Optimizing Admission Control in Balanced Networks

Matthieu Jonckheere[1],
Joint work with Jean Mairesse [2]

[1]Technical University of Eindhoven, [2]LIAFA, CNRS, Paris

YEQT III, november 2009

Virtues of the Erlang formula

- Robustness : insensitivity to fine traffic statistical properties.
- Computationally simple:

Recursive formula

Consider a birth and death process on $\{0, 1, \ldots, y\}$ with birth rates ν and death rates $\phi(x)$, then the blocking probability (probability to be in state y) can be recursively evaluated as follows:

$$B(y)^{-1} = 1 + \frac{\phi(y)}{\nu} B(y-1)^{-1}. \quad (1)$$

Virtues of the Erlang formula

- Robustness : insensitivity to fine traffic statistical properties.
- Computationally simple:

Recursive formula

Consider a birth and death process on $\{0, 1, \ldots, y\}$ with birth rates ν and death rates $\phi(x)$, then the blocking probability (probability to be in state y) can be recursively evaluated as follows:

$$B(y)^{-1} = 1 + \frac{\phi(y)}{\nu} B(y-1)^{-1}. \quad (1)$$

Following Erlang's steps

From a trunk to networks

- Loss networks, (Gibbens, Kelly, Ross...)
- Bandwidth sharing networks, (Roberts, Massoulie, Bonald, Proutiere, Virtamo...)

We aim at obtaining performance evaluation formula and optimization tools for multi-class networks with admission control and load balancing, **restricting the set of policies** to the ones being:

- robust,
- recursively computable.

Following Erlang's steps

From a trunk to networks

- Loss networks, (Gibbens, Kelly, Ross...)
- Bandwidth sharing networks, (Roberts, Massoulie, Bonald, Proutiere, Virtamo...)

We aim at obtaining performance evaluation formula and optimization tools for multi-class networks with admission control and load balancing, **restricting the set of policies** to the ones being:

- robust,
- recursively computable.

Model

Network model

Network with a finite set of processor sharing nodes $\mathcal{I}$.
$\mathcal{I}$ is partitioned into finitely many non-empty subsets $\mathcal{I}_k, k \in \mathcal{K}$, each customer has a class which is an element of $\mathcal{K}$, and a customer of class k has to be served by one of the nodes in $\mathcal{I}_k$.

The process of the number of customers

X a continuous-time birth and death process, on a **finite, coordinate-convex** state space $\mathcal{X}$, with infinitesimal generator $Q = (q(x,y))_{x,y\in\mathcal{X}}$ given by: $\forall x \in \mathcal{X}$,

$$\begin{cases} q(x, x-e_i) = \phi_i(x) & \text{if } x - e_i \in \mathcal{X} \\ q(x, x+e_i) = \lambda_i(x) & \text{if } x + e_i \in \mathcal{X} \\ q(x,y) = 0 & \text{if } y \in \mathcal{X},\ y \neq x - e_i, x + e_i . \end{cases} \tag{2}$$

The scalars $\lambda_i(x)$ define the routing/admission policy

Model

Network model

Network with a finite set of processor sharing nodes $\mathcal{I}$.
$\mathcal{I}$ is partitioned into finitely many non-empty subsets $\mathcal{I}_k, k \in \mathcal{K}$, each customer has a class which is an element of $\mathcal{K}$, and a customer of class k has to be served by one of the nodes in $\mathcal{I}_k$.

The process of the number of customers

X a continuous-time birth and death process, on a **finite, coordinate-convex** state space $\mathcal{X}$, with infinitesimal generator $Q = (q(x,y))_{x,y\in\mathcal{X}}$ given by: $\forall x \in \mathcal{X}$,

$$\begin{cases} q(x, x - e_i) = \phi_i(x) & \text{if } x - e_i \in \mathcal{X} \\ q(x, x + e_i) = \lambda_i(x) & \text{if } x + e_i \in \mathcal{X} \\ q(x, y) = 0 & \text{if } y \in \mathcal{X},\ y \neq x - e_i, x + e_i\ . \end{cases} \tag{2}$$

The scalars $\lambda_i(x)$ define the routing/admission policy

Total routing intensity

Arrival intensities constraints

$$\forall k \in \mathcal{K}, \quad \sum_{i \in \mathcal{I}_k} \lambda_i(x) \le \nu_k \tag{3}$$

Maximum total routing intensity

The *intensity* $h : \mathcal{X} \longrightarrow \mathbb{R}_+^*$ of a routing is defined by

$$h(x) = \sum_{i \in \mathcal{I}} \lambda_i(x). \tag{4}$$

The *maximum routing intensity* $\nu : \mathcal{X} \longrightarrow \mathbb{R}_+^*$ is defined by

$$\nu(x) = \sum_{k \in \mathcal{K}} \nu_k \mathbf{1}_{\{\exists i \in \mathcal{I}_k, x + e_i \in \mathcal{X}\}} . \tag{5}$$

Clearly, the intensity h of any routing satisfies: $\forall x \in \mathcal{X}, \ h(x) \le \nu(x)$.

Total routing intensity

Arrival intensities constraints

$$\forall k \in \mathcal{K}, \quad \sum_{i \in \mathcal{I}_k} \lambda_i(x) \leq \nu_k \tag{3}$$

Maximum total routing intensity

The *intensity* $h : \mathcal{X} \longrightarrow \mathbb{R}_+^*$ of a routing is defined by

$$h(x) = \sum_{i \in \mathcal{I}} \lambda_i(x). \tag{4}$$

The *maximum routing intensity* $\nu : \mathcal{X} \longrightarrow \mathbb{R}_+^*$ is defined by

$$\nu(x) = \sum_{k \in \mathcal{K}} \nu_k \mathbf{1}_{\{\exists i \in \mathcal{I}_k, x+e_i \in \mathcal{X}\}} \,. \tag{5}$$

Clearly, the intensity h of any routing satisfies: $\forall x \in \mathcal{X},\ h(x) \leq \nu(x)$.

Robustness

Reversibility conditions

We suppose that the service rates are given and balanced:

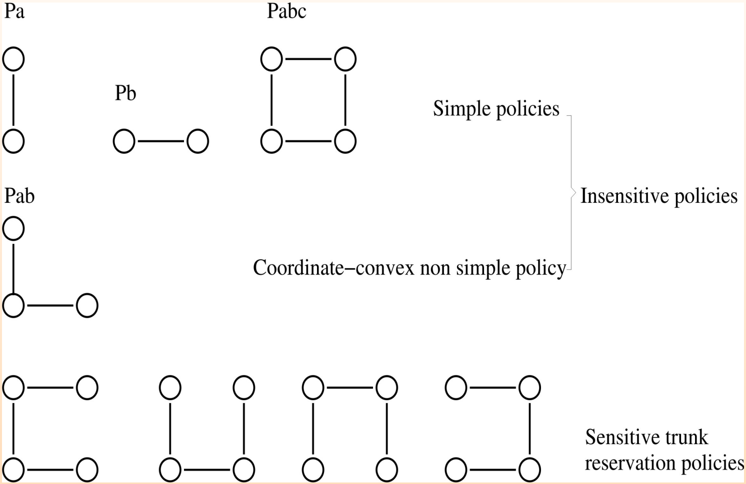
$$\phi_i(x) = \frac{\Phi(x - e_i)}{\Phi(x)} > 0.$$

Then the network is insensitive to the service distribution if and only if:

$$\lambda_i(x) = \frac{\Lambda(x + e_i)}{\Lambda(x)}.$$

We aim at finding a (sub)-optimal Λ.

A 2classes / four states example



Robustness

Characterization of the balance function

A 'robust' policy π can be characterized by:

- Its state space $\mathcal{X}^\pi \subset \mathcal{X}$.
- Its routing intensity h:

$$\Lambda(x) = \frac{1}{h(x)} \sum_{i=1}^{N} \Lambda(x + e_i).$$

Outline

- Bounds using rectangular functions.
- Recursive formula.
- The case of one arrival process.
- The case of several arrival processes with admission control (only).

Basic functions

Rectangular balance functions

Define a policy having an hyper-rectangle $\{x \leq y\}$ as state space and routing intensity g.
The corresponding *rectangular* balance function $\tilde{\Lambda}^{y,g} : \mathbb{N}^{\mathcal{I}} \to \mathbb{R}_+$ associated with y and g is defined by:

$$\tilde{\Lambda}^{y,g}(x) = \begin{cases} 1 & \text{if } x = y \\ g(x)^{-1} \sum_{i \in \mathcal{I}} \tilde{\Lambda}^{y,g}(x + e_i) & \text{if } x \leq y, x \neq y \\ 0 & \text{otherwise} \end{cases} .$$

This policy is not necessarily admissible !
For instance if $g(x) = \nu(x), \forall x$.

Basic functions

Rectangular balance functions

Define a policy having an hyper-rectangle $\{x \leq y\}$ as state space and routing intensity g.
The corresponding *rectangular* balance function $\tilde{\Lambda}^{y,g} : \mathbb{N}^{\mathcal{I}} \to \mathbb{R}_+$ associated with y and g is defined by:

$$\tilde{\Lambda}^{y,g}(x) = \begin{cases} 1 & \text{if } x = y \\ g(x)^{-1} \sum_{i \in \mathcal{I}} \tilde{\Lambda}^{y,g}(x + e_i) & \text{if } x \leq y, x \neq y \\ 0 & \text{otherwise} \end{cases} .$$

This policy is not necessarily admissible !
For instance if $g(x) = \nu(x), \forall x$.

Recursive evaluation

Consider the restriction of a rectangular balance function to a set $\mathcal{Y} \subset \mathcal{X}$.

Proposition

The blocking probability of the policy associated with $\tilde{L}_{\mathcal{Y}}^{y,g}$ satisfies

$$B_p(L_{\mathcal{Y}}^{y,g}) = 1 - \frac{\sum_{j\in\mathcal{K}} p_j \nu_j^{-1} \sum_{i\in\mathcal{I}_j} P^i(y,g)}{C(y,g)} \,. \tag{6}$$

The quantities P^i and C can be computed using the recursive schemes:

$$C(y,g) = \mathbf{1}_{\{y\in\mathcal{X}\}} + \sum_i C(y-e_i,g)\phi_i(y)g(y-e_i)^{-1}, \tag{7}$$

$$P_j(y,g) = \phi_j(y)\mathbf{1}_{\{y-e_j\in\mathcal{X}\}} + \sum_i P_j(y-e_i,g)\phi_i(y)g(y-e_i)^{-1} \,. \tag{8}$$

Upper Bounds

For any balance function define the (weighted) blocking probability by:

$$B_p = \sum_{x \in \mathcal{X}} \pi(x) \sum_{k \in \mathcal{K}} p_k \left(1 - \frac{\sum_{i \in \mathcal{I}_k} \lambda_i(x)}{\nu_k}\right). \quad (9)$$

Theorem

For any 'robust' admissible policy π associated with a balance function Λ.

$$B_p(\Lambda) \geq \min_{y \in \mathcal{X}} B_p(\Lambda^{y,\nu}). \quad (10)$$

Proof. One can decompose any balance function Λ as:
$\Lambda = \sum_{y \in \mathcal{X}} c_y \Lambda^{y,\nu}$.
The blocking probability is a linear function of *normalized* balance functions. □

Upper Bounds

For any balance function define the (weighted) blocking probability by:

$$B_p = \sum_{x \in \mathcal{X}} \pi(x) \sum_{k \in \mathcal{K}} p_k \left(1 - \frac{\sum_{i \in \mathcal{I}_k} \lambda_i(x)}{\nu_k}\right). \qquad (9)$$

Theorem

For any 'robust' admissible policy π associated with a balance function Λ.

$$B_p(\Lambda) \geq \min_{y \in \mathcal{X}} B_p(\Lambda^{y,\nu}). \qquad (10)$$

Proof. One can decompose any balance function Λ as:
$\Lambda = \sum_{y \in \mathcal{X}} c_y \Lambda^{y,\nu}$.
The blocking probability is a linear function of *normalized* balance functions. □

Lower Bounds

Decentralized balance functions

The decentralized policies works as follows. Do not accept customers outside the region $y^{\downarrow}$. Inside the region $y^{\downarrow} \cap \mathcal{X}$, all possible customers are accepted, *except* in points $x \in y^{\downarrow} \cap \mathcal{X}$ such that

$$\exists k \in \mathcal{K}, \exists i, j \in \mathcal{I}_k, \quad x + e_i \in y^{\downarrow} \cap \mathcal{X},\ x + e_j \in y^{\downarrow} \cap \mathcal{X}^c. \tag{11}$$

These policies can be thought of as restriction (to the state space) of rectangular policies. Define g as the arrival intensity of such policies.

Lower bound

$$B_p(\Lambda^*) \leq \min_{y \in \mathcal{X}} B_p(\Lambda^{y,g}). \tag{12}$$

Lower Bounds

Decentralized balance functions

The decentralized policies works as follows. Do not accept customers outside the region $y^{\downarrow}$. Inside the region $y^{\downarrow} \cap \mathcal{X}$, all possible customers are accepted, *except* in points $x \in y^{\downarrow} \cap \mathcal{X}$ such that

$$\exists k \in \mathcal{K}, \exists i, j \in \mathcal{I}_k, \quad x + e_i \in y^{\downarrow} \cap \mathcal{X}, \; x + e_j \in y^{\downarrow} \cap \mathcal{X}^c. \tag{11}$$

These policies can be thought of as restriction (to the state space) of rectangular policies. Define g as the arrival intensity of such policies.

Lower bound

$$B_p(\Lambda^*) \leq \min_{y \in \mathcal{X}} B_p(\Lambda^{y,g}). \tag{12}$$

Outline

- Bounds using rectangular functions.
- Recursive formula.
- The case of one arrival process.
- The case of several arrival processes with admission control (only).

One class of arrivals

In that case, the rectangular balance functions with intensity $\nu(\cdot)$ are **admissible** and extremal. Hence, the upper bound and the lower bound defined previously coincide.

Recursive formula for the **optimal** blocking probability

$$(B(\Lambda^y))^{-1} = 1 + \sum_{i \in \mathcal{I}} \frac{\phi_i(y)}{\nu} (B(\Lambda^{y-e_i}))^{-1}\,, \tag{13}$$

One class of arrivals

In that case, the rectangular balance functions with intensity $\nu(\cdot)$ are **admissible** and extremal. Hence, the upper bound and the lower bound defined previously coincide.

Recursive formula for the **optimal** blocking probability

$$(B(\Lambda^y))^{-1} = 1 + \sum_{i \in \mathcal{I}} \frac{\phi_i(y)}{\nu} (B(\Lambda^{y-e_i}))^{-1} , \tag{13}$$

Outline

- Bounds using rectangular functions.
- Recursive formula.
- The case of one arrival process.
- The case of several arrival processes with admission control (only).

Coordinate convex balance functions

Ferrers set policies

A *Ferrers set* is a finite subset E of $\mathbb{N}^k$ such that:

$$\left[x \in E, x_i > 0\right] \implies x - e_i \in E \; .$$

Denote $\mathcal{F}(\mathcal{X})$ the set of Ferrers set contained in $\mathcal{X}$.

Definition

Consider a Ferrers set $\mathcal{C} \in \mathcal{F}(\mathcal{X})$. The *coordinate-convex* balance function associated with $\mathcal{C}$ is defined by,

$$\tilde{\Lambda}^{\mathcal{C}}(x) = \prod_i \nu_i^{x_i} \mathbf{1}_{x \in \mathcal{C}} \; .$$

Corresponds to a *coordinate-convex* policy: if $x + e_i \in \mathcal{C}$, then $\lambda_i(x) = \nu_i$, if $x + e_i \notin \mathcal{C}$, then $\lambda_i(x) = 0$.

Coordinate convex balance functions

Ferrers set policies

A *Ferrers set* is a finite subset E of $\mathbb{N}^k$ such that:

$$\left[x \in E, x_i > 0\right] \implies x - e_i \in E \, .$$

Denote $\mathcal{F}(\mathcal{X})$ the set of Ferrers set contained in $\mathcal{X}$.

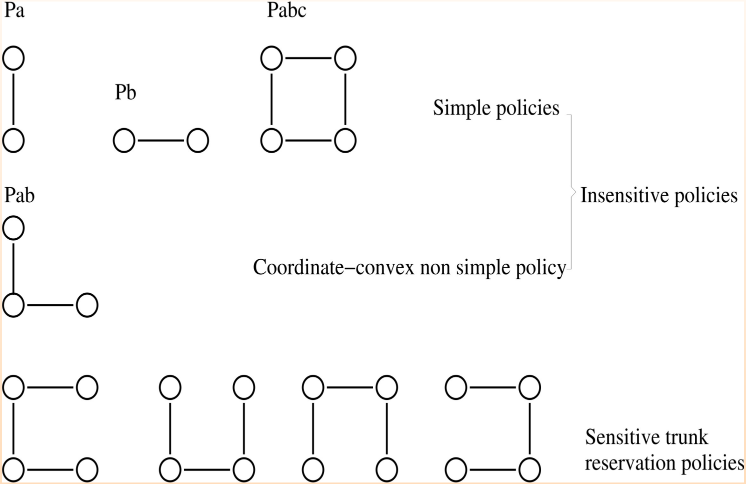
Definition

Consider a Ferrers set $\mathcal{C} \in \mathcal{F}(\mathcal{X})$. The *coordinate-convex* balance function associated with $\mathcal{C}$ is defined by,

$$\tilde{\Lambda}^{\mathcal{C}}(x) = \prod_i \nu_i^{x_i} \mathbf{1}_{x \in \mathcal{C}} \, .$$

Corresponds to a *coordinate-convex* policy: if $x + e_i \in \mathcal{C}$, then $\lambda_i(x) = \nu_i$, if $x + e_i \notin \mathcal{C}$, then $\lambda_i(x) = 0$.

A 2classes / four states example



Extremal balance functions

Extremal policies

Theorem

An admissible balance function Λ can be decomposed as:

$$\Lambda(x) = \sum_{\mathcal{C} \in \mathcal{F}(\mathcal{X})} \beta(\mathcal{C}) \Lambda^{\mathcal{C}}(x),$$

with $\beta(\mathcal{C}) \geq 0$ for all $\mathcal{C}$ and $\sum_{\mathcal{C} \in \mathcal{F}(\mathcal{X})} \beta(\mathcal{C}) = 1$.

Recursive evaluation

Recursion from $\mathcal{C} \cup \{x\}$ to $\mathcal{C}$

Lemma

Consider a Ferrers set $\mathcal{C} \in \mathcal{F}(\mathcal{X})$ and the corresponding coordinate-convex policy. For a point $x \notin \mathcal{C}$ such that $\mathcal{C} \cup \{x\}$ is also a Ferrers set, we have:

$$C(\mathcal{C} \cup \{x\}) = C(\mathcal{C}) + \tilde{\Lambda}_d(x)\Phi(x), \tag{14}$$

$$P_j(\mathcal{C} \cup \{x\}) = P_j(\mathcal{C}) + \tilde{\Lambda}_d(x)\Phi(x - e_j)\,. \tag{15}$$

Comparing $\mathcal{C}$ and $\mathcal{C} \cup \{x\}$

Lemma

Consider the coordinate-convex policy associated with $\mathcal{C} \in \mathcal{F}(\mathcal{X})$ and let x be a point such that $\mathcal{C} \cup \{x\} \in \mathcal{F}$. Let $X^{\mathcal{C}}$ be a r.v. distributed as the stationary number of customers. We have :

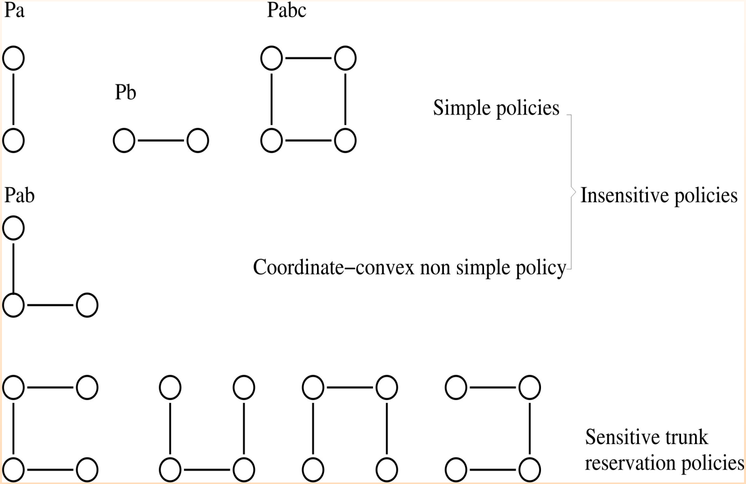
$$1 - B_p(\Lambda^{\mathcal{C}}) = \sum_{i \in \mathcal{I}} \frac{p_i}{\nu_i} E[\phi_i(X^{\mathcal{C}})].$$

Furthermore, the blocking probabilities satisfy:

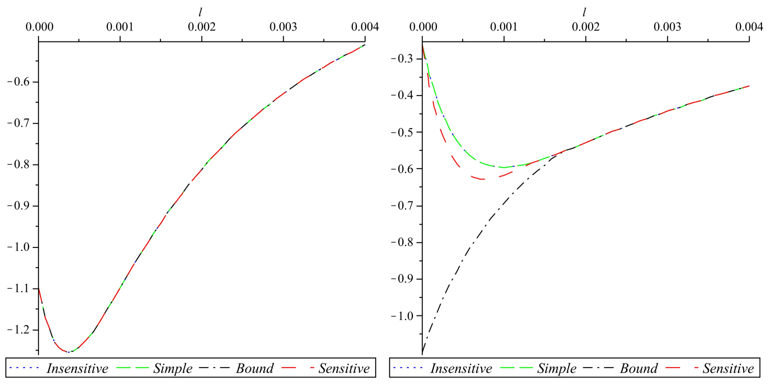
$$\left[B_p(\Lambda^{\mathcal{C} \cup \{x\}}) \leq B_p(\Lambda^{\mathcal{C}})\right] \iff \left[\sum_{i \in \mathcal{I}} \frac{p_i}{\nu_i} E[\phi_i(X^{\mathcal{C}})] \leq \sum_{i \in \mathcal{I}} \frac{p_i}{\nu_i} \phi_i(x)\right]. \quad (16)$$

Allows to give conditions under which a complete sharing is optimal.

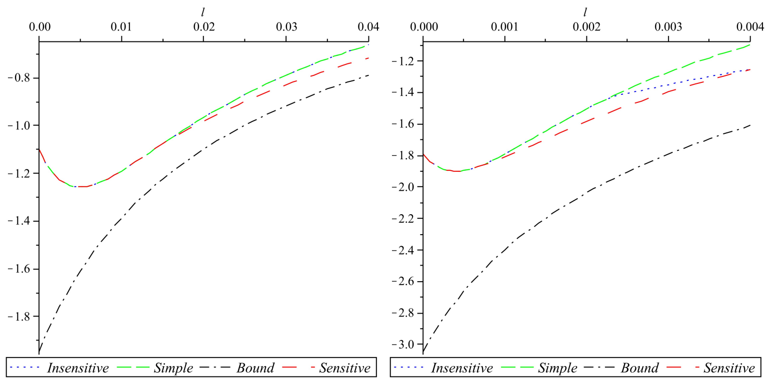
A 2classes / four states example



A 2classes / four states example



A 2classes / four states example



Conclusion

- The situation becomes more complex for several arrival processes...
- Generic structure of the optimal 'robust' policy still unknown.

- Computable bounds, tight for large networks at moderate loads.
- Complete characterization of extremal/optimal policies for networks with admission control only. Theoretical grip on the structures of optimal policies.

Open question

- When are decentralized policies optimal? (Leino&Virtamo examples).