Aalto University
School of Electrical Engineering

On the Optimal Trade-off between SRPT and Opportunistic Scheduling

Samuli Aalto, Aleksi Penttinen, Pasi Lassila, Prajwal Osti
Aalto University, Finland

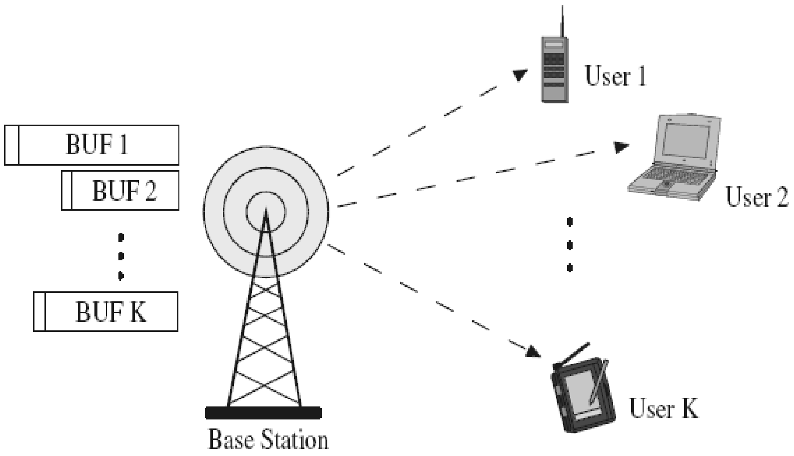
SIGMETRICS'11
7– 11 June 2011
San Jose, CA

Outline

- Introduction
- Optimal scheduling problem
- Solution
- Examples
- Lower and upper bounds
- Summary

Research problem

- Downlink data transmission in a cellular system
- Traffic = elastic flows
  - file transfers using TCP
- Scheduling decisions in each time slot
  - time scale of milliseconds
- Traffic dynamics in a longer time scale
  - time scale of seconds+
- Optimal scheduler for flow-level performance?



Source: Hu et al. (2004)

Flow-level performance

- Performance is expressed as throughput or flow delay
  - Mean flow delay would describe how long file transfers on the average last

- Importance of the time scale
  - Users do not care about delays of individual packets, but only about the total time to transmit a file of a given size

- Flow-level models try to characterize the system at the time-scale where users experience the performance

Schedulers

- Channel-aware schedulers
  - Scheduling based on channel information
  - Scheduler may prefer users with a good channel
  - Opportunistic scheduling
  - Examples: MR, PF

- Size-based schedulers
  - Scheduling based on flow size information
  - Scheduler may prefer users with a short flow
  - Example: SRPT

Fundamental trade-off

- Opportunistic scheduling
  - Select the user that has instantaneously good channel
  - Aggregate mean service rate increases with the number of users (opportunistic gain, multiuser diversity gain)
  - However, a user with a long remaining service time blocks the others

- SRPT
  - Select the user that has the least remaining service time
  - The number of flows is reduced most efficiently
  - However, opportunistic gain is lost due to suboptimal channel (later on also due to a smaller number of flows)

Combining opportunistic and size-based scheduling

- Tsybakov (2003)
  – Dynamic programming approach (time-slot scale)
- Hu et al. (2004)
  – Heuristic approach: TAOS (time-slot scale)
- Lassila and Aalto (2008)
  – Another heuristic approach: SRPT-P (time-slot scale)
- Sadiq and de Veciana (2010)
  – Time-scale separation (flow scale)
  – Transient system
  – Optimality result for nested polymatroids
  – Cf. optimality of SRPT-FM, Raj et al. (2004)

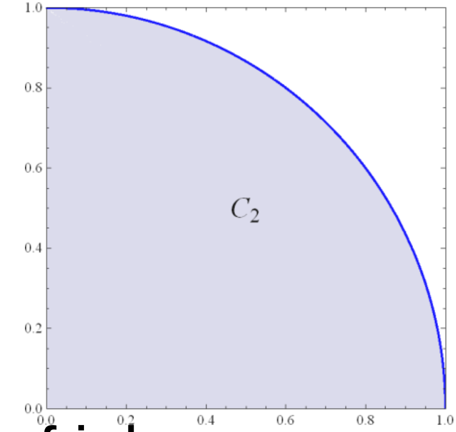
Our contribution

- Time-scale separation (flow scale)
  - In fact, abstract capacity regions
- Transient system
- Optimality result for compact and symmetric capacity regions
  - includes nested polymatroids
  - requires an implicit condition related to capacity regions
  - optimal policy applies the SRPT-FM principle
- Conservative upper bound for the mean delay

Outline

- Introduction
- Optimal scheduling problem
- Solution
- Examples
- Lower and upper bounds
- Summary

Model



- Service system where the service capacity is adjustable depending on the current number of jobs
- When there are k jobs with sizes

$$s_1 \geq \ldots \geq s_k$$

choose a rate vector

$$\mathbf{c}_k = (c_{k1}, \ldots, c_{kk}) \in C_k$$

and serve job i with rate c_{ki}

- Assume: Capacity regions C_k compact and symmetric

Transient system

- Assume that there are n jobs in the system at time 0
- What is the optimal way to make the system empty?
- Our objective: Minimize the mean delay (or flow time)
- Define: Flow time (or total completion time) for policy π

$$T^{\pi} = \sum_{i=1}^{n} t_i^{\pi}$$

where t_i is the completion time of job i

- Define: Operating policies

$$\Pi_n = \{\pi = (\mathbf{c}_1, \ldots, \mathbf{c}_n) : \mathbf{c}_k \in C_k \text{ for all } k\}$$

Outline

- Introduction
- Optimal scheduling problem
- Solution
- Examples
- Lower and upper bounds
- Summary

Trivial case: One job

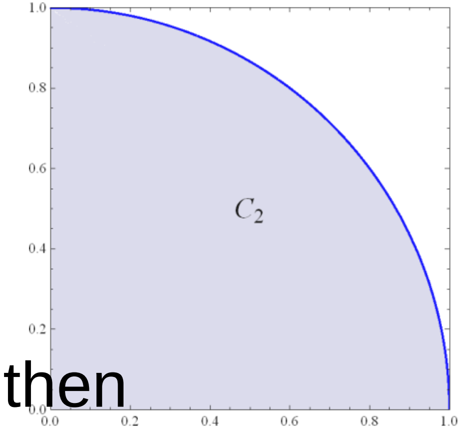
- Define:

$$G_1^* = \frac{1}{c_1^*}, \quad c_1^* = \max_{c_1 \in C_1} c_1$$

- Now

$$T^* = \min_{\pi \in \Pi_1} T^\pi = s_1 G_1^*, \quad \pi^* = (\mathbf{c}_1^*)$$

Simple case: Two jobs



- If job 2 (i.e., the shorter one) completes first, then

$$T^{\pi} = 2\frac{s_2}{c_{22}} + \left(s_1 - \frac{s_2}{c_{22}} c_{21}\right)\frac{1}{c_1^*} = \frac{s_2}{c_{22}}\left(2 - \frac{c_{21}}{c_1^*}\right) + \frac{s_1}{c_1^*}$$

- Otherwise

$$T^{\pi} = 2\frac{s_1}{c_{21}} + \left(s_2 - \frac{s_1}{c_{21}} c_{22}\right)\frac{1}{c_1^*} = \frac{s_1}{c_{21}}\left(2 - \frac{c_{22}}{c_1^*}\right) + \frac{s_2}{c_1^*}$$

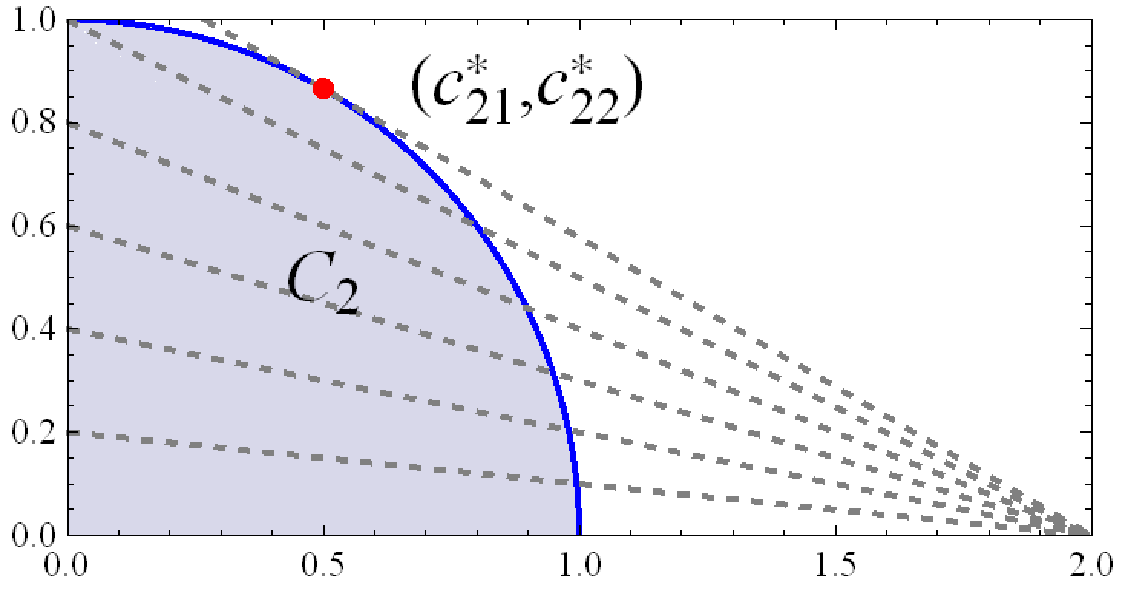
- Let us minimize (a function not depending on sizes!)

$$g(\mathbf{c}_2) = \frac{1}{c_{22}}\left(2 - \frac{c_{21}}{c_1^*}\right), \quad \mathbf{c}_2 \in C_2$$

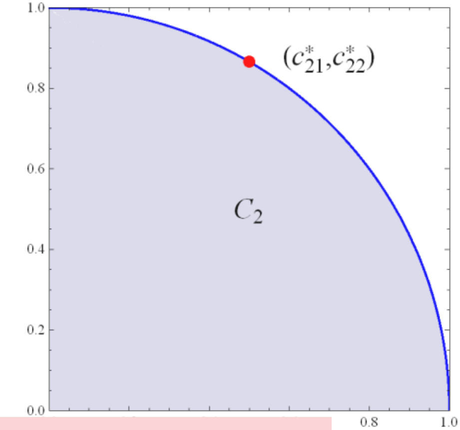
Aalto University
School of Electrical Engineering

Simple case: Two jobs (cont.)

- Geometric interpretation



Simple case: Two jobs (cont.)



- Define:

$$G_2^* = g(\mathbf{c}_2^*) = \min_{\mathbf{c}_2 \in C_2} g(\mathbf{c}_2)$$

- Result: Now **if**

$$G_1^* < G_2^*$$

then (due to the symmetry property!)

$$T^* = \min_{\pi \in \Pi_2} T^\pi = s_2 G_2^* + s_1 G_1^*, \; \pi^* = (\mathbf{c}_1^*, \mathbf{c}_2^*), \; c_{21}^* \le c_{22}^*$$

Simple case: Two jobs (cont.)

- Justification:

$$T^{\pi} \geq \min\{s_2 g(c_{21}, c_{22}) + s_1 G_1^*, s_1 g(c_{22}, c_{21}) + s_2 G_1^*\}$$

$$\geq \min\{s_2 G_2^* + s_1 G_1^*, s_1 G_2^* + s_2 G_1^*\}$$

$$= s_2 G_2^* + s_1 G_1^* \qquad [\text{since } G_2^* > G_1^*]$$

$$T^{\pi^*} = s_2 g(c_{21}^*, c_{22}^*) + s_1 G_1^* \qquad [\text{since } c_{22}^* \geq c_{21}^*]$$

$$= s_2 G_2^* + s_1 G_1^*$$

Simple case: Two jobs (cont.)

- Required additional result:

$$\frac{1}{c_{22}^{*}}\left(2-\frac{c_{21}^{*}}{c_{1}^{*}}\right) \leq \frac{1}{c_{21}^{*}}\left(2-\frac{c_{22}^{*}}{c_{1}^{*}}\right) \Leftrightarrow$$

$$c_{21}^{*}\left(2-\frac{c_{21}^{*}}{c_{1}^{*}}\right) \leq c_{22}^{*}\left(2-\frac{c_{22}^{*}}{c_{1}^{*}}\right) \Leftrightarrow$$

$$(c_{22}^{*}-c_{21}^{*})\left(2-\frac{c_{21}^{*}}{c_{1}^{*}}-\frac{c_{22}^{*}}{c_{1}^{*}}\right) \geq 0 \Leftrightarrow$$

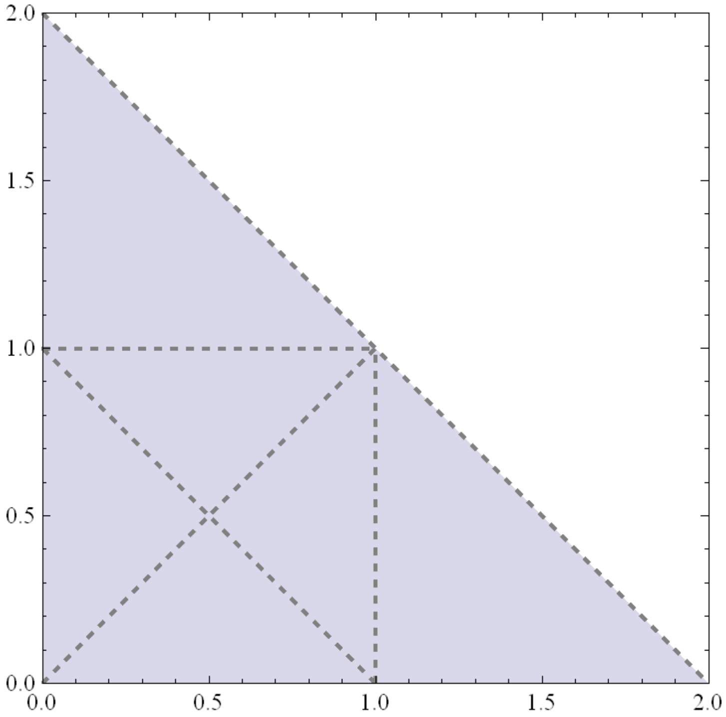
$$c_{22}^{*}(c_{22}^{*}-c_{21}^{*})(G_{2}^{*}-G_{1}^{*}) \geq 0$$

Simple case: Two jobs (cont.)

- Equivalent condition:

$$G_2^* > G_1^* \quad \Leftrightarrow$$
$$c_{21} + c_{22} < 2 \cdot c_1^*$$

- Suffient condition:
  nested capacity regions
- Note: However, capacity regions are not required to be nested



General case: n jobs

- Define (recursively):

$$G_k^* = \min_{\mathbf{c}_k \in C_k} g_k(\mathbf{c}_k), \quad g_k(\mathbf{c}_k) = \frac{1}{c_{kk}}\left(k - \sum_{i=1}^{k-1} c_{ki} G_i^*\right)$$

- Theorem 1: **If**

$$G_1^* < \ldots < G_n^*$$

then

$$T^* = \min_{\pi \in \Pi_2} T^\pi = \sum_{k=1}^{n} s_k G_k^*, \quad \pi^* = (\mathbf{c}_1^*, \ldots, \mathbf{c}_n^*)$$

General case: n jobs (cont.)

- In addition,

$$c^*_{k1} \le \ldots \le c^*_{kk} \quad \text{for all } k$$

- Thus, the optimal policy applies the SRPT-FM principle:
  - the shortest job is served with the highest rate,
  - the second shortest job is served with the second highest rate,
  - etc.

- Note also that the optimal rate vector does not depend on the absolute sizes (only on their order)

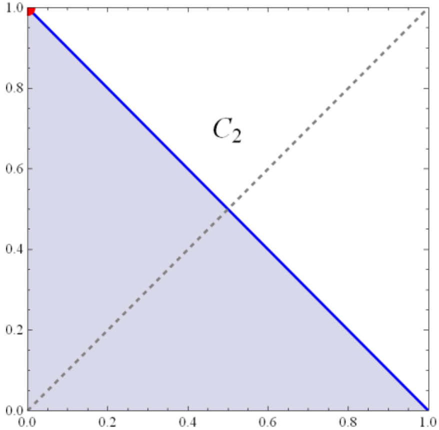
General case: n jobs (cont.)

- Necessary condition:

$$G_1^* < \ldots < G_k^* \quad \Rightarrow \quad c_{k1} + \ldots + c_{kk} < k \cdot c_1^*$$

Outline

- Introduction
- Optimal scheduling problem
- Solution
- Examples
- Lower and upper bounds
- Summary



Single-server queue

- Consider capacity regions

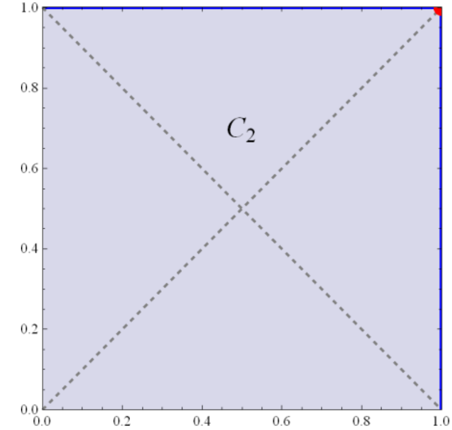
$$C_k = \{\mathbf{c}_k \geq 0 : \sum_{j=1}^{k} c_{kj} \leq 1\}$$

- Now

$$G_k^* = k \qquad \text{(increasing in } k\text{)}$$

$$c_{kj}^* = \begin{cases} 0, & j < k \\ 1, & j = k \end{cases} \qquad \text{(increasing in } j\text{)}$$

Infinite-server queue



- Consider capacity regions

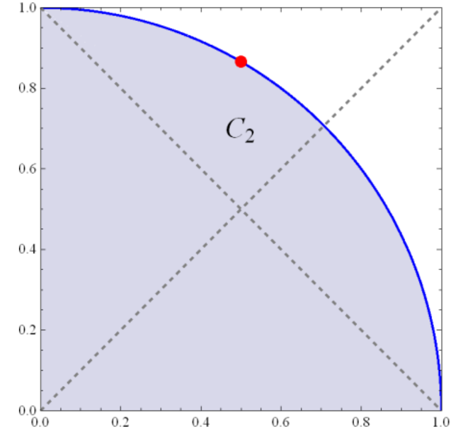
$$C_k = \{\mathbf{c}_k \geq 0 : \sum_{j=1}^{k} c_{kj} \leq k, c_{kj} \leq 1 \, \forall j\}$$

- Now

$$G_k^* = 1 \qquad \text{(constant)}$$

$$c_{kj}^* = 1 \qquad \text{(constant)}$$

Alpha-balls



- Let $\alpha > 1$ and consider capacity regions

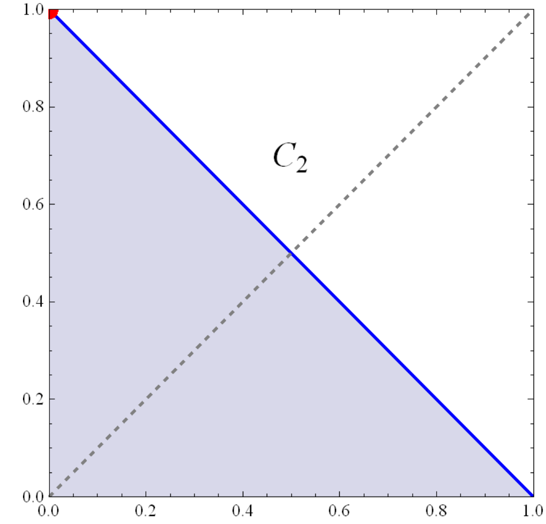
$$C_k = \{\mathbf{c}_k \geq 0 : \sum_{j=1}^{k} c_{kj}^{\alpha} \leq 1\}$$

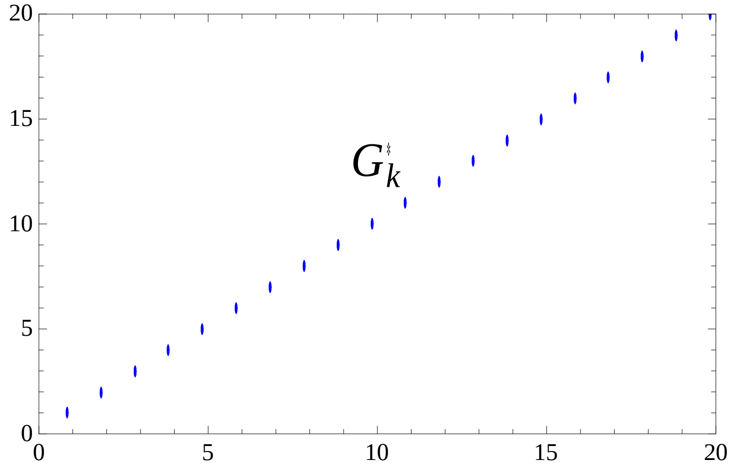
- Now

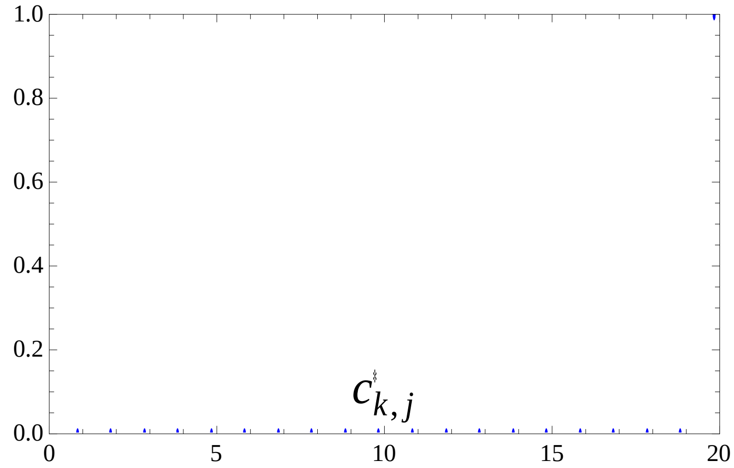
$$G_k^* = \left(k^{\frac{\alpha}{\alpha-1}} - (k-1)^{\frac{\alpha}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \quad \text{(increasing in } k\text{)}$$

$$c_{kj}^* = \left(\frac{G_j^*}{k} \right)^{\frac{1}{\alpha-1}} \quad \text{(increasing in } j\text{)}$$

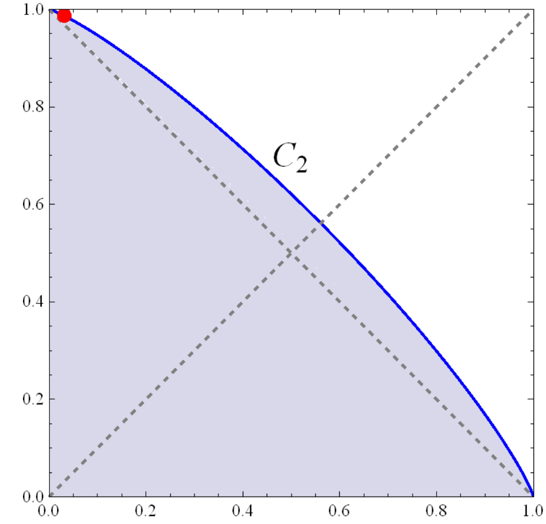
Alpha = 1.0
(single-server queue)



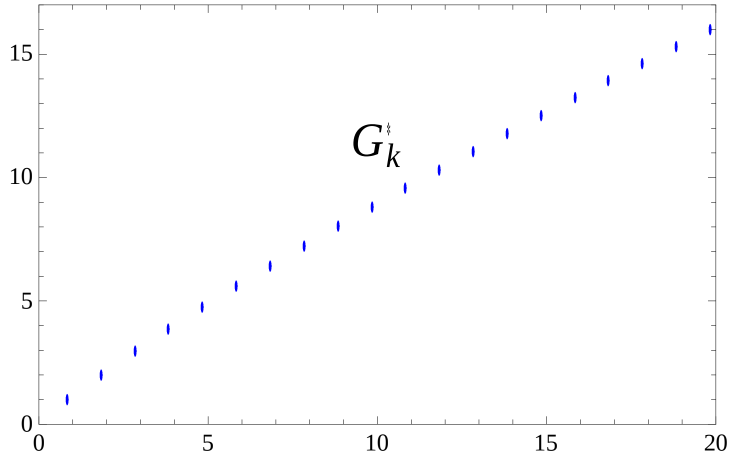




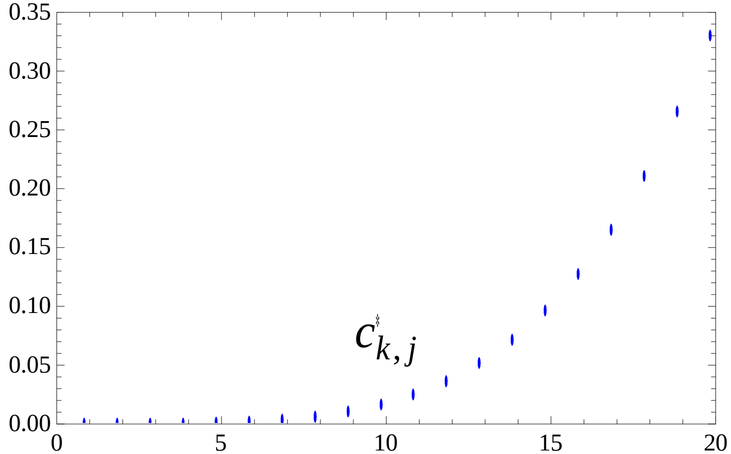
Alpha = 1.2



C_2
0.0
0.2
0.4
0.6
0.8
1.0

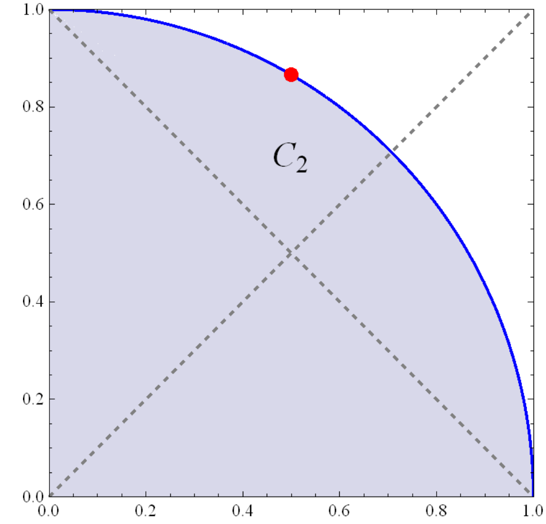


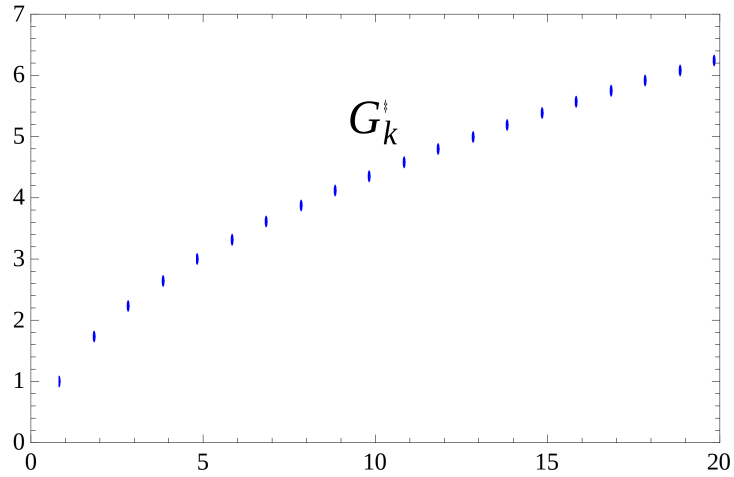
$G_k^{\ddagger}$
0
5
10
15
20

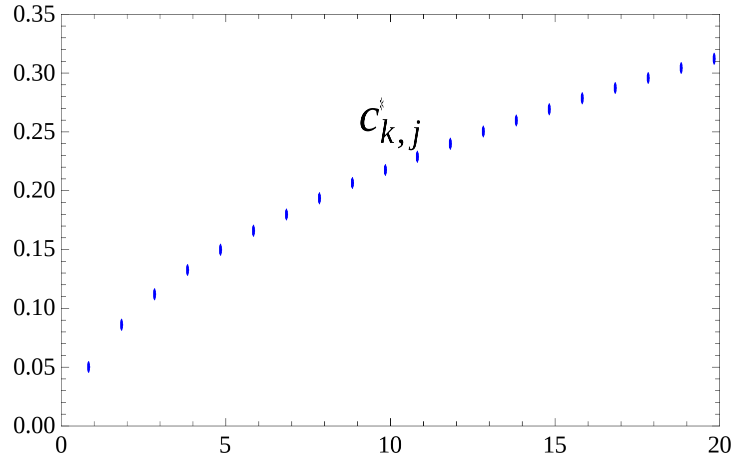


$c_{k,j}^{\ddagger}$
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
5
10
15
20

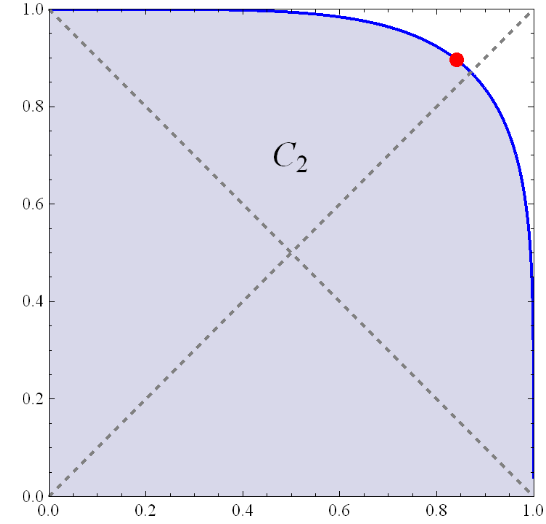
Alpha = 2.0

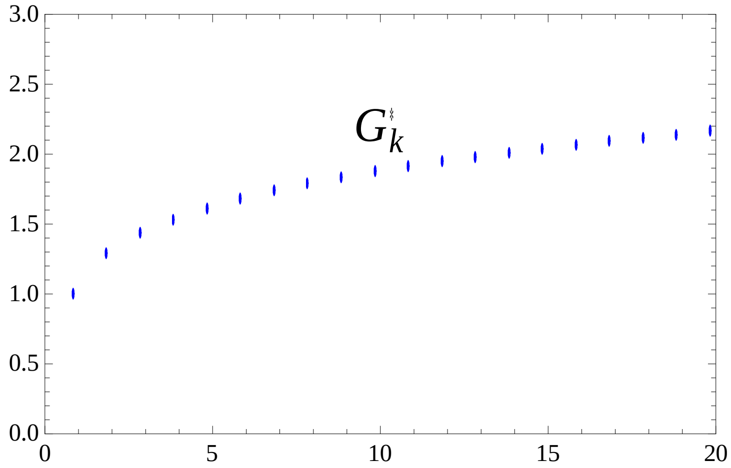

C_2
0.0
0.2
0.4
0.6
0.8
1.0

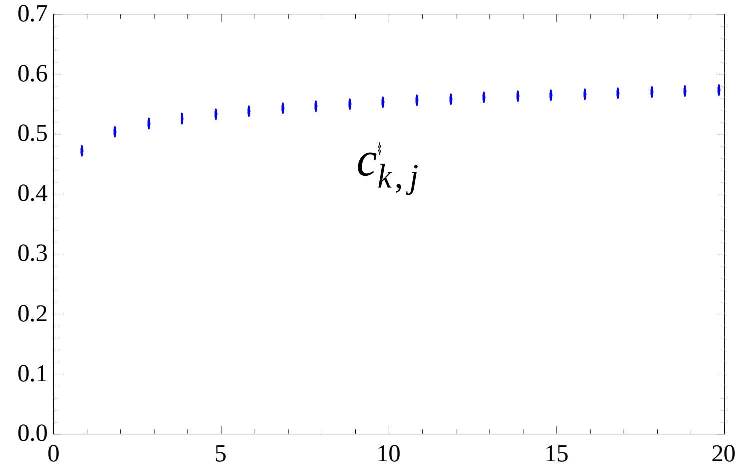

$G_k^{\ddagger}$
0
1
2
3
4
5
6
7
0
5
10
15
20


$c_{k,j}^{\ddagger}$
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
5
10
15
20

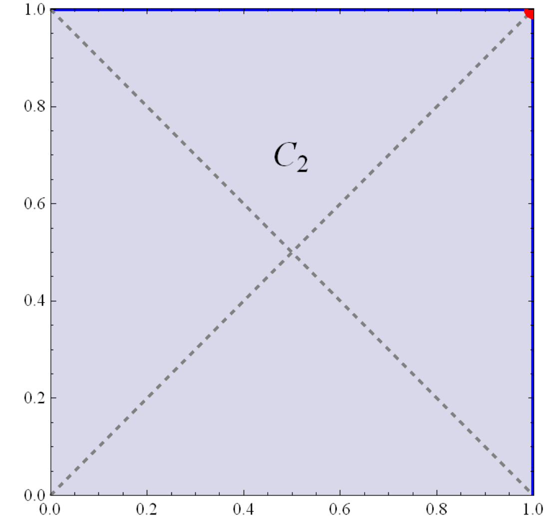
Alpha = 5.0

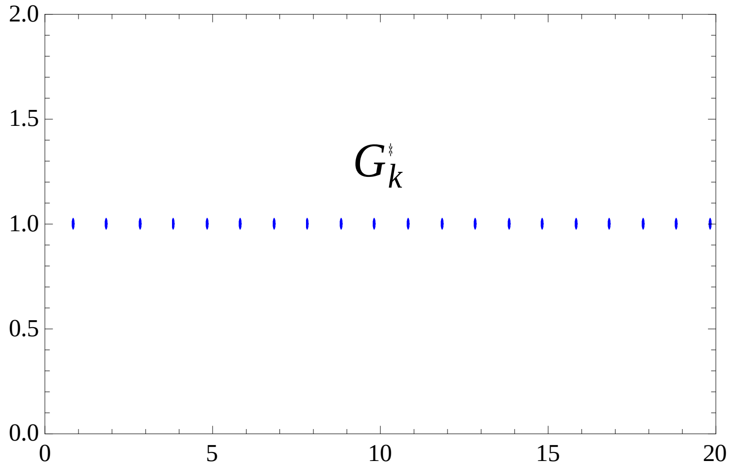

C2
0.0
0.2
0.4
0.6
0.8
1.0

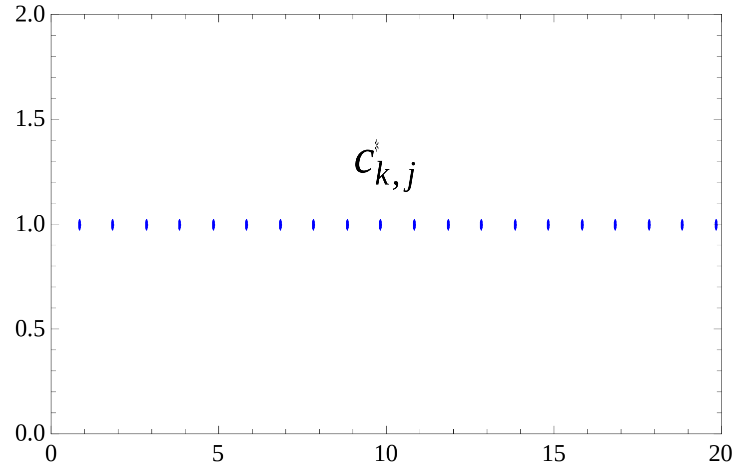

Gk
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
5
10
15
20


ck, j
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
5
10
15
20

Alpha = infinite (infinite-server queue)



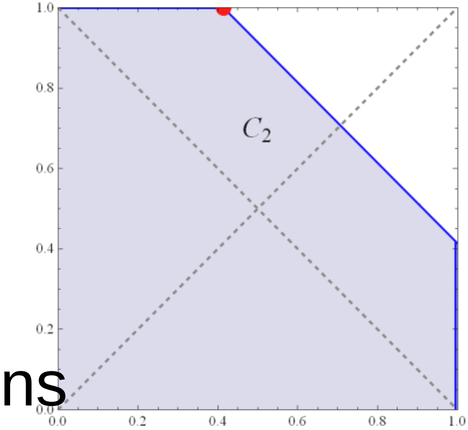




Outline

- Introduction
- Optimal scheduling problem
- Solution
- Examples
- Lower and upper bounds
- Summary

Symmetric polymatroids



- Let $\gamma_1 < \ldots < \gamma_n$ and consider capacity regions

$$C_k = \{\mathbf{c}_k \ge 0 : \sum_{i \in I} c_{ki} \le \gamma_{|I|}, I \subset \{1,\ldots,n\}\}$$

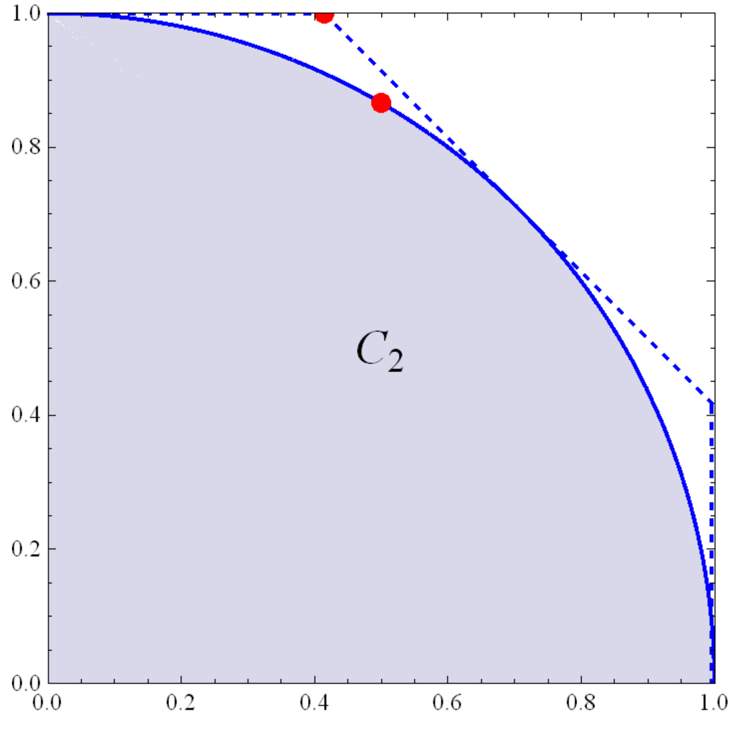
- Theorem 2: If $\gamma_1 > \gamma_2 - \gamma_1 > \ldots > \gamma_n - \gamma_{n-1}$, then

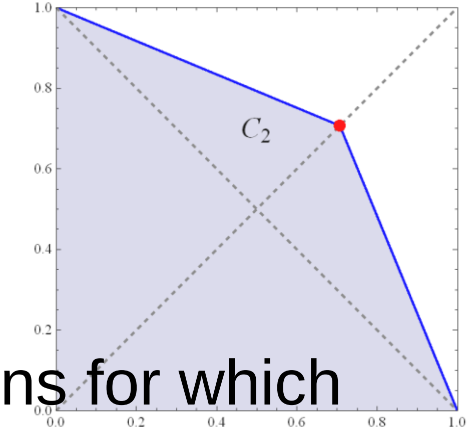
$$G_1^* < \ldots < G_n^* \qquad \text{(increasing in } k\text{)}$$

$$c_{kj}^* = \gamma_{k-j+1} - \gamma_{k-j} \qquad \text{(increasing in } j\text{)}$$

- Optimality result of Sadiq and de Veciana (2010)

Optimistic (lower) bound





Symmetric OPS-limited polytopes

- Let $\gamma_1 < \ldots < \gamma_n$ and consider capacity regions for which C_k is the convex hull of all permutations of rate vectors

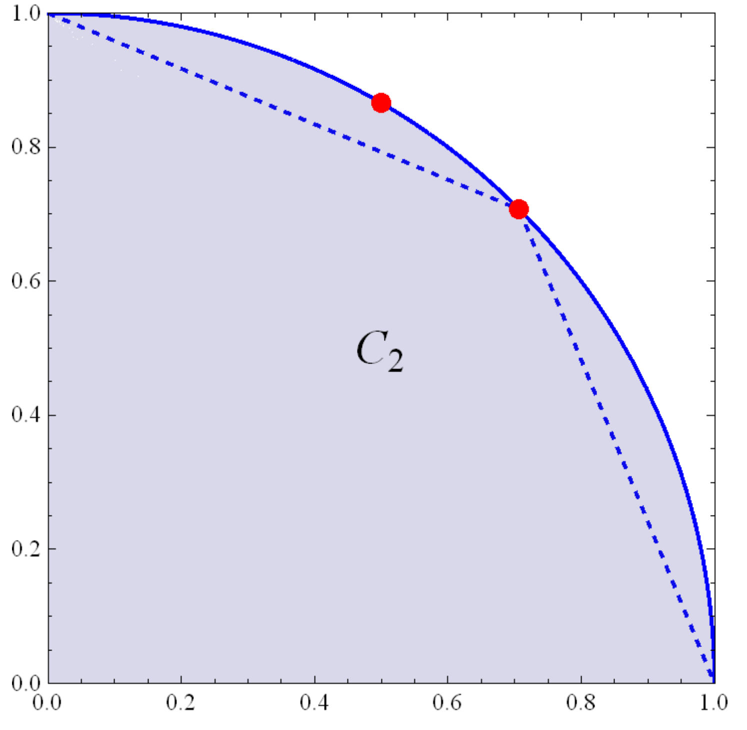
$$(0,\ldots 0, \frac{\gamma_j}{j}, \ldots, \frac{\gamma_j}{j}), \quad j = 0,\ldots,k$$

- Theorem 3: If $G_1{}^* < \ldots < G_n{}^*$, then

$$\mathbf{c}_k^* = (0,\ldots 0, \frac{\gamma_{j_k^*}}{j_k^*}, \ldots, \frac{\gamma_{j_k^*}}{j_k^*})$$

- SRPT-OPS policy introduced in Sadiq and de Veciana (2010)

Conservative (upper) bound



Outline

- Introduction
- Optimal scheduling problem
- Solution
- Examples
- Lower and upper bounds
- Summary

Summary

- Assumptions:
  - Abstract capacity regions (time-scale separation)
  - Transient system
- Results:
  - Optimality result for compact and symmetric capacity regions
  - Optimal rate vectors (that do not depend on absolute sizes of the flows) for each phase
  - Conservative upper bound for the mean delay
- Open questions:
  - Is it possible to make the implicit condition explicit?
  - Any idea about the truly dynamic system with random arrivals?

The End

New results: Making the implicit condition explicit

- The implicit condition is, indeed, satisfied under the assumption that the channel conditions for different users are independent and identically distributed.
- In addition, there is a recursive algorithm for the optimal flow-level rate vectors that directly utilizes the time slot level channel model.
- It is also possible to determine explicitly how to implement the optimal rate vectors in the time slot level opportunistic scheduler.
- But this is another story …