Algorithmique du Network Calculus

Eric Thierry

LIAFA, Univ. Paris 7 & IXXI, LIP, ENS Lyon

Plan

1. Présentation du Network Calculus
   - Les modèles étudiés.
   - Les techniques d'analyse.
   - Comparaisons avec d'autres modèles/analyses.
2. Trois exemples de problèmes algorithmiques.
   - Calcul d'opérations $(\min, +)$ élémentaires.
   - Un problème de routage optimal.
   - Questions de stabilité.

Plan

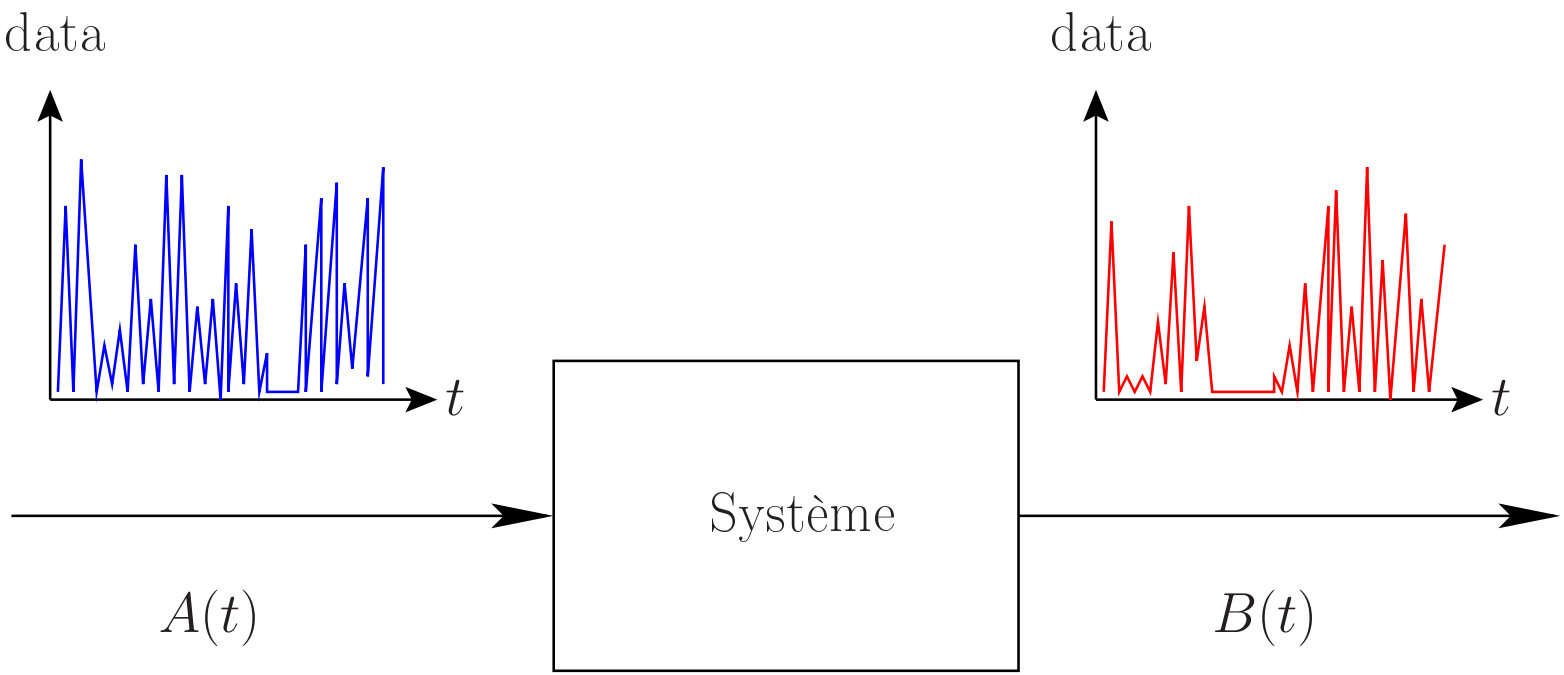
1. Présentation du Network Calculus
   - Les modèles étudiés.
   - Les techniques d'analyse.
   - Comparaisons avec d'autres modèles/analyses.
2. Trois exemples de problèmes algorithmiques.
   - Calcul d'opérations $(\min, +)$ élémentaires.
   - Un problème de routage optimal.
   - Questions de stabilité.

Pas de Network Calculus Stochastique aujourd'hui.

Présentation du Network Calculus

- **Objectif général :** analyse pire cas (déterministe) dans les réseaux de communication.
- **Modèles analysés :** réseaux avec serveurs & flux de données + jeu de contraintes très spécifiques.
- **Analyse systématique :** calculs à l'aide notamment d'opérations $(\min, +)$.
- **Applications projetées :**
  - Grands réseaux de communication (Internet/IntServ/DiffServ).
  - Réseaux embarqués (AFDX).

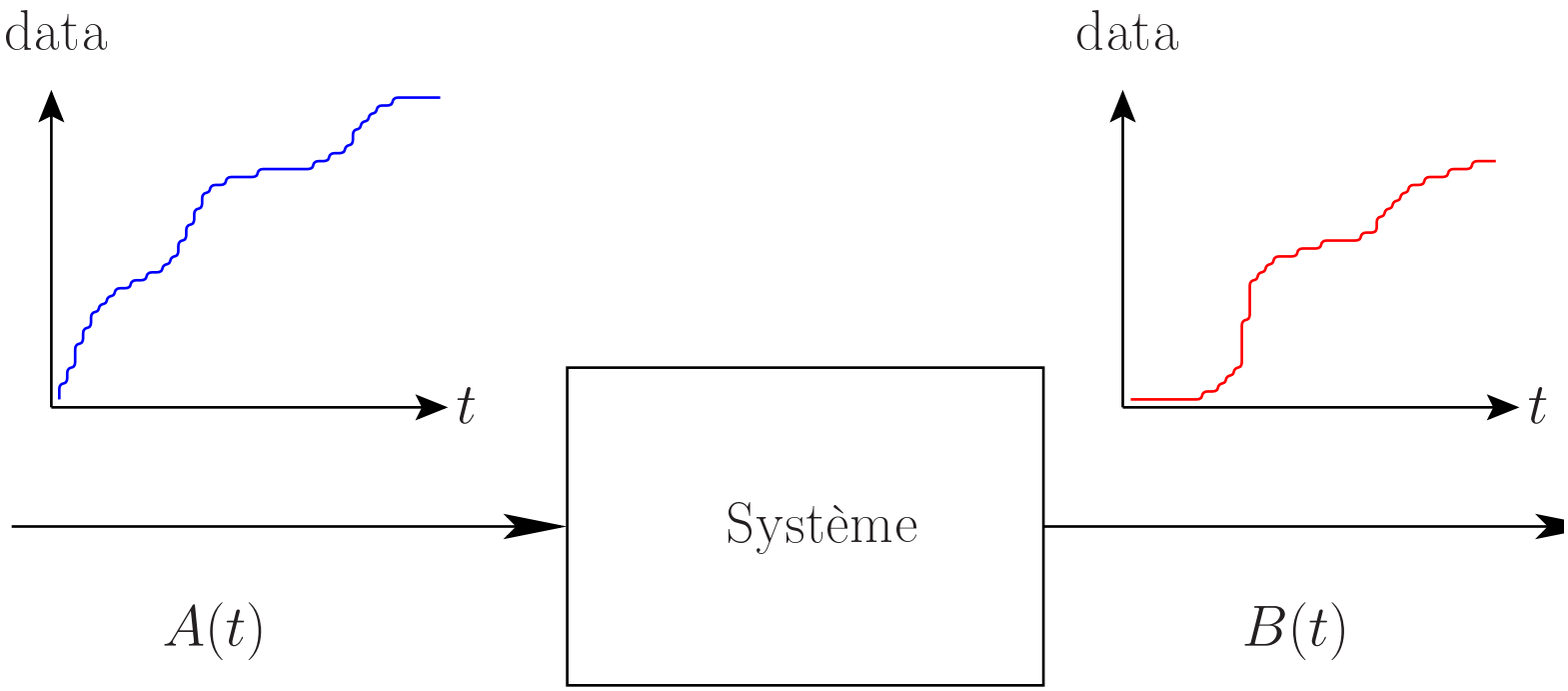
Les courbes de traffic



nbre de bits entrant
à l'instant t

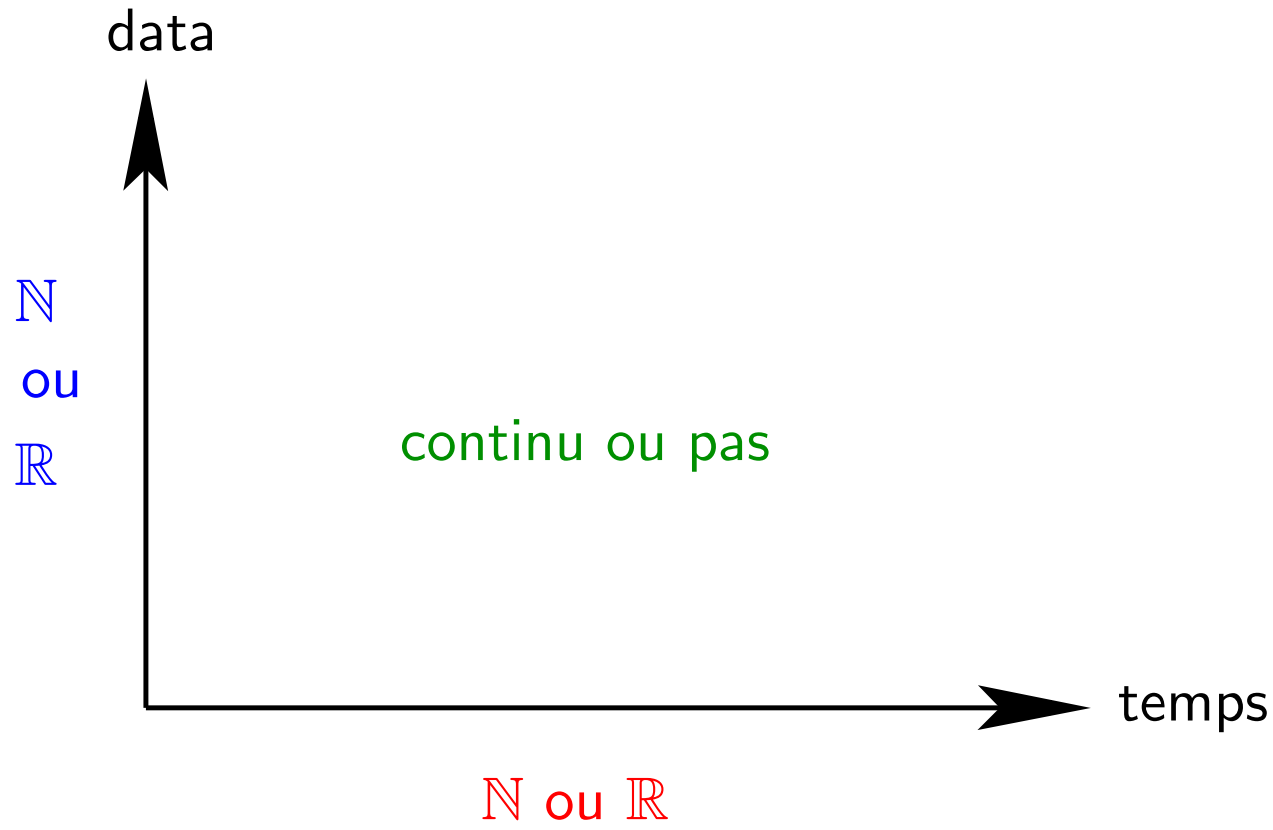
nbre de bits sortant
à l'instant t

Les courbes de traffic



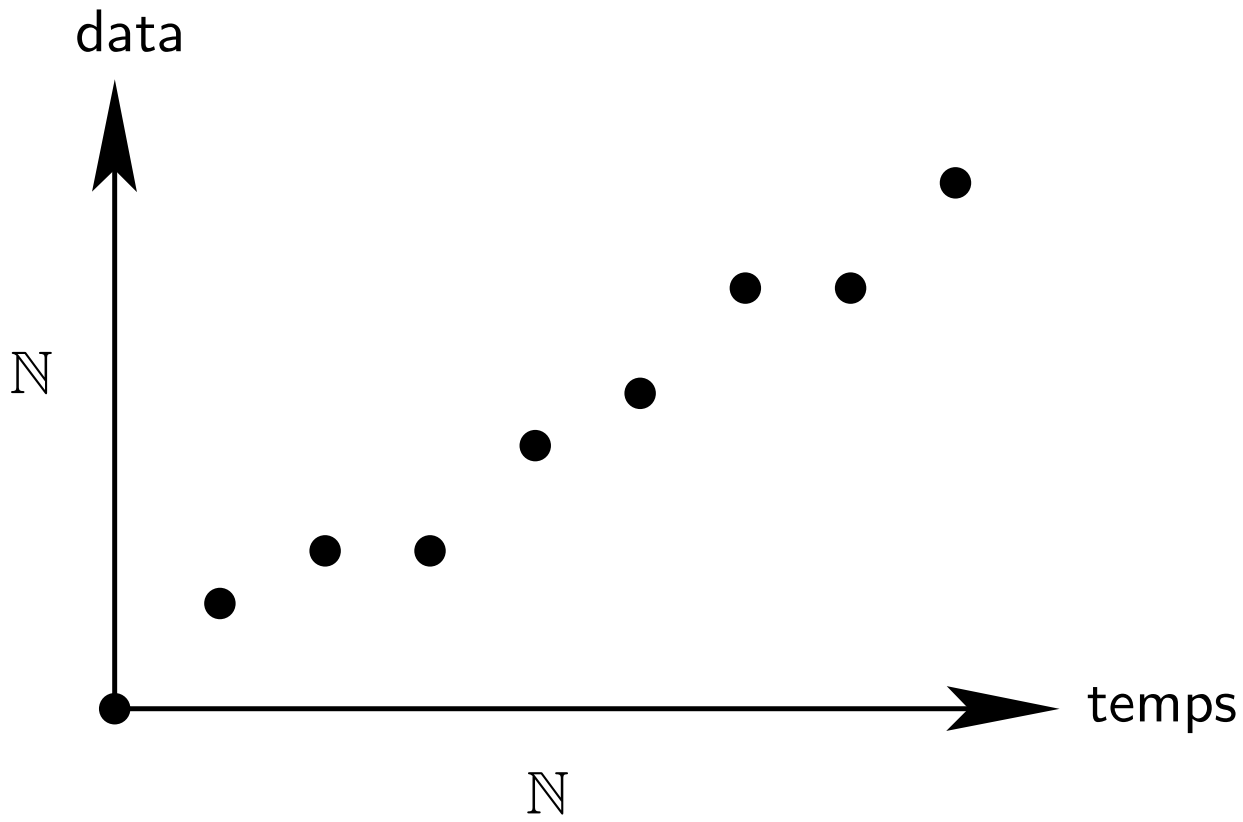
Courbes cumulatives (fonctions croissantes)

L'espace des courbes



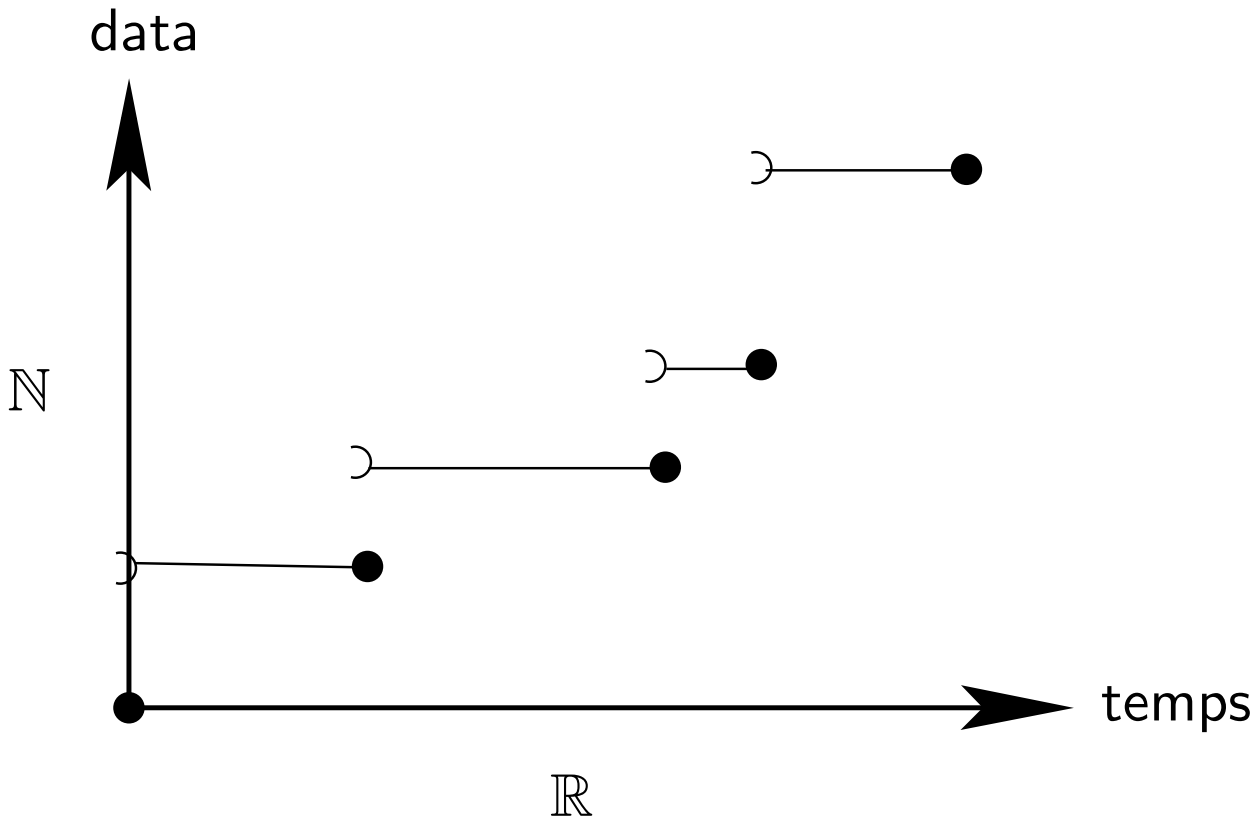
- $\mathcal{D}$ = fonctions de $\mathbb{N}$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$.
- $\mathcal{F}$ = fonctions de $\mathbb{R}_+$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$.

L'espace des courbes



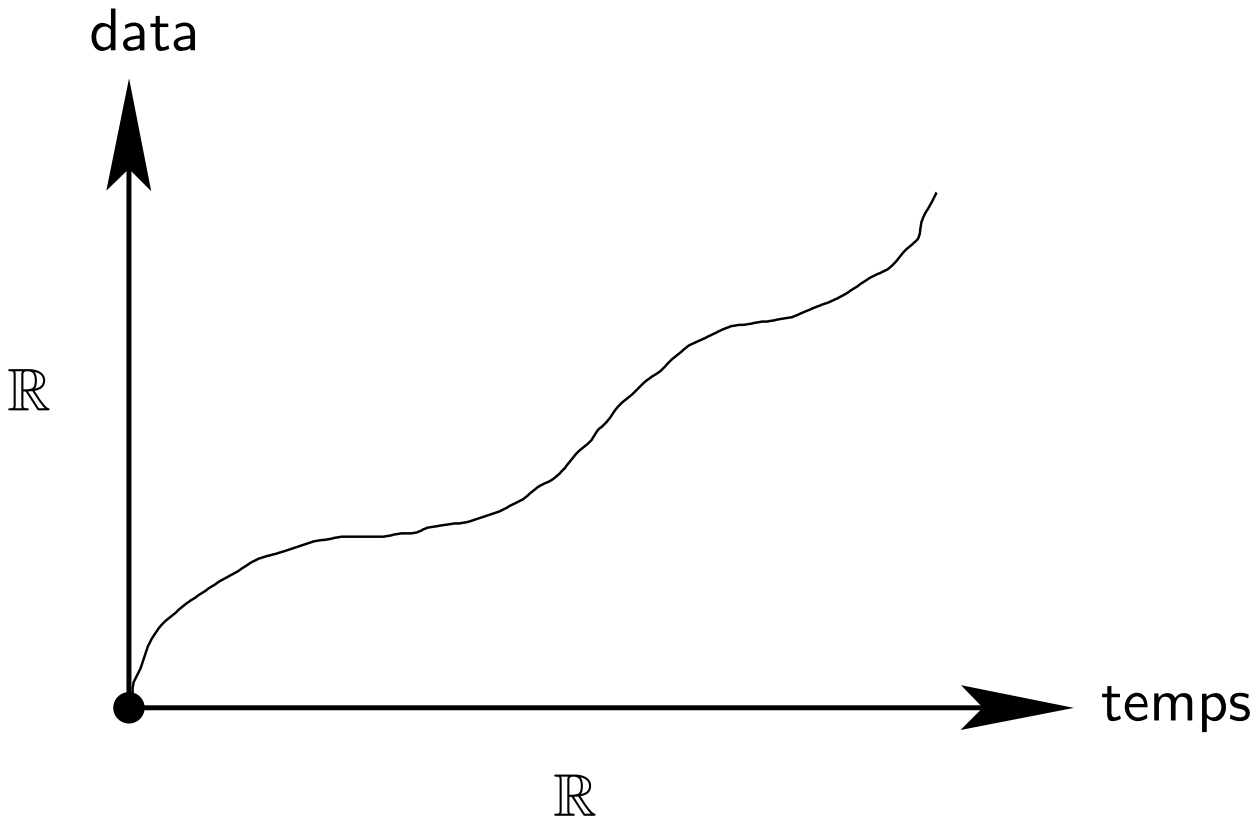
- $\mathcal{D}$ = fonctions de $\mathbb{N}$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$.
- $\mathcal{F}$ = fonctions de $\mathbb{R}_+$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$.

L'espace des courbes



- $\mathcal{D}$ = fonctions de $\mathbb{N}$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$.
- $\mathcal{F}$ = fonctions de $\mathbb{R}_+$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$.

L'espace des courbes



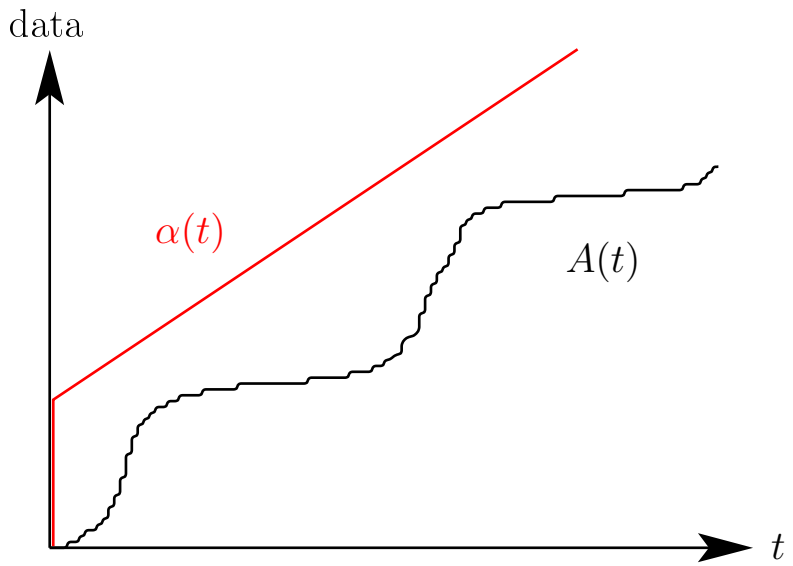
- $\mathcal{D}$ = fonctions de $\mathbb{N}$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$.
- $\mathcal{F}$ = fonctions de $\mathbb{R}_+$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$.

Les contraintes : courbes d'arrivée

Courbe d'arrivée α : soit A une courbe de traffic dans le réseau, A est contrainte par la courbe d'arrivée α (ou α-régulée) si

$$\forall\ 0 \leq s \leq t,\ A(t) - A(s) \leq \alpha(t-s)$$

Exemple : $\alpha(t) = \sigma + \rho t,\ \sigma, \rho \geq 0.$

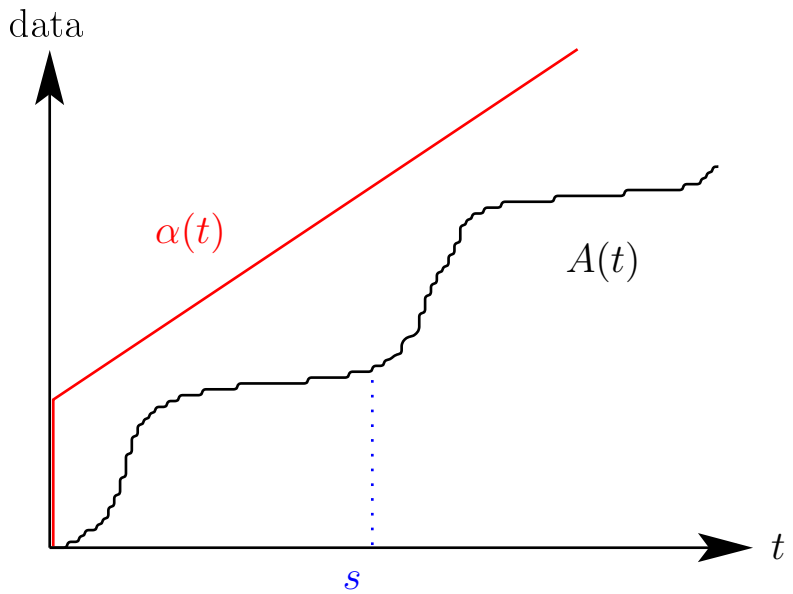


Les contraintes : courbes d'arrivée

Courbe d'arrivée α : soit A une courbe de traffic dans le réseau, A est contrainte par la courbe d'arrivée α (ou α-régulée) si

$$\forall\ 0 \leq s \leq t,\ A(t) - A(s) \leq \alpha(t-s)$$

Exemple : $\alpha(t) = \sigma + \rho t,\ \sigma, \rho \geq 0.$

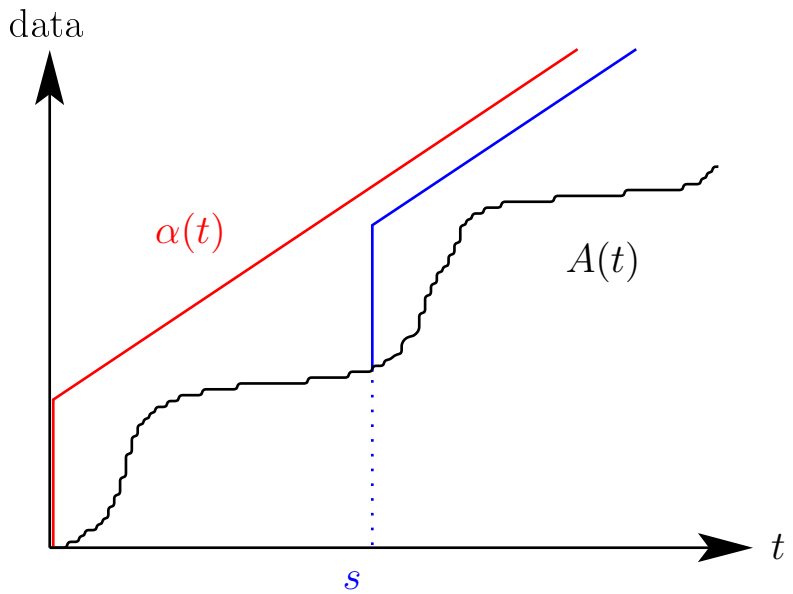


Les contraintes : courbes d'arrivée

Courbe d'arrivée α : soit A une courbe de traffic dans le réseau, A est contrainte par la courbe d'arrivée α (ou α-régulée) si

$$\forall\ 0 \leq s \leq t,\ A(t) - A(s) \leq \alpha(t-s)$$

Exemple : $\alpha(t) = \sigma + \rho t,\ \sigma, \rho \geq 0.$

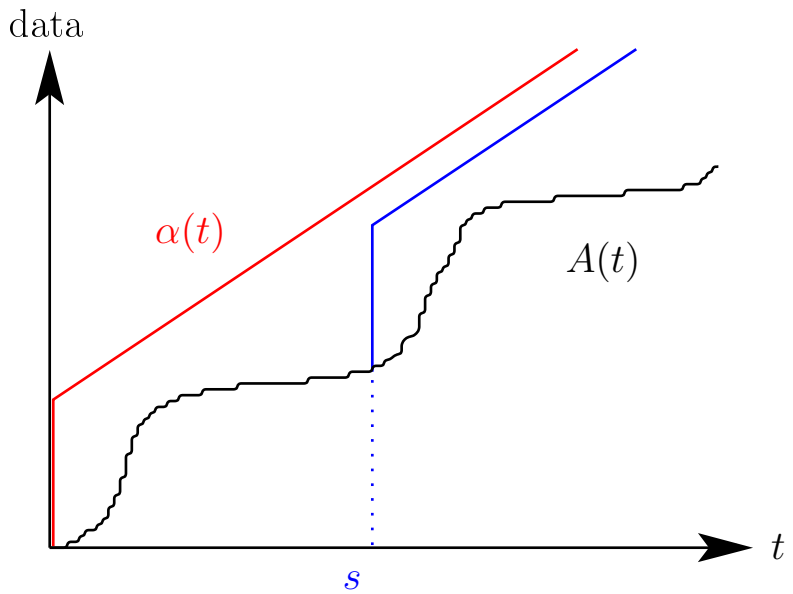


Les contraintes : courbes d'arrivée

Courbe d'arrivée α : soit A une courbe de traffic dans le réseau, A est contrainte par la courbe d'arrivée α (ou α-régulée) si

$$\forall\ 0 \leq s \leq t,\ A(t) - A(s) \leq \alpha(t-s)$$

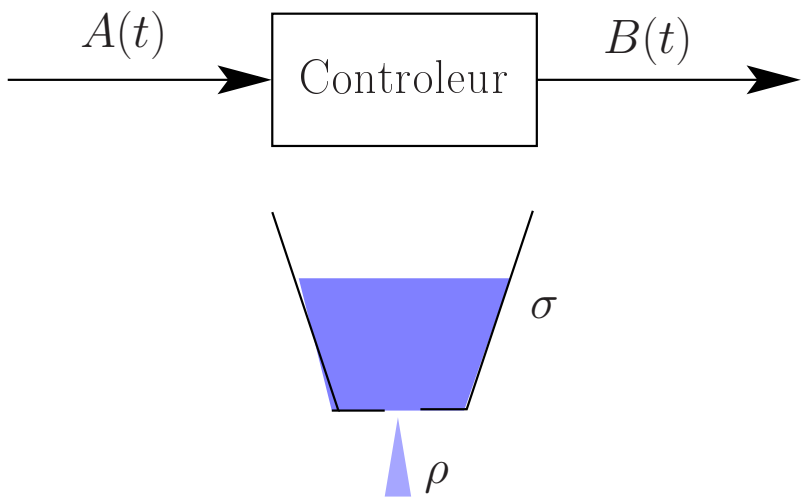
Exemple : $\alpha(t) = \sigma + \rho t,\ \sigma, \rho \geq 0.$



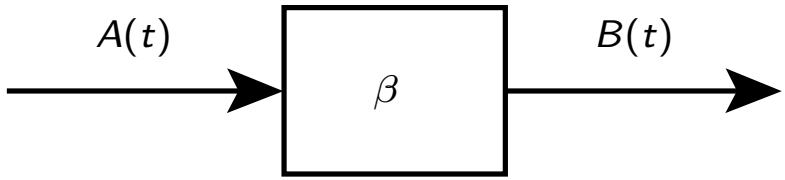
Autre écriture : $A \leq A * \alpha$, où $(f * g)(t) = \inf_{0 \leq s \leq t} (f(s) + g(t-s))$.

Les contraintes : exemples

- Est-ce que toute courbe de traffic $A(t)$ admet une courbe d'arrivée ? OUI $\rightarrow \alpha(t) = \sup_{s \geq 0} A(t+s) - A(s)$, quantité max de données pouvant arriver dans un intervalle de durée t.
- Sceau percé (*leaky bucket*) :
  - Seau de taille σ d'où liquide sécoule à un débit ρ.
  - Contrôleur laisse passer les données en remplissant le seau d'autant de liquide.Sa contrainte : le seau ne doit pas déborder.
  - Conséquence : traffic en sortie régulé par $\alpha(t) = \sigma + \rho t$.



Les contraintes : courbes de service



- Le système garantit une courbe de service minimum β si:

$$\forall t \geq 0,\ B(t) \geq (A * \beta)(t) = \inf_{0 \leq s \leq t} (A(s) + \beta(t-s)).$$

- Le système garantit une courbe de service strict β si, pendant toute période $[s, t]$ chargée ($\forall s < u < t,\ B(u) - A(u) > 0$),

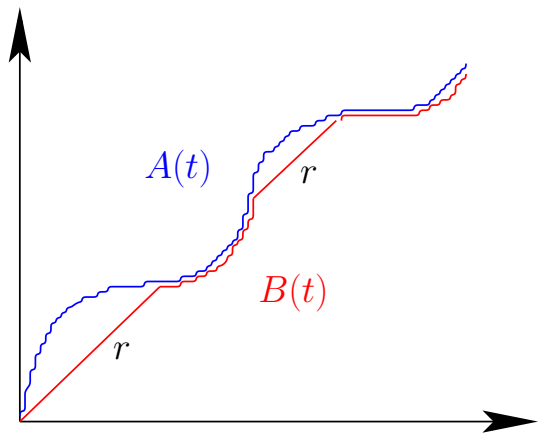
$$B(t) - B(s) \geq \beta(t-s).$$

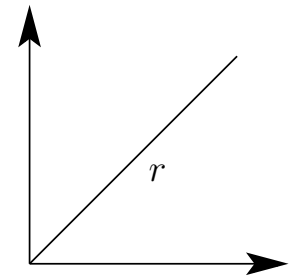
- β service strict $\Rightarrow$ β service minimum.

Remarque : il existe aussi une définition de service maximum.

Les contraintes : exemples de serveurs

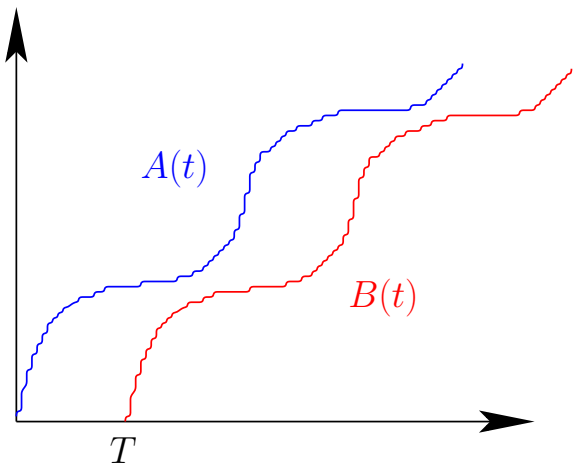
Débit $\geq r$

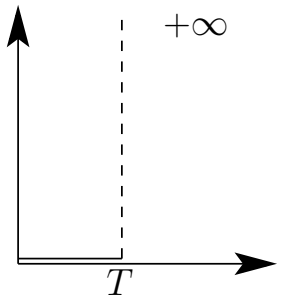




β simple =strict

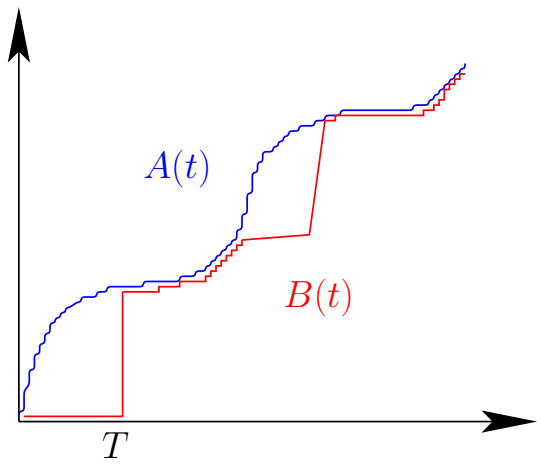
Attente $\leq T$

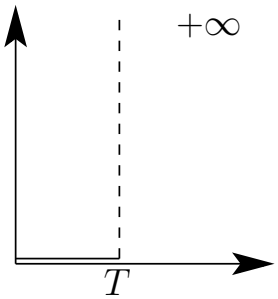




β simple

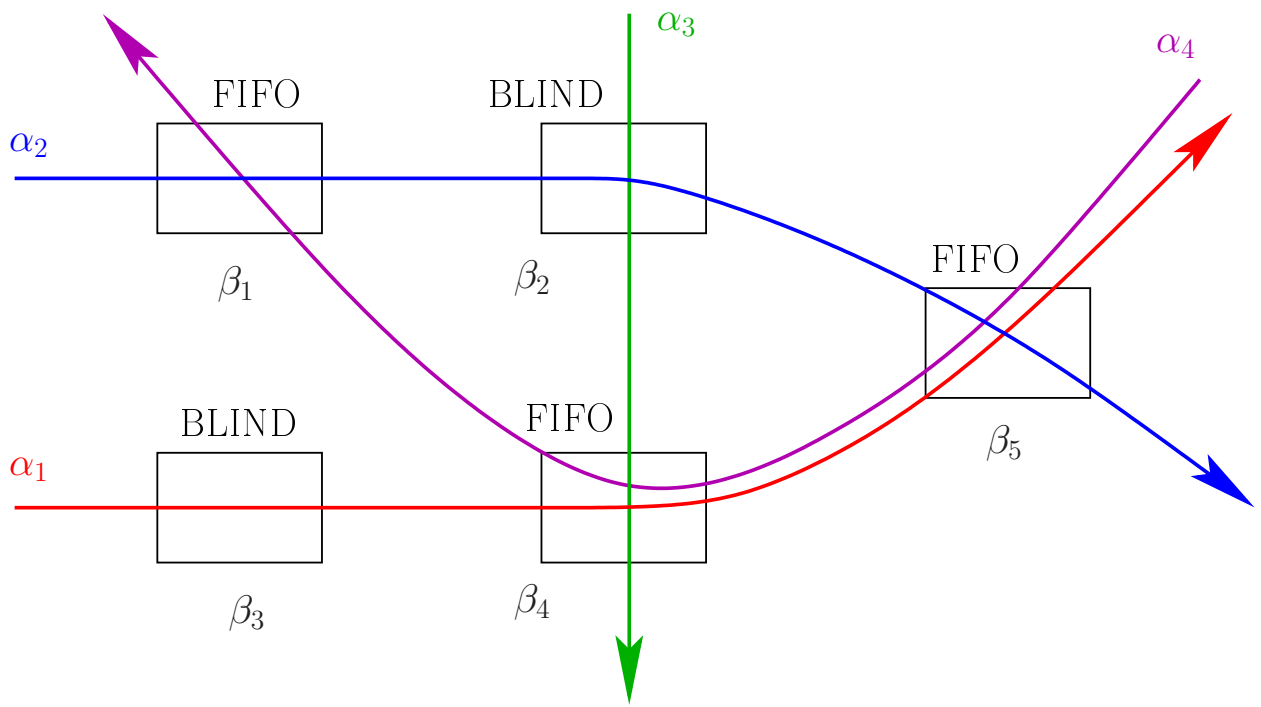
Période chargée $\leq T$





β strict

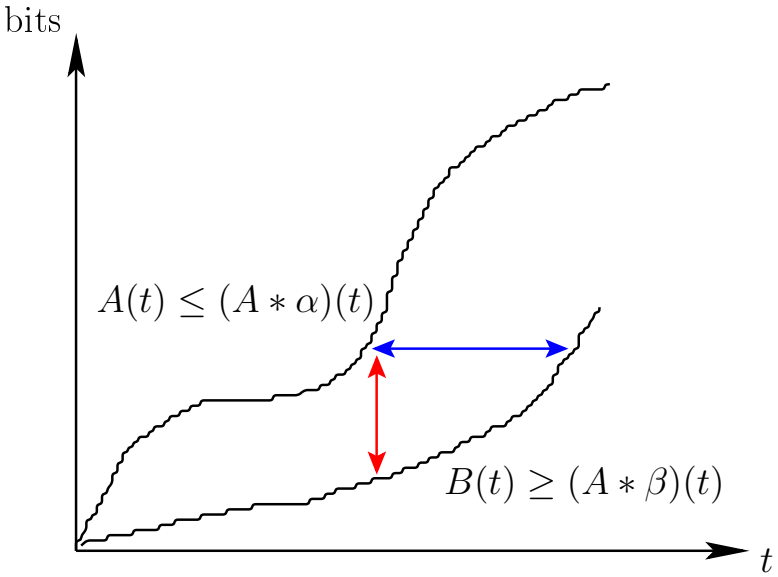
Modèles analysés



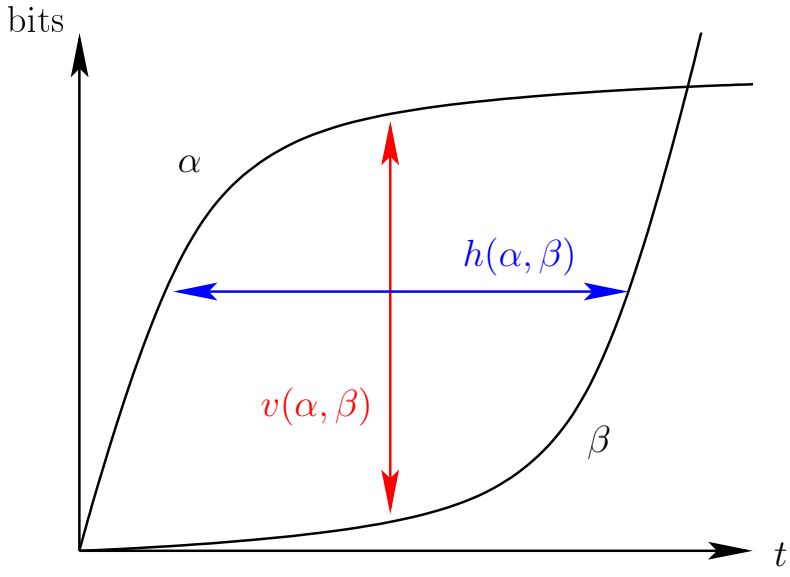
Exemples de questions :

- Quel est le délai maximum de bout en bout d'un paquet du flux 2 ?
- Quel est la taille maximum du buffer au niveau du serveur 5 ?

Des bornes à partir des contraintes



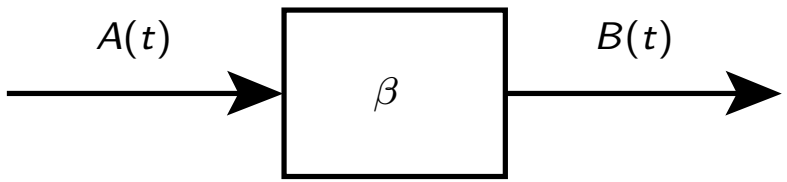
Courbes de traffic



Contraintes

- Charge maximum $\leq v(\alpha, \beta) = \sup\{\alpha(t) - \beta(t) \mid t \geq 0\}$
- Délai maximum $\leq h(\alpha, \beta) = \sup\{\beta^{-1}(z) - \alpha^{-1}(z) \mid z \geq 0\}$

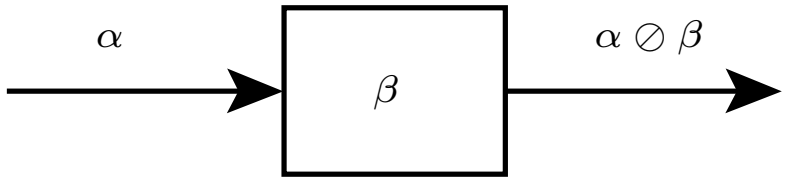
  (sous réserve que flux servi en FIFO et $f^{-1}(z) \stackrel{def}{=} \inf\{t \geq 0 \mid f(t) \geq z\}$)



Propagation des contraintes

Théorème

Un flux contraint par α qui traverse un système de service minimum β ressort contraint par $\alpha \oslash \beta$.

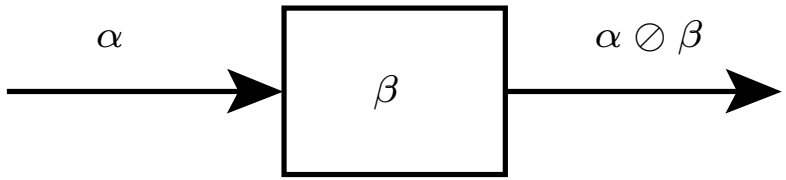


Définition : $(f \oslash g)(t) = \sup_{s \geq 0} (f(t+s) - g(s))$.

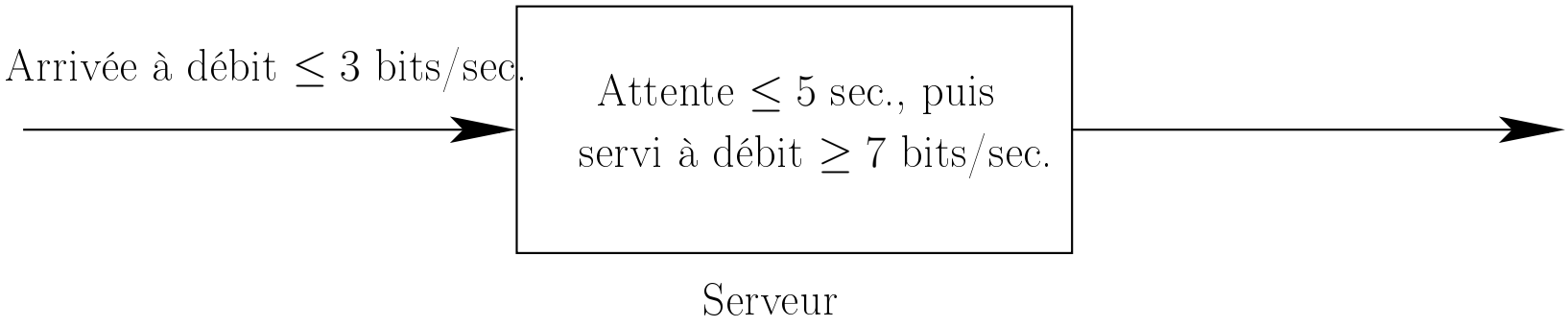
Propagation des contraintes

Théorème

Un flux contraint par α qui traverse un système de service minimum β ressort contraint par $\alpha \oslash \beta$.



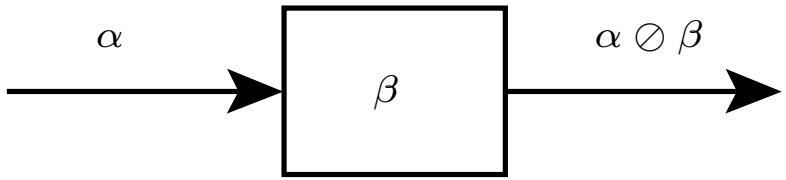
Définition : $(f \oslash g)(t) = \sup_{s \geq 0} (f(t+s) - g(s))$.



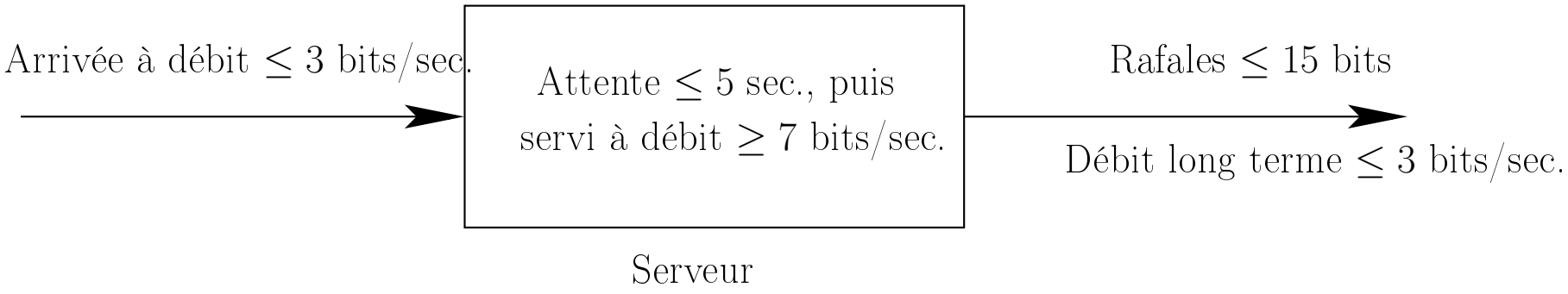
Propagation des contraintes

Théorème

Un flux contraint par α qui traverse un système de service minimum β ressort contraint par $\alpha \oslash \beta$.



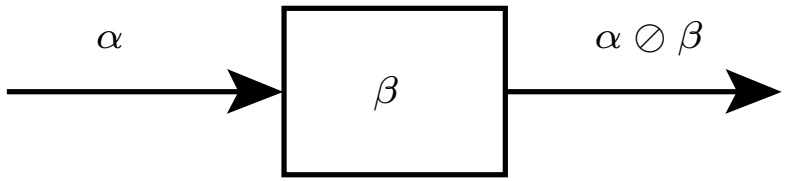
Définition : $(f \oslash g)(t) = \sup_{s \geq 0} (f(t+s) - g(s))$.



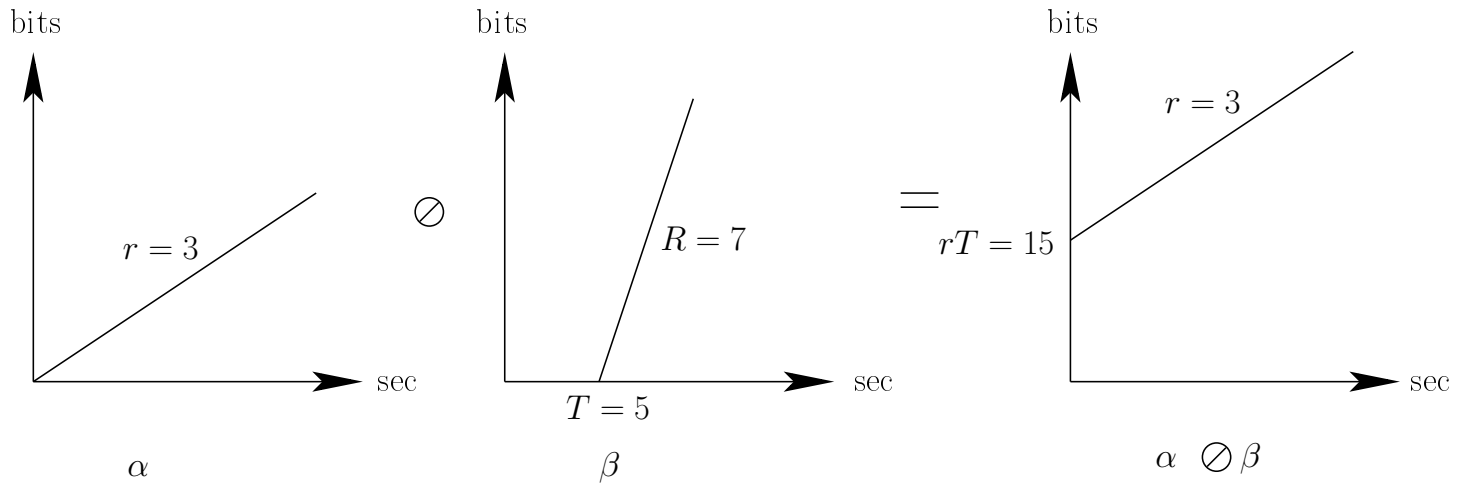
Propagation des contraintes

Théorème

Un flux contraint par α qui traverse un système de service minimum β ressort contraint par $\alpha \oslash \beta$.



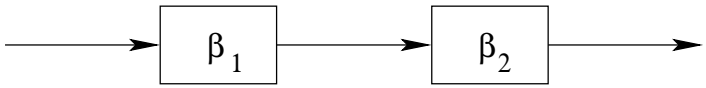
Définition : $(f \oslash g)(t) = \sup_{s \geq 0} (f(t+s) - g(s))$.



Les formules : d'autres règles de combinaison

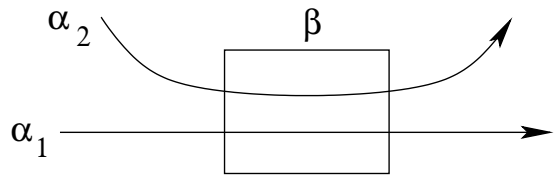
Théorème (Tandem)

La concaténation de deux serveurs de services minimum resp. β_1 et β_2 offre un service minimum $\beta_1 * \beta_2$ au flux.



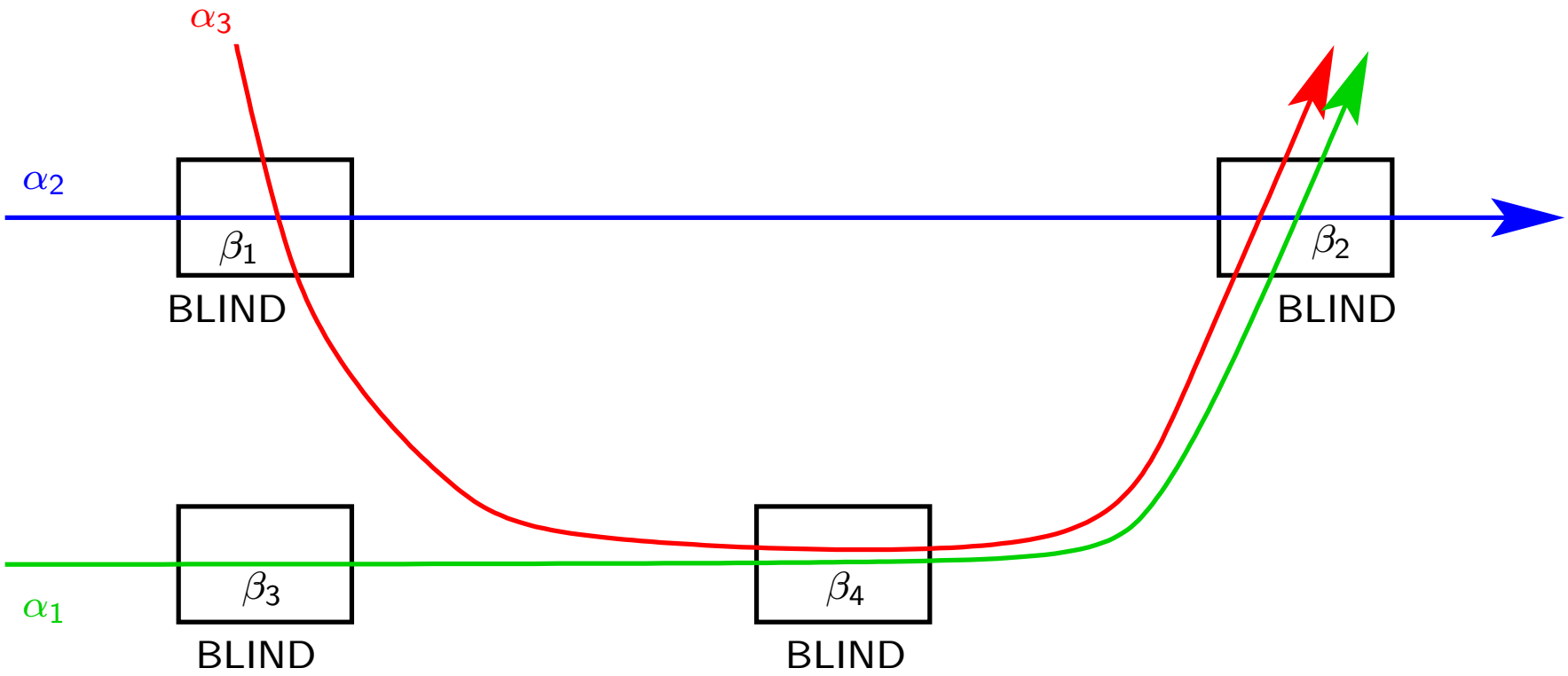
Théorème (Service résiduel)

Si deux flux de courbes d'arrivée α_1, α_2 traversent un serveur de service strict β, alors une courbe de service pour le flux 1 est $\beta_1 = (\beta - \alpha_2)_+$.



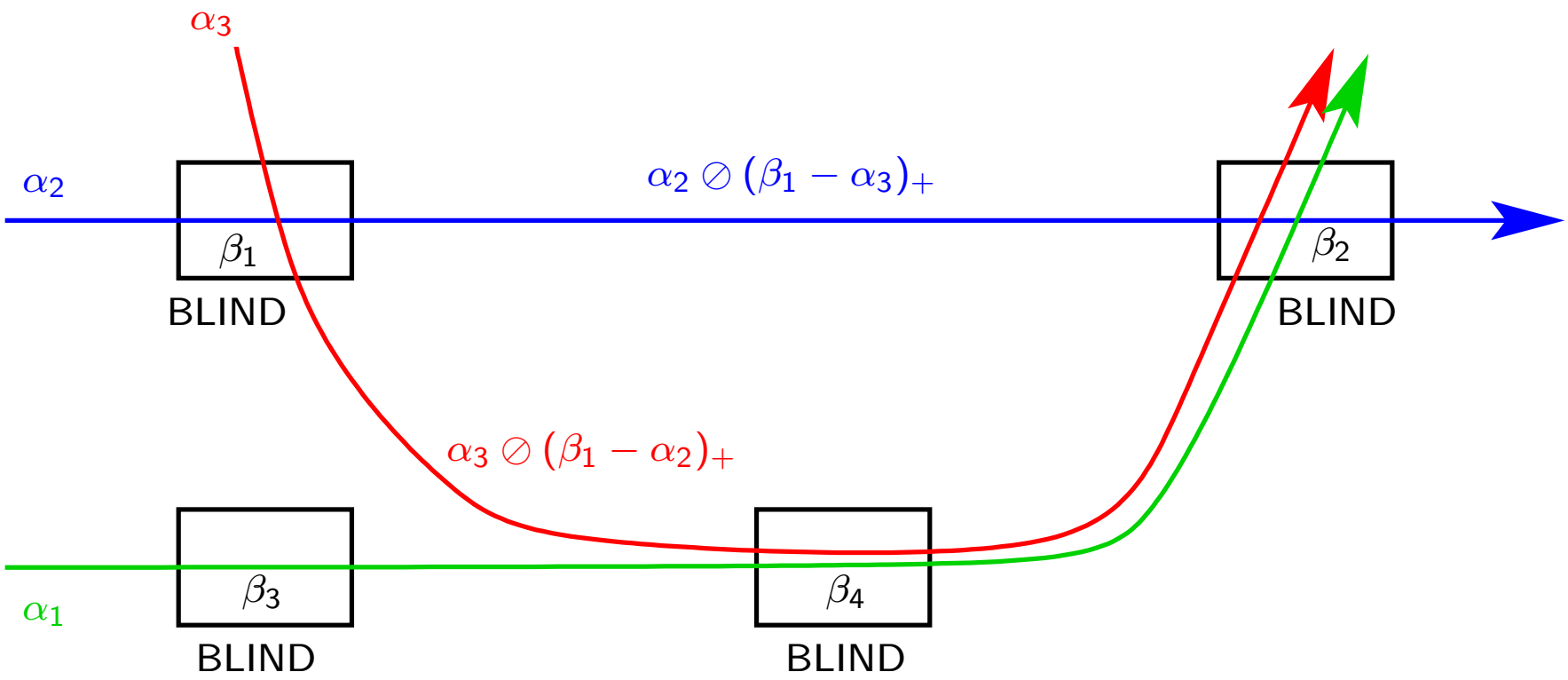
Notation : $x_+ = \max(x, 0)$.

Les formules : un exemple



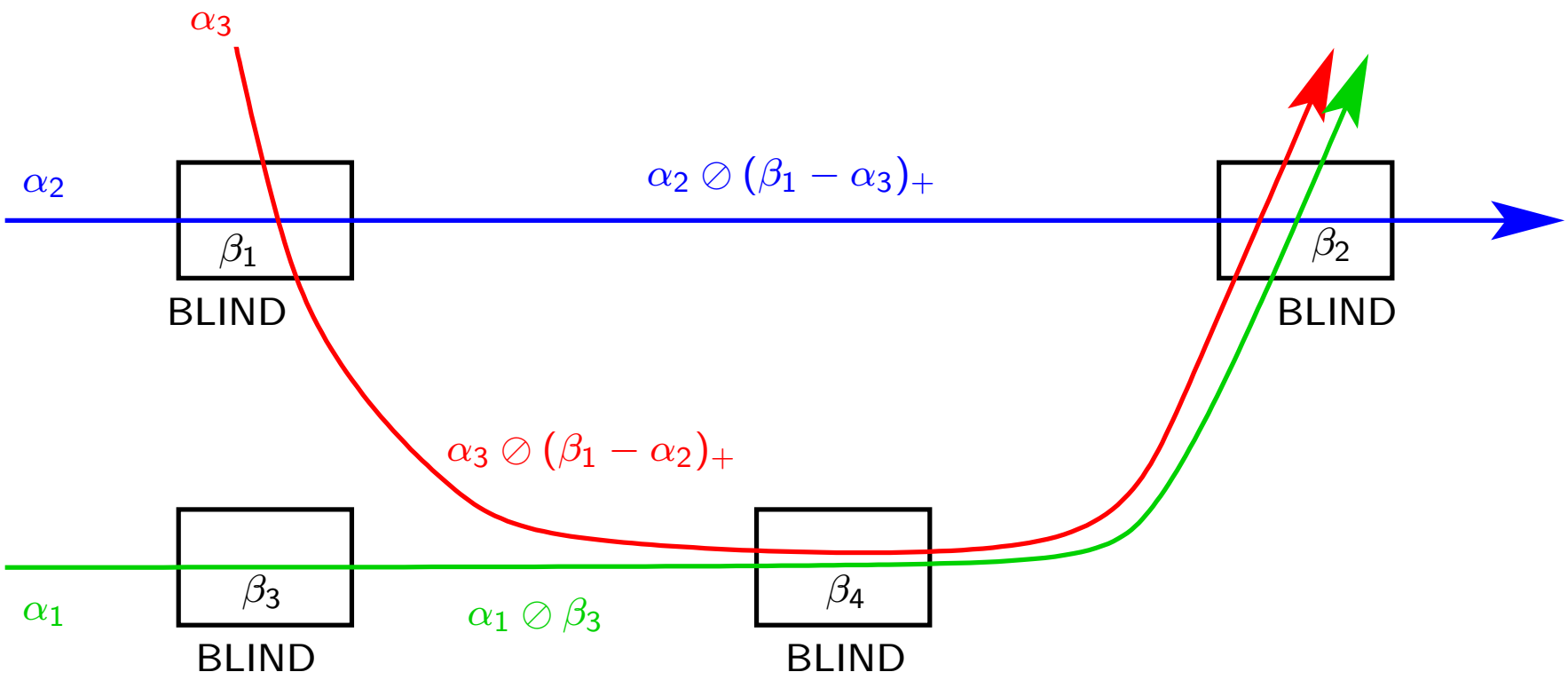
Délai max flux 2 ?

Les formules : un exemple



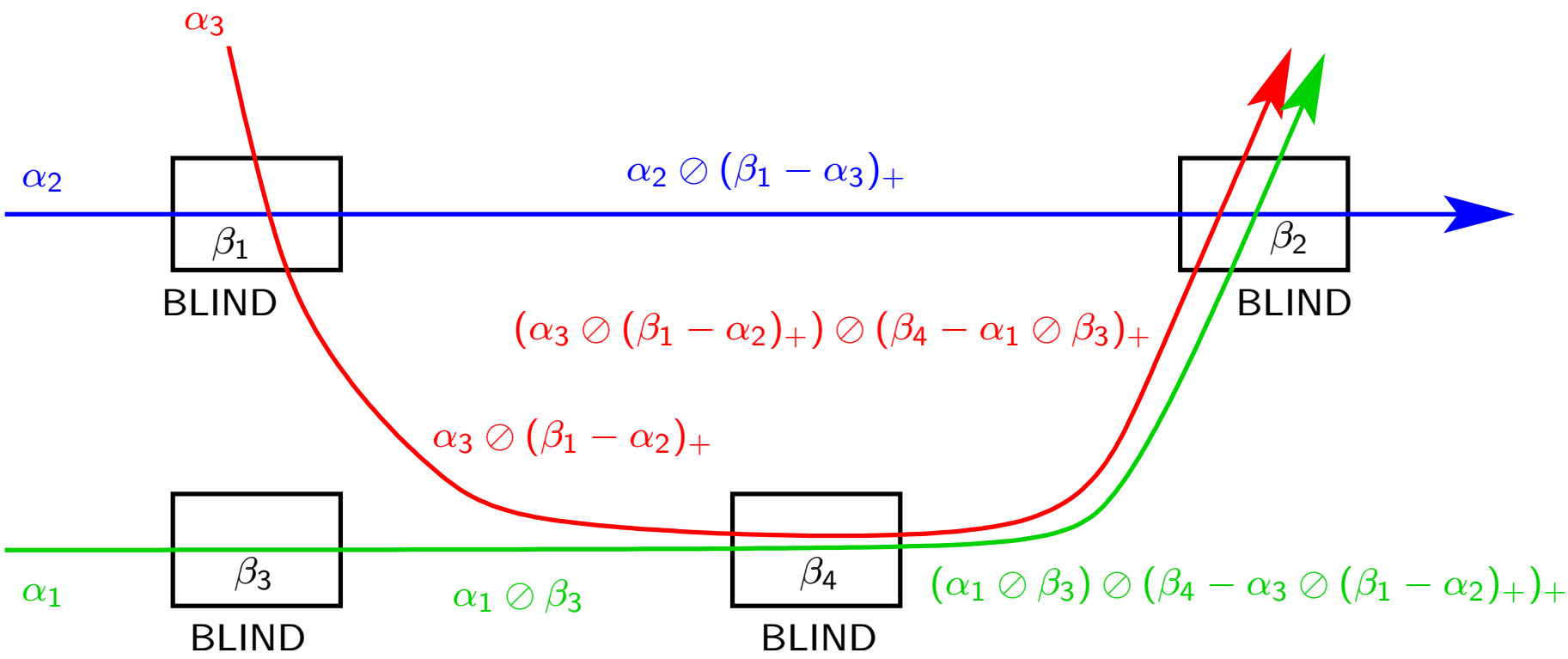
Délai max flux 2 ?

Les formules : un exemple



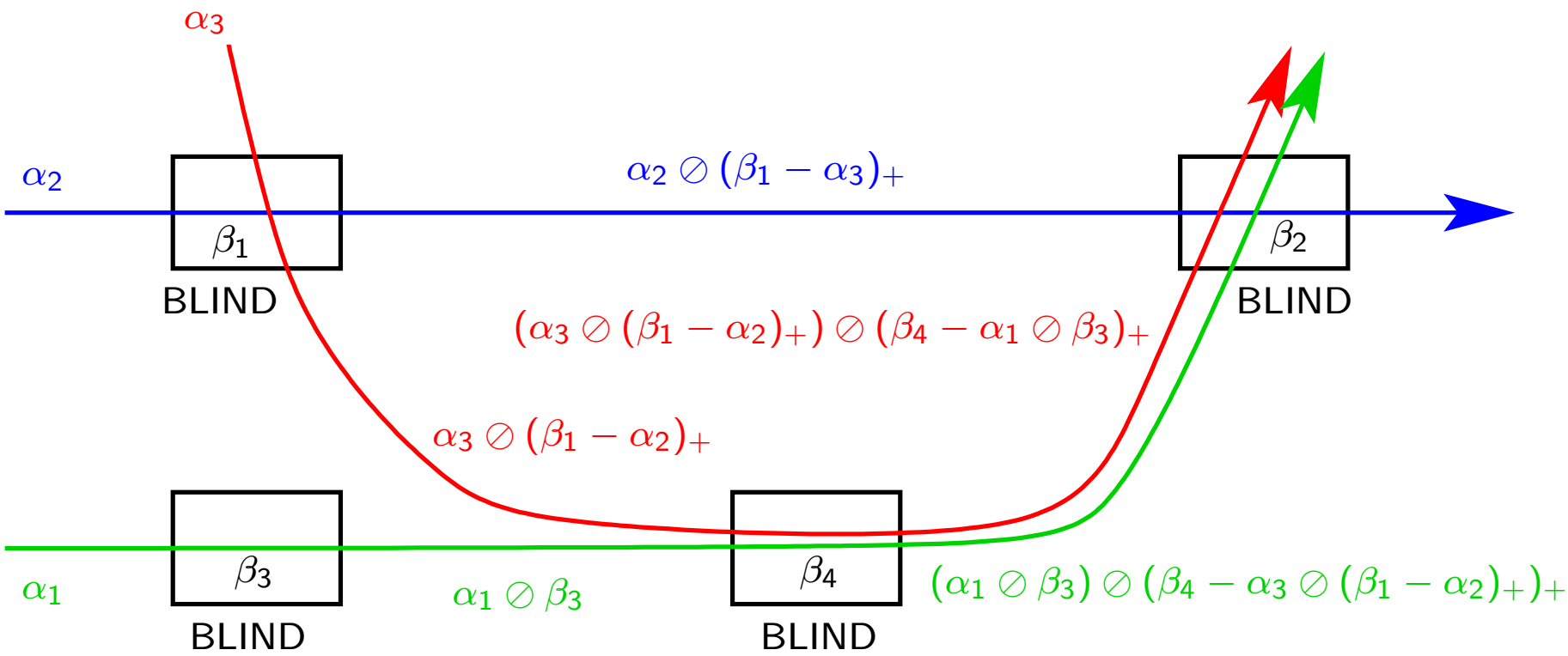
Délai max flux 2 ?

Les formules : un exemple



Délai max flux 2 ?

Les formules : un exemple



Délai max flux 2 ?

$\leq h(\alpha_2, (\beta_1 - \alpha_3)_+ * (\beta_2 - (\alpha_3 \oslash (\beta_1 - \alpha_2)_+) \oslash (\beta_4 - \alpha_1 \oslash \beta_3)_+ - (\alpha_1 \oslash \beta_3) \oslash (\beta_4 - \alpha_3 \oslash (\beta_1 - \alpha_2)_+)))$

Les formules : seule technique ?

Intérêt :

- Formule $\longrightarrow$ algorithme.
- Possibilité de calcul formel.
- Informations qualitatives et pas uniquement quantitatives.

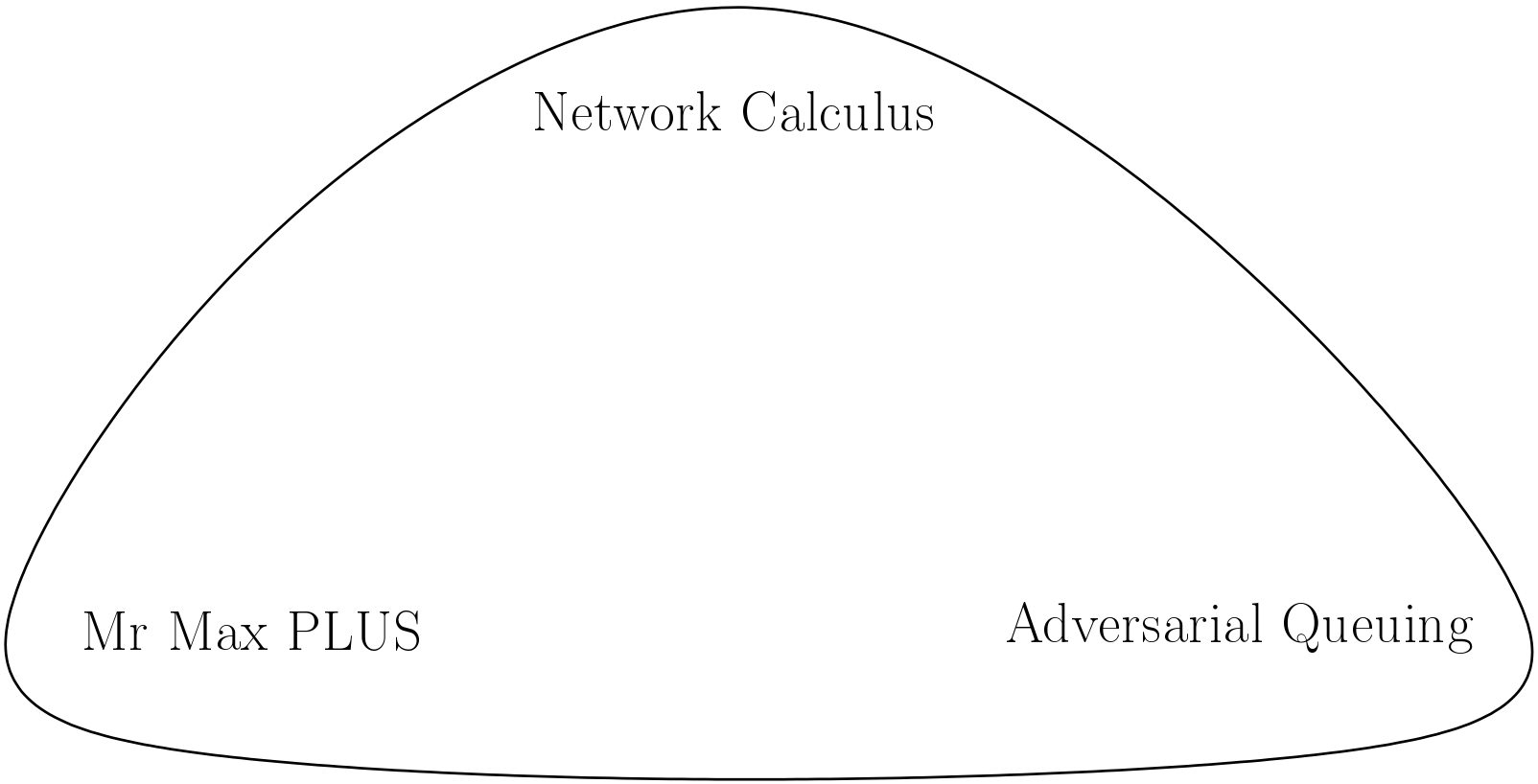
Limites ?

- BLIND : scénarios simples dont bornes précises apparemment pas exprimables par formules simples [Schmitt et al, 2008].
- FIFO : difficultés d'avoir des bornes précises $\longrightarrow$ autres outils & opérateurs [Lenzini et al, 2008] & [Le Boudec, Rizzo, 2008].

Comparatif : échelle d'optimisme

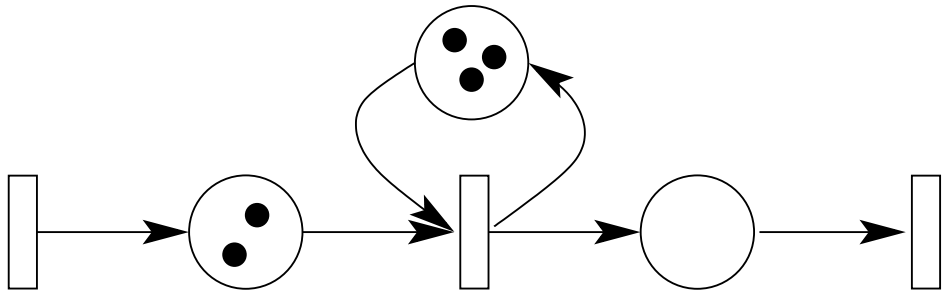
- Théorie des files d'attentes, avec hypothèses markoviennes :
  $+$ optimiste que Network Calculus.
- Théorie "On-line", avec un adversaire tout permis :
  - optimiste que Network Calculus.

Comparatif : problématiques proches



Network Calculus vs. Max PLUS

Proche de la théorie des systèmes $(\min, +)$ linéaires (Exemples : certains réseaux de Petri temporisés).



Commentaire :

- Même opérations élémentaires comme $*$ ou $\oslash$.
- Des systèmes avec souvent $B = \beta * A$ plutôt que $B \geq \beta * A$, et parfois $(\max, +)$ au lieu de $(\min, +)$.
- Travail plutôt directement sur les courbes de traffic que sur les contraintes (courbes d'arrivée et de service).
- Moins la problématique de flux pouvant s'aggréger à des nœuds.

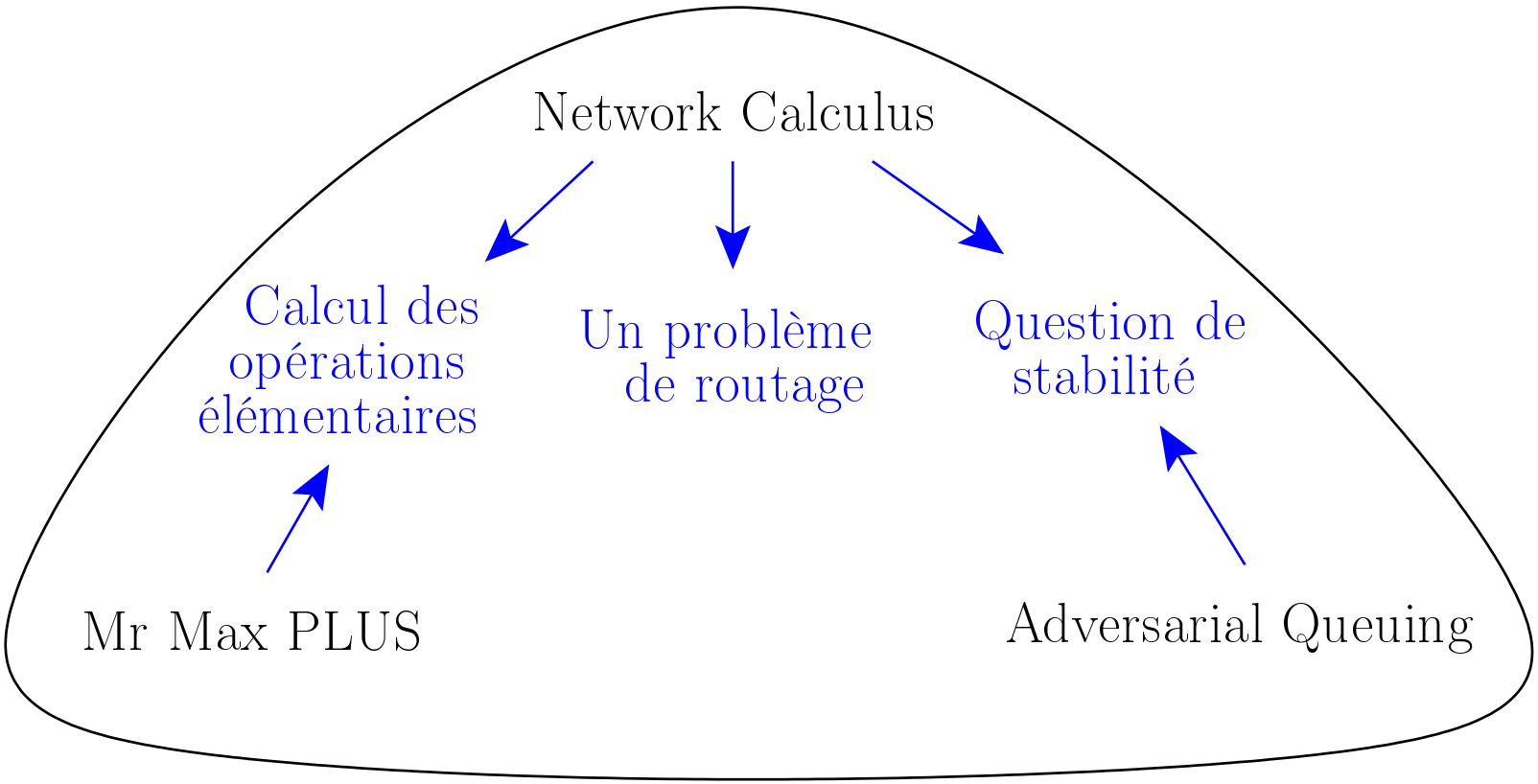
Network Calculus vs. Adversarial Queuing Theory

Modèles variés :

- Réseau :
  - Topologie (graphe).
  - Serveurs à débit constant, délais sur les liens/aux serveurs.
- Traffic :
  - Sessions (flux) sur chemins fixes ou pas.
  - Injection des données par un adversaire avec contraintes.
- Politiques de service : FIFO/LIFO/NTG/LTG/NTS/LTS/SIS/LIS

Question : stabilité ? (quantité de donnée dans le réseau est bornée ?)
PSM (Permanent Session Model) : sessions sur chemins fixes & nbre de données injectées dans flux i pendant $[s, s+t[$ est $\leq \sigma_i + \rho_i t$.
AQM (Adversarial Queuing Model) : sessions sur chemins fixes & injection vérifiant que pour tout arc e et période $[s, s+t[$, le nbre de données devant passer par e est $\leq w + rt$.
Commentaire : lien clair entre modèles PSM et Network Calculus.
Résultats PSM : souvent recherche de contre-exemples.

Trois problèmes algorithmiques



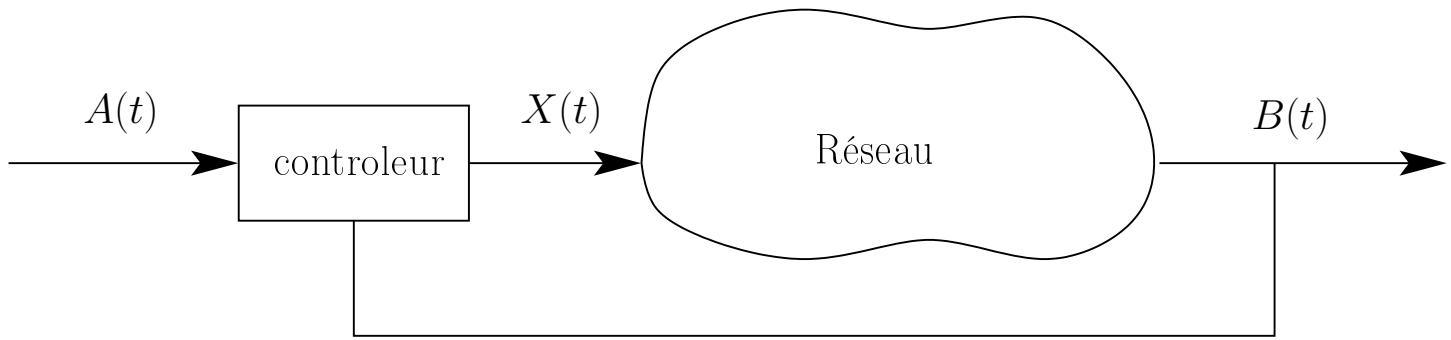
Définition

Soient f, g *de* X *dans* $\mathbb{R} \cup \{-\infty, +\infty\}$ *où* $X = \mathbb{N}$ *ou* $\mathbb{R}_+$,

- $\min(f,g)(t) = \min(f(t), g(t))$.
- $\max(f,g)(t) = \max(f(t), g(t))$.
- $(f+g)(t) = f(t) + g(t)$.
- $(f-g)(t) = f(t) - g(t)$.
- $(f * g)(t) = \inf_{0 \le s \le t}(f(s) + g(t-s))$ **(Convolution)**.
- $(f \oslash g)(t) = \sup_{s \ge 0}(f(t+s) - g(s))$ **(Déconvolution)**.
- $f^*(t) = \inf(f^{(0)}, f^{(1)}, f^{(2)}, f^{(3)}, \ldots)$ **(Clôture sous-additive)**
  où $f^{(0)}(t) = 0$ *si* $t = 0$ *and* $= +\infty$ *si* $t > 0$,
  et $f^{(k)} = \underbrace{f * \cdots * f}_{k \text{ fois}}$ *pour* $k \ge 1$.

Clôture sous-additive : exemples

- A α-régulée $\Rightarrow$ $\alpha * \alpha$-régulée $\Rightarrow \cdots \Rightarrow$ α^*-régulée ($\alpha^* \leq \alpha$).
- Contrôle d'admission tel que charge dans réseau toujours $\leq W$ $\longrightarrow$ une courbe de service minimum de l'ensemble [contrôleur + réseau] est $\beta * (\beta + W)^*$.

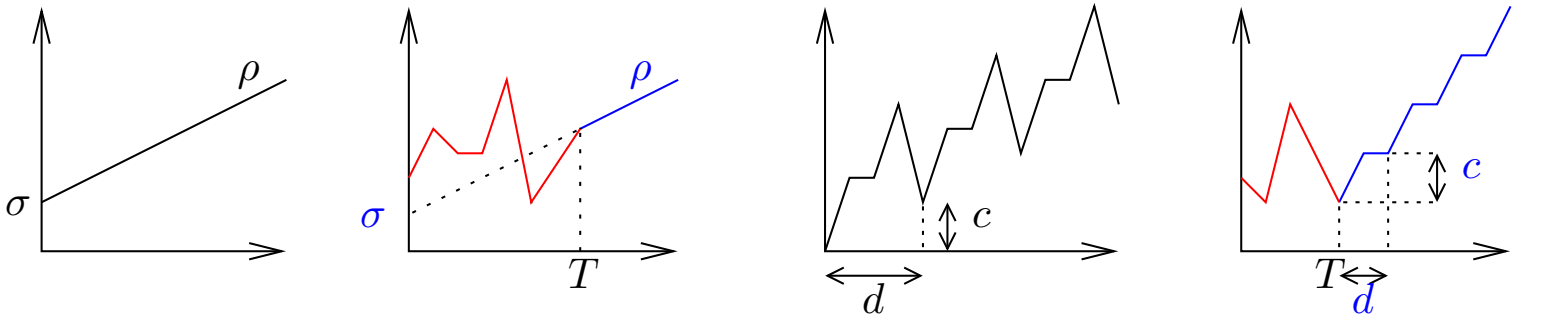


Quelques classes de fonctions

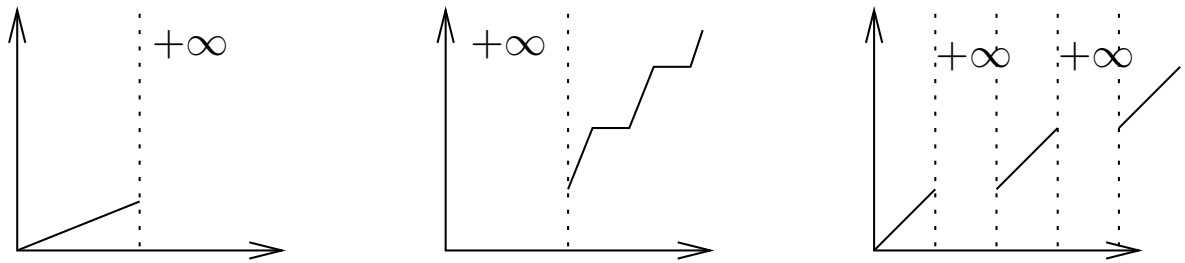
Soit f une fonction de X dans $\mathbb{R} \cup \{-\infty, +\infty\}$ où $X = \mathbb{N}$ ou $\mathbb{R}_+$, alors, en notant $X^* = X \setminus \{0\}$:

- f est *affine* si $\exists \sigma, \rho \in \mathbb{R}, \forall t \in X, f(t) = \rho t + \sigma$ ou $\forall t \in X, f(t) = +\infty$ (resp. $-\infty$).
- f est *ultimement affine* si $\exists T \in X$, $\exists \sigma, \rho \in \mathbb{R}$, $\forall t > T, f(t) = \rho t + \sigma$ ou $\forall t > T, f(t) = +\infty$ (resp. $-\infty$).
- f est *pseudo-periodique* si $\exists (c, d) \in \mathbb{R} \times X^*$, $\forall t \in X, f(t + d) = f(t) + c$.
- f est *ultimement pseudo-periodique* si $\exists T \in X$, $\exists (c, d) \in \mathbb{R} \times X^*$, $\forall t > T, f(t + d) = f(t) + c$.
- f est *ultimement franche* if $\exists T \in X$, $\forall t > T$, $f(t) \in \mathbb{R}$, ou $\forall t > T$, $f(t) = +\infty$, ou $\forall t > T$, $f(t) = -\infty$.
- f est *franche* si elle est ultimement franche, et $\forall\ 0 \leq t < T$, $f(t) \in \mathbb{R}$, et $f(T) \in \mathbb{R}$ ou éventuellement $f(T) = +\infty$ (resp. $-\infty$) si $\forall t > T$, $f(t) = +\infty$ (resp. $-\infty$).

Illustrations des classes



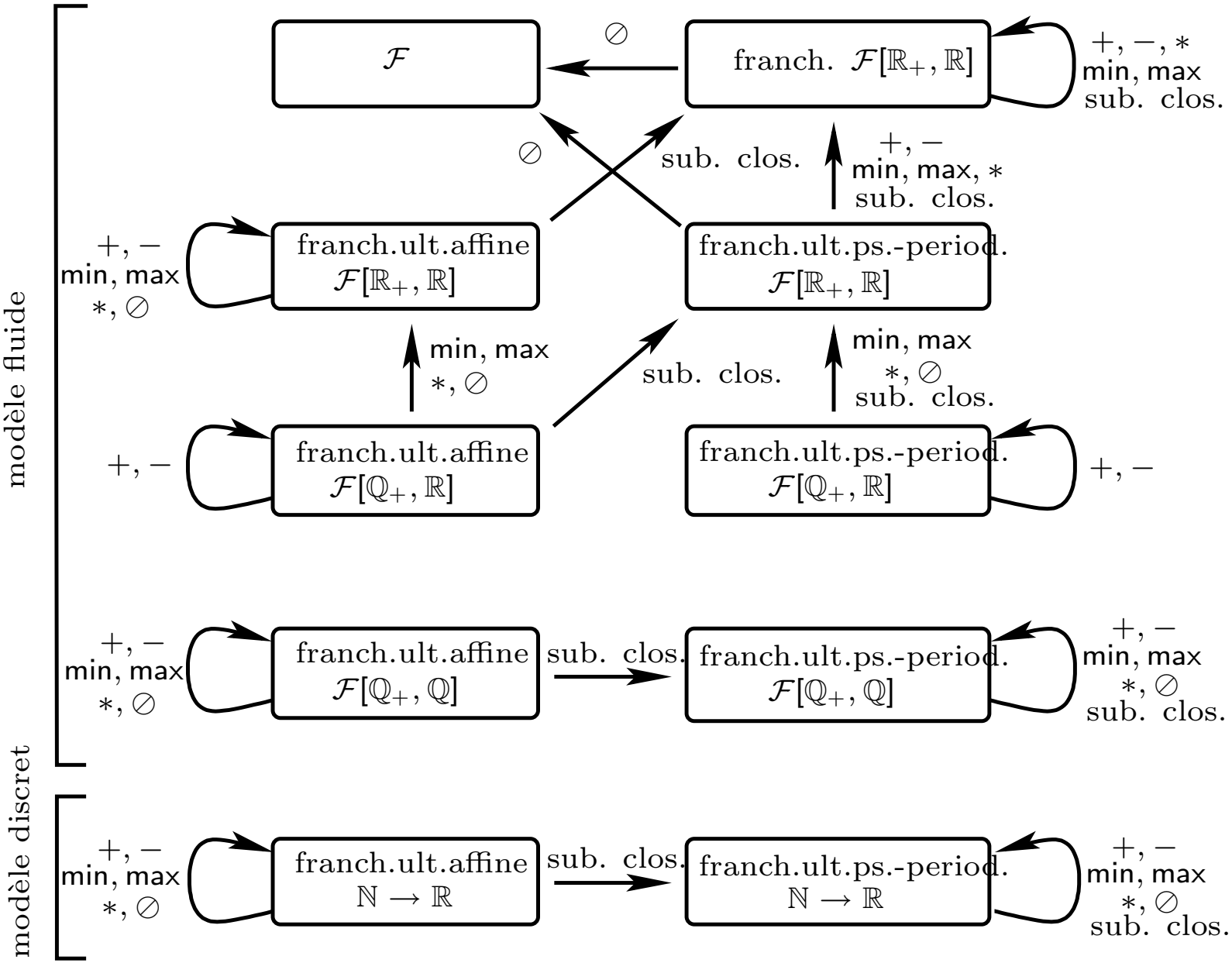
affine | ultimement affine | pseudo-périodique | ultimement pseudo-périodique



franche | pas franche | pas franche

Notation : $\mathcal{F}[X, Y]$ = fonctions de $\mathbb{R}_+$ dans $\mathbb{R} \cup \{-\infty, +\infty\}$ affines par morceaux avec des sauts ou changements de pentes à des points d'abscisses dans X et d'ordonnées dans Y.

Navigation entre les différentes classes



Théorèmes de clôture

Deux classes stables pour $\min, \max, +, -, *, \oslash, .^*$ et couvrant les exemples rencontrés en Network Calculus :

- Dans $\mathcal{F}$, les fonctions affines par morceaux franches et ultimement pseudo-périodiques.
- Dans $\mathcal{D}$, les fonctions franches et ultimement pseudo-périodiques.

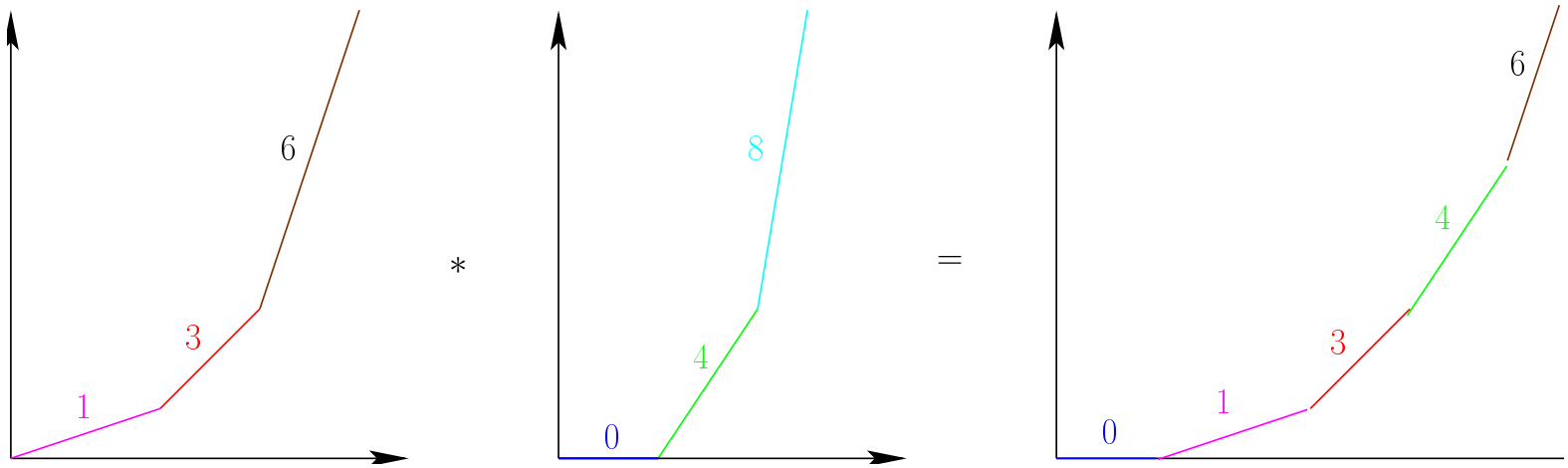
On peut difficilement faire plus petit pour rester stable :

Théorème

Dans $\mathcal{F}$, la clotûre des fonctions affines pour les opérations $\min, \max, +, -, *, \oslash, .^*$ est la classe des fonctions affines par morceaux continues (sauf éventuellement en 0) ultimement pseudo-périodiques.

Question : quels algorithmes pour implémenter les opérations élémentaires sur ces classes de fonctions ?

Algorithme de convolution : cas particulier

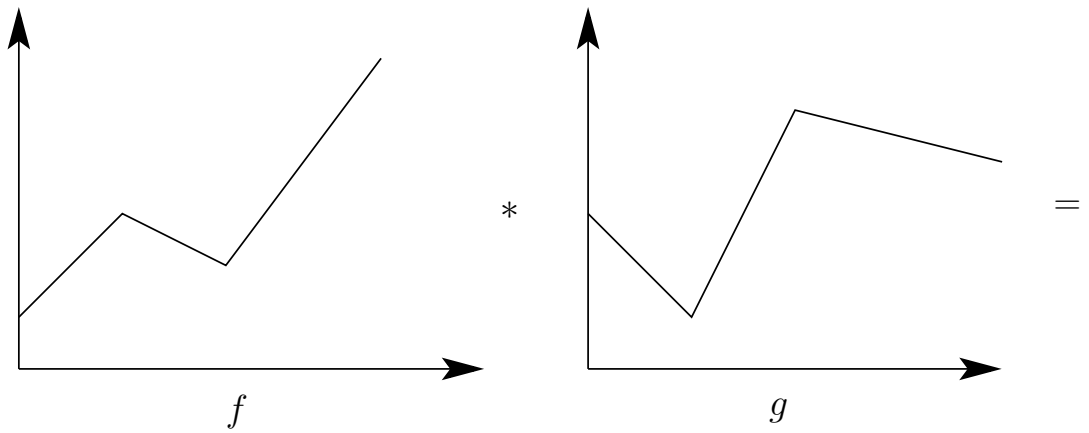


Théorème (Le Boudec, Thiran, 2001)

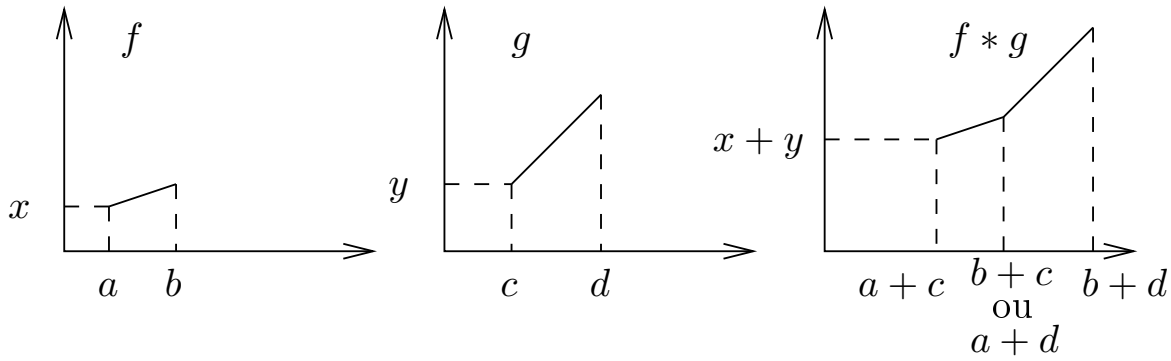
Convolution de f et g convexes et affine par morceaux = concaténation des différents morceaux de f et g triés par pentes croissantes.

$\rightarrow$ Complexity: $\mathcal{O}(|f| + |g|)$ où $|f|$ (resp. $|g|$) est le nombre de morceaux de f (resp. g).

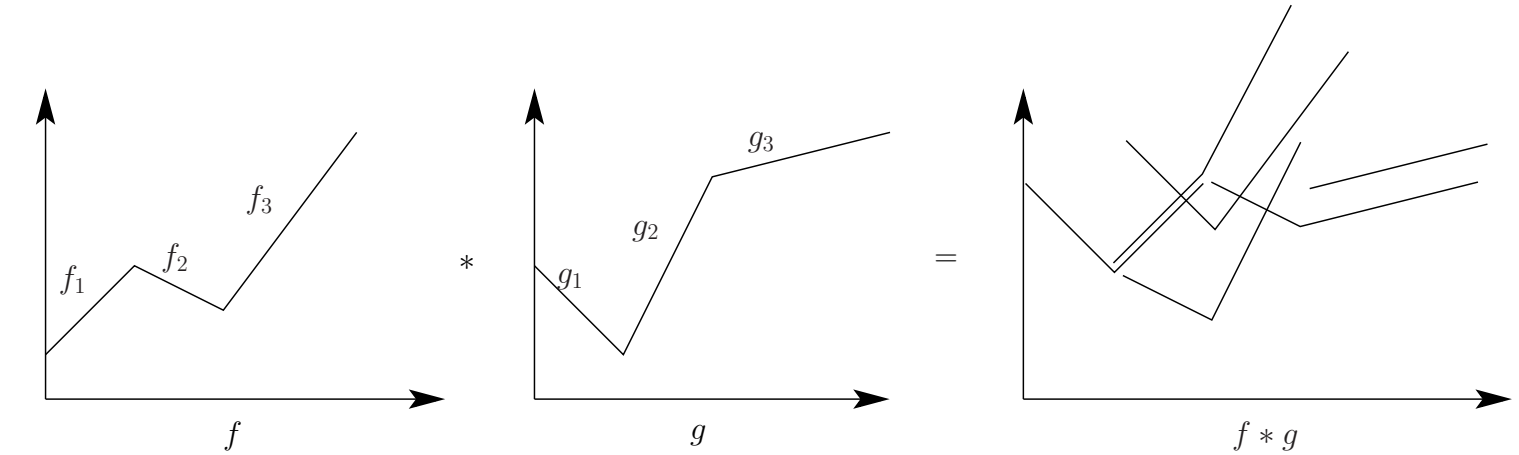
Algorithme de convolution : cas général



Distributivité : $\min(f_1, f_2, \ldots, f_n) * \min(g_1, g_2, \ldots, g_m) = \min_{i,j}(f_i * g_j)$.



Algorithme de convolution

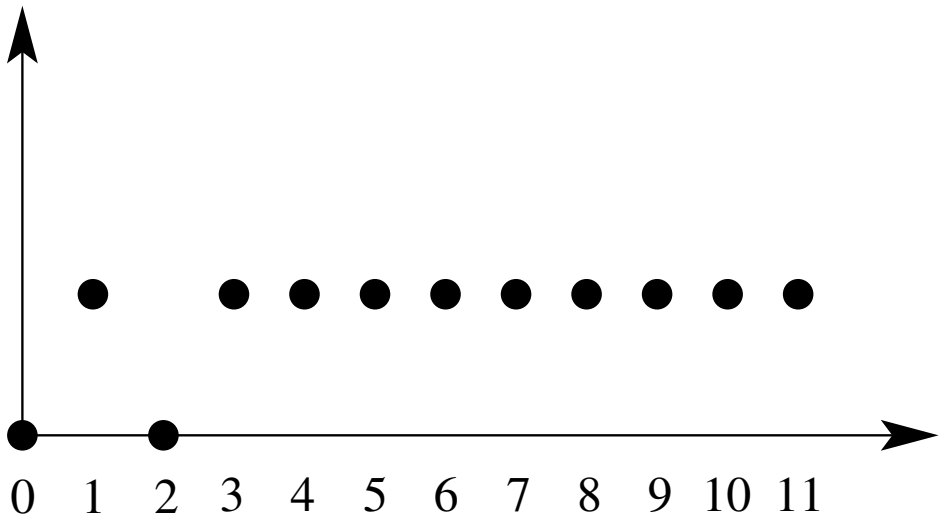


Nbre de morceaux de cette enveloppe : $\ell(N)$

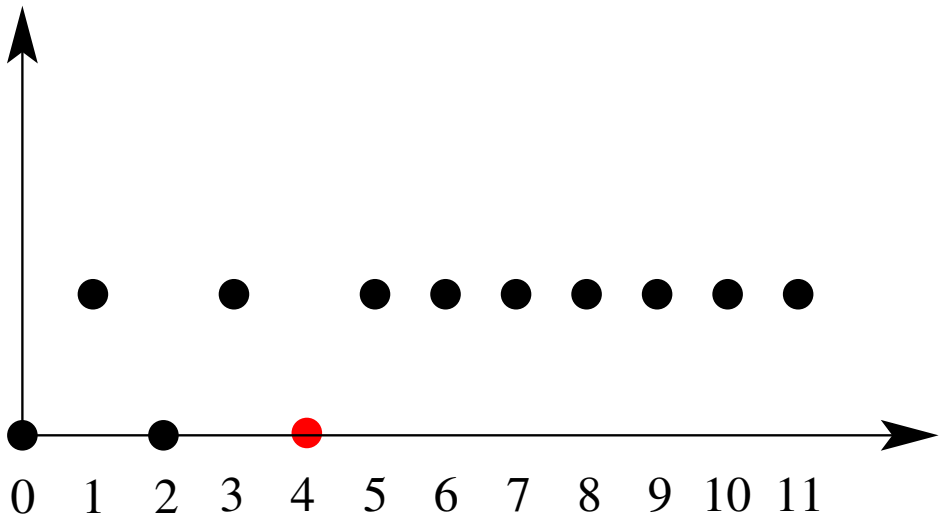
1. min(N morceaux, M morceaux) $\leq 2(N+M)$ morceaux, donc $\ell(N) \leq 2(\ell(N/2)+\ell(N/2)) = 4\ell(N/2) \rightarrow \ell(N) \leq N^2$.
2. $\ell(N) \leq \lambda_3(N) \leq 3N\alpha(N)$, où $\lambda_3(N)$= plus grand mot sur N lettres sans sous-séquence du type *abab* [Davenport, Schinzel, 1960].

Algorithmes [Herschberger, 1989] & [Nielsen, Yvinec, 1998]
$\rightarrow$ complexité $\mathcal{O}(N \log N)$.

Algorithme de clôture sous-additive

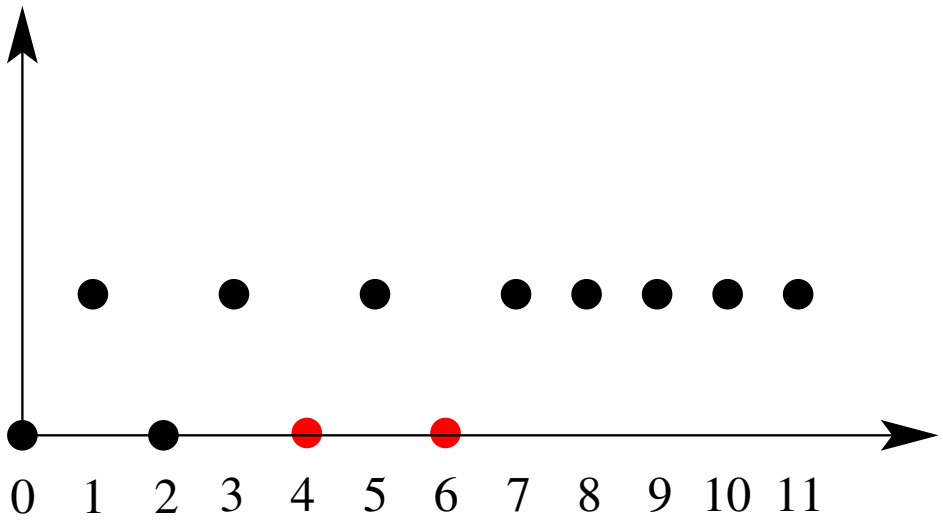


Algorithme de clôture sous-additive



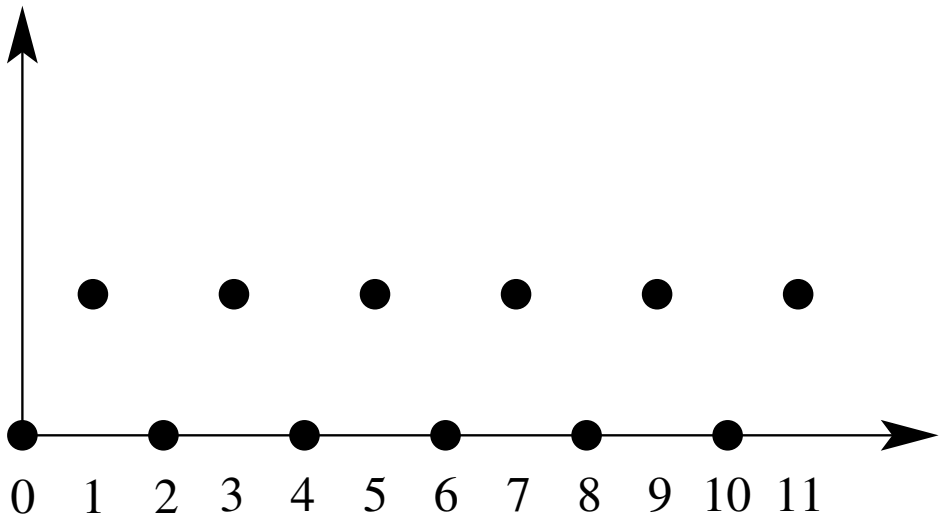
- $(f * f)(t) = \inf_{0 \leq s \leq t} f(s) + f(t-s) = \inf_{t_1+t_2=t} f(t_1) + f(t_2)$.

Algorithme de clôture sous-additive



- $(f * f)(t) = \inf_{0 \leq s \leq t} f(s) + f(t-s) = \inf_{t_1+t_2=t} f(t_1) + f(t_2)$.
- $(f * f * f)(t) = \inf_{t_1+t_2+t_3=t} f(t_1) + f(t_2) + f(t_3)$.

Algorithme de clôture sous-additive



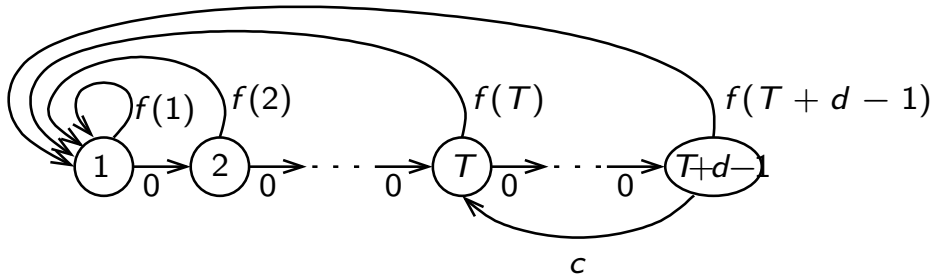
- $(f * f)(t) = \inf_{0 \leq s \leq t} f(s) + f(t-s) = \inf_{t_1+t_2=t} f(t_1) + f(t_2)$.
- $(f * f * f)(t) = \inf_{t_1+t_2+t_3=t} f(t_1) + f(t_2) + f(t_3)$.
- ...
- $\nexists k,\ f^{(k)} = f^*$.

Algorithme de clôture sous-additive

Entrée : $f \in \mathcal{D}$, $\forall t \geq T$, $f(t+d) = f(t) + c$.

Calcul de f^* :

$f^*(t) = \inf_{k \geq 0} f^{(k)}(t) = \inf_{t_1+t_2+\cdots+t_k=t} (f(t_1) + f(t_2) + \cdots + f(t_k))$.



Puissance de matrice : $f^*(t) = M^t_{0,0}$ où M matrice du graphe valué et produit de matrice dans $(\min, +)$ $((PQ)_{i,j} = \min_k(P_{i,k} + Q_{k,j}))$.

Théorème (Cohen,Dubois,Quadrat,Viot, 1985)

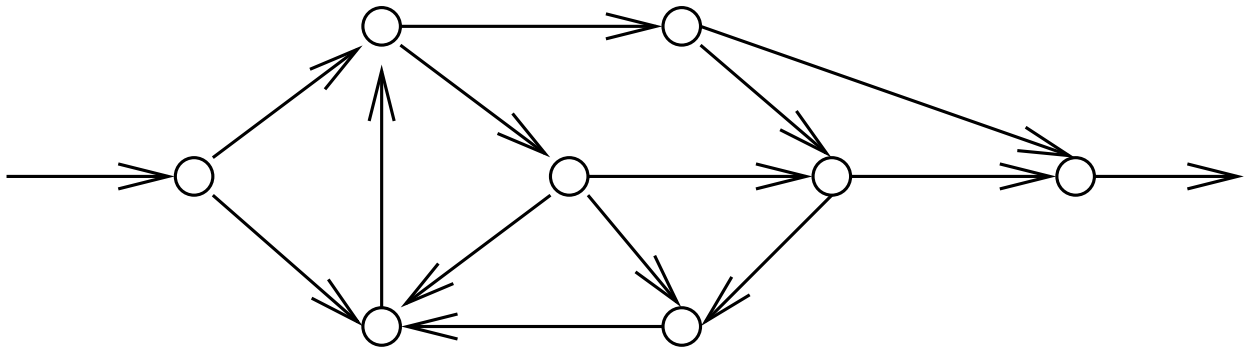
Pour M irréductible, il existe $d^*, T^* \in \mathbb{N}$ et $\lambda \in \mathbb{R}$ tels que $\forall t \geq T^*$, $M^{t+d^*} = M^t + \lambda d^*$.

$\rightarrow \forall t \geq T^*$, $f^*(t + d^*) = f^*(t) + c^*$ avec $c^* = \lambda d^*$.

$\rightarrow$ Un algorithme : calculer incrémentalement les M^t (très coûteux).

Un problème de routage

Problème : étant donné un réseau (sans traffic) et un flux de source et destination connues, calculer une route qui donne la meilleur garantie sur le délai maximum (resp. sur la charge maximum).

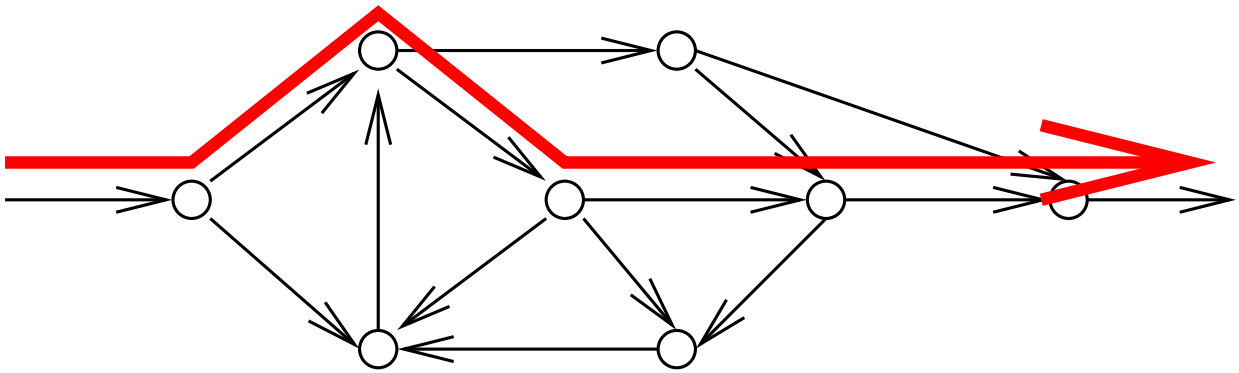


Entrées:

- le réseau sous forme d'un graphe $G = (V, A)$,
- un service minimum β_v à chaque nœud v,
- source x, destination y et courbe d'arrivée α pour le flux à router.
- courbes affines par morceaux, arrivée α concave, services β_v convexe.

Un problème de routage

Problème : étant donné un réseau (sans traffic) et un flux de source et destination connues, calculer une route qui donne la meilleur garantie sur le délai maximum (resp. sur la charge maximum).



Entrées:

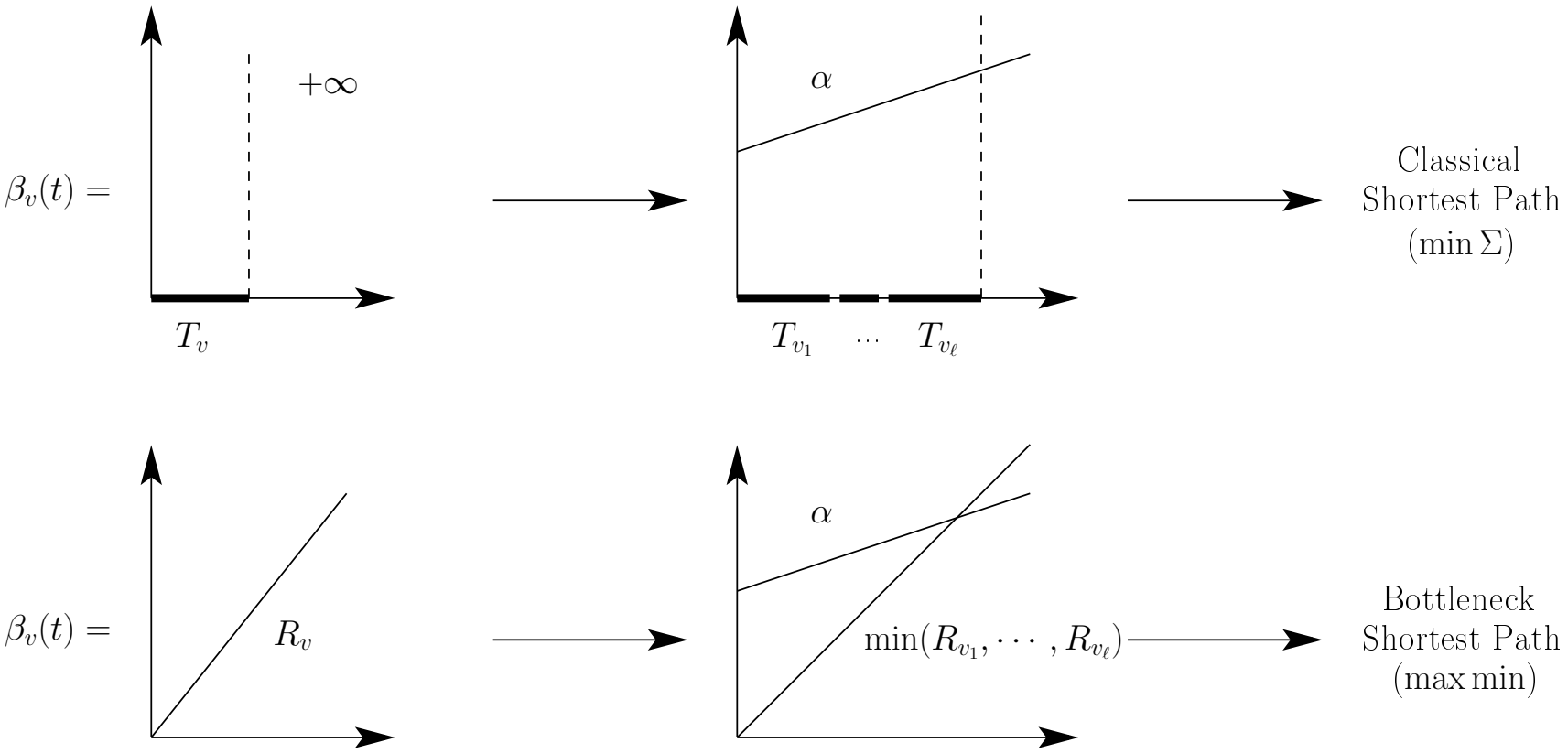
- le réseau sous forme d'un graphe $G = (V, A)$,
- un service minimum β_v à chaque nœud v,
- source x, destination y et courbe d'arrivée α pour le flux à router.
- courbes affines par morceaux, arrivée α concave, services β_v convexe.

Routage : forme de la function à optimiser

Optimisation de la borne sur les délais :

$$
\begin{aligned}
OPT &= \inf_{\substack{\textit{chemin} \\ (x, v_1, \ldots, v_\ell, y)}} h(\alpha, \beta_{v_1} * \cdots * \beta_{v_\ell}) \\
&= \inf_{\substack{\textit{chemin} \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \geq 0} \{(\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z)\}
\end{aligned}
$$

Routage : cas particuliers



Routage : forme de la function à optimiser

Optimisation de la borne sur les délais :

$$OPT = \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} h(\alpha, \beta_{v_1} * \cdots * \beta_{v_\ell})$$

Routage : forme de la function à optimiser

Optimisation de la borne sur les délais :

$$
\begin{aligned}
OPT &= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} h(\alpha, \beta_{v_1} * \cdots * \beta_{v_\ell}) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \geq 0} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right)
\end{aligned}
$$

Routage : forme de la function à optimiser

Optimisation de la borne sur les délais :

$$
\begin{aligned}
OPT &= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} h(\alpha, \beta_{v_1} * \cdots * \beta_{v_\ell}) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \geq 0} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \in C} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right)
\end{aligned}
$$

Routage : forme de la function à optimiser

Optimisation de la borne sur les délais :

$$
\begin{aligned}
OPT &= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} h(\alpha, \beta_{v_1} * \cdots * \beta_{v_\ell}) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \geq 0} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \in C} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \sup_{z \in C} \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right)
\end{aligned}
$$

Routage : forme de la function à optimiser

Optimisation de la borne sur les délais :

$$
\begin{aligned}
OPT &= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} h(\alpha, \beta_{v_1} * \cdots * \beta_{v_\ell}) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \geq 0} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \in C} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \sup_{z \in C} \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \sup_{z \in C} \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right)
\end{aligned}
$$

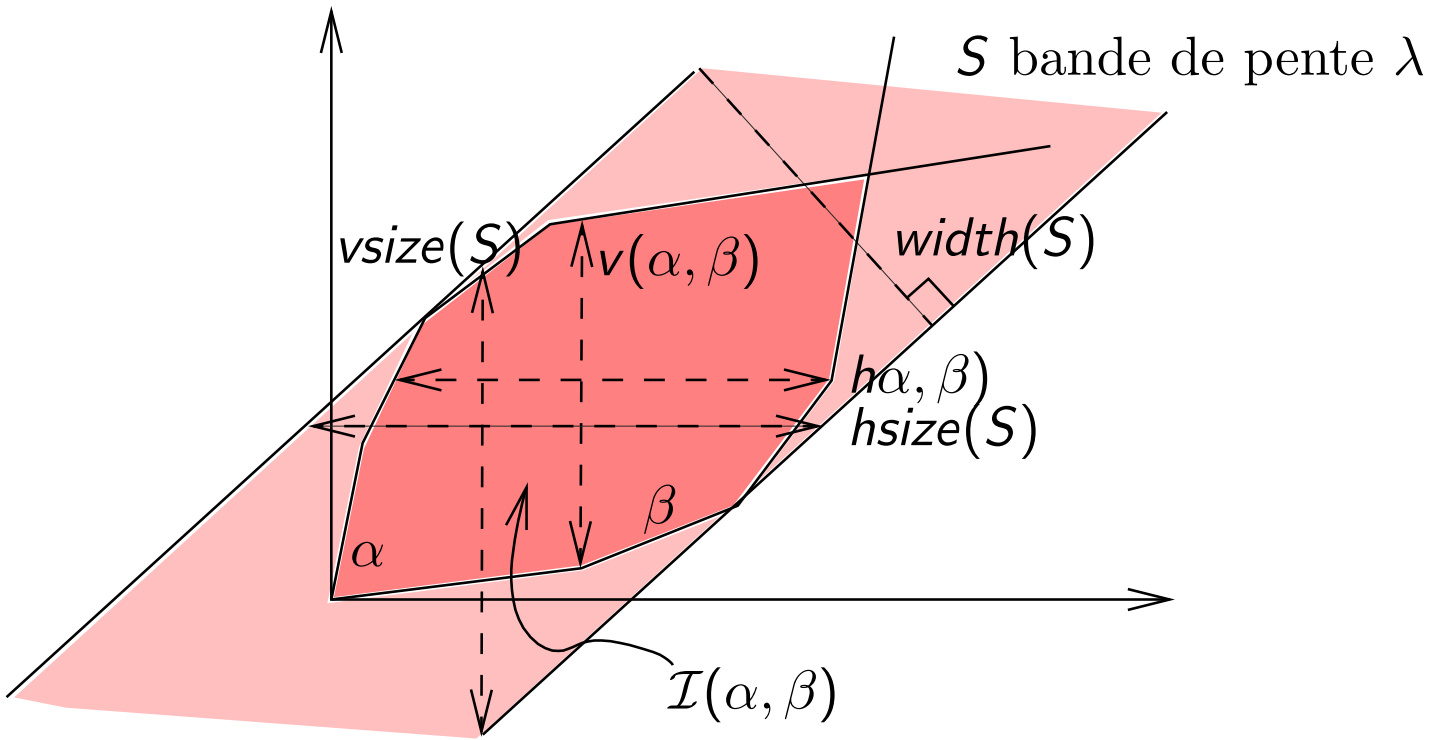
Optimisation de la borne sur les délais :

$$
\begin{aligned}
OPT &= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} h(\alpha, \beta_{v_1} * \cdots * \beta_{v_\ell}) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \geq 0} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \in C} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \sup_{z \in C} \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \sup_{z \in C} \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right)
\end{aligned}
$$

En fait,

$$
\neq \sup_{z \in C} \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right)
$$

Routage : un théorème min-max

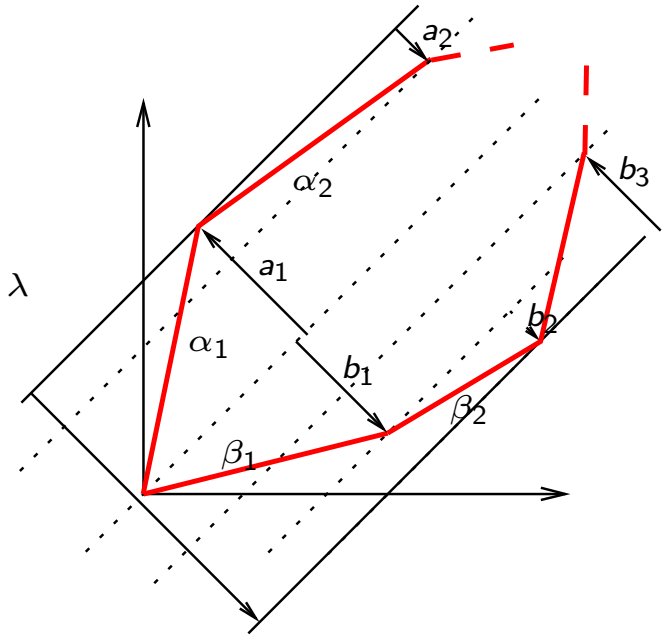


Théorème

Si α concave et β convexe, alors

$$\sup_{z\geq 0}\{\beta^{-1}(z)-\alpha^{-1}(z)\} = \inf_{\lambda\in C}\{hsize(S) \mid \mathcal{I}(\alpha,\beta)\subseteq S \text{ bande de pente } \lambda\}.$$

Routage : résolution



$$\inf\{hsize(S) \mid I(\alpha,\beta) \subseteq S,\ S \text{ bande de pente } \lambda\} = -proj_\lambda(\alpha) + \sum_{i=1}^{\ell} (proj_\lambda(\beta_{v_i}))_+$$

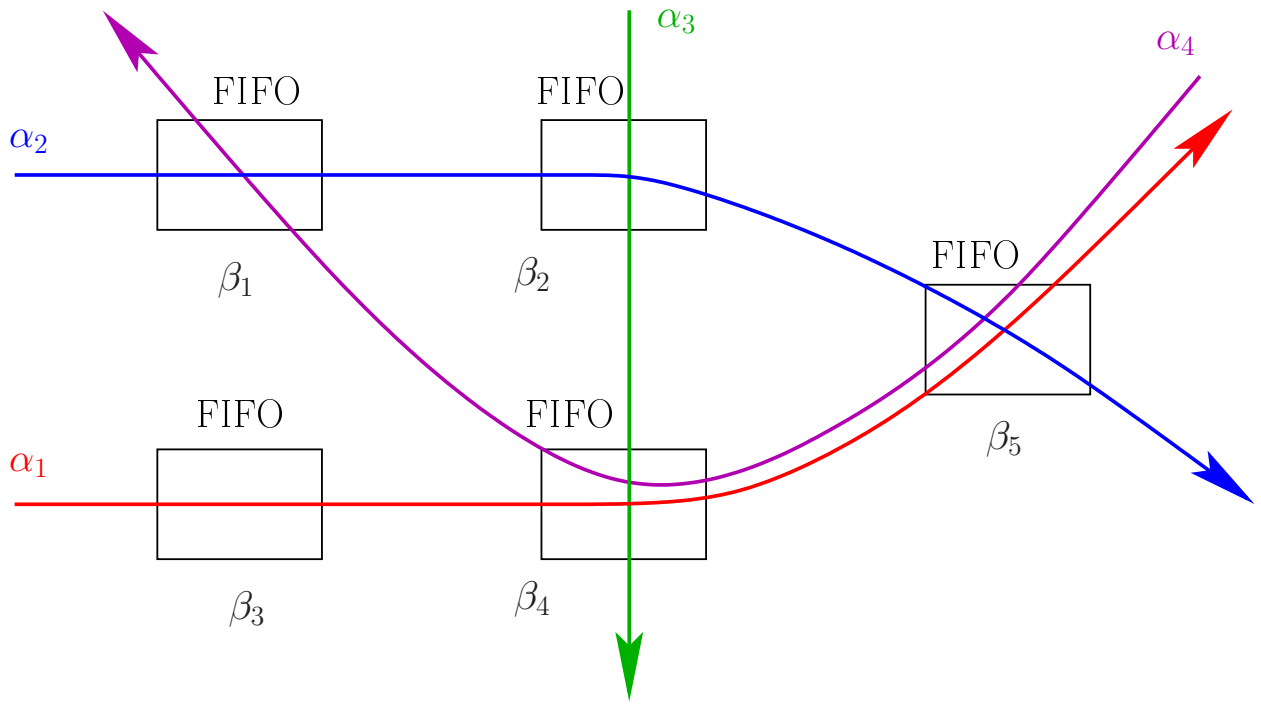
Routage : résolution

$$
\begin{aligned}
OPT &= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \sup_{z \geq 0} \left((\beta_{v_1} * \cdots * \beta_{v_\ell})^{-1}(z) - \alpha^{-1}(z) \right) \\
&= \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \inf_{\lambda \in C} \{ hsize(S) \mid I(\alpha, \beta) \subseteq S \text{ bande de pente } \lambda \} \\
&= \inf_{\lambda \in C} \inf_{\substack{chemin \\ (x, v_1, \ldots, v_\ell, y)}} \left(- proj_\lambda(\alpha) + \sum_{i=1}^{\ell} (proj_\lambda(\beta_{v_i}))_+ \right)
\end{aligned}
$$

Algorithme :

1. Pour toute pente λ du modèle, remplacer β_v par $(proj_\lambda(\beta_v))_+$ et calculer plus court chemin classique dans le graphe.
2. Minimum des valeurs calculées pour ces pentes λ.

Le problème de stabilité



Modèle : arrivées $\alpha_i(t) = \sigma_i + \rho_i t$, services stricts $\beta_v = R_i(t - T_i)_+$, liens avec délais exacts et capacités, FIFO.

Stabilité : quantité de données dans réseau = bornée ?

Critère important : facteur d'utilisation

$$\nu = \max_{\text{serveur } v} \frac{\sum_{i \text{ passant par } v} \rho_i}{R_v}.$$

Stabilité : configurations sans circuit

Une condition nécessaire de stabilité : $\nu \leq 1$.

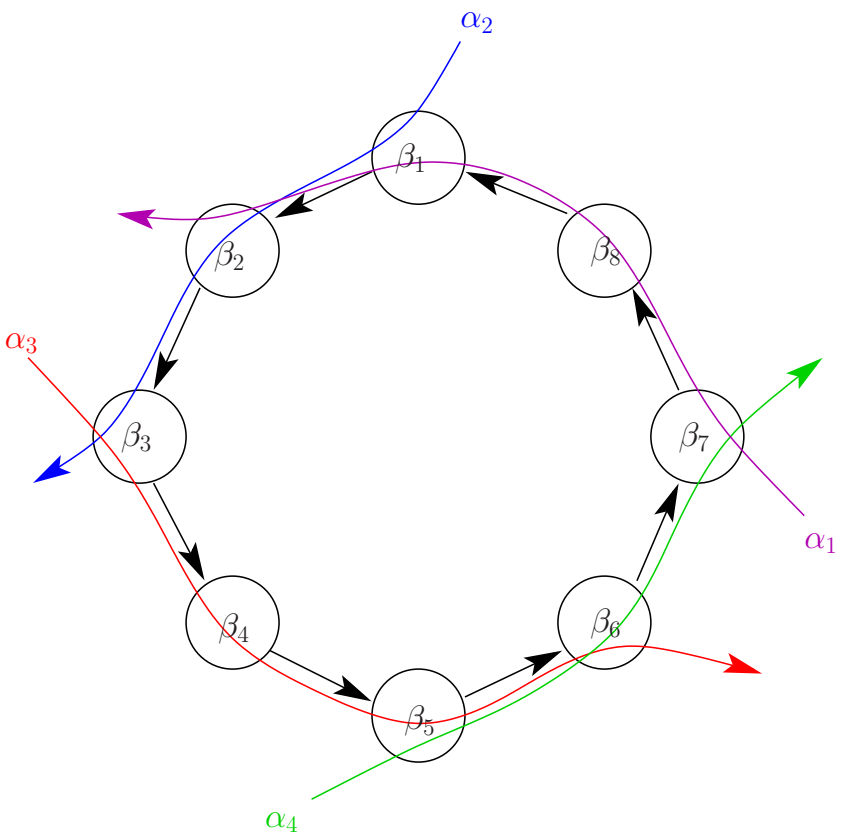
Théorème

Si les flux ne forment pas de circuit, alors le modèle est stable si $\nu < 1$.

Stabilité : l'anneau

Théorème (Tassiulas,Georgiadis,1996)

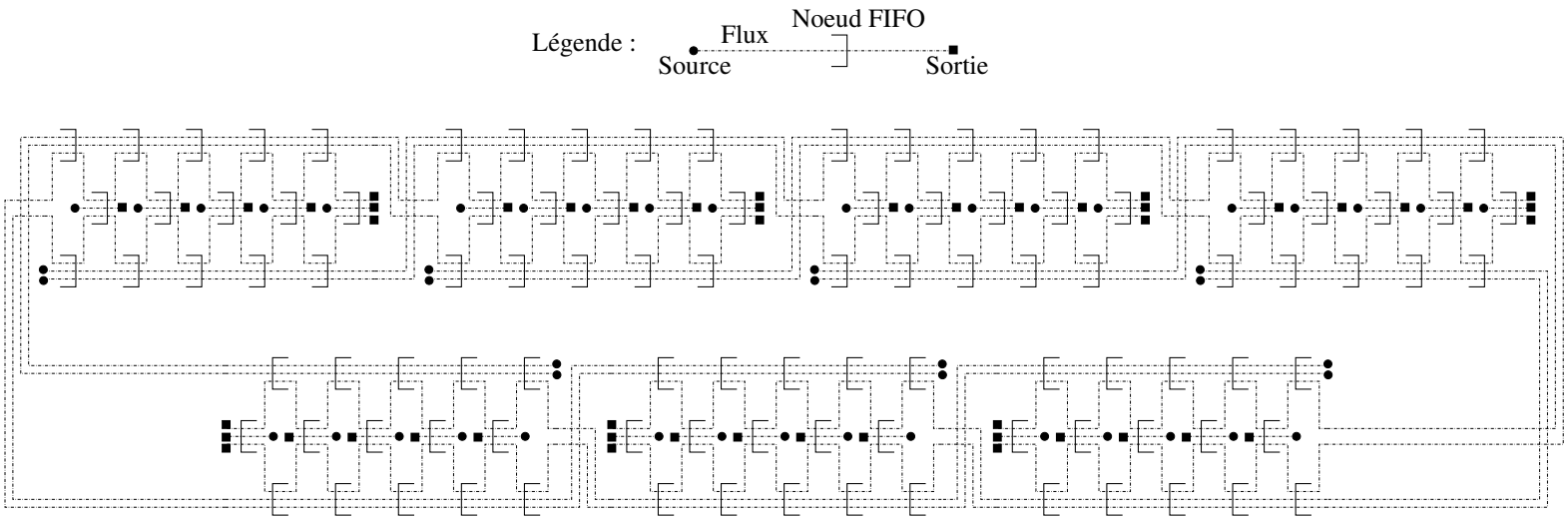
Si le réseau a une structure d'anneau (dirigé dans un sens), alors le modèle est stable si $\nu < 1$.



Stabilité : instabilité de FIFO

Théorème (Andrews,2000)

Il existe un modèle (réseau+traffic) instable de facteur d'utilisation $\nu = 1 - 3 \times 10^{-9}$.



Théorème (Andrews,2007)

Pour tout $r > 0$, *il existe un modèle (réseau+traffic) instable de facteur d'utilisation* $\nu < r$.

Stabilité : à ce jour

Résultat récent :[Le Boudec,Rizzo,2008]

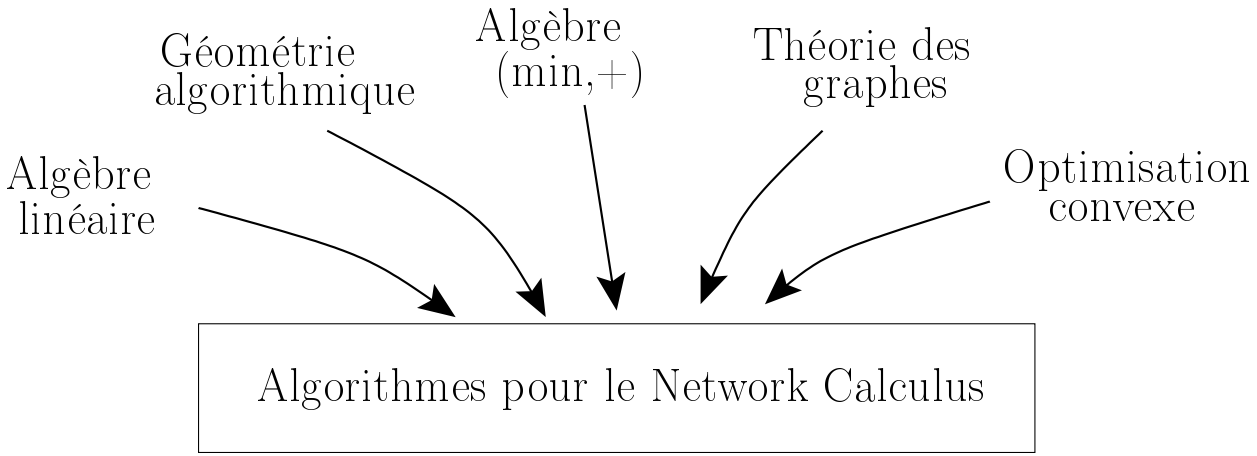
- Algorithme calculant une borne sur ν garantissant la stabilité.
- Possibilités de calculs de bornes sur les délais.
- Introduction d'un opérateur décrivant la dynamique du modèle, borné ensuite par un autre opérateur linéaire dans $(+, \times)$.

Question ouverte : pour ce genre de modèle de réseau & traffic, stabilité $=$ décidable ?

Logiciels

- DISCO Network Calculator (Java) [Schmitt et al, 2006].
- RTC - Real Time Calculus Toolbox (Java/Matlab) [Thiele et al, 2006].
- CyNC - Cyclic Network Calculus (Matlab/Simulink) [Schioler et al, 2005].
- Séries formelles $\gamma\delta$ (C++/Scilab) [Gaubert, Hardouin et al, 2007]
- COINC - COmputational Issues in Network Calculus (C++/Scilab) [Gaujal et al, 2007].

Conclusion



- Automatisation de l'analyse.
- Précision des bornes.
- Introduction de nouveaux opérateurs/algorithmes, au delà des opérations $(\min, +)$ élémentaires.

Petite bibliographie

- Network Calculus
  - J.-Y. Le Boudec & P. Thiran, Network Calculus: A Theory of Deterministic Queuing Systems for the Internet, LNCS 2050, Springer (2001).
  - C.-S. Chang, Performance Guarantees in Communication Networks, TNCS, Springer (2000).
- Max Plus Theory
  - MaxPlus Algebra Home Page (`www-rocq.inria.fr/MaxplusOrg/`).
- Adversarial Queuing Theory
  - Bibliographie de Matthew Andrews, Bell Labs.