A mezoscopic vehicles pursuit model for managing traffic during a massive evacuation

Houssein Alaeddine[a,b], Mindjid Maizia[a], Kamal Serrhini[a], Emmanuel Néron[b]

[a] *Laboratoire Citères de l'université de Tours*
[b] *Laboratoire Informatique de l'université de Tours*
Tours, France

Abstract

Network flows over time comprising many variants form a wide area in evacuation domain. However the macroscopic type that characterizes these models forms their main drawback comparing to traffic assignment models that treat traffic situation in a more realistic way where selected evacuation paths as well as flow-dependent transit times on links are taken into consideration. Evacuation paths that may not be respected in dynamic flow models was studied in a previous paper. In this paper we are interested on the development of a mezoscopic vehicles pursuit model (VPM) belonging to traffic assignment models which is capable to capture the flow of vehicles on the network combining both discrete and continuous time aspects. This model is based on a discrete scheduling carried out by a lexicographic maximum flow algorithm according to an evacuation priorities list. VPM then transforms the discrete process to a continuous one avoiding overlap between successive time intervals while minimizing the departure times of vehicles from sources.

Keywords:
traffic assignment model, flow routing problem, discrete and continuous time, time minimization, evacuation

1. Introduction

There are nearly half a century that Ford-Fulkerson [16] addressed the construction of dynamic flow in network from static one [15], with constant transit time on arcs and fixed discrete time horizon. Dynamic flow, also called flows over time, is traditionally solved by a time-expanded network, introduced by Ford and Fulkerson (1958), that contains a copy of the static network for each discrete time step $t \in \{0,1,\ldots,\mathrm{T}-1\}$, where T is a defined evacuation time horizon i.e. no flow can leave its source if its last particle does not reach its destination at the latest at T.

Given a static network $\mathrm{G} = (\mathrm{N},\mathrm{A},\tau,c)$ defined by a set of nodes N, a set of arcs A and a constant transit time τ and capacity c of arcs, a time-expanded network $\mathrm{G}_{\mathrm{T}-1}$ is built so that $\mathrm{N}_{\mathrm{T}-1} = \left\{j(t) : j \in \mathrm{N}, t \in \{0,1,\ldots,\mathrm{T}-1\}\right\}$ and $\mathrm{A}_{\mathrm{T}-1} = \left\{(i(t), j(t+\tau_{i,j})), t \in \{0,\ldots,\mathrm{T}-1-\tau_{i,j}\}\right\}$. The problem of dynamic maximum flow during T is a classical problem of maximum flow from source $s(0)$ to destination $p(\mathrm{T}-1)$ [14] with a pseudo-polynomial complexity i.e. the execution time of algorithms based on time-expanded network is a polynomial function of the number of time steps T [23].

The independent-flow transit time on arcs in this network is converted in number of time steps. Therefore to achieve an accuracy in model a too high number of T is needed. However it results in a huge graph with fractional capacities of arcs (for instance, $c_{i,j} = 0.38888$veh/sec). Moreover since the number of arcs in time-expanded network depends on the travel time, this type of network is no longer valid if this latter depends on flow. To avoid the huge size of a time-expanded graph, Ford and Fulkerson have introduced the flow temporally repeated that can be solved by one minimum cost flow computation in a static network. The construction of this flow is based on a decomposition paths algorithm [10] enabling to send an amount of flow through each path p_i at every $t \in \{0,1,\ldots,\mathrm{T}-1-\tau(p^i)\}$, where $\tau(p^i)$ corresponds to the total travel time on the path p^i.

Despite that flow temporally repeated solves the problem concerning the size of network, therefore it meets the problem of multi-sources that cannot be reduced to a single source. Indeed the flow temporally repeated algorithm determines for every source an amount of flow to be routed in the network at each time step. This amount may equal zero for some origins, this is first, and second, an origin with a low quantity of vehicles does not need all the horizon time to remove them (network under-utilization).

Many authors were addressed the dynamic flow problem enriching the literature by developing different efficient approaches (dynamic maximum flow [14, 18, 17], universal dynamic maximum flow [12, 15, 19, 23, 27, 28, 36], quickest dynamic flows [7, 8, 13, 23], lexicographic dynamic maximum flow [21, 22, 23, 27]). Those approaches, in addition to their difference in problem treated and resolution method, differ in terms of transit time considered as flow-dependent or independent, time-dependent or independent capacity, etc.

Kohler et al. [25] distinguished between two flow-dependent transit times aspects, the first one is adopted by many models in the literature based on relatively simple assumptions i.e. traversal time on a road link is treated as a function depending only on the flow rate at time of entry to the arc (for example, see [9]). While in contrast, a fully realistic model of flow-dependent transit times must take density, velocity and flow rate on roads into consideration. To be as realistic as possible, a flow-dependent time model must

Email addresses: houssein.alaeddine@hotmail.fr (Houssein Alaeddine), mindjid.maizia@univ-tours.fr (Mindjid Maizia), kamal.serrhini@univ-tours.fr (Kamal Serrhini), emmanuel.neron@univ-tours.fr (Emmanuel Néron)

consider the density, speed and flow on the roads [20]. They have proposed a formulation based on a model developed by *U.S. Bureau of Public Roads*, describing the time depending on flow. Based on flow, free-flow travel time and capacity of road, this formulation computes transit time on road. This model is of limited use because first, the definition of flow rate on the arc at a specific moment in time is not explicit, and second, it's not reasonable to assume that the transit time on a road link depends only on the current flow rate. According to the authors, even if the flow rate which enters a road is low compared to the maximal flow rate on this latter, the transit time may nevertheless be huge due to traffic jams caused by a large number of vehicles entered previously. That model is, of course, not able to capture all properties of traffic since no delay on any road caused by congestion is considered.

Carey and Subrahmanian [9] introduced a generalized time-expanded network to meet the issue of flow-dependent transit time. At each time step, several copies of each arc corresponding to different transit times are created. However this approach cannot be solved by standard network flow techniques but by a general linear programming solver due to the necessity of special capacity constraints in order to respect flow-dependent transit time. This is a serious drawback because of the inefficiency and the difficulty of applying such a model.

Kohler et al. [26] proposed a closely related model which overcomes the drawback of Carey and Subrahmanian's model introducing additional 'regulating' arcs which enable to observe the flow-dependent transit time without resorting to generalized capacity constraints.

Although this approach does not fully take into consideration the behavior of flow over time, it has a satisfactory result. This incompleteness is a consequence of discrete time function used on arcs which takes only a few values. Thus, besides the disadvantage of size of time-expanded graph, the number of arcs is much higher in the case of a generalized time-expanded network. The complexity of this network is a polynomial function of the number of time steps T multiplied by the number of values that the time function can take.

The nature of time between, discrete and continuous, forms beside the flow-dependent transit time a second important element in flow modeling. Define here the function $f_a(t)$ for measuring the flow rate (quantity of flow per time unit) entering the arc a at time t. In discrete time, $f_a(t)$ is defined over a discrete time set : $\{0,1,\ldots,\mathrm{T}-1\} \rightarrow \mathbb{R}^+$, while in continuous time it is defined on a continuous interval : $[0,\mathrm{T}-1] \rightarrow \mathbb{R}^+$. The figure 1, slightly modified from that proposed by [33], explains the cause of adopting a discrete time interval $\{0,1,\ldots,\mathrm{T}-1\}$ despite a horizon equal to T.

This example consists of a single arc a of capacity $u_a = 2$ veh/sec, a travel time $\tau_a = 3$ sec and a time horizon $\mathrm{T} = 5$. This figure shows the circulation of dynamic flow on the arc a. At $t = 0$, the arc a is empty and we inject a flow equal to its capacity. At $t = 1$, the first flow circulates on the arc and we send a second flow of two units, then we stop.

We stop at $t = 1$ and not at $t = 2$, in order to avoid that the third flow arrives at its destination too late i.e. after T. Therefore, the last time step T is the time required for network for absorbing flow, since a flow is composed of several particles. In other words, the time period between two successive injections of vehicles in the network is equal to one time unit.

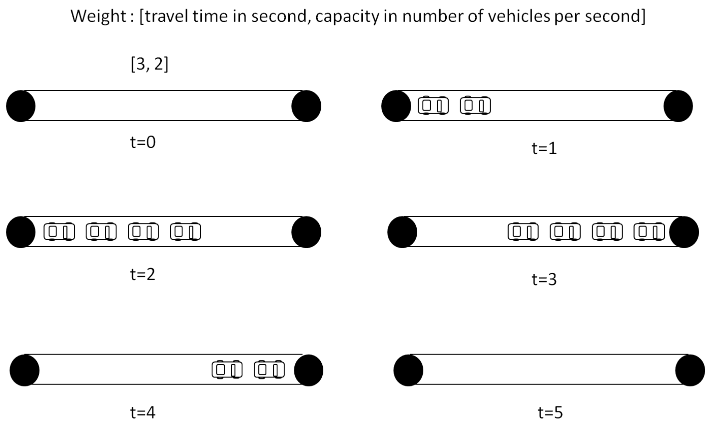


FIGURE 1: Circulation of dynamic flow through a road link

Fleischer and Tardos [14] noted a strong interaction between discrete and continuous models and introduced a natural transformation of discrete time in continuous one. Let $f_a(t)$ a discrete flow rate entering arc a at time $t \in \{1,\ldots,\mathrm{T}-2\}$ and $f'_a(t')$ the continuous flow rate corresponding to $f_a(t)$ for $t' \in [t,t+1)$. Assuming constant network settings (capacities of arcs, travel times), we get an identical continuous flow rate with this transformation for both functions f et f' and for any interval $[t,t+k), t \in \{1,\ldots,\mathrm{T}-2\}, k \in \mathbb{N}$. For instance, the capacity (or maximal flow rate) expressed in number of vehicles per one half hour is equal to 30 times the capacity expressed in number of vehicles per minute.

Philpott [29] and Anderson et al. [5] formulated and solved dynamic flow problems in continuous time with zero transit times and time-varying arc capacities and storage at nodes. That research was later extended by Philpott [30] to arbitrary transit times.

Koch et al. [24] devopped a model which combines both discrete and continuous models based on measure theory. They studied the maximum flow over time in a directed graph where the flow on arcs is a Borel measure in $\mathbb{R}$. The aspect of storage of vehicles and the wait of these latter in nodes are considered with a time-dependent cost. However transit time on arcs is fixed in their model and does not depend on flow. Philpott [31] also studied this problem with fixed transit times in the space of finite Borel measures introducing a dual program.

Sherali and Carter [32] formulated a non-linear mixed integer program for evacuation planning under hurricane/flood conditions. The approach developed aims to minimize the total traffic-jams during an evacuation process. Travel time on roads is computed using the same traffic model in [25]. Daganzo and so [11] developed a socially « fair » as well as an optimal model for managing traffic network in real-time during emergency evacuations. That model combines between the two principles of Wardrop [35] ; the user equilibrium and the system optimum.

We address in this paper the problem of evacuation of buildings located in zone exposed to a natural disaster and more particularly to a risk of flood. The transportation network

not allowing a simultaneously evacuation of all buildings, a scheduling evacuation system based on a priorities list established is developed in order to organize the evacuation. The strategy applied in this system is to evacuate at each time slot (15 minutes, 30 minutes, etc.) the lexicographic maximal flow from each building not yet completely evacuated. This system was studied in a previous papers (see [1][4])under some constraints in order to meet several objectives. Since the sources from which flow incoming may change over time, an overlap between successive time slots can be occurred and therefore the period time between two successive injections of vehicles in the network is greater than or equal to one time slot. This required to develop a mezoscopic vehicles pursuit model (VPM) to organize traffic in network tracing the trajectory of vehicles incoming from different buildings at each time slot. VPM, which is based on a polynomial traffic model enabling the computation of flow-dependent transit time, aims to minimize the evacuation departure times of every building while avoiding any congestion in the network. This model combines both discrete and continuous time aspects. Our evacuation model STOM[1] computes the minimum possible total evacuation time.

2. Problem formulation

Two main objectives are pursued in the VPM : (i) minimizing evacuation departure dates of buildings, and (ii) ensuring continuity of flow in the evacuation network without congestion.

2.1. Graph Construction

Let us consider $\mathbf{B} = \{\mathscr{B}_1,\ldots,\mathscr{B}_n\}$ a priorities list of n buildings to be evacuated towards $\mathbf{S} = \{\mathscr{S}_1,\ldots,\mathscr{S}_m\}$ a set of m shelters. Each shelter $\mathscr{S}_j \in \mathbf{S}$ is defined by its capacity of vehicles c_j and each building $\mathscr{B}_i \in \mathbf{B}$ with a number of vehicles v_i is connected to one or more shelters $\mathscr{H}_i$. A set of minimum paths $\mathscr{K}_{i,j}$ between each building $\mathscr{B}_i$ and each associated shelter $\mathscr{S}_j \in \mathscr{H}_i$ is computed. We denote by $v_{i,j}$ and $\eta_{i,j}$ successively, the number of vehicles to be evacuated and the dimension of minimum paths set from $\mathscr{B}_i$ to $\mathscr{S}_j$. We represent the k-th-path connecting $\mathscr{B}_i$ to $\mathscr{S}_j$ by the vector of arcs $Arc^k_{i,j}$.

Let $\mathscr{G} = (\mathscr{N},\mathscr{A})$ be the graph representing the evacuation network established. We define $\hat{\phi}_l, l \in \mathrm{A}$, the maximal flow rate (capacity) of arc l expressed in number of vehicles per unit of time (veh/h, etc.). This capacity is provided by a polynomial traffic model applied in STOM such as congestion on roads is not allowed (see section 2.3).

We extend the graph $\mathscr{G}$ to $\mathscr{G}^* = (\mathscr{N}^*,\mathscr{A}^*)$ by adding 1) a super node s and a super destination p, 2) a sub-set of virtual nodes $\mathbf{M}$ of dimension $\sum_{i=1}^{n} |\mathscr{H}_i|$ including one node $\mathscr{M}_{j,i}$ by each safety point $\mathscr{S}_j$ associated to each building $\mathscr{B}_i$, 3) a sub-set of virtual arcs between s and $\mathbf{M}$ which represent the number of vehicles to be evacuated from buildings to safety points, 4) a sub-set of arcs of dimension $\eta_{i,j}$ between each node $\mathscr{M}_{j,i} \in \mathbf{M}$ and $\mathscr{B}_i$ which represent the number of vehicles to be evacuated through the paths $Arc^k_{i,j}$, and finally 5) a sub-set of virtual arcs among shelters and p which represent the capacities of shelters in terms of number of vehicles.

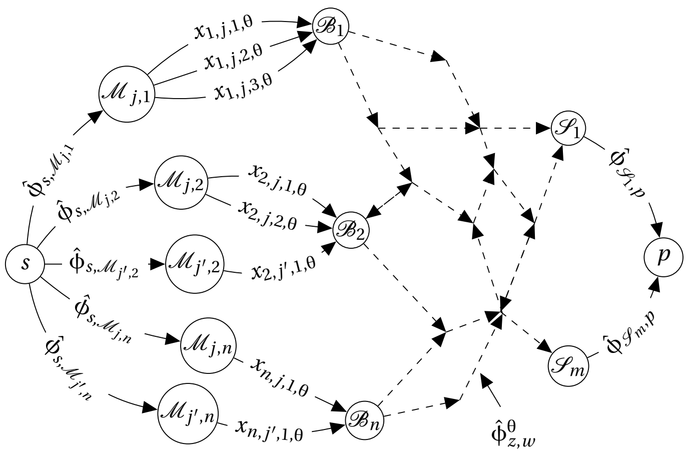


FIGURE 2: Buildings to be evacuated

2.2. Notation

- i : An index for building, $\mathscr{B}_i$
- j : An index for shelter, $\mathscr{S}_j$
- k : An index for k-th best path, $Arc^k_{i,j}$
- θ : time slot
- l : Road link (arc)
- $\hat{\phi}_l$: Capacity of road link l
- ϕ^θ_l : Flow on arc l at time slot θ
- $\mathscr{M}_{j,i} : \mathscr{B}_i \rightarrow \mathscr{S}_j$

The vehicles pursuit model developed here is based, as we mentioned previously, on a evacuation scheduling model providing (i) the number of time slot Θ for evacuating all buildings (ii) the number of vehicles evacuated through each path from each building at each time slot $\theta \in \{1,\ldots,\Theta\}$, and (iii) the network load (assigned flow on arcs) at each slot.

2.2.1. Input data from the evacuation scheduling system

- $x_{i,j,k,\theta}$: number of vehicles evacuated from building $\mathscr{B}_i$ to safety point $\mathscr{S}_j$ using the k-th path connecting $\mathscr{B}_i$ and $\mathscr{S}_j$ at time slot θ.
- ϕ^θ_l : number of vehicles crossing road link l at time slot θ

2.2.2. Decision variables

- $t_{i,j,k,\theta}$: departure time of vehicles evacuated from building $\mathscr{B}_i$ to safety point $\mathscr{S}_j$ using the k-th path connecting $\mathscr{B}_i$ to $\mathscr{S}_j$ at time slot θ
- $\tau_{i,j,k,\theta}$: travel time between $\mathscr{B}_i$ and $\mathscr{S}_j$ through the k-th minimal path at slot θ
- $\lambda_{i,j,k,\theta}$: flow rate of vehicles outgoing from building $\mathscr{B}_i$ and evacuated to safety point $\mathscr{S}_j$ using the k-th path connecting $\mathscr{B}_i$ to $\mathscr{S}_j$ at slot θ
- t_θ : start date of time slot θ
- δ_θ : duration of time slot θ
- Γ_θ : end date of slot θ
- τ^θ_l : travel time on arc l at time slot θ
- $\hat{\epsilon}^\theta_l$: entry time of last vehicle crossing l at slot θ

1. A spatio-Temporel Optimization Model for the evacuation of population exposed to natural disasters. Developed in ACCELL project funded by Région Centre and FEDER, France

- ϵ_l^θ : release time of arc l at time slot θ

Auxiliary variables :

- Δ_l^ϕ : journey time through a road link l, given an arrival flow ϕ.
- $e_{i,\theta} = \begin{cases} 1 & \text{if } \mathscr{B}_i \text{ is effectively evacuated, completely} \\ & \text{or partially, at time slot } \theta \\ 0 & \text{otherwise} \end{cases}$

Parameters :

- $\beta_{i,j} = \begin{cases} 1 & \text{if } \mathscr{B}_i \text{ is connected to } \mathscr{S}_j \\ 0 & \text{otherwise} \end{cases}$
- $\alpha_{i,j,k,l} = \begin{cases} 1 & \text{if } l \in Arc_{i,j}^k \\ 0 & \text{otherwise} \end{cases}$
- $\lambda_{i,j,k}$: constant flow rate of vehicles outgoing from $\mathscr{B}_i$ and evacuated to $\mathscr{B}_j$ using the k-th path.

2.3. *Traffic model*

This section focuses on the presentation of a traffic polynomial model which allows to determine the maximal flow rate on each arc depending on free-flow speed and jam density. In addition this model computes speed on roads depending on density which in turn is depending on flow of vehicles.

By hypothesis, the network is considered here loaded by a variable number of vehicles depending on the start date of the evacuation.

Transport network as it is known has a capacity limited to a number of vehicles per unit of time. In other words there is no buffer or waiting space, and therefore as soon as a vehicle enters the road link, it starts obtaining the service [34]. Any overload on any road leads to a congestion on all or part of network. Our model prevents the traffic-jams on network respecting maximal flow rate on each road. The calculation of flow rate q is crucial, it depends on two elements, firstly the density k expressed in number of vehicles per unit of distance, and secondly, the speed v in unit of distance per unit of time.

$$q = k \times v$$

The traffic model used here is the polynomial traffic model of order 3. The relationship between velocity and density is a second degree polynomial equation [6] :

$$v = v_f + bk + ck^2$$

where : v_f is the free-flow speed, b and c are constants.

Jam-density on a road of network road can be computed in a *go & stop* condition, when by approximation, the inter-vehicular distance closes the average length of a vehicle (see figure 3).

According to *L'Argus*[2], the average french car length in 2011 is 4.19 m. Given a road of length L = 1 km and a car of length $x = 4.19\ m$, the jam-density $k_j = 119.8\ veh/km/lane$ is determined by the next formula :

$$k_j x + (k_j - 1)x = \mathrm{L}$$

2. www.largus.fr

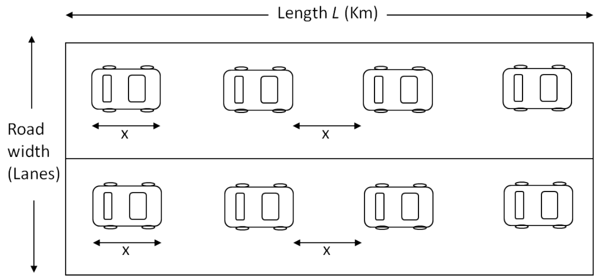


FIGURE 3: Representation of a road link in case of congestion

The speed $vs.$ density relationship may be expressed as a quadratic equation of the form [6] :

$$v = v_f(1 - \frac{k^2}{k_j^2})$$

The maximal flow rate q_m is obtained when $\frac{dq}{dk} = 0$.

The two following figures 4 and 5 show successively the relationship between firstly speed and density, and secondly, density and flow rate.

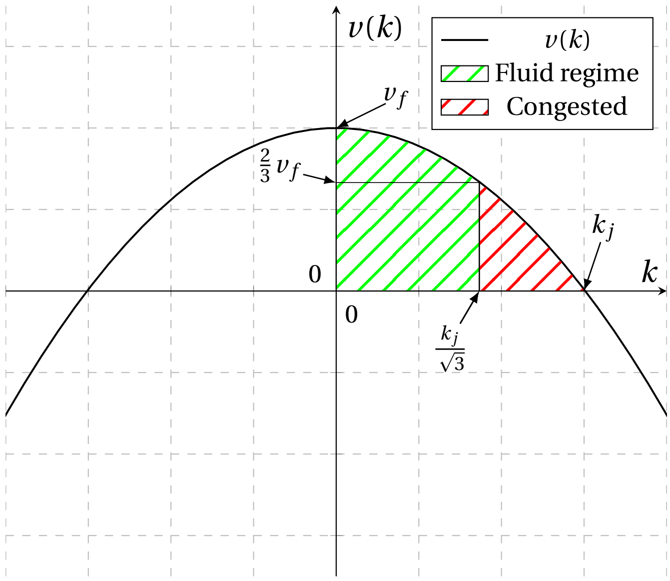


FIGURE 4: Speed $vs.$ density relationship

3. Time routing optimization

To achieve the stated objectives in the vehicles pursuit model, two methods behaving differently, from organizational point of view, are developed here.

Let's define for each time slot θ :

- $\mathbf{B}^\theta = \{\mathscr{B}_i / e_{i,\theta} > 0\}$ the set of buildings evacuated completely or partially. This set is a result from the evacuation scheduling model.
- $\boldsymbol{\tau}^\theta = \{\tau_{i,j,k,\theta}/x_{i,j,k,\theta} > 0, \forall \mathscr{B}_i \in \mathbf{B}^\theta, \forall \mathscr{S}_j \in \mathscr{H}_i, \forall k \in \{1,\cdots,\eta_{i,j}\}\}$ the set of journey times on routes connecting buildings to shelters
- $\boldsymbol{\lambda}^\theta = \{\lambda_{i,j,k,\theta}/x_{i,j,k,\theta} > 0, \forall \mathscr{B}_i \in \mathbf{B}^\theta, \forall \mathscr{S}_j \in \mathscr{H}_i, \forall k \in \{1,\cdots,\eta_{i,j}\}\}$ the set of vehicles flow rates on paths of buildings

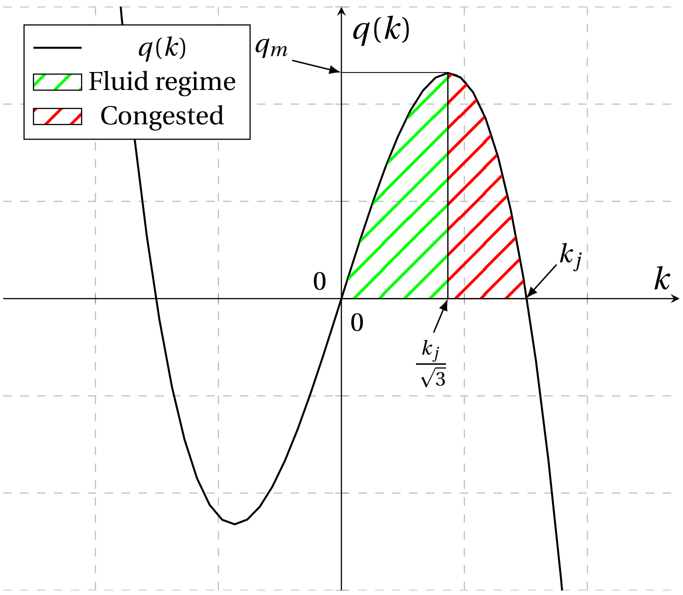


FIGURE 5: Flow rate *vs.* density relationship

- $\mathbf{t}^\theta = \{t_{i,j,k,\theta} / x_{i,j,k,\theta} > 0, \forall \mathscr{B}_i \in \mathbf{B}^\theta, \forall \mathscr{S}_j \in \mathscr{H}_i, \forall k \in \{1, \cdots, \eta_{i,j}\}\}$ the set of departure dates of evacuated vehicles through the paths of buildings.

$\mathbf{B}^\theta$ is the input data of VPM ; $\boldsymbol{\tau}^\theta$, $\boldsymbol{\lambda}^\theta$ and $\mathbf{t}^\theta$ constitute the output of this model.

The set of start times of vehicles which use the k-th path connecting $\mathscr{B}_i$ and $\mathscr{S}_j$ at slot θ is denoted by $[t_{i,j,k,\theta}, t_{i,j,k,\theta} + \lambda_{i,j,k,\theta} \cdot (x_{i,j,k,\theta} - 1)\ ]$ where the first vehicle on this path leaves at $t_{i,j,k,\theta}$ and the last one at $t_{i,j,k,\theta} + \lambda_{i,j,k,\theta} \cdot (x_{i,j,k,\theta} - 1)$. The waiting time between two start times is determined by $\lambda_{i,j,k,\theta}$. The set of arrival times of vehicles using the k-th path connecting $\mathscr{B}_i$ and $\mathscr{S}_j$ at slot θ is given by $[t_{i,j,k,\theta} + \tau_{i,j,k,\theta}, t_{i,j,k,\theta} + \lambda_{i,j,k,\theta} \cdot (x_{i,j,k,\theta} - 1) + \tau_{i,j,k,\theta}]$.

We denote by $\mathscr{I}_\theta = [t_\theta, \Gamma_\theta]$ the time interval corresponding to slot θ where :

$$t_\theta = \mathrm{M}in\{t_{i,j,k,\theta}\ ,\ \forall \mathscr{B}_i \in \mathbf{B}^\theta, \forall \mathscr{S}_j \in \mathscr{H}_i, \forall 1 \leq k \leq \eta_{i,j}\}$$

$$\Gamma_\theta = \mathrm{M}ax\{t_{i,j,k,\theta} + \lambda_{i,j,k,\theta} \cdot (x_{i,j,k,\theta} - 1) + \tau_{i,j,k,\theta}\ ,\ \forall \mathscr{B}_i \in \mathbf{B}^\theta\}$$

Travel time τ_l^θ on a road link l at slot θ is computed by the traffic model (see section 2.3) depending on flow rate of vehicles crossing this road at this slot.

Recall that the capacity of each arc at each slot was observed by the evacuation scheduling model. This constraint ensures a traffic circulation in the best possible conditions (without congestion) during the time interval $[\tau_\theta, \Gamma_\theta]$.

3.1. *Empty evacuation network*

Since the origins of vehicles are not always the same during successive time intervals, the capacity constraint may be violated by overlap between two successive time intervals.

To overcome this constraint, a new time interval will be allocated when the previous one is completely finished. This constraint is expressed as follows :

$$\forall \theta \in \{1, \ldots, \Theta - 1\}, \mathscr{I}_\theta \cap \mathscr{I}_{\theta+1} = \emptyset \Leftarrow \tau_{\theta+1} = \Gamma_\theta \qquad (1)$$

The total evacuation time T_e according to this scheduling is given by :

$$\mathrm{T}_e = \Gamma_\Theta - \tau_1$$

3.2. *Vehicles pursuit model : availability of network roads*

In the evacuation model STOM, minimizing the total clearance time depends on two elements : (i) minimizing the number of slots by maximizing flow in the network at each time interval under some constraints (scheduling evacuation model[3]) , and (ii) minimizing the start times of vehicles from buildings at each time slot, this property also guarantees a continuity of flow in network.

The first solution represented by equation 1 (see above), responsible of avoiding overlap between successive time intervals, is not optimal especially when safe points are away from buildings to be evacuated (evacuation network underutilized).

To exploit the network, a vehicles pursuit model is developed. It allows to trace the flow of vehicles along the transport network in order to determine the availability of roads during each time interval. This solution ensures the continuity of flows from buildings while avoiding overlap between successive time intervals.

VPM is based on the concept of availability of road links : A road link (arc) l is considered available at a given time if and only if the release time ϵ_l of this road is less than or equal to this time.

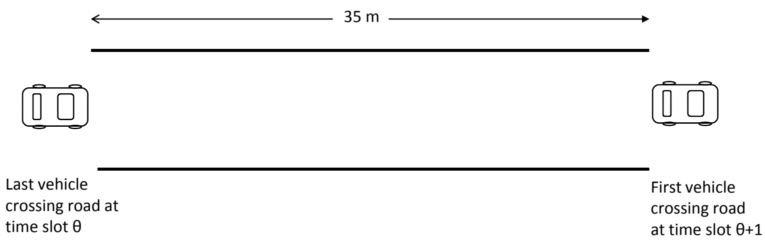


FIGURE 6: Availability of a road

According to this definition, the road link l cannot be used before the complete crossing of the last vehicle. Therefore the more the distance of road is big, the more the exploitation of this latter is low (see figure 6).

The maximal flow rate on each arc is formed with a density $k = \frac{k_j}{\sqrt{3}}$ (k_j : jam-density), which means that the inter-vehicular distance is equal to 2.49 times the average length of a vehicle (see figure 7).

This leads to divide each arc l into a number of identical portions of length greater than or equal to 2.49 times the average car length (x). The lower and upper bounds of length of a portion are given by [2.49x, 2×2.49x[. Note that travel time on each portion l_p of road link l is obviously the same (see figures 6 and 8).

3. This part of STOM was addressed in a previous papers

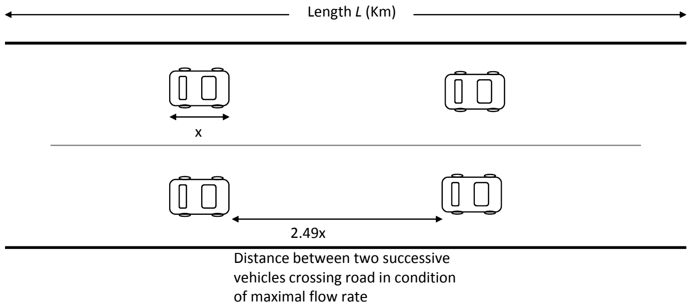


FIGURE 7: Distance between two successive vehicles in case of maximal flow rate

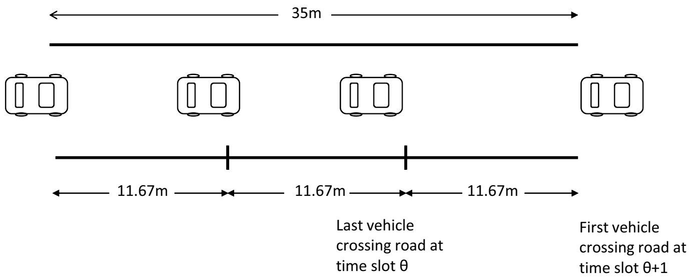


FIGURE 8: Maximum continuous flow of vehicles on the network

The objective of the VPM, which corresponds to minimizing the departure dates of vehicles at each time interval, is carried out for each route ensuring both a maximal flow rate of vehicles on roads and a continuous flow between successive time intervals. This condition of maximal flow rate is ensured between the last vehicle crossing road at previous time interval and the first vehicle in the current time interval : it guarantees the absence of overlap (see figure 8). The continuity of flow on roads during each time interval is ensured by the traffic model (see section 2.3).

By definition, an arc is considered shared between two paths at time slot θ, if and only if, it receives flow from both these two paths :

$Arc^{k}_{i,j} \cap Arc^{k'}_{i',j'} \neq \emptyset$ during $\mathscr{I}_\theta$ if and only if it $\exists\ l \in Arc^{k}_{i,j}$ such as $\alpha_{i',j',k',l} \cdot x_{i',j',k',\theta} > 0$

The importance of this definition is described later.

Theorem 1 *The flow rate of vehicles on a path, in a given time interval, depends on the maximal flow rate on minimum arc (in terms of maximal flow rate) of this path, if this latter does not share any arc with another during this time interval.*

The next paragraph details this theorem.

The second objective of VPM is to organize the evacuation on the network and especially on shared arcs between several paths, by a specific passage time allocation. This requires an harmonization of flow in network and in particular on the shared arcs. This is discussed below.

3.3. Harmonization of flow between parallel and sequential methods for intervals of long periods

The organization of flow on arcs shared by several paths (muli-source) differs from that on arcs that belong to a single path. To harmonize the flow on the road network, two approachs have been developed. A so-called parallel approach to inject in network, in a parallel manner, vehicles from multiple sources with a flow rate more or less low. And a sequential approach simulating a sequential circulation of vehicles on the network with a higher flow rate.

3.4. Parallel Harmonization of flow

The first method in the parallel approach is to harmonize flow on all arcs per unit of time (hour, half hour, etc..). For a unit of time equal to one hour, vehicles must be removed with a flow rate given by the equation :

$$\lambda_{i,j,k,\theta} = \frac{3600}{x_{i,j,k,\theta}} \quad with \ \ \lambda_{i,j,k,\theta} \in \boldsymbol{\lambda}^{\theta} \tag{2}$$

Due to the formulation of $\lambda_{i,j,k,\theta}$ which depends on time unit on arcs, a waste of time at each interval may then be met. We consider in the example below a building $\mathscr{B}_i$ with 700 vehicles to be evacuated towards $\mathscr{S}_j$ through the path $Arc^{k}_{i,j}$ at slot θ (figure 9) :

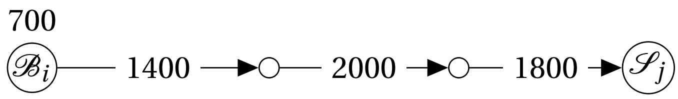


FIGURE 9: Flow rate of vehicles on a road

As the capacity unit of arcs is expressed in $veh/hour$, so the vehicles from $\mathscr{B}_i$ will need an hour to leave the building, which leads to $\lambda_{i,j,k,\theta} = \frac{3600}{700} = 5.14$. Since density of vehicles on road is low, speed becomes higher reducing the travel time (see section 2.3). According to this method (equation 2), the total evacuation time of vehicles from $\mathscr{B}_i$ is equal to one hour increased travel time of the last vehicle from $\mathscr{B}_i$ to $\mathscr{S}_j$ through $Arc^{k}_{i,j}$.

Looking closely, we note that it is possible to evacuate vehicles of $\mathscr{B}_i$ in a shorter time by injecting vehicles with a constant flow rate (independent of assigned flow at each time slot θ) equal to the maximal flow rate of minimum arc (in terms of maximal flow rate).

$$\lambda_{i,j,k} = \frac{3600}{Min_{\hat{\phi}} Arc^{k}_{i,j}} \tag{3}$$

The total discharge time corresponds to a half hour plus the travel time of the last vehicle. In this case (equation 3), the travel time on $Arc^{k}_{i,j}$ is greater than the previous one (case of equation 2) since the density on road is much higher.

Therefore the loss of time in the evacuation process resulting from the use of equation 2 can be solved by applying the equation 3. However this method is no longer valid if the evacuation paths share arcs and especially minimum arcs. The figure 10 shows an example of two buildings $\mathscr{B}_1$ and $\mathscr{B}_2$ with 700 vehicles each. The two paths of these two buildings share a minimum road.

According to the maximal flow rates of minimum arcs of paths $Arc^{k}_{1,j}$ and $Arc^{k}_{2,j}$, both buildings will evacuate their vehicles, at the current time slot θ, with a flow rate $\lambda_{1,j,k} = \lambda_{2,j,k} = 2.57$ and therefore all vehicles from $\mathscr{B}_1$ and $\mathscr{B}_2$ will be evacuated within an half hour.

However as the capacity of common arc allows to receive one vehicle per 2.57 seconds, a congestion will be occurred

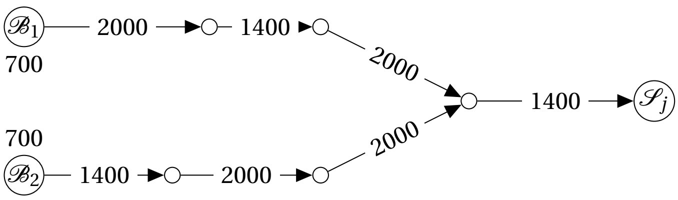


FIGURE 10: Flow rates on paths sharing arcs

theoretically on this arc in a half hour and reaches its maximum after one hour of the beginning of the evacuation. Hence this method cannot be used to generate an evacuation plan (without congestion).

3.4.1. Evacuation network decomposition algorithm

To solve the loss of time present in the equation 2, a modification of this equation ($\lambda_{i,j,k,\theta} = \frac{3600}{x_{i,j,k,\theta}}$) has been performed based on the decomposition of the evacuation network $\mathscr{A}$, at each time slot, into several sub-networks $\{\mathscr{A}_1,\ldots,\mathscr{A}_z\}$. Each sub-network $\mathscr{A}_x$ has its own period (bounded by]0,3600]) enabling vehicles to leave their buildings. The construction of the set of sub-networks Π^θ at each time slot θ is given by the following.

Definitions :

1. $\mathscr{A}_x$: sub-network.
2. $\mathscr{A}_x \cap \mathscr{A}_y \neq \emptyset$: if it exists at least a common arc between the two sub-networks $\mathscr{A}_x$ and $\mathscr{A}_y$.

We denote by Π^θ={$\mathscr{A}_1,\ldots,\mathscr{A}_z$} the z sub-network(s) built by the algorithm at time slot θ, $z \geq 1$..

The progress of this algorithm is given by the following :

Algorithm 1 Decomposition of evacuation network

procedure CONSTRUCTION OF SUB-NETWORKS
  $\mathscr{Q} = \cup_{i=1}^{n} \cup_{\mathscr{R}_j \in \mathscr{H}_i} \cup_{k=1}^{\eta_{i,j}} Arc_{i,j}^k$ ▷ Set of remaining paths
  $\Pi^\theta = \{\}$ ▷ Initialization
  while $\mathscr{Q} \neq \emptyset$ **do**
    Take a path $Arc_{i,j}^k \in \mathscr{Q}$
    if $x_{i,j,k,\theta} > 0$ **then**
      Build the sub-network $\mathscr{A}_{z+1}=\left\{Arc_{i,j}^k\right\}$
      Find the sub-set $E_{\Pi^\theta} \subseteq \Pi^\theta$ which satisfies the following :
        • $\mathscr{A}_{z+1} \cap \mathscr{A}_x \neq \emptyset,\ \forall\ \mathscr{A}_x \in E_{\Pi^\theta}$
        • $\mathscr{A}_{z+1} \cap \mathscr{A}_x = \emptyset,\ \forall\ \mathscr{A}_x \in \Pi^\theta \setminus E_{\Pi^\theta}$
      $\mathscr{A}_t = \mathscr{A}_{z+1} \cup E_{\Pi^\theta}$
      Sort $\Pi^\theta \setminus E_{\Pi^\theta}$
      $z = z+1-|E_{\Pi^\theta}|$
      $\mathscr{A}_z = \mathscr{A}_t$
      $\Pi^\theta = (\Pi^\theta \setminus E_{\Pi^\theta}) \cup \mathscr{A}_z$
    end if
    $\mathscr{Q} = \mathscr{Q} \setminus \left\{Arc_{i,j}^k\right\}$
  end while
end procedure

The result of this algorithm is a set of sub-networks $\Pi^\theta = \{\mathscr{A}_1,\ldots,\mathscr{A}_z\}$ which satisfies the following property :

$$\forall(\mathscr{A}_x,\mathscr{A}_y) \in (\Pi^\theta \times \Pi^\theta)\ \ \mathscr{A}_x \cap \mathscr{A}_y = \emptyset,\ \ x \neq y$$

We define the following family of Boolean variables :

$$\gamma_{i,j,k,s} = \begin{cases} 1 & \text{if } Arc_{i,j}^k \text{ belongs to } \mathscr{A}_s \\ 0 & \text{otherwise} \end{cases}$$

And $0 < Max_{\frac{\phi}{\hat{\phi}}} \mathscr{A}_s \leq 1,\ \forall \mathscr{A}_s \in \Pi^\theta$.

The new equation to calculate the flow rate of vehicles on paths at each time slot θ is therefore as follows :

$$\lambda_{i,j,k,\theta} = \frac{3600}{x_{i,j,k,\theta}} \cdot Max((Max_{\frac{\phi}{\hat{\phi}}} \mathscr{A}_s) \cdot \gamma_{i,j,k,s})\ \ \forall \mathscr{A}_s \in \Pi^\theta \qquad (4)$$

Now the organization of flow in network is optimized without congestion by the equation 4, we present in the following the algorithm of the parallel approach developed in the vehicles pursuit model (see algorithm 2). This algorithm enables to minimize the departure times of buildings to be evacuated ensuring a continuity of flow in network without traffic-jams.

Theorem 2 *The velocity of a flow of vehicles ϕ_l crossing a road link l with a capacity $\hat{\phi}_l \cdot \alpha\ veh/(hour \cdot \alpha)$, where $\alpha > 0$, is equal to the velocity of a flow of vehicles $\frac{\phi_l}{\alpha}$ crossing l with l has a capacity $\hat{\phi}_l\ veh/hour$*

This theorem which can be proved based on the traffic model (see section 2.3) is shown by the line 18 of the algorithm 2. This algorithm takes as input at each time slot θ the set of buildings $\mathbf{B}^\theta$. Its purpose is to compute the flow rate of vehicles $\lambda_{i,j,k,\theta}, \mathscr{B}_i \in \mathbf{B}^\theta$ from $\mathbf{B}^\theta$ (line 10) and the departure times of these vehicles $t_{i,j,k,\theta}$ (line 59, algorithm 2). This algorithm ensures traffic fluidity while minimizing the total evacuation time. It assumes that all vehicles of all buildings in all time intervals leave at the beginning of the evacuation (line 11, algorithm 2). At each time slot $\theta \in \{1,\ldots,\Theta\}$ and for every path $Arc_{i,j}^k / x_{i,j,k,\theta} > 0$, this algorithm simulates in real time the trajectory of flow of vehicles in order to compute its start time. On each arc belonging to this path, we check if vehicles of previous time intervals sharing this arc have left before the first vehicle on route $Arc_{i,j}^k$ enters this arc at the current time interval (lines 17-18, algorithm 2).

If this is not the case, we divide the current arc in several portions (see figure 8) and compute the difference time i.e. the waiting time of the first vehicle of flow outgoing from $\mathscr{B}_i$ on $Arc_{i,j}^k$ before leaving (Lines 19-31, algorithm 2). Ans so on all the arcs of path $Arc_{i,j}^k$.

In order not to update the start and release times of arcs on the current path at each time difference founded and for technical reasons (to keep the release times of these arcs by vehicles that passed at the previous time slots and not those who are passing in the current time slot) we compute the start time of vehicles and update the release times just after simulate the paths of all outgoing vehicles from buildings $\mathbf{B}^\theta$ (Lines 39-64).

3.4.2. Tests of the parallel method with and without decomposition of evacuation network

A sample evacuation network is given by the figure 11. In this network we consider 4 buildings to be evacuated, the priority list established is $\{\mathscr{B}_2,\mathscr{B}_1,\mathscr{B}_4,\mathscr{B}_3\}$. Being $\mathscr{B}_4$ must be evacuated to two shelters so an other level of priority is established and the list can be seen as

Algorithm 2 Parallel pursuit of vehicles ▷ Part I

st =start_time() ▷ Start time of the evacuation (h : m : s : ms)
$\hat{\epsilon}_l^0 = \epsilon_l^0 = st, \forall l \in \mathscr{A}$ ▷ Entry and release times of arc l initialized to st

procedure PARALLEL METHOD
  $Decomposition()$ ▷ Build Π^θ, $\theta \geq 1$
  $\hat{\epsilon}_l^\theta = \hat{\epsilon}_l^{\theta-1}, \epsilon_l^\theta = \epsilon_l^{\theta-1}, \forall l \in \mathscr{A}$
  for $\mathscr{B}_i \in \mathbf{B}^\theta$ **do**
    for $\mathscr{S}_j \in \mathscr{H}_i$ **do**
      for $k = 1$ **to** $\eta_{i,j}$ **do**
        if $x_{i,j,k,\theta} > 0$ **then**
          $\lambda_{i,j,k,\theta} = \dfrac{3600 \cdot \mathrm{MAX}((\mathrm{MAX}_{\frac{\phi}{\hat{\phi}}} \mathscr{A}_s) \cdot \gamma_{i,j,k,s})}{x_{i,j,k,\theta}}, \forall \mathscr{A}_s \in \Pi^\theta$ ▷ Flow rate
          $ct = st$ ▷ Initialization of current time
          $sum\_dt = 0$
          for $l \in \mathrm{Arc}_{i,j}^k$ **do**
            $dt = 0$ ▷ Initialization of difference time
            $ctt = ct$
            $hms = cmp\_time(ct, \epsilon_l)$
            if $hms = \vec{0}$ **then** ▷ $ct \geq \epsilon_l^{\theta-1}$
              $ct = ct + \Delta_l^{\frac{\phi_l^\theta \times 3600}{\lambda_{i,j,k,\theta} \cdot x_{i,j,k,\theta}}}$ ▷ Add the travel time on arc l
            else
              $c\hat{\epsilon}_l = \hat{\epsilon}_l^{\theta-1}$ ▷ Copy
              for $p = 1$ **to** $l.prt\_nb()$ **do** ▷ Number of portions of road l
                $c\hat{\epsilon}_l = c\hat{\epsilon}_l + \hat{t}_{l_1}$ ▷ Transit time on each portion : $\hat{t}_{l_1} = \ldots = \hat{t}_{l_p}$
                $hms = cmp\_time(ct, c\hat{\epsilon}_l^{\theta-1})$
                if $hms \neq \vec{0}$ **then**
                  $dt = dt + hms$ ▷ Increment the difference time
                  $ct = ct + hms$
                  $sum\_dt = sum\_dt + hms$
                end if
                $ct = ct + \Delta_{l_1}^{\frac{\phi_l^\theta \times 3600}{\lambda_{i,j,k,\theta} \cdot x_{i,j,k,\theta}}}$ ▷ Add the travel time
              end for
            end if
            $v\_\epsilon.add(ct)$, $v\_\hat{\epsilon}.add(ctt)$, $v\_dt.add(dt)$
          end for
          $v\_sum\_dt.add(sum\_dt)$
        end if
      end for
    end for
  end for
  $it = iv = 1$
  for $\mathscr{B}_i \in \mathbf{B}^\theta$ **do**
    for $\mathscr{S}_j \in \mathscr{H}_i$ **do**
      for $k = 1$ **to** $\eta_{i,j}$ **do**
        if $x_{i,j,k,\theta} > 0$ **then**
          $sub\_sum = 0$
          for $l \in \mathrm{Arc}_{i,j}^k$ **do**
            $v\_\hat{\epsilon}[it] = v\_\hat{\epsilon}[it] + v\_sum\_dt[iv] - sub\_sum$ ▷ 1^{st} veh entering
            $v\_\hat{\epsilon}[it] = v\_\hat{\epsilon}[it] + \lambda_{i,j,k,\theta} \cdot (x_{i,j,k,\theta} - 1)$ ▷ Last vehicle entering
            $sub\_sum = sub\_sum + v\_dt[it]$
            $v\_\epsilon[it] = v\_\hat{\epsilon}[it] + v\_sum\_dt[iv] - sub\_sum$ ▷ 1^{st} vehicle leaving
            $v\_\epsilon[it] = v\_\epsilon[it] + \lambda_{i,j,k,\theta} \cdot (x_{i,j,k,\theta} - 1)$ ▷ Last vehicle leaving
            if $v\_\epsilon[it] \geq \epsilon_l^\theta$ **then** ▷ Update entry, release and travel times
              $\epsilon_l^\theta = v\_\epsilon[it]$
              $\hat{\epsilon}_l^\theta = v\_\hat{\epsilon}[it]$
              $\hat{t}_{l_1} = \Delta_{l_1}^{\frac{\phi_l^\theta \times 3600}{\lambda_{i,j,k,\theta} \cdot x_{i,j,k,\theta}}}$
            end if
            $it = it + 1$
          end for
          $f = \mathrm{Arc}_{i,j}^{k,1}$ ▷ First arc in path $\mathrm{Arc}_{i,j}^k$
          $t_{i,j,k,\theta} = \hat{\epsilon}_f - \lambda_{i,j,k,\theta} \cdot (x_{i,j,k,\theta} - 1)$ ▷ Departure time of first vehicle
          $iv = iv + 1$
        end if
      end for
    end for
  end for
end procedure

$\{\mathscr{M}_{1,2}, \mathscr{M}_{1,1}, \mathscr{M}_{1,4}, \mathscr{M}_{2,4}, \mathscr{M}_{2,3}\}$. The input of this network is given in figure 12.

The figure 13 shows the evacuation scheduling of the set of buildings $\{\mathscr{B}_1, \mathscr{B}_2, \mathscr{B}_3, \mathscr{B}_4\}$ performed by an evacuation scheduling system developed in the evacuation model STOM. The number of time slots Θ for evacuating these buildings is 2, and for each time slot $\theta \in \{1, \ldots, \Theta\}$ we know $x_{i,j,k,\theta}$ the number of vehicles evacuated by path $\mathrm{Arc}_{i,j}^k, 1 \leq k \leq \eta_{i,j}$ and a priori $\hat{\phi}_{s,\mathscr{M}_{j,i}} = \sum_{k=1}^{\eta_{i,j}} x_{i,j,k,\theta}$ the number of vehicles evacuated from $\mathscr{B}_i$ to $\mathscr{S}_j$, $1 \leq i \leq 4, \mathscr{S}_j \in \mathscr{H}_i$.

The figure 14 shows, at first time slot, the approach used to compute the departure moments of vehicles from buildings $\{\mathscr{B}_1, \mathscr{B}_2\}$.

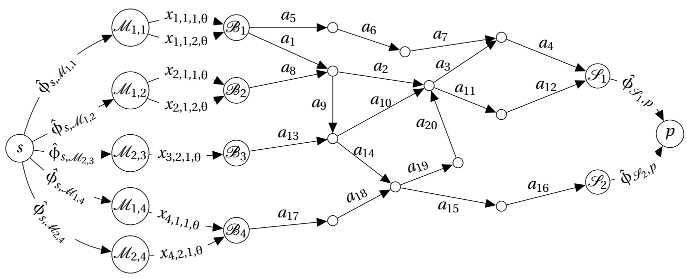


FIGURE 11: Sample evacuation network $\mathscr{G}^* = (\mathscr{N}^*, \mathscr{A}^*)$

3.4.2.1. First slot.

The decomposition of graph, at first slot, generates the same graph with $\mathrm{MAX}_{\frac{\phi}{\hat{\phi}}} \mathscr{A}_1 = 1$. The expression for computing the flow rates of vehicles gives $\lambda_{i,j,k,\theta} = \frac{900}{x_{i,j,k,\theta}}$ (the time unit of arcs capacities is here equal to 15 minutes).

The departure moments of vehicles from all buildings, at first slot, are given by the table 1.

i	j	k	θ	$x_{i,j,k,\theta}$	$\lambda_{i,j,k,\theta}$	$t_{i,j,k,\theta}$
1	1	1	1	200	4.5	10 : 0 : 0 : 0
1	1	2	1	350	2.571	10 : 0 : 0 : 0
2	1	1	1	150	6	10 : 0 : 0 : 0
2	1	2	1	350	2.571	10 : 0 : 0 : 0
4	2	1	1	600	1.5	10 : 0 : 0 : 0

TABLE 1: Departure times of vehicles from all buildings at first slot : parallel method

Similarly, the entry, release and travel times on roads at first time slot, are given by the table 2.

3.4.2.2. Second slot.

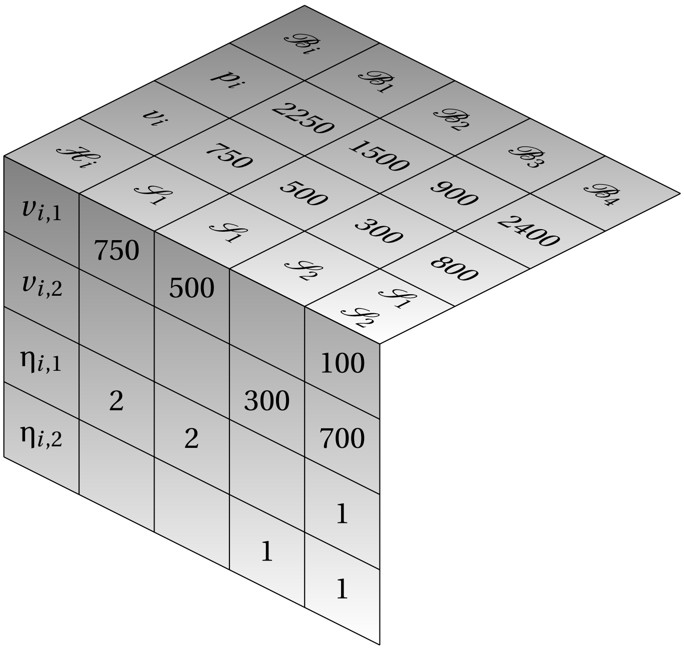
The main objective of VPM is to minimize, at each slot, the departure times of vehicles avoiding overlap (congestion) in evacuation network. Departure times of vehicles from all buildings evacuated at the first time interval are identical (10 : 0 : 0 : 0), because the network is considered empty (this does not eliminate the hypothesis of loading of evacuation network by a some number of vehicles, but we assume here that the capacities of arcs are computed taking into account this load). The departure dates of vehicles from buildings evacuated at the second slot are different from those of the first one.

$Arc^1_{1,1}$	$Arc^2_{1,1}$	$Arc^1_{2,1}$	$Arc^2_{2,1}$	$Arc^1_{3,2}$	$Arc^1_{4,1}$	$Arc^1_{4,2}$
a_1	a_5	a_8	a_8	a_{13}	a_{17}	a_{17}
a_2	a_6	a_2	a_9	a_{14}	a_{18}	a_{18}
a_{13}	a_7	a_3	a_{10}	a_{15}	a_{19}	a_{15}
a_4	a_4	a_4	a_{11}	a_{16}	a_{20}	a_{16}
			a_{12}		a_3	
					a_4	

a: Paths between buildings and shelters

l	L_l (Km)	$\hat{\phi}_l$ $(veh/\frac{1}{4}h)$	l	L_l (Km)	$\hat{\phi}_l$ $(veh/\frac{1}{4}h)$
a_1	1.5	500	a_{11}	2	700
a_2	1.4	350	a_{12}	3	700
a_3	1.2	350	a_{13}	3	350
a_4	1.3	700	a_{14}	1	500
a_5	3	350	a_{15}	3	600
a_6	1.4	500	a_{16}	2	600
a_7	2	500	a_{17}	2	600
a_8	3	500	a_{18}	1.5	600
a_9	2	350	a_{19}	1	500
a_{10}	1.7	350	a_{20}	1	350

b: Road links caracteristics



c: Number of vehicles to be evacuated from buildings to shelters

FIGURE 12: Evacuation model input

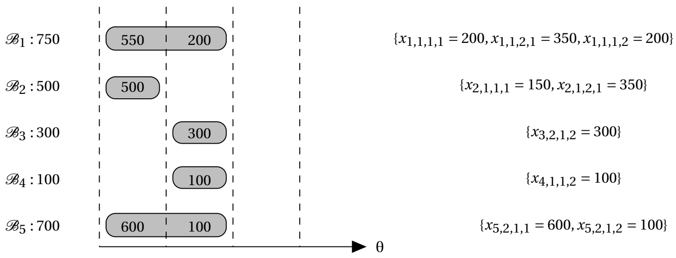


FIGURE 13: Scheduling evacuation

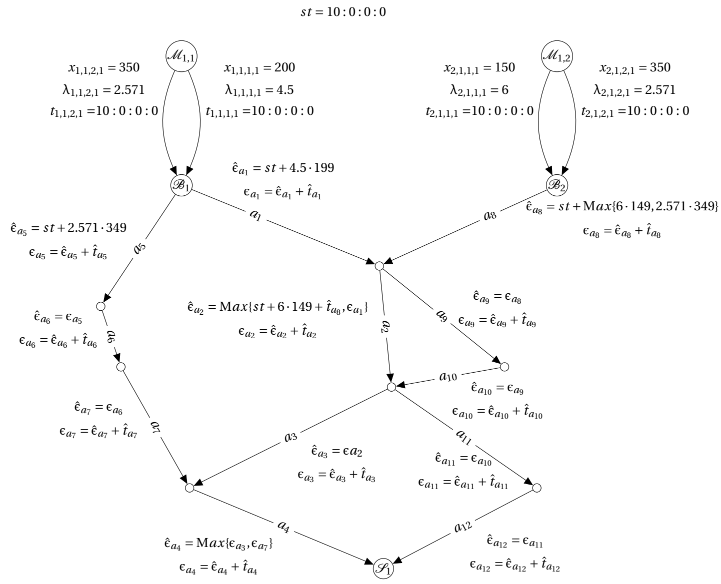


FIGURE 14: Calculation of departure times $t_{1,1,1,1}$, $t_{1,1,2,1}$, $t_{2,1,1,1}$, $t_{2,1,2,1}$

l	$\hat{\epsilon}_l$ (time)	ϵ_l (time)	$\hat{t}_l$ (seconds)
a_1	10 : 14 : 55 : 500	10 : 17 : 3 : 183	127.683
a_2	10 : 21 : 7 : 499	10 : 25 : 16 : 498	249
a_3	10 : 25 : 16 : 498	10 : 28 : 49 : 926	213.428
a_4	10 : 29 : 0 : 124	10 : 30 : 55 : 730	115.607
a_5	10 : 14 : 57 : 428	10 : 23 : 50 : 998	533.571
a_6	10 : 23 : 50 : 998	10 : 25 : 58 : 285	127.287
a_7	10 : 25 : 58 : 285	10 : 29 : 0 : 124	181.839
a_8	10 : 14 : 57 : 428	10 : 21 : 10 : 927	373.499
a_9	10 : 21 : 10 : 927	10 : 27 : 6 : 640	355.714
a_{10}	10 : 27 : 6 : 640	10 : 32 : 8 : 996	302.357
a_{11}	10 : 32 : 8 : 996	10 : 34 : 12 : 534	123.538
a_{12}	10 : 34 : 12 : 534	10 : 37 : 17 : 841	185.307
a_{15}	10 : 21 : 1 : 623	10 : 26 : 12 : 872	311.25
a_{16}	10 : 26 : 12 : 872	10 : 29 : 40 : 371	207.5
a_{17}	10 : 14 : 58 : 500	10 : 18 : 25 : 999	207.5
a_{18}	10 : 18 : 25 : 999	10 : 21 : 1 : 623	155.625

TABLE 2: Entry, release and travel times on roads at first slot : parallel method

According to the formula 2 and for the second slot, table 3 provides the departure moments. Beside, table 4 indicates, for each road link, the entry, release and travel times. On the other side, the application of the formula 4 is the result of all information illustrated by tables 5 and 6. The evacuation network decomposition algorithm led to the construction of a single sub-network with a maximum value of ratio of

flow to capacity equal to 0.857 (unlike the first slot where the maximum ratio of flow to capacity were equal to 1). This is due to a sub-optimal use of arcs (there is no more vehicles to be evacuated through this sub-network). In this case, vehicles leave buildings during $0.857 \cdot \frac{1}{4}h$, and speed on arcs is computed according to theorem 2 (see also algorithm 2). For example the speed on arcs whose flow capacity ratio is 0.857, is equal to that on the same arcs when flow is equal to capacity (see theorem 2).

i	j	k	θ	$x_{i,j,k,\theta}$	$\lambda_{i,j,k,\theta}$	$t_{i,j,k,\theta}$
1	1	2	2	200	4.5	10 :21 :46 :618
3	2	1	2	300	3	10 :14 :59 :345
4	1	1	2	100	9	10 :19 :19 :896
4	2	1	2	100	9	10 :19 :20 :618

TABLE 3: Departure times at second time slot without decomposition of evacuation network

l	$\hat{\epsilon}_l$ (time)	ϵ_l (time)	$\hat{t}_l$ (seconds)
a_1	10 :36 :42 :119	10 :38 :49 :616	127.683
a_2	10 :38 :49 :616	10 :41 :44 :999	175.487
a_3	10 :41 :44 :999	10 :44 :32 :509	167.593
a_4	10 :44 :32 :509	10 :45 :51 :830	79.361
a_{13}	10 :29 :56 :346	10 :36 :55 :328	418.982
a_{14}	10 :36 :55 :328	10 :38 :23 :644	88.3365
a_{15}	10 :38 :23 :664	10 :42 :8 :345	224.894
a_{16}	10 :42 :8 :345	10 :44 :38 :81	149.93
a_{17}	10 :34 :11 :619	10 :36 :32 :144	140.731
a_{18}	10 :36 :32 :144	10 :38 :17 :663	105.548
a_{19}	10 :38 :16 :940	10 :39 :40 :440	83.5007
a_{20}	10 :39 :40 :440	10 :41 :40 :499	120.06

TABLE 4: Road links at second time slot without decomposition of evacuation network

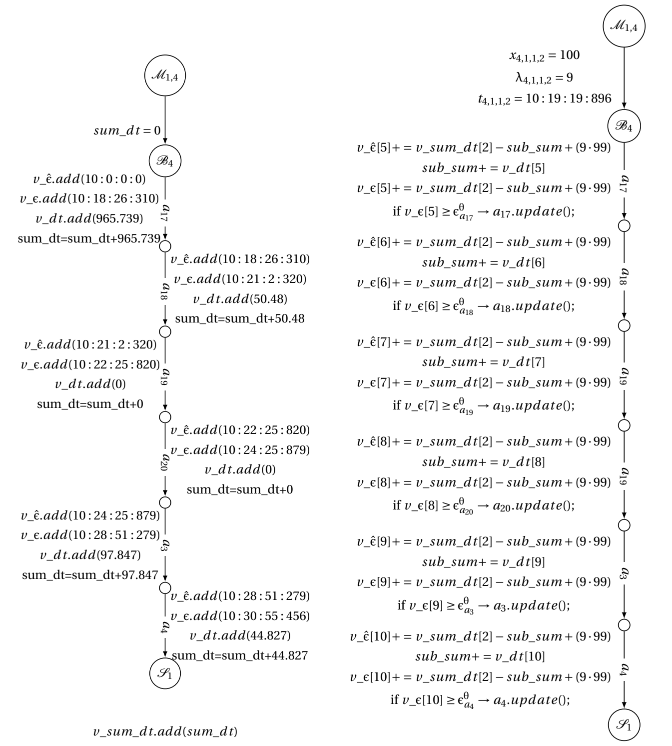
i	j	k	θ	$x_{i,j,k,\theta}$	$\lambda_{i,j,k,\theta}$	$t_{i,j,k,\theta}$
1	1	2	2	200	3.857	10 :20 :54 :223
3	2	1	2	300	2.571	10 :12 :46 :606
4	1	1	2	100	7.714	10 :18 :31 :23
4	2	1	2	100	7.714	10 :19 :3 :544

TABLE 5: Departure times at second time slot with decomposition of evacuation network

A detailed calculation of the start time $t_{4,1,1,2}$ of vehicles outgoing from $\mathscr{B}_4$ to $\mathscr{S}_1$ at second slot is given by the figure 15. The table 7 provides some attributes of path $Arc^1_{4,1}$. We note that no attribute on arcs $\{a_{19}, a_{20}\}$ is provided since these two arcs were unoccupied at the first slot (the algorithm will not enter in the bloc of $else$ (lines 19-31, algorithm 2)).

l	$\hat{\epsilon}_l$ (time)	ϵ_l (time)	$\hat{t}_l$ (seconds)
a_1	10 :33 :41 :795	10 :35 :50 :638	128.964
a_2	10 :35 :50 :638	10 :38 :50 :458	179.915
a_3	10 :38 :50 :458	10 :42 :23 :732	213.428
a_4	10 :42 :23 :732	10 :43 :43 :910	80.2998
a_{13}	10 :25 :35 :464	10 :34 :29 :34	533.571
a_{14}	10 :34 :29 :34	10 :35 :59 :953	90.9196
a_{15}	10 :35 :59 :953	10 :39 :54 :16	234.258
a_{16}	10 :39 :54 :16	10 :42 :29 :970	156.172
a_{17}	10 :31 :47 :259	10 :34 :8 :720	141.662
a_{18}	10 :34 :8 :720	10 :35 :54 :810	106.246
a_{19}	10 :35 :22 :289	10 :36 :45 :975	83.6861
a_{20}	10 :36 :45 :975	10 :38 :46 :601	120.626

TABLE 6: Road links at second time slot with decomposition of evacuation network



a: Part I b: Part II

FIGURE 15: The steps for computing $t_{4,1,1,2}$

3.4.3. Simple example of parallel pursuit of vehicles

The figure 16 shows, for a small example, the steps performed by the algorithm from the second slot. For simplification reasons, we will not use in this example all the mobilized variables in the algorithm 2. This example does not attempt to identify all scenarios. For a more comprehensive understanding, we invite the reader to refer to the algorithm 2 and previous examples. The example assumes a part of network shared by two paths of two different buildings. Also for

l	$prt\_nb()$	L_{l_1} *meter*	$\hat{t}_l$ *second*	$\hat{t}_{l_1}$ *second*
a_3	115	10.434	213.428	1.855
a_4	124	10.483	115.606	0.932
a_{17}	191	10.471	207.499	0.736
a_{18}	143	10.489	155.624	0.738
a_{19}	-	-	-	-
a_{20}	-	-	-	-

TABLE 7: Characteristics on arcs belonging to path $Arc^1_{4,1}$ at second time slot (with decomposition)

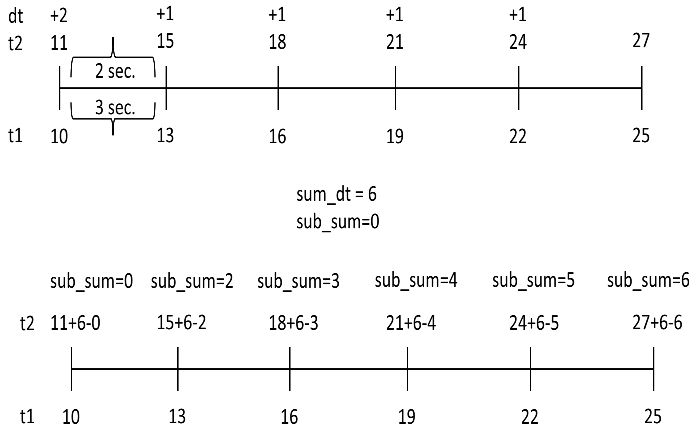


FIGURE 16: Simple example of pursuit of vehicles, $\theta > 1$

simplification reasons, we assume that the number of portions is equal to one for each road link, and that part of network is first used by vehicles outgoing from $\mathscr{B}_1$ at time slot θ. we denote by t1, sequentially, the release time of each arc by the last vehicle outgoing from $\mathscr{B}_1$. Similar for $\mathscr{B}_2$ with the difference that this building takes this part of network at slot $\theta + 1$. The first values of t1 and t2 (10 and 11 respectively) correspond to the release time(s) of arc(s) whose origin node of first arc (belonging to this part of network) corresponds to destination node of these arcs. The maximal flow rate and the length of every road link in this part of network are considered equal, the journey time at a given time slot is therefore the same on every road. The journey time (figure 16) differs from one slot to another because of the difference of flow, where the journey time was 3 seconds at time slot θ against 2 seconds at time slot $\theta + 1$. This result is due to a flow outgoing from $\mathscr{B}_1$ higher than that from $\mathscr{B}_2$. The difference time expressed by the variable dt is the waiting time needed for that the first vehicle outgoing from $\mathscr{B}_2$ can use the current arc (current portion in case if an arc is divided in several portions). At slot $\theta + 1$ and for each arc belonging to this part of network, dt and $t2$ are computed as follows :

$$dt = t1 - t2 \qquad and \qquad t2 = t2 + dt + 2$$

The waiting in network nodes is only authorized from an algorithmic point of view. In other words, at each slot, the variables will be updated once all paths of all buildings have been processed. Finally, the algorithm applies a back strategy to update time variables on arcs (see figure 16), and then to set the start time of vehicles through the given path. This strategy (delay start time) guarantees the absence of overlap between the two time intervals $\mathscr{I}_\theta$ and $\mathscr{I}_{\theta+1}$ (network without congestion).

3.5. Sequential method of pursuit of vehicles and harmonization of flow

The modification represented by the construction of a set of sub-networks at each time slot reduces the total clearance time, but not entirely, because all vehicles using paths belonging to the same sub-network have the same period time for being evacuated. In addition, for a large time unit of slot (for example 1 hour) and a small number of vehicles to be evacuated from buildings (for example 10 vehicles), the organization of the evacuation becomes difficult : these buildings will be forced to take 1 hour to evacuate 10 vehicles only. Finally, the organization of departure times of vehicles on roads sharing common arcs requires a system of periodic tasks within a given time slot. The figure 10 (page 7) provides an explication on this periodic scheduling. The capacity of the common arc in this example is equal to one vehicle each 2.57 seconds, and the flow rate of vehicles of buildings is given by :

$$\lambda_{1,j,k,\theta} = \lambda_{2,j,k,\theta} = \frac{3600 \cdot \mathrm{M}ax\{\mathrm{M}ax_{\frac{\phi}{\Phi}} Arc^k_{1,j}, \mathrm{M}ax_{\frac{\phi}{\Phi}} Arc^k_{2,j}\}}{x_{1,j,k,\theta}} = 5.14$$

According to this scheduling method, $\mathscr{B}_1$ begins to evacuate first its vehicles then $\mathscr{B}_2$ does the same 2.57 seconds later. Thereby, this strategy ensure that only on vehicle crosses the common arc each 2.57 seconds, unlike the case where $\mathscr{B}_1$ and $\mathscr{B}_2$ begin simultaneously (2 vehicles crossing the common arc each 5.14 seconds).

This periodic scheduling must take into account the distance among buildings and common arcs. It should be noted that such a clarification in case of massive evacuation is not required since the behavior of evacuees are not generally calibrated to the scale of the second. In addition, traffic flow model adopted allows such phenomena to occur in the network, without risk of congestion on roads.

To solve the problem of too few vehicles, a sequential evacuation method has been developed here. The flow rate of vehicles $\lambda_{i,j,k}$ is a constant equal to the maximal flow rate on minimum arc of path $Arc^k_{i,j}$ (see formula 3). At each time slot, the departure of vehicles on a path, which shares arcs with another path, starts a period of time (the time needed to avoid an overlap) after the last vehicle using the other path (from a priority building or from the current building) leaves its building (see algorithm 3).

The overload problem between vehicles in the sequential method is not only between successive time slots (the case of parallel method) but within time slots themselves. Solving of this problem within a time slot follows the same strategy applied to solve it between successive time slots (see algorithm 3). The entry and release times of arcs belonging to a path are updated once the algorithm simulate this path (contrary to the parallel method where these times are not updated before that all paths from all buildings are simulated).

We take back the example (see figure 10) of $\mathscr{B}_1$ and $\mathscr{B}_2$, we see that according to the sequential method, $\mathscr{B}_1$ will evacuate its vehicles in a half hour, then $\mathscr{B}_2$ will start just after $\mathscr{B}_1$ exits its last vehicle. The sequential method organizes better the evacuation, it allows to define for each building a minimum evacuation time interval with a maximal flow rates on paths

(maximal flow rates of minimum arcs).

Algorithm 3 Sequential pursuit of vehicles

```
procedure SEQUENTIAL METHOD
    st=start_time()                                  ▷ Departure time of hte evacuation (h : m : s : ms)
    ε̂_l = ε_l = st, for θ = 1, ∀l ∈ 𝒜

    for ℬ_i ∈ B^θ do
        for 𝒮_j ∈ ℋ_i do
            for k = 1 to η_{i,j} do
                if x_{i,j,k,θ} > 0 then
                    λ_{i,j,k} = 3600 / Min_φ̂ Arc^k_{i,j}       ▷ Flow rate of vehicles on Arc^k_{i,j}
                    ct = st                          ▷ Initialization of current time
                    sum_dt = 0
                    for l ∈ Arc^k_{i,j} do
                        dt = 0                       ▷ Initialization of difference time
                        hms = cmp_time(ct, ε_l)
                        if hms = 0⃗ then                ▷ ct ≥ ε_l
                            ctt = ct
                            ct = ct + Δ_l^{Min_φ̂ Arc^k_{i,j}}   ▷ Add the travel time on arc l
                        else
                            ctt = ct
                            for p = 1 to l.prt_nb() do      ▷ Number of portions of arc l
                                ε̂_l = ε̂_l + t̂_{l_1}    ▷ Travel time on each portion : t̂_{l_1} = ... = t̂_{l_p}
                                hms = cmp_time(ct, ε̂_l)
                                if hms ≠ 0⃗ then
                                    sum_dt = sum_dt + hms
                                    dt = dt + hms
                                    ct = ct + hms
                                end if
                                ct = ct + Δ_{l_1}^{Min_φ̂ Arc^k_{i,j}}   ▷ Add the travel time
                            end for
                        end if
                        ε̂_l = ctt + (λ_{i,j,k} · (x_{i,j,k,θ} − 1))
                        ε_l = ct + (λ_{i,j,k} · (x_{i,j,k,θ} − 1))
                        t̂_l = Δ_l^{Min_φ̂ Arc^k_{i,j}}
                        v_dt.add(dt)
                    end for
                    sub_sum = 0, it = 1
                    for l ∈ Arc^k_{i,j} do
                        ε̂_l = ε̂_l + (sum_dt − sub_sum)      ▷ Update entry time
                        sub_sum = sub_sum + v_dt[it]
                        ε_l = ε_l + (sum_dt − sub_sum)      ▷ Update release time
                        it = it + 1
                    end for
                    f = Arc^{k,1}_{i,j}              ▷ First arc in the k-th path connecting ℬ_i to 𝒮_j
                    t_{i,j,k,θ} = ε̂_f                ▷ Departure time of first vehicle
                    v_dt.clear()
                end if
            end for
        end for
    end for
end procedure
```

3.5.1. Tests on sequential method

The application of the sequential method to the evacuation problem present in figures 11 and 12 results in the following four tables. The table 8 provides the departure times of vehicles from all buildings at first time slot, while the table 9 gives the entry, release and travel times on road links at this same time slot. Similarly the two tables 10 and 11 provide these information at the second time slot.

3.6. Harmonization of flow for intervals of short periods

Despite flow rates of vehicles are maximal in the sequential method, the total evacuation time increases. This is due to the fact that this method does not benefit from common arcs (non-parallel evacuation). Indeed, the problem of increase of the evacuation time occurs when common arc between several paths has a capacity greater than the capacity of minimum arc

i	j	k	θ	$x_{i,j,k,\theta}$	$\lambda_{i,j,k,\theta}$	$t_{i,j,k,\theta}$
1	1	1	1	200	2.571	10 : 0 : 0 : 0
1	1	2	1	350	2.571	10 : 2 : 34 : 813
2	1	1	1	150	2.571	10 : 16 : 1 : 226
2	1	2	1	350	2.571	10 : 21 : 0 : 288
4	2	1	1	600	1.5	10 : 0 : 0 : 0

TABLE 8: Departure moments of vehicles from all buildings at first time slot : sequential method

l	$\hat{\epsilon}_l$ (time)	ϵ_l (time)	$\hat{t}_l$ (seconds)
a_1	10 : 6 : 38 : 0	10 : 8 : 54 : 379	136.379
a_2	10 : 25 : 31 : 984	10 : 29 : 40 : 889	249
a_3	10 : 29 : 40 : 889	10 : 33 : 14 : 214	213.428
a_4	10 : 33 : 14 : 214	10 : 34 : 34 : 384	80.299
a_5	10 : 14 : 12 : 814	10 : 23 : 6 : 384	533.571
a_6	10 : 23 : 6 : 384	10 : 25 : 13 : 671	127.287
a_7	10 : 25 : 13 : 671	10 : 28 : 15 : 510	181.839
a_8	10 : 32 : 38 : 288	10 : 37 : 10 : 825	372.759
a_9	10 : 37 : 10 : 825	10 : 43 : 6 : 538	355.714
a_{10}	10 : 43 : 6 : 538	10 : 48 : 8 : 894	302.357
a_{11}	10 : 48 : 8 : 894	10 : 50 : 12 : 432	123.538
a_{12}	10 : 50 : 12 : 432	10 : 53 : 17 : 739	185.307
a_{15}	10 : 16 : 2 : 123	10 : 21 : 13 : 372	311.25
a_{16}	10 : 21 : 13 : 372	10 : 24 : 40 : 871	207.5
a_{17}	10 : 9 : 59 : 0	10 : 13 : 26 : 499	207.5
a_{18}	10 : 13 : 26 : 499	10 : 16 : 2 : 123	155.625

TABLE 9: Entry, release and travel times on the roads at first slot : sequential method

i	j	k	θ	$x_{i,j,k,\theta}$	$\lambda_{i,j,k,\theta}$	$t_{i,j,k,\theta}$
1	1	2	2	200	2.571	10 : 23 : 17 : 620
3	2	1	2	300	2.571	10 : 8 : 10 : 800
4	1	1	2	100	2.571	10 : 27 : 37 : 500
4	2	1	2	100	1.5	10 : 30 : 56 : 266

TABLE 10: Departure dates at second slot : sequential method

of one or more of these paths. The example in the figure 17 shows this case.

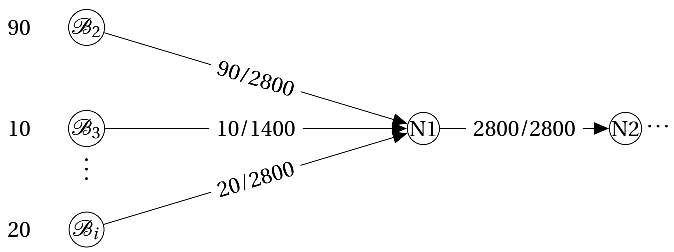


FIGURE 17: Problem of common arcs, weight : flow/capacity(in hour)

l	$\hat{\epsilon}_l$ (time)	ϵ_l (time)	$\hat{t}_l$ (seconds)
a_1	10 : 29 : 55 : 620	10 : 32 : 11 : 899	136.379
a_2	10 : 32 : 11 : 899	10 : 36 : 20 : 813	249
a_3	10 : 39 : 40 : 717	10 : 43 : 13 : 991	213.428
a_4	10 : 43 : 13 : 991	10 : 44 : 34 : 219	80.299
a_{13}	10 : 18 : 8 : 801	10 : 27 : 2 : 371	533.571
a_{14}	10 : 27 : 2 : 371	10 : 28 : 33 : 290	90.919
a_{15}	10 : 38 : 38 : 276	10 : 43 : 49 : 525	311.25
a_{16}	10 : 43 : 49 : 525	10 : 47 : 17 : 24	207.5
a_{17}	10 : 32 : 35 : 266	10 : 36 : 2 : 692	207.5
a_{18}	10 : 36 : 2 : 692	10 : 38 : 38 : 276	155.625
a_{19}	10 : 35 : 11 : 942	10 : 36 : 42 : 861	90.919
a_{20}	10 : 36 : 42 : 861	10 : 39 : 40 : 717	177.857

TABLE 11: Road links at second slot : sequential method

According to parallel method, $\mathcal{B}_3$ must evacuate its 10 vehicles during one hour with a flow rate $\lambda_{3,j,k,\theta} = \frac{3600}{10} =$ 360 seconds per vehicle. While the sequential method allows $\mathcal{B}_3$ to inject its 10 vehicles during 25.7 seconds with a flow rate $\lambda_{3,j,k} = \frac{3600}{1400} = 2.57$. The evacuation departure of $\mathcal{B}_3$ in this case is just after the departure time of the last vehicle from $\mathcal{B}_2$ (priority list of buildings). We see that the increase of evacuation time in the sequential method is due to the fact that $\mathcal{B}_3$ does not comply with the time period dedicated by the arc (N1, N2). Indeed the capacity of common arc (N1, N2) is twice that of minimum arc ($\mathcal{B}_3$, N1) of path from $\mathcal{B}_3$.

To sum up, the parallel approach allows to minimize the total evacuation time using the advantage of common arcs, however it is less effective from an organizational point of view. In contrast, the sequential method is more efficient in organizational aspect but less interesting in terms of time optimization. Hence the interest to bring these two approaches together in order to simultaneously benefit of their advantages. Thus, the gap in the first approach can be filled based on the principle of the sequential method by the use of a smaller slot time unit (for example : 1 minute, 5 minutes, 15 minutes, etc.).

The example of figure 18 compares the time unit « hour » with decomposition of the evacuation network to time unit « minute ».

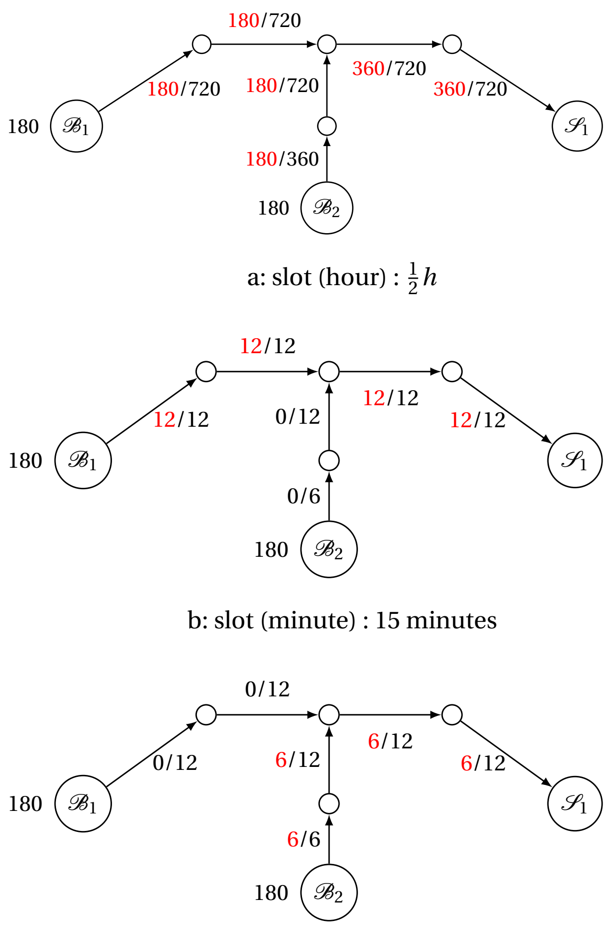
We note that slot of an hour enables a parallel evacuation of vehicles from $\mathcal{B}_1$ and $\mathcal{B}_2$ during 30 minutes, while the slot of one minute leads to a sequential evacuation during 15 minutes for $\mathcal{B}_1$ plus 30 minutes for $\mathcal{B}_2$ (travel time on arcs are not considered here in this comparison).

Note that if $\mathcal{B}_2$ has priority over $\mathcal{B}_1$, the total clearance time remains always equal to 30 minutes.

The other figure 19, in turn, compares the slots hour - minute and illustrates the interest of the latter in terms of reducing the total evacuation time.

In the case of slot of one hour, the part (a) shows that buildings $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ are evacuated during on hour. While through the slot of one minute (part b), theses buildings take only $Max(30, 40, 50) = 50$ minutes.

The two cases (hour and minute) consider the same list of priorities $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$. However, with another list of priorities



a: slot (hour) : $\frac{1}{2}h$

b: slot (minute) : 15 minutes

c: slot minute : 30 minutes

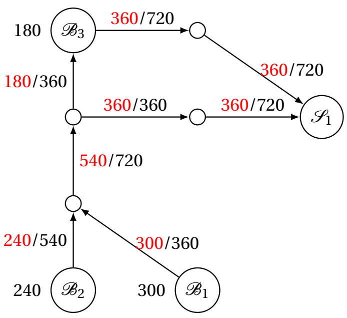
FIGURE 18: Preference of slot hour over slot minute

$\{\mathcal{B}_3, \mathcal{B}_2, \mathcal{B}_1\}$, parts (c) and (d) show an evacuation during 55 minutes.

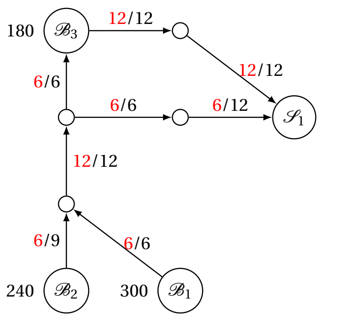
The transition to slot minute requires a real variables in order to keep the total capacity of network. In the context of a real evacuation plan, it should use for a building, an appropriate slot in the order of 15, 30 minutes, etc. The aggregation of evacuated vehicles per minute (real values), in slot of 15 minutes for example, gives a real number where the rounding by raising the lower bound is necessary. If the aggregation of a building results in a integer number of vehicles, however the total time of this aggregation does not necessarily correspond to time step of 15 minutes. In addiction, such aggregation is only possible if the evacuation slots (minute) of this building are consecutive (non-preemption evacuation). Faced with this dilemma, an average slot (15 minutes, 30 minutes, etc.) associated with an evacuation per homogeneous areas (neighborhoods, districts, etc.) can be a good alternative. The two main objectives of the vehicles pursuit model, (1) minimization of evacuation departure dates of buildings and (2) continuity of flow in the network without congestion, will be illustrated on a real evacuation network int the next and last section.

4. Applications and results

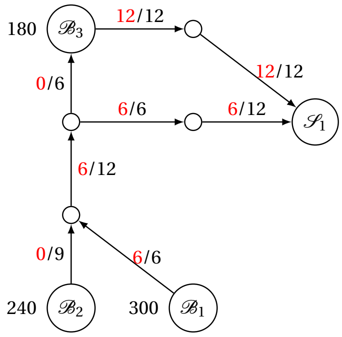
This paper, as we have previously mentioned, focuses on the conversion of the discrete result, provided by an



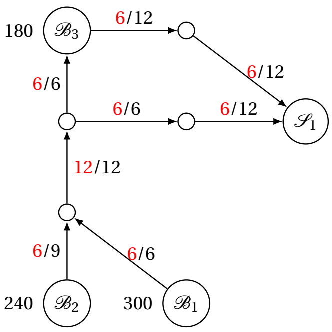
a: slot (hour) : 1 hour



b: slot (minute) : 30 min for $\mathscr{B}_3$, 40 min for $\mathscr{B}_2$ and 50 min for $\mathscr{B}_1$



c: slot (minute) : 15 minutes for $\mathscr{B}_3$, 15 minutes for $\mathscr{B}_1$



d: slot (minute) : 40 min for $\mathscr{B}_2$ and 35 min for $\mathscr{B}_1$

FIGURE 19: Preference of slot minute over slot hour

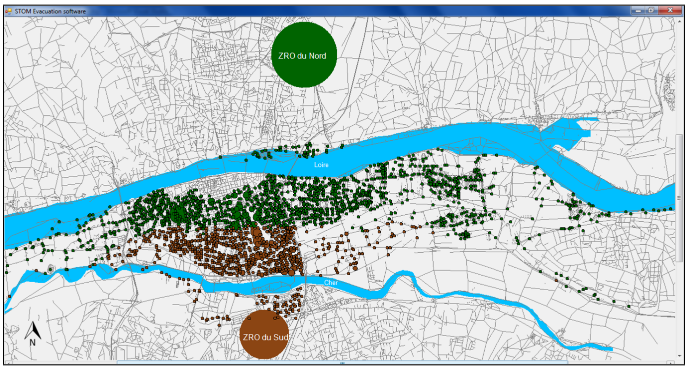


FIGURE 20: The valley of Tours, FR 37

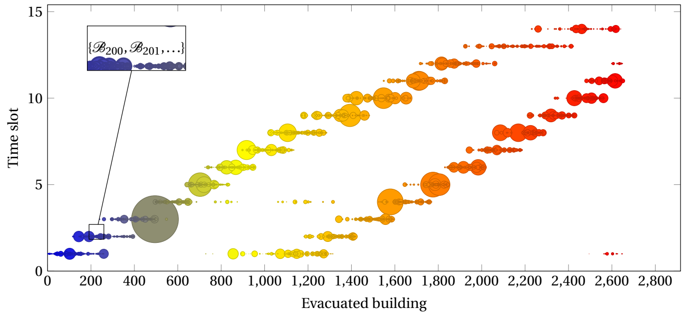


FIGURE 21: Evacuation of the valley of Tours, France 37

evacuation scheduling model (flow, time slot), to a continuous one using a mezoscopic pursuit vehicles model (flow rate, time interval). This conversion, which aims to minimize the departure times of vehicles and to ensure a continuity of flow in the network without overlap, will be exposed in this section on a real site, the valley of Tours [4] (see [3]). The evacuation network established involves : 9825 buildings grouped in 2753 buildings [5], 2 safety points, 13446 nodes and 22676 arcs (3-best paths predetermined between buildings and shelters [2]). The figure 20 shows this area and the buildings to be evacuated to two shelters (ZRO [6]), one in the North and other in the South.

Recall that the evacuation scheduling model is based on a buildings priority list established according to several criteria. The slot time unit is equal to 30 minutes.

The figure 21 shows the parallel evacuation of the valley of Tours (France, 37) to two shelters (North and South). This evacuation is achieved by an evacuation scheduling algorithm. The size of each point in this figure indicates the number of evacuated vehicles from each given building at each given time slot. The evacuated buildings are classified by descending order of evacuation priority and each color corresponds to a building. For further information about the evacuation scheduling system, we invite the reader to consult [1][4]

Now the discrete process is completed (flow, time slot), we present the evacuation network load over time (see figure 22). We remark some roads which load up to the maximum at every time slot, they are obviously the arcs with smaller capacities. The size of each point shown in this figure corresponds to the capacity of each given arc, and its color, per slot, provides an information on the loading of this arc.

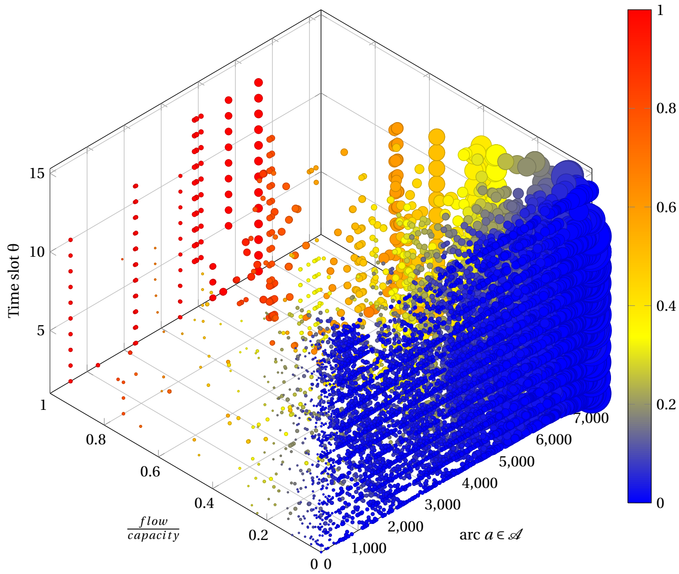


FIGURE 22: Loading of the evacuation network over time

The results of computation of flow rate and departure, travel and arrival times [7] are given respectively in the figures 23, 24, 25

4. France 37
5. By assigning each building to the nearest network node
6. Zone de Regroupement et d'Orientation

7. The departure time of a building trough a path corresponds to the departure time of the first vehicle while the arrival time corresponds to the

et 26. The three colors in these figures correspond to the K-paths (K = 3) and the size of a given point corresponds to the number of vehicles evacuated from a given building through a given path at a given time interval. The dimension j in the variables $t_{i,j,k,\theta}$ (departure time, 10 :00, 10 :30, etc.), $\tau_{i,j,k,\theta}$ (travel time in minutes) and $\lambda_{i,j,k,\theta}$ (vehicle flow rate, 0.1 minute per vehicle, 1.5 min/veh, etc.) is eliminated here since each building is associated to only one safe point.

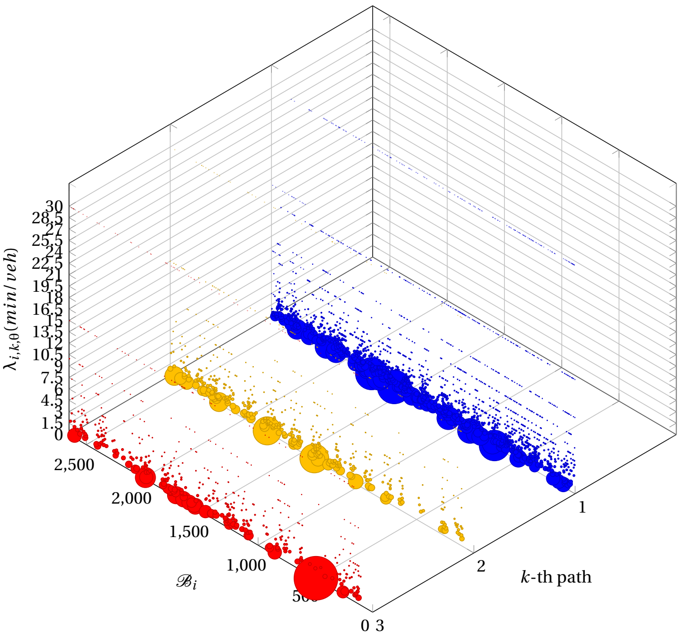


FIGURE 23: Evacuation : flow rate

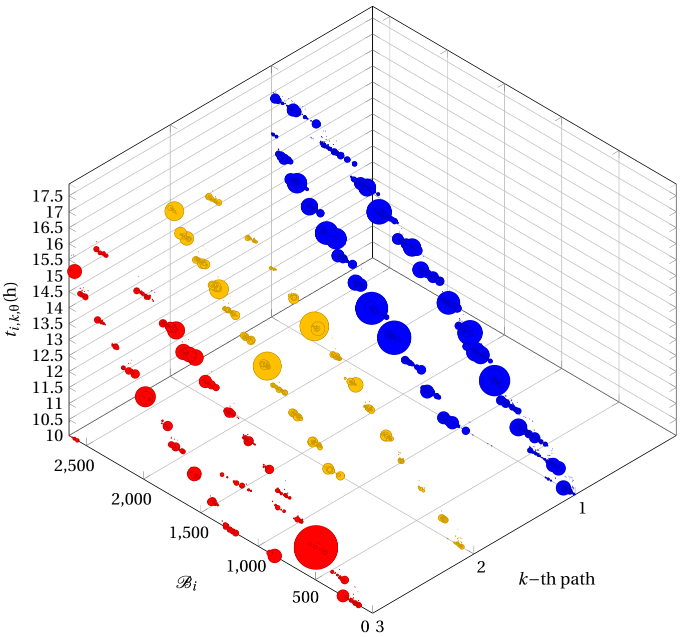


FIGURE 24: Evacuation : departure times

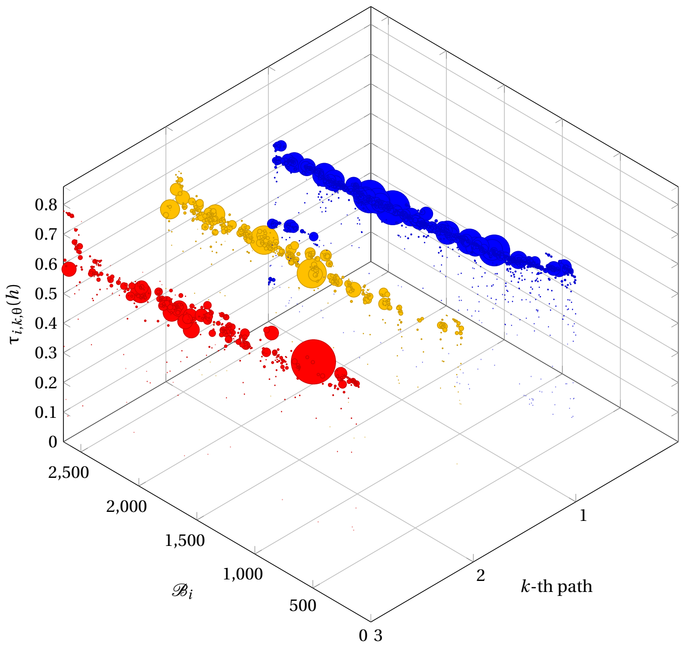


FIGURE 25: Evacuation : travel time

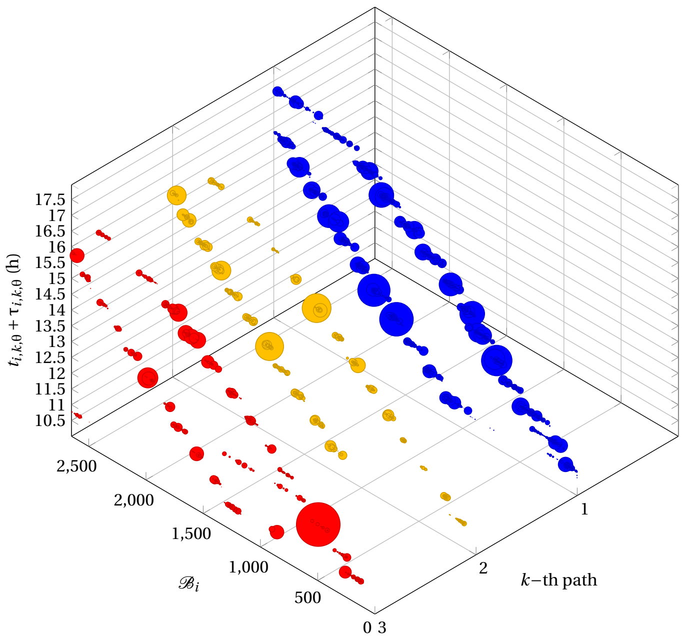


FIGURE 26: Evacuation : arrival times

We remark that flow rates of vehicles incoming in the network are almost equal and higher over time (see figure 23). Furthermore, the continuity of traffic flow is achieved over time by the VPM as shown in figures 24 and 26. We also note that the journey times on the minimum 3-paths of all buildings are very close (see figure 25).

Conclusion

Based on the foregoing, the minimization of departure dates and the continuity without congestion of traffic flow in the network, two stated objectives, are achieved by the mezoscopic pursuit vehicles model using the parallel method with decomposition (see section 3.4). It should be noted

arrival time of the last vehicle to the destination (safe point associated)

that using the sequential method (see section 3.5) on the same evacuation network (valley of Tours) provides a total evacuation time nearly double of that provided by the parallel approach with decomposition.

Acknowledgement

We would like to thank LA REGION CENTRE (France) and FEDER (Europe) for funding this research.

References

[1] Alaeddine, H., Néron, E., Serrhini, K., and Maïzia, M. (2014a). A novel polynomial algorithm for the lexicographic maximum dynamic flow problem with several sources applied to evacuation planning. *portail.scd.univ-tours.fr/iii/encore/ ?lang=eng.*

[2] Alaeddine, H., Serrhini, K., Maïzia, M., and Néron, E. (2014b). Finding the k-best paths in evacuation network. *portail.scd.univ-tours.fr/iii/encore/ ?lang=eng.*

[3] Alaeddine, H., Serrhini, K., Maïzia, M., and Néron, E. (2014c). A spatio-temporel optimization model for the evacuation of the population exposed to natural disasters. *portail.scd.univ-tours.fr/iii/encore/ ?lang=eng.*

[4] Alaeddine, H., Serrhini, K., Néron, E., and Maïzia, M. (2014d). Traffic assignment algorithms for planning a mass vehicular evacuation. *portail.scd.univ-tours.fr/iii/encore/ ?lang=eng.*

[5] Anderson, E. J., Nash, P., and Philpott, A. B. (1982). A class of continuous network flow problems. *Mathematics of Operations research*, 7 Issue 4 :501–514.

[6] Ardekani, S., Ghandehari, M., and Nepal, S. (2011). Macroscopic speed-flow models for characterization of freeway and managed lane. *The University of Texas, Arlington, USA Department of Civil Engineering.*

[7] Bretschneider, S. (2012). Mathematical models for evacuation planning in urban areas. *Lecture Notes in Economics and Mathematical Systems 659.*

[8] Burkard, R., Dlaska, K., and Klinz, B. (1993). The quikest flow problem. *ZOR-Methods and Models of Operations Research*, 37 :31–58.

[9] CAREY, M. and Subrahmanian, E. (2000). An approach to modeling time-varying flows on congested networks. *Transportation Research Part B*, 34 :58–69.

[10] Catell, K., Dinneen, M., and Fellows, M. (1994). A simple linear-time algorithm for finding paths-decompsitions of small width. *Department of Computer Science, University of Victoria, Victoria, British Columbia, V8W 3P6, Canada.*

[11] Daganzo, C. and So, S. (2011). Managing evacuation networks. *Transportation Research Part B*, 45 :1424–1432.

[12] Fleischer, L. (2001). Universally maximum flow with piecewise-constant capacities. *Networks*, 38(3) :115–125.

[13] Fleischer, L. and Skutella, M. (2003). Quickest flows over time.

[14] Fleischer, L. and Tardos, E. (1996). Efficient continuous-time dynamic network flow algorithms.

[15] Ford, L. and Fulkerson, D. (1955). Maximal flow through a network. *Canadian Journal of Mathematics*, 8 :399–404.

[16] Ford, L. and Fulkerson, D. (1958a). Constructing maximal dynamic flows from static flows. *Oper. Res*, 6 :419–433.

[17] Ford, L. and Fulkerson, D. (1958b). Constructing maximal dynamic flows from static flows. *Operations Research*, 6 :419–433.

[18] Ford, L. and Fulkerson, D. (1962). Flows in networks. *Princeton University Press, Princeton, NJ.*

[19] Gale, D. (1959). Transient flows in networks. *The Michigan Mathematical Journal*, 6 :59–63.

[20] Gartner, N., Messer, C., and Rathi, A. (2001). Traffic flow theory a state-of-the-art report. *http ://www.tft.pdx.edu/docs/revised_monograph_2001.pdf.*

[21] Hamacher, H. and Tjandra, S. (2001). Mathematical modelling of evacuation problems : A state of art. *Beritche des Fraunhofer Istitut Techno- und Wirtschaftsmathematik, Nr. 24.*

[22] Hamacher, H. and Tufekci, S. (1987). On the use of lexicographic min cost flows in evacuation modeling. *Department of Industrial and System Engineering, University of Florida, Gainesville, Florida 32611.*

[23] Hoppe, B. and Tardos, E. (1994). Polynomial time algorithms for some evacuation problems. *Proc. of 5th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 433–441.

[24] Koch, R., Nasrabadi, E., and Skutella, M. (2010). Continuous and discrete flows over time. a general model based on measure theory. *Technische universitat Berlin.*

[25] Kohler, E. (2005). Flows over time with load-dependent transit times. *SIAM J. OPTIM*, 15, No. 4 :1185–1202.

[26] Kohler, E., Langkau, K., and Skutella, M. (2002). Time-expanded graphs for flow-dependent transit times. *Technische Universitat Berlin Institut fur Mathematik, Sekr. MA 6–1, Max-Planck-Institut fur Informatik Stuhlsatzenhausweg 85, 66123 Saarbr¨ucken, Germany.*

[27] Minieka, E. (1973). Maximal, lexicographic, and dynamic network flows. *Operations Research*, 21 :517–527.

[28] Orda, A. and Rom, R. (1995). On continuous network flows. *Operations Research Letters*, 17 :27–36.

[29] Philpott, A. B. (1982). *Algorithms For Continuous Network Flow Problems.* PhD thesis, University site of Cambridge, UK.

[30] Philpott, A. B. (1990). Continuous-time flows in networks. *Mathematics of Operations Research*, 15 :No. 4.

[31] Philpott, A. B. (1994). Continuous-time shortest path problems and linear programming. *SIAM Journal on Control and Optimization*, 32 :538–552.

[32] Sherali, H., Carter, T., and Hobeika, A. (1991). A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. *Transportation Research Part B : Methodological*, 25, Issue 6 :439–452.

[33] Skutella, M. (2008). An introduction to network flows over time.

[34] Stepanov, A. and Smith, J. (2008). Multi-objective evacuation routing in transportation networks. *European Journal of Operational Research*.

[35] Wardrop, J. (1952). Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers, Part II*, 1(36) :352–378.

[36] Wilkinson, W. (1971). An algorithm for universal maximal dynamic flows in a network. *Operations Research*, 19 :1602–1612.