

18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015,
Delft, The Netherlands

Optimizing emergency transportation through multicommodity quickest paths

Anna Melchiori ^{a,*}, Antonino Sgalambro ^a

^aNational Research Council of Italy - Istituto per le Applicazioni del Calcolo "Mauro Picone", Via dei Taurini 19, Roma 00185, Italy

Abstract

In transportation networks with limited capacities and travel times on the arcs, a class of problems attracting a growing scientific interest is represented by the optimal routing and scheduling of given amounts of flow to be transhipped from the origin points to the specific destinations in minimum time. Such problems are of particular concern to emergency transportation where evacuation plans seek to minimize the time evacuees need to clear the affected area and reach the safe zones.

Flows over time approaches are among the most suitable mathematical tools to provide a modelling representation of these problems from a macroscopic point of view. Among them, the Quickest Path Problem (*QPP*), requires an origin-destination flow to be routed on a single path while taking into account inflow limits on the arcs and minimizing the makespan, namely, the time instant when the last unit of flow reaches its destination.

In the context of emergency transport, the *QPP* represents a relevant modelling tool, since its solutions are based on unsplittable dynamic flows that can support the development of evacuation plans which are very easy to be correctly implemented, assigning one single evacuation path to a whole population. This way it is possible to prevent interferences, turbulence, and congestions that may affect the transportation process, worsening the overall clearing time. Nevertheless, the current state-of-the-art presents a lack of studies on multicommodity generalizations of the *QPP*, where network flows refer to various populations, possibly with different origins and destinations.

In this paper we provide a contribution to fill this gap, by considering the Multicommodity Quickest Path Problem (*MCQPP*), where multiple commodities, each with its own origin, destination and demand, must be routed on a capacitated network with travel times on the arcs, while minimizing the overall makespan and allowing the flow associated to each commodity to be routed on a single path. For this optimization problem, we provide the first mathematical formulation in the scientific literature, based on mixed integer programming and encompassing specific features aimed at empowering the suitability of the arising solutions in real emergency transportation plans.

A computational experience performed on a set of benchmark instances is then presented to provide a proof-of-concept for our original model and to evaluate the quality and suitability of the provided solutions together with the required computational effort. Most of the instances are solved at the optimum by a commercial MIP solver, fed with a lower bound deriving from the optimal makespan of a splittable-flow relaxation of the *MCQPP*.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of Delft University of Technology

Keywords: Transportation, Emergency, Evacuation, Quickest Path, Quickest Flow

* Corresponding author. Tel.: +39 06 49270969; fax: +39-064404306.

E-mail address: a.melchiori@iac.cnr.it (Anna Melchiori).

1. Introduction

In transportation networks with limited capacities and travel times on the arcs, a class of problems attracting a growing scientific interest is represented by the optimal routing and scheduling of given amounts of flow to be transhipped from the origin points to the specific destinations in minimum time. Such problems are of particular concern to emergency transportation where evacuation plans seek to minimize the time evacuees need to clear the affected area and reach the safe zones.

Flows over time approaches are among the most suitable mathematical tools to provide a modelling representation of these transportation problems from a macroscopic point of view. Firstly introduced by Ford and Fulkerson (1958, 1962), these methods rightly capture the dynamic evolution of the scenario e.g. the flow variation over time and the time needed for the transshipment, for a complete survey see Aronson (1989), Koehler et al. (2009). Polynomial-time algorithms have been proposed to solve some of these problems; however, in many general cases their NP-hardness has been proven, see Ford and Fulkerson (1962), Hoppe (1995), Hoppe and Tardos (2000), Hall et al. (2003), Klinz and Woeginger (2004). Flows over time problems may be applied to various real-life situation, e.g. road and air traffic control, communication networks, production systems, and financial flows, see Powell et al. (1995), Aronson (1989).

Among this class of approaches, the Quickest Flow Problem (*QFP*) asks for sending a s - t flow (namely, a flow directed from a source node s to a sink node t) taking into account the limitations of inflow on the arcs and such that the last unit of flow arrives at destination as quickly as possible, thus minimizing the makespan of the evacuation process. The *QFP* was shown to be solvable in strongly polynomial time by Burkard et al. (1993). A recent formulation can be found in Lin and Jaillet (2015). The Evacuation Problem is an extension of the *QFP* to multiple origins and a single sink, see Chalmet et al. (1982), Hamacher and Tjandra (2002). For the multicommodity generalization of the *QFP*, namely, the Quickest Multicommodity Flow Problem (*MCQFP*), a valid fully polynomial time approximation scheme has been provided in Fleischer and Skutella (2002, 2003) and Hall et al. (2003). Nevertheless, in a real management of emergency transportation operations, a relevant drawback of the solutions provided by the *QFP* lies in the frequent assignment of multiple and bifurcated paths to the s - t flow demand, requiring thus a group of evacuees to be divided and routed on different ways to destination.

The Quickest Path Problem (*QPP*), extensively treated in the literature, see Chen and Chin (1990) and Pascoal et al. (2006), copes with this issue requiring the origin-destination flow to be routed on a single path.

State-of-the-art presents a lack of studies on the multicommodity generalization of the *QPP*, where each commodity, with its own origin, destination, and flow demand, has to be routed on a single path sharing the same capacitated arcs with the other commodities, while minimizing the makespan. We refer to this as the Multicommodity Quickest Path Problem (*MCQPP*), which reveals its dramatic relevance when unsplitable dynamic flows are necessary to prevent all those sources of interference and congestion that may affect the transportation process worsening the overall clearing time.

The literature presents a related problem for static network flows, the Multicommodity Unsplittable Flow problem introduced by Kleinberg (1996) and the Multicommodity k -splittable flow problem, see Baier et al. (2005), Caramia and Sgalambro (2008, 2010), and Gamst (2014). The unsplitable flow request has been studied by Mawson (2005) and Hamacher et al. (2011) for clusters of people evacuating an area in a single-source graph. Their Quickest Cluster Flow Problem is formulated as a non-linear model that accounts for one cluster size and single flow units (clusters with size 1). For each given cluster a unique path to destination is determined. The NP-hardness has been proven for general and tree networks i.e. single source-multiple sinks directed networks and with several sizes for clusters, see Leiner and Ruzika (2011).

In this paper we provide, to the best of our knowledge, the first mathematical formulation in the scientific literature for the *MCQPP*, based on mixed integer linear programming. Our aim is providing a valid mathematical support to the development of evacuation plans to be used in real emergency transportation scenarios. A relevant issue to be considered is therefore the operational implementation of the evacuation plans, that could be affected by human behaviour as a reaction to the emergency situation. To favour an easy compliance of the interested populations to the emergency transport plans, it is important to avoid as much as possible phenomena such as interferences, turbulence, and congestions, arising in some cases for both vehicles and crowd flows. To this aim, we introduce to the above

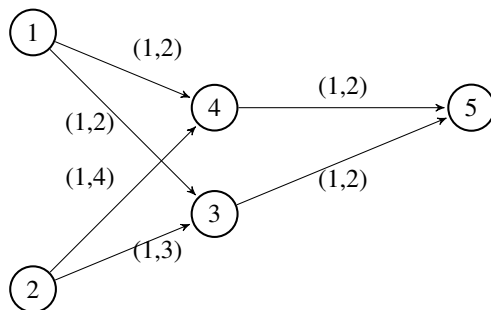


Fig. 1. Arc label: (delay, capacity).

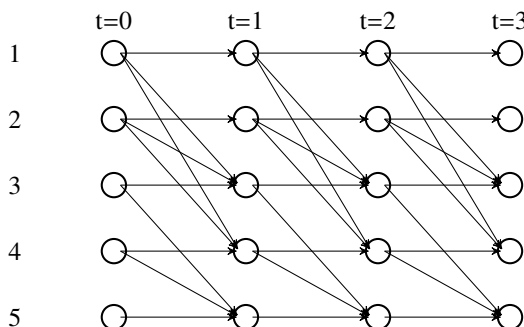


Fig. 2. Time expansion.

mentioned problem additional features allowing each population to move along a continuous and unified path from the origin to the destination, also avoiding those chaotic situations caused by the concurrent overlapping on the same shared arcs of flows from different populations. More in detail, we assume that a whole population might wait at the source before its evacuation process starts, but then all the members of the population are routed through the same unique path in an uninterrupted time sequence and further holdovers are not allowed. Moreover, the concurrent assignment of flows from different commodities to the same arc at the same time is forbidden.

The *MCQPP* is described in detail in Section 2: preliminaries are introduced in Section 2.1, a complete description of the features of the considered problem is provided in Section 2.2, while in Section 2.3 we give the notation and the detailed mixed integer formulation of our constrained *MCQPP*. In Section 3 we present a computational experience performed on a set of benchmark instances to study the effects of the unsplitable flow constraints and other additional restrictions and to provide a proof-of-concept for our original model. Instances are solved by a commercial MIP solver and fed with a lower bound deriving from the optimal makespan of the *MCQFP*. The quality and suitability of the provided solutions are evaluated and compared with the *MCQFP* in terms of solution quality, optimal makespan, and required computational effort. Section 4 presents our conclusions and further developments.

2. The problem

2.1. Preliminaries and notation

In this paper we focus on dynamic network flow models with a discrete representation of the time horizon. A discrete-time dynamic network flow problem is defined on a directed graph $D = (\mathcal{V}, \mathcal{A}, \mathcal{T})$ with \mathcal{V} representing the set of nodes, \mathcal{A} the set of arcs, and \mathcal{T} the time horizon discretized into a finite number of intervals. For every arc $(i, j) \in \mathcal{A}$ we assume its attributes to be time independent: the capacity $c_{ij} \in \mathbb{Q}^+$ defines the maximum number of flow

units that can enter the arc at the same time, and the delay $\lambda_{ij} \in \mathbb{Z}^+$ specifies the time intervals needed to traverse the arc from node i to node j with the flow on the arc progressing at a uniform rate. We assume the capacity of each node to be unlimited. A discrete-time dynamic network flow is the amount of flow at each time interval for each arc in the digraph, formally:

$$x : (\mathcal{A}, \mathcal{T}) \longrightarrow \mathbb{Q}^+$$

Thus, the value of the x_{ijt} variable represents the rate of flow, per time unit, crossing arc (i, j) at time t . A discrete-time dynamic network flow problem can be solved through a time-expansion procedure that transforms it into a static network flow problem on a time-expanded graph, introduced by Ford and Fulkerson (1958, 1962). Given a directed graph $D = (\mathcal{V}, \mathcal{A}, \mathcal{T})$ with constant attributes and a time horizon $\mathcal{T} = [0, \dots, T]$, its time-expanded network (TEN) is a directed graph $D_{\mathcal{T}} = (\mathcal{V}_{\mathcal{T}}, \mathcal{A}_{\mathcal{T}})$ that contains T -replica of the original network. Formally:

$$\begin{aligned} \mathcal{V}_{\mathcal{T}} &:= \mathcal{V} \times \mathcal{T} = \{(i, t) \mid i \in \mathcal{V}, t \in \mathcal{T}\} \\ \mathcal{A}_M &:= \mathcal{A} \times \mathcal{T} = \{(i, t), (j, t') \mid (i, j) \in \mathcal{A}, t' = t + \lambda_{ij} \leq T\} \\ \mathcal{A}_H &:= \mathcal{V} \times \mathcal{T} = \{(i, t), (i, t + 1) \mid i \in \mathcal{V}, t = 0, \dots, T - 1\} \\ \mathcal{A}_{\mathcal{T}} &= \mathcal{A}_M \cup \mathcal{A}_H. \end{aligned}$$

where $\mathcal{V}_{\mathcal{T}}$ contains one copy of the node set of the underlying 'static' network for each discrete time step and \mathcal{A}_M contains the replica for every discrete time step of the original arcs with the same constant attributes. The additional set \mathcal{A}_H contains the *holdover arcs* that allow the flow to wait at a node over a one time period. The capacity of an holdover arc (i, i) is set equal to the i -node capacity. A discrete dynamic network flow can be thus represented as a static flow in the corresponding time-expanded network. This reduction allows to solve some dynamic network flow problems with standard well-known pseudo-polynomial algorithms, see Koehler et al. (2009), Ford and Fulkerson (1958, 1962). The drawback of this approach is the dimension of the static network flow problem to be solved: the more discretized the time horizon is, the larger the time-expanded network is, with relevant consequence on the solvability of the problem (even prohibitive in many cases). On the other side a non accurate partition of the time horizon usually presents a lack of precision to give consistent solutions to real applications.

Figure 2 depicts the time-expansion procedure of the digraph in Figure 1 over a time horizon of three time intervals: the original five nodes, placed in the first vertical column, and arcs are reproduced along the time dimension (the horizontal axis) making a separate copy for every discrete time instant and according to the arcs attributes. Holdover arcs are added to connect each node in the time-expanded network with its replica at the next time layer.

2.2. Problem Description

In this paper we consider the Multicommodity Quickest Path Problem (MCQPP) providing a formal definition as follows. We are given a time horizon discretized into a finite number of intervals, a network with constant attributes over time (limited capacities and travel times on the arcs), and a set of commodities, each with its own source and destination node and with an associated flow demand. Flows must be routed and scheduled on the network in such a way as to respect at any interval time the inflow capacity of shared arcs and minimize the makespan, namely, the time instant when the last unit of flow reaches its destination. Furthermore, each commodity must be assigned to a single path, that is, the entire flow of a commodity has to be enrouted through the same unique path.

The problem represents a relevant and easy to implement tool in the context of emergency transport as it strictly assigns a unique evacuation path to a whole population. Due to operational reasons related to the need of minimizing situations of interferences, turbulence, and congestion during the transportation process, we account for the following additional constraints.

Uninterrupted Scheduling. The presence of limited inflow capacities on the arcs may not allow an entire population to leave its source at the same instant. In this case a commodity can be split into subsets, each of them being scheduled at different time intervals. The first unit of flow leaving the source univocally determines the path for the whole original demand of the commodity. The Uninterrupted Scheduling feature forces subsets of the same commodity to be scheduled on consecutive and uninterrupted time intervals and it prohibits further holdovers. This first additional request is

meant to increase the regularity of the evacuation process and reduce every possible interruption and perturbation in the process once it started. Note that by these constraints a feasible flow for a given commodity corresponds in the time-expanded network to a temporarily and compact repetition of the path, where the number of copies depends on how the flow demand has been scheduled at its origin.

Non-contemporaneity of flows. When more commodities share the same arc at the same time, there is a node in the time-expanded network where the flows coming from different arcs have to be merged. This situation is a recognized source of turbulence in flow dynamics. Hence, in our formulation we impose the non-contemporaneity of different flows on the same arc. This way we also prevent commodities from being mixed, allowing a safer routing of populations through the network.

2.3. Problem Formulation

In the remainder of this section we present our formulation in detail.

Nomenclature

- D = $(\mathcal{V}, \mathcal{A}, \mathcal{T})$ original network
- \mathcal{K} = $\{1, \dots, K\}$ set of commodities
- (s_k, d_k, σ_k) = (origin, destination, demand) of commodity k
- \mathcal{S} = $\{s_k : k \in \mathcal{K}\}$ origin nodes
- \mathcal{D} = $\{d_k : k \in \mathcal{K}\}$ destination nodes
- \mathcal{N} intermediate nodes
- \mathcal{V} = $\{\mathcal{S} \cup \mathcal{N} \cup \mathcal{D}\}$ set of nodes
- \mathcal{A} = $\{(i, j) : i, j \in \mathcal{V}\}$ set of arcs
- \mathcal{T} = $\{1, \dots, T\}$ time horizon

- c_{ij} capacity of the arc (i, j)
- λ_{ij} delay of the arc (i, j)
- $\delta^+(i) = \{j : (i, j) \in \mathcal{A}\}$ outbound arcs from node i
- $\delta^-(i) = \{j : (j, i) \in \mathcal{A}\}$ inbound arcs of node i

- x_{ijt}^k units of flow of commodity k leaving node i at time t heading to node j
- y_{ijt}^k binary variable equals 1 iff commodity k leaves node i at time t heading to node j

Basic Assumptions. For each commodity $k \in \mathcal{K}$ we suppose that no arc is departing from the destination node, that is $\delta_{d_k}^+ = \emptyset$ and no arc enters the origin node except loops, $\delta_{s_k}^- = \{s_k\}$. Furthermore, loops at source nodes have by default delay equal to one, $\lambda_{s_k s_k} = 1 \quad \forall k \in \mathcal{K}$. We optimize the evacuation process asking for each commodity to reach its safe destination within a time horizon \mathcal{T} .

Objective function. The *MCQPP* seeks to minimize the makespan, namely, the time needed to completely clear the given network with each unit of flow demands arrived at its destination. The makespan is represented in the objective function by the ν term.

We also consider an additional term defined as a weighted sum of all the arcs among the time horizon, where holdover arcs at the source nodes present a smaller weight than the other arcs, see (1):

$$C := \sum_{k \in \mathcal{K}, t \in \mathcal{T}} \frac{\sum_{(i,j) \in \mathcal{A}: i \neq j} \lambda_{ij} x_{ijt}^k + \sum_{(i,j) \in \mathcal{A}: i=j} \frac{\lambda_{ij}}{2} x_{ijt}^k}{D |\mathcal{A}| |\mathcal{T}|}$$

where $D = \max_{(i,j) \in \mathcal{A}} \lambda_{ij}$ represents the maximal value of delay among all arcs. The aim of this term is to improve the quality of the proposed solutions penalising evacuation plans that present long evacuation paths, in terms of number of visited nodes, and forcing when possible the model to select solutions with equal makespan and lower number of nodes.

$$\min v + C \tag{1}$$

$$\tilde{c}_{jt}^k (y_{jd_k t}^k - y_{jd_k(t+1)}^k) \leq v \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, j \in \delta^-(d_k). \tag{2}$$

$$\sum_{j \in \delta^+(s_k)} x_{s_k j 0}^k = \sigma_k \quad \forall k \in \mathcal{K}. \tag{3}$$

$$\sum_{j \in \delta^-(d_k)} \sum_{t=0}^{T-\lambda_{jd_k}} x_{jd_k t}^k = \sigma_k \quad \forall k \in \mathcal{K}. \tag{4}$$

$$\sum_{j \in \delta^+(s_k), j \neq s_k} y_{s_k j 0}^k \leq 1 \quad \forall k \in \mathcal{K}. \tag{5}$$

$$y_{s_k s_k(t+1)}^k \leq y_{s_k s_k t}^k \quad \forall k \in \mathcal{K}, t \in \mathcal{T}. \tag{6}$$

$$\sum_{j \in \delta^+(s_k), j \neq s_k} y_{s_k j(t+1)}^k \leq y_{s_k s_k t}^k \quad \forall k \in \mathcal{K}, t \in \mathcal{T}. \tag{7}$$

$$\sum_{j \in \delta^+(i)} y_{ijt}^k \leq 1 \quad \forall k \in \mathcal{K}, i \in \mathcal{N}, t \in \mathcal{T}. \tag{8}$$

$$y_{iit}^k = 0 \quad \forall k \in \mathcal{K}, i \in \mathcal{N}, t \in \mathcal{T}. \tag{9}$$

$$1 + y_{s_k j(t+1)}^k \geq y_{s_k s_k t}^k + y_{s_k j t}^k \quad \forall k \in \mathcal{K}, j \in \delta^+(s_k), j \neq s_k, t \in \mathcal{T}. \tag{10}$$

$$2 + y_{ij(t+1)}^k \geq y_{ijt}^k + y_{mi(t-\lambda_{mi})}^k + y_{mi(t+1-\lambda_{mi})}^k \quad \forall k \in \mathcal{K}, i \in \mathcal{N}, j \in \delta^+(i), m \in \delta^-(i), t \in \mathcal{T} : t - \lambda_{mi} \geq 0. \tag{11}$$

$$\sum_{k \in \mathcal{K}} y_{ijt}^k \leq 1 \quad \forall (i, j) \in \mathcal{A}, i \neq j, t \in \mathcal{T}. \tag{12}$$

$$\sum_{k \in \mathcal{K}} x_{ijt}^k \leq c_{ij} \quad \forall (i, j) \in \mathcal{A}, t \in \mathcal{T}. \tag{13}$$

$$\sum_{j \in \delta^-(i)} x_{ji(t-\lambda_{jt})}^k - \sum_{j \in \delta^+(i)} x_{ijt}^k = 0 \quad \forall k \in \mathcal{K}, i \in \mathcal{N}, t \in \mathcal{T} : t - \lambda_{jt} \geq 0. \tag{14}$$

$$\sum_{j \in \delta^-(i)} y_{ji(t-\lambda_{ji})}^k - \sum_{j \in \delta^+(i)} y_{ijt}^k = 0 \quad \forall k \in \mathcal{K}, i \in \mathcal{N}, t \in \mathcal{T} : t - \lambda_{ji} \geq 0. \tag{15}$$

$$\sum_{j \in \delta^-(s_k)} x_{js_k(t-\lambda_{js_k})}^k - \sum_{j \in \delta^+(s_k)} x_{s_k j t}^k = 0 \quad \forall k \in \mathcal{K}, t \in \mathcal{T} : t > 0. \tag{16}$$

$$x_{ijt}^k \leq c_{ij} y_{ijt}^k \quad \forall k \in \mathcal{K}, (i, j) \in \mathcal{A}, t \in \mathcal{T}. \tag{17}$$

$$y_{ijt}^k \in \{0, 1\} \quad \forall k \in \mathcal{K}, (i, j) \in \mathcal{A}, t \in \mathcal{T}. \tag{18}$$

$$x_{ijt}^k \geq 0 \quad \forall k \in \mathcal{K}, (i, j) \in \mathcal{A}, t \in \mathcal{T}. \tag{19}$$

$$v \geq 0. \tag{20}$$

Constraints. Linear Constraints (2) estimate the makespan of the transportation process. Specifically, they determine the clearing time of each commodity, i.e. the exact instant time in which the last unit of flow reaches its destination. The parameters are defined as $\tilde{c}_{jt}^k = (t + \lambda_{jd_k})$. Constraints (3) impose the origin evacuation node for each commodity; note that they don't force the flow to start the evacuation process at time zero as the set of outbound arcs includes loops. Constraints (4) impose the destination node for the flow of each commodity. Constraints (5), (6) and (7) cope with the path conservation at the source nodes. Specifically, they allow to postpone the start of the transportation process of each commodity by enrouting the original demand through the holdover arc of the source

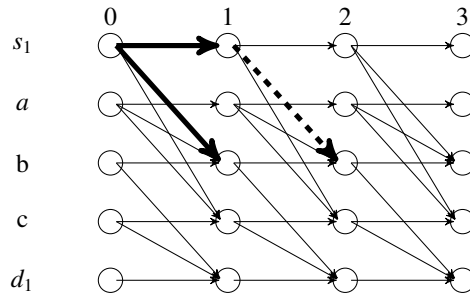


Fig. 3. Effect of constraints (10).

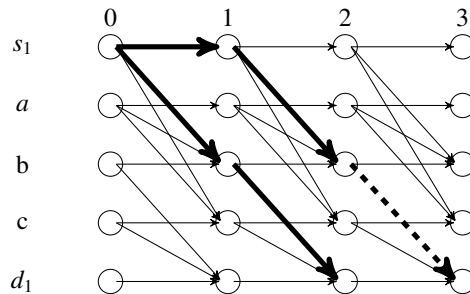


Fig. 4. Effect of constraints (11).

node. The evacuation starts whenever the entire flow or a subset of it is enrouted through a single path. Constraints (8) ensure that every commodity doesn't further split its flow into subsets among the evacuation process. Constraints (9), (10), and (11) represent the core of our formulation as they deal with the routing of each commodity through the same unique path with no holdovers at intermediate nodes and the uninterrupted scheduling of its flow demand. Figures 3 and 4 clarify how these constraints work for a given commodity. We refer to the same original and time-expanded network of Figures 1 and 2 with node 1 as the source and node 5 as the destination. Intermediate nodes are labelled as $a, b,$ and c . Bold arcs in Figure 3 show that at time zero a subset of the original flow demand has been scheduled to start its evacuation through the arc (s_1, b) whereas the rest waits at the source. In this situation the constraint of type (10) with $t = 0, s_k = s_1, j = b$ forces to enrout at the next time interval another subset through the same arc (bold dashed arc). This constraint ensures the uniqueness of the path right from the source avoiding large distances between subsets of the same original flow.

The same idea is applied to intermediate nodes: Figure 4 depicts the effect of Constraints (11) in an hypothetical next step with $t = 1, i = b, m = s_1, j = d_1$. Two subsets of flow of the same commodity are enrouted through the same arc in consequent time intervals and the leading subset is further enrouted through the arc (b, d_1) at time 1. Again, the subset behind is forced to be assigned to the same arc at the next time interval (bold dashed arc). The combination of Constraints (10) and (11) ensures that once a first unit of flow starts its evacuation process, parallel and uninterrupted paths are assigned till the entire original flow has been routed. Constraints (12) impose the non-contemporaneity of flows associated to different commodities for each arc at the same time.

The remainder of the formulation simply impose standard flow constraints in a capacitated time-dependent network: constraints (13) impose the inflow arc capacity limitations; Constraints (14) and (16) ask for the conservation of flows for each commodity both at intermediate and source nodes, Constraints (15) ensure the path conservation at intermediate nodes. For each commodity at any time Constraints (17) impose the amount of flow to be null whenever the arc is not assigned.

In some networks sources might be transition points in the evacuation path of other different commodities. To account for this general situation, we simply have to extend for each commodity $k \in \mathcal{K}$ our Constraints (8), (9), (11),

(14), (15) to the case $i \in \mathcal{N} \cup \mathcal{S}$ with $i \neq s_k$. We still suppose this cannot happen for destination nodes as they represent safe areas.

3. Computational Results

In this section we present results obtained by optimizing evacuation processes both with the Multicommodity Quickest Flow Problem (*MCQFP*) and with the presented formulation of the Multicommodity Quickest Path Problem (*MCQPP*). The computational tests have been conducted on a benchmark set solved to optimality using ILOG CPLEX v. 12.6.0.0 solver in parallel opportunistic mode (up to 20 threads) on a 64bit Intel Xeon CPU at 2.80GHz with 32 GB memory, running Ubuntu 14.04.2, with a time limit of 3 hours for each test.

Our benchmark testbed is represented by a set of 20 instances based on two directed grid networks, the first with 50 nodes and 186 arcs and the second with 100 nodes and 292 arcs. Networks are populated with $k = 1, 2, 3, 4, 5$ commodities and two different levels of flow, namely *Level a* and *Level b*, with the second presenting a higher volume of flows. The time horizon considered for the evacuation is divided into 16 time intervals and the length of all arcs is fixed as equal to one time period.

The aim of this section is to provide a proof-of-concept for the presented model and to evaluate the quality of the provided solutions. Note that the value of the optimal solution of the *MCQFP* model represents a valid lower bound for the makespan of the *MCQPP* model on the same instance. Hence, we improve the resolution process of our *MCQPP* model by feeding the solver with the optimal solution of the *MCQFP* on the same instance as a lower bound.

In Table 1 we report for each instance the structure of the network: number of nodes $|\mathcal{V}|$, arcs $|\mathcal{A}|$, and commodities $|\mathcal{K}|$. Column *MCQFP Makespan* represents the optimal objective function obtained with the *MCQFP* and the *CPU Time* column the computation time, expressed in seconds, required to solve the instance. The remaining columns similarly refer to the *MCQPP* model. If the instance is solved to proven optimality within the time limit we provide the *GAP* as a percentage. The last column represents the ratio of the *MCQPP* makespan over the *MCQFP* makespan values. This way we compare the *MCQFP* makespan with the best optimal feasible solution obtained with our *MCQPP* model.

The computational experiments show how the *MCQFP* was able to solve all instances to optimality in a very low computational time, on average 0.5 seconds with both levels of population.

The introduction of the additional specific constraints related to the *MCQPP*, such as flow unsplittability, Non-contemporaneity, and Uninterrupted Scheduling, produced a moderately increase in the makespan of all the instances and a considerable increase in the computational time with a total of 17 out of 20 instances solved to proven optimality within the time limit. In terms of objective function values, the makespan of the *MCQPP* deteriorated by less than 50% on the average, with a mean ratio equal to 1.49 with the *Level a* of population and by less than 70% with a ratio equal to 1.68 with the *Level b*. With these values the *MCQPP* proves to be more sensitive than the *MCQFP* to changes in the level of population.

The overall experiments prove the validity of our model to correctly represent the original presented problem. Furthermore, one can argue that the *MCQPP* proposed in this paper provides solutions with a limited increase in the makespan offering a drastic reduction of the risk of interferences, congestions, bottlenecks and the production of chaotic situations with respect to the *MCQFP* solutions.

Additional computational tests were performed in order to evaluate the impact of Constraints (12) on the quality of the solutions. All of the considered instances present an equal value for the makespan (objective function) when Constraints (12) are relaxed, while the only remarkable difference is related to the computational effort, that is increased by the presence of Constraints (12).

4. Conclusion and outlook

In this paper we considered the Multicommodity Quickest Path Problem (*MCQPP*). Our research is motivated by real applications to emergency transportation processes that aim at providing a decision support system for the routing and scheduling of populations with their own origin, destination, and flow through single unique paths. We presented an original mixed integer linear programming formulation for the *MCQPP* with specific constraints introduced in

Table 1. Computational experiments with *Level a* and *Level b* of population.

Name	Instance			MCQFP		MCQPP			Ratio
	$ V $	$ A $	$ K $	Makespan (opt *)	CPU Time (sec.)	Makespan (opt *)	CPU Time (sec.)	GAP	
2-1-a	50	186	1	6*	0.06	8*	11.75	–	1.33
2-2-a	50	186	2	5*	0.15	8*	6.75	–	1.60
2-3-a	50	186	3	6*	0.25	9*	69.42	–	1.50
2-4-a	50	186	4	5*	0.30	8*	1058.89	–	1.60
2-5-a	50	186	5	6*	0.41	9*	1247.52	–	1.50
3-1-a	100	292	1	6*	0.19	9*	109.60	–	1.50
3-2-a	100	292	2	6*	0.36	10*	862.62	–	1.67
3-3-a	100	292	3	6*	0.55	7*	7472.38	–	1.17
3-4-a	100	292	4	6*	0.57	9*	3705.80	–	1.50
3-5-a	100	292	5	7*	2.13	–	10800.00	∞	–
2-1-b	50	186	1	6*	0.03	10*	5.18	–	1.67
2-2-b	50	186	2	6*	0.09	10*	15.30	–	1.67
2-3-b	50	186	3	6*	0.11	11*	54.53	–	1.83
2-4-b	50	186	4	6*	0.17	9*	990.41	–	1.50
2-5-b	50	186	5	6*	0.19	10*	3388.02	–	1.67
3-1-b	100	292	1	6*	0.10	11*	54.87	–	1.83
3-2-b	100	292	2	6*	0.21	12*	5169.38	–	2.00
3-3-b	100	292	3	6*	0.31	8*	1016.92	–	1.33
3-4-b	100	292	4	6*	0.39	16	10800.00	61.82%	–
3-5-b	100	292	5	8*	2.20	–	10800.00	∞	–

order to prevent the generation of interferences, turbulence, and congestions: Uninterrupted Scheduling of subsets of flows from the same commodity and Non-contemporaneity of different populations on the same arcs at the same time. We then performed a computational experience to provide a proof-of-concept of our proposed model and evaluate the quality of its solutions. The benchmark testbed collects connected grid networks with increasing number of commodities and values of populations. We compared the makespan for the Multicommodity Quickest Flow Problem (*MCQFP*) with the complete formulation of the Multicommodity Quickest Path Problem (*MCQPP*) introduced in this paper. The results of the computational experiments confirm the suitability and the quality of the solutions of the proposed model for the specific applications arising in the field of emergency transportation which motivated our work.

An important direction for future researches in this field consists in the improvement of the proposed formulation, by studying the relevant properties of the model, and the design of meta-heuristic approaches to compute valid upper bounds for the considered problem. Moreover, both micro-simulation and macro-simulation approaches (see for instance Barcelo (2005) (AIMSUN) and Caramia et al. (2010) (Fluidsim)) could be adopted to assess and compare the expected outcomes of the transportation plans produced by the *MCQFP* and the *MCQPP* optimization problems.

References

- Aronson, J.E., 1989. A survey of dynamic network flows. *Annals of Operations Research*, 20:166.
- Baier, G., Kohler, E., Skutella, M., 2005. On the k -splittable flow problem. *Algorithmica*, 42:231-248.
- Barcelo, J., Casas, J., 2005. *Dynamic Network Simulation with AIMSUN*. Simulation Approaches in Transportation Analysis Operations Research/Computer Science Interfaces Series Volume 31, pp. 57-98.
- Burkard, E. R., Dlaska, K., Klinz, B., 1993. The quickest flow problem, *ZOR Zeitschrift für Operations, Research Methods and Models of Operations Research*, Volume 37, Number 1, pp. 31-58.
- Caramia, M., Sgalambro, A., 2008. An exact approach for the maximum concurrent k -splittable flow problem. *Optimization Letters* 251-265.
- Caramia, M., Sgalambro, A., 2010. A fast heuristic algorithm for the maximum concurrent k -splittable flow problem. *Optimization Letters* 37-55.
- Caramia, M., D'Apice, C., Piccoli, B., Sgalambro, A., 2010. Fluidsim: A Car Traffic Simulation Prototype Based on FluidDynamic. *Algorithms*, 294-310.
- Chalmet, L., Francis, R., Saunders, P., 1982. Network Model for Building Evacuation. *Management Science*, Volume 28, pp. 86-105.
- Chen, Y.L., Chin, Y.H., 1990. The quickest path problem, *Computers and Operations Research* 17, 153-161.

- Fleischer, L., Skutella, M., 2002. The Quickest Multicommodity Flow Problem in Integer Programming and Combinatorial Optimization, Springer Berlin Heidelberg 36-53.
- Fleischer, L., Skutella, M., 2003. Quickest Flows Over Time in *SIAM Journal on Computing*.
- Ford, L. R., Fulkerson, D. R., 1958. Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419-433.
- Ford, L. R., Fulkerson, D. R., 1962. *Flows in Networks*. Princeton University Press, Princeton, NJ.
- Gamst, M., 2014. A local search heuristic for the Multi-Commodity k -splittable Maximum Flow Problem. *Optimization Letters*, 919-937.
- Hall, A., Langkau, K., Skutella, M., 2003. An FPTAS for Quickest Multicommodity Flows with Inflow-Dependent Transit Times. *Approximation, Randomization, and Combinatorial Optimization, Volume 2764 of Lecture Notes in Computer Science* 71-82.
- Hall, A., Hippler, S., Skutella, M., 2003. Multicommodity flows over time: Efficient algorithms and complexity. *Theoretical Computer Science*. To appear. An extended abstract appeared in *Proceedings of ICALP 2003*.
- Hamacher, H., Tjandra, S., 2002. Mathematical Modelling of Evacuation Problems A State of the Art. In: "*Pedestrian and Evacuation Dynamics*", Springer, Heidelberg, pp. 227-266.
- Hamacher, H., Leiner, K., Ruzika, S., 2011. Quickest Cluster Flow Problems. In "*Proceedings of PED 2010 Conference*".
- Hoppe, B., 1995. Efficient dynamic network flow algorithms. PhD thesis, Cornell University.
- Hoppe, B., Tardos, E., 2000. Polynomial time algorithms for some evacuation problems. *Operations Research*, 25:36-62.
- Kleinberg, J.M., 1996. Single-Source Unsplittable Flow. *Annual Symposium on Foundations of Computer Science*. 37, 68-77.
- Klinz, B., Woeginger, G.J., 2004. Minimum-cost dynamic flows: The series-parallel case. *Networks*, 43:153-162.
- Koehler, E., Moehring, R. H., Skutella, M., 2009. Traffic networks and flows over time. In "*J. Lerner, D. Wagner, and K. A. Zweig, editors*". *Algorithmics of Large and Complex Networks, LNCS 5515*, Springer, pp. 166-196.
- Leiner, K., Ruzika, S., 2011. Quickest Cluster Flow Problems on Tree Networks. In *INOC(2011)* 243-255.
- Lin, M., Jaillet, P., 2015. On the Quickest Flow Problem in Dynamic Networks: A Parametric Min-cost Flow Approach. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 1343–1356.
- Mawson, A., 2005. Understanding Mass Panic and Other Collective Responses to Threat and Disaster. *Psychiatry: Interpersonal and Biological Processes* 68(2), 95-113.
- Pascoal, M. M. B., Captivo, M. E. V., Climaco, J. C. N., 2006. A comprehensive survey on the quickest path problem, *Annals of Operations Research* 147, pp. 5-21.
- Powell, W. B., Jaillet, P., Odoni, A., 1995. Stochastic and dynamic networks and routing. In "*M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors*", *Network Routing, Volume 8 of Handbooks in Operations Research and Management Science*, pp. 141-295. North-Holland, Amsterdam, The Netherlands.