DM865 – Spring 2018

Heuristics and Approximation Algorithms

Introduction to Scheduling: Terminology and Classification

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

1. Definitions
2. Classification
3. Exercises
4. Schedules

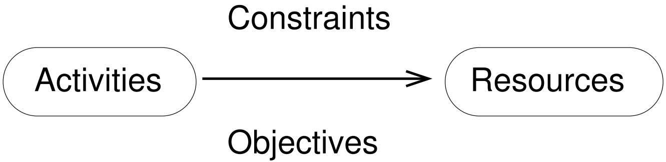
Scheduling

- Manufacturing
  - Project planning
  - Single, parallel machine and job shop systems
  - Flexible assembly systems
    Automated material handling (conveyor system)
  - Lot sizing
  - Supply chain planning

- Services
  - personnel/workforce scheduling
  - public transports

$\Rightarrow$ different models and algorithms

Problem Definition



Problem Definition

Given: a set of jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$ to be processed by a set of machines $\mathcal{M} = \{M_1, \ldots, M_m\}$.

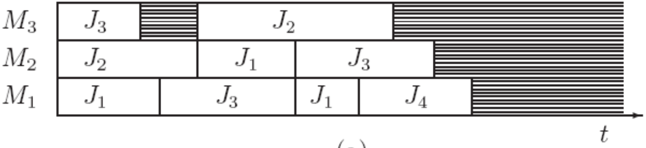
Task: Find a schedule, that is, a mapping of jobs to machines and processing times, that satisfies some constraints and is optimal w.r.t. some criteria.

Notation:
n, j, k jobs
m, i, h machines

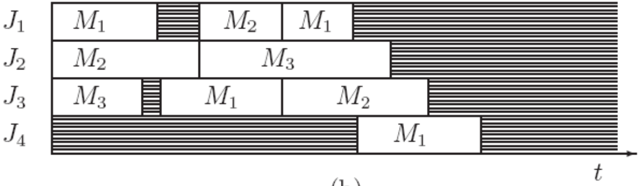
Visualization

Scheduling are represented by Gantt charts

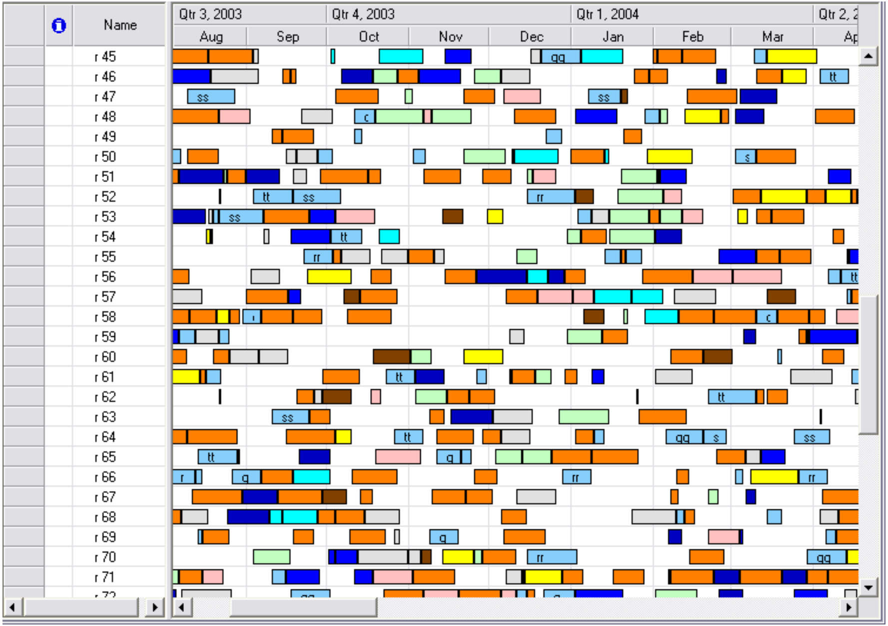
(a) machine-oriented
(b) job-oriented



(a)



(b)


Name
Qtr 3, 2003
Qtr 4, 2003
Qtr 1, 2004
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
r 45
r 46
r 47
r 48
r 49
r 50
r 51
r 52
r 53
r 54
r 55
r 56
r 57
r 58
r 59
r 60
r 61
r 62
r 63
r 64
r 65
r 66
r 67
r 68
r 69
r 70
r 71

Data Associated to Jobs

- Processing time p_{ij}
- Release date r_j
- Due date d_j (called deadline, if strict)
- Weight w_j
- Cost function $h_j(t)$ measures cost of completing J_j at t
- A job J_j may also consist of a number n_j of operations $O_{j1}, O_{j2}, \ldots, O_{jn_j}$ and data for each operation.
- A set of machines $\mu_{jl} \subseteq \mathcal{M}$ associated to each operation
  - $|\mu_{jl}| = 1$ dedicated machines
  - $\mu_{jl} = \mathcal{M}$ parallel machines
  - $\mu_{jl} \subseteq \mathcal{M}$ multipurpose machines

Data that depend on the schedule

- Starting times S_{ij}
- Completion time C_{ij}, C_j

Outline

1. Definitions
2. Classification
3. Exercises
4. Schedules

Problem Classification

A scheduling problem is described by a triplet $\alpha \mid \beta \mid \gamma$.

- α machine environment (one or two entries)
- β job characteristics (none or multiple entry)
- γ objective to be minimized (one entry)

[R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan (1979): Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 4, 287-326.]

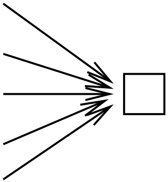
$\alpha \mid \beta \mid \gamma$ Classification Scheme

Machine Environment

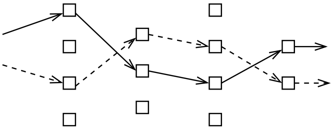
$\boldsymbol{\alpha_1 \alpha_2} \mid \beta_1 \ldots \beta_{13} \mid \gamma$

- single machine/multi-machine ($\alpha_1 = \alpha_2 = 1$ or $\alpha_2 = m$)
- parallel machines: identical ($\alpha_1 = P$), uniform p_j/v_i ($\alpha_1 = Q$), unrelated p_j/v_{ij} ($\alpha_1 = R$)
- multi operations models: Flow Shop ($\alpha_1 = F$), Open Shop ($\alpha_1 = O$), Job Shop ($\alpha_1 = J$), Mixed (or Group) Shop ($\alpha_1 = X$), Multi-processor task sched.

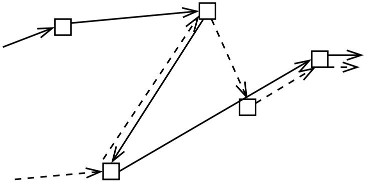
Single Machine



Flexible Flow Shop ($\alpha = FFc$)



Open, Job, Mixed Shop



$\alpha \mid \beta \mid \gamma$ Classification Scheme

Job Characteristics

$\alpha_1\alpha_2 \mid \beta_1 \ldots \beta_{13} \mid \gamma$

- $\beta_1 = prmp$ presence of preemption (resume)
- β_2 precedence constraints between jobs acyclic digraph $G = (V, A)$
  - $\beta_2 = prec$ if G is arbitrary
  - $\beta_2 = \{chains, intree, outtree, tree, sp\text{-}graph\}$
- $\beta_3 = r_j$ presence of release dates
- $\beta_4 = p_j = p$ preprocessing times are equal
- ($\beta_5 = d_j$ presence of deadlines)
- $\beta_6 = \{s\text{-}batch, p\text{-}batch\}$ batching problem
- $\beta_7 = \{s_{jk}, s_{jik}\}$ sequence dependent setup times

$\alpha \mid \beta \mid \gamma$ Classification Scheme

Job Characteristics (2)

$\alpha_1\alpha_2 \mid \boldsymbol{\beta_1 \ldots \beta_{13}} \mid \gamma$

- $\beta_8 = brkdwn$ machine breakdowns
- $\beta_9 = M_j$ machine eligibility restrictions (if $\alpha = Pm$)
- $\beta_{10} = prmu$ permutation flow shop
- $\beta_{11} = block$ presence of blocking in flow shop (limited buffer)
- $\beta_{12} = nwt$ no-wait in flow shop (limited buffer)
- $\beta_{13} = recrc$ recirculation in job shop

$\alpha \mid \beta \mid \gamma$ Classification Scheme

Objective (always $f(C_j)$) $\alpha_1\alpha_2 \mid \beta_1\beta_2\beta_3\beta_4 \mid \gamma$

- Lateness $L_j = C_j - d_j$
- Tardiness $T_j = \max\{C_j - d_j, 0\} = \max\{L_j, 0\}$
- Earliness $E_j = \max\{d_j - C_j, 0\}$
- Unit penalty $U_j = \begin{cases} 1 & \text{if } C_j > d_j \\ 0 & \text{otherwise} \end{cases}$

$\alpha \mid \beta \mid \gamma$ Classification Scheme

Objective

$\alpha_1\alpha_2 \mid \beta_1\beta_2\beta_3\beta_4 \mid \gamma$

- Makespan: Maximum completion $C_{max} = \max\{C_1, \ldots, C_n\}$
  tends to max the use of machines
- Maximum lateness $L_{max} = \max\{L_1, \ldots, L_n\}$
- Total completion time $\sum C_j$ (flow time)
- Total weighted completion time $\sum w_j \cdot C_j$
  tends to min the av. num. of jobs in the system, ie, work in progress, or also the throughput time
- Discounted total weighted completion time $\sum w_j(1 - e^{-rC_j})$
- Total weighted tardiness $\sum w_j \cdot T_j$
- Weighted number of tardy jobs $\sum w_j U_j$

All regular functions (nondecreasing in $C_1, \ldots, C_n$) except E_i

$\alpha \mid \beta \mid \gamma$ Classification Scheme

Other Objectives

$\alpha_1\alpha_2 \mid \beta_1\beta_2\beta_3\beta_4 \mid \gamma$

Non regular objectives

- Min $\sum w'_j E_j + \sum w"_j T_j$ (just in time)
- Min waiting times
- Min set up times/costs
- Min transportation costs

Outline

Exercises

- TSP: $1 \mid s_{jk} \mid C_{\max}$
- Knapsack: $1 \mid d_j = d \mid \sum w_j U_j$
- Project planning (CPM, PERT): $P\infty \mid prec \mid C_{\max}$
- $Jm \mid\mid C_{\max}$
- $Fm \mid p_{ij} = p_j \mid C_{\max}$
- $FJc \mid r_j, s_{ijk} \mid \sum w_j T_j$
- $1 \mid r_j, prmtn | L_{\max}$

Outline

1. Definitions
2. Classification
3. Exercises
4. Schedules

Solutions

Distinction between

- sequence
- schedule
- scheduling policy

If no preemption allowed, schedule defined by vector $S = (S_i)$

Feasible schedule

A schedule is feasible if no two time intervals overlap on the same machine, and if it meets a number of problem specific constraints.

Optimal schedule

A schedule is optimal if it is feasible and it minimizes the given objective.

Classes of Schedules

Semi-active schedule

A feasible schedule is called semi-active if no operation can be completed earlier without changing the order of processing on any one of the machines. (local shift)

Active schedule

A feasible schedule is called active if it is not possible to construct another schedule by changing the order of processing on the machines and having at least one operation finishing earlier and no operation finishing later. (global shift without preemption)

Nondelay schedule

A feasible schedule is called nondelay if no machine is kept idle while an operation is waiting for processing. (global shift with preemption)

- There are optimal schedules that are nondelay for most models with regular objective function.
- There exists for $Jm||\gamma$ (γ regular) an optimal schedule that is active.
- nondelay $\Rightarrow$ active but active $\not\Rightarrow$ nondelay

Summary

- Scheduling Definitions (jobs, machines, Gantt charts)
- Classification
- Classes of schedules