Advanced Algorithms

Nicole Megow (Universität Bremen)

SoSe 2025

Scheduling Problems

Lecture 09

Scheduling

Definition Lawler, Lenstra, Rinnooy Kan and Shmoys (1993)

Sequencing and scheduling is concerned with the optimal allocation of scarce resources to activities over time.

Very general field of computer science and operations research, active since 1960s, in theory and practice:

- Proccesses to CPUs on a machine
- Workloads to machines in a cloud system
- Clients to taxis, pizza orders to drivers, students to projects
- Production planning, project management

General scheduling representation

- set of m machines: $\{M_1, \ldots, M_m\}$ or $\{1, \ldots, m\}$
- set of n jobs: $\{J_1, \ldots, J_n\}$ or $\{1, \ldots, n\}$
- A job j has processing time p_{ij} when processing on machine i.
- Schedule typically represented by Gantt chart (Henry Gantt 1910s)

Gantt chart: 3 machines, 7 jobs

Scheduling terminology

A schedule is an assignment of jobs to machines such that no machine can process more than one job at a time t.

Non-preemptive jobs: the processing of a job must not be interrupted

Preemptive jobs: The processing of a job may be interrupted and resumed later even on a different machine

What do we optimize?

- Minimizing the makespan (aka load balancing): minimize the latest completion time over all jobs, $C_{\max} := max_{j \in [n]} C_j$
- Minimize the sum of completion times $\sum_{j \in [n]} C_j$
- and many more

Universität Bremen

Classification of scheduling problems

Three field notation $\alpha|\beta|\gamma$ introduced by Graham et al. (1979)

α is the machine environment:

- 1 – a single machine;
- P – m identical machines;
- Q – related machines with speeds;
- R – unrelated machines.

β is the job specification:

- r_j, d_j, $p_j = 1$ (unit-size jobs), $p_j = p$ (equal-size jobs);
- **pmtn** – jobs can be preempted;
- **prec** – precedence constraints (partial order for processing);

γ is the objective function:

- $C_{\max}$ – minimize makespan.
- $\sum w_j C_j$ – minimize weighted sum of completion times;

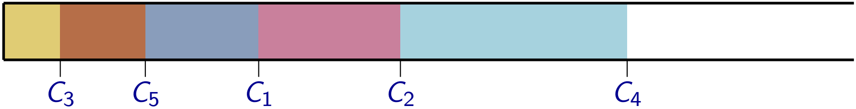
Min-Sum Scheduling

Minimizing the sum of completion times

Min-Sum Scheduling

Input: set of jobs with processing requirements p_j

Task: sequence jobs on a single machine



Objective: Minimize sum of completion times $\sum_j w_j C_j$.

Smith's Rule: Sequence jobs in non-increasing order of their ratios $\frac{w_j}{p_j}$.

Theorem

In an optimal schedule for $1 \mid\mid \sum w_j C_j$, jobs must be scheduled in order of Smith's Rule.

... at the board.

Universität Bremen

Unrelated Maschines

We consider the problem $R||\sum C_j$.

Input: m heterogenous machines,
n jobs: job j with processing time p_{ij} on machine i

Task: assign each job to one of the machines and order them

Objective: minimize the sum of completion times $\sum_{j=1}^{n} C_j$

Theorem (Horn 1973)

The scheduling problem $R \mid\mid \sum C_j$ can be solved in polynomimal time.

... at the board.

Minimizing the Makespan

Identical Parallel Machines

We consider the problem $P|\text{ptmn}|C_{\max}$.

Input: m identical parallel machines,
n jobs with processing times $p_1, \ldots, p_n$

Task: Schedule all jobs possibly using preemption and migration

Objective: Minimize the maximum load over all machines, where the load of machine i is $L_i = \sum_{j\,:\,j \to i} p_j$.

Denote $C_{\max} = \max_{i \in [m]} L_i$ as the makespan.

Identical and Unrelated Parallel Machines

At the board.

Theorem (McNaughton 1959)

The Wrap Around Rule solves the problem $P \mid \mathrm{pmtn} \mid C_{\max}$ optimally in polynomial time.

Jobs j may have individual release dates r_j, denoting the earliest possible starting time.

Theorem (Horn 1974)

The problem $P \mid r_j, \mathrm{pmtn} \mid C_{\max}$ can be solved optimally in polynomial time.

Summary & Outlook

This lecture

- Introduction to scheduling
- Min-sum scheduling:
  - $1|(\text{pmtn})|\sum w_j C_j$ Smith's Rule
  - $R||\sum C_j$ assignment problem, matching
- Makespan
  - $P|\text{pmtn}|C_{\max}$ McNaughton's Wrap Around Rule
  - $P|r_j, \text{pmtn}|C_{\max}$ Maximum flow & McNaughton's Rule

Outlook

- Most other scheduling problems are NP-hard.
  $\rightarrow$ approximation algorithms
- How to cope with uncertainty in the input data (e.g., processing time, machine speed)?