WORKING PAPER

A Trade-off between Average and Maximum Arc Congestion Minimization in Traffic Assignment with User Constraints

E. Angelelli
V. Morandi
M.G. Speranza

WPDEM 1/2018

A trade-off between average and maximum arc congestion minimization in traffic assignment with user constraints

E. Angelelli (1) *V. Morandi* (2) *M.G. Speranza* (3)

(1) *Department of Economics and Management, University of Brescia, Italia, enrico.angelelli@unibs.it*

(2) *Department of Economics and Management, University of Brescia, Italia, valentina.morandi1@unibs.it*

(3) *Department of Economics and Management, University of Brescia, Italia, grazia.speranza@unibs.it*

Abstract

In system optimal traffic assignment of traffic flows with user constraints the total travel time is minimized on a set of paths with bounded length ensuring a certain level of fairness for users. Minimizing the total travel time may lead to experience large travel times on some arcs. On the other hand, when minimizing the maximum arc travel time, the total travel time cannot be controlled. The increase of arc travel time with respect to arc free-flow travel time is related to the arc congestion level that is one of the main issues in road networks. In this paper we propose a new model that is a compromise between minimizing the average arc congestion level and the worst arc congestion level. The model is a linear program that minimizes the average arc congestion over a given percentage of the most congested arcs. A heuristic algorithm aimed at reducing the number of paths that feed the model is also proposed. A computational study is performed that shows the flexibility of the model and the quality of the heuristic.

Keywords: traffic assignment, congestion, linear programming, CVaR.

1 Introduction

Traffic congestion regulation is one of the most important issues to be considered in a city environment. Route guidance is a natural way to alleviate congestion on a road network. The IoV (Internet of Vehicles), i.e. the infrastructure providing information on traffic level, helps in finding the best route for a single driver given the current traffic situation. However, when sat-nav devices are fed with the same information, they provide the same route for drivers travelling from an origin to a destination. As a result, congestion may be simply shifted to other arcs of the road network. Thus, users coordination, by means of a traffic assignment policy, is crucial to alleviate congestion. The technological advances in vehicles design and the advent of self-driving vehicles make the perspective of coordination of users paths more realistic than only a few years ago.

Traffic assignment on a road network consists in assigning demand from each origin to each destination (OD pair) to paths in such a way that an objective function is optimized. Among traffic assignment models proposed in the literature, we investigate a system optimal assignment of traffic flows with user constraints since it embeds the system point of view (minimizing a function related to the travel time experienced on the road network) and the users point of view (allowing only paths that ensure a certain level of fairness). The first attempt to compute a system optimal assignment of traffic flows with user constraints is due to Jahn et al. (2000) and successively improved in Jahn et al. (2005). Other contributions can be found in Schulz and Stier-Moses (2006) and Lujak et al. (2015). These papers propose non-linear models. The first attempt to use linear programming models was proposed in Angelelli et al. (2016a) where a hierarchical approach using a constant latency function has been developed. This approach is considered reliable only if the arc traffic flow is lower than a certain threshold. In order to overcome this limitation, in Angelelli et al. (2016b) a linear programming model, in which a flow-dependent latency function is embedded, is presented.

Traditionally, system optimum traffic assignment optimizes the total travel time over all network arcs (see Jahn et al. (2005), Lujak et al. (2015) and Angelelli et al. (2016b)). In some cases the optimization concerns only the arc experiencing the worst travel time (or related measures) in the network (as in Correa et al. (2007) and Angelelli et al. (2016a)). Optimizing a measure involving all network arcs could lead to situations in which a small set of arcs is highly delayed with respect to the free-flow travel time. On the other hand, optimizing the worst case could lead to a very poor result in terms of average value. In a traffic assignment context, it is important to have a good average value, but controlling the worst cases is equally important especially when specific regulations are imposed to the traffic planner (a threshold on air pollutant emissions, noise pollution control in restricted areas, etc.). Furthermore, as shown in Zhang and Batterman (2013), the air pollution and noise pollution on an arc are strongly related to the arc congestion, intended as the increase of travel time with respect to the free-flow travel time. Thus, controlling the arc congestion in terms of increase of travel time means controlling also these two environmental factors.

In this paper a compromise solution between optimizing the average arc congestion and the maximum arc congestion is proposed. The idea is to minimize the average congestion over a given percentage of the most congested arcs. Traditionally, the experienced travel time on an arc is evaluated through the *latency function* in which the experienced travel time depends on several parameters, related to the type of road, and on the arc traffic flow rate. The congestion on arc (i, j) coping with an arc flow rate x_{ij} is defined as the ratio between the travel time experienced by users $t_{ij}(x_{ij})$ and the free-flow travel time $t_{ij}^{FF} = t_{ij}(0)$, known as the travel time index, weighted by the arc flow rate $t_{ij}(x_{ij})/t_{ij}^{FF} \times x_{ij}$. The model proposed in this paper adopts a piecewise approximation of the latency function, similar to the one proposed in Angelelli et al. (2016b). The percentage of the most congested arcs over which the congestion is minimized is a choice of the regulator. The model, called β-Average Constrained System Optimum model (A-C-SO(β)), allows us to minimize as particular cases, on one extreme, the average arc congestion, and, on the other extreme, the maximum arc congestion. The model also limits the user inconvenience by considering only a selected subset of the possible paths for each OD pair guaranteeing a certain level of fairness, and turns out to be a Mixed Integer Linear Program. The idea of optimizing over a percentage of the worst outcomes was proposed in the financial field in Rockafellar and Uryasev (2000) (an useful tutorial can be found in Sarykalin et al. (2008)), where the measure adopted is called Conditional Value-at-Risk. Several applications in other fields have been also proposed (see Filippi et al. (2017) for a survey). In Filippi et al. (2016), the use of the measure in Mixed Integer Linear Programming is discussed and the measure is called β-average.

The remainder of the paper is organized as follows. In Section 2, we introduce and discuss the A-C-SO(β) model with some basic properties. In Section 3, we propose a heuristic for the A-C-SO(β) model. In Section 4, we discuss the results of an extensive study of the A-C-SO(β) model and the heuristic. Finally, in Section 5, we present some concluding remarks.

2 The β-average constrained system optimum model

The problem is defined on a graph $G = (V,A)$ representing the road network. Vertices V represent road junctions while arcs A represent oriented road segments between junctions. For each arc $(i,j) \in A$, a decision variable x_{ij} gives the rate of vehicles entering arc (i,j) in a steady-state situation. A latency function $t_{ij}(x_{ij})$, evaluating the arc (i,j) traversing time as a non-decreasing convex function of the entering flow rate, is assigned to each arc (i,j). A set C of OD pairs is defined where each OD pair $c \in C$ has origin $O_c \in V$, destination $D_c \in V$ and positive demand d_c defining the rate of vehicles from O_c to D_c.

The relative difference between the length of path k and the shortest path for OD pair c is called *path inconvenience*. For a fixed percentage γ, called *maximum inconvenience*, the set of eligible paths K_c^γ associated with each OD pair $c \in C$ is defined as the set of paths from O_c to D_c such that their inconvenience is less than or equal to γ. For each OD pair $c \in C$ and for each path $k \in K_c^\gamma$, the demand routed on k is specified by the decision variable y_{ck}. Variables x_{ij} and y_{ck} are linked by the parameter a_{ij}^{ck} whose value is 1 if arc (i,j) belongs to path $k \in K_c^\gamma$, and 0 otherwise.

Arc congestion experienced by a single user on a given arc (i,j) is evaluated using the travel time index, i.e. the ratio $t_{ij}(x_{ij})/t_{ij}^{FF}$, weighted by the arc flow rate. This measure can be used in comparing arc congestion on different roads and it quantifies, for each unit of free-flow travel time, how much time is needed to traverse the road for all vehicles traversing the arc. The arc congestion measure is clearly a non-linear function of the arc flow rate x_{ij}. In order to linearize the model, an approximated version of it is adopted: $\varepsilon_{ij}(x_{ij}) \approx \frac{t_{ij}(x_{ij})}{t_{ij}^{FF}} x_{ij}$ (for the linearization technique, see Angelelli et al. (2016b)).

The idea is to fix an upper bound on the flow rate x_{ij}, U_{ij}, and to approximate each non-linear term $\frac{t_{ij}(x_{ij})}{t_{ij}^{FF}} x_{ij}$ on the range $[0, U_{ij}]$ of the potential flow rate on arc (i,j) by a piecewise linear function.

The range $[0, U_{ij}]$ is partitioned in n flow rate intervals at fixed break-points $B = \{b_{ij}^0 = 0, b_{ij}^1, ..., b_{ij}^{n-1}, b_{ij}^n = U_{ij}\}$ with corresponding values $\varepsilon_{ij} = \{\varepsilon_{ij}^0 = 0, \varepsilon_{ij}^1 = \frac{t_{ij}(b_{ij}^1)}{t_{ij}^{FF}} b_{ij}^1, ..., \varepsilon_{ij}^n = \frac{t_{ij}(U_{ij})}{t_{ij}^{FF}} U_{ij}\}$. The interval width is denoted by $\Delta_{ij}^h = b_{ij}^h - b_{ij}^{h-1}$ $(h = 1, \ldots, n)$.

As pointed out in the introduction, when minimizing the average arc congestion value among all arcs, a few arcs may turn out to be heavily congested and the remaining arcs almost empty in view of having the average arc congestion value as small as possible. On the other hand, a model in which the maximum arc congestion is minimized could help in controlling the worst case but may lead to a high average arc congestion. In order to alleviate these drawbacks, a linear programming model that is able to return a trade-off solution is proposed in this section. The trade-off consists in minimizing the average arc congestion of the $\lceil \beta |A| \rceil$ most congested arcs in the network. This allows us to control the arcs in the right tail of the arc congestion distribution. We follow the method to transform a generic mixed-integer linear programming into a new mixed-integer linear programming optimizing the $\lceil \beta |A| \rceil$ worst terms of objective function, recently proposed in Filippi et al. (2016). The result of applying this method to the average arc congestion minimization is the following β-Average Constrained System Optimum model (A-C-SO(β)).

The A-C-SO(β) model

$$\Gamma_\beta \equiv \min \lceil \beta |A| \rceil \, \omega + \sum_{(ij)\in A} \eta_{ij}$$

$$\text{s.t.} \quad \lceil \beta |A| \rceil \, (\omega + \eta_{ij}) \geq \varepsilon_{ij} \qquad \forall (i,j) \in A \qquad (1)$$

$$x_{ij} = \sum_{h=1}^{n} \lambda_{ij}^{h} \qquad \forall (i,j) \in A \qquad (2)$$

$$\varepsilon_{ij} = \sum_{h=1}^{n} \frac{\varepsilon_{ij}^{h} - \varepsilon_{ij}^{h-1}}{\Delta_{ij}^{h}} \lambda_{ij}^{h} \qquad \forall (i,j) \in A \qquad (3)$$

$$0 \leq \lambda_{ij}^{h} \leq \Delta_{ij}^{h} \qquad \forall (i,j) \in A \quad \forall h = 1,\ldots,n \qquad (4)$$

$$x_{ij} = \sum_{c\in C} \sum_{k\in K_c^\gamma} a_{ij}^{ck} y_{ck} \qquad \forall (i,j) \in A \qquad (5)$$

$$d_c = \sum_{k\in K_c^\gamma} y_{ck} \qquad \forall c \in C \qquad (6)$$

$$x_{ij} \geq 0 \qquad \forall (i,j) \in A \qquad (7)$$

$$y_{ck} \geq 0 \qquad \forall c \in C \quad \forall k \in K_c^\gamma \qquad (8)$$

$$\eta_{ij} \geq 0 \qquad \forall (i,j) \in A. \qquad (9)$$

Auxiliary variable λ_{ij}^h represents the amount of the flow rate x_{ij} assigned to interval $[b_{ij}^{h-1}, b_{ij}^{h}]$ as in constraints (4). The arc flow rate on arc x_{ij} is evaluated as the sum over all λ_{ij}^h variables as in constraints (2). Constraints (3) set the arc congestion on arc (i,j) as the sum over all the pieces in the piecewise function of the h-th slope multiplied by the corresponding λ_{ij}^h. Constraints (5) link the flow rate x_{ij} to flow rates on paths containing the arc (i,j). Constraints (6) ensure that the demands d_c is completely routed. Constraints (7)-(8) are non-negativity constraints. Variables η_{ij} and variable ω are auxiliary variables. Variables η_{ij} are set to be non-negative as in constraints (9) while ω is unbounded.

We point out that the average arc congestion minimization and the maximum arc congestion minimization are particular cases of the A-C-SO(β) model in which $\beta = 1$ and $\beta = \frac{1}{|A|}$, respectively. In Table 1 the notation used in the model is summarized.

When optimizing with respect to a certain β value, the traffic planner could also look at the behaviour of the arc congestion on a broader or on a smaller arc set. In order to do this, we propose an additional measure, denoted by Γ_β^κ, representing the average arc congestion value among the most congested arcs resulting from the A-C-SO(β) model. This measure allow us to derive some considerations on the accuracy of the A-C-SO(β) assignment when the traffic regulator is also interested in optimizing the κ most congested arcs.

In the following, some properties for the A-C-SO(β) model are derived.

Definition 1. *Let Γ_β^κ be the average arc congestion on the κ-th quantile of the most congested arcs for the traffic assignment produced by the* A-C-SO(β) *model for $\beta \in (0,1]$ and $\kappa \in (0,1]$.*

The following remark trivially states that the average arc congestion value among the κ most congested arcs resulting from the A-C-SO(β) assignment is always greater than the one obtained by using κ as β parameter in the A-C-SO(β) model. Since Γ_κ is the assignment for which the average arc congestion on the κ most congested arcs is minimized, any other assignment will trivially produce an average arc congestion on the κ most congested arcs that is greater or equal to Γ_κ.

Remark 1. *For any $\beta \in (0,1]$ and $\kappa \in (0,1]$, the inequality*

$$\Gamma_\beta^\kappa \geq \Gamma_\kappa$$

holds.

The following proposition states that, considering the assignment produced by the A-C-SO(β) model, the larger the κ value considered is the smaller the value the correspondent average arc congestion value is.

Proposition 1. *Let $\beta \in (0,1]$ and $\kappa^{'}, \kappa^{''} \in (0,1]$ with $\kappa^{'} \leq \kappa^{''}$. Then,*

$$\Gamma_\beta^{\kappa^{'}} \geq \Gamma_\beta^{\kappa^{''}}.$$

Proof. Given the assignment produced by the A-C-SO(β) model, arcs can be sorted in decreasing order with respect to ε_{ij} forming the vector $f = [a_1, ..., a_{|A|}]$ with $a_n \geq a_{n+1}$ for $n = 1, .., |A| - 1$. Let $k^{'} = \left\lceil \kappa^{'} |A| \right\rceil$ and $k^{''} = \left\lceil \kappa^{''} |A| \right\rceil$. Trivially, $k^{'} \leq k^{''}$. By A-C-SO(β) model definition, $\Gamma_\beta^{\kappa^{'}} = \frac{1}{k^{'}} \sum_{n=1}^{k^{'}} a_n$ and $\Gamma_\beta^{\kappa^{''}} = \frac{1}{k^{''}} \sum_{n=1}^{k^{''}} a_n$. The equivalence between the objective function Γ_β and the average arc congestion of the $\lceil \beta |A| \rceil$ most congested arcs is shown in Filippi et al. (2016).

Now,

$$\begin{aligned}
\Gamma_{\beta}^{\kappa''} &= \frac{1}{k''}(\sum_{n=1}^{k'} a_n + \sum_{n=k'+1}^{k''} a_n) = \frac{1}{k''}(k'\Gamma_{\beta}^{\kappa'} + \sum_{n=k'+1}^{k''} a_n) \\
&\le \frac{k'}{k''}\Gamma_{\beta}^{\kappa'} + \frac{1}{k''}\sum_{n=k'+1}^{k''} a_{k'} = \frac{k'}{k''}\Gamma_{\beta}^{\kappa'} + \frac{k''-k'}{k''}a_{k'} \\
&\le \frac{k'}{k''}\Gamma_{\beta}^{\kappa'} + \frac{k''-k'}{k''}\Gamma_{\beta}^{\kappa'} = \Gamma_{\beta}^{\kappa'}.
\end{aligned}$$

In the chain of inequalities we used the fact that $\max\{a_n \mid n = k'+1, \dots, k''\} \le a_{k'}$ and $a_{k'} \le \Gamma_{\beta}^{\kappa'}$. □

Finally, the following proposition shows that the objective function of the A-C-SO(β) model is a monotonic decreasing function of β.

Proposition 2. *Let β' and β'' be two parameters in $[0,1]$ with $\beta' \le \beta''$. Then,*

$$\Gamma_{\beta''} \le \Gamma_{\beta'}.$$

Proof. From Proposition 1 we have $\Gamma_{\beta'}^{\beta''} \le \Gamma_{\beta'}^{\beta'}$

Given $\beta' \le \beta''$, for Proposition 1, $\Gamma_{\beta}^{\beta''} \le \Gamma_{\beta}^{\beta'}$ for any β. In particular, it is true for β' and, hence, $\Gamma_{\beta'}^{\beta''} \le \Gamma_{\beta'}^{\beta'}$. From Proposition 1 we have $\Gamma_{\beta''} \le \Gamma_{\beta'}^{\beta''}$ and, trivially, $\Gamma_{\beta'}^{\beta'} = \Gamma_{\beta'}$. Hence, $\Gamma_{\beta''} \le \Gamma_{\beta'}^{\beta''} \le \Gamma_{\beta'}^{\beta'} = \Gamma_{\beta'}$. □

3 A heuristic for the β-average constrained system optimum model

The presented model requires the complete enumeration of all eligible paths from origin to destination for each OD pair. However, as proved in Angelelli et al. (2016a), the number of paths may grow exponentially with the instance size, and the model becomes computationally intractable when large size instances are considered. Furthermore, computational experiments show that only a very small number of the generated paths are used in the optimal solution.

Notation

Sets

V	set of vertices
A	set of arcs
C	set of OD pairs
K_c^γ	set of eligible paths for $c \in C$ with maximum inconvenience γ

Parameters

γ	maximum inconvenience
d_c	demand for OD pair $c \in C$
U_{ij}	maximum flow rate allowed on arc $(i, j) \in A$
t_{ij}^{FF}	free-flow travel time of arc $(i, j) \in A$
a_{ij}^{kc}	1 if path $k \in K_c^\gamma$ contains arc $(i, j) \in A$, 0 otherwise
n	number of intervals
b_{ij}^h	h-th breakpoint related to the flow range $\left[0, U_{ij}\right]$, $(h = 0, \dots, n))$
ε_{ij}^h	value of the function $\frac{t_{ij}(x_{ij})}{t_{ij}^{FF}} x_{ij}$ at breakpoint b_{ij}^h, $(h = 0, \dots, n)$
$\Delta_{ij}^h = b_{ij}^h - b_{ij}^{h-1}$	h-th interval size, $(h = 1, \dots, n)$

Decision variables

y_{ck}	flow rate of OD pair $c \in C$ routed on path $k \in K_c^\gamma$
x_{ij}	total flow rate entering arc $(i, j) \in A$: $x_{ij} = \sum\limits_{c \in C} \sum\limits_{k \in K_c^\gamma} a_{ij}^{kc} y_{ck}$
ε_{ij}	piecewise linear approximation of the arc congestion $\frac{t_{ij}(x_{ij})}{t_{ij}^{FF}} x_{ij}$
λ_{ij}^h	amount of flow rate in the h-th flow interval of arc (i, j)
η_{ij}	auxiliary variable for the A-C-SO(β) model associated with arc (i, j)
ω	auxiliary variable for the A-C-SO(β) model.

Table 1. Notation related to the proposed model

Since most of the computational issues are related to the number of paths, the idea is to generate only those paths that are more likely involved in the optimal solution. In this section, a heuristic method called *Heuristic β-average* algorithm (H-A(β)), able to produce a path set on which the A-C-SO(β) model returns a near-optimal solution, is proposed.

Algorithm H-A(β) solves a sequence of restricted versions of the A-C-SO(β) model in which only a small subset of the eligible paths are considered. The idea is to start with a solution provided by a minimal set of paths and to iteratively add a few paths to the current path set in order to improve the current solution.

The algorithm stops when no new paths are found or a maximum number of iterations is reached.

The H-A(β) algorithm is sketched in Algorithm 1. The most relevant variables are: variable P^{Curr} (initialized to $\emptyset$) represents the restricted path set; variable x^{Curr} represents the optimal solution of the restricted A-C-SO(β); variable Γ_{β}^{Curr} represents the objective value of the restricted A-C-SO(β); variable L represents the set of new promising paths found in an attempt to enlarge set P^{Curr} and improve the current solution and is initialized with the set of the shortest paths for each OD pair in the network G; variable C_{crit} represents the set of the critical pairs, that are those OD pairs with at least one active path (with non-zero flow) traversing one or more of the $\lceil \beta |A| \rceil$ most congested arcs and, finally, variable $Iter$ is an iteration counter.

At each iteration the set L containing the new paths is added to the restricted set P^{Curr}. Then, the restricted A-C-SO(β) model is solved on the new set P^{Curr}. Variables x^{Curr} and Γ_{β}^{Curr} are updated and variables x^{Curr} are used to evaluate the new congestion values ε_{ij} for each arc and to construct the set of the critical OD pairs C_{crit}. The set of improving paths L is searched for by means of the routines $LeastcongestedPaths$ and $ModifiedShortestPaths$ that will be described in Section 3.1. Finally, the counter $Iter$ is updated. The algorithm stops as soon as one of the termination criteria is met.

3.1 Searching for an improving path set

At each iteration of the H-A(β) algorithm, the optimal solution x^{Curr} of the restricted A-C-SO(β) model is computed and the set C_{crit} is updated. At this point, two path searching routines are run with the aim to find, for at least one critical OD pair $c \in C_{crit}$, some paths avoiding at

Algorithm 1: H-A(β) algorithm

```
input  : $G$ : graph of the road network,
         $C$ : set of OD pairs,
         $\gamma$ : maximum inconvenience,
         $MaxIter$,
         $ConvThreshold$
output: $\Gamma_\beta^H$ heuristic objective value of A-C-SO($\beta$)
global : $G, C, \gamma, P^{Curr}$
– $P^{Curr} := \emptyset$;
– $L :=$ set of shortest paths for each $c \in C$;
– $Iter := 0$;
do
    – $P^{Curr} := P^{Curr} \bigcup L$;
    – Solve the restricted A-C-SO($\beta$) on path set $P^{Curr}$;
    – $\Gamma_\beta^{Curr} :=$ objective function value;
    – $x^{Curr} :=$ optimal solution of the restricted A-C-SO($\beta$);
    for $(i, j) \in A$ do
        $-\varepsilon_{ij} = \frac{t_{ij}(x_{ij}^{Curr})}{t_{ij}^{FF}} x_{ij}^{Curr}$;
    – $C_{crit} :=$ set of critical OD pairs;
    – $L^1 := LeastcongestedPaths(x^{Curr}, C_{crit})$;
    – $L^2 := ModifiedShortestPaths(x^{Curr}, C_{crit})$;
    – $L := L_1 \bigcup L_2$;
    – $Iter++$;
while $(L \neq \emptyset \wedge Iter < MaxIter)$
– $\Gamma_\beta^H := \Gamma_\beta^{Curr}$;
– return $\Gamma_\beta^H$
```

least one of the arcs for which the OD pair is considered critical.

For each $c \in C_{crit}$, the idea is, for each most congested arc traversed by paths of the OD pair, to remove the arc from the network and, then, to search for the cheapest path for the OD pair first in terms of arc congestion ε_{ij} (*LeastcongestedPaths* routine), evaluated using the current x^{Curr} solution, and, secondly, in terms of free-flow travel time (*ModifiedShortestPaths* routine). In both routines, we check the feasibility of new paths. If no path with the required properties is found, the routines return an empty set. Routine *LeastcongestedPaths* is sketched in Algorithm 2. The set L^1 represents an auxiliary path set in which new paths are added and is initialized

Algorithm 2: LeastcongestedPaths

```
input  : $x^{Curr}$ : traffic assignment to arcs, $C_{crit}$
output: $L^1$ improving path set
global : $G, C, \gamma, P^{Curr}$
– $L^1 := \emptyset$;
for $c \in C_{crit}$ do
    – $H_c :=$ find the arcs in the $\lceil \beta |A| \rceil$ most congested arcs traversed by the OD pair $c$;
    for $h \in H_c$ do
        – $A' := A \setminus \{h\}$;
        – $G' = (V, A')$;
        – $p :=$ shortest path in $G'$ from $O_c$ to $D_c$ with respect to arc weights $\varepsilon_{ij}$;
        – $nlp :=$ length of $p$ with respect to arc length;
        – $nlsp :=$ length of shortest path from $O_c$ to $D_c$ with respect to arc length;
        if $nlp \leq (1+\gamma)nlsp \wedge p \notin P^{Curr}$ then
            – $L^1 := L^1 \bigcup \{p\}$;
– return $L^1$
```

as empty. Let H_c be the subset of the most congested arcs traversed by c and C_{crit} the set of pairs whose paths contain at least one of the arc in H_c. For each OD pair in C_{crit}, the algorithm removes the arc from the graph G and searches for the shortest path on a network considering ε_{ij} as arc weight. If eligible, it is added to L^1.

Routine *ModifiedShortestPaths* is sketched in Algorithm 3. As for routine *LeastcongestedPaths*, the set L^2 represents an auxiliary path set in which new paths are added and it is initialized as empty. The only difference with respect to the previous routine is that, in searching for the shortest path, we use t_{ij}^{FF} as arc weight and, if the new path is eligible, it is added to L^2.

Algorithm 3: ModifiedShortestPaths

```
input  : $x^{Curr}$ : traffic flow rate, $C_{crit}$
output: $L^2$ improving path set
global : $G, C, \gamma, P^{Curr}$
– $L^2 := \emptyset$;
for $c \in C_{crit}$ do
    – $H_c$ := find the arcs in the $\lceil \beta |A| \rceil$ most congested arcs traversed by the OD pair $c$;
    for $h \in H_c$ do
        – $A' := A \setminus \{h\}$;
        – $G' = (V, A')$;
        – $p$ := shortest path in $G'$ from $O_c$ to $D_c$ with respect to arc weights $t_{ij}^{FF}$;
        – $nlp$ := length of $p$ with respect to arc length;
        – $nlsp$ := length of shortest path from $O_c$ to $D_c$ with respect to arc length;
        if $nlp \leq (1+\gamma) nlsp \wedge p \notin P^{Curr}$ then
            – $L^2 := L^2 \bigcup \{p\}$;
– return $L^2$
```

4 Computational results

A benchmark of 40 fixed size instances and a benchmark of 8 growing size instances has been used in a computational study to assess the performance of the presented model and the heuristic algorithm. The 40 fixed size instances used in Section 4.2 are 150 nodes networks and differ in terms of number of nodes and distribution, OD pairs and demand patterns. The 8 growing size instances (from 120 to 330 nodes) have been generated and analyzed in Section 4.3 in order to show the efficiency of the H-A(β) algorithm. Instances are available at `http://or-brescia.unibs.it/instances`. Details on the instance generation process can be found in Angelelli et al. (2016a). For each instance, a traffic assignment has been found using a restricted path set with maximum inconvenience γ ranging from 0% to 25% with increments of 5% and β values in $B \equiv \{\frac{1}{|A|}, 0.01, 0.03, 0.05, 0.07, 0.1, 0.25, 1\}$ (i.e., 48 traffic assignments). The linear programs are solved using CPLEX 12.6.0. The experiments were conducted on a Windows 64-bit computer with Intel Xeon processor E5-1650, 3.50 GHz, and 16 GB Ram. For all experiments, the latency function proposed by the U.S. Bureau of Public Roads,

$$t_{ij}(x_{ij}) = t_{ij}^{FF}[1 + 0.15(\frac{x_{ij}}{u_{ij}})^4],$$

has been used. Shape parameters $t_{ij}^{FF} = t_{ij}(0)$ (free-flow travel time) and u_{ij} are associated to each arc (details on u_{ij} parameters generation can be found in Angelelli et al. (2016a)). We considering as natural upper bound on the arc (i, j) flow $U_{ij} = 4u_{ij}$. The maximum number of iterations allowed for the H-A(β) algorithm is $MaxIter = 20$.

The statistics collected for each instance are described in Section 4.1. Results for the A-C-SO(β) model are presented and discussed in Sections 4.2. Results for the proposed heuristic are presented in Section 4.3. For an easier understanding, results presentation relies on graphics and tables.

4.1 Statistics

In the following all the computed and collected statistics are defined.

- **Arc congestion distribution**
  - Γ_β: optimal value of the A-C-SO(β) model, i.e. average arc congestion on the $\lceil \beta |A| \rceil$ most congested arcs for the traffic assignment produced by the A-C-SO(β) model.
  - Γ_β^κ: average arc congestion on the $\lceil \beta |A| \rceil$ most congested arcs for the traffic assignment produced by the A-C-SO(β) model.
  - Γ_β^H: heuristic value produced by the H-A(β) algorithm.
  - $R_\beta^\kappa = \frac{\Gamma_\beta^\kappa - \Gamma_\kappa}{\Gamma_\kappa}$: relative difference of Γ_β^κ with respect to Γ_κ.
  - $A_\beta = \sum_{\kappa \in K} \alpha_\kappa R_\beta^\kappa$: performance measure of the optimal solution produced by the A-C-SO(β) model obtained as a convex combination ($\sum_{\kappa \in K} \alpha_\kappa = 1$) of R_β^κ values.
  - ε_{ij}: congestion on arc (i, j) resulting from the A-C-SO(β) model.
- **User experience for each OD pair** $c \in C$
  - t_{ck}: experienced travel time on path $k \in K_c^\gamma$.
  - t_{cSP}: free-flow travel time on the shortest path for OD pair c.

- $I_{FF} = \frac{1}{\sum_{c \in C} d_c} \sum_{c \in C} \sum_{k \in K_c^{\gamma}} y_{ck} \frac{t_{ck} - t_{cSP}}{t_{cSP}}$: total weighted experienced inconvenience with respect to free-flow travel time.

- **H-A(β) approximation error**
  - $\tau_\beta = \frac{\Gamma_\beta^H - \Gamma_\beta}{\Gamma_\beta}$: the relative error produced by algorithm H-A(β) with respect to A-C-SO(β).

- **Computational time**
  - A-C-SO(β) computational time: accounts for the time to generate the paths and to solve the A-C-SO(β) linear model.
  - H-A(β) computational time: accounts for the iterated path generation and model solution.

- **Memory usage** The number of generated paths.

4.2 Optimal solutions

The traffic assignment produced by the A-C-SO(β) model is analyzed for maximum inconvenience γ ranging from 0% to 25% with increments of 5% and for β values in

$$B \equiv \{\frac{1}{|A|},\ 0.01,\ 0.03,\ 0.05,\ 0.07,\ 0.1,\ 0.25,\ 1\}.$$

Reported values are averaged over the 40 instances. Figure 1 shows the objective function, Γ_β, as a function of γ. Let us observe the behaviour of Γ_β with $\beta = \frac{1}{|A|}$. As expected, Γ_β is monotone non-increasing with respect to γ (the larger γ is the more relaxed the model is). We observe the most relevant drop in Γ_β when passing from $\gamma = 0\%$, where only the shortest paths are eligible, to $\gamma = 5\%$. The decrease rate of Γ_β is getting lower and lower as γ increases; from $\gamma = 15\%$, Γ_β becomes almost steady. The same behaviour can be observed with increasing value of β. We observe that the curves become more and more flat as β increases. Dominance between curves with different values of β is guaranteed by Proposition 2.

In Tables 2-6, the behaviour of the R_β^κ statistics, i.e. the relative difference of Γ_β^κ with respect to Γ_κ, with $\beta \in B$ and $\kappa \in B$, is reported. Each table is computed for a different value

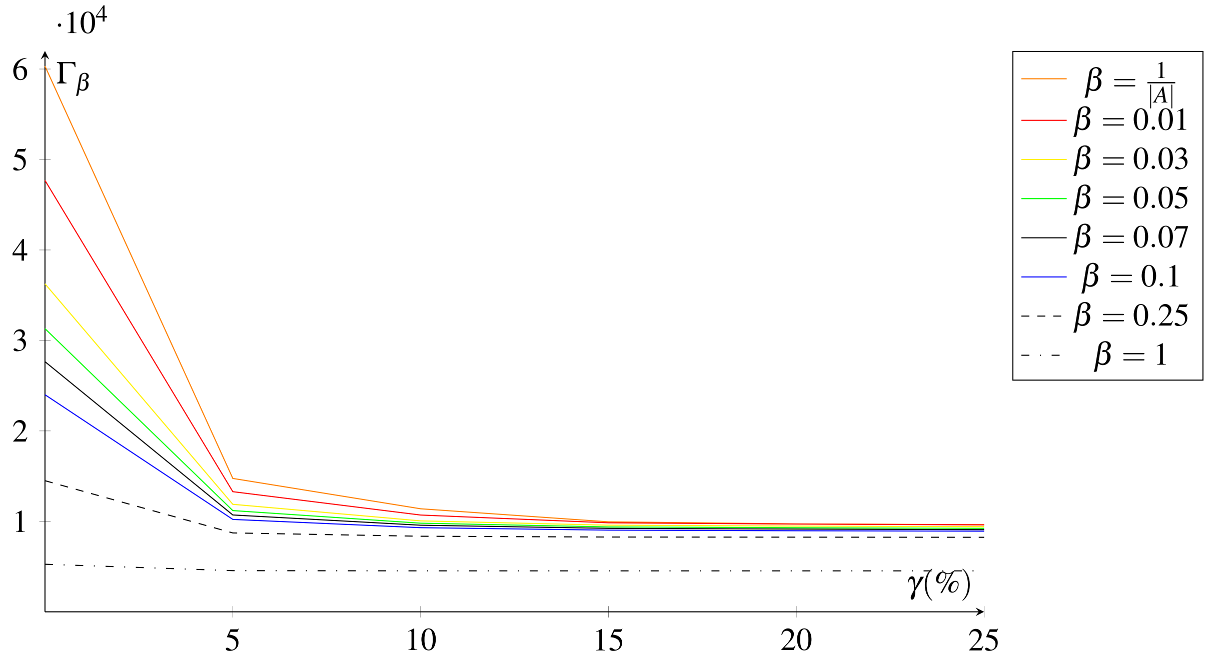


Figure 1. Γ_β as a function of γ

of γ. However, a table for $\gamma = 0\%$ is not reported as in this case only shortest paths are allowed and no choice is allowed with respect to β. Values of β are reported through rows and values of κ are reported through columns. Thus, in each row we have a fixed value of β and we are considering the same optimal solution of A-C-SO(β). Elements on the same rows show, for each κ, the average arc congestion increase of the κ-quantile with respect to the potential value that could be obtained if we choose $\beta = \kappa$. Conversely, each column shows the behaviour of R^κ_β when κ is fixed and β changes. Obviously, the entries in the main diagonal of each table equal 0 as $\Gamma^\beta_\beta = \Gamma_\beta$. Accordingly, it seems reasonable to expect monotonicity in entries: reading either along a row or a column we may expect a smaller value as we approach the main diagonal. This is true reading by rows, while it is not always true reading by columns. Monotonicity by columns is broken a few times. For instance, consider $\kappa = \frac{1}{|A|}$. We note that using $\beta = 1$ is better than using $\beta = 0.1$ or $\beta = 0.25$. However, exceptions of this kind are very rare in Tables 2-6, and happen only for small κ and big β. In general, the closer the κ value is to the γ value the lower the R^κ_β value is but this behaviour is not monotonic. Greater values are obtained for $\gamma = 5\%$. Looking at Table 2, where R^κ_β statistics for $\gamma = 5\%$ are shown, we see that with $\beta = \frac{1}{|A|}$, the overall average arc congestion ($\kappa = 1$) is about 221% of the average arc congestion produced having $\beta = 1$ while, allowing $\beta = 0.01$, the resulting average arc congestion is about 156% of the average arc congestion produced having $\beta = 1$. In Tables 3-6 we recognize the same pattern as in Table 2, but values are generally lower with bigger values of γ.

As already pointed out, the closer κ is to a fixed β the lower the statistic R^{κ}_{β} is. The key point in choosing the right β value is the trade-off between the different R^{κ}_{β} values with the whole set of κ values.

	κ	$\frac{1}{\lvert A\rvert}$	0.01	0.03	0.05	0.07	0.1	0.25	1
	$\frac{1}{\lvert A\rvert}$	0	9.3	21.7	29.4	35.5	42.2	92.4	220.5
	0.01	3.1	0	5.4	10.6	14.9	20	52	156.4
	0.03	5.5	1.7	0	1.2	3.2	6.3	21.2	64.8
β	0.05	6.9	3.1	0.7	0	0.5	2.4	21.5	102.9
	0.07	7.9	4	1.6	0.6	0	0.8	14.5	89.2
	0.1	9.5	4.7	2.6	2	0.9	0	9.3	77
	0.25	13.5	7.9	5.7	5.3	4.3	3.2	0	35.2
	1	8.3	5.5	4.9	4.9	4.3	3.6	0.8	0

Table 2. R statistics (%) with $\gamma = 5\%$ for different κ and β values

	κ	$\frac{1}{\lvert A\rvert}$	0.01	0.03	0.05	0.07	0.1	0.25	1
	$\frac{1}{\lvert A\rvert}$	0	5.8	12.7	15.6	18.4	21.8	35.5	77.7
	0.01	5.1	0	2	3.8	5.8	8.6	20.3	60.8
	0.03	9.7	1.9	0	0.7	2.1	4.4	15.1	55.5
β	0.05	12.5	5.8	1	0	0.4	2	11.4	50.9
	0.07	14	8.6	3.3	0.6	0	0.7	8.6	47.2
	0.1	17.5	11.1	6.5	3.2	1	0	4.7	42.1
	0.25	27.3	19	12.9	9.6	7.2	4.9	0	29.8
	1	20	16.9	14.4	11.7	9.7	8	3.8	0

Table 3. R statistics (%) with $\gamma = 10\%$ for different κ and β values

	κ	$\frac{1}{\lvert A\rvert}$	0.01	0.03	0.05	0.07	0.1	0.25	1
	$\frac{1}{\lvert A\rvert}$	0	0.9	3.6	5.7	7.5	9.8	19.9	65.1
	0.01	1.4	0	1.6	3.5	5.1	7.3	17.1	61.3
	0.03	12.1	3	0	0.8	1.9	3.7	12.5	57.3
β	0.05	18.7	8.5	0.7	0	0.3	1.5	9.2	53
	0.07	20	11.4	2.7	0.5	0	0.5	7.1	50.2
	0.1	23.2	14.3	6.1	2.8	0.6	0	4.5	46.2
	0.25	39.5	25.3	16	12	9	6.1	0	31.8
	1	32.8	25.7	19	15.7	13.1	10.5	4.6	0

Table 4. R statistics (%) with $\gamma = 15\%$ for different κ and β values

To this aim, in Figure 2, the function A_{β}, with increasing γ values, is plotted for each β value. The function A_{β} represents a performance measure of the optimal solution produced by

	κ	$\frac{1}{\lvert A\rvert}$	0.01	0.03	0.05	0.07	0.1	0.25	1
β	$\frac{1}{\lvert A\rvert}$	0	0.1	2.4	4.4	6.1	8.2	17.5	65
	0.01	0.5	0	2	3.9	5.6	7.7	16.8	63.3
	0.03	13.8	3	0	0.9	2	3.6	11.9	58.2
	0.05	20.3	9.8	0.8	0	0.1	1.1	8.1	54.3
	0.07	22.9	12.3	2.4	0.4	0	0.5	6.7	52.3
	0.1	27.2	16.4	7	3	0.8	0	3.9	48.2
	0.25	40.6	26.2	16.8	12.5	9.4	6.3	0	34.3
	1	36	27.4	20.4	17	14.2	11.4	4.7	0

Table 5. R statistics (%) with $\gamma = 20\%$ for different κ and β values

	κ	$\frac{1}{\lvert A\rvert}$	0.01	0.03	0.05	0.07	0.1	0.25	1
β	$\frac{1}{\lvert A\rvert}$	0	0.1	2.2	4.3	5.9	7.8	16.7	62.3
	0.01	0.7	0	1.7	3.7	5.3	7.2	15.9	59.8
	0.03	12.9	2.7	0	1	2.1	3.6	11.6	56.5
	0.05	21.1	9.7	0.6	0	0.2	1.1	7.9	52.8
	0.07	23.7	13.4	2.8	0.6	0	0.2	5.9	49.7
	0.1	26.9	15.8	6.4	2.8	0.8	0	3.9	46.9
	0.25	40.6	26.3	16.8	12.8	9.6	6.4	0	34.1
	1	37.2	28.7	21.3	17.8	14.9	11.9	4.9	0

Table 6. R statistics (%) with $\gamma = 25\%$ for different κ and β values

the A-C-SO(β) model obtained as a convex combination of R^{κ}_{β} values. This measure reflects the impact of choosing a certain β value on all the other κ-quantiles considered. Furthermore, weights can be chosen depending on the preferences of the traffic regulators assigning an importance ranking to different κ-quantiles. In case of recurrent bottlenecks, the focus should be on the most congested arcs and, hence, on low number of arcs κ-quantiles while, in case of widespread congestion, the focus should be on κ-quantiles that consider a higher number of arcs. Here we consider a generic case in which weights are all equal to $\frac{1}{|K|}$. Using the proposed weights, the highest performance function A_{β} value is obtained with $\beta = 1$ as long as the maximum inconvenience γ remains under 10%. Once higher values of γ are considered, the performance function A_{β} value is lower for $\beta < 0.1$. This means that, if we consider a broader path set ($\gamma \geq 10\%$) it is convenient to consider the A-C-SO(β) model with small β values in terms of performance function.

Even though the proposed model is focused on a system perspective in which the arc congestion on a set of arcs is minimized, the users satisfaction is a key point in considering a traffic assignment. In Figure 3 the total weighted experienced inconvenience with respect to free-flow

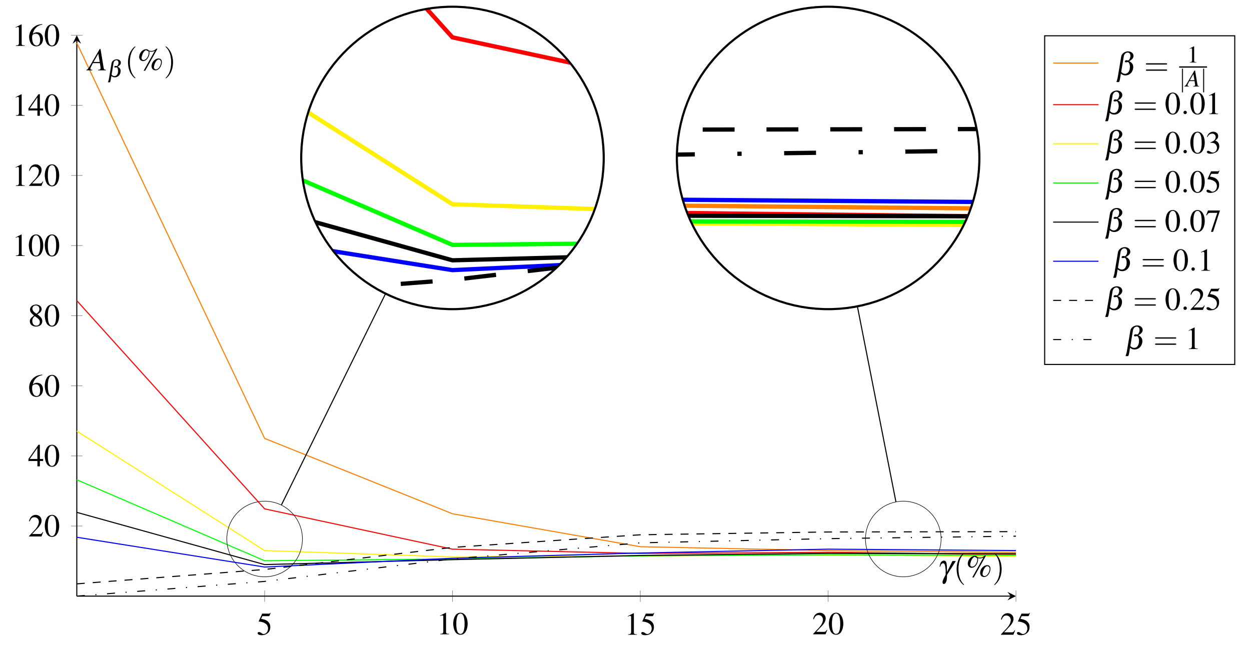


Figure 2. A_β curves for different β values

travel time I_{FF} is shown. We recall that the experienced inconvenience is evaluated as the relative difference between the user experienced travel time and the user free-flow travel time from origin to destination. The proposed plot highlights that, until γ values under 10% are considered, the assignments are substantially indifferent in terms of inconvenience. From that point, the smaller β is, the higher the average inconvenience experienced by users is. The reason is that, when $\beta = 1$, the model will tend to choose paths containing less arcs as in the following example. Let the network be the one in Figure 4 and let the flow rate from O to D be 20. In Figure 4(a) the free-flow travel time and parameter u of each arc is shown. When $\beta = 1$, we are considering the average among all arc congestion values. Sending OD pair flow rate on the shortest path, as in Figure 4(b), will produce an average arc congestion level equal to $\frac{68}{5} = 13.6$. On the other hand, sending OD pair flow rate on the longest path, as in Figure 4(c), will produce an average arc congestion level equal to $\frac{92}{5} = 18.4$. Thus, choosing the shortest path is more convenient than the longest path when $\beta = 1$. On the contrary, when lower enough values of β are considered, it is more convenient to send flow on the longest path since the objective is to minimize the average over the $\lceil \beta|A| \rceil$ most congested arcs. For instance, with $\beta = 25\%$, the objective is to minimize the average over the two most congested arcs. Choosing the shortest path the objective value is equal to $\frac{68+0}{2} = 34$ while choosing the longest path the objective

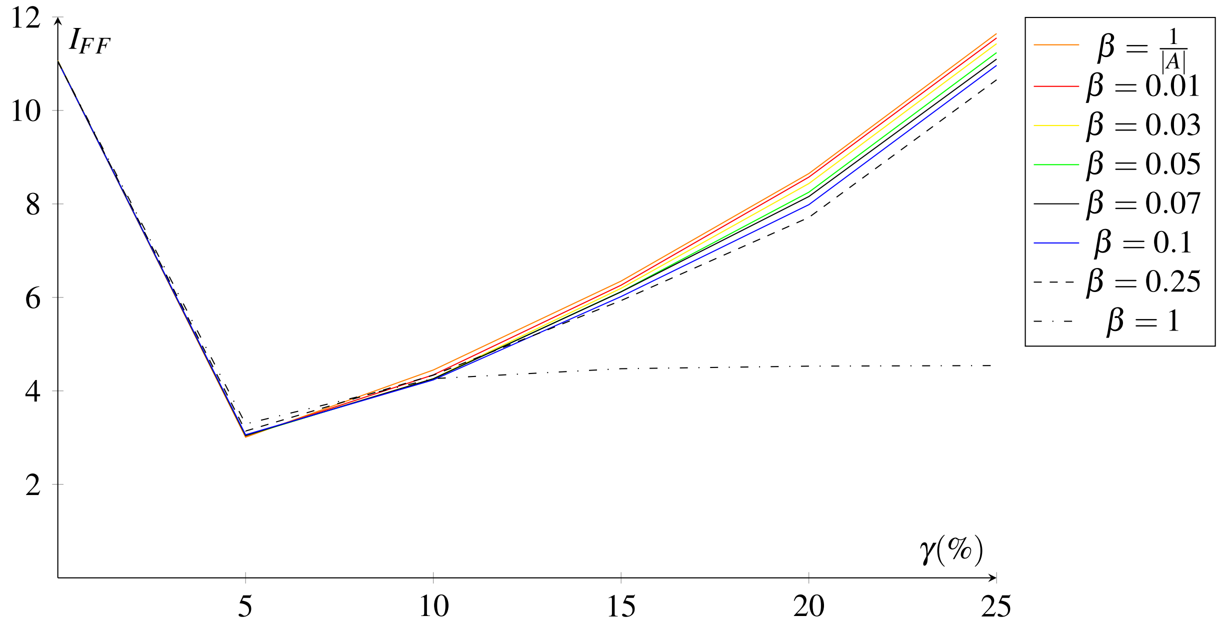
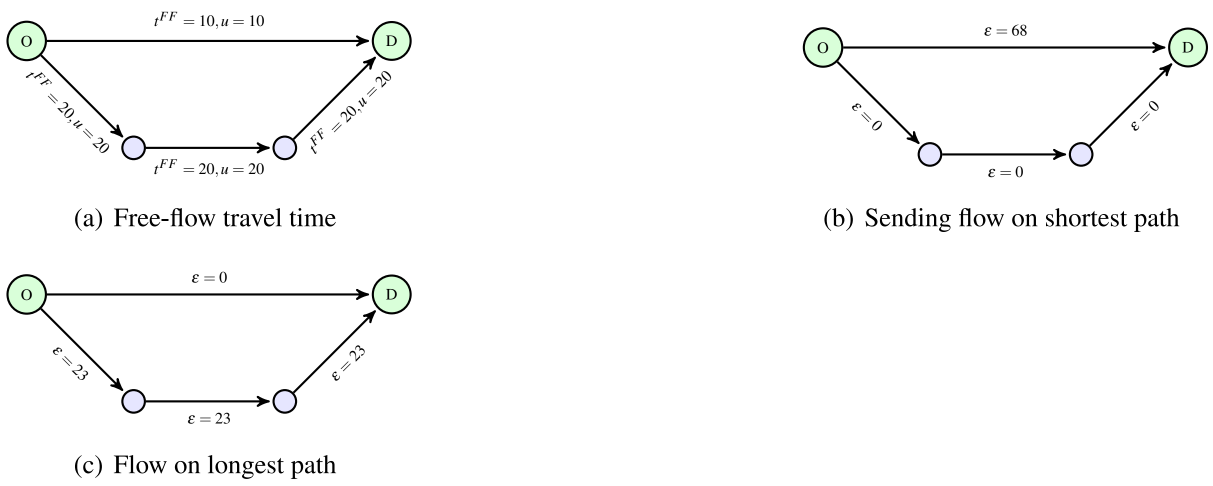


Figure 3. I^{FF} curves for different β values

value is equal to $\frac{23+23}{2} = 23$.



(a) Free-flow travel time

(b) Sending flow on shortest path

(c) Flow on longest path

Figure 4. An example

In this section, we examined the objective function values, the A_β measures and the experienced inconvenience averaged over all instances. To better explore the arc congestion distribution, we next investigate a single instance for $\gamma = 25\%$.

The experiment uses an instance with 150 nodes, 480 arcs, and 1170 OD pairs, and gen-

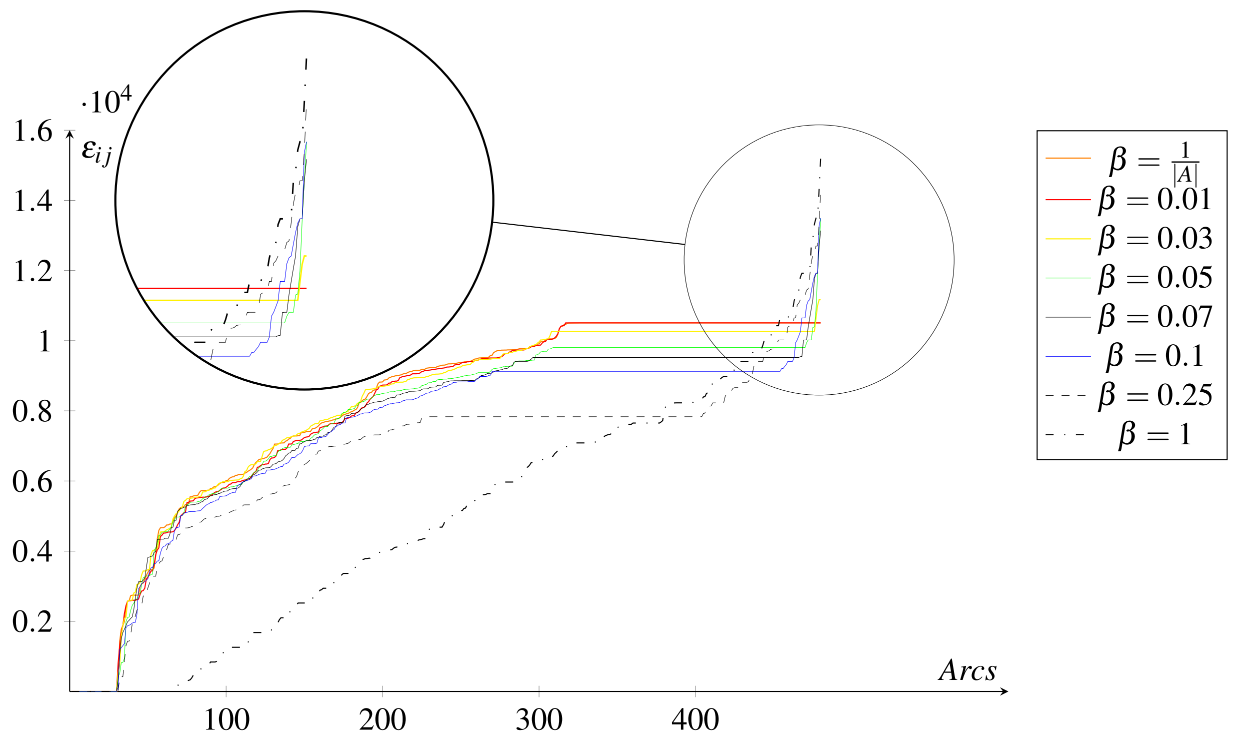


Figure 5. $\gamma = 25\%$ curves

eration parameter values: oligo-centric, in-peak, and high in-city traffic (the instance name is RG_Big_oligo_heavytraffic_10_A from the same benchmark instances repository used for averaged results and explanations about used parameters can be found in Angelelli et al. (2016a)). In Figure 5 the arc congestion distribution for $\gamma = 25\%$ is shown. The arc congestion distribution is obtained showing the arc congestion level of all arcs increasingly sorted by the arc congestion value, i.e. from the least to the most congested arc. For large part of the arcs, considering $\beta = 1$, the arc congestion level is lower than considering other β values. However, a set of arcs have an arc congestion value that is greater than the one obtained with smaller β values. In the magnified part, the set of arcs for which the phenomenon can be observed is shown. In this particular instance, the difference between the maximum arc congestion value obtained using $\beta = 1$ with respect to $\beta = \frac{1}{|A|}$ is around the 45% (15230 versus 10507).

4.3 Performance of the heuristic against optimal solutions

In this section we summarize the results obtained by algorithm H-A(β) on 40 networks with 150 nodes and values of γ ranging from 0% to 25% with step 5%.

β	Time (sec)	γ 0%	5%	10%	15%	20%	25%
$\frac{1}{\|A\|}$	H-A(β)	56.6	81.3	96.6	91.1	77.3	75.4
	CE	2.3	13.1	31.9	75.5	115.2	313.7
0.01	H-A(β)	63.3	82.3	95.4	85.7	76.7	75.4
	CE	2.4	13.6	30.6	76	114.9	313.9
0.03	H-A(β)	61.6	85	81.9	83.7	75.7	81.2
	CE	2.2	13.7	30.4	77.1	118.8	313.6
0.05	H-A(β)	58.8	86.3	93.1	87	79.2	82.9
	CE	2	13.6	31.7	75	114.5	319.6
0.07	H-A(β)	60.2	95.2	101.1	88.6	83.3	87.8
	CE	2.8	15	36.1	70.1	110.6	296.3
0.1	H-A(β)	61.3	92.4	98	88.3	84.6	88.1
	CE	2	15.6	36	70.2	112.6	298.4
0.25	H-A(β)	68.4	75.12	82.8	96.1	97.2	99.2
	CE	2.2	15.7	31.0	64.0	107.8	263.4
1	H-A(β)	73.8	93.7	105.6	101	98.3	99.2
	CE	2.5	14.5	34.4	65.7	114	269

Table 7. Computational time (sec)

From now on, we denote with CE the A-C-SO(β) model requiring the complete enumeration of all feasible paths from origin to destination. In Table 7, the CE and H-A(β) computational times are shown for different γ and β values. If small γ values are considered, it seems to be better to use the CE model since computational times are better in most cases. However, algorithm H-A(β) is less time consuming when $\gamma > 15\%$ and time savings grow as growing values of γ are considered and the advantage in using the H-A(β) becomes clear. Moreover, the computational time needed by algorithm H-A(β) is almost stable with γ increasing values. The reason is that, considering greater γ values, the number of paths grows and the CE requires a huge number of variables (one for each considered path) and, hence, computational time rapidly grows. On the other hand, algorithm H-A(β) generates a number of paths that is polynomial (in the worst case, at each iteration, it is proportional to the number of arcs in the objective function multiplied by the number of OD pairs).

The memory usage of algorithm H-A(β) is compared to CE in Table 8, where, for different values of β and γ, we report the number of paths generated by the H-A(β) algorithm and the percentage of generated paths with respect to CE. The first row shows the number of paths generated by CE which rapidly grows with increasing values of γ. Note that β does not influence the number of eligible paths. Thus, we showed this number in a single row as far as CE is

		γ					
β	Paths	0%	5%	10%	15%	20%	25%
All βs	CE	1202.5	16389.4	43305	91046.4	160740.1	452059.3
$\frac{1}{\lvert A \rvert}$	H-A(β)	1202.5	7363.3	10708.7	10300.8	10892.9	11626.1
	% on CE	100	44.93	24.73	11.31	6.78	2.57
0.01	H-A(β)	1202.5	7286.3	10550.1	10052.5	10693.1	11439.4
	% on CE	100	44.46	24.36	11.04	6.65	2.53
0.03	H-A(β)	1202.5	7184.1	9802.1	10053.3	10728.9	11544.9
	% on CE	100	43.83	22.64	11.04	6.67	2.55
0.05	H-A(β)	1202.5	6939.7	9672.8	10026.8	11057.2	12106.2
	% on CE	100	42.34	22.34	11.01	6.88	2.68
0.07	H-A(β)	1202.5	6750.9	9509.9	10165.6	11099.2	12361.9
	% on CE	100	41.19	21.96	11.17	6.91	2.73
0.1	H-A(β)	1202.5	6471	9136.4	10041.7	11343.6	12451.1
	% on CE	100	39.48	21.1	11.03	7.06	2.75
0.25	H-A(β)	1202.5	5149.7	7691.4	9996.9	11506.1	12953.1
	% on CE	100	31.42	17.76	10.98	7.16	2.87
1	H-A(β)	1202.5	4862	6674.5	7263.5	7597.3	8454.6
	% on CE	100	29.67	15.41	7.98	4.73	1.87

Table 8. Number of generated paths

concerned. Subsequent rows contain, for each β value, the number of paths generated by the H-A(β) algorithm and the percentage of the H-A(β) generated paths with respect to CE model one for different γ values. If $\gamma = 5\%$, memory savings with respect to the CE are not very high (around a half of generated paths). When γ increases, memory savings become more and more significant. Using $\gamma = 25\%$, memory saving for all β values is around 50 times the CE memory consumption. Observe that for $\gamma \geq 10\%$ the number of generated paths remains almost steady which confirms the analysis about computational times.

In Table 9 statistics on the H-A(β) average relative error τ_β for different β and γ values are shown. Note that, for all β values and $\gamma = 0\%$, the average τ_β is zero. This is because the H-A(β) algorithm (after one iteration) and CE generate the same path set. Considering greater γ values, the average τ_β has a decreasing trend with increasing γ values. However, for some γ values, the behaviour is different. For instance, the H-A(β) algorithm produces an average τ_β with $\gamma = 5\%$ that is lower than one obtained with $\gamma = 10\%$ when $\beta = 0.01$ is considered.

In Figure 6, the τ_β density function on all 1920 H-A(β) runs (Cartesian product between 40 instances, the 8 β values and 6 γ values) is shown.

Note that, in the 20% of the runs, the problem is optimally solved by the H-A(β) algorithm

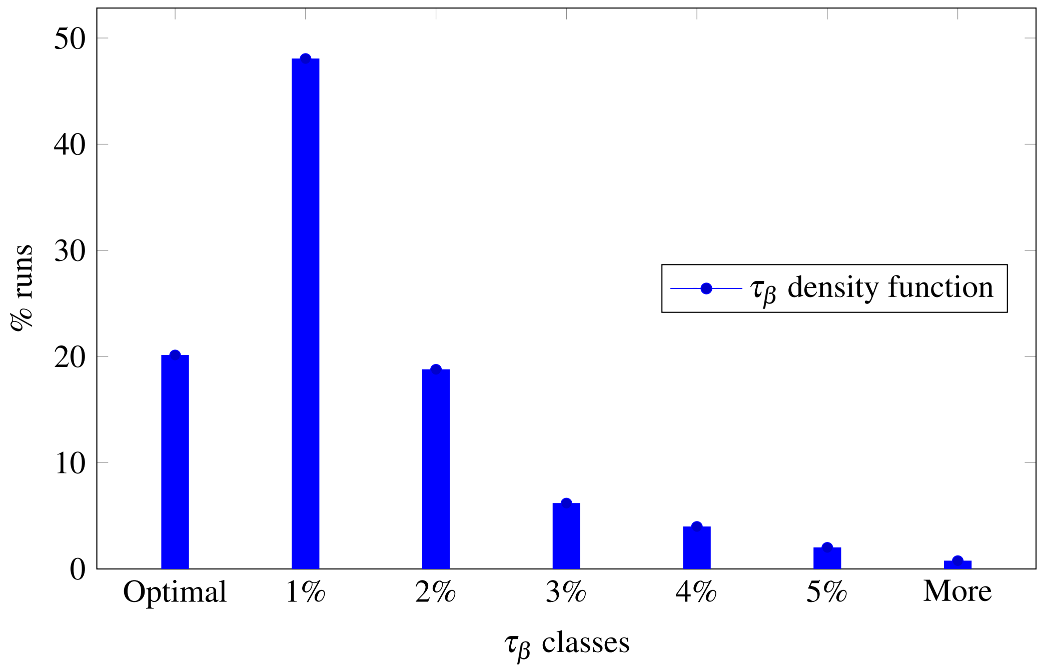


Figure 6. τ_β density function

while, in the 96.97% of the runs, τ_β is lower than 4%. The percentage of runs in which the τ_β value is greater than 5% is very low, around 0.7%. Moreover, this happens only with γ value equal to 5% and 10% where the A-C-SO(β) model can be easily optimally solved.

	γ					
β	0%	5%	10%	15%	20%	25%
$\frac{1}{\|A\|}$	0	0.71	2.31	1.45	0.85	1.07
0.01	0	0.91	1.92	1.32	1.04	1
0.03	0	1.46	1.71	1.15	1.05	1.09
0.05	0	2.1	1.59	1.1	1.05	1.11
0.07	0	2.13	1.38	0.99	0.97	1.09
0.1	0	2.56	1.32	0.73	0.81	0.78
0.25	0	1.92	0.86	0.23	0.24	0.13
1	0	0.21	0.09	0.03	0.03	0.02

Table 9. Average relative error $\tau_\beta(\%)$

4.4 Performance of the heuristic on larger instances

Experiments were carried out considering 8 networks, with a number of nodes that ranges from 120 to 330 with step 30, and γ ranging from 0% to 25% with step 5%. Parameter β is fixed at 0.25. In Tables 10 and 11 the number of generated paths and the computational time for CE and H-A(β) algorithm are reported as well as the relative error $\tau_{0.25}$. Note that, for instances with a number of nodes greater than 210 and for some values of γ, the statistics for algorithm CE are not shown because either the path generation procedure ran out of memory or the solver running time exceeded a time threshold of 10800 seconds (3 hours). The number of paths generated by algorithm CE grows dramatically fast, with respect to algorithm H-A(β), as the number of nodes increases. In rows 'Time CE (sec)' the computational time (including the time needed in generating the path set) is shown, while in rows 'Time H-A(β) (sec)' the H-A(β) computational time is shown in seconds. Rows 'Paths CE' and rows 'Paths H-A(β) ' represent the number of generated paths by the two algorithms. Finally, rows 'Relative error $\tau_{0.25}$ (%)' represent the relative error produced by H-A(β) with respect to the CE with $\beta = 0.25$. With $\gamma = 25\%$, for the 150 nodes instance the number of paths generated by the H-A(β) algorithm (13125) is approximately 2.5% of the number of generated paths by CE (524067) while for the 180 nodes instance the percentage is approximately 0.87% (16403 versus 1896053).

When the instance size grows to 210 nodes this percentage decreases to 0.37% (22462 versus 6104204) and, when it grows to 240 nodes the A-C-SO(β) model runs out of memory. This means that the H-A(β) algorithm produces a number of paths that is less than 2 orders of magnitude of the paths generated by CE. With higher instance sizes there is no available data for $\gamma = 25\%$ since the solver exceeded the time threshold or ran out of memory. However, regardless of the value of γ, the percentage of H-A(β) generated paths with respect to CE continues to decrease with the increase of the instance size.

Regarding the computational time, the time required by algorithm CE grows fast, with respect to H-A(β), as the number of nodes increases. With $\gamma = 25\%$, for the 150 nodes instance the H-A(β) algorithm computational time is approximately 28.64% of the CE computational time while for the 180 nodes instance it is approximately 9.48%. When the instance size grows to 210 nodes the percentage decreases to 4.45% and when it grows to 240 nodes the A-C-SO(β) model runs out of memory. This means that algorithm H-A(β) for large instances takes a time that is around 2 orders of magnitude lower than the time required by CE. Small instances should be solved using the CE since the computational time is reasonable. On the contrary, the number of iterations set for the H-A(β) makes the algorithm slower when small instances are considered. However, a significant number of iterations have to be set in order to let the algorithm converge to the heuristic solution. Like for the number of paths, with larger instance sizes there is no available data for $\gamma = 25\%$ but the percentage continues to decrease with the increase of the instance size. For example, with a 330 nodes instance and $\gamma = 10\%$ the computational time required by H-A(β) is 20% of the time required only for the generation of the eligible path set by algorithm CE and generated paths are the 0.97% of the paths required by CE.

After the analysis of H-A(β) memory and time savings, some conclusions also on its effectiveness in generating high quality solutions can be derived. The relative error τ_β is quite stable with growing instances and relative error is always under 6% for all tested instances. We recall that τ_β is a relative error evaluated on a measure that is a percentage itself. This means that the relative error is a percentage of a percentage. The average computed relative error for all γ and instance sizes is around 1.68%.

5 Conclusions

In this paper a model for the system optimal traffic assignment with users constraints to minimize the congestion over a given percentage of the most congested arcs is proposed. Computational experiments show that the solution produced by the proposed model is a compromise solution between those obtained by the minimization of the average arc congestion and the minimization of the maximum arc congestion. As the computational complexity of the model is mainly affected by the number of paths, we propose a heuristic for the path generation. Computational experiments show that the proposed heuristic reduces by orders of magnitude the number of generated paths and, consequently, by orders of magnitude the amount of memory usage and computational time, while allowing high quality solutions of the model.

Future research directions include the design of a column generation algorithm for the solution of the proposed model. Moreover, an interesting alternative model might minimize the congestion on the most congested paths as an alternative to the most congested arcs, moving the focus from the system to the users inconvenience. Another interesting research direction is

# Nodes	Stats	γ					
		0%	5%	10%	15%	20%	25%
120	Paths CE	942	4526	11143	18796	26797	38973
	Paths H-A(25%)	942	3289	5871	7143	7590	8567
	Time CE (sec)	0.99	2.02	3.13	4.92	6.63	10.05
	Time H-A(25%) (sec)	13.5	16.80	17.65	18.62	19.11	20.43
	Relative error $\tau_{0.05}$ (%)	0	1.29	0.16	0.05	0.0	0.0
150	Paths CE	1188	17340	46741	99111	177108	524067
	Paths H-A(25%)	1188	5165	7742	9874	11199	13125
	Time CE (sec)	1.44	6.22	13.13	27.37	48.07	137.75
	Time H-A(25%) (sec)	28.79	31.13	31.52	33.89	36.54	39.45
	Relative error $\tau_{0.05}$ (%)	0	2.38	0.58	0.25	0.28	0.06
180	Paths CE	1422	17073	78518	243062	652019	1896053
	Paths H-A(25%)	1422	6699	9810	12573	14014	16403
	Time CE (sec)	1.79	9.27	26.43	72.03	206.30	617.79
	Time H-A(25%) (sec)	45.65	47.42	45.51	49.83	53.73	58.6
	Relative error $\tau_{0.25}$ (%)	0	4.88	1.24	0.65	0.91	0.68
210	Paths CE	1650	23914	145011	560441	1813283	6104204
	Paths H-A(25%)	1650	8355	12717	17132	20051	22462
	Time CE (sec)	2.30	14.31	51.14	188.30	657.68	2135.11
	Time H-A(25%) (sec)	73.02	67.08	74.57	83.21	90.28	94.91
	Relative error $\tau_{0.25}$ (%)	0	3.65	2.71	0.80	0.92	0.55

Table 10. Results for different number of nodes in the network

# Nodes	Stats	γ					
		0%	5%	10%	15%	20%	25%
240	Paths CE	18860	39959	284623	1361928	5415058	*
	Paths H-A(25%)	1886	10462	15004	20231	24862	27551
	Time CE (sec)	3.15	25.58	114.986	541.03	2509.53	*
	Time H-A(25%) (sec)	114.38	97.77	106.88	118.50	127.46	133.51
	Relative error $\tau_{0.25}$ (%)	0	4.20	4.58	1.30	1.56	*
270	Paths CE	2142	71463	655685	3612381	15763662	*
	Paths H-A(25%)	2142	13787	19113	25808	30308	33719
	Time CE (sec)	4.27	49.06	290.60	1593.42	9710.30	*
	Time H-A(25%) (sec)	197.1	159.68	167.79	186.41	241.46	203.09
	Relative error $\tau_{0.25}$ (%)	0	5.16	3.83	1.61	1.23	*
300	Paths CE	2367	118394	1260039	7714606	*	*
	Paths H-A(25%)	2367	16761	22153	27695	34085	38547
	Time CE (sec)	5.25	85.09	613.88	3763.67	*	*
	Time H-A(25%) (sec)	281.10	228.73	239.82	249.21	253.34	263.25
	Relative error $\tau_{0.25}$ (%)	0	5.72	3.78	2.41	*	*
330	Paths CE	2669	210832	2853271	*	*	*
	Paths H-A(25%)	2669	20137	27742	36341	41806	46060
	Time CE (sec)	7.38	162.64	1627.74	*	*	*
	Time H-A(25%) (sec)	428.47	304.21	335.58	335.14	386.58	406.28
	Relative error $\tau_{0.25}$ (%)	0	5.64	5.55	*	*	*

Table 11. Results for different number of nodes in the network

a time-dependent extension of the proposed model.

References

E. Angelelli, A. Idil, V. Morandi, S. Martin, and M. G. Speranza. Proactive route guidance to avoid congestion. *Transportation Research Part B: Methodological*, 94:1 – 21, 2016a.

E. Angelelli, V. Morandi, M. Savelsbergh, and M. G. Speranza. System optimal routing of traffic flows with user constraints using linear programming. Technical Report 5, University of Brescia, Department of Economics and Management, 2016b. URL http://www.unibs.it/dipartimenti/economia-e-management/ricerca/working-paper.

J. R. Correa, A. S. Schulz, and N. E. Stier-Moses. Fast, fair, and efficient flows in networks. *Operations Research*, 55:215–225, 2007.

C. Filippi, M. G. Speranza, and W. Ogryczak. Discrete conditional value-at-risk. Technical Report 2, University of Brescia, Department of Economics and Management, 2016. URL http://www.unibs.it/dipartimenti/economia-e-management/ricerca/working-paper.

C. Filippi, G. Guastaroba, and M. G. Speranza. Applications of conditional value-at-risk beyond finance: A literature review. Technical Report 2, University of Brescia, Department of Economics and Management, 2017. URL http://www.unibs.it/dipartimenti/economia-e-management/ricerca/working-paper.

O. Jahn, R. H. Möhring, and A. S. Schulz. Optimal routing of traffic flows with length restrictions in networks with congestion. *Operations Research Proceedings 1999*, pages 437–442, 2000.

O. Jahn, R. H. Möhring, A. S. Schulz, and N. E. Stier-Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations Research*, 53:600–616, 2005.

M. Lujak, S. Giordani, and S. Ossowski. Route guidance: Bridging system and user optimization in traffic assignment. *Neurocomputing*, 151:449–460, 2015.

R. Rockafellar and S. Uryasev. Optimization of Conditional Value-at-Risk. *Journal of Risk*, 2: 21–42, 2000.

S. Sarykalin, G. Serraino, and S. Uryasev. Value-at-Risk vs. Conditional Value-at-Risk in risk management and optimization. In Z.-L. Chen and S. Raghavan, editors, *Tutorials in Operations Research*, pages 270–294. INFORMS, 2008.

A. S. Schulz and N. E. Stier-Moses. Efficiency and fairness of system-optimal routing with user constraints. *Networks*, 48:223–234, 2006.

K. Zhang and S. Batterman. Air pollution and health risks due to vehicle traffic. *Science of the total Environment*, 450:307–316, 2013.