heSRPT: Parallel Scheduling to Minimize Mean Slowdown

Ben Berg, Rein Vesilo, Mor Harchol-Balter

Carnegie Mellon University Computer Science Department

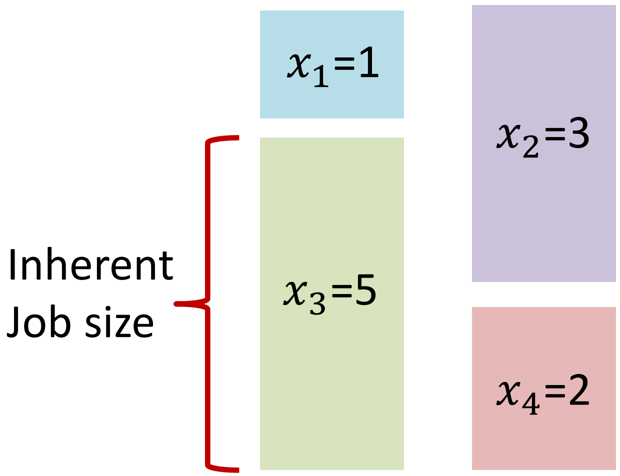
Carnegie Mellon University

Parallel Jobs Are Ubiquitous

- Multicore chips
- Supercomputing centers
- Servers in a datacenter

Workloads are <u>increasingly parallel</u>

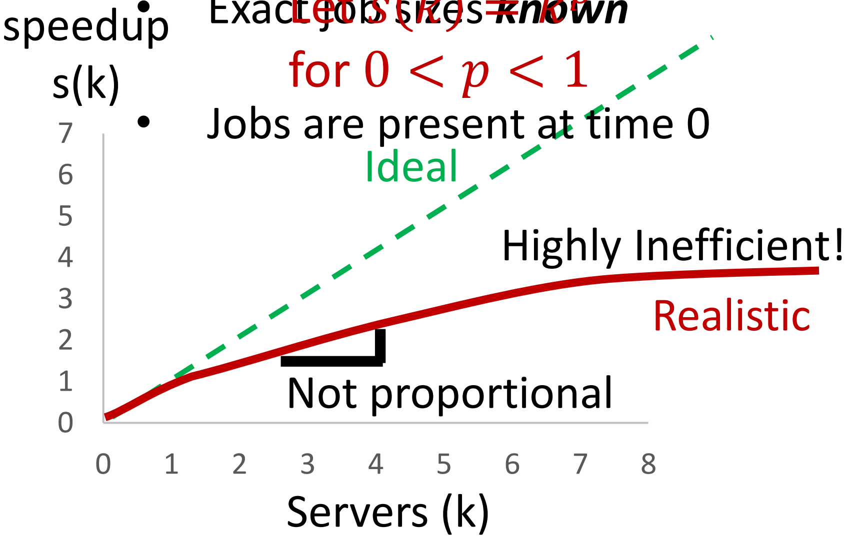
Our Model of Server Allocation

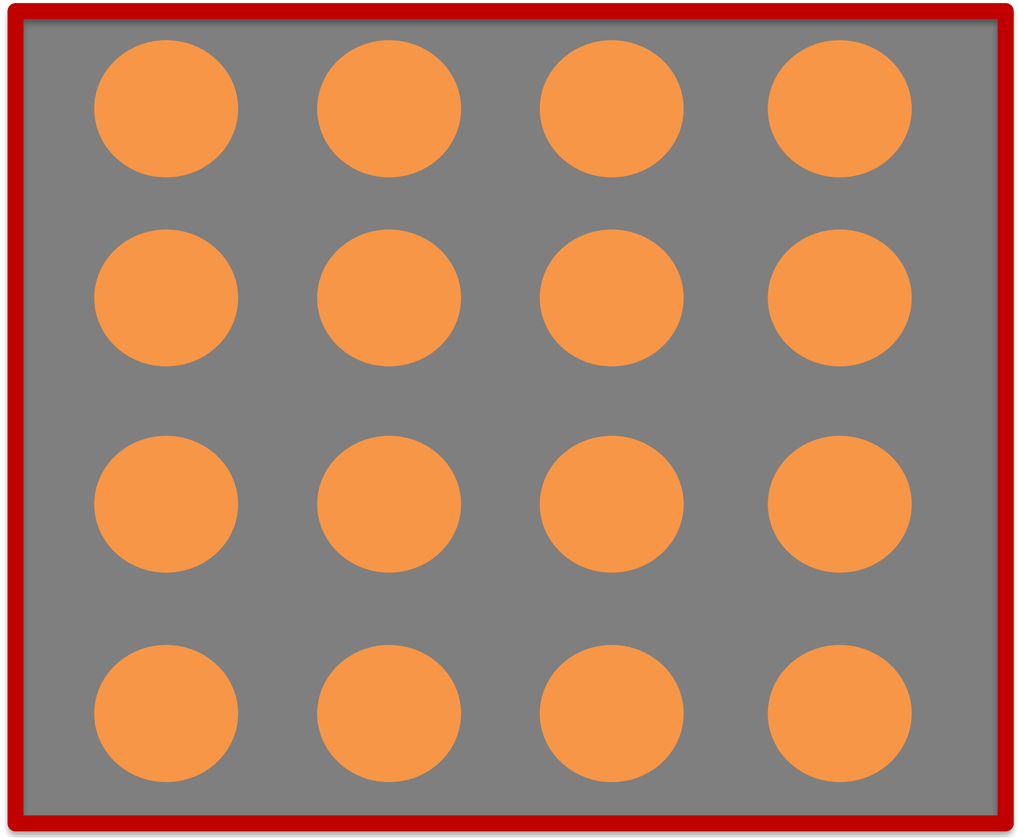


M = 4 Jobs

- Exact job sizes ***known***
- Jobs are present at time 0

Let $s(k) = k^p$ for $0 < p < 1$



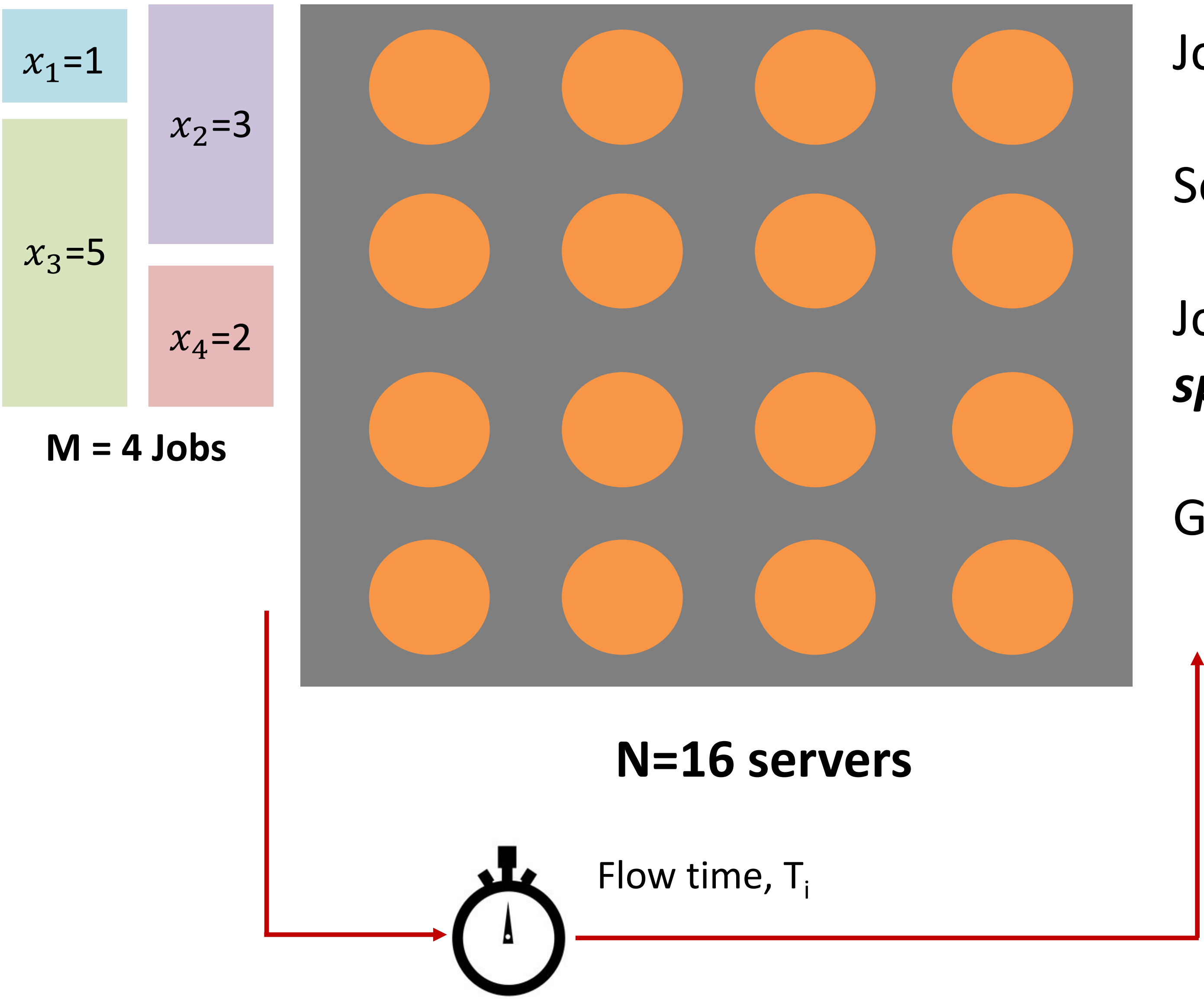


N=16 servers

Big Question:

How to allocate servers to jobs?

An Allocation Policy



Jobs are ***malleable***

Servers are ***divisible***

Jobs follow a ***single speedup function***

Goal?

Carnegie Mellon University

Metrics of Interest

Goal: Minimize Flow Time

Minimize $\sum_{i=1}^{M} T_i$

First: Mean Flow Time

- Classic, intuitive metric
- Good "overall" system performance
- Well understood by both theory and systems communities

Goal: Minimize Slowdown

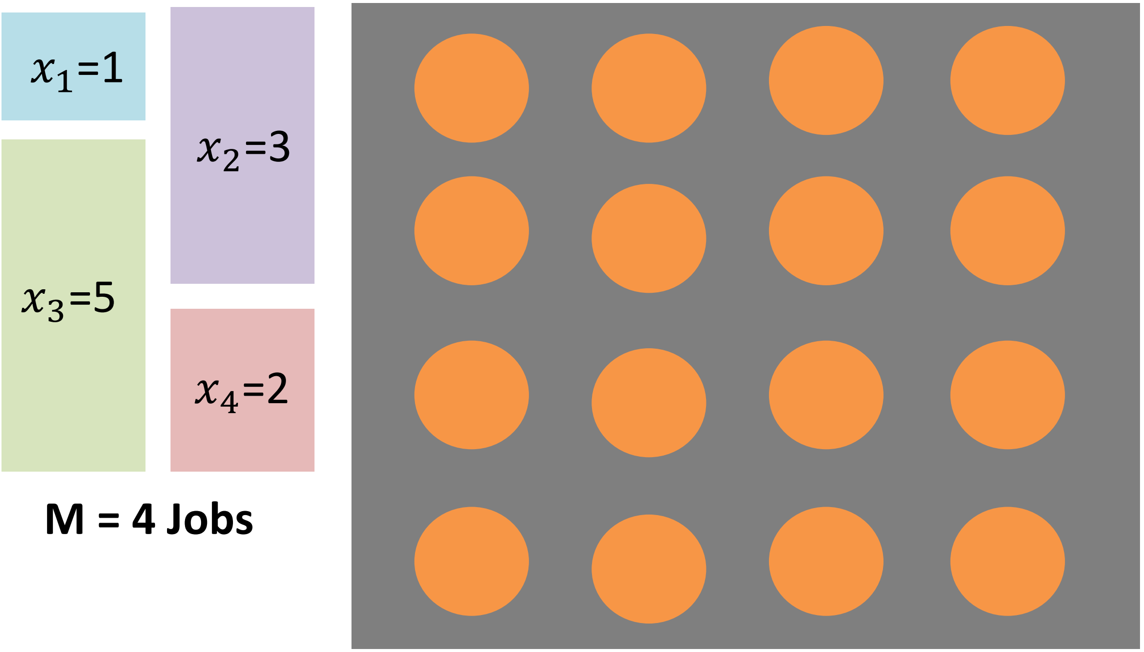
Minimize $\sum_{i=1}^{M} \frac{T_i}{x_i/s(N)}$

Second: Mean Slowdown

- Long jobs tolerate long waits
- Popular in multiuser systems
    - Seems "Fair"
- ***Expansion Factor*** in HPC

Carnegie Mellon University

Minimizing Mean Flow Time



- Optimal allocation policy?

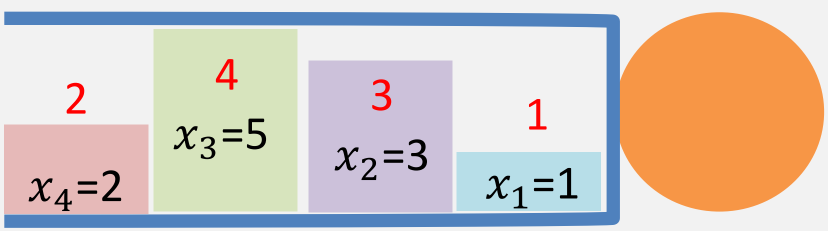
EQUI?

More servers to big jobs?

More servers to small jobs?

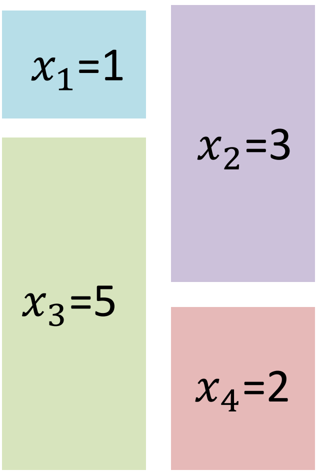
Detour: Shortest Remaining Processing Time

SRPT is optimal!

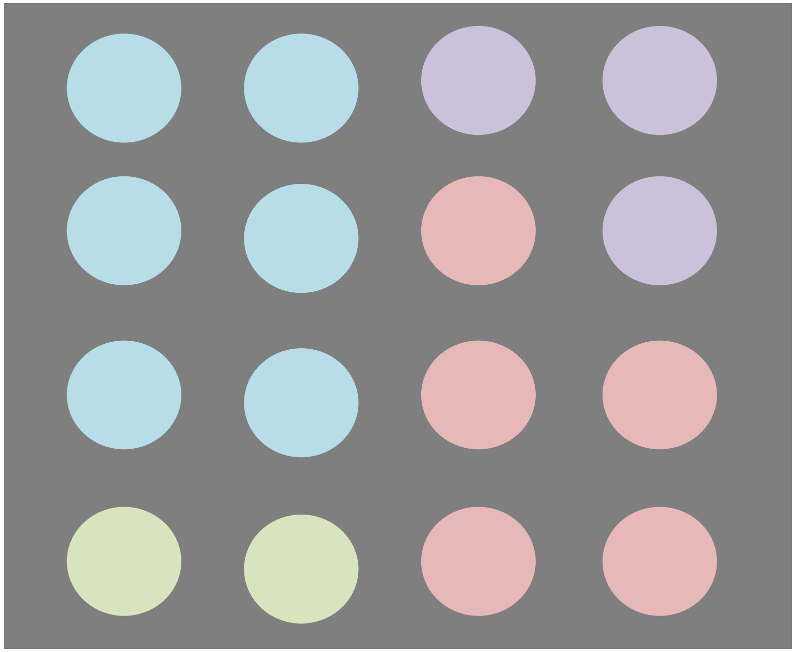


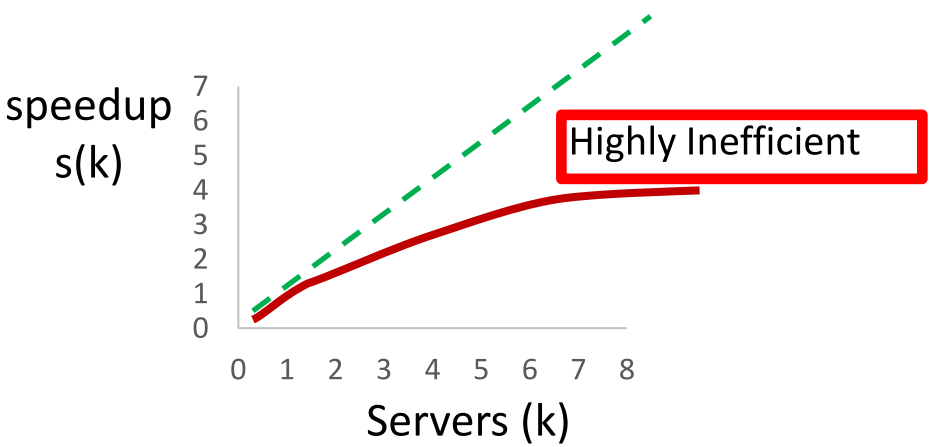
Carnegie Mellon University

Minimizing Mean Flow Time



M = 4 Jobs





- Optimal allocation policy?

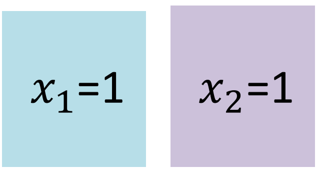
EQUI?

More servers to big jobs?

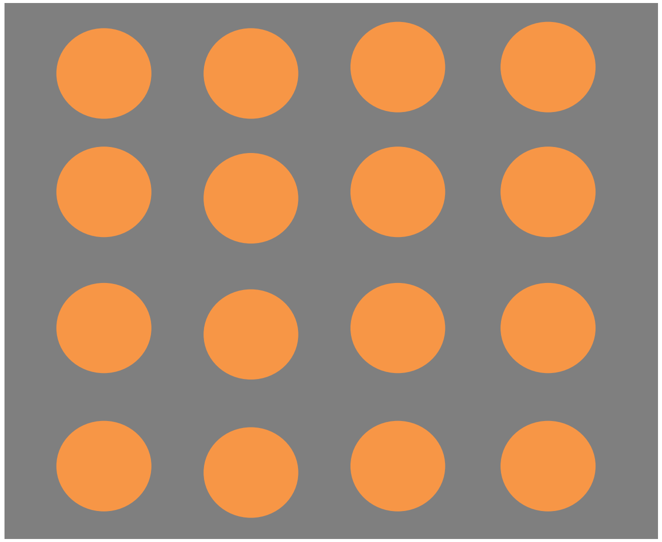
More servers to small jobs?

SRPT: Give all to shortest job?

Consider an easier problem…



M=2 jobs

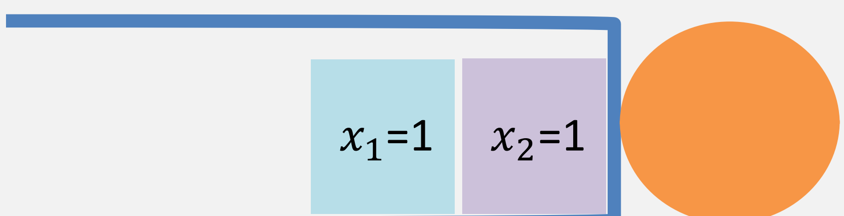


- Let $s(k) = \sqrt{k}$, N=100
- Optimal allocation policy?

EQUI?

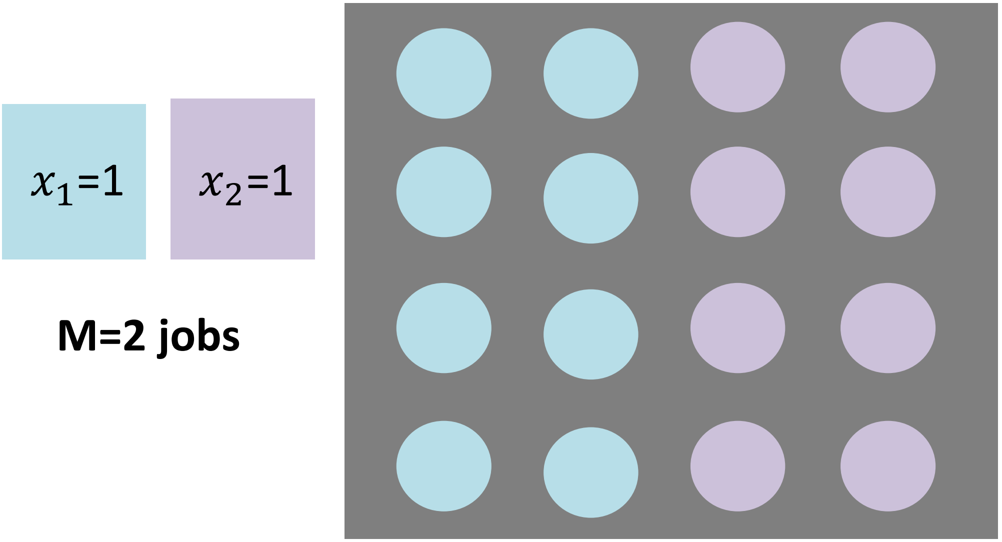
Detour: Processor Sharing

$T_{PS} = 4$ $T_{FCFS} = 3$



Carnegie Mellon University

Consider an easier problem...

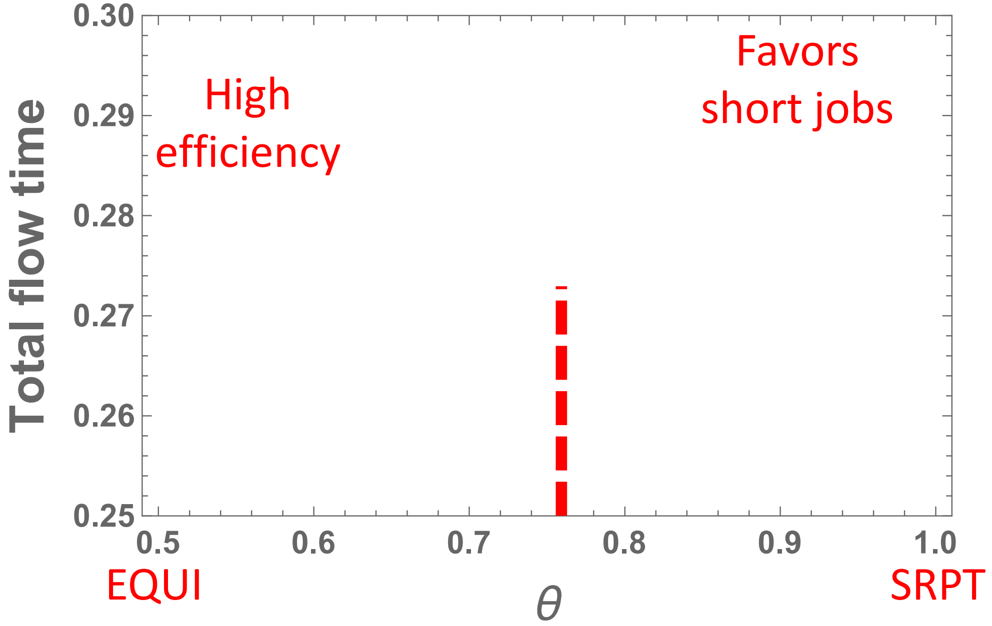


M=2 jobs

- Let $s(k) = \sqrt{k}$, N=100
- Optimal allocation policy?

EQUI?

- Let θ be fraction given to **blue job**



Optimal allocation: $\theta^* = .75$

Best of both worlds:
high efficiency SRPT (heSRPT)

Carnegie Mellon University

The heSRPT Optimization Problem

Recall: $s(k) = k^p$

O(M²) Variables!

$$\theta^* = 1 - \left(\frac{1}{2}\right)^{\frac{1}{1-p}}$$

$T[x_1, x_2]$

$$T[x_1, x_2, x_3] = \frac{3x_3}{s(\theta_3^3 N)} + \frac{2\left(x_2 - \frac{s(\theta_2^3 N)x_3}{s(\theta_3^3 N)}\right)}{s(\theta_2^2 N)} + \frac{x_1 - \frac{s(\theta_1^3 N)x_3}{s(\theta_3^3 N)} - \frac{s(\theta_1^2 N)\left(x_2 - \frac{s(\theta_2^3 N)x_3}{s(\theta_3^3 N)}\right)}{s(\theta_2^2 N)}}{s(N)}$$

Carnegie Mellon University

The heSRPT Optimization Problem

What if job 2 finishes first?

One optimization problem ***per completion order***

Recall: $s(k) = k^p$

O(M²) Variables!

O(M!) Optimization Problems

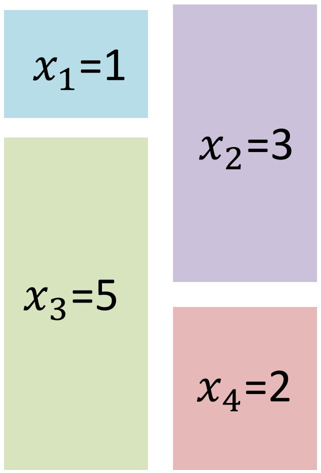
$$T[x_1, x_2, x_3] = \frac{3x_3}{s(\theta_3^3 N)} + \frac{2\left(x_2 - \frac{s(\theta_2^3 N)x_3}{s(\theta_3^3 N)}\right)}{s(\theta_2^2 N)} + \frac{x_1 - \frac{s(\theta_1^3 N)x_3}{s(\theta_3^3 N)} - \frac{s(\theta_1^2 N)\left(x_2 - \frac{s(\theta_2^3 N)x_3}{s(\theta_3^3 N)}\right)}{s(\theta_2^2 N)}}{s(N)}$$

How do you enforce a completion order?

O(M) constraints ***per optimization problem***

Carnegie Mellon University

Reducing the Search Space



M = 4 Jobs

Thm: optimal completion order

Optimal completion order to minimize mean flow time is Shortest Job First

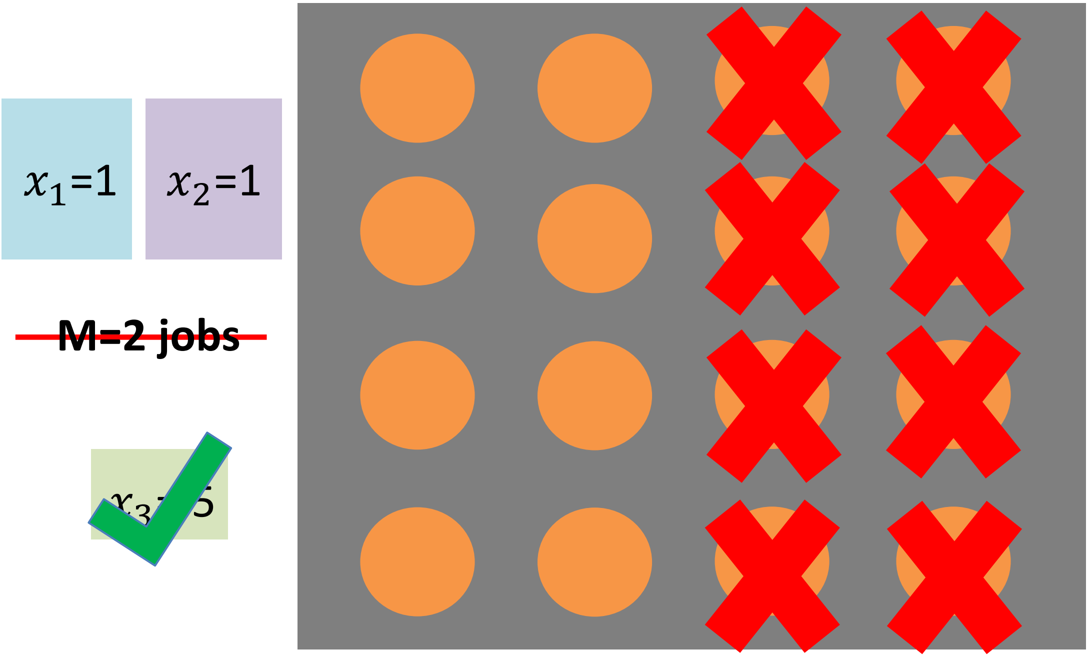
$$T[x_1, x_2, x_3] \rightarrow T[x_1, x_2]$$

Thm: optimal substructure

Scale-free property

Carnegie Mellon University

Theorem: The Scale-Free Property



θ^* does not depend on N

Smaller system?
Same fraction θ^*!

Changing system?
Unclear…

Turns out θ^* remains constant

$$\theta^* = 1 - \left(\frac{1}{2}\right)^{\frac{1}{1-p}}$$

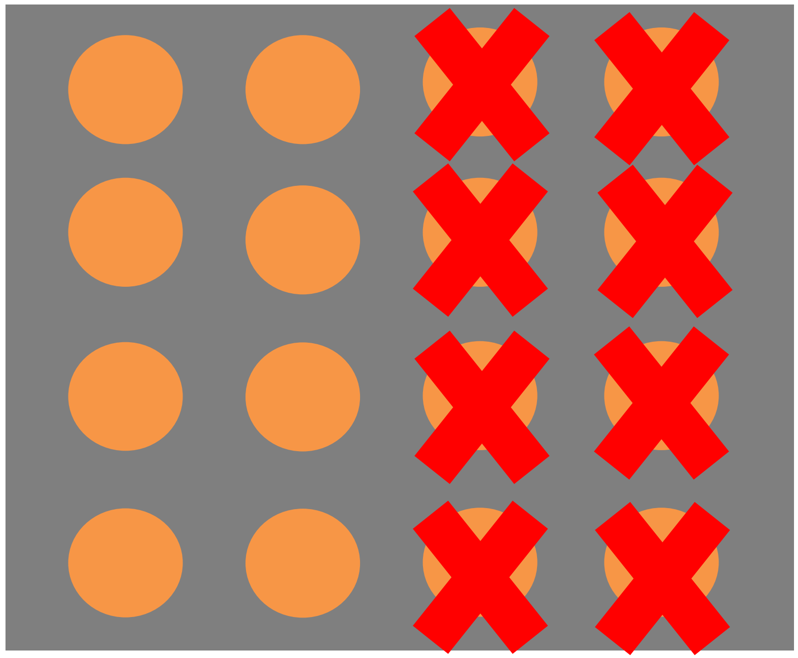
Why do we care?

Claim: $\frac{\text{allocation to job i}}{\text{allocation to jobs larger than i}} = \omega_i \rightarrow$ ***scale-free constant***

Using The Scale-Free Property

x_1=1 x_2=1

M=2 jobs



O(M²) variables -> exactly M scale-free constants

Much easier to write total flow time

Explicitly solve for optimal scale-free constants

$$T = \frac{1}{s(N)} \sum_{i=1}^{M} x_i \cdot [is(1+\omega_i) - (i-1)s(\omega_i)]$$

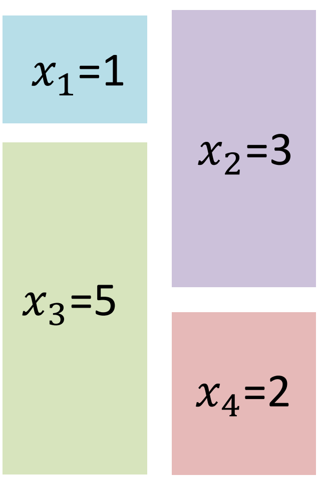
M scale-free constants precisely define heSRPT!

The Optimal Allocation

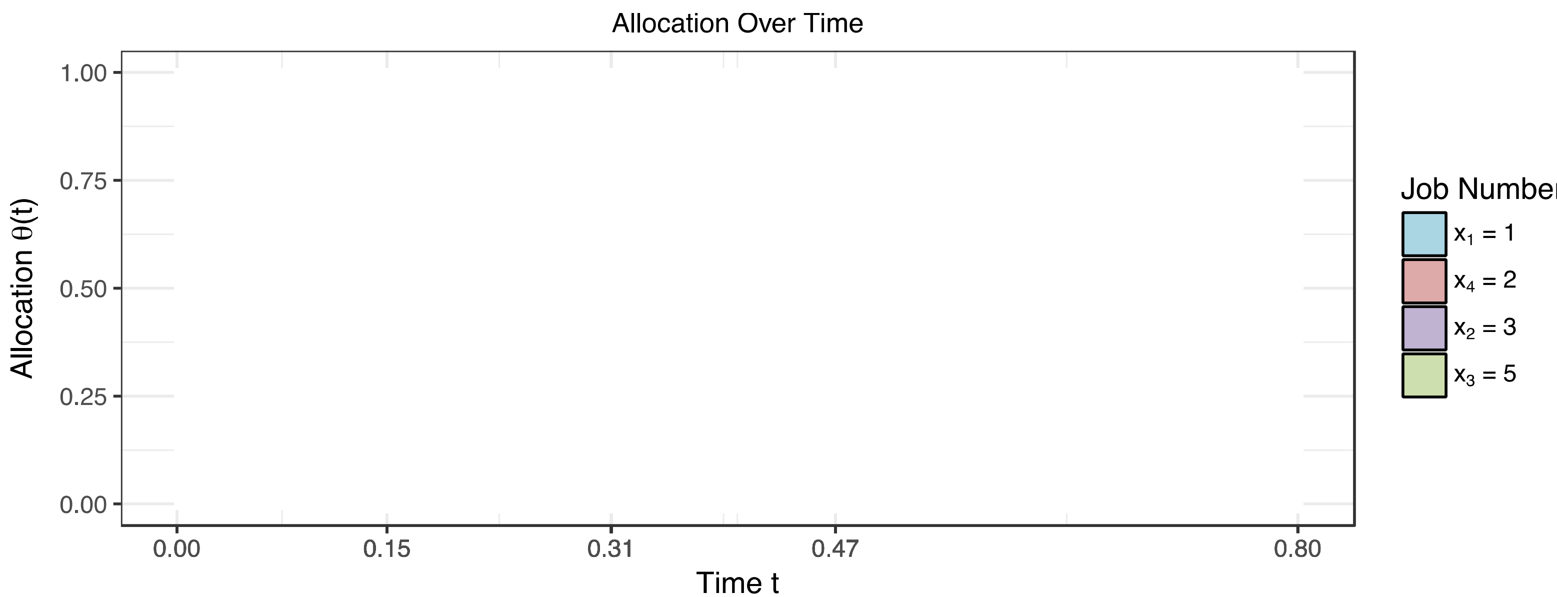
$$\theta_i^*(t)$$

for the ith largest job   At time t (when $m(t)$ jobs are in the system)

$$\theta_i^*(t) = \left(\frac{i}{m(t)}\right)^{\frac{1}{1-p}} - \left(\frac{i-1}{m(t)}\right)^{\frac{1}{1-p}} \quad \forall 1 \le i \le m(t)$$



M = 4 Jobs



Summary of Results

Goal: Minimize Mean Flow Time

Thm: optimal completion order

Shortest Job First

Thm: optimal substructure

Scale-free property

Thm: optimal allocation:

$$\theta_i^*(t) = \left(\frac{i}{m(t)}\right)^{\frac{1}{1-p}} - \left(\frac{i-1}{m(t)}\right)^{\frac{1}{1-p}} \quad \forall 1 \le i \le m(t)$$

Goal: Minimize Mean Slowdown

→ Optimal completion order to minimize mean slowdown is ***also*** Shortest Job First

→ Holds for ***any weighted flow time*** metric (including Slowdown)

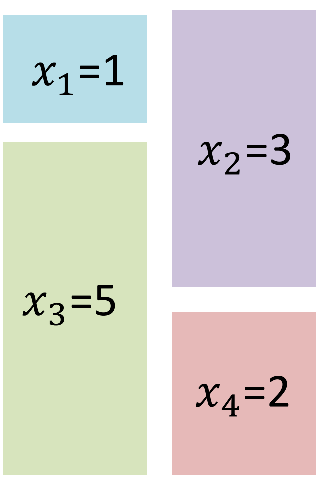
Optimal Allocation for Slowdown

Minimizes mean flow time

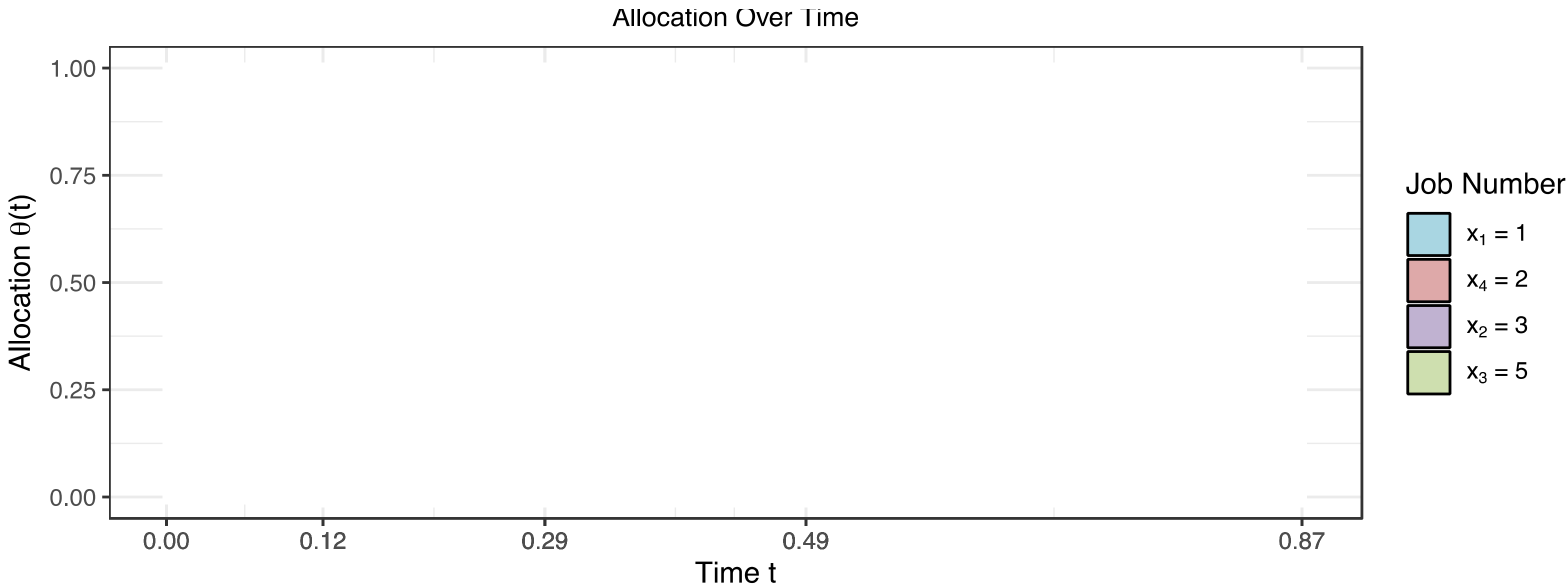
$$\theta_i^*(t) = \left(\frac{i}{m(t)}\right)^{\frac{1}{1-p}} - \left(\frac{i-1}{m(t)}\right)^{\frac{1}{1-p}} \quad \forall 1 \leq i \leq m(t)$$

$$z(i) = \sum_{j=1}^{i} \frac{1}{x_i S(N)} \qquad \theta_i^*(t) = \left(\frac{z(i)}{z(m(t))}\right)^{\frac{1}{1-p}} - \left(\frac{z(i-1)}{z(m(t))}\right)^{\frac{1}{1-p}} \quad \forall 1 \leq 1 \leq m(t)$$

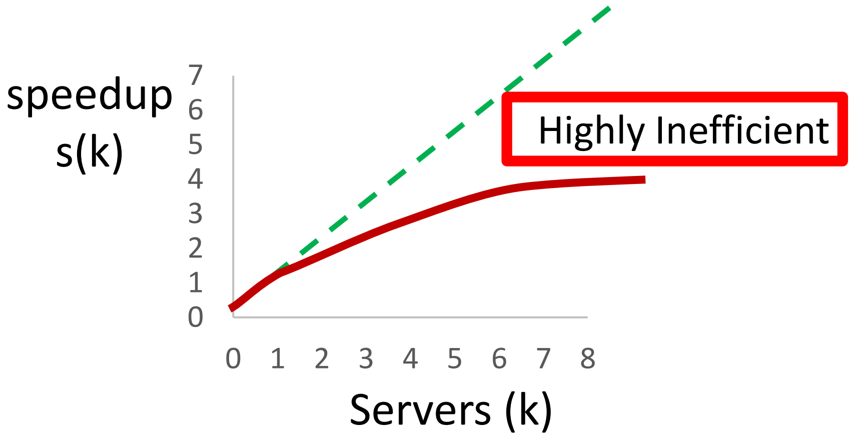
Rate at which slowdown is accrued

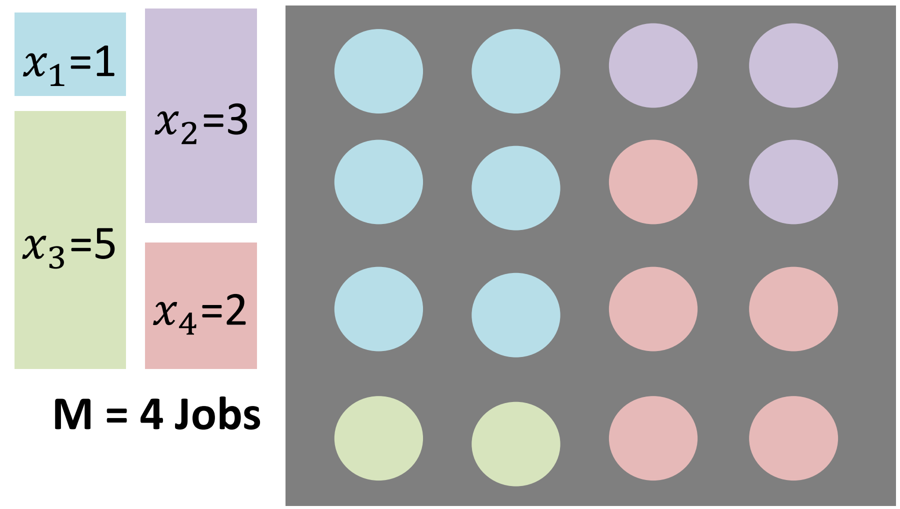


M = 4 Jobs



Conclusion





Goal:

Minimize mean flow time, slowdown

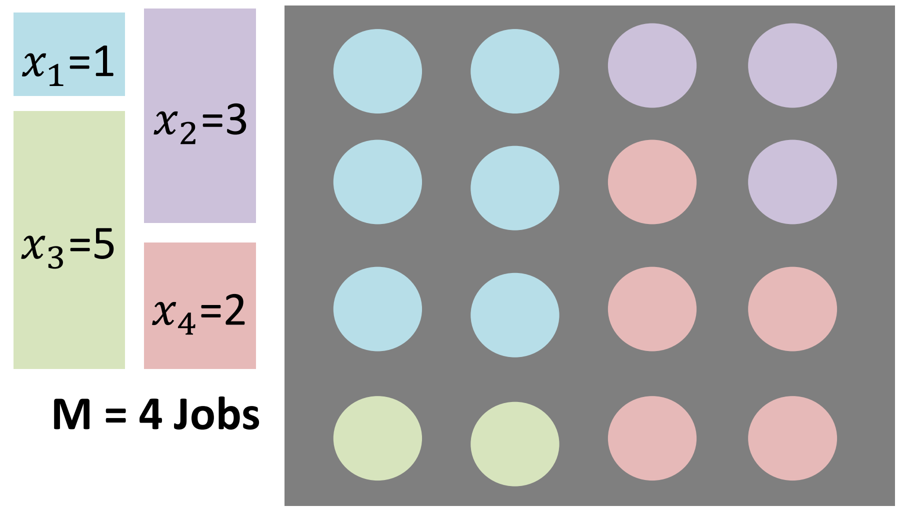
We have a closed-form solution for both cases

Reducing the search space:

- Derive Optimal Completion Order
- Find Optimal Substructure

Carnegie Mellon University

Questions?



Goal:

Minimize mean flow time, slowdown

Idea: balance EQUI and SRPT (**heSRPT**)

We have a closed-form solution for both cases

Reducing the search space:

- Derive Optimal Completion Order
- Find Optimal Substructure

Carnegie Mellon University