Online Scheduling with QoS Constraints

Uwe Schwiegelshohn

Scheduling for large-scale systems
Knoxville, TN, USA
May 14, 2009

Problem Description

- Online scheduling $r_{j,online}$
  - Jobs are submitted over time.
  - At its submission r_j, job J_j is immediately allocated to an eligible machine.

- Parallel identical machines P_m
  - Each job J_j has the same processing time p_j on each eligible machine.

- Ordered machine eligibility
  - There is a fixed order of the machines: 1,2,…,m
  - Using this order, the first machine eligible to execute job J_j is machine k_j.
  - Every machine i with $i \geq k_j$ is also eligible to execute job J_j: $M_j=\{i|i \geq k_j\}$

- Makespan C_{max}
  - It is the goal to minimize the makespan of the schedule.

$$P_m|r_{j,online},M_j|C_{max}$$

Previous Results

- $P_m|M_j|C_{max}$ with no restrictions on M_j.
  - The problem is NP-hard as $P_m||C_{max}$ is already NP-hard.
- $P_m|p_j=1,M_j|C_{max}$ with nested machine eligibility constraints.
  - $M_j = M_k$, $M_j \subset M_k$, $M_j \supset M_k$, or $M_j \cap M_k = ø$.
  - The Least Flexible Job First (LFJ) rule optimally solves this problem.
  - M. Pinedo: Scheduling: Theory, Algorithms, and Systems, Prentice Hall, 2002.
- $P_m|M_j|C_{max}$ with ordered eligibility.
  - Least eligibility – longest processing time order guarantees the approximation factor 2-1/(m-1).
  - H-C. Hwang, S.Y. Chang, K. Lee. Parallel machine scheduling under a grade of service provision, Computer & Operations Research 31, 2055-2061 (2004).
- $P_m|r_{j,online},M_j|C_{max}$ with no restrictions on M_j.
  - Competitive ratio log n for deterministic and randomized cases.
  - Y. Azar, J. Naor, R. Ron. The Competitiveness of On-Line Assignments, Journal of Algorithms 18, 221-237 (1995).

Relevance of the Problem

- Shall I allocate my precious resources to somebody not paying enough for them or run the risk that these resources are not used at all?
- In practice, this is a fixed capacity problem with customer rejection.
  - This problem is different from the utilization of a fixed number of machines.
  - There is a close connection with utilization if there ins no rejection.
  - The makespan can represent this objective.

Application Examples

- Packaging in lattice boxes
  - The granularity of the material determines the required size of the lattice.

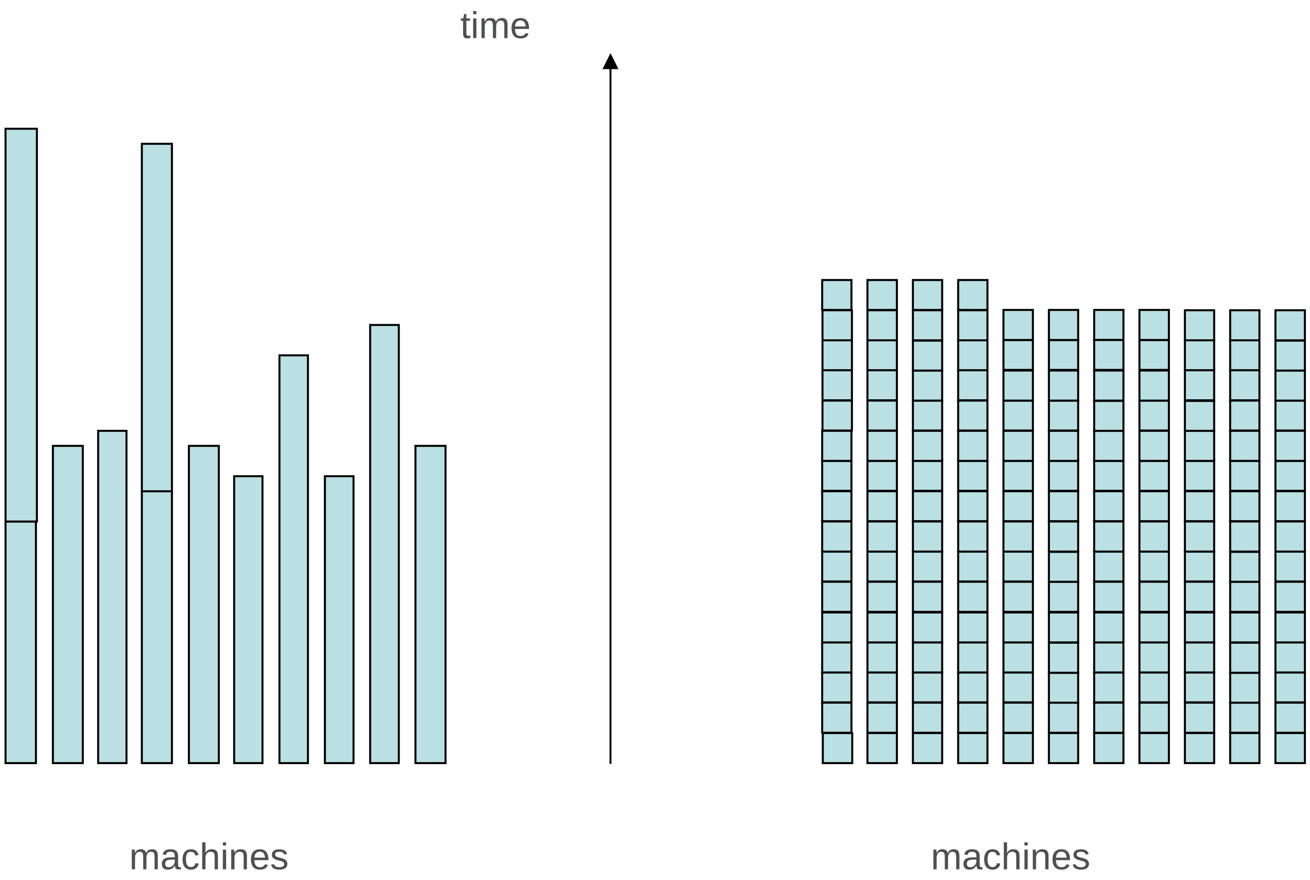
- Servers with different amount of main memory
  - The storage requirement of a job determines the eligibility of a server.

- Class of transportation
  - The ticket determines the class of transportation.

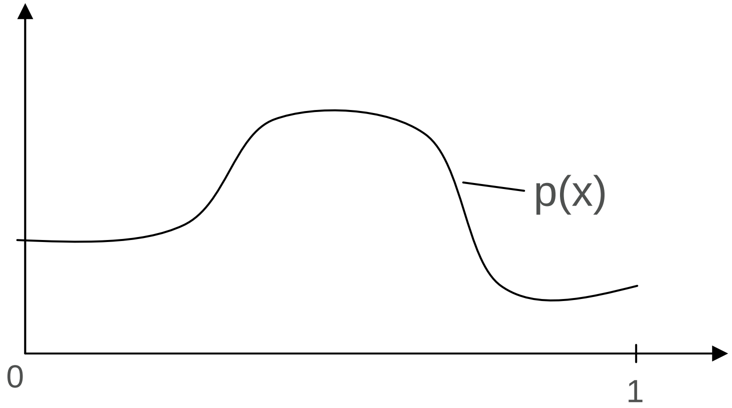
Continuous Model for Large-scale Systems

- We allow fractional machines and normalize the machine space.
  - The machine space is represented by the interval [0,1] of real numbers.
  - Job J_j can only be allocated to the interval $[k_j,1]$ with $0 \le k_j \le 1$.
- Each job has a very short processing time.
  - Job allocation does not need to consider individual processing times.
- Machine eligibility is represented by the job density function p(x)
  - $p(x): [0,1] \rightarrow \mathbb{R}^{\ge 0}$: Total processing time of jobs with $k_j = x$.
  - Release dates are not considered within the job density function.
- There is a completion time function that determines the makespan.
  - $c_S(x): [0,1] \rightarrow \mathbb{R}^{\ge 0}$: Completion time function of schedule S
  - $C_{max}(S) = \max\{c_S(x) | 0 \le x \le 1\}$: Makespan of schedule S
  - Idle times are included.

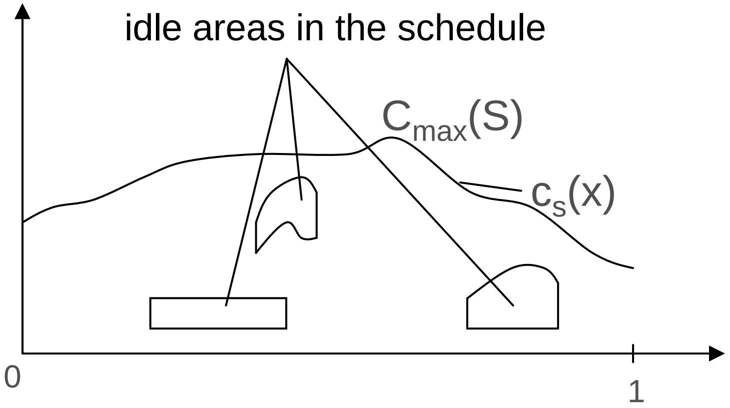
Long and Short Processing times



Job Density and Completion Time Functions



Job density function

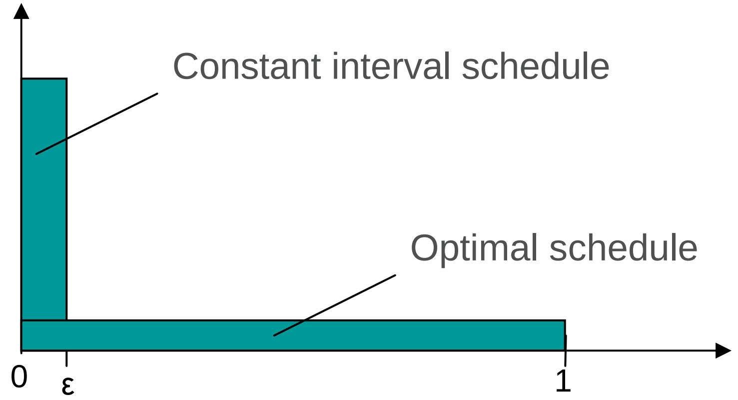


Completion time function

$$\int_0^1 p(x)dx = \int_0^1 c_S(x)dx$$ if there are no intermediate idle areas in the schedule.

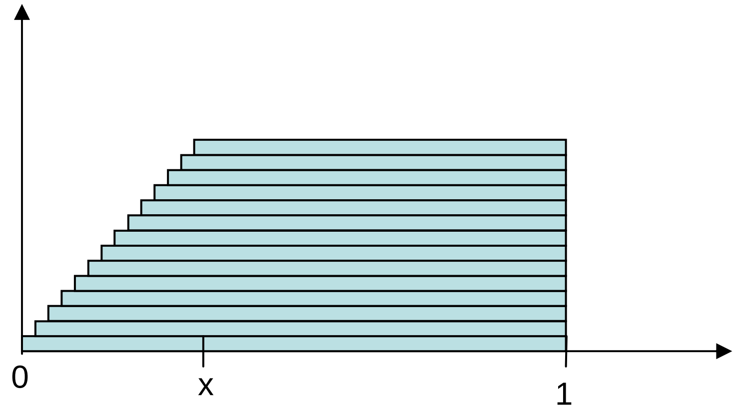
Simple Approaches with Bad Results

- Constant interval approach: A new job J_j is allocated such that the maximum of the function $c_s(x)$ is increased the least in the interval $[k_j, \max\{k_j+\varepsilon,1\})$.
  - The competitive factor is ε^{-1}.



- Greedy approach: A new job J_j is allocated such that the maximum of the function $c_s(x)$ is increased the least in the interval $[k_j,1)$.
  - $p(x)=1$, jobs are submitted in quick succession in order of k_j.
  - The competitive factor is not constant.

Greedy Approach



$$x = \int_0^x c_S(t)dt + (1-x)\cdot c_S(x)$$

$$1 = c_S(x) - c_S(x) + (1-x)\cdot \frac{dc_S(x)}{dx}$$

$$\frac{dc_S(x)}{dx} = \frac{1}{1-x} \Rightarrow c_S(x) = C\cdot \ln\frac{1}{1-x}$$

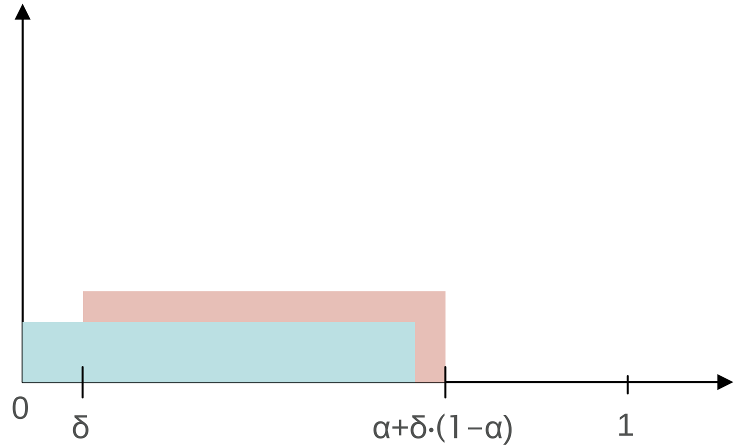
$$\int c_S(x)dx = C\cdot\big((1-x)\cdot\ln(1-x)+x\big)$$

$$\frac{1}{2} = \frac{1}{2}C\cdot(1-\ln 2) + \frac{1}{2}C\cdot\ln 2 \Rightarrow C = 1$$

$$\lim_{x\to 1} c_S(x) \to \infty$$

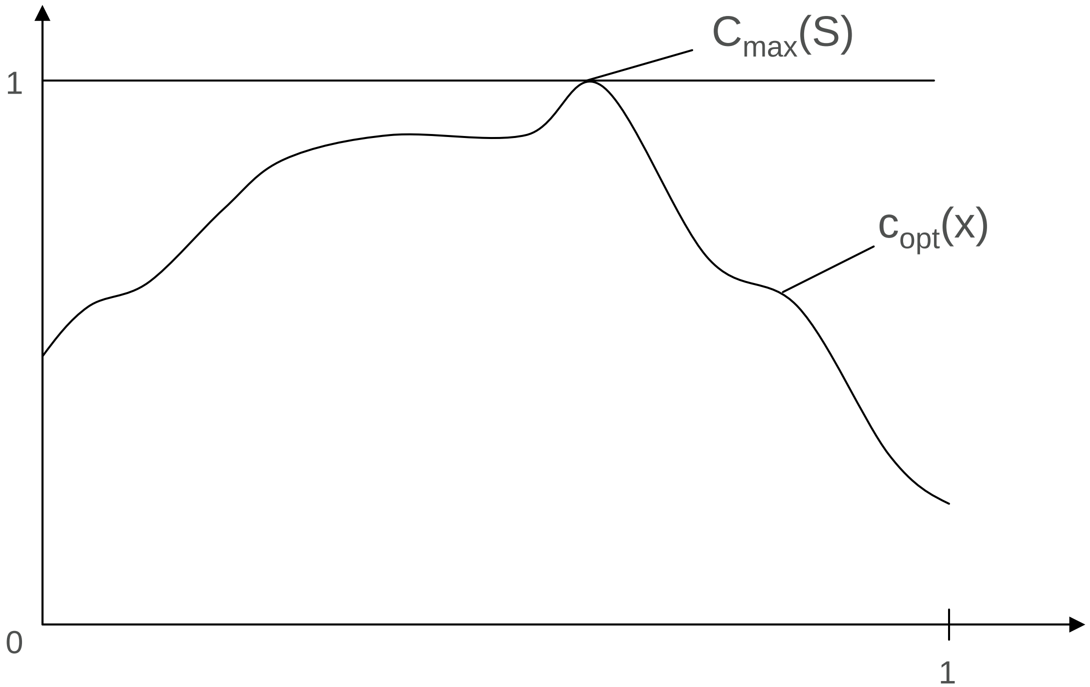
Interval Approach

- A job J_j is only executed in the interval $[k_j, k_j+(1-k_j)/\alpha)$ with $\alpha>1$.
- In this approach, additional jobs cannot decrease the makespan of a schedule.
- Example: Assume that a group of jobs with $k_1=0$ is released at time 0 and immediately followed by another group of jobs with $k_2=\delta$



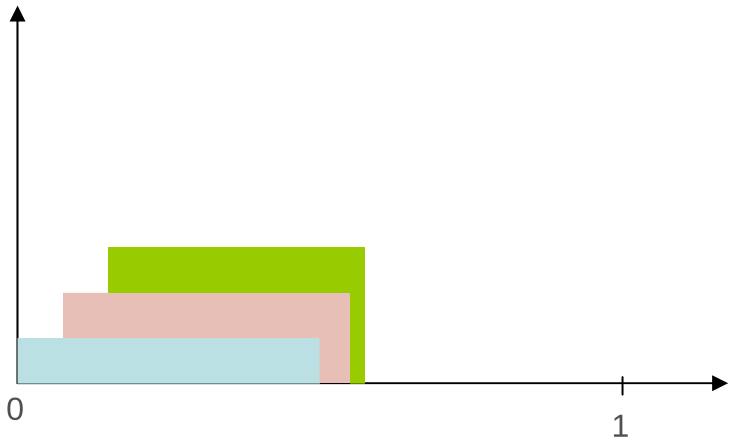
Making a Schedule Worse

- Jobs are added until $c_{opt}(x)$=const for all x and the schedule contains no idle areas.
  - The ratio $\max_x\{c_S(x)\}$ to $\max_x\{c_{opt}(x)\}$ cannot decrease.
- We normalize the job density function such that $c_{opt}(x)=1$.

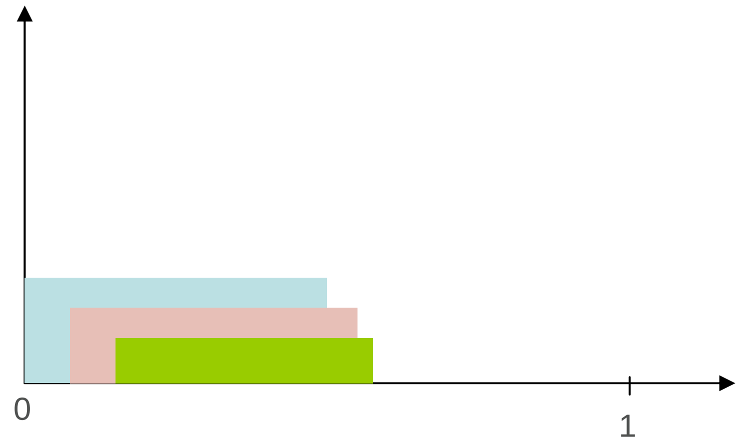


Worst Case: Job Submission Order

- The jobs are submitted in quick succession in increasing order of k_j.
  - The difference between the starting values of two intervals k_2-k_1 is larger than the difference between the ending values of these intervals $(1-1/\alpha)\cdot(k_2-k_1)$.
  - An increasing order of k_j produces larger C_{max} values than a decreasing order.



Increasing order of k_j



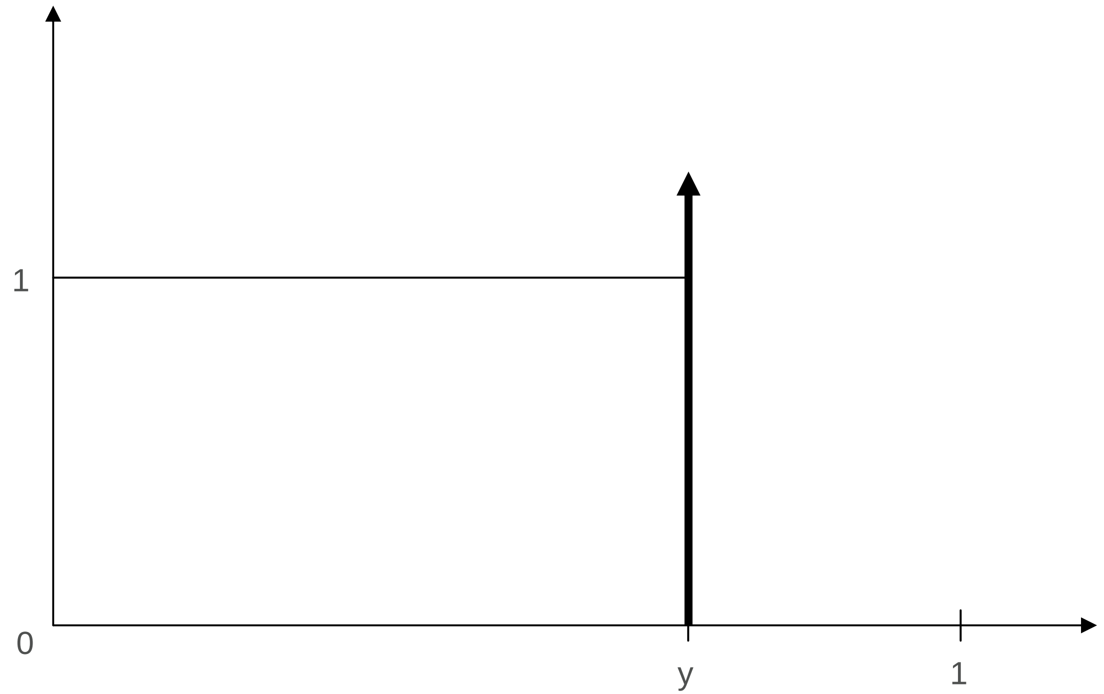
Decreasing order of k_j

Example with α=2

Worse Case Input Data

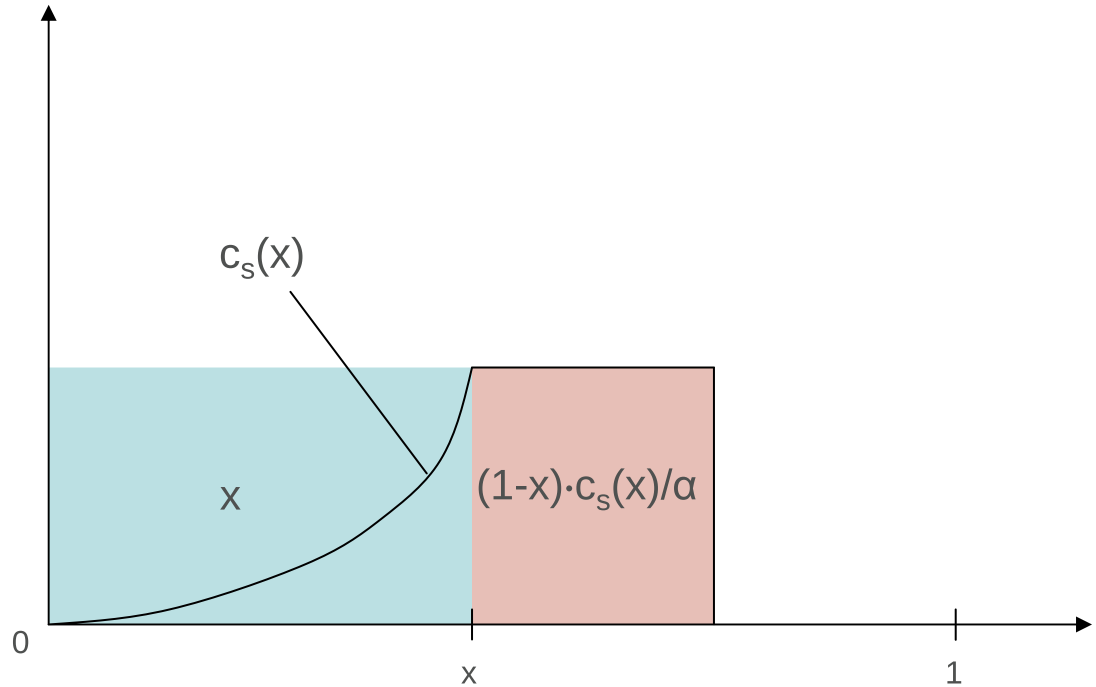
- Assume $y=\arg\{\max_x\{c_S(x)\}\}$.
  - $x \geq y$: Every job with that executes in the optimal schedule on a machine with a number greater than y is changed to a job with $k_j=y$.
    - The makespan of the optimal schedule remains unchanged.
    - $C_{max}(S)$ cannot decrease as jobs with $k_j>y$ do not contribute to $c_s(y)$.
  - $x<y$: If the eligibility bound k_j of a job J_j is less than the machine number x at which it is executed in the optimal schedule then k_j is increased to x.
    - The makespan of the optimal schedule remains unchanged.
    - $C_{max}(S)$ cannot decrease as this transformation can only increase the machine number on which a job is executed in schedule S.
- The job density function of worst case input data is 1 for $x<y$ and a Dirac pulse for $x=y$ such that the area of the pulse is $(1-y)$.

Job Density Function of Worst Case Input Data



Differential Equation of the Interval Approach

$$x = \int_0^x c_S(t)dt + \frac{(1-x)}{\alpha} \cdot c_S(x)$$



Differential Equation of the Interval Approach

$$x = \int_0^x c_S(t)dt + \frac{(1-x)}{\alpha} \cdot c_S(x)$$

$$1 = c_S(x) - \frac{1}{\alpha} \cdot c_S(x) + \frac{(1-x)}{\alpha} \cdot \frac{dc_S(x)}{dx}$$

$$\frac{dc_S(x)}{dx} \cdot \frac{1-x}{\alpha} = 1 - c_S(x) \cdot \left(1 - \frac{1}{\alpha}\right)$$

$$\frac{dc_S(x)}{dx} = \frac{\alpha}{1-x} - \frac{\alpha}{1-x} \cdot \left(1 - \frac{1}{\alpha}\right) \cdot c_S(x) = \frac{1-\alpha}{1-x} \cdot c_S(x) + \frac{\alpha}{1-x}$$

Solution of the Differential Equation

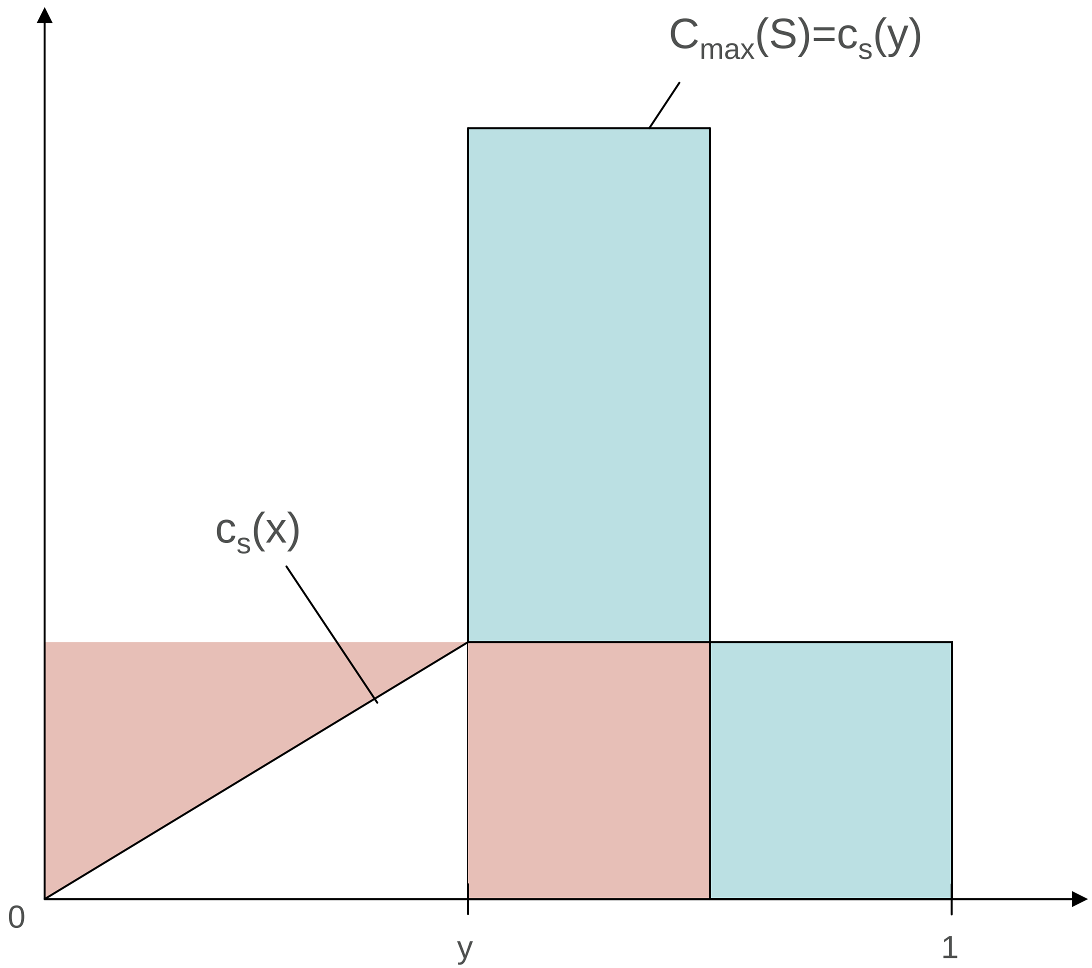
$$c_S^h(x) = \exp\left(\int \frac{1-\alpha}{1-x}\,dx\right) = \exp\big((\alpha-1)\cdot\ln(1-x)\big) = (1-x)^{\alpha\text{-}1}$$

$$c_S^p(x) = (1-x)^{\alpha\text{-}1}\cdot\int \frac{\frac{\alpha}{1-x}}{(1-x)^{\alpha\text{-}1}}\,dx = (1-x)^{\alpha\text{-}1}\cdot\frac{\alpha}{\alpha-1}\cdot(1-x)^{1\text{-}\alpha} = \frac{\alpha}{\alpha-1}$$

$$c_S(x) = C\cdot c_S^h(x) + c_S^p(x) = C\cdot(1\text{-}x)^{\alpha\text{-}1} + \frac{\alpha}{\alpha-1}$$

$$c_S(0) = 0 \Rightarrow C = \frac{\alpha}{1-\alpha} \qquad c_S(x) = \frac{\alpha}{1-\alpha}\cdot\left((1-x)^{\alpha\text{-}1} - 1\right)$$

Interval Schedule S for α=2 and a Given y



Determination of the Optimal Value for α

$$f(y) = c_S(y) + \alpha \Rightarrow \frac{df(y)}{dy} = \alpha \cdot (1-y)^{\alpha-2} > 0$$

$$y \to 1 \Rightarrow f(y) \to \frac{\alpha}{\alpha-1} + \alpha = \frac{\alpha^2}{\alpha-1}$$

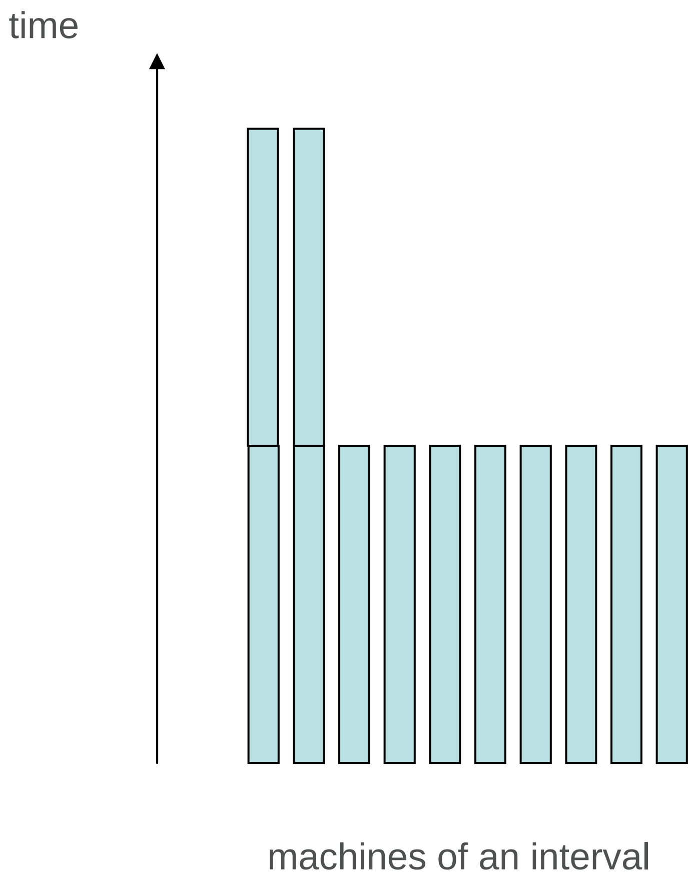
$$\frac{d\frac{\alpha^2}{\alpha-1}}{d\alpha} = \frac{(\alpha-1)\cdot 2\alpha - \alpha^2}{(\alpha-1)^2} = \frac{\alpha^2 - 2\alpha}{(\alpha-1)^2} \Rightarrow \alpha = 2$$

$$y \to 1 \Rightarrow f(y) \to \frac{2}{2-1} + 2 = 4$$

Transformation to the Discrete Case

- The interval allocation is approximated by allowing a job to be allocated to a machine if the continuous interval of the job would contain at least of fraction of this machine and afterwards applying list scheduling.
  - Some machines may receive more total processing time than in the continuous case.
  - The additional processing time is upper bounded by the processing time of the longest job.
- Due to the different processing times of the individual jobs, not all machines of an interval may achieve the same makespan although this might have happen in the continuous case.
  - The difference between the makespan of such two machines is at most the processing time of the longest job (list scheduling).
- Altogether the approximation cannot increase the competitive factor by more than 1.

Consequence of the Approximation



Conclusion

- For very large systems, online job scheduling on parallel identical machines with ordered machine eligibility achieves a competitive factors of 5.
- For the proof, we used a generalization to a continuous case and derived and solved a differential equation.
- We approximated the continuous case by simply applying list scheduling.
- For systems with few machines, the competitive factor is smaller as the value of y in the continuous case is bounded by m-1.