is tackled through a micro-simulation approach able to face large networks. A scenario-based robust optimization approach is used in (Karoonsoontawong & Waller, 2007) on a model formulated as a two-stage stochastic program. Their work is however limited to small, not realistically-sized networks, and considers demand uncertainty only. Other works (e.g., (Quian, Shen, &

Zhang, 2012) or (Lim, Rungta, & Baharnemati, 2015)) consider a path-based formulation instead of a cell-based one. In (Ben-Tal, Chung, Mandala, & Yao, 2011), the AARC method proposed in (Ben-Tal, Goryashko, Guslitzer, & Nemirovski, 2004) is applied to the SO-DTA problem where decisions are taken at each time interval. Alternative approaches, based on Petri networks (Julvez & Boel, 2010; Kim, Kato, Okuma, & Narikiyo, 2008) or on dynamic network flow models (Ma, Cui, & Cheng, 2004) do not explicitly account for uncertainties in the modeling phase (Ma et al., 2004), or try to compensate their effect with techniques inspired to model predictive control (Julvez & Boel, 2010; Kim et al., 2008), which is computationally intensive, requiring the computation of the optimal control input at each time step. In (Tuerprasert & Aswakul, 2010) a CTM generalization for the multiclass vehicles case is presented. In (Islam, Vu, Panda, & Ngoduy, 2018) a linear programming model for optimizing the SO-DTA is proposed, where the control signal is designed for minimizing the vehicle-discharged emissions.

In most of the mentioned works, the stochastic data accounted for are only the traffic demands, and the related uncertainty is usually modeled through a uniform distribution bounded in an uncertainty box. Moreover, these methods are usually computationally intensive and, when robustness features with respect to the uncertainty sources are considered, their solutions lean towards the satisfaction of the worst case, limiting in fact the effectiveness in managing the road network in the vast majority of cases. In fact, in the classic framework of robust optimization (Ben-Tal, Ghaoui, & Nemirovski, 2000; Ben-Tal & Nemirovski, 1998), uncertain parameters in the mathematical program are assumed to be unknown-but-bounded quantities and solutions are sought such that relevant constraints are satisfied *for all* possible values of the parameters, including the worst-case scenarios. Thus, a cost objective that is optimal in this robust sense (for instance, the total travel time), provides a performance level that is guaranteed for all the considered scenarios.

In this work, we propose a strategy to robustly solve, in a probabilistic sense, the SO-DTA problem, based on the CTM (Ziliaskopoulos, 2000), in presence of uncertainty on both the traffic demand and the road link capacities. To this aim, we leverage random convex programming (RCP) (Calafiore, 2010), a robust optimization technique based on the solution of convex optimization problems subject to a *finite number of random constraints.* The strength of the proposed approach relies in the possibility of accounting for any given or even unknown distribution (as long as samples from such distribution are available) of the stochastic parameter set included in the constraints of the problem, and to find a solution that guarantees a small level of *violation probability*, that is, the probability that the solution is no longer optimal if a further random constraint is added to the problem. This implies that: *i)* the solution obtained through our approach will be valid for *most of* the scenarios occurring in the road network; *ii)* the solution will not be heavily conditioned by the inclusion of worst-case scenarios that tend to occur with a very low probability; and *iii)* the probability with which a situation not considered in the optimization problem and for which the obtained solution is unfeasible is known and can be tuned at will through a suitable choice of the number of random constraints to be generated when the optimization problem is solved.

We also show that the proposed approach is computationally affordable, and that it can be used in applications where the computational time is a critical factor. Further, we enhance the efficiency of the proposed optimization strategy by adopting an effective constraint removal procedure. Such procedure is in general computationally hard for a generic convex problem, but can be applied here in an affordable time, by exploiting the specific model structure. The technique used for deriving this model is a novel result valid not only for this specific problem, but for all the models with the same structure, namely linear programming models with uncertainties in the constant terms of the constraints.

The rest of the paper is structured as follows. In Section 2, the SO-DTA cell model is introduced, and a new equivalent formulation with a reduced number of decision variables is derived. In Section 3, the probabilistic version of the problem is considered and the corresponding RCP solution model is derived. All the computational results on problems with increasing sizes are presented in Section 4. Finally, our conclusions are drawn in Section 5.

2. Problem setup

We start by reviewing the linear programming formulation of the SO-DTA problem introduced in (Ziliaskopoulos, 2000), based on the CTM defined in (Daganzo, 1994, 1995). In this context, a *link* represents a path connecting two distinct points on a planar space, such as a street in a streets grid. A *cell* is a segment of a link having length equal to the distance traveled by a vehicle during a fixed time interval τ if no other vehicles are on the destination node (free-flowing vehicle). A *network* is described as the set $\mathcal{C}$ of interconnected cells, indexed by $i = 1, \ldots, C$, with $C \doteq |\mathcal{C}|$ and $|\cdot|$ indicating the set cardinality operator. The system state $x(t) \in \mathbb{N}_0^C$ is a vector whose components $x_i(t) \in \mathbb{N}_0$ represent the number of vehicles contained in each cell $i \in \mathcal{C}$ at the discrete time $t \in \mathbb{N}_0$, with $t = 0, 1, \ldots, T$, where $T \in \mathbb{N}$ is the time horizon over which the system is analyzed. Besides its state, a cell is characterized by several parameters and variables. In particular, we indicate with $\Gamma(i)$ and $\Gamma^{-1}(i)$, the index sets of cells following or preceding cell $i \in \mathcal{C}$, respectively (see Table 1). The set $\mathcal{C}_{\mathrm{OR}} \subseteq \mathcal{C}$ of *ordinary cells* comprises the cells that have exactly one predecessor and one successor, i.e., $\mathcal{C}_{\mathrm{OR}} = \{i \in \mathcal{C} : |\Gamma(i)| = |\Gamma^{-1}(i)| = 1\}$. *Diverging cells* have exactly one predecessor and more than one successor, i.e., their related set $\mathcal{C}_{\mathrm{DV}} \subseteq \mathcal{C}$ is defined as $\mathcal{C}_{\mathrm{DV}} = \{i \in \mathcal{C} : |\Gamma(i)| > 1, |\Gamma^{-1}(i)| = 1\}$. *Merging cells* have more than one predecessor and exactly one successor, i.e., their related set $\mathcal{C}_{\mathrm{MR}} \subseteq \mathcal{C}$ is defined as $\mathcal{C}_{\mathrm{MR}} = \{i \in \mathcal{C} : |\Gamma(i)| = 1, |\Gamma^{-1}(i)| > 1\}$. *Source cells* do not have predecessors and have exactly one successor, i.e., their related set $\mathcal{C}_{\mathrm{SR}} \subseteq \mathcal{C}$ is defined as $\mathcal{C}_{\mathrm{SR}} = \{i \in \mathcal{C} : |\Gamma(i)| = 1, |\Gamma^{-1}(i)| = 0\}$, and *sink cells* have exactly one predecessor and do not have successors, i.e., their related set $\mathcal{C}_{\mathrm{SN}} \subseteq \mathcal{C}$ is defined as $\mathcal{C}_{\mathrm{SN}} = \{i \in \mathcal{C} : |\Gamma(i)| = 0, |\Gamma^{-1}(i)| = 1\}$. The traffic demand is associated with source cells, through the variable $d_i(t)$, which represents the demand (inflow) at cell i in the t-th time interval. To simplify the notation, we consider the existence of a variable $d_i(t)$ for each cell, although this variable is set to zero for all non-source cells. Initial cell occupation in the system is set by the initial condition $x(0) = \xi$. Sink cells are assumed to have infinite holding capacity and to allow infinite input flows ($N_i(t) = \infty$, $Q_i(t) = \infty$, $\forall t$ and $\forall i \in \mathcal{C}_{\mathrm{SN}}$), and source cells are also assumed to have infinite holding capacity ($N_i(t) = \infty$, $\forall t$ and $\forall i \in \mathcal{C}_{\mathrm{SR}}$).

In the proposed formulation, the objective of the SO-DTA problem is to determine network states $x_i(t)$ and flows $y_{ij}(t)$ that minimize the total travel time in the network, that is the sum of the travel times experienced by all vehicles in the network. Since $x_i(t)$ vehicles stay in the i-th cell for τ units of time and sink cells do not contribute

Symbol	Description
$N_i(t)$	maximum number of vehicles (holding capacity) in cell i at time t
$Q_i(t)$	maximum number of vehicles that can flow in or out cell i during the t-th time interval
$\delta_i(t)$	free-flow to backward propagation speed ratio for cell i at time t
$\Gamma(i)$	index set of successor cells to cell i
$\Gamma^{-1}(i)$	index set of predecessor cells to cell i
$d_i(t)$	demand (inflow) of vehicles at cell i in the t-th time interval
$x_i(t)$	number of vehicles in cell i in the t-th time interval
$y_{ij}(t)$	number of vehicles moving from cell i to cell j in the t-th time interval

Table 1. Cell parameters and variables

to total system time, the cost objective function to be minimized is

$$J = \sum_{t=1}^{T} \sum_{i \in \mathcal{C} \setminus \mathcal{C}_{\mathrm{SN}}} \tau x_i(t). \tag{1}$$

To tackle this problem, we start by considering the linear programming (LP) model for the single-destination SO-DTA problem presented in (Ziliaskopoulos, 2000):

$$\min J \tag{2}$$

$$\text{subject to, } \forall i \in \mathcal{C}, t = 1, \dots, T :$$

$$x_i(t) = x_i(t-1) + \textstyle\sum_{j \in \Gamma^{-1}(i)} y_{ji}(t-1) - \sum_{j \in \Gamma(i)} y_{ij}(t-1) + d_i(t-1) \tag{3}$$

$$\textstyle\sum_{j \in \Gamma(i)} y_{ij}(t) \leq x_i(t) \tag{4}$$

$$\textstyle\sum_{j \in \Gamma(i)} y_{ij}(t) \leq Q_i(t) \tag{5}$$

$$\textstyle\sum_{j \in \Gamma^{-1}(i)} y_{ji}(t) \leq Q_i(t) \tag{6}$$

$$\textstyle\sum_{j \in \Gamma^{-1}(i)} y_{ji}(t) \leq (N_i(t) - x_i(t))\delta_i(t) \tag{7}$$

$$x_i(t) \geq 0 \tag{8}$$

$$y_{ij}(t) \geq 0, \tag{9}$$

with initial conditions $x_i(0) = \xi_i$, $y_{ij}(0) = 0$, $\forall i, j \in \mathcal{C}$, and where $d_i(t) = 0$ for all non-source nodes. Equation (3) represents the flow balance at cells, equations (4–5) constrain the output flow from cell i to be smaller than $\min(x_i(t), Q_i(t))$, whereas equations (6–7) impose that the input flow to cell i must not exceed the maximum input flow capacity $Q_i(t)$ and a fraction of the remaining holding capacity at the target node. The last two constraints impose that all the states and flows must be non-negative.

Some remarks are in order. Constraint (7) formalizes a congestion model based on a triangular dependency between the link capacity and the remaining holding capacity at the destination node (Daganzo (1995)). Other studies model traffic congestion with a different (usually concave) dependency, corresponding to a model in which $\delta_i(t)$ is not constant, but depends on the destination holding capacity (Arbib, Muccini, and Moghaddam (2018)). In this case, it is still possible to model the CTM as a LP through a proper interval linearization. This would not affect the applicability of the methods derived in this paper. We also observe that the CTM model is presented in the literature for single destination networks. In fact, CTM does not allow to impose the choice of a destination node to each vehicle, unless a FIFO mechanism is implemented, resulting in a non-convex model (Carey, Bar-Gera, Watling, and Balijepalli (2014)). A slight extension of the method can be put forward when multiple destination cells are present, but the choice of the destination cell is not relevant, so that it can be assumed that all the destination cells are connected to a unique virtual destination cell. Consider for instance the use of the model for planning evacuation in case of emergency (fires, floods, etc.). In this case, it is not important which of the multiple destinations is reached by a vehicle, but only that a safe destination cell is reached. Thus, we can consider that all the set of destination nodes are equivalent and connected with a single virtual *out of danger* cell (Bayram (2016)). More recently, Long, Wang, and Szeto (2018) propose a alternative intersection-movement formulation for the simultaneous route and departure time choice (DSO-SRDTC), which is able to consider both vehicle holding constraints and FIFO constraints. However, the latter model leads to nonconvex nonlinear programming problems. Long and Szeto (2019) introduce a Branch and Bound algorithm for the link-based SO-DTA, which is able to capture FIFO constraints and optimally solve the problem for small dimension networks, up to 43 nodes.

In the following subsections, we present a novel reformulation of the CTM model, which allows us to reduce the number of model variables. This reduction will result to be very useful later in reducing the number of samples to be generated in order to ensure a certain probability of constraint violation.

2.1. Model reformulation in compact matrix form

The LP model previously defined can be recast in a compact matrix form. To this aim, let $Y(t) \in \mathbb{N}_0^{C,C}$ be a matrix with non-negative entries and such that $Y_{i,j}(t) = y_{ij}(t) = 0$ whenever $j \notin \Gamma(i)$, and denote by $\mathcal{Y}$ the set of non-negative matrices having this sparsity pattern. Then, the total flow coming out of cell i at t is $\sum_{j=1}^{C} Y_{i,j}(t)$, while the total flow entering cell i at t is $\sum_{j=1}^{C} Y_{j,i}(t)$. Let also form the following vectors by stacking node-related quantities defined in Table 1: $d(t) = [d_1(t) \cdots d_C(t)]^\top$, $Q(t) = [Q_1(t) \cdots Q_C(t)]^\top$, $N(t) = [N_1(t) \cdots N_C(t)]^\top$, $\Delta(t) = \mathrm{diag}(\delta_1(t), \dots, \delta_C(t)$. Finally, let $e \in \mathbb{N}_0^C$ be a vector such that $e_i = 1$ if $i \notin \mathcal{C}_{\mathrm{SN}}$ and $e_i = 0$ otherwise. Denoting with $\mathbf{1}$ a vector of ones of length C, the SO-DTA LP can be cast in the equivalent matrix format

$$\min_{x(t)\geq 0, Y(t)\in\mathcal{Y}} \sum_{t=1}^{T} e^\top x(t) \tag{10}$$

$$\text{subject to, for } t = 1, \dots, T:$$

$$x(t) = x(t-1) + [Y^\top(t-1) - Y(t-1)]\mathbf{1} + d(t-1) \tag{11}$$

$$Y(t)\mathbf{1} \leq x(t) \tag{12}$$

$$Y(t)\mathbf{1} \leq Q(t) \tag{13}$$

$$Y^\top(t)\mathbf{1} \leq Q(t) \tag{14}$$

$$Y^\top(t)\mathbf{1} \leq \Delta(t)[N(t) - x(t)]. \tag{15}$$

2.2. Reduction of the number of variables

The problem formulated in the previous sections has $T(C^2 + C - |\mathcal{C}_{\rm SN}|)$ variables. In the following, we present a strategy to reformulate the problem reducing the number of variables. Let $f_{\rm in}(t) \doteq Y^\top(t)\mathbf{1}$, $f_{\rm out}(t) \doteq Y(t)\mathbf{1}$ denote the vectors of total inflows and total outflows at cells, respectively, and $f_{{\rm in},i}(t)$, $f_{{\rm out},i}(t)$ their i-th components. Then, applying Eq. (11) recursively, for all $t = 1, \ldots, T$, we obtain

$$x(t) = x(0) + \sum_{\nu=0}^{t-1} \left(f_{\rm in}(\nu) - f_{\rm out}(\nu)\right) + \sum_{\nu=0}^{t-1} d(\nu), \tag{16}$$

with $x(0) = \xi$, $f_{\rm in}(0) = f_{\rm out}(0) = 0$. We observe that, making the standard assumption that diverging and merging cells cannot be directly connected (Ziliaskopoulos, 2000), the knowledge of the total inflows and outflows at cells is sufficient for reconstructing all the individual flows in $Y(\cdot)$. Notably, source, sink, and ordinary cells have only one inflow or outflow link, which would obviously have a corresponding entry in $f_{\rm in}(\cdot)$, $f_{\rm out}(\cdot)$. Diverging cells have multiple output flows, but each of them is uniquely determined by the corresponding inflow at the successor cells. Similarly, merging cells have multiple inflows, each of them corresponding to the single outflow of a predecessor cell.

Therefore, substituting (16) into (10)–(15), we obtain a LP where the $x(\cdot)$ variables have been eliminated and only the total flows $f_{\rm in}(t)$, $f_{\rm out}(t)$ appear. The individual flows $Y(t)$ can be reconstructed a posteriori on the basis of the network topology. If we let the cumulative demand up to time t be $\psi(t) \doteq \sum_{\nu=0}^{t} d(\nu)$, and the model objective function be $\gamma = \sum_{t=1}^{T} e^\top x(t)$, then our LP takes the following form:

$$\min_{f_{\rm in}\geq 0, f_{\rm out}\geq 0, \gamma \geq 0} \gamma \tag{17}$$

subject to:

$$\sum_{t=1}^{T} e^\top \xi + \sum_{t=1}^{T} \sum_{\nu=0}^{t-1} e^\top \left(f_{\rm in}(\nu) - f_{\rm out}(\nu)\right) + \sum_{t=1}^{T} e^\top \psi(t-1) \leq \gamma \tag{18}$$

$$f_{\rm out}(t) \leq \xi + \sum_{\nu=0}^{t-1} \left(f_{\rm in}(\nu) - f_{\rm out}(\nu)\right) + \psi(t-1) \qquad \forall t \in 1, \ldots, T \tag{19}$$

$$f_{\rm out}(t) \leq Q(t) \qquad \forall t \in 1, \ldots, T \tag{20}$$

$$f_{\rm in}(t) \leq Q(t) \qquad \forall t \in 1, \ldots, T \tag{21}$$

$$f_{\rm in}(t) \leq \Delta(t) \Big(N(t) - \xi + \sum_{\nu=0}^{t-1} \left(f_{\rm in}(\nu) - f_{\rm out}(\nu)\right) + \psi(t-1)\Big) \qquad \forall t \in 1, \ldots, T \tag{22}$$

$$\sum_{j\in\Gamma(i)} f_{{\rm in},j}(t) = f_{{\rm out},i}(t) \qquad \forall t \in 1, \ldots, T; \forall i \in \mathcal{C}_{\rm DV} \tag{23}$$

$$\sum_{i\in\Gamma^{-1}(j)} f_{{\rm out},i}(t) = f_{{\rm in},j}(t) \qquad \forall t \in 1, \ldots, T; \forall j \in \mathcal{C}_{\rm MR} \tag{24}$$

$$f_{{\rm out},i}(t) = f_{{\rm in},j}(t) \qquad \forall t \in 1, \ldots, T; \forall (i,j) \in \{(\mathcal{C}_{\rm OR}, \mathcal{C}_{\rm OR}), (\mathcal{C}_{\rm MR}, \mathcal{C}_{\rm DV})\}, i \in \Gamma^{-1}(j) \tag{25}$$

where constraints (19)-(22) are the result of the substitution of (16) in (11)-(15), while constraints (23), (24) and (25) are necessary to keep information about the flows between cells.

We observe that this new formulation uses $2 \cdot C \cdot T + 1$ variables only. Forming the vector

$$X = [f_{\rm in}(1), \ldots, f_{\rm in}(T), f_{\rm out}(1), \ldots, f_{\rm out}(T), \gamma]^\top,$$

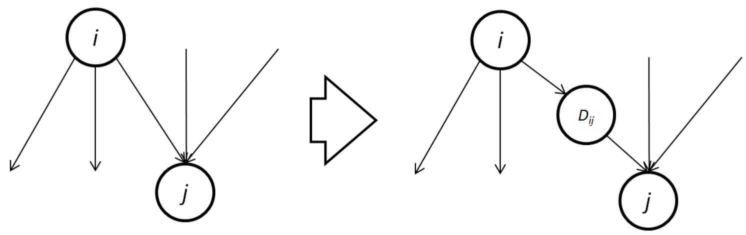

Figure 1. Transformation for directly connected diverging and merging cells.

the previous LP can be written in compact form as

$$\min c^\top X \quad \text{subject to:} \tag{26}$$
$$AX \leq B + E\Theta, \tag{27}$$

where the dependence on time has been dropped to enhance readability, $c^\top \doteq [0\,0 \cdots 0\,1]$, and A, B, E are coefficient matrices that may be easily inferred from (17)–(25). In particular, if we consider stochastic values for demands d and capacities N and Q, the constant terms of constraints (27) are formed through the sum of two contributions, where vector B contains all the deterministic terms and vector Θ contains all the stochastic ones.

If the cumulative demand $\psi(\cdot)$ and the cell capacities $N(\cdot)$ and $Q(\cdot)$ are exactly known in advance, the SO-DTA problem is then solved by the LP (26)-(27) in the total flux variables. In the next section, we will instead consider the case of uncertainty in those input data, and seek for solutions that are guaranteed to be robust in a probabilistic sense, using the RCP approach.

As already stated, the presented variable reduction relies on the assumption that diverging and merging cells cannot be directly connected. In case such condition does not hold, the reduction strategy can be put forward by adding dummy ordinary cells D_{ij} in between the generic diverging node i and merging node j, as illustrated in Figure 1. These additional cells do not contain vehicles in any finite time interval, and are represented in the LP model by additional variables $f_{D_{ij}}(t)$, which appear as both the input and the output flow of such cells at the same time, thus not changing the network behavior. Hence, the following procedure can be adopted:

- For each couple on nodes $i, j : i \in \mathcal{C}_{\text{DV}}, j \in \mathcal{C}_{\text{MR}}$, variables $f_{D_{ij}}(t) \geq 0, t \in 1, \ldots, T$ are added to the model.
- Constraints (23) and (24) are modified as follows:

$$\textstyle\sum_{j\in\Gamma(i), j\notin\mathcal{C}_{\text{MR}}} f_{\text{in},j}(t) + \sum_{j\in\Gamma(i), j\in\mathcal{C}_{\text{MR}}} f_{D_{ij}}(t) = f_{\text{out},i}(t) \qquad \forall t \in 1, \ldots, T; \forall i \in \mathcal{C}_{\text{DV}} \tag{28}$$
$$\textstyle\sum_{i\in\Gamma^{-1}(j), i\notin\mathcal{C}_{\text{DV}}} f_{\text{out},i}(t) + \sum_{i\in\Gamma^{-1}(j), i\in\mathcal{C}_{\text{DV}}} f_{D_{ij}}(t) = f_{\text{in},j}(t) \quad \forall t \in 1, \ldots, T; \forall j \in \mathcal{C}_{\text{MR}} \tag{29}$$

3. Traffic assignment under uncertainty

In the following, we will provide a formulation for the robust solution of problem (26)-(27). At first (Sections 3.1 and 3.2) the problem is approached by considering as robustness criterion the classical *worst-case robustness* one. Then, the proposed randomized scenario based method is presented in the subsequent sections.

3.1. Robust (worst-case) optimization

A generic formulation for the robust solution of problem (26)-(27) with uncertainties can be given as follows. Consider the optimization problem

$$\min c^\top X \quad \text{subject to:} \tag{30}$$

$$AX \leq B + E\Theta, \text{with } \Theta \in \mathcal{U} \subset \mathbb{R}^m. \tag{31}$$

Vector Θ represents the source of uncertainty of the problem, and the only available information is that it belongs to a given *uncertainty set* $\mathcal{U} \subset \mathbb{R}^m$. In the framework of robust optimization, constraint (31) must be always satisfied, irrespective of the specific realization of the uncertainty $\Theta \in \mathcal{U}$. In this context, we say that X is a robustly feasible solution to the uncertain optimization problem (30)-(31) if it satisfies constraint (31) for all possible realizations of Θ, i.e.,

$$AX \leq B + E\Theta, \ \forall \Theta \in \mathcal{U}. \tag{32}$$

A robustly optimal solution X^* for (30)-(31) is a robustly feasible solution that minimizes the objective, that is

$$X^* = \min_X \left\{ \sup_{\Theta \in \mathcal{U}} c^\top X : AX \leq B + E\Theta, \ \forall \Theta \in \mathcal{U} \right\}. \tag{33}$$

Generic robust optimization problems are not always computationally tractable, often resulting in NP-hard formulations, see, e.g., (Ben-Tal & Nemirovski, 1998).

3.2. A simple sub-case

Although problem (33) is computationally hard in general, there are special cases in which it can be solved efficiently. This occurs, for instance, if the uncertainty vector is bounded to lie entry-wise within independent intervals. Indeed, if we assume that the cumulative demand $\psi(\cdot)$ and capacities contained in Θ are bounded within intervals, i.e., $\Theta_{\text{low}} \leq \Theta \leq \Theta_{\text{up}}$, where Θ_{low}, Θ_{up} are vectors containing the lower and upper bounds on the entries of Θ, respectively, then we can determine a robust solution of the SO-DTA problem that is guaranteed to perform for all the values comprised in the uncertainty box, including the worst-case scenario. We seek in this case a solution to the following *interval* robust LP problem:

$$\min c^\top X \quad \text{subject to:} \tag{34}$$

$$AX \leq B + E\Theta, \quad \forall \Theta : \ \Theta_{\text{low}} \leq \Theta \leq \Theta_{\text{up}}. \tag{35}$$

The interval LP above can be readily cast and solved as a standard LP, with no additional computational effort with respect to a deterministic one with no uncertainties, as detailed in the next proposition.

Proposition 3.1. *A solution for the robust SO-DTA in* (34)-*(35) can be obtained by solving the standard LP*

$$\min c^\top X \quad \textit{subject to:} \tag{36}$$

$$AX \le B + E_+\Theta_{\text{low}} - E_-\Theta_{\text{up}}, \tag{37}$$

where

$$[E_+]_{ij} = \begin{cases} [E]_{ij} & \textit{if } [E]_{ij} > 0 \\ 0 & \textit{otherwise} \end{cases},$$

$$[E_-]_{ij} = \begin{cases} -[E]_{ij} & \textit{if } [E]_{ij} < 0 \\ 0 & \textit{otherwise} \end{cases}.$$

Proof. The proof is straightforward: considering the i-th row of constraints (35) and denoting the i-th row of matrix A with $A_{i,:}$, we have

$$\begin{aligned}
A_{i,:}X - B_i &\le \sum_j E_{ij}\Theta_j, \quad \forall\, \Theta_{\text{low},j} \le \Theta_j \le \Theta_{\text{up},j} \\
&\Updownarrow \\
A_{i,:}X - B_i &\le \min_{\Theta_{\text{low}} \le \Theta \le \Theta_{\text{up}}} \sum_j E_{ij}\Theta_j \\
&= \sum_j \min_{\Theta_{\text{low},j} \le \Theta_j \le \Theta_{\text{up},j}} E_{ij}\Theta_j \\
&= \sum_j \begin{cases} E_{ij}\Theta_{\text{low},j} & \text{if } E_{ij} > 0 \\ E_{ij}\Theta_{\text{up},j} & \text{if } E_{ij} \le 0 \end{cases}
\end{aligned}$$

from which the statement easily follows. □

3.3. Randomized scenario-based optimization

As discussed in Section 3.1, the robust worst-case approach to the SO-DTA is in general computationally unaffordable (except in some very special cases, such as the one discussed in Section 3.2). Besides, such a worst-case approach can be very conservative, since constraints are enforced for all possible realizations of the uncertainty, even the rarest or very unlikely ones. Further, since such a solution is basically grounded on the worst set of constraints, it will most likely yield to an underuse of the potential of the road network. Contrary, the probabilistic approach based on RCP (Calafiore, 2010) aims at obtaining an optimal solution that is valid for most, yet not all the cases. Moreover, the RCP approach provides a bound on the probability of failure of the optimal solution, in terms of a *violation probability,* as it is detailed in the following paragraphs. The RCP approach also allows to manage a wider category of constraints. In fact, here we do not limit the uncertainty vector Θ to be limited within two bounds, but assume Θ to be a realization of a stochastic variable with some probability distribution, which may be even unknown to the problem solver, as long as samples from this distribution are available.

The main concept underlying RCP is to seek for solutions of a given program that result feasible with a given probability on new, unseen scenarios. More formally, referring to problem (30)-(31), the first step is to impose the constraint (31) to hold at least with an assigned probability $1-\epsilon$, where $\epsilon \in (0,1)$ is a given small level. The problem formulation resulting from such an approach is a so-called chance-constrained optimization problem of the form (see, e.g., (Prekopa, 1970))

$$\min c^\top X \quad \text{subject to:} \tag{38}$$
$$\text{Prob}\{AX \leq B + E\Theta\} \geq 1-\epsilon, \tag{39}$$

with the understanding that the probability constraint is here imposed *jointly* on all rows of the constraints $AX \leq B + E\Theta$. This joint chance-constrained LP problem, however, is very hard to solve exactly, see, e.g., (Luedtke, Ahmed, & Nemhauser, 2010).

A solution strategy that permits to alleviate the conservativeness of the deterministic worst-case approach, while also avoiding the complexity of the joint chance-constrained LP formulation is the so called Random Convex Programming with Violated constraints (RCPV) methodology, described in Section 4 of (Calafiore, 2010). According to the RCPV method, one considers S samples $\Theta^{(s)}$, $s = 1, \ldots, S$, of Θ, generated according to its probability distribution, and then solves a standard LP of the form

$$\min c^\top X \quad \text{subject to:} \tag{40}$$
$$AX \leq B + E\Theta^{(s)}, \ s \in \mathcal{S}, \tag{41}$$

where $\mathcal{S} = \{1, \ldots, S\} \backslash \mathcal{R}$, being $\mathcal{R}$ a suitably selected subset of $\{1, \ldots, S\}$ of cardinality R. The idea is to impose the constraints $AX \leq B + E\Theta^{(s)}$ for all but R of the sampled Θs. The R constraints that are neglected should be suitably selected among the ones in $\{1, \ldots, S\}$ in order to reduce the objective function value.

Consider a generic vector $\bar{\Theta}$, which may be used to form a constraint for the optimization problem. Once the solution $X^*_{S,R}$ of problem (40)-(41) is computed, $\bar{\Theta}$ may keep the problem feasible, or make it unfeasible. Hence, we define the *constraint violation probability* for the optimal solution as

$$V^* = \text{Prob}\{\bar{\Theta} : AX^*_{S,R} \not\leq B + E\bar{\Theta}\}.$$

A fundamental theoretical result establishes a relationship between the number of randomly generated samples S and the violation probability V^*, as detailed in the following proposition.

Proposition 3.2. *Consider problem* (40)-(41)*, and two scalars* $\beta \in (0,1)$ *and* $\epsilon \in (0,1)$*. If the number* S *of random constraints considered in the problem is such that*

$$S \geq \frac{2}{\epsilon} \ln \beta^{-1} + \frac{4}{\epsilon}(R + \zeta - 1), \tag{42}$$

where ζ *is equal to the number of decision variables in* X*, then it is guaranteed that*

$$\text{Prob}\{V^* \leq \epsilon\} > 1 - \beta. \tag{43}$$

Proof. *The proposition is proved in Corollary 5.1 of (Calafiore, 2010).* □

Remark 1. Since β^{-1} appears under a logarithm in (42), it is customary in RCPV applications to choose a very small level, say $\beta = 10^{-9}$, so that the event $\{V^* \leq \epsilon\}$ is almost certain to all practical purposes. RCPV theory then guarantees that by solving problem (40) one finds an optimal solution $X^*_{S,R}$ that almost certainly satisfies the chance constraint $\text{Prob}\{AX \leq B + E\Theta\} \geq 1 - \epsilon$, and provides a bound on the number S of constraints to be generated in order to guarantee such probability.

3.4. Constraint filtering and optimal removal

In this section we describe the specific approach used for dealing with the constraints in (41), and for solving problem (40). Two issues are discussed next. The first issue is the application of the RCVP method, where we show that a high number of constraints in (41) is redundant, and can be eliminated before starting the actual optimization phase, thus greatly reducing the computational burden. The second issue is related to the way in which the set of R samples to be a posteriori discarded is selected. Here, this is done optimally, by suitably recasting the problem with the addition of a vector of boolean decision variables and a filtering procedure in order to reduce the number of candidate samples.

3.4.1. RCPV application

Let S be the number of generated samples, according to Eq. (42). Let $\Theta^{(s)}$ be the vector representing the s-th sample of Θ. The robust problem is

$$\min \quad c^\top X \quad \text{subject to:} \tag{44}$$
$$AX \leq B + E\Theta^{(s)}, \qquad \forall s = \{1, \ldots, S\} \setminus \mathcal{R}. \tag{45}$$

Consider first $R = 0$. Problem (44) is easily solvable, being a standard LP. However, if S is large (e.g., in the order of the millions), the number of constraints in the LP is $S \cdot M$, where M is the number of rows in (27), hence it may be hard even to store in memory the problem description. However, the following result holds.

Proposition 3.3. *A solution for the probabilistically robust SO-DTA in (44), with $R = 0$, can be obtained by solving the following LP*

$$\min \quad c^\top X \quad \textit{subject to:} \tag{46}$$
$$AX \leq B + \gamma_{\min} \tag{47}$$

where $\gamma_{\min}$ is the vector whose components $m = 1, \ldots, M$ are

$$\gamma_{\min,m} = \min_{s=1,\ldots,S} E\Theta_m{}^{(s)}. \tag{48}$$

Proof. In (44) each constraint is repeated S times with a different constant term only. Hence, only the constraint having the minimal constant term has to be considered, while all the others are redundant. From this observation, the statement easily follows. □

3.4.2. Optimal selection of the removed samples

Let now $R > 0$ be the number of samples to be a posteriori removed from the problem formulation in order to improve the solution quality. We derive here a filtering procedure to reduce the number of removal candidates, and a mixed-integer linear programming (MILP) model in order to optimally choose, among the remaining samples, the best R to be removed.

At first, note that not all the M constraints are stochastic. In fact, some of them, depending on the problem structure, may be deterministic and hence not modified by the generated samples values. Let us denote by $\mathcal{M}_{\mathrm{DT}}$ and $\mathcal{M}_{\mathrm{ST}}$ the sets of constraints with a deterministic or stochastic constant term, respectively, and define the size of those sets by M_{DT} and M_{ST} respectively ($M = M_{\mathrm{DT}}+M_{\mathrm{ST}}$). Let us define the constant term of a constraint m resulting from a sample s as

$$\gamma_{m,s} = \sum_k E_{mk}\Theta_k^{(s)}. \tag{49}$$

While a constraint belonging to $\mathcal{M}_{\mathrm{ST}}$ has different $\gamma_{m,s}$ for each sample s, a constraint $m \in \mathcal{M}_{\mathrm{DT}}$ is independent on the sample, and can be defined as:

$$\gamma_m = \gamma_{m,s} \quad \forall s \in 1,\dots,S. \tag{50}$$

Consider now the m-th constraint $m \in \mathcal{M}_{\mathrm{ST}}$ (repeated S times, one for each sample)

$$A_{m,:}X - B_m \quad \leq \quad \gamma_{m,s} \quad \forall s = 1,\dots,S. \tag{51}$$

Due to the specific constraint structure, not all the S samples are suitable candidates for removal. In fact, only removing one of the R samples with the minimal values of $\gamma_{m,s}$ among the S removal candidates can result in a constraint relaxation and, consequently, in a possible improvement in the objective function. Hence, it is possible to derive a samples filtering procedure, only keeping the only possible candidates for elimination.

Let $\gamma_{mp}^{\mathrm{val}}$ be the value of the p-th (in non decreasing order) right-side constant term of constraint m among all the samples $s = 1,\dots,S$, and $\gamma_{mp}^{\mathrm{ind}}$ the sample index that generated that term. For both matrices we consider all the constraints ($m = 1,\dots,M$) and the first $R+1$ positions only ($p = 1,\dots,R+1$). Note that the constraints referring to position $R+1$ is not a candidate for removal, but we need to store its value, as it will be clarified in the following proposition. Hence, the possible candidates for elimination are $\mathcal{S}^* = \cup_{m:1,\dots,M,p:1,\dots R}(\gamma_{mp}^{\mathrm{ind}})$, which are at most $M \cdot R$, in the general case where $M = M_{\mathrm{ST}}$. The following proposition introduces a MILP model able to optimally choose the samples to be removed in order to improve as much as possible the objective function.

Proposition 3.4. *Let Z be a vector of additional binary decision variables of size $|S^*|$ ($Z_s \in \{0,1\}$, $\forall s \in S^*$) with the following meaning:*

$$Z_s = \begin{cases} 1 & \textit{if sample } s \textit{ is removed from the sample set,} \\ 0 & \textit{otherwise.} \end{cases} \tag{52}$$

The optimal solution for the robust SO-DTA with $R > 0$ in (44)-(45) can be obtained *by solving the following LP:*

$$\min \qquad c^\top X \quad \textit{subject to:} \tag{53}$$
$$A_{m,:}X - B_m \le \gamma^{\text{val}}_{mp} + (\gamma^{\text{val}}_{m,R+1} - \gamma^{\text{val}}_{mp}) Z_{\gamma^{\text{ind}}_{mp}}, \quad \forall p \in 1,\dots,R, \forall m \in \mathcal{M}_{\text{ST}} \tag{54}$$
$$A_{m,:}X - B_m \le \gamma_m \quad \forall m \in \mathcal{M}_{\text{DT}} \tag{55}$$
$$\textstyle\sum_{s\in S^*} Z_s = R. \tag{56}$$

Proof. The objective function (53) is the same of the original problem. Each stochastic constraint (54) is repeated R times, one for each candidate sample: if a sample s is not selected for removal ($Z_s = 0$), the constraint results in

$$A_{m,:}X - B_m \le \gamma^{\text{val}}_{mp} \quad \forall p \in 1,\dots,R \tag{57}$$

enforcing the sample constraints, whereas if sample s is selected for removal ($Z_s = 1$) the constant term of the constraint becomes $\gamma^{\text{val}}_{s,R+1}$ and the effect of the constraint is nullified (in case all of the candidates at positions $1,\dots,R$ are removed from a single constraint m, the constraint constant term $\gamma_{m,s}$ will be the $(R+1)$-th one). Deterministic constraints (55) are repeated only once (the constant term is the same for all samples). Constraint (56) sets R as the number of constraints to be removed. This formulation is then able to optimally select the R samples to be removed from the samples set in order to achieve the best improvement in the objective function. □

The overall model (53) results in $M_{\text{DT}}+M_{\text{ST}}\cdot R$ constraints, and a maximum of $M_{\text{ST}}\cdot R$ binary variables, while the original number of continuous variables is not changed. Note that while generating the samples, in order to avoid using a huge quantity of memory to store dominated and consequently useless samples, it is not necessary to store all of them, yet we can filter them keeping track of the real candidates $\mathcal{S}^*$ only. This way, we obtain a greatly improved possibility of increasing the number of generated samples. The overall generation procedure is described in Algorithm 1.

Algorithm 1 Samples generation and filtering.

```
for count = 1, ..., S do
  generate sample s
  for m = 1, ..., M do
    if m ∉ M_ST then
      continue
    end if
    compute γ_{m,s}
    if γ_{m,s} > γ^val_{m,R+1} then
      continue
    end if
    order insert γ_{m,s} in γ^val_m
    update γ^ind_m accordingly
  end for
end for
```

Proposition 3.5. *The computational complexity of Algorithm 1 is* $S \cdot o(C \cdot T \cdot \log(R))$.

Proof. The insert operation in an already ordered vector is realized with a $o(\log(R+1))$

complexity algorithm, which is repeated at maximum $M_{\mathrm{ST}} \cdot S$ times. Finally, $M_{\mathrm{ST}} \leq M = 4 \cdot C \cdot T + 1$, from which the proposition follows. □

Hence, the worst case expected CPU time needed for the samples generation is proportional to the number of generated samples, the number of cells and the number of considered time periods, while R can be increased with a lesser impact (logarithmic) on the additional time.

3.4.3. Heuristic selection of the removed samples

On large instances, or with a large R setting, solving the integer programming model of proposition 3.4 to optimality could require a long computational time. It is always possible to set a time limit for the MILP solver: when the optimal solution is not yet found when the limit is met, the solver will return the best solution found up to that moment, and its optimality gap (ratio between the objective function value and the worst-case lower bound for all the open nodes of the search tree). Note that the Branch and Bound algorithm embedded in MILP solvers often spends a large amount of time verifying the optimality of an already found solution, see (Ozaltin, Hunsaker, & Schaefer, 2011). Hence, a good solution, if not the optimal one, may already be available when the time limit is met. In case not enough time is available for obtaining a good solution, we introduce a simple heuristic algorithm able to find a good solution in a limited amount of time. The main steps are:

- Relax the integrality of Z variables, substituting the definition $Z_s \in \{0,1\} \quad \forall s \in S$ with $0 <= Z_s <= 1 \quad \forall s \in S$. The resulting model is a continuous variables LP model (computationally easily solvable).
- The optimal solution of the relaxed problem is unlikely integer, but the information about the values of the continuous variables Z_s can be used to derive an integer solution: if a variable has a value nearer to one, it is more likely to be one in a good solution than a variable with a lower value.
- We then select the K_{fix} variables Z_s with the highest value, and we add a constraint on those variables setting them to 1.
- The model is then solved again, and the fixing procedure is repeated, until we have exactly Z variables with value 1.
- The number K_{fix} of variables to be fixed at each iteration can derived by computational testings.

The detailed pseudocode of the overall procedure is provided in Algorithm 2.

Algorithm 2 Heuristic Algorithm

- consider model (53).
- substitute the integrality definition $Z_s \in \{0,1\}$ with $0 \leq Z_s \leq 1$, $\forall s \in S$.
- solve the obtained (continuous variables) model to optimality.

while $R_1 = \sum_{s \in S : Z_s = 1.0} Z_s \neq R$ **do**

  - order Z_s in non-decreasing order. Let S_z be the set of indexes s referring to the first $\min(K_{\mathrm{fix}}, R - R_1)$ values such that $Z_s \neq 1.0$.
  - add the constraints $Z_s = 1.0$, $\forall s \in S_z$.
  - solve the updated (continuous variables) model to optimality.

end while

4. Computational results

The presented approach has been tested on different networks. The first is an ad-hoc synthetic network, which can be dynamically generated in different sizes. The second is a network taken from the literature ((Sun et al., 2014), (Waller & Ziliaskopoulos, 2006)), derived from a real geographical area. Then, in order to test the approach applicability on larger, realistically sizes, a scalability test has been performed using both synthetic networks and a larger 620 cells one. In all the experiments, we obtain a set of solutions using different values for ϵ and R, and we compare them with the solutions obtained using the deterministic model (expected values of the uncertain data), and the one obtained through the worst-case optimization approach presented in Section 3.1. Note that the worst-case approach can be applied only if all the uncertainties are assumed to be bounded in intervals; the introduced probabilistic approach is instead applicable for general uncertainty distributions. All the generated solutions are then tested a posteriori for feasibility on a new set of 5000 samples, hence measuring the capacity of the solution to remain feasible on new scenarios (i.e., the solution robustness). Clearly, the results of the deterministic method in terms of objective function only will never be worse than the ones obtained with any robust method (if things goes exactly as predicted, the solution is then optimal), but solutions will have a very high violation probability for new samples. In terms of traffic, any violated constraint would result in traffic left behind in the previous nodes, generating congestions. The set of obtained solutions has been evaluated in a bi-objective setting, considering the original objective function, and the associated percentage of unfeasible new scenarios. Recall that a solution is *Pareto efficient* if none of the objective functions can be improved in value without degrading the other one (Ehrgott, 2005). Any solution worse than an efficient one for all the considered objective functions is a *dominated solution.* Then, it is possible to check all the obtained solutions with the aim of identifying the Pareto efficient ones. Note that, without additional subjective preference information, all Pareto optimal solutions are considered equally good.

4.1. A synthetic network

The synthetic network in this example is composed by a graph of cells organized in layers. A parameter K_{gen} controls the network dimension and the network is generated through the following layers:

- Layer 1: K_{gen} source cells. Each source cell is connected to a single cell in the second level.
- Layer 2: K_{gen} diverging cells. Each cell is indirectly connected to a single cell in level 4, through one and only one cell of layer 3.
- Layer 3: K_{gen}^2 ordinary cells.
- Layer 4: K_{gen} merging cells. Each cell is connected to a single sink node at layer 5.
- Layer 5: K_{gen} sink cells.

As previously stated, the model can be applied on multiple sink cells networks, as far as it is immaterial which destination node the incoming traffic is routed to. This is also equivalent to the model obtained adding a single final node, connected from all the sink nodes, with infinite capacity incoming links, as in the synthetic network with $K_{\text{gen}} = 2$ exemplified in Figure 2 where the number of generated cells is $C = K_{\text{gen}}^2 + 4K_{\text{gen}} = 12$.

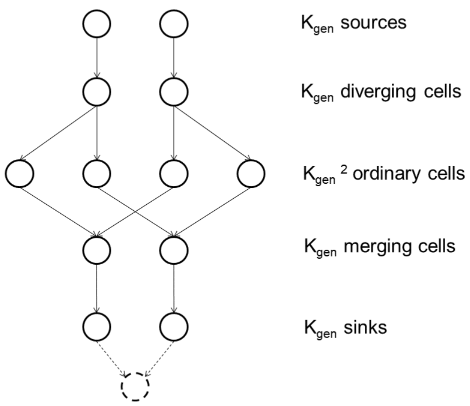

Figure 2. Cell representation of a synthetic network with $K_{\text{gen}} = 2$.

Other network input data are:

- The capacities $Q_i(t)$ are set to ∞ for source and sink cells, and to 10 vehicles per time interval for all other cells.
- The free-flow to backward propagation speed ratio $\delta_i(t)$ is set to 1, without loss of generality.
- The considered time horizon T is 30 time units.
- The demand $d_i(t)$ at the source cells is initially described by a uniform distribution $U(50, 200)$ applied to each of the first 5 time intervals. Then, it is set to 0.
- The cell holding capacities $N_i(t)$ are set to ∞ for source and sink cells, whereas a uniform distribution $U(15, 25)$ is used for the cells at layer 3 of the network, and to 20 for all other cells.

The software for generating the samples has been implemented in C++ language and the tool FICO XPress-MP version 7.9 has been used on a 8GB memory Intel i5-6200U 2.80GHz PC system in order to solve the MILP model described in Proposition 3.4.

Two networks have been generated, using $K_{\text{gen}} = 3$ (21 cells) and $K_{\text{gen}} = 4$ (32 cells), respectively. The results are summarized in Tables 2 and 3. The result tables present the required violation probability ϵ (while β is set to 10^{-6} in all experiments), the number of samples that have to be generated in order to respect that probability (in accordance to eq. (42)), the number R of samples to be removed in the improvement procedure, the number of samples remaining from the filtering phase (also, the number of binary variables of the samples removal model), the CPU time T_{gen} required for samples generation, the optimal objective function $O.F.$ obtained by the solver, its improvement with respect to the worst case solution approach, the CPU time T_{sol} required by the solver to reach the optimal solution, and the number of samples, among the new 5000 test samples, for which the solution resulted to be unfeasible. The bold cells identify the Pareto efficient solutions. Both the efficient and dominated solutions are also represented in Figures 3 and 4, where the axes represent the optimal objective function (horizontal axis) and the number of unfeasible new samples (vertical axis). Notice that the only solution that is listed in the tables but not represented in the figures is the nominal one, since it would appear as out of scale compared to all the others.

In the first network the constraints are 2521. Among them, due to the stochasticity in demands and capacities, 451 constraints are stochastic. In the second network the constraints are 3841, and 721 of them are stochastic.

The sample generation to obtain the requested robustness level is fast and effectively filters the samples, reducing the number of binary variables of the sample removal model.

As expected, while the generation is always fast (below 10 seconds), the sample removal model becomes harder as R increases, due to a larger number of binary variables associated to the reduced samples. Anyway, the solver is always able to find the optimal solution in a few seconds.

Moreover, note that the removal procedure is particularly effective on the solution objective function. For instance, the objective function of the solution obtained with the maximum tested robustness ($\epsilon = 0.05$) when $R = 200$ is always better than the solution with a lower one (up to $\epsilon = 0.25$) without using the removal procedure ($R = 0$).

The results of the a posteriori testing of the solutions on new samples show that most of the solutions are much more robust than predicted by the theory. In particular, the theoretically guaranteed feasibility on new samples is around $1 - \epsilon$, while all the solutions show experimentally a feasibility greater than 0.98. Particular cases are represented by the nominal solution (based on the expected values of the stochastic distributions), which shows, as expected, a high level of unfeasibility, and the worst-case robust solution which is guaranteed to be always feasible on any new problem sample. This higher-than-expected robustness can be explained by the fact that the theory is conservative, providing only a guaranteed bound on the violation probability.

The objective function improvement with respect to the worst case robust solution is ranging from 17% to 28%, depending on the algorithm setup.

Finally, note that a large number among the Pareto efficient solutions have been obtained by the models with $R > 0$, and often with the largest tested values of R. This means that the a posteriori constraint removal method is particularly efficient in order to achieve better solutions than the one that are obtainable simply increasing or decreasing the number of generated samples.

4.2. 62 Cells Network

As a second example, we tested the performance of our approach on a network introduced in (Waller & Ziliaskopoulos, 2006) and (Sun et al., 2014), which is derived from a real geographical area. The network is composed by 62 cells, and is represented in Figure 5, where the node-arc representation on the left, with 22 nodes, is translated into the cell representation on the right. Note that the arcs $(57, 9)$ and $(57, 58)$ directly connect a diverging and a merging cell. As described in Section 2.2, in order to apply the variable reduction, two dummy cells $D_{57,9}$ and $D_{57,58}$ have been added to the network.

Similarly to the papers where this network has been already used for testing, only the demand is supposed to be stochastic in this case study. The demand distributions at the source cells are $U(400, 800)$ for node 1 (cell 54), and $U(200, 250)$ for nodes 8 and 13 (cells 55 and 56). This demand is generated at time $t = 0, \dots, 5$ only. The

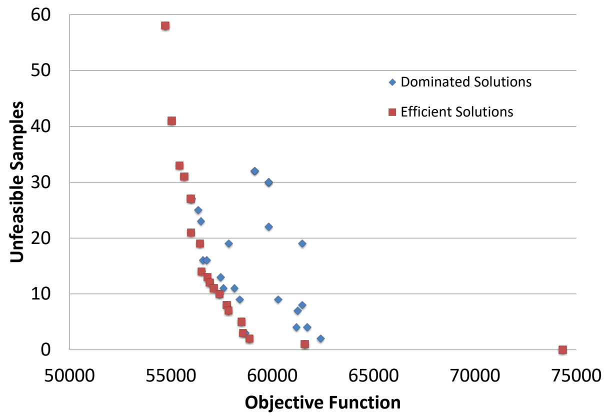

Figure 3. Efficient and dominated solutions for the 21 cells network of Section 4.1.

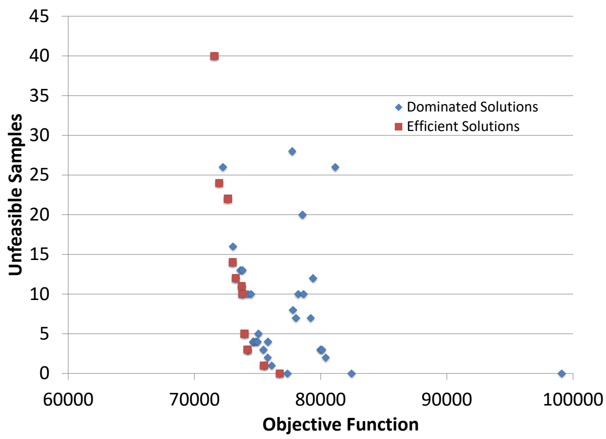

Figure 4. Efficient and dominated solutions for the 32 cells network of Section 4.1.

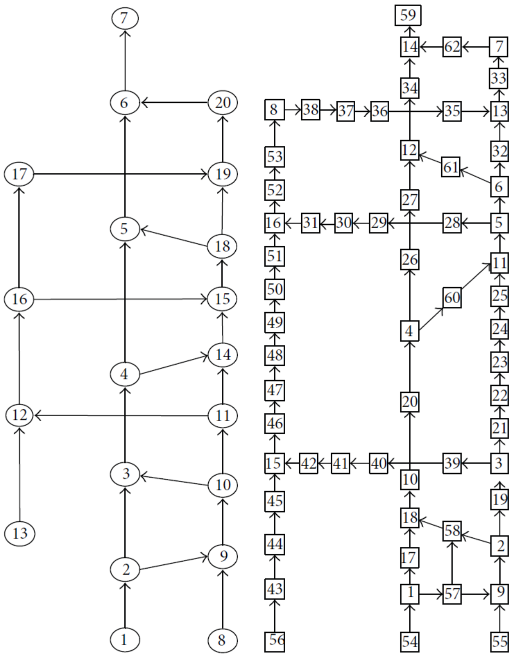

Figure 5. Link and cell representation for the realistic network described in Section 4.2.

time horizon is set as $T = 105$. The network parameters configuration is summarized in table 4. Results are presented in Table 5, while Figure 6 shows the selection of Pareto efficient solutions.

Note that, compared to the previous synthetic examples, the requested samples to be generated for achieving the same robustness is highly increased. This mainly happens because of equation (42) due to the increase in variables of the original problem (more cells, and a larger time horizon). Hence, the computational time needed for generating the samples increased as well. Anyway, the filtering procedure is very effective in terms of samples reduction, and the time requested by the solver to reach the optimal solution is of the order of a few seconds. The reason for this very good filtering algorithm result is that the demand here is generated at time $t = 0, \ldots, 5$, and on three cells only, with a less connected network graph than in the previous examples. Hence, most of the problem constraints do not have a stochastic term. Moreover in most of the stochastic constraints the demand appears with a similar constraint structure, which means that a few samples (basically, the ones with lower demand) dominate all the other ones. The quality of the solutions, again, is improved using higher R values. For the same reason, the improvement from the worst case solution is still substantial but smaller than in the synthetic problem case, ranging from 5% to 10%.

The lower stochasticity of the problem can be seen in the a posteriori solutions feasibility. Again, the real robustness of the solution is greatly higher than the theoretically guaranteed one. And again, most of the efficient solutions are found using higher R values showing the great potential of the constraint elimination method.

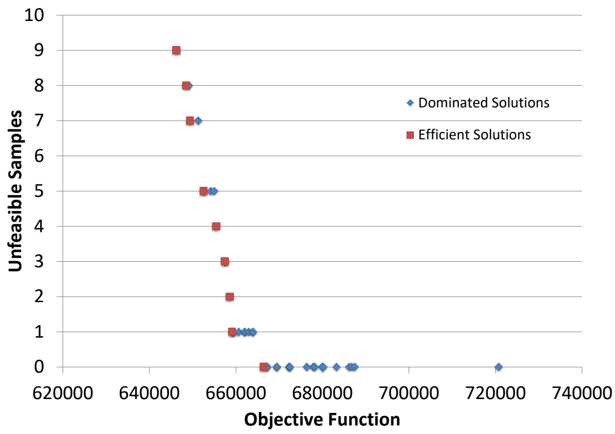

Figure 6. Efficient and dominated solutions for the 62 cells realistic network described in Section 4.2.

4.3. Scalability Test on Synthetic Networks

In order to test the scalability of the proposed approach, additional testings have been performed for networks defined as in Section 4.1, for K_{gen} ranging from 5 (45 nodes) to 16 (320 nodes), using four different algorithm configurations, with $\epsilon = 0.1$ or 0.2, and $R = 50$ or 100. Results are presented in Figure 7, where the time requested by the samples generation algorithm T_{gen}, by the optimal constraints removal algorithm T_{sol} and the total one $T_{\text{tot}} = T_{\text{gen}} + T_{\text{sol}}$ on all the generated instances are presented, as a function of K_{gen}. As expected, the generation algorithm is more influenced by ϵ than by R, because of the increasing number of samples to be generated. On the contrary, the optimal constraints removal is larger when R is larger, because of the cardinality of the variables array Z generated in the LP model. T_{sol} seems less predictable than the time needed by the generation algorithm but remains computationally tractable. For the presented experiments, a maximum time of 3000 seconds has been set, but it has never been reached. Also, the graph representing the test network is more connected (ratio between the cardinality of arcs set and node set in the graph) than the networks used for transportation problems in literature (see for instance (Transportation Networks for Research Core Team, 2019)), creating models with more random constraints than in a real network. Hence, the computational time presented here can be considered as a sort of upper bound on the time needed on a real network with the same size.

4.4. A large 620 cells network

In order to stress the algorithm performances on a large, realistically sized network, the previously introduced 62 cells network have been used in order to generate a larger network. The original network has been duplicated 10 times and the different copies have been connected as illustrated in Figure 8: on the left, the connections from/to a single subnetwork are presented, while on the right the overall network is illustrated. The result is a network composed by 620 cells (7 sources) and 692 links. The network configuration (summarized in Table 6) has stochasticity in both the demand and the cell holding capacity. In particular, note that $N_i(t)$ has been defined through a discrete distribution on the cells belonging to the central freeway for all the sub-networks: its value is set to 20 with probability 0.5, or 30 otherwise, modeling, for instance, the

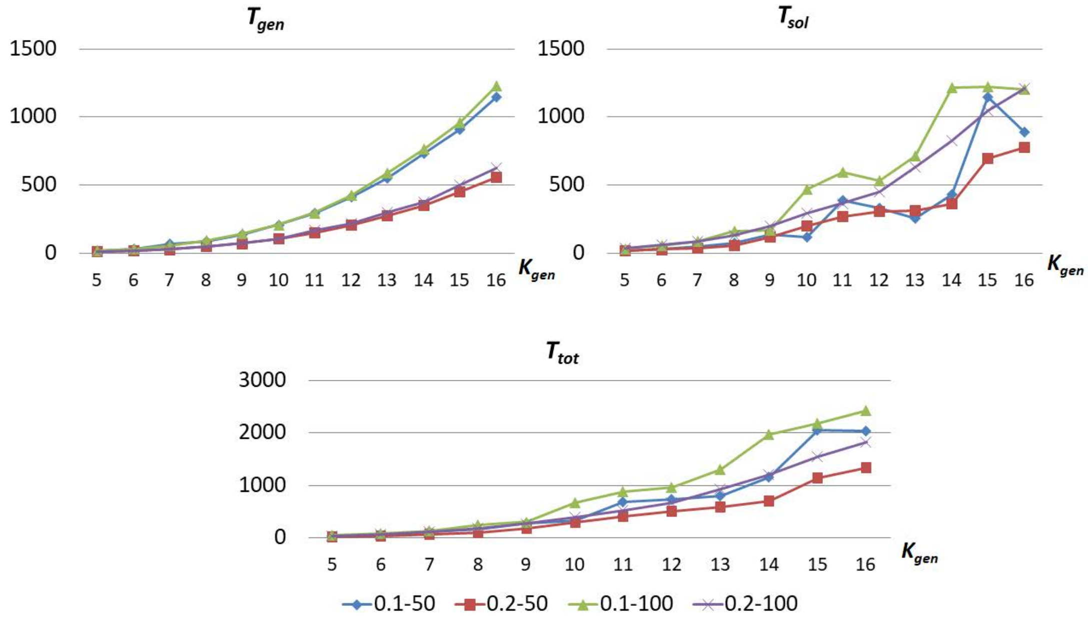

Figure 7. Scalability testing results on synthetic networks.

opening/closing of an additional lane. One of the strength of our approach is that it can be used independently of the distribution of the uncertain parameters. Note that the resulting linear programming model is quite large, already for $R = 0$, counting 74421 continuous variables and 297601 constraints. The constraints with a stochastic constant term are 14881. For this reason, and considering that, for all previous testings, the method seems to be more conservative than theoretically expected, higher values of the parameter ϵ have been considered, compared to the previous tests: 0.25 and 0.5, with R ranging from 0 to 20. A time limit of $T_{lim} = 3000$ s has been set for the solver. Results are presented in table 7 and figure 9. Note that the time required to generate the samples increased proportionally with the number of samples, while the good behavior of the filtering procedure, in terms of samples reduction, is confirmed. The solver time limit has been reached in only one instance, and the solution that the solver attained in that case is actually the optimal one (it would have required about 600 additional seconds to prove it). The quality of the solutions, again, is improved using higher R values. The improvement from the worst case solution is ranging from 4% to 9%. It is also confirmed that the real robustness of the solution is consistently higher than the theoretically guaranteed one. In order to further compare the solutions' quality, and relate it to transportation-related performance indices, we carry out an analysis on the average number of uncongested cells (i.e., cells that are in the free-flowing speed condition). Figure 10 summarizes the results of such an analysis. On the horizontal axis, the solutions of Table 7 have been sorted according to their improvement, in terms of objective function, with respect to the robust (worst-case) solution. On the vertical axis, for each solution, we plot the average percentage of time slots in which cells are in the uncongested state. This quantity is averaged on all cells, on freeway cells only, and on non-freeway cells only, respectively. We observe that solutions with an improved value in the objective function generally correspond to solutions that favor the inception of uncongested states. Some remarkable exception is constituted by freeway cells, which are sometimes more congested in correspondence

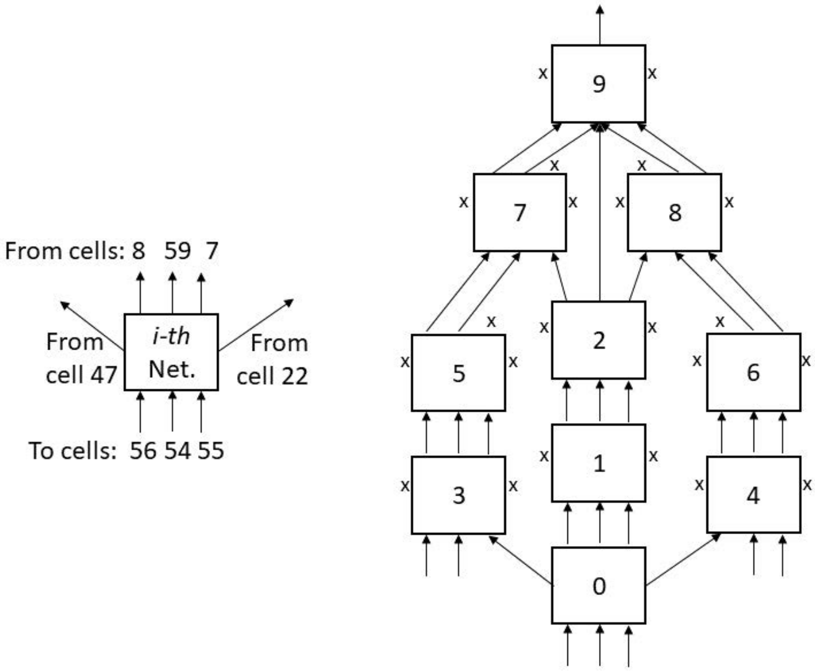

Figure 8. The 620 cells network

of solutions with a better value of the objective function. However, in these cases, the congestion is much more mitigated over secondary, non-freeway roads.

4.5. Heuristic algorithm computational test

The heuristic algorithm presented in section 3.4.3 has been tested on the samples removal problems of the 620 cells network. Results are summarized in table 8 where objective functions and required CPU time of the exact (T_{sol}) and heuristic (T_{heur}) approaches are compared. Preliminary testings indicated that the best algorithm configuration was with $K_{\text{fix}} = 20$. Hence, the algorithm performs only one iteration. The obtained solutions are already tight enough to the optimal values, and using more iterations would results in a very small improvement. It is also interesting that about half of the time T_{heur} is actually used by the solver only to load the problem structure in memory, and would be probably reduced using a solver with a more efficient memory management.

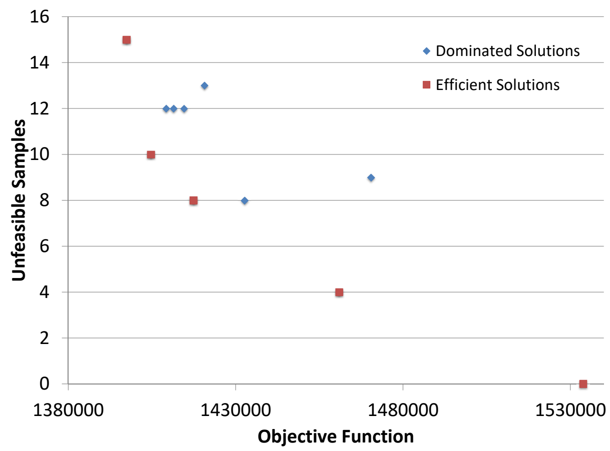

Figure 9. Efficient and dominated solutions for the 620 cells network

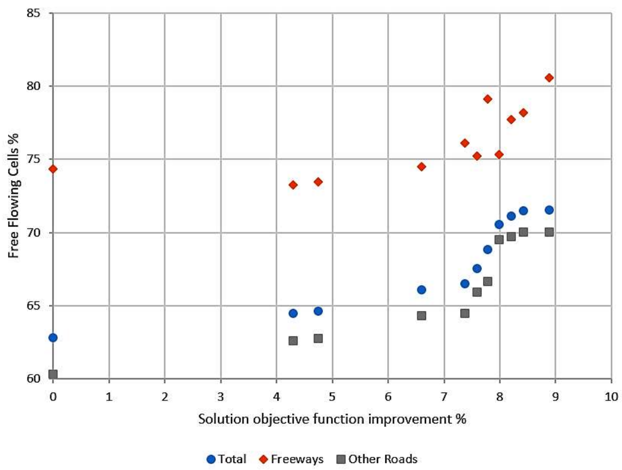

Figure 10. Congestion analysis for the 620 cells network

Table 2. Results with $K_{\text{gen}} = 3$ (21 cells) for the synthetic network of Section 4.1

ϵ	R	Samples	Reduced Samples	T_{gen} (sec)	O.F.	Impr.	T_{sol} (sec)	Unf. New Samples
Deterministic	-	-	-	-	**43956.11**	-	0	**4830**
Worst Case	-	-	-	-	74331.11	-	0	0
0.05	0	101433	299	4	**61608.17**	17%	0	**1**
0.05	20	103033	6091	5	**58870.88**	21%	2	**2**
0.05	40	104633	11455	6	58674.10	21%	4	3
0.05	60	106233	16720	6	**58482.19**	21%	7	**5**
0.05	80	107833	21568	6	**57758.88**	22%	9	**8**
0.05	100	109433	26338	6	57765.85	22%	10	8
0.05	200	117433	46488	7	**56921.70**	23%	20	**12**
0.1	0	50717	298	2	61731.29	17%	0	4
0.1	20	51517	5879	2	58403.28	21%	2	9
0.1	40	52317	10893	3	**57843.91**	22%	5	**7**
0.1	60	53117	15357	3	**57455.71**	23%	5	**13**
0.1	80	53917	19352	3	57394.31	23%	7	10
0.1	100	54717	23053	3	**56803.06**	24%	9	**13**
0.1	200	58717	37438	3	**55992.04**	25%	18	**21**
0.15	0	33811	297	1	62384.17	16%	0	2
0.15	20	34345	5679	2	58556.11	21%	2	3
0.15	40	34878	10295	2	57597.44	23%	4	11
0.15	60	35411	14132	2	**57119.14**	23%	5	**11**
0.15	80	35945	17563	2	56587.95	24%	7	16
0.15	100	36478	20396	2	**56518.38**	24%	9	**14**
0.15	200	39145	30594	2	**55418.25**	25%	17	**33**
0.2	0	25359	295	1	61197.34	18%	0	4
0.2	20	25759	5499	1	58131.67	22%	2	11
0.2	40	26159	9718	1	57404.21	23%	5	10
0.2	60	26559	13051	1	**56432.46**	24%	6	**19**
0.2	80	26959	15782	1	56350.58	24%	7	25
0.2	100	27359	18146	1	**55982.73**	25%	9	**27**
0.2	200	29359	25376	2	**55040.46**	26%	16	**41**
0.25	0	20287	291	0	61250.03	17%	0	7
0.25	20	20607	5395	1	57856.59	22%	1	19
0.25	40	20927	9217	1	56768.13	24%	5	16
0.25	60	21247	12195	1	56482.08	24%	6	23
0.25	80	21567	14446	1	56025.20	25%	6	27
0.25	100	21887	16295	1	**55659.21**	25%	9	**31**
0.25	200	23487	21596	1	**54725.90**	26%	18	**58**
0.3	0	16906	296	0	60291.18	19%	0	9
0.4	0	12680	291	0	61485.82	17%	0	8
0.5	0	10144	291	0	61485.82	17%	0	19
0.6	0	8453	293	0	59823.58	20%	0	22
0.7	0	7246	294	0	59823.58	20%	0	30
0.8	0	6340	294	0	59823.58	20%	0	30
0.9	0	5636	293	0	59133.01	20%	0	32

Table 3. Results with $K_{\text{gen}} = 4$ (32 cells) for the synthetic network of Section 4.1

ϵ	R	Samples	Reduced Samples	T_{gen} (sec)	O.F.	Impr.	T_{sol} (sec)	Unf. New Samples
Deterministic	-	-	-	0	**58608.15**	-	0	**4783**
Worst Case	-	-	-	0	99108.15	-	0	0
0.05	0	154233	517	11	82458.80	17%	0	0
0.05	20	155833	10455	12	77371.53	22%	2	0
0.05	40	157433	19804	13	76121.99	23%	8	1
0.05	60	159033	28609	14	75598.89	24%	12	1
0.05	80	160633	36832	14	**75511.85**	24%	15	**1**
0.05	100	162233	44579	15	75077.95	24%	20	5
0.05	200	170233	77507	17	74248.03	25%	41	3
0.1	0	77117	518	6	80416.49	19%	0	2
0.1	20	77917	10137	6	**76771.43**	23%	3	**0**
0.1	40	78717	18562	6	75815.37	24%	7	2
0.1	60	79517	25908	7	75021.20	24%	10	4
0.1	80	80317	32560	7	74683.69	25%	15	4
0.1	100	81117	38398	7	**74212.57**	25%	20	**3**
0.1	200	85117	59879	9	**73285.28**	26%	46	**12**
0.15	0	51411	520	3	80122.73	19%	0	3
0.15	20	51945	9780	4	75831.46	23%	4	4
0.15	40	52478	17305	4	74879.50	24%	7	4
0.15	60	53011	23701	5	**73981.08**	25%	10	**5**
0.15	80	53545	29116	4	**73808.19**	26%	14	**10**
0.15	100	54078	33360	5	73763.64	26%	18	11
0.15	200	56745	47576	6	**72651.43**	27%	40	**22**
0.2	0	38559	516	2	78048.53	21%	0	7
0.2	20	38959	9472	3	75477.78	24%	3	3
0.2	40	39359	16352	3	74492.61	25%	8	10
0.2	60	39759	21603	3	**73761.44**	26%	12	**11**
0.2	80	40159	25888	3	73650.75	26%	15	13
0.2	100	40559	29313	3	**73029.91**	26%	21	**14**
0.2	200	42559	38728	4	**71971.30**	27%	41	**24**
0.25	0	30847	514	2	79993.84	19%	0	3
0.25	20	31167	9193	2	74652.74	25%	3	4
0.25	40	31487	15432	2	74241.88	25%	8	10
0.25	60	31807	19989	3	73833.30	26%	11	13
0.25	80	32127	23429	3	73065.35	26%	13	16
0.25	100	32447	25940	3	72266.53	27%	19	26
0.25	200	34047	32424	3	**71590.93**	28%	42	**40**
0.3	0	25706	511	1	77813.32	21%	0	8
0.4	0	19280	512	1	79221.24	20%	0	7
0.5	0	15424	511	1	78230.44	21%	0	10
0.6	0	12853	511	1	78653.80	21%	0	10
0.7	0	11017	506	0	79413.89	20%	0	12
0.8	0	9640	502	0	78568.00	21%	0	20
0.9	0	8569	501	0	77754.12	21%	0	28

Table 4. Data for 62 cells network. Freeway cells are all the cells on the direct path between source cell 54 and sink cell 59

Data	Source Cells	Sink Cells	Freeway Cells	Other Cells
$N_i(t)$	∞	∞	20	10
$Q_i(t)$	∞	∞	12	3
$\delta_i(t)$	1	1	1	1
$d_i(t)$	U(400,800) for freeway source (t=1,...,5) U(200,250) for other sources (t=1,...,5)	0	0	0

Table 5. Results for the 62 cells realistic network described in Section 4.2.

ϵ	R	Samples	Reduced Samples	T_{gen} (sec)	O.F.	Impr.	T_{sol} (sec)	Unf. New Samples
Deterministic	-	-	-	0	**568489.00**	-	8	**2822**
Worst Case	-	-	-	0	720739.00	-	8	0
0.05	0	1042233	37	1293	683274.52	5%	8	0
0.05	25	1044233	862	1656	672292.93	7%	8	0
0.05	50	1046233	1693	1745	669436.57	7%	10	0
0.05	100	1050233	3271	1805	**666408.17**	8%	10	**0**
0.05	200	1058233	6316	1832	663996.57	8%	14	1
0.05	300	1066233	9292	1771	660637.19	8%	20	1
0.05	400	1074233	12316	1801	**659155.04**	9%	27	**1**
0.05	500	1082233	15173	1816	**658578.57**	9%	34	**2**
0.1	0	521117	35	640	687476.91	5%	8	0
0.1	25	522117	861	816	669508.77	7%	8	0
0.1	50	523117	1662	883	667330.48	7%	10	0
0.1	100	525117	3159	906	662952.36	8%	10	1
0.1	200	529117	6134	920	659232.22	9%	14	1
0.1	300	533117	8965	894	**657455.67**	9%	20	**3**
0.1	400	537117	11849	906	**655485.65**	9%	27	**4**
0.1	500	541117	14544	914	654169.43	9%	34	5
0.2	0	260559	35	319	686815.14	5%	8	0
0.2	25	261059	835	407	667119.54	7%	8	0
0.2	50	261559	1606	442	662040.13	8%	9	1
0.2	100	262559	3095	445	659554.46	8%	10	1
0.2	200	264559	5950	460	654988.31	9%	14	5
0.2	300	266559	8658	454	652614.34	9%	20	5
0.2	400	268559	11131	451	651315.32	10%	27	7
0.2	500	270559	13904	454	649173.26	10%	34	8
0.3	0	173706	35	215	686183.78	5%	8	0
0.3	25	174039	836	274	663900.23	8%	8	1
0.3	50	174373	1559	295	662045.29	8%	9	1
0.3	100	175039	3019	299	657580.70	9%	13	3
0.3	200	176373	5732	308	**652572.36**	9%	14	**5**
0.3	300	177706	8293	377	**649438.29**	10%	20	**7**
0.3	400	179039	10894	304	**648511.64**	10%	27	**8**
0.3	500	180373	13318	310	**646251.28**	10%	34	**9**
0.4	0	130280	34	158	672384.75	7%	8	0
0.5	0	104224	35	127	680220.42	6%	8	0
0.6	0	86853	35	106	677912.11	6%	8	0
0.7	0	74446	32	90	679979.64	6%	8	0
0.8	0	65140	31	79	672541.71	7%	8	0
0.9	0	57902	38	70	678237.81	6%	8	0

Table 6. Data for the 620 cells network. Freeway cells are all the cells on the direct path between source cell 54 and sink cell 59, for all sub-networks

Data	Source Cells	Sink Cells	Freeway Cells	Other Cells
$N_i(t)$	∞	∞	20 or 30 (p=0.5)	15
$Q_i(t)$	∞	∞	20	6
$\delta_i(t)$	1	1	1	0,5
$d_i(t)$	U(400,800) for freeway sources (t=1,...,5) U(200,250) for other sources (t=1,...,5)	0	0	0

Table 7. Results for the large 620 cells network (* = unfinished)

ϵ	R	Samples	Reduced Samples	T_{gen} (sec)	O.F.	Impr.	T_{sol} (sec)	Unf. New Samples
Deterministic	-	-	-	0	**1101016**	-	170	**4812**
Worst Case	-	-	-	0	1533800	-	167	0
0.5	0	1190464	940	6304	1470479	4.3%	168	9
0.5	5	1190504	4850	6676	**1417355**	7.59%	271	**8**
0.5	10	1190544	8400	7048	1409241	8.21%	1678	12
0.5	15	1190584	11990	7312	**1404681**	8.42%	1381	**10**
0.5	20	1190624	15460	7436	**1397442**	8.89%	1985	**15**
0.25	0	2380927	930	12328	**1460958**	4.75%	189	**4**
0.25	5	2381007	4910	13472	1432622	6.60%	461	8
0.25	10	2381087	8590	14080	1420702	7.37%	802	13
0.25	15	2381167	12240	14440	1414512	7.78%	2294	12
0.25	20	2381247	16100	14928	1411436	7.98%	3000*	12

Table 8. Heuristic Algorithm Results on the 620 cells network (* = unfinished)

ϵ	R	Optimal O.F.	T_{sol} (sec)	Heuristic O.F.	Gap	T_{heur} (sec)
0.5	5	1417355	271	1417355	0.00‰	175
0.5	10	1409241	1678	1409241	0.00‰	247
0.5	15	1404681	1381	1404914	0.16‰	286
0.5	20	1397442	1985	1397542	0.07‰	289
0.25	5	1432622	461	1432622	0.00‰	168
0.25	10	1420702	802	1420702	0.00‰	203
0.25	15	1414512	2294	1415497	0.69‰	237
0.25	20	1411436	3000*	1412888	1.02‰	273

5. Conclusions

In this paper, the system-optimum dynamic traffic assignment (SO-DTA) problem in the presence of time-dependent uncertainties on both traffic demands and road link capacities has been considered. Building on an earlier formulation of the problem based on the cell transmission model, the SO-DTA problem is robustly solved in the probabilistic sense in the framework of random convex programs. A specific model able to obtain a robust solution for medium to large scale networks with a low desired violation probability in a tractable computation time has been introduced. Moreover, exploiting the specific model structure, a samples filtering method and a mixed integer linear programming model have been derived in order to improve the expected objective function, while maintaining the requested violation probability. The same methodology is applicable to any problem formulated through a linear programming model having stochasticity in the constant terms of the constraints only. The obtained solutions have been tested in order to evaluate a posteriori the probability that the addition of further samples make the computed solution unfeasible. The results show that the requested violation probability is always met, and that the samples removal method is particularly effective in finding efficient solutions for both total cost and robustness objectives, as shown by the selection of the Pareto efficient solutions. We observe that in a realistic or simulated setting a non feasible solution could be actuated anyway, using a proper control algorithm, able to adjust it in real time to feasible values (see for instance (Ziliaskopoulos, 2000)) or a vehicle behavior model, in case traffic is not controllable.

On the other hand, from a practical point of view, our solutions may yield some undesired feature, such as the traffic holding back behavior. Such solutions may require vehicles to mandatorily stop at nodes even if the surrounding traffic conditions would practically allow to proceed their journey toward neighboring nodes. However, it has been proved that it is always possible to convert such solutions to a holding-free assignment pattern, without increasing the total system cost (Shen and Zhang (2014)). Development of such algorithms constitute a line research line of its own, which is outside the scope of the present work. Further research directions involve testing the method effectiveness in the case of non-triangular traffic congestion relationships, or on different problem settings, as the UE-DTA model developed in Ukkusuri and Waller (2008).

Acknowledgements

This research has been partially supported by Compagnia di San Paolo, Torino, Italy.

References

Arbib, C., Muccini, H., & Moghaddam, M. T. (2018). Applying a network flow model to quick and safe evacuation of people from a building: a real case. In *Proceedings of the GEOSAFE workshop on robust solutions for fire fighting, RSFF 2018, l'aquila, italy, july 19-20, 2018.* (p. 50-61). Retrieved from `http://ceur-ws.org/Vol-2146/paper02.pdf`

Bayram, V. (2016). Optimization models for large scale network evacuation planning and management: A literature review. *Surveys in Operations Research and Management Science*, *21*, 63-84.

Bell, M., & Iida, Y. (1997). *Transportation network analysis.* New York: John Wiley & Sons.

Ben-Tal, A., Chung, B. D., Mandala, S., & Yao, T. (2011). Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. *Transportation Research Part B*, *45*, 1177-1189.

Ben-Tal, A., Ghaoui, L. E., & Nemirovski, A. (2000). Robust semidefinite programming. In L. V. H. Wolkowicz R. Saigal (Ed.), *Handbook of semidefinite programming.* Kluwer.

Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, *99*, 351-376.

Ben-Tal, A., & Nemirovski, A. (1998). Robust convex optimization. *Mathematics of operations research*, *23*(4), 769–805.

Calafiore, G. (2010). Random convex programs. *Siam Journal of Optimization*, *20*, 3427-3464.

Carey, M., Bar-Gera, H., Watling, D., & Balijepalli, C. (2014). Implementing first-in-first-out in the cell transmission model for networks. *Transportation Research Part B Methodological*, *65*, 105-118.

Chung, B., Yao, T., Xie, C., & Thorsen, A. (2011). Robust optimization model for a dynamic network design problem under demand uncertainty. *Network and Spatial Economics*, *11*, 371-389.

Como, G., Lovisari, E., & Savla, K. (2016). Convexity and robustness of dynamic traffic assignment and freeway network control. *Transportation Research Part B: Methodological*, *91*, 446–465.

Daganzo, C. (1994). The cell transmission model: a simple dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation research, Part B*, *28*, 269-287.

Daganzo, C. (1995). The cell transmission model. II: network traffic. *Transportation research, Part B*, *29*, 79-93.

Ehrgott, M. (2005). *Multicriteria optimization* (Vol. 491). Springer Science and Business Media.

Florian, M., Mahut, M., & Tremblay, N. (2008). Application of a simulation-based dynamic traffic assignment model. *European Journal of Operational Research*, *189*(1), 1381-1392.

Friesz, T., Kwon, C., & Bernstein, D. (2007). Analytical dynamic traffic assignment models. In D. Hensher & K. Button (Eds.), *Handbook of transport modelling: 2nd edition* (Vol. 1, p. 221-237). Emerald Insight.

Garrett, M., & Wachs, M. (1996). *Transportation planning on trial: The clean air act and travel forecasting.* SAGE Publications.

Islam, T., Vu, H. L., Panda, M., & Ngoduy, D. (2018). A study of realistic dynamic traffic assignment with signal control, time-scale, and emission. *Journal of Intelligent Transportation Systems*, *22*(5), 446-461.

Julvez, J., & Boel, R. K. (2010). A continuous petri net approach for model predictive control of traffic systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *40*(4), 686–697.

Kachroo, P., & Ozbay, K. (2006). Modeling of network level system-optimal real-time dynamic traffic routing problem using nonlinear h-∞ feedback control theoretic approach. *Journal of Intelligent Transportation Systems*, *10*(4), 159-171.

Karoonsoontawong, A., & Waller, S. (2007). Robust dynamic continuous network design problem. *Transportation Research Record: Journal of the Transportation Research Board*, *2029*, 58-71.

Kim, Y., Kato, T., Okuma, S., & Narikiyo, T. (2008). Traffic network control based on hybrid dynamical system modeling and mixed integer nonlinear programming with convexity analysis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *38*(2), 346–357.

Knoop, V., Hoogendoorn, S., & van Zuylen, H. (2008). Capacity reduction at incidents: empirical data collected from a helicopter. *Transportation Research Record: Journal of the Transportation Research Board*(2071), 19–25.

Lim, G., Rungta, M., & Baharnemati, R. (2015). Reliability analysis of evacuation routes under capacity uncertainty of road links. *IIE Transactions*, *1*, 50-63.

Long, J., & Szeto, W. (2019). Link-based system optimum dynamic traffic assignment problems in general networks. *Operations Research*, *67*(1), 167-182.

Long, J., Wang, C., & Szeto, W. (2018). Dynamic system optimum simultaneous route and departure time choice problems: Intersection-movement-based formulations and comparisons. *Transportation Research Part B Methodological*, *115*, 166-206.

Luedtke, J., Ahmed, S., & Nemhauser, G. L. (2010). An integer programming approach for linear programs with probabilistic constraints. *Mathematical Programming*, *122*(2), 247–272.

Ma, Z., Cui, D., & Cheng, P. (2004). Dynamic network flow model for short-term air traffic flow management. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *34*(3), 351–358.

Merchant, D., & Nemhauser, G. (1978a). A model and an algorithm for the dynamic traffic assignment problems. *Transportation Science*, *12*(3), 183-199.

Merchant, D., & Nemhauser, G. (1978b). Optimality conditions for a dynamic traffic assignment model. *Transportation Science*, *12*(3), 200-207.

Oppenheim, N., et al. (1995). *Urban travel demand modeling: from individual choices to general equilibrium.* John Wiley and Sons.

Ozaltin, O., Hunsaker, B., & Schaefer, A. (2011, Summer 2011). Predicting the solution time of branch-and-bound algorithms for mixed-integer programs. *INFORMS Journal on Computing*, *23*(3), 392-403.

Peeta, S. (1994). *System optimal dynamic traffic assignment in congested networks with advanced information systems.* Ph.D. Thesis, University of Texas at Austin.

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

Peeta, S., & Ziliaskopoulos, A. (2001). Foundations of dynamic traffic assignment: the past, the present and the future. *Networks and Spatial Economics*, *1*, 233-265.

Prekopa, A. (1970). On probabilistic constrained programming. In *Proceedings of the princeton symposium on mathematical programming* (pp. 113–138).

Quian, Z., Shen, W., & Zhang, H. (2012). System-optimal dynamic traffic assignment with and without queue spillback: Its path-based formulation and solution via approximate path marginal cost. *Transportation Research Part B*, *46*, 874-893.

Rasouli, S., & Timmermans, H. (2012). Uncertainty in travel demand forecasting models: literature review and research agenda. *Transportation Letters*, *4*(1), 55-73.

Samaranayake, S., Reilly, J., Krichene, W., Lespiau, J., Delle Monache, M., Goatin, P., & Bayen, A. (2015). Discrete-time system optimal dynamic traffic assignment (so-dta) with partial control for horizontal queuing networks. In *American control conference (acc), 2015* (pp. 663–670).

Shen, W., & Zhang, H. (2014). System optimal dynamic traffic assignment: Properties and solution procedures in the case of a many-to-one network. *Transportation Research Part B Methodological*, *65*, 1-17.

Smith, B., Qin, L., & Venkatanarayana, R. (2003). Characterization of freeway capacity reduction resulting from traffic accidents. *Journal of Transportation Engineering*, *129*(4), 362–368.

Sun, H., Gao, Z., & Zhao, F. (2014). Dynamic network design problem under demand uncertainty: An adjustable robust optimization approach. *Hindawi - Discrete Dynamics in Nature and Society*, *2014*, Article ID 739656.

Transportation Networks for Research Core Team. (2019). *Transportation networks for research.* `https://github.com/bstabler/TransportationNetworks`. (Accessed: 2019-03-22)

Tuerprasert, K., & Aswakul, C. (2010). Multiclass cell transmission model for heterogeneous mobility in general topology of road network. *Journal of Intelligent Transportation Systems*, *14*(2), 68-82.

Ukkusuri, S., Mathew, T., & Waller, S. (2007). Robust transportation network design under demand uncertainty. *Computer-Aided Civil and Infrastructure Engineering*, *22*, 6-18.

Ukkusuri, S., & Waller, S. (2008). Linear programming models for the user and system optimal dynamic network design problem: Formulations, comparisons and extensions. *Networks and Spatial Economics*(8), 383-406.
Waller, S., & Ziliaskopoulos, A. (2006). A chance-constrained based stochastic dynamic traffic assignment model: Analysis, formulation and solution algorithms. *Transportation research, Part C*, *14*(6), 418-427.
Wardrop, J. (1952). Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers*, *1*(3), 325–362.
Yang, C., Chen, A., Xu, X., & Wong, S. (2013). Sensitivity-based uncertainty analysis of a combined travel demand model. *Transportation Research Part B: Methodological*, *57*(Supplement C), 225 - 244. Retrieved from http://www.sciencedirect.com/science/article/pii/S0191261513001215
Yazici, A., Kamga, C., & Ozbay, K. (2015). Evaluation of incident management impacts using stochastic dynamic traffic assignment. *Transportation Research Procedia*, *10*, 186–196.
Yazici, M., & Ozbay, K. (2007). Impact of probabilistic road capacity constraints on the spatial distribution of hurricane evacuation shelter capacities. *Transportation Research Record: Journal of the Transportation Research Board*, *2022*, 55-62.
Zhao, Y., & Kockelman, K. (2002). The propagation of uncertainty through travel demand models: an exploratory analysis. *The Annals of regional science*, *36*(1), 145–163.
Ziliaskopoulos, A. (2000). A linear programming model for the single destination system optimum dynamic traffic assignment problem. *Transportation science*, *34*(1), 37-49.