Computation of a \((\min,+)\) multi-dimensional convolution for end-to-end performance analyzes
Anne Bouillard, Laurent Jouhet, Eric Thierry

To cite this version:
Anne Bouillard, Laurent Jouhet, Eric Thierry. Computation of a \((\min,+)\) multi-dimensional convolution for end-to-end performance analyzes. 2008. hal-00281355

HAL Id: hal-00281355
https://hal.science/hal-00281355
Preprint submitted on 21 May 2008
Computation of a \((\min,+)\) multi-dimensional convolution for end-to-end performance analysis.

Anne Bouillard  
ENS Cachan / IRISA  
Campus de Beaulieu  
35000 Rennes, France  
Anne.Bouillard@irisa.fr

Laurent Jouhet  
ENS Lyon / IXXI  
46 Allée d’Italie  
69007 Lyon, France  
Laurent.Jouhet@ens-lyon.fr

Eric Thierry  
LIAFA & ENS Lyon / IXXI  
46 Allée d’Italie  
69007 Lyon, France  
Eric.Thierry@ens-lyon.fr

ABSTRACT

Network Calculus is an attractive theory to derive deterministic bounds on end-to-end performance measures. Nevertheless bounding tightly and quickly the worst-case delay or backlog of a flow over a path with cross-traffic remains a challenging problem.

This paper carries on with the study of configurations where a main flow encounters some cross-traffic flows which interfere over connected sub-paths. We also assume that no information is available about scheduling policies at the nodes (blind multiplexing). Such configurations were first analyzed in \([25, 27]\) where a “Pay Multiplexing Only Once” (PMOO) phenomenon was identified, and then in \([6, 7]\) where a \((\min,+)\) multi-dimensional operator was introduced to compute a minimum service curve for the whole path. Under usual assumptions (convex arrival curves and convex service curves), we prove some properties of this new operator and we show how to use it to derive bounds on delays and backlogs in polynomial time.

We also discuss the simpler case when there is no cross-traffic. Then the analysis is known to boil down to the \((\min,+)\) convolution of all the service curves over the path. If we want to take into account the interference due to the cross-traffic and not only the delays or backlogs, we need to consider more general operators such as \((\min,+)\) or \((\max,+)\) and prove that the complexity of computing performances bounds in this case is infinite.

1. INTRODUCTION

Network Calculus can be presented as a deterministic queuing theory based on the \((\min,+)\) semi-ring, and aimed at worst-case performance analyzes in communication networks. From a mathematical point of view, it consists in combining curves which locally describe the shape of the traffic and the services, with \((\min,+)\) or \((\max,+)\) operations in order to predict the global behavior of the networks, and in particular end-to-end measures. This theory has managed to gather several results of the performance evaluation literature within a common framework and it has yielded interesting new results about QoS in networks \([12, 13, 9, 5]\). Originally intended for Internet quality of service \([14]\), its use seems promising for other types of networks: embedded systems like networks processors \([30]\), switched Ethernet networks \([15]\), sensor networks \([24, 28]\). Besides those achievements and potential applications, some mathematical and algorithmic issues remain open. One main issue is the difficulty to analyze networks where several flows are multiplexed at some nodes. Such configurations also involve the type of scheduling policy used to serve data from those different flows when they cross. As an example, the complexity of deciding the stability of a network (i.e. checking that the amount of data backlogged in the network will never grow to infinite) is still open for simple Network Calculus constraints and FIFO policy everywhere \([1, 5]\). Even for feed-forward networks where a simple criteria of stability is known \([5]\), providing quickly tight bounds on end-to-end measures like delays is very challenging in Network Calculus.

The analysis of one flow which encounters some cross-traffic on its path is one basic issue that has been addressed with various assumptions. For FIFO multiplexing at the nodes of the path (which are the servers), configurations with arbitrary cross-traffic have been studied in \([3, 10, 16]\) where bounds on end-to-end delays can be derived only for small utilization factors. The reader is also referred to \([18, 19, 20]\) for the latest results about FIFO multiplexing: tight bounds are provided for configurations where each cross-traffic flow interferes over a sub-path which is a suffix of the main path. Our work rather lies in the framework of blind multiplexing, that is nothing is known about multiplexing policies at the nodes. Despite this lack of information, one can provide bounds on end-to-end performance measures like delays or backlogs. The analysis of a path reduced to a single node with cross-traffic and under blind multiplexing can be found in \([5]\). This analysis was extended to configurations where cross-traffic flows interfere over connected sub-paths \([25, 27]\). Those authors identified a phenomenon called “Pay Multiplexing Only Once” (PMOO). To give a short description of this phenomenon, let us say that when the routed flow merges with some cross-traffic flows, its service may be strongly reduced at the first node. However at the next nodes, the interference due to the cross-traffic cannot be as severe since the competition for the resource...
The second part of the paper (Section 4) focuses on paths without any cross-traffic. Such configurations represent the brand image of Network Calculus: it is well-known that one can analyze the performances by replacing the whole path by a single node offering as minimum service curve the convolution of all the service curves of the initial path. It captures for instance the famous "Pay Burst Only Once" phenomenon [5]. When the initial service curves are convex and piecewise affine with a finite number of pieces, the convolution comes to concatenating all the pieces of the different curves sorted by non-decreasing slopes (up to removing a few pieces at the end). This theorem stated in [5] has been used by various authors [6, 8, 17, 21, 22, 25, 27]. Unfortunately the proof presented in [5] is incomplete (Remark 2). Consequently we present three different proofs of this theorem (Theorem 5) which illustrate different types of tools used by various authors [6, 8, 17, 21, 22, 25, 27]. Unfortunately the algorithm had a flaw explained in [7].

Service curves. Consider $B$ the cumulative departure function of the flow, i.e. the amount of data $B(t)$ that has left the system until time $t$, with $B(0) = 0$. A function $\alpha \in \mathcal{F}$ is an arrival curve for $A$ if $\forall s,t \in \mathbb{R}_+$, $s \leq t$, we have $A(t) - \alpha(s) \leq \alpha(t-s)$. It means that the amount of data arriving between time $s$ and $t$ is at most $\alpha(t-s)$. An example of arrival curve is the affine function $\alpha(t) = \sigma + \rho t$, $\sigma,\rho \in \mathbb{R}_+$ (sometimes called leaky-bucket arrival curve). In this case, $\sigma$ can be interpreted as the maximal number of bits that can arrive simultaneously (maximal burst) and $\rho$ the maximal long-term rate of arrivals.

Convex/concave/piecewise affine assumptions. In this paper, we will often assume that the input curves are piecewise affine with a finite number of pieces, and convex or concave. These are recurrent assumptions in Network Calculus [6, 7, 9, 5, 25, 30] due to the fact that such functions can be easily generated from combinations of linear functions and that convex/concave properties are often preserved by the Network Calculus operations.

Note that since we consider functions from $\mathbb{R}_+$ into $\mathbb{R}_{\text{min}}$, our definition of a convex (resp. concave) function $f$ is that its support (the set where it is finite) is $\mathbb{R}_+$, or an interval $[0,T]$, $T \in \mathbb{R}_+$, and $f$ is convex on its support. In the same way, a function $f$ from $\mathbb{R}_+$ into $\mathbb{R}_{\text{min}}$ is piecewise affine if its support (the set where it is finite) is $\mathbb{R}_+$, or an interval $[0,T]$, $T \in \mathbb{R}_+$, and $f$ is piecewise affine on its support. The number of affine pieces of $f$ is denoted $|f|$.

Bounds on delays and backlogs. Bounds on worst-case backlogs and delays can be easily derived from the Network Calculus constraints.

**Definition 1.** Let $A$ be the cumulative arrival function of a flow entering a system and let $B$ be its corresponding cumulative departure function. Then the backlog of the flow at time $t$ is

$$b(t) \triangleq A(t) - B(t)$$

and the delay (assuming FIFO order when serving data of

- Convolution: $(f * g)(t) = \inf_{0 \leq s \leq t} (f(s) + g(t-s))$.
- Deconvolution: $(f \circ g)(t) = \sup_{s \geq 0} (f(t+u) - g(u))$.

**Arrival curves.** Given a data flow traversing a system, let $A \in \mathcal{F}$ be its cumulative arrival function, i.e. $A(t)$ is the total amount of data that has arrived in the system until time $t$, with $A(0) = 0$. A function $\alpha \in \mathcal{F}$ is an arrival curve for $A$ if $\forall s,t \in \mathbb{R}_+$, $0 \leq s \leq t$, we have $A(t) - \alpha(s) \leq \alpha(t-s)$. It means that the amount of data arriving between time $s$ and $t$ is at most $\alpha(t-s)$. An example of arrival curve is the affine function $\alpha(t) = \sigma + \rho t$, $\sigma,\rho \in \mathbb{R}_+$ (sometimes called leaky-bucket arrival curve). In this case, $\sigma$ can be interpreted as the maximal number of bits that can arrive simultaneously (maximal burst) and $\rho$ the maximal long-term rate of arrivals.

**Convergence.** Given a data flow traversing a system, let $A \in \mathcal{F}$ be its cumulative arrival function, i.e. $A(t)$ is the total amount of data that has arrived in the system until time $t$, with $A(0) = 0$. A function $\alpha \in \mathcal{F}$ is an arrival curve for $A$ if $\forall s,t \in \mathbb{R}_+$, $0 \leq s \leq t$, we have $A(t) - \alpha(s) \leq \alpha(t-s)$. It means that the amount of data arriving between time $s$ and $t$ is at most $\alpha(t-s)$. An example of arrival curve is the affine function $\alpha(t) = \sigma + \rho t$, $\sigma,\rho \in \mathbb{R}_+$ (sometimes called leaky-bucket arrival curve). In this case, $\sigma$ can be interpreted as the maximal number of bits that can arrive simultaneously (maximal burst) and $\rho$ the maximal long-term rate of arrivals.

**Service curves.** Consider $B$ the cumulative departure function of the flow, i.e. the amount of data $B(t)$ that has left the system until time $t$, with $B(0) = 0$. The system is said to provide a (minimum) service curve $\beta \in \mathcal{F}$ if $B \geq A \circ \beta$. A strict service curve $\beta$ for the system is a non-negative function in $\mathcal{F}$ such that during any backlogged period of duration $u$, at least $\beta(u)$ data is served. More formally, for any $s,t \in \mathbb{R}_+$, $s \leq t$, if $\forall s < v < t$, $A(v) - B(v) > 0$, then $B(t) \geq B(s) + \beta(t-s)$. A strict service curve is clearly always a service curve for the system (by considering starts of backlogged periods), but the converse is not true [5].

Note that although it may seem unusual or hard to interpret, nothing prevents from using service curves and strict service curves which fail to be non-decreasing or non-negative.

**Convergence.** Given a data flow traversing a system, let $A \in \mathcal{F}$ be its cumulative arrival function, i.e. $A(t)$ is the total amount of data that has arrived in the system until time $t$, with $A(0) = 0$. A function $\alpha \in \mathcal{F}$ is an arrival curve for $A$ if $\forall s,t \in \mathbb{R}_+$, $0 \leq s \leq t$, we have $A(t) - \alpha(s) \leq \alpha(t-s)$. It means that the amount of data arriving between time $s$ and $t$ is at most $\alpha(t-s)$. An example of arrival curve is the affine function $\alpha(t) = \sigma + \rho t$, $\sigma,\rho \in \mathbb{R}_+$ (sometimes called leaky-bucket arrival curve). In this case, $\sigma$ can be interpreted as the maximal number of bits that can arrive simultaneously (maximal burst) and $\rho$ the maximal long-term rate of arrivals.

**Service curves.** Consider $B$ the cumulative departure function of the flow, i.e. the amount of data $B(t)$ that has left the system until time $t$, with $B(0) = 0$. The system is said to provide a (minimum) service curve $\beta \in \mathcal{F}$ if $B \geq A \circ \beta$. A strict service curve $\beta$ for the system is a non-negative function in $\mathcal{F}$ such that during any backlogged period of duration $u$, at least $\beta(u)$ data is served. More formally, for any $s,t \in \mathbb{R}_+$, $s \leq t$, if $\forall s < v < t$, $A(v) - B(v) > 0$, then $B(t) \geq B(s) + \beta(t-s)$. A strict service curve is clearly always a service curve for the system (by considering starts of backlogged periods), but the converse is not true [5].

Note that although it may seem unusual or hard to interpret, nothing prevents from using service curves and strict service curves which fail to be non-decreasing or non-negative.

**Convergence.** Given a data flow traversing a system, let $A \in \mathcal{F}$ be its cumulative arrival function, i.e. $A(t)$ is the total amount of data that has arrived in the system until time $t$, with $A(0) = 0$. A function $\alpha \in \mathcal{F}$ is an arrival curve for $A$ if $\forall s,t \in \mathbb{R}_+$, $0 \leq s \leq t$, we have $A(t) - \alpha(s) \leq \alpha(t-s)$. It means that the amount of data arriving between time $s$ and $t$ is at most $\alpha(t-s)$. An example of arrival curve is the affine function $\alpha(t) = \sigma + \rho t$, $\sigma,\rho \in \mathbb{R}_+$ (sometimes called leaky-bucket arrival curve). In this case, $\sigma$ can be interpreted as the maximal number of bits that can arrive simultaneously (maximal burst) and $\rho$ the maximal long-term rate of arrivals.
the flow) at time $t$ is

$$d(t) \overset{\text{def}}{=} \inf\{s \geq 0 \mid A(t) \leq B(t + s)\}.$$ 

Given an arrival curve and a service curve, it is possible to compute with the Network Calculus operations an upper bound on the maximal backlog (resp. delay). Moreover, one can also compute the arrival curve of the departure flow.

Theorem 1 ([5, 9]). Let $A \in \mathcal{F}$ be the cumulative arrival function with an arrival curve $\alpha$ for a flow entering a system with service curve $\beta$. Let $B \in \mathcal{F}$ be the cumulative departure function. Then,

1. $B$ has an arrival curve $\alpha \odot \beta$.

2. $b(t) \leq B_{\max}(\alpha, \beta) \overset{\text{def}}{=} \sup\{\alpha(t) - \beta(t) \mid t \geq 0\} = (\alpha \odot \beta)(0)$.

3. $d(t) \leq D_{\max}(\alpha, \beta) \overset{\text{def}}{=} \inf\{d \geq 0 \mid \forall t \geq 0, \alpha(t) \leq \beta(t + d)\}^2 = \inf\{d \geq 0 \mid (-\beta) \odot (-\alpha)(d) \leq 0\}$.

As illustrated on Fig. 1, the backlog bound $B_{\max}(\alpha, \beta)$ is the maximal vertical distance between $\alpha$ and $\beta$ while the delay bound $D_{\max}(\alpha, \beta)$ is given by the maximal horizontal distance between those two functions.

Figure 1: Bounds on worst-case backlog and delay.

2.2 Quantifying the “Pay Multiplexing Only Once” phenomenon

There exists an interesting formula which gives a end-to-end service curve for a path in case the cross-traffic flows interfere over sub-paths, i.e. sets of consecutive nodes. It generalizes a formula obtained in [27] for a small example with three nodes and two cross-traffic flows. Here are the configurations we study (and called PMOO) and the notation we use, illustrated by Fig. 2:

PMOO Configurations:

- The main flow $F_0$ follows a path $p$ of $n$ nodes indexed $1, 2, \ldots, n$ with respective service curves $\beta_i$, $1 \leq i \leq n$, which can be strict or not.

- The cross-traffic interfering with the path is composed of $k$ flows $F_i$, $1 \leq i \leq k$, with respective arrival curve $\alpha_i$.

- Each cross-traffic flow $F_i$ intersects the path $p$ over a set of consecutive nodes of $p$. It lines the path at node $s_i$, and leaves it just after node $e_i$ (in particular $s_0 = 1$ and $e_n = n$).

- Consider a flow $i$, $1 \leq i \leq k$, the amount of data that have been served by node $j$ until time $t$ is denoted $A_i^{(j)}(t)$. The amount of data that have entered node $s_i$ until time $t$ is denoted $A_i^{(s_i)}(t)$.

- As in the rest of the paper, we assume blind multiplexing at the nodes: the scheduling policy applied to the input flows entering a node is unknown (we only assume that, for each flow, it is FIFO with regard to data from this flow).

Figure 2: Path where each cross-traffic flow $i$ interferes over a sub-path.

The next theorem is proved in [6] with the assumption that all the service curves $\beta_i$ are strict. A careful look at the proof enables to rewrite the theorem as follows (it emphasizes the role of the strict service curve assumption):

Theorem 2 (PMOO multi-dimensional operator [6]).

For a PMOO configuration, a service curve offered by the whole path $p$ to flow $F_0$ is:

$$\psi(t) = \inf_{u_1, \ldots, u_n \geq 0} \sum_{j=1}^{n} \beta_j(u_j) - \sum_{j=1}^{k} \sum_{s_i \leq t} \alpha_i(\sum_{j=x_i}^{e_i} u_j).$$

If all the service curves $\beta_i$ are strict, a service curve offered by the whole path is $\phi(t) = \psi(t)_+$, where $x_+ \overset{\text{def}}{=} \max(x, 0)$.

Two classical results of Network Calculus can be seen as corollaries of this theorem. The first one corresponds to $n = 2$ and no cross traffic, and the second one to $n = k = 1$.

Corollary 1 (Tandem [5, 9]). Consider a flow crossing two nodes in tandem with respective service curves $\beta_1$ and $\beta_2$. Then the concatenation of the two nodes offers a minimum service curve $\beta_1 \odot \beta_2$ to the flow.

Corollary 2 (Residual service [5, 9]). Consider a node offering a strict service curve $\beta$ and two flows entering that server, with respective arrival curves $\alpha_1$ and $\alpha_2$. Then a service curve for flow 1 is $\beta_1 = (\beta - \alpha_2)_+$. 

---

1 with the convention $(+\infty) - (+\infty) = -\infty$

2 with the convention $\inf \emptyset = +\infty$
3. PERFORMANCE ANALYSIS WITH A MULTI-DIMENSIONAL CONVOLUTION

3.1 Multi-dimensional convolution

Definition 2. Let \( J = \{1, \ldots, n\} \) and \( I = \{1, \ldots, k\} \), Let \( \{f_i\}_{i \in I} \) be a family of functions in \( \mathcal{F} \), and \( \{J_i\}_{i \in I} \) be a family of subsets of \( J \). The multi-dimensional convolution associated with those families is the function \( \psi \in \mathcal{F} \) defined as:

\[
\psi(t) = \min_{u_1, \ldots, u_n \geq 0} \sum_{i \in I} f_i(x) \quad \text{for all} \quad t = x \text{ such that } u_1 + \cdots + u_n = t.
\]

Theorem 3. With the notation of Definition 2, if all the functions \( f_i \) are convex, then \( \psi \) is also convex. Moreover if they are convex and piecewise affine with a finite number of pieces, then \( \psi \) is a convex piecewise affine function which can be computed in exponential time.

Proof. For all \( j \in J \), we will denote \( I_j = \{i \in I \mid j \in J_i\} \). We can suppose w.l.o.g. that for all \( 1 \leq i \leq k \), \( f_i(0) = 0 \). Otherwise it is sufficient to notice that \( \psi \) is equal to the same formula with \( f_i(x) = f_i(x) - f_i(0) \) plus the constant \( \sum_{i=1}^k f_i(0) \). Then we may also suppose w.l.o.g. that all functions are non-decreasing. If some functions \( f_i \) are not non-decreasing, let \( \lambda < 0 \) be the smallest slope among them. For all \( t \in \mathbb{R}_+ \), it can be checked that

\[
\psi(t) = kM + \min_{u_1, \ldots, u_n \geq 0} \sum_{i \in I} f_i(x) \quad \text{for all} \quad t = x \text{ such that } u_1 + \cdots + u_n = t.
\]

In this way, up to adding the negative linear function \( t \to kM \), \( \psi \) is the multi-dimensional convolution of \( n + k \) non-decreasing convex piecewise functions. Note that those transformations yield new instances which can be stored in linear space with regard to the initial instance.

Now for each function \( f_i \), \( 1 \leq i \leq k \), with \( \ell_i = |f_i| \) pieces on its support, let \( r_{i,\ell} \) (resp. \( \tau_{i,\ell} \)) be the slope (resp. the horizontal length, possibly \(+\infty\)) of the \( \ell \)-th piece starting from the left. Since all \( f_i \)'s are convex and non-decreasing, for a fixed \( t \in \mathbb{R}_+ \), the value \( \psi(t) \) is exactly the solution of a linear programming instance (the primal one). It uses the non-negative variables \( v_{i,\ell} \), \( 1 \leq i \leq k \), \( 1 \leq \ell \leq \ell_i \), for each \( 1 \leq i \leq k \). It can be easily checked that the value \( \psi(t) \) is equal to \( \inf \sum_{i=1}^k \sum_{\ell=1}^{\ell_i} r_{i,\ell} v_{i,\ell} \varphi \), where \( \varphi \) is any feasible solution of the linear constraints:

\[
\sum_{\ell=1}^{\ell_i} v_{i,\ell} \leq u_i, \quad \varphi(0) = 0, \quad \varphi(t) = \psi(t) + kM, \quad \text{for all} \quad t \geq 0.
\]

We first wish to find the support of \( \psi \); we only have to replace the objective function by \( \sup t \) with the same constraints, the solution \( T \) (possibly \(+\infty\)) can be found in polynomial time by solving this linear programming instance. Whenever \( t > T \) we know that \( \psi(t) = +\infty \).

Now the Strong Duality theorem for linear programming [29] ensures that, when it is finite, \( \psi(t) \) is also equal to the optimum of the dual instance defined for the non-negative variables \( y \), \( (y_i)_{1 \leq i \leq k} \) and \( (y_{i,\ell})_{1 \leq \ell \leq \ell_i} \). The value \( \psi(t) \) is equal to \( \sup yt - \sum_{i=1}^k \sum_{\ell=1}^{\ell_i} y_{i,\ell} r_{i,\ell} \) under the constraints:

\[
\begin{align*}
y_i &\leq y_{i,\ell} + r_{i,\ell} t \\
y_{i,\ell} &\leq \sum_{\ell'=1}^{\ell_i} y_{i,\ell'} \\
\end{align*}
\]

for all \( 1 \leq i \leq k \), \( 1 \leq \ell \leq \ell_i \), \( 1 \leq \ell \leq \psi \), \( 1 \leq t \leq n \). Those \( c = n + \sum_{i=1}^k \ell_i \) linear constraints over \( d = c + k + 1 \) variables define a convex polyhedron in \( \mathbb{R}^d_+ \), which is independent of \( t \) (note that this polyhedron is not bounded). We already know the support of \( \psi \); thus we are only interested in finite values of \( \psi \). It is known that the supremum (if finite) is necessarily reached at an extremal point of the polyhedron in \( \mathbb{R}^d_+ \). Moreover the set of extremal points is finite, its cardinal can be exponential in \( d \), and this set can be computed in exponential time in \( d [4, 29] \). Each extremal point \( (y_i, (y_{i,\ell})_{1 \leq i \leq k}, (y_{i,\ell})_{1 \leq i \leq k}) \in \mathbb{R}^d_+ \) defines an affine function \( t \to yt - \sum_{i=1}^k \sum_{\ell=1}^{\ell_i} y_{i,\ell} r_{i,\ell} \) and over its support, \( \psi(t) \) is the supremum of this finite set of affine functions. Consequently \( \psi \) is a convex piecewise affine function which can be computed in exponential time in \( d \).

For now, when the functions \( f_i \) are convex and piecewise affine, if one wishes to compute and store the whole function \( \psi \), we only have an exponential algorithm except from very specific cases where a polynomial algorithm is known:

- \( J_i = J \) for all \( i \) (it comes to the addition of convex piecewise affine functions).
- \( I = J \) and \( J_i = \{1\} \) for all \( i \) (it comes to the classical convolution - see Section 4).

Nevertheless we conjecture that it is possible to compute the whole function \( \psi \) in polynomial time (which implies that the number of pieces of \( \psi \) is polynomial with the size of the input functions). As a matter of fact, we can already compute, in polynomial time, worst-case bounds on delays and backlogs using \( \psi \) but without computing explicitly the whole curve, as shown in the next section.

3.2 Bounds on delays and backlogs

Theorem 4. In a PMOO configuration, if all the service (resp. arrival) curves \( \beta_j \) (resp. \( \alpha_i \), including \( \alpha \) the arrival curve of the main flow \( F_0 \)) are convex (resp. concave) piecewise affine functions with a finite number of pieces, then the worst-case bounds \( D_{\text{max}}(\alpha, \psi) \) and \( B_{\text{max}}(\alpha, \psi) \) for the main flow \( F_0 \) can be computed in polynomial time by solving one linear programming instance with \( O(\sum_{i=1}^k |\alpha_i| + \sum_{j=1}^d |\beta_j|) \) linear constraints over \( O(\sum_{i=1}^k |\alpha_i| + \sum_{j=1}^d |\beta_j|) \) variables (replace \( \psi \) by \( \varphi = \psi_+ \), if service curves are strict).

Proof. Consider \( C(\alpha, \psi) \equiv \{(t, z) \in \mathbb{R}_+ \times \mathbb{R} \mid \psi(t) \leq z \leq \alpha(t)\} \) the area between the arrival curve \( \alpha \) and the end-to-end service curve \( \psi \) defined by Theorem 2. This function \( \psi \) is a particular case of the multi-dimensional convolution and due to the hypotheses on \( \alpha_i \) and \( \beta_j \), Theorem 3 ensures that \( \psi \) is convex (as well as \( \psi_+ \)). Since \( \alpha \) is concave, \( C(\alpha, \psi) \) is a convex set. As a matter of fact, let \( \alpha(t) \equiv \min_{1 \leq i \leq k} (\alpha_i(t) + b_i) \), then \( (t, z) \in C(\alpha, \psi) \) if and only if \( z \leq \alpha(t) + b \), for all \( 1 \leq p \leq d \) and \( (t, z) \) satisfies the constraints of the primal linear programming...
Let \( f, g \in cF \) be two convex functions. As convex functions, \( f \) and \( g \) are continuous and admit a left and a right derivative. For a function \( f \) admitting a derivative on the left (resp. right),

\[
\text{First proof (Local differentiation). Let } f, g \in cF \text{ be two convex functions. As convex functions, } f \text{ and } g \text{ are continuous and admit a left and a right derivative. For a function } f \text{ admitting a derivative on the left (resp. right), let us denote by } f'_L \text{ (resp. } f'_R) \text{ the left (resp. right) derivative function of } f. \text{ If } f \text{ is convex, } f'_L \text{ and } f'_R \text{ are non-decreasing and } \forall u \in \mathbb{R}_+, f'_L(u) \leq f'_R(u) \text{ and are respectively left and right-continuous. By convention, one set } f'_L(0) = g'_L(0) = -\infty.

The next two lemmas apply to arbitrary convex functions in } F.

**LEMMA 1 (Folklore).** Let \( f \) be a convex function, \( \forall u, v \in \mathbb{R}_+, \) with \( u < v, \) one has

\[
f'_L(u) \leq \frac{f(v) - f(u)}{v - u} \leq f'_R(v).
\]

**LEMMA 2.** Let \( f, g \in F \) be two convex functions. Then, for all \( t \in \mathbb{R}_+ \) and \( u, v \in \mathbb{R}_+ \) s.t. \( u + v = t, \) the next two points are equivalent:

- \( f \ast g \) is convex,
- \( f \ast g \) is continuous on the left (resp. right).

![Figure 3: Convolution of convex piecewise affine functions.](image)

For all \( t \in \mathbb{R}_+, \) there always exist such \( u, v \in \mathbb{R}_+. \)

**Proof.** Since \( f \) and \( g \) are continuous and \( f \ast g \) is an infimum over the compact set \([0, t]\), the existence of \( u \) and \( v \) stated at the end is proved.

Suppose that one can find \( u \) and \( v \) such that \( u + v = t, \) \( g'_L \leq f'_L(u) \) and \( f'_R(u) \leq g'_R(v) \). Consider \( u' \) and \( v' \) such that \( f \ast g(t) = f(u') + g(v') \) with \( u' + v' = t \) (then \( u - u' = v' - v \)). Without loss of generality, one can suppose that \( u' \leq u. \) Then,

\[
f(u') + g(v') = f(u') + [f(u) - f(u')] + g(v') + [(g(v) - g(v'))] = f(u') + g(v') + (u - u') [\frac{f(u) - f(u')}{u - u'} - \frac{g(v) - g(v')}{v' - v}]
\]

\[
\leq f(u') + g(v') + (u - u') [f'_L(u) - g'_R(v')]
\]

\[
\leq f(u') + g(v').
\]

As a consequence, \( f(u) + g(v) = f \ast g(t) \).

Conversely, let \( f \ast g(t) = f(u) + g(v), \) \( u + v = t. \) Then

\[
f(u) + g(v) \leq \inf_{0 < \epsilon < t} f(u - \epsilon) + g(v + \epsilon). \]

It implies \( 0 \leq \inf_{\epsilon > 0} [-f(u) + f(u - \epsilon) + g(v + \epsilon) - g(v)]. \) Due to convexity, \( g(v + \epsilon) - g(v) \leq \epsilon g'_L(v + \epsilon) \) and \( f(u) - f(u - \epsilon) \geq \epsilon f'_R(u - \epsilon). \) Thus \( f'_L(u - \epsilon) \leq g'_R(v + \epsilon). \) The function \( f'_L \) (resp. \( g'_R) \) is continuous on the left (resp. on the right). By
letting $\varepsilon$ go to 0, we get $f'_t(u) \leq g'_t(v)$. In a symmetric way, we can prove $g'_t(v) \leq f'_t(u)$.

We now apply this theorem to piecewise affine functions. Let $t \in \mathbb{R}_+$ and $u$ and $v$ such that $f \ast g(t) = f(u) + g(v)$. We look at continuity of the derivative of $f$ and $g$ at those points.

- If $f'$ exists at $u$ and is continuous on $[u, u + \varepsilon]$, then $f'(u) = f'_t(u + \varepsilon)$ and $f(u + \varepsilon) \leq f'_t(u) + f'_t(u + \varepsilon)$. So, $f \ast g(t + \varepsilon) = f(u + \varepsilon) + g(v)$ and $f \ast g$ is affine of slope $f'(u)$ on $[t, t + \varepsilon]$. One can chose $\varepsilon$ such that there is a change of slope in $f$ at $u + \varepsilon$.

- The same holds for $g$ and $v$ if $g'$ is continuous on an interval $[v, v + \varepsilon]$ by symmetry.

- If $f'$ and $g'$ don’t exist at $u$ and $v$, and if $f'_u(u) \leq g'_u(v)$ (the role of $f$ and $g$ are symmetric), then $f_t(u + \varepsilon) = f'_t(u) \leq g'_t(v)$ and $g'_t(u) \leq f'_t(u) = f'_t(u + \varepsilon)$. So $f \ast g(t) = f(u + \varepsilon) + g(v)$ and the slope of $f \ast g$ after $t$ is $f'_t(u)$. This holds for any $\varepsilon$ smaller that the length of the segment of slope $f'_t(u)$ in $f$. If this segment is of infinite length, it holds for every $\varepsilon \in \mathbb{R}_+$.

The three items show that one can construct $f \ast g$ by letting both $u$ and $v$ grow (sometimes alternatively). The first two items show that the complete pieces of $f$ and $g$ appear in the function $f \ast g$. The third item indicates the rule of concatenation: the piece of smallest slope appears first.

Second proof (Algorithmic proof). Consider the first segment of $f$, of slope $s_f$ and the first segment of $g$ of slope $s_g$ and suppose that $s_f \leq s_g$ and that the length of the first segment of $f$ is $\ell_f \in [0, \infty)$. Then, for every $t \in [0, \ell_f]$, we have:

$$f \ast g(t) = \inf_{u + v = t, u, v \geq 0} f(u) + g(v)$$

$$= \inf_{u + v = t, u, v \geq 0} f(0) + s_f u + g(0) + v \frac{g(v) - g(0)}{v - 0}.$$

As $g$ is convex, $\frac{g(v) - g(0)}{v - 0} \geq s_g \geq s_f$ so $f \ast g(t) = f(0) + f(t)$.

If $\ell_f = \infty$, then $f \ast g$ is an affine function consisting in the segment of $f$ and $g$ smallest slope.

Otherwise, $f \ast g$ consist of the segment of smallest slope on $[0, \ell_f]$.

Let $t > \ell_f$ and suppose that $f \ast g(t) = f(u) + g(v)$, $u + v = t$ and $u < \ell_f$.

$$f(\ell_f) + g(t - \ell_f) = [f(\ell_f) - f(u)] + f(u) + [g(t - \ell_f) - g(v)] + g(v)$$

$$= f(u) + g(v) + s_f (\ell_f - u) - (\ell_f - u) \frac{g(v) - g(0)}{v - 0}.$$
quence \((r_i)_{1 \leq i \leq n}\) is increasing to locate the value \(t\) for which the supremum is reached.

From Property (LF1) and Lemma 3, for all \(\lambda \in \mathbb{R}_{+}\), \(\int f \circ g(\lambda) = f(\lambda) + g(\lambda) = \sum_{(r_\rho, h_\rho) \in C_{\rho}} (\lambda - r_\rho) + \tau + \sum_{(r_\psi, h_\psi) \in C_{\psi}} (\lambda - r_\psi) + \tau\) where \(C_f\) (resp. \(C_g\)) contains the couples (slope, horizontal length) of all the pieces of \(f\) (resp. \(g\)).

We also know that \(f \circ g\) is convex. This is true for any convex functions \(f, g \in \mathcal{F}\) (even if not piecewise affine) since \(f \circ g\) is the infimum of a convex function over a convex set [23].

We now introduce \(h\) the concatenation from point \((0, 0)\) of all the pieces of \(f\) and \(g\), sorted by non-decreasing slopes. If \(f\) or \(g\) has a semi-infinite piece, that is \((\rho_f, +\infty) \in C_f\) or \((\rho_g, +\infty) \in C_g\), then we ignore the pieces with slopes larger than \(\rho_f\) or \(\rho_g\) and \(h\) ends by the semi-infinite piece of smallest slope (denoted \(\rho_h\)). Since \(h\) sorts the pieces by non-decreasing slopes \(h\) is convex by construction and Lemma 3 applies: for all \(\lambda \in \mathbb{R}_{+}\), \(\hat{h}(\lambda) = \sum_{(r_\rho, h_\rho) \in C_{\rho}} (\lambda - r_\rho) + \tau\) where \(C_h\) denotes the set of the couples (slope, horizontal length) of the pieces of \(h\).

One can easily check that, for all \(\lambda \in \mathbb{R}_{+}\), \(\int f \circ g(\lambda) = \hat{h}(\lambda)\). If \(f\) and \(g\) have no semi-infinite, \(C_h \subseteq C_f \cup C_g\). Otherwise \(C_h \subseteq C_f \cup C_g\), but the terms which only appear in the formula of \(f \circ g(\lambda)\) do not play any role: due to their form \((\lambda - r_\rho) + \tau\) with \(r > \rho_h\), they can be taken up only for \(\lambda > \rho_h\), but both \(f \circ g\) and \(\hat{h}\) contain the term \((\lambda - \rho_h) + \tau\) which is equal \(+\infty\) for such \(\lambda\).

To sum up, \(f \circ g\) and \(h\) are both convex functions of \(\mathcal{F}^+\) and \(f \circ g = \hat{h}\). Since the Legendre-Fenchel transform is injective on \(\mathcal{F}^+\), we have \(f \circ g = h\).

Remark 2. Theorem 5 is presented in [5] (Theorem 3.1.6 rule 9, pages 113-115). However the proof for convex piecewise affine functions is incomplete. Two functions are defined from the convex functions \(f\) and \(g\): \(h \overset{\text{def}}{=} f \circ g\) and \(h'\) which is the concatenation from \(f(0) + g(0)\) of all the pieces of \(f\) and \(g\) sorted by non-decreasing slopes. The proof defines the respective epigraphs \(\Sigma_1, \Sigma_2, \Sigma\) of \(f, g, h, h'\) (and their boundaries \(\partial \Sigma_1, \partial \Sigma_2, \partial \Sigma, \partial \Sigma')\). It is shown that \(\partial \Sigma' \subseteq \partial \Sigma_1 + \partial \Sigma_2\). It implies that \(h' \geq f \circ g\), but unfortunately it does not prove the inverse inequality. The proof that \(\partial \Sigma' \subseteq \partial \Sigma_1 + \partial \Sigma_2\) only uses the fact that \(h'\) is a prefix of a shuffle of the two sequences of pieces of \(f\) and \(g\) (seen as two words). It does not use the sorting of slopes. Although at some point it is noted that \(h'\) is convex by construction, this property is not used later.

4.2 Bounds on delays and backlogs

Theorem 6. In the simple PMSO configurations where there is no cross-traffic, a service curve for the whole path is \(\beta = \beta_1 \ast \beta_2 \ast \cdots \beta_n\). If all the service curves are convex piecewise affine functions with a finite number of pieces, the size of \(\nu\) satisfies \(|\nu| \leq \sum_{\ell=1}^n |\beta_\ell|\) and \(\nu\) can be computed in \(O(\log n \sum_{\ell=1}^n |\beta_\ell|)\) time.

Given an concave piecewise affine arrival curve \(\alpha\) with a finite number of pieces, if the pieces of \(\alpha\) (resp. \(\psi\)) are stored in an array sorted w.r.t. abscissa, one can compute \(D_{\max}(\alpha, \beta)\) and \(B_{\max}(\alpha, \beta)\) in \(O((\log^2(\|\alpha\| + |\beta|)))\) time.

Proof. Thanks to Theorem 5, the computations of \(\beta = \beta_1 \ast \beta_2 \ast \cdots \beta_n\) comes to merging \(n\) sorted lists of respective sizes \(|\beta|\). The complexity stated in the theorem can be achieved by progressing from left to right on the lists while maintaining a binary heap which contains the first available piece of each list and enable to find efficiently the one of smallest slope [11].

In order to compute the bound \(B_{\max}(\alpha, \beta)\), one can use a kind of dichotomical algorithm. Two pointers \(l_\alpha \leq r_\alpha\) (resp. \(l_\beta \leq r_\beta\)) are associated with each sorted array, they points at two pieces of the corresponding function and indicates the extremities on the interval where the maximum vertical distance may be reached. Those pointers will move during the search of this maximum, they are initialized at both extremities of each array, i.e. \(l_\alpha = 1\) and \(r_\alpha = |\alpha|\), \(l_\beta = 1\) and \(r_\beta = |\beta|\). A elementary step of the algorithm can be decomposed as follows:

- Choose the function which maximizes \(max(r_\alpha - l_\alpha, r_\beta - l_\beta)\). Say for instance that it is \(\alpha\).
- Consider the median piece between \(l_\alpha\) and \(r_\alpha\), i.e. the piece of index \(|l_\alpha + r_\alpha|\), and find the piece(s) of \(\beta\) below each extremities of this piece thanks to a dichotomical search in the sorted array of \(\beta\).
- By comparing the slopes of the median piece of \(\alpha\) and of the piece(s) of below its extremities (in some cases one may have to check the slopes of the adjacent pieces), one can decide whether the maximum vertical distance is located on the left, on the right or below the median piece. The pointers of \(\alpha\) can be appropriately updated.

Each step removes at least \(1/4\) of the number of pieces of \(\alpha\) and \(\beta\), and each dichotomical search to locate pieces w.r.t. the median piece is in \(O(\log(|\alpha| + |\beta|))\) time. Thus the overall time complexity is bounded by \(O(\log^2(|\alpha| + |\beta|)) \log_{1/4}(|\alpha| + |\beta|))\).

The same kind of algorithm applies to the computation of \(D_{\max}(\alpha, \beta)\).

5. REFERENCES


