Analysis of resource pooling games via a new extension of the Erlang loss function

Frank Karsten, Marco Slikker and Geert-Jan van Houtum

School of Industrial Engineering
Eindhoven University of Technology

Second Israeli-Dutch Workshop on Queueing Theory,
30-09-2010

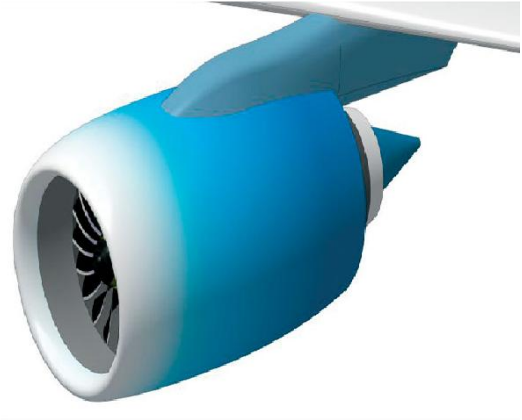
Motivation: resource pooling (example)

Medical departments in a hospital that require a clinical ward

Motivation: resource pooling (example)

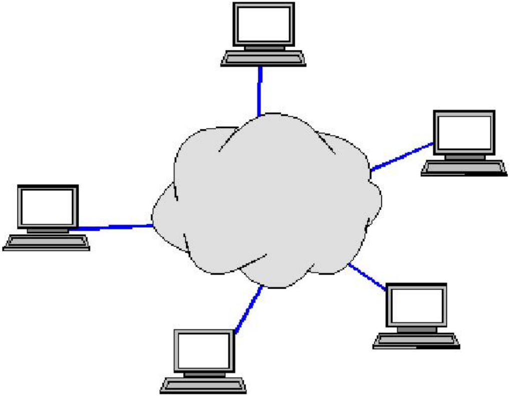
Call centers pooling their telephone operators

Motivation: resource pooling (example)



Airline companies pooling their inventories of spare engines

Motivation: resource pooling (example)



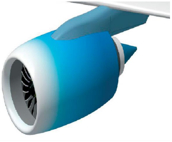
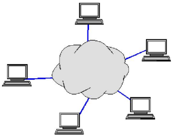
University departments sharing computing facilities

Motivation: resource pooling (example)

Several cities operating a joint fire department

Research question




- Several independent decision makers that may collaborate by pooling resources.
- How should they fairly distribute the joint costs of the pooled system amongst each other?

  $\Rightarrow$ Cooperative game theory

Related literature

Cooperative games in single-server queueing systems

- García-Sanz et al. (2008)
- Yu, Benjaafar & Gerchak (2009)
- Anily & Haviv (2009)

Outline

Setting

- There is a group of independent decision makers, the players.

Setting

- There is a group of independent decision makers, the players.
- Customers for each player arrive according to a Poisson process.

Setting

- There is a group of independent decision makers, the players.
- Customers for each player arrive according to a Poisson process.
- Arriving customers that find all servers occupied are lost.

Setting

- There is a group of independent decision makers, the players.
- Customers for each player arrive according to a Poisson process.
- Arriving customers that find all servers occupied are lost.
- Players can collaborate by fully pooling their servers and arrival streams.
    - Customers can be freely transferred between servers

Setting

- There is a group of independent decision makers, the players.
- Customers for each player arrive according to a Poisson process.
- Arriving customers that find all servers occupied are lost.
- Players can collaborate by fully pooling their servers and arrival streams.
  - Customers can be freely transferred between servers
- There is an infinite time horizon.

Setting

- There is a group of independent decision makers, the players.
- Customers for each player arrive according to a Poisson process.
- Arriving customers that find all servers occupied are lost.
- Players can collaborate by fully pooling their servers and arrival streams.
  - Customers can be freely transferred between servers
- There is an infinite time horizon.
- A cooperating group of players picks the cost-minimizing number of servers.

Resource pooling situations

A resource pooling situation is a tuple $(N, (\lambda_i)_{i \in N}, \mu, h, p)$, where

- N is the finite set of players
- λ_i is the arrival rate of customers of player $i \in N$
- $1/\mu$ is the expected service time
- h is the costs to hold one server per unit time
- p is the penalty costs incurred for a lost customer

Cooperative games

Cooperative cost game: (N, c)

- Player set N
  - A subset $M \subseteq N$ is called a coalition
  - $M = N$ refers to the grand coalition
- Characteristic cost function $c : 2^N \to \mathbb{R}$
  - Assigns to every coalition M its cost $c(M)$
  - $c(\emptyset) = 0$

Costs for a group of players

Erlang's loss function

The function $\pi_0 : \mathbb{N}_0 \times (0, \infty) \to [0, 1]$ with

$$\pi_0(s, a) = \frac{a^s/s!}{\sum_{y=0}^{s} a^y/y!}$$

gives the blocking probability in an M/G/*s*/*s* queue with load *a*.

Costs for a group of players

Erlang's loss function

The function $\pi_0 : \mathbb{N}_0 \times (0, \infty) \to [0, 1]$ with

$$\pi_0(s, a) = \frac{a^s/s!}{\sum_{y=0}^{s} a^y/y!}$$

gives the blocking probability in an M/G/s/s queue with load a.

Costs for a coalition

- Coalition $M \subseteq N$ with $\lambda_M = \sum_{i \in M} \lambda_i$
- Suppose they jointly use S servers
- Expected costs per unit time in steady state:

$$h \cdot S + \pi_0(S, \lambda_M/\mu) \cdot \lambda_M \cdot p$$

Definition (Resource pooling games)

Let $(N, (\lambda_i)_{i \in N}, \mu, h, p)$ be a resource pooling situation.
The game (N, c) with

$$c(M) = \min_{S \in \mathbb{N}_0} \{h \cdot S + \pi_0(S, \lambda_M/\mu) \cdot \lambda_M \cdot p\}$$

for all $M \subseteq N$ is called the associated resource pooling game.
An optimal number of servers for coalition M is denoted as S_M^*.

Example resource pooling game

Example

Situation:

- $N = \{1, 2, 3\}$
- $\lambda_1 = 1$
- $\lambda_2 = 2$
- $\lambda_3 = 3$
- $\mu = 1$
- $h = 2$
- $p = 3$

Associated resource pooling game and optimal number of servers:

Coalition M	S_M^*	$c(M)$
{1}	0	3
{2}	1	6
{3}	1	$8\frac{3}{4}$
{1,2}	1	$8\frac{3}{4}$
{1,3}	2	$11\frac{5}{13}$
{2,3}	3	$13\frac{223}{236}$
{1,2,3}	4	$16\frac{52}{155}$

Cost allocations

Definition (Core)

The core is the set of all allocations $(x_i)_{i \in N} \in \mathbb{R}^N$ that are

- Efficient: $\sum_{i \in N} x_i = c(N)$
- Stable: $\sum_{i \in M} x_i \leq c(M)$ for all $M \subseteq N$

Cost allocations

Definition (Core)

The core is the set of all allocations $(x_i)_{i \in N} \in \mathbb{R}^N$ that are

Efficient: $\sum_{i \in N} x_i = c(N)$

Stable: $\sum_{i \in M} x_i \leq c(M)$ for all $M \subseteq N$

Definition (Allocation proportional to arrival rates)

For a resource pooling situation $(N, (\lambda_i)_{i \in N}, \mu, h, p)$ we allocate

$$\mathcal{A}_i = c(N) \cdot \frac{\lambda_i}{\lambda_N}.$$

to each player $i \in N$.

Example resource pooling game

Example

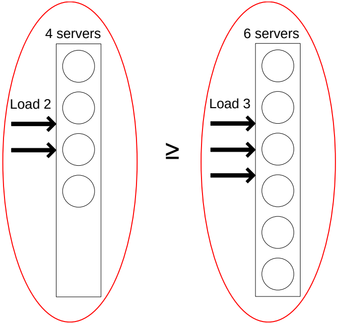
Situation:

- $N = \{1, 2, 3\}$
- $\lambda_1 = 1$
- $\lambda_2 = 2$
- $\lambda_3 = 3$
- $\mu = 1$
- $h = 2$
- $p = 3$

In this example, $\mathcal{A} = (2\frac{112}{155}, 5\frac{69}{155}, 8\frac{26}{155})$.
Is it stable (in the core)?

Coalition M	S^*_M	$c(M)$	$\sum_{i \in M} \mathcal{A}_i$
{1}	0	3	$2\frac{112}{155}$
{2}	1	6	$5\frac{69}{155}$
{3}	1	$8\frac{3}{4}$	$8\frac{26}{155}$
{1,2}	1	$8\frac{3}{4}$	$8\frac{26}{155}$
{1,3}	2	$11\frac{5}{13}$	$10\frac{138}{155}$
{2,3}	3	$13\frac{223}{236}$	$13\frac{95}{155}$
{1,2,3}	4	$16\frac{52}{155}$	$16\frac{52}{155}$

Scaling property of of the Erlang loss function



Theorem (Smith and Whitt, 1981)

$\pi_0(s_1, a) \geq \pi_0(s_2, a \cdot s_2/s_1)\ \forall\ a > 0$ *and* $s_1, s_2 \in \mathbb{N}_0$ *with* $s_1 \leq s_2$.

Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M)$$

$$\geq \qquad c(N)$$

Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \pi_0 \quad \left(S_M^*, \frac{\lambda_M}{\mu} \right) \qquad \cdot \lambda_N \cdot p$$

$$\geq \qquad c(N)$$

Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \pi_0 \quad \left(S_M^*, \frac{\lambda_M}{\mu} \right) \qquad \cdot \lambda_N \cdot p$$

$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \pi_0 \quad \left(\frac{\lambda_N}{\lambda_M} \cdot S_M^*, \frac{\lambda_N}{\mu} \right) \quad \cdot \lambda_N \cdot p \qquad c(N)$$

Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \pi_0 \quad \left(S_M^*, \frac{\lambda_M}{\mu} \right) \qquad \cdot \lambda_N \cdot p$$

$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \pi_0 \quad \left(\frac{\lambda_N}{\lambda_M} \cdot S_M^*, \frac{\lambda_N}{\mu} \right) \quad \cdot \lambda_N \cdot p \geq c(N)$$

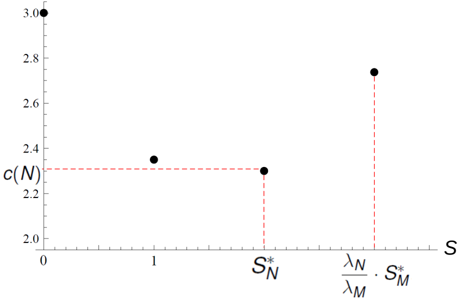
Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S^*_M + \pi_0 \left(S^*_M, \frac{\lambda_M}{\mu} \right) \cdot \lambda_N \cdot p$$

$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S^*_M + \pi_0 \left(\frac{\lambda_N}{\lambda_M} \cdot S^*_M, \frac{\lambda_N}{\mu} \right) \cdot \lambda_N \cdot p \geq c(N)$$

Costs for grand coalition N when using S servers



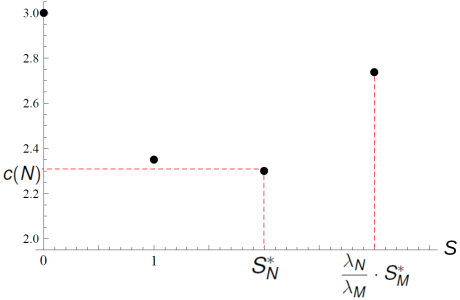
Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S^*_M + \pi_0 \left(S^*_M, \frac{\lambda_M}{\mu} \right) \cdot \lambda_N \cdot p$$

$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S^*_M + \pi_0 \left(\frac{\lambda_N}{\lambda_M} \cdot S^*_M, \frac{\lambda_N}{\mu} \right) \cdot \lambda_N \cdot p \geq c(N)$$

Costs for grand coalition N when using S servers



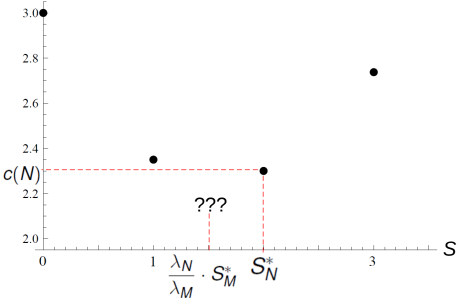
Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \pi_0 \left(S_M^*, \frac{\lambda_M}{\mu} \right) \cdot \lambda_N \cdot p$$

$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \pi_0 \left(\frac{\lambda_N}{\lambda_M} \cdot S_M^*, \frac{\lambda_N}{\mu} \right) \cdot \lambda_N \cdot p \geq c(N)$$

Costs for grand coalition N when using S servers



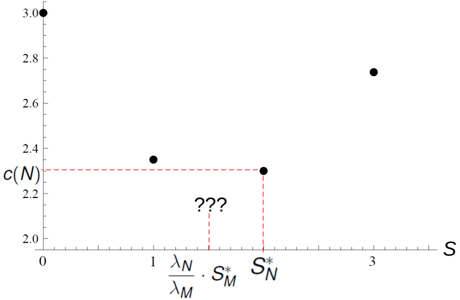
Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \quad X\left(S_M^*, \frac{\lambda_M}{\mu}\right) \quad \cdot \lambda_N \cdot p$$

$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + \quad X\left(\frac{\lambda_N}{\lambda_M} \cdot S_M^*, \frac{\lambda_N}{\mu}\right) \quad \cdot \lambda_N \cdot p \geq c(N)$$

Costs for grand coalition N when using S servers



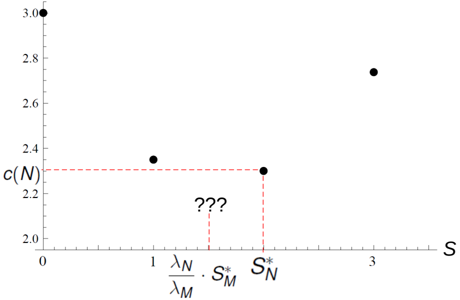
Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + X\left(S_M^*, \frac{\lambda_M}{\mu}\right) \cdot \lambda_N \cdot p$$

$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + X\left(\frac{\lambda_N}{\lambda_M} \cdot S_M^*, \frac{\lambda_N}{\mu}\right) \cdot \lambda_N \cdot p \geq c(N)$$

Costs for grand coalition N when using S servers



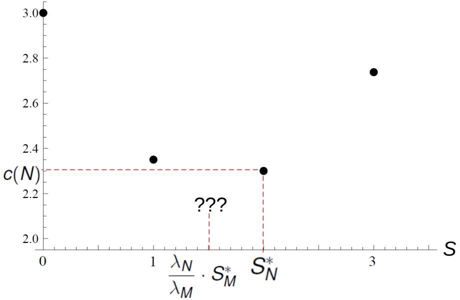
Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + X\left(S_M^*, \frac{\lambda_M}{\mu}\right) \cdot \lambda_N \cdot p$$

$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + X\left(\frac{\lambda_N}{\lambda_M} \cdot S_M^*, \frac{\lambda_N}{\mu}\right) \cdot \lambda_N \cdot p \geq c(N)$$

Costs for grand coalition N when using S servers



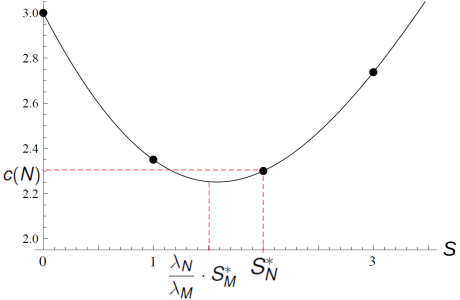
Proof methodology for core inclusion of $\mathcal{A}$

We have to show stability for each coalition $M \subseteq N$:

$$\frac{\lambda_N}{\lambda_M} \cdot c(M) = h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + X\left(S_M^*, \frac{\lambda_M}{\mu}\right) \cdot \lambda_N \cdot p$$

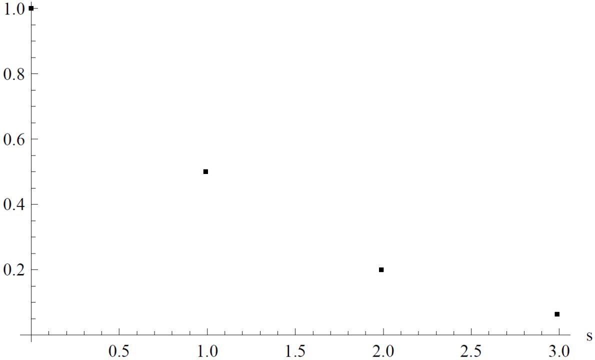
$$\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + X\left(\frac{\lambda_N}{\lambda_M} \cdot S_M^*, \frac{\lambda_N}{\mu}\right) \cdot \lambda_N \cdot p \geq c(N)$$

Costs for grand coalition N when using S servers



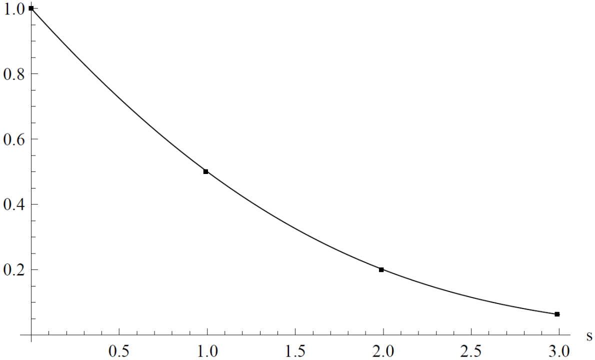
Known extensions of the Erlang loss function

- $\pi_0(s,a) = (a^s/s!)/(\sum_{y=0}^{s} a^y/y!)$, $s \in \mathbb{N}_0$



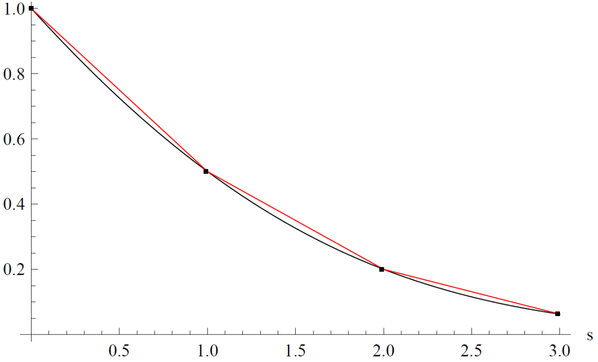
Known extensions of the Erlang loss function

- $\pi_0(s,a) = (a^s/s!)/(\sum_{y=0}^{s} a^y/y!),\ s \in \mathbb{N}_0$
- $B(s,a) = \left(a \int_0^\infty e^{-ax}(1+x)^s dx\right)^{-1}$



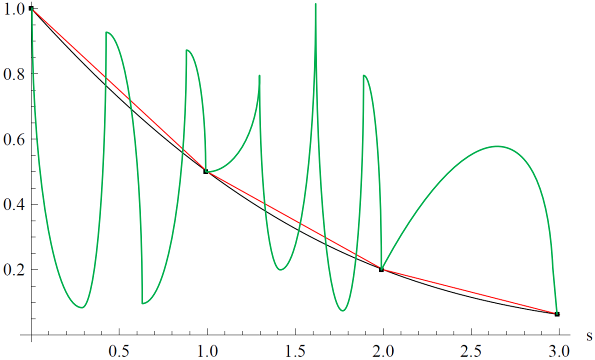
Known extensions of the Erlang loss function

- $\pi_0(s, a) = (a^s/s!)/(\sum_{y=0}^{s} a^y/y!),\ s \in \mathbb{N}_0$
- $B(s, a) = \left(a \int_0^\infty e^{-ax}(1 + x)^s dx\right)^{-1}$
- $L(s, a) = (1 - (s - \lfloor s \rfloor)) \cdot \pi_0(\lfloor s \rfloor, a) + (s - \lfloor s \rfloor) \cdot \pi_0(\lceil s \rceil, a)$

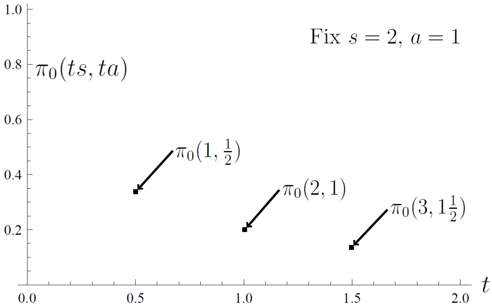


Known extensions of the Erlang loss function

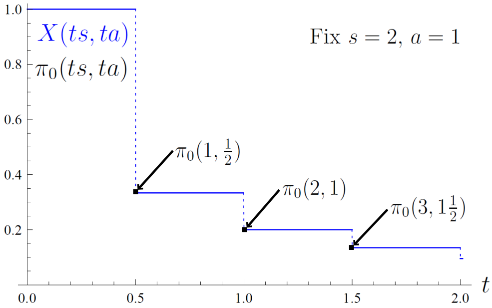
- $\pi_0(s,a) = (a^s/s!)/(\sum_{y=0}^{s} a^y/y!),\ s \in \mathbb{N}_0$
- $B(s,a) = \left(a \int_0^\infty e^{-ax}(1+x)^s dx\right)^{-1}$
- $L(s,a) = (1-(s-\lfloor s \rfloor)) \cdot \pi_0(\lfloor s \rfloor, a) + (s - \lfloor s \rfloor) \cdot \pi_0(\lceil s \rceil, a)$
- Nonsense$(s,a) = \ldots$



A new extension



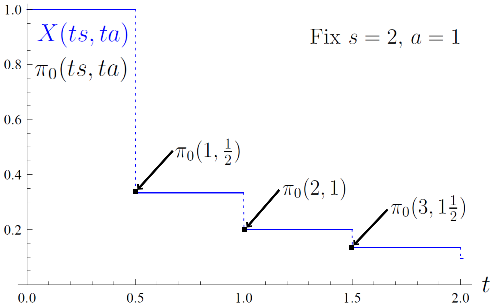
A new extension



A new extension

Define the function $X : [0, \infty) \times (0, \infty) \to [0, 1]$ with

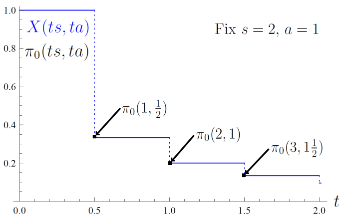
$$X(s, a) = \begin{cases} \pi_0(\lfloor s \rfloor, a \cdot \lfloor s \rfloor / s) & \text{if } s \geq 1; \\ 1 & \text{if } s \in [0, 1). \end{cases}$$



A new extension

Define the function $X : [0, \infty) \times (0, \infty) \to [0, 1]$ with

$$X(s,a) = \begin{cases} \pi_0(\lfloor s \rfloor, a \cdot \lfloor s \rfloor / s) & \text{if } s \geq 1; \\ 1 & \text{if } s \in [0,1). \end{cases}$$



Obvious properties of this function

Extension: $X(s,a) = \pi_0(s,a) \; \forall \; s \in \mathbb{N}_0, a > 0$

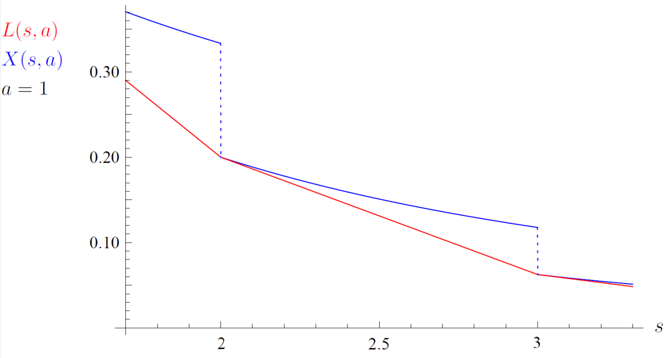
Scaling: $X(ts, ta)$ is non-increasing in $t \; \forall \; s \in [0, \infty), a > 0$

An important property

Theorem

$X(s,a) \geq L(s,a)$ for all $s \in [0,\infty)$ and $a > 0$.

Proof sketch

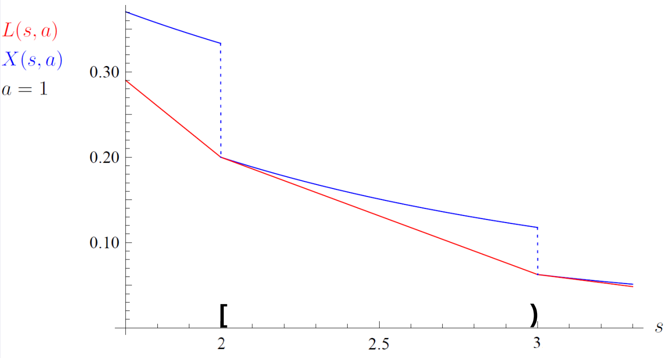


An important property

Theorem

$X(s, a) \geq L(s, a)$ for all $s \in [0, \infty)$ and $a > 0$.

Proof sketch

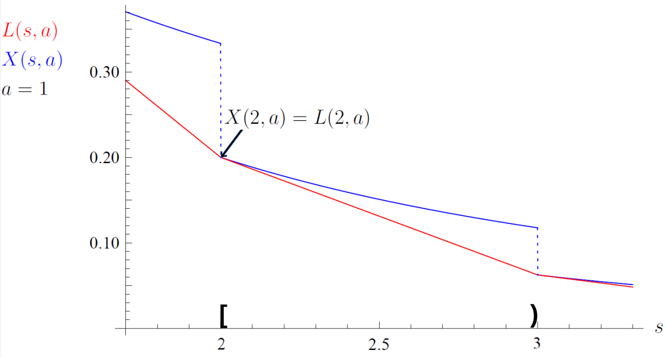


An important property

Theorem

$X(s, a) \geq L(s, a)$ for all $s \in [0, \infty)$ and $a > 0$.

Proof sketch

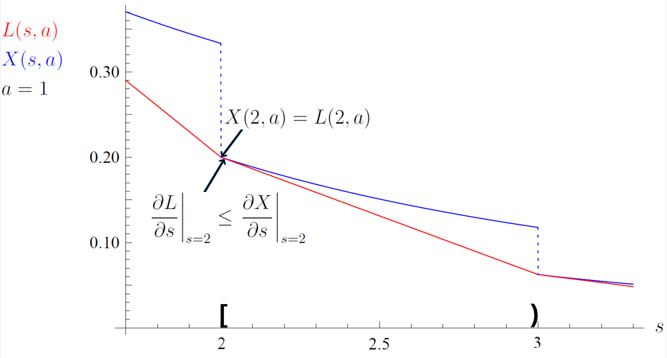


An important property

Theorem

$X(s, a) \geq L(s, a)$ for all $s \in [0, \infty)$ and $a > 0$.

Proof sketch

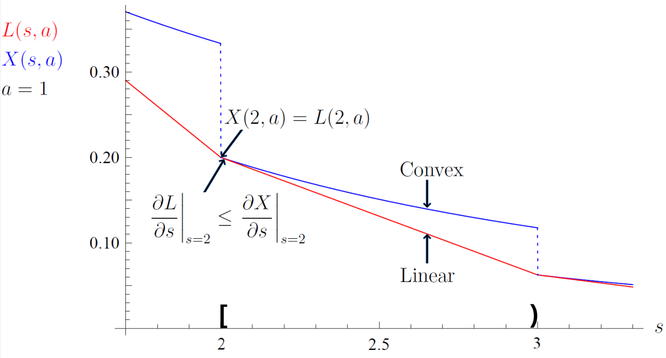


An important property

Theorem

$X(s,a) \geq L(s,a)$ for all $s \in [0,\infty)$ and $a > 0$.

Proof sketch



The main theorem

Theorem

Let $(N, (\lambda_i)_{i \in N}, \mu, h, p)$ be a resource pooling situation. Then $\mathcal{A}$ is in the core of the associated resource pooling game.

Proof.

As before,

$$\begin{aligned}
\frac{\lambda_N}{\lambda_M} \cdot c(M) &= h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + X\left(S_M^*, \frac{\lambda_M}{\mu}\right) && \cdot \lambda_N \cdot p \\
&\geq h \cdot \frac{\lambda_N}{\lambda_M} \cdot S_M^* + X\left(\frac{\lambda_N}{\lambda_M} \cdot S_M^*, \frac{\lambda_N}{\mu}\right) && \cdot \lambda_N \cdot p \geq c(N)
\end{aligned}$$

using the new extension X. □

Conclusion

Setting - several independent decision makers that

- face costs for servers and lost customers
- collaborate by optimizing and pooling their joint servers

Main results:

- A new extension of the Erlang loss function
- A cost allocation is in the core of a resource pooling game

Future research:

- Costly customer transfers or servers with limited skills
- Asymmetric penalty costs or service times
- Waiting or overflow rather than balking

Thank you for your attention

Analysis of resource pooling games via a new extension of the Erlang loss function

Frank Karsten, Marco Slikker and Geert-Jan van Houtum

School of Industrial Engineering
Eindhoven University of Technology

Second Israeli-Dutch Workshop on Queueing Theory,
30-09-2010