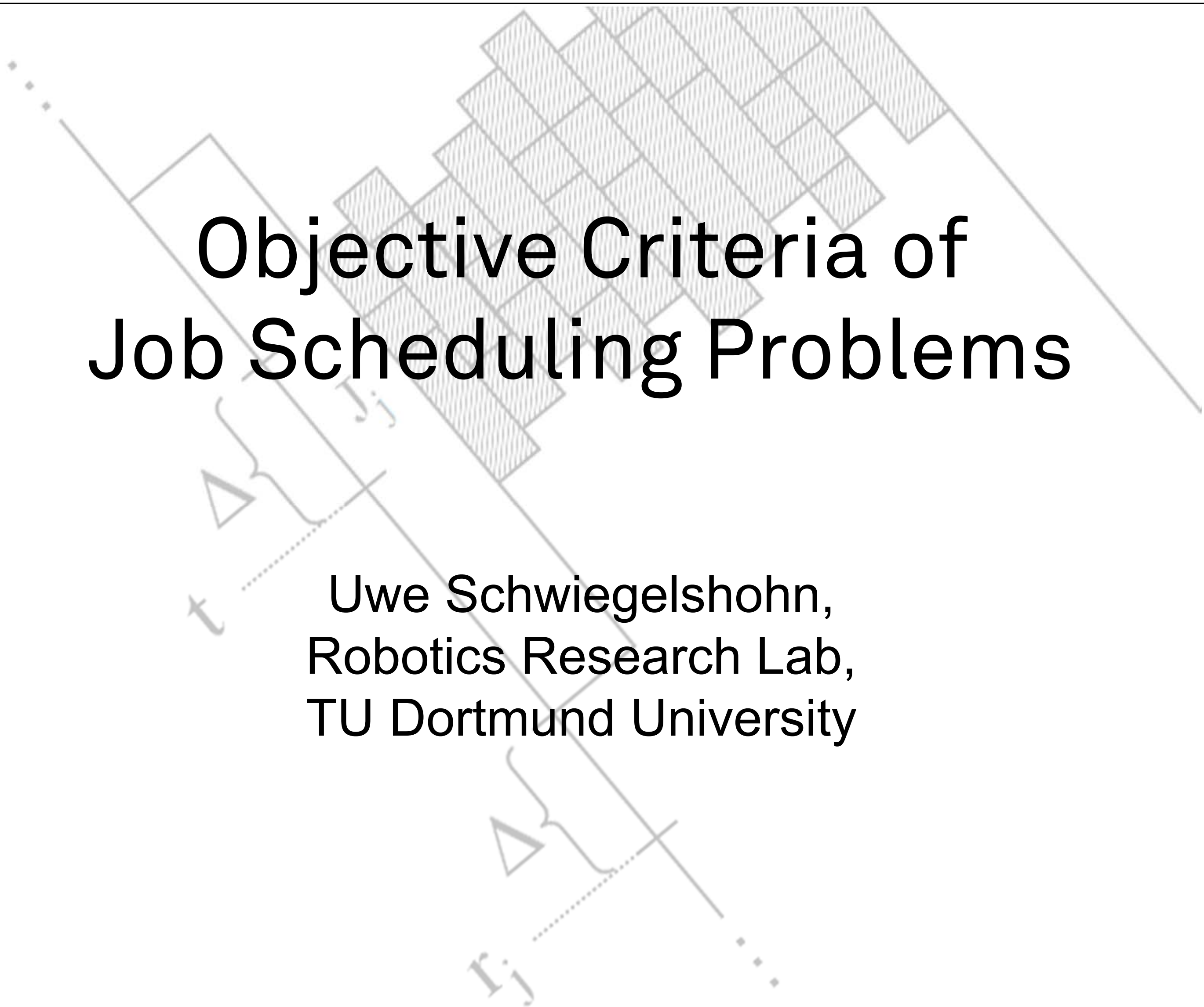


Objective Criteria of Job Scheduling Problems

Uwe Schwiegelshohn,
Robotics Research Lab,
TU Dortmund University

Jobs and Users in Job Scheduling Problems

- Independent users
  - No or unknown precedence constraints between different jobs
- Online scheduling
  - Jobs are unknown until they are submitted ($r_{j,online}$-condition).
- Nonclairvoyant scheduling
  - The processing time of a job is unknown until its completion (ncv-condition).
- Coarse granular scheduling
  - Fine granular scheduling is responsibility of the user or the OS of the user (virtualization).
  - A job is a single entity or consists of few stages.
- Resource (pre-)selection by the user
  - A job requires more than one machine in parallel ($size_j$-condition).

Machines in Job Scheduling Problems

- Cloud federation or computational grid (often GP_m-model)
  - Job allocation to a group of machines (central allocation or bids of different owners)
- A group of machines belongs to a single owner (often P_m-model).
  - System centric primary objective
  - Secondary objectives may consider user interests (service level agreements).
- Heterogeneity within a group of machines is usually invisible to the user.
  - Storage system, processors and cores.
  - Consideration of the dominant resource (virtual machine)
- Virtually exclusive access to machines
  - Machine sharing (fine grain preemption) is invisible to the user.

What is the Purpose of an Objective Criterion?

- Primary properties
  - Appropriate representation of **system** goals
  - Quantitative evaluation of the schedule
- Secondary characteristics
  - Easy evaluation ← important for online scheduling
    - Little volatility
  - Robustness regarding different scenarios
    - The overhead to determine general results and/or develop frameworks must pay off.
  - Extensibility to include more complex criteria
    - Inclusion of secondary user criteria

Analysis of Job Scheduling Problems

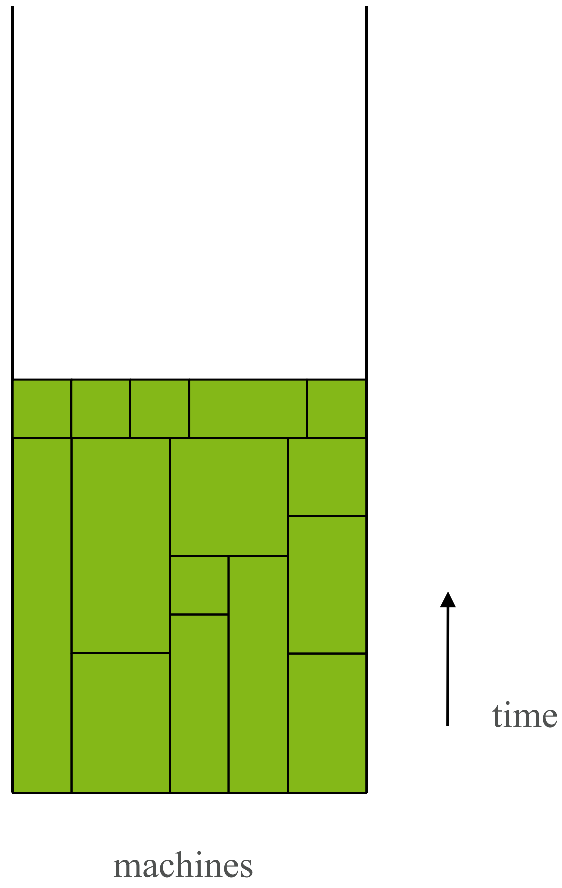
- Evaluation on a real system
  - Real systems of sufficient size are rarely available for experiments.
- Simulation experiments
  - Sampling of the solution space
  - Selection of input data to generate a good cover of the solution space.
    - Random data often do not represent real problems.
    - Only few real workload data are available.
- Theoretical evaluation
  - Stochastic scheduling
    - Real workload cannot be modeled by simple distributions
  - Competitive analysis

Competitive Analysis

- Worst case analysis
  - Information about stability of the approach
  - Possibly little indication about applicability in practice
- Similarity to approximation algorithms
  - Determination of a competitive factor
- Methodology
  - For all problem instances, we determine an upper bound for the ratio between
    - the objective value of the schedule generated by the algorithm to
    - the objective value of the optimal schedule for this instance.
- Example for makespan
  - $C_{max}(S) \leq c \cdot C_{max}(OPT)$ for all instances with c being the competitive factor

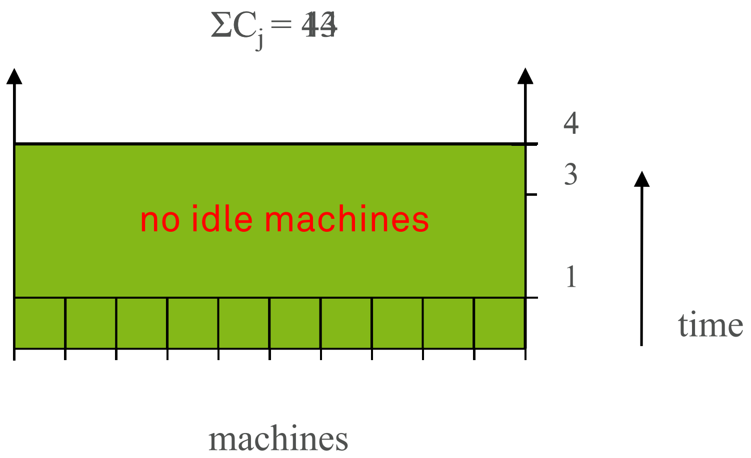
A Common Objective: Makespan

- Makespan corresponds to machine utilization.
- It is easy to determine the makespan of a schedule.
- Schedules with an optimal makespan may not be *good* schedules.
- It is difficult to incorporate secondary objectives.
- The objective may be highly volatile in an online scenario.



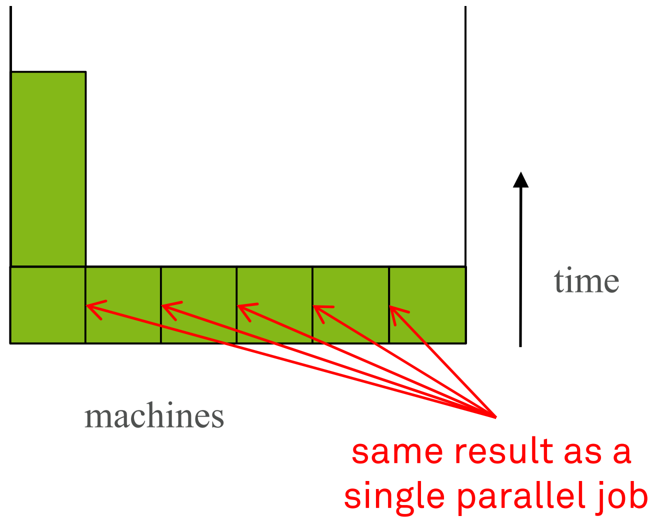
Another Common Objective: Total Completion Time

- The total completion time objective considers all jobs.
- Little volatility
- There may be different completion time results even in optimally utilized schedules.
  - Bias towards certain schedules
- Extension with job weights
  - Who selects the weights?



Total Completion Time with Resource Weights

- $w_j = p_j$ (sequential jobs)
  $w_j = p_j \cdot size_j$ (parallel jobs)
- Some known analysis results (see Queyranne and Kawaguchi and Kyan)
- Unbiased if the resource occupation remains unchanged
  - vertically dividing a parallel job does not change the objective.
  - horizontally dividing a long running job is invariant of the schedule.



Selection of a Reference Value

- Utilization must particularly address time periods of high demand (and their neighboring time periods).
- Makespan: Start time (time 0) of the schedule
  - In an online scenario, the makespan is lower bounded by the last release date plus the corresponding processing time.
    - Simple transformation of offline results into online results (see Shmoys et al.)
- Completion time (C_j) or flow time (C_j-r_j)?
  - Same optimal schedule
  - Significant differences in competitive factors (see Kellerer et al. and Becchetti and Leonardi)
  - System representation: completion time with an appropriate start time (see makespan)
  - User representation: flow time

$P_m|r_{j,online}, ncv|*$

- Comparison using the competitive factor for list scheduling

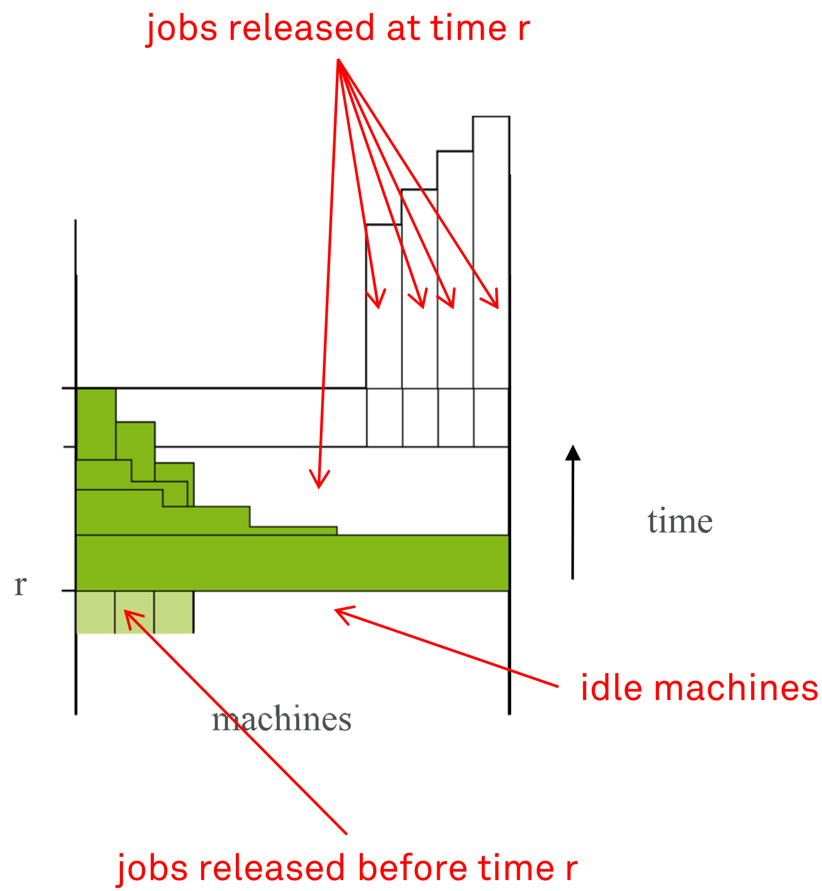
- Makespan ($*= C_{max}$)
  - 2-1/m (tight bound, see Graham)

- Utilization until the actual time ($*=$ occupied machine time / total machine time = actual time · number of machines)
  - 1.333 (tight bound, see Hussein et al.)

- Resource weight metric ($*= \sum p_j \cdot C_j$)
  - 1.25 (small gap, 1.207 is a lower bound, see Kawaguchi and Kyan)

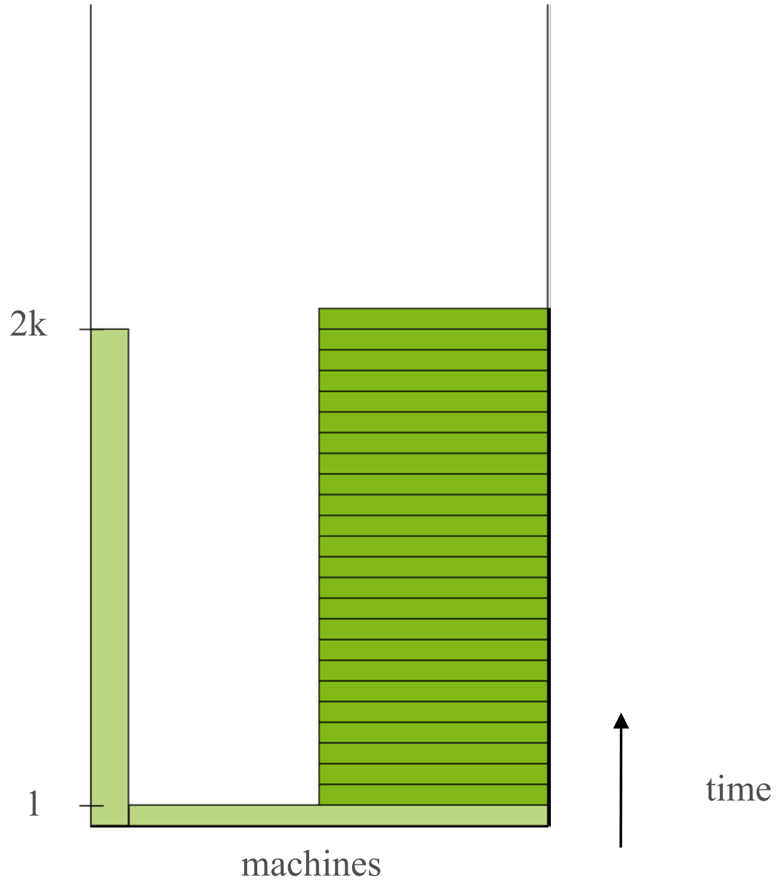
$P_m | r_{j,online}, ncv | \sum p_j C_j$

- The analysis helps to determine an appropriate machine overprovisioning in the system.
- Induction by the number of different release dates
- Use of the utilization result
  - At most 25% of the resources in an interval are left idle by list scheduling and are used in the optimal schedule.



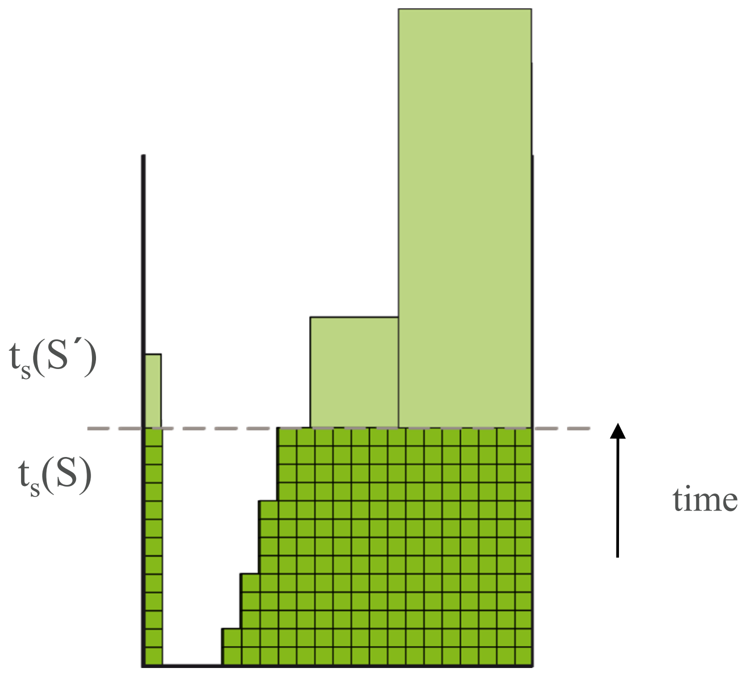
P_m|$size_j$,ncv|*

- Makespan (*= C_{max})
  - 2-1/m (tight bound, see Graham)
- Utilization
  - 1.333 until C_{max}(OPT)
- Resource weight metric (*= $\sum p_j \cdot C_j$)
  - 2 (jobs are scheduled in decreasing degree of parallelism)



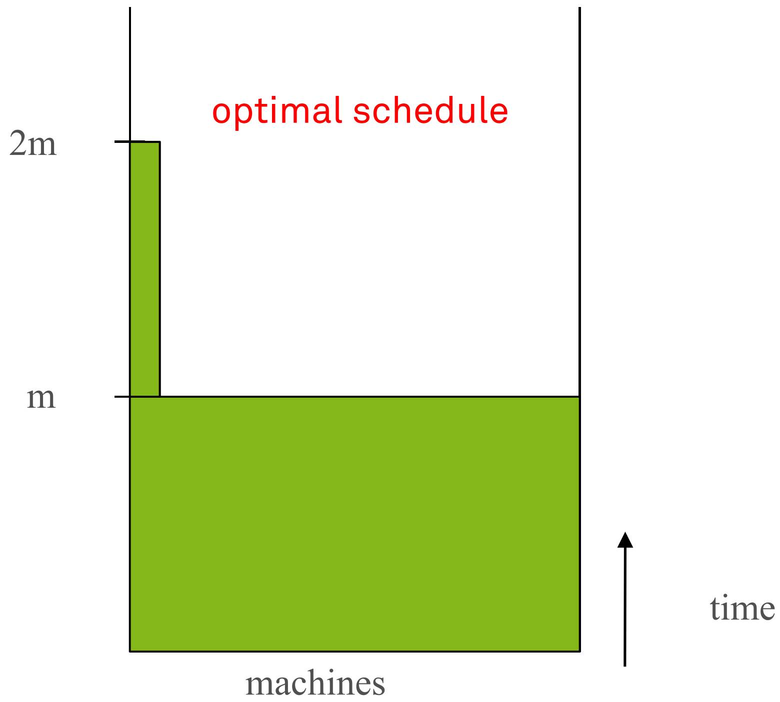
$P_m|size_j, ncv|\sum p_j \cdot size_j \cdot C_j$

- Vertical splitting of parallel jobs
- Horizontal splitting of some long jobs
  - no reduction of the competitive factor
- Combination of the remaining long jobs
  - neutral to the objective value
- Determination of the competitive factor by numerical optimization (2 variables)



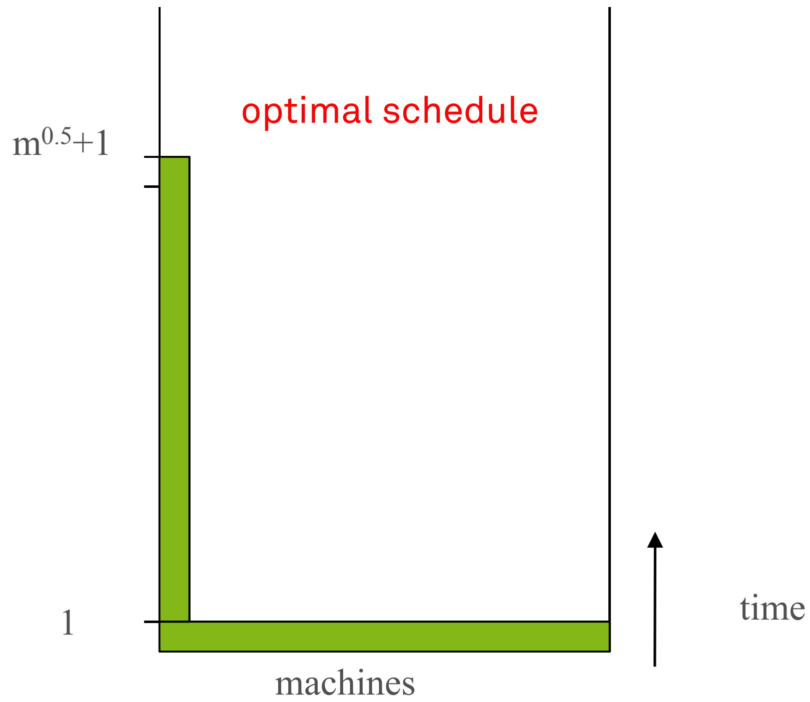
$P_m | r_{j,online}, size_j, ncv | *$

- Makespan ($* = C_{max}$)
  - 2-1/m (tight bound, see Naroska et al.)
- Utilization until the actual time m
  - $\Omega(m)$



$P_m | r_{j,online}, size_j, ncv | *$

- Makespan ($* = C_{max}$)
  - 2-1/m (tight bound, see Naroska et al.)
- Utilization until the actual time
  - $\Omega(m)$
- Resource weight metric ($* = \sum p_j \cdot C_j$)
  - 2 if $size_j \leq m/2$ for all jobs (see Turek et al.)
  - $\Omega(m^{0.5})$ in the general case
  - 3.562 with fine granular preemption (see Schwiegelshohn and Yahyapour)



Challenges and Status

- Discussion of several metrics for online nonclairvoyant job scheduling problems — done
- Comparison of the metrics based on competitive analysis — mostly done
- Testing of the metrics for real multiprocessor scheduling — mostly open
  - Real workloads
  - Heuristic algorithms
- Extension of the metrics — open
  - Different job classes with additional weight factors
  - Consideration of service level agreements