

Concepts

Helmut Simonis

email: helmut.simonis@insight-centre.org
homepage: <http://insight-centre.org/>

ENTIRE EDIH
Insight SFI Centre for Data Analytics
School of Computer Science and Information Technology
University College Cork
Ireland

Constraint Based Production Scheduling

Acknowledgments

This publication was developed as part of the ENTIRE EDIH project, which received funding from Enterprise Ireland and the European Commission.

Part of this work is based on research conducted with the financial support of Science Foundation Ireland under Grant number 12/RC/2289-P2 at Insight the SFI Research Centre for Data Analytics at UCC, which is co-funded under the European Regional Development Fund.

Part of this work is based on research conducted within the ASSISTANT European project, under the framework program Horizon 2020, ICT-38-2020, Artificial intelligence for manufacturing, grant agreement number 101000165.

Key Points

- We introduce the core concepts used in scheduling
- Different layers of description
 - What we are doing (jobs, tasks, resources)
 - Why we are scheduling (orders, products, processes)
- Temporal Relations
- Process description
- Problem classification
- Visualization

1 Core Concepts

1.1 Jobs, Tasks and Resources

Most basic description of scheduling problem

- *Job*
 - Collection of activities required to manufacture one object/lot/order
 - Overall start/end determined by starts and ends of its tasks
- *Task*
 - Individual activities required for manufacture
 - Have defined start, end (typical: variables) and duration (sometimes fixed)
 - Often performed on one specific resource (more on that later)
- *Resources*
 - Resources are needed to perform the tasks
- Very compact representation of scheduling problem
- But, where does that information come from?

1.2 Orders, Products, Processes

Scheduling orders

- An *order* specifies a need for a certain *product* at a given time in a specific quantity
- There may be multiple ways of making the *product* (multiple *processes*)
- We assume that the process to use is decided when placing the order
- Each order corresponds to a job, with its constituent tasks
- There may be limited visibility of future orders

Process Description

- Each *process* consists of one or more *process steps*
- A process step contains a duration formula to describe how long it lasts
- The order of *process steps* is defined by *process sequences*
- The resources needed are defined by *resource needs* (described later on)
- Tasks are created for each process step, their duration is based on the duration formula and order quantity

Where do the orders come from?

- Made to order
 - Each order is caused by a customer request
 - Defines due date, release date often implied
- Made to stock
 - Orders are satisfied from stock
 - Inventory control strategy decides when to make product
 - Often called stock orders
 - More complex variant integrates production planning and detailed scheduling
 - Example later in course

2 Temporal Relations

Temporal Relations

- Temporal constraints between tasks and/or jobs
- Defined by the manufacturing process
- In simple cases
 - A single sequence of process steps performed in that order
 - Each task must finish before the next one can start



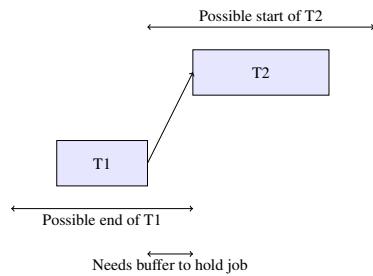
Annotations on Features

- ✓Currently available in scheduling tool
- (✓)Will be available shortly
- ✗Currently not available, may be added in future version

2.1 Relations between Tasks

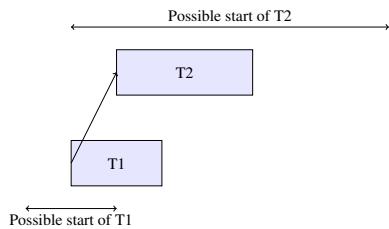
The Most Common Relation: EndBeforeStart ✓

- States that one task (T1) must end before the next one (T2) can start
- Typical for manufacturing process based on the same item
- Addition: offset
 - Wait at least offset units between end and start
 - For example cooling, drying time outside a machine



Less Common: StartBeforeStart ✓

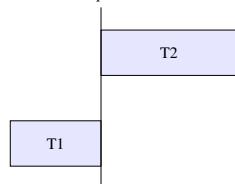
- States that one task (T2) can start any time after the start of another task (T1)
- Uncommon in manufacturing, occurs in project management
- Example later on on assembly line balancing



NoWait ✓

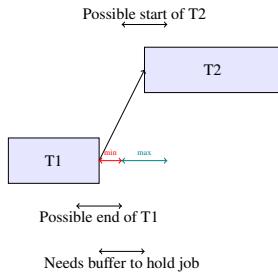
- Sometimes, two steps must follow each other immediately
- The item made would spoil
 - Product specific
- There is no space to hold item
 - Machine specific, buffers
- End of one task (T1) must be equal to start of next task (T2)
- May mean delay of start of task T1

Start of T2 is equal to End of T1



MaxWait (✓)

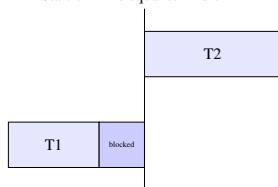
- Limit how long we can wait between tasks
 - Cooling enough, but not too much
 - Baking: rise time
- Impose both lower and upper waiting time limit
- Makes it more difficult to find solutions



Blocking ✓

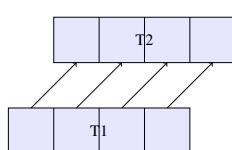
- Sometimes, two steps must follow each other immediately
- There is no space to store item between machines
- Keep item on previous machine until needed
- That machine is now *blocked*
- Duration of task T1 is extended until start of T2
- *Use with caution! Easy to deadlock*

Start of T2 is equal to End of T1



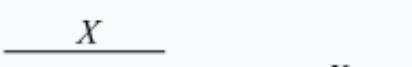
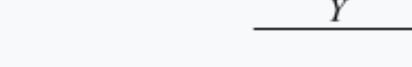
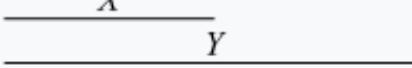
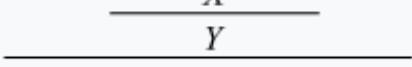
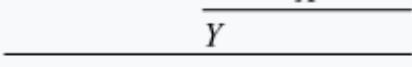
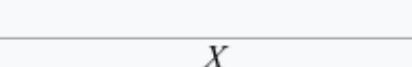
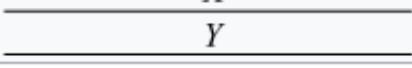
Special Case: Pipelining ✗

- Sometimes, we can start on the next task while the first is still running
- Possible if one job produces multiple items (lots,...)
- As soon as the first item is finished, take it to the next machine to process it there
- Overlaps T1 and T2 as much as possible
- Details can get complex



More General: Relations between Intervals

- First introduced by Allen (1983)
- 13 relations between intervals
- Allows composition of relations
- Constraint reasoning on sets of relations

Relation	Illustration	Interpretation
$X < Y$	 A horizontal line with 'X' at the left end and 'Y' at the right end.	X precedes Y
$Y > X$	 A horizontal line with 'Y' at the left end and 'X' at the right end.	Y is preceded by X
$X \mathbf{m} Y$	 A horizontal line with 'X' and 'Y' overlapping.	X meets Y
$Y \mathbf{mi} X$	 A horizontal line with 'Y' and 'X' overlapping.	Y is met by X (<i>i</i> stands for <i>inverse</i>)
$X \mathbf{o} Y$	 A horizontal line with 'X' and 'Y' overlapping.	X overlaps with Y
$Y \mathbf{oi} X$	 A horizontal line with 'Y' and 'X' overlapping.	Y is overlapped by X
$X \mathbf{s} Y$	 A horizontal line with 'X' to the left of 'Y'.	X starts Y
$Y \mathbf{si} X$	 A horizontal line with 'X' to the left of 'Y'.	Y is started by X
$X \mathbf{d} Y$	 A horizontal line with 'X' to the left of 'Y'.	X during Y
$Y \mathbf{di} X$	 A horizontal line with 'X' to the left of 'Y'.	Y contains X
$X \mathbf{f} Y$	 A horizontal line with 'X' to the left of 'Y'.	X finishes Y
$Y \mathbf{fi} X$	 A horizontal line with 'X' to the left of 'Y'.	Y is finished by X
$X = Y$	 A horizontal line with 'X' and 'Y' overlapping.	X is equal to Y

from Wikipedia: https://en.wikipedia.org/wiki/Allen%27s_interval_algebra

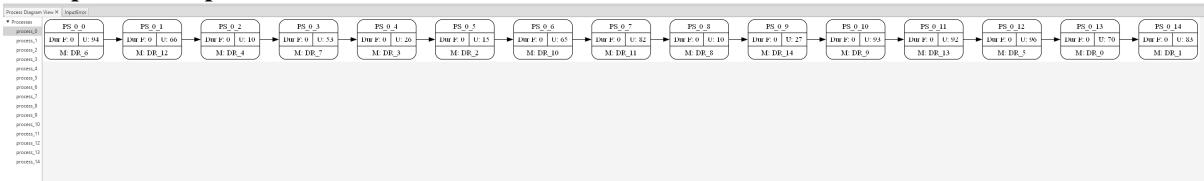
2.2 Relation between Tasks and Jobs

Start and End of Jobs

- The start of a job is equal to the start of the earliest task of the job

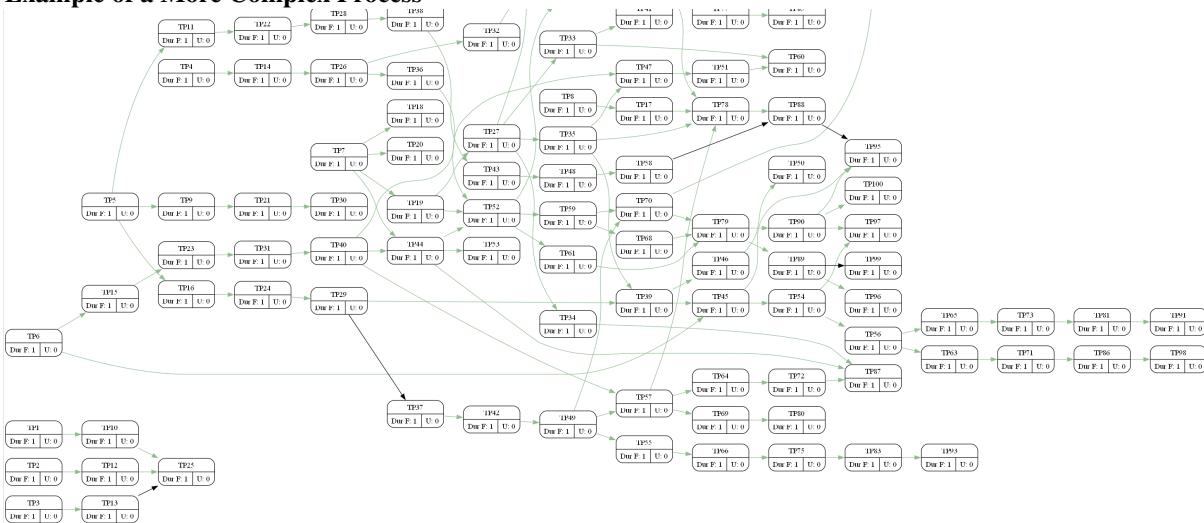
- The end of a job is equal to the latest end of any of its tasks
 - Also called: the job *spans* its tasks
 - Sometimes very simple
 - Start of job is start of first process step
 - End of job is end of last process step
 - But, do we know which steps will be first or last?

An Example of a Simple Process



- The steps form a precedence chain
 - Easy to identify first and last step

An Example of a More Complex Process



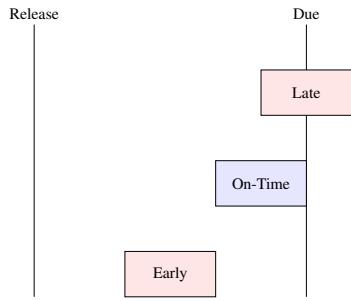
- There is no clear first or last process step

2.3 Jobs: Release and Due Date

Jobs: Release and Due Dates ✓

- The execution of a job may be constrained in time
 - *Release dates* states earliest time a job can start
 - *Due dates* states latest time a job can end
 - These may or may not be hard constraints!
 - A job will be *late* if it ends after the due date
 - A job will be *early* if it ends before the due date

- A job will be *on-time* if it ends at the due date



2.4 Relations between jobs

Relations between Jobs X

- There may be relations between jobs as well
- For example, jobs for the same product may be arranged by due date
- Do not allow to run job for a later due date before any job with an earlier due date
- Orders for the same customer, but different products, may be constrained
- Most common:
 - Jobs for intermediate products must finish in time for their use later on

3 Alternative Processes, Bill of Materials

More Complexity

- We have ignored a lot of potential complications
 - Alternative processes
 - Alternative process paths
 - Alternative resources
- Intermediate products
- Impact of raw material availability

Intermediate products X

- Some production operations are assembly steps
- Combine multiple intermediate products together
- These intermediate products need to be made as well
- There are processes for those products

Raw materials ✗

- Sometimes, a process step needs certain raw materials
- These are not made within the scheduled part of the plant
- They come from stock, inventory control problem
- Do we schedule production and then order raw materials?
- Do we schedule based on the available raw materials?

Bill of Materials (BoM), Bill of Processes ✗

- Enterprise systems will describe which items are needed to make a product
- Tree like structure, indicates the intermediate product/raw material needed and its quantity
- *BoM explosion* derive all required input materials for a given set of orders
- We may want to know at which step of process we need which materials (Bill of processes)
- This is where you use SAP, big database, trivial calculation
- Becomes hard if processes not fixed

4 Problem Classification

Problem Classification

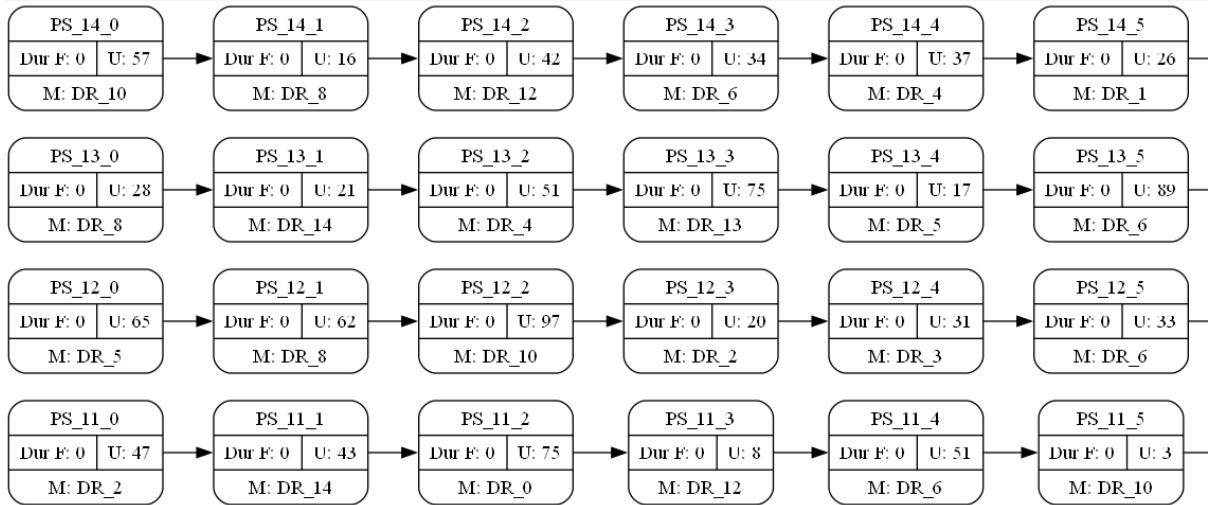
- Most real-world problems are messy, with many special conditions and exceptions
- Academic research prefers well-structured problems
- Scheduling research often focuses on well-structured problem types
 - Easier to understand
 - Possible to exploit structure
 - Easier to compare results
- A small number of problem types are very common in research

4.1 Job-Shop

Job-Shop ✓

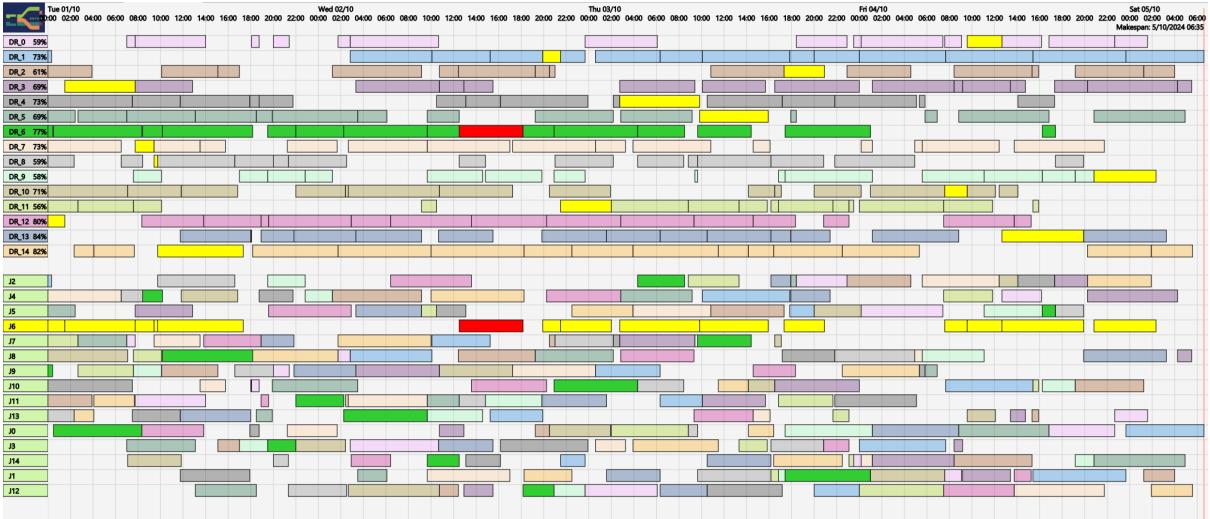
- Consists of a number of jobs and a number of machines
- Each job visits each machine, but possibly in a different order, depending on process
- Tasks of a job are linked as a precedence chain
- Objective is to minimize overall end, the *makespan*

Example Job-Shop Process



- Note that the order of machines visited is different for each process

Example Job-Shop Solution

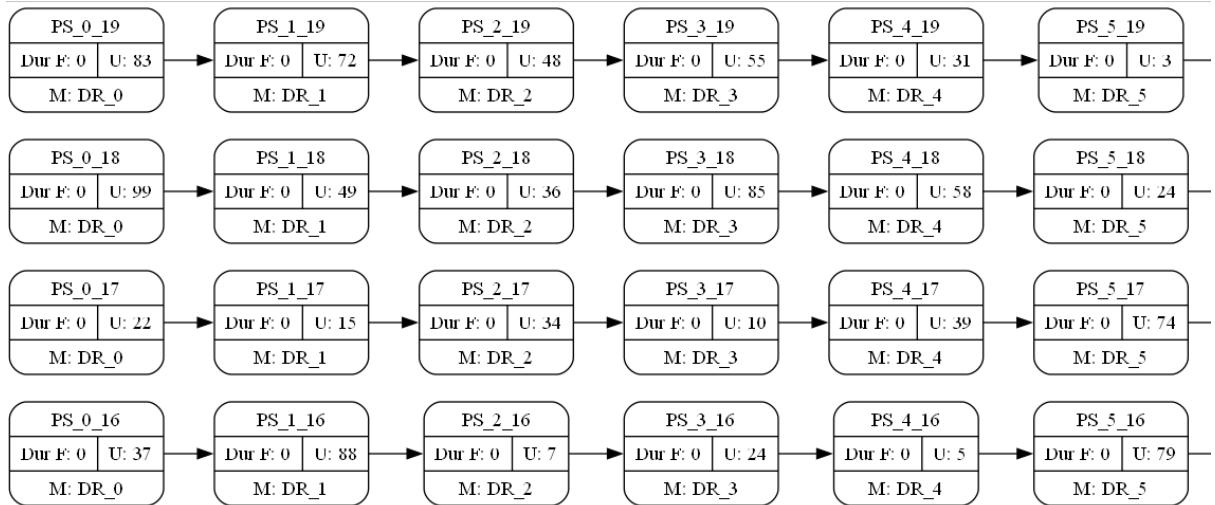


- One task is selected (in red), in both Machine and Job Gantt Chart
- Tasks are colored by machine, note coloring in jobs is different for each job

4.2 Flow-Shop

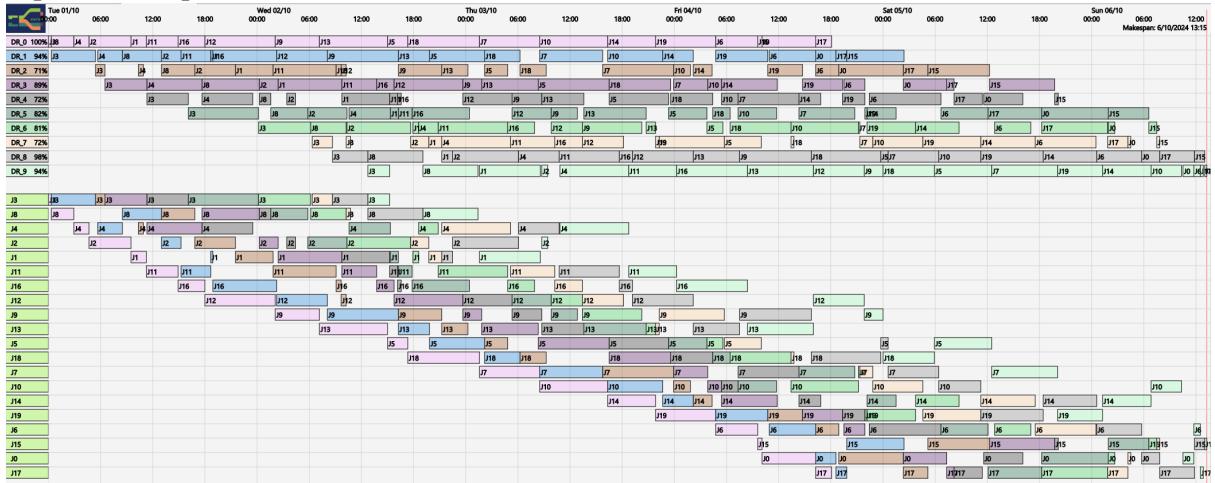
Flow-Shop ✓

- Consists of a number of jobs and a number of machines
- Each job visits each machine, all jobs in the same order
- Tasks of a job linked in a precedence chain
- Objective is to minimize overall end, the *makespan*



- Note that each process visits the machines in order DR_0, DR_1, ...

Example Flow-Shop Solution



- Tasks are colored by machine, note the regular pattern in the Job Gantt Chart

4.3 Open-Shop

Open-Shop ✓

- Consists of a number of jobs and a number of machines
- Each job visits each machine, we have to choose the sequence individually for each order
- There are no temporal constraints between tasks, but tasks of the same job cannot overlap
- Objective is to minimize overall end, the *makespan*

Open Shop Example Process

- Only showing details of one process
- No prescribed sequence between process steps

- Easier to find a task to run next
- Much larger search space

▼ Processes

process_0

process_1

process_2

process_3

process_4

process_5

process_6

PS_0_6	
Dur F: 0	U: 56
M: DR_4	

PS_0_5	
Dur F: 0	U: 92
M: DR_5	

PS_0_4	
Dur F: 0	U: 71
M: DR_0	

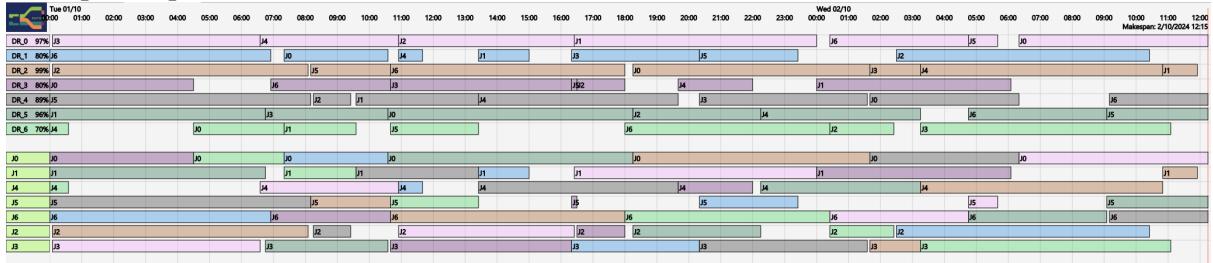
PS_0_3	
Dur F: 0	U: 34
M: DR_6	

PS_0_2	
Dur F: 0	U: 54
M: DR_3	

PS_0_1	
Dur F: 0	U: 39
M: DR_1	

PS_0_0	
Dur F: 0	U: 89
M: DR_2	

Open-Shop Example Solution



- Example solution for 7x7 open shop example
- Order of tasks within jobs not constrained
- Note that machines are still idle in optimal solution

4.4 RCPSP

Resource Constrained Project Scheduling Problem (RCPSP) (✓)

- Problem class from project management
- One project (one job), many tasks
- Precedence graph is arbitrary DAG
- Cumulative as well as disjunctive resources
- Variants with process alternatives

4.5 $\alpha/\beta/\gamma$ Notation

$\alpha/\beta/\gamma$ Notation

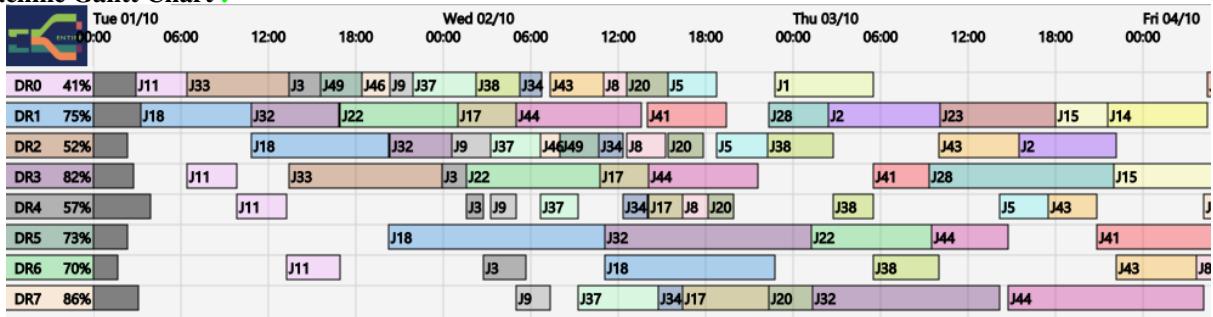
- The previous classes are good for algorithm research, but not very practical
- General scheme to describe problem type introduced in 1979
- Based on three parameters
 - α resource structure, stages
 - β temporal relations
 - γ objective
- $P2/r_j, \bar{d}_j/C_{\max}$: One stage, two identical parallel machines, hard release and due dates, objective makespan
- More detailed description at <https://encyclopedia.pub/entry/30497>

5 Key Visualization Methods

Visualization

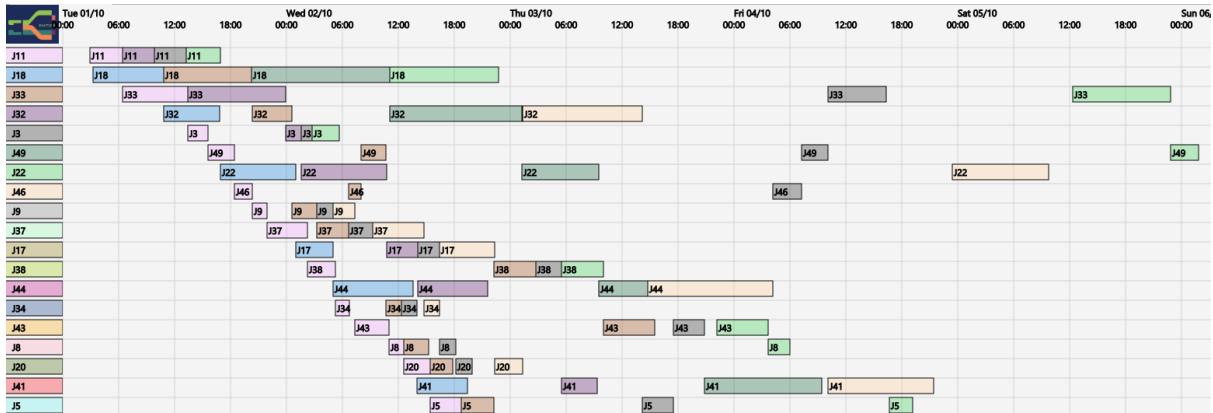
- Visualization is key to present and to understand results
- Many different ways to give an overview of schedule, and highlight problems
- Some diagrams types are used a lot, and are provided in our generic scheduling tool
- Customization is key

Machine Gantt Chart ✓



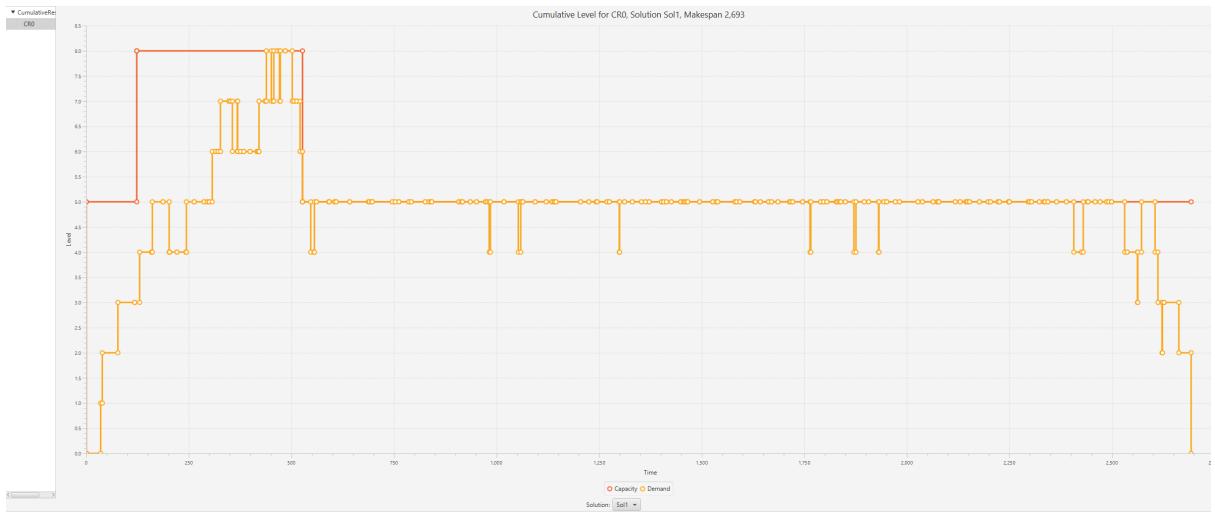
- Shows all tasks that are assigned to each machine
- Tasks should not overlap
- Also shows work in progress (WiP), down-times
- Optional display of setup and idle times

Job Gantt Chart ✓



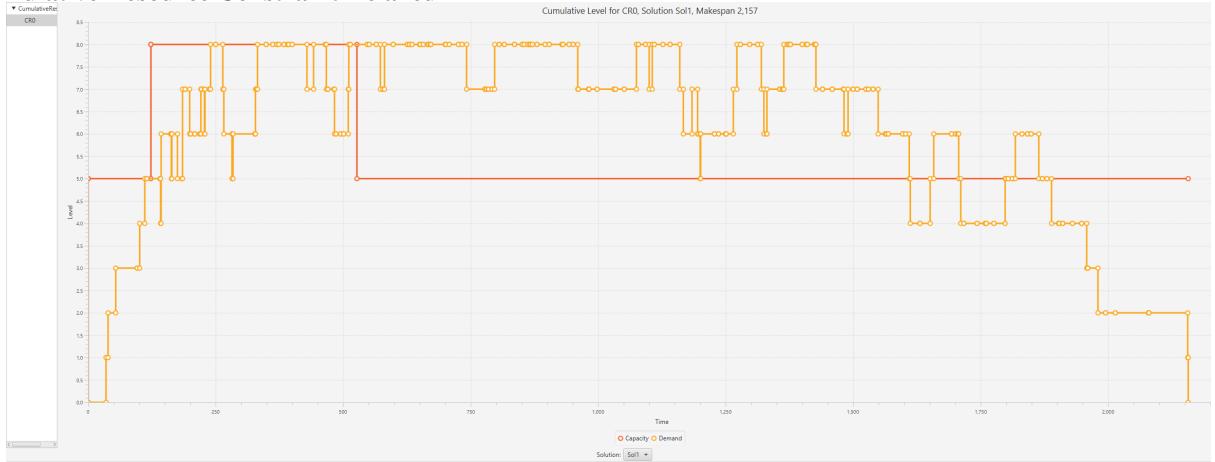
- Shows all tasks of a job in one line
- Only works for single chain of process steps
- Possible display of earliness, lateness
- Optional display of waiting and transport times

Cumulative Resource Chart ✓

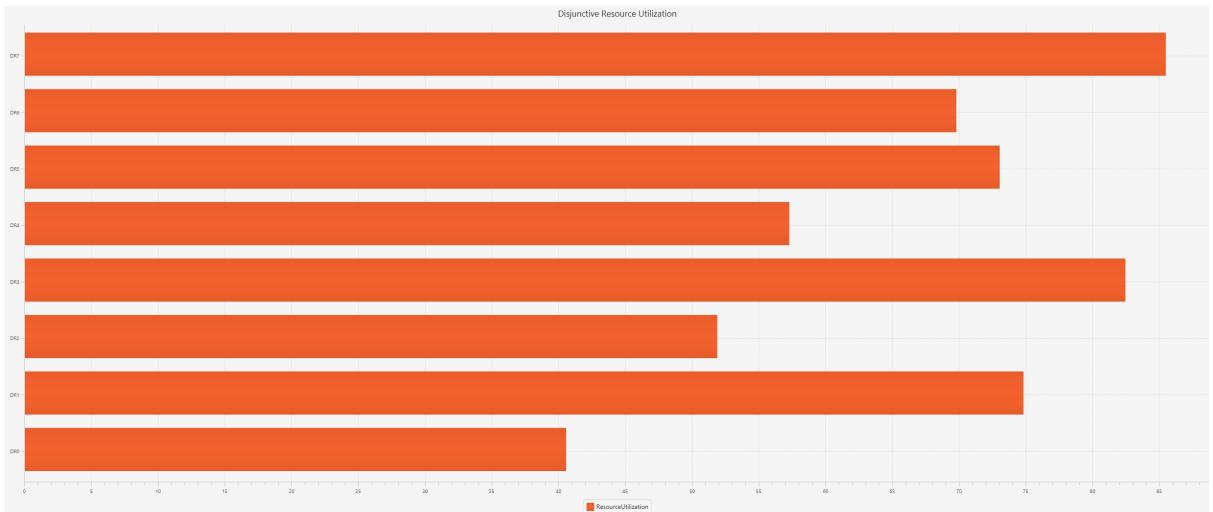


- Shows resource utilization of cumulative resource over time
- Utilization should be below capacity profile
- Unless we relax the cumulative resource constraint

Cumulative Resource Constraint Relaxed

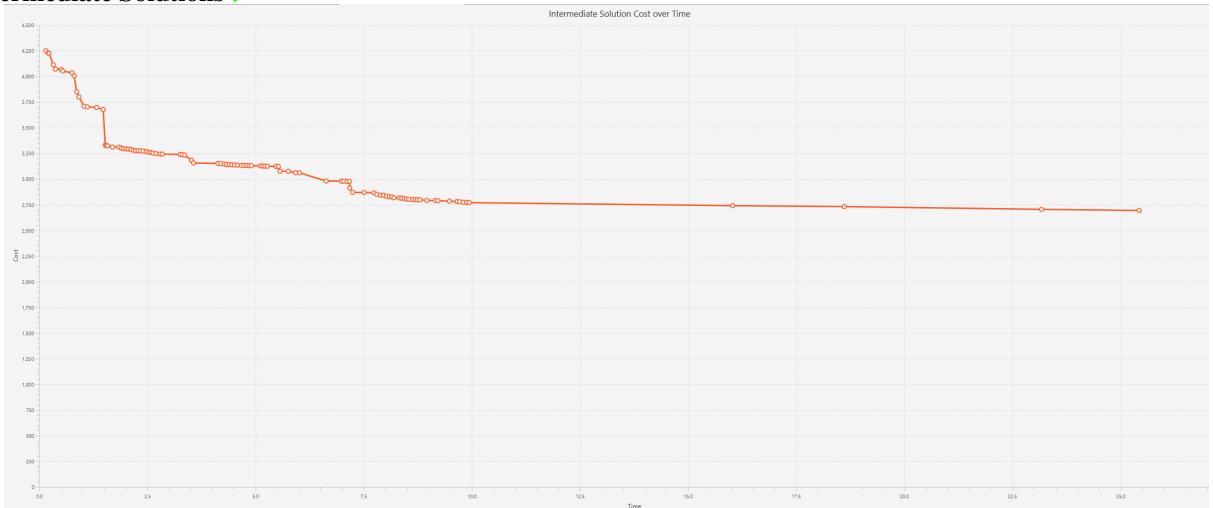


Resource Utilization ✓



- Shows utilization of machines as percentage of active time
- Helpful to identify bottleneck machines
- Information also shown in Machine Gantt

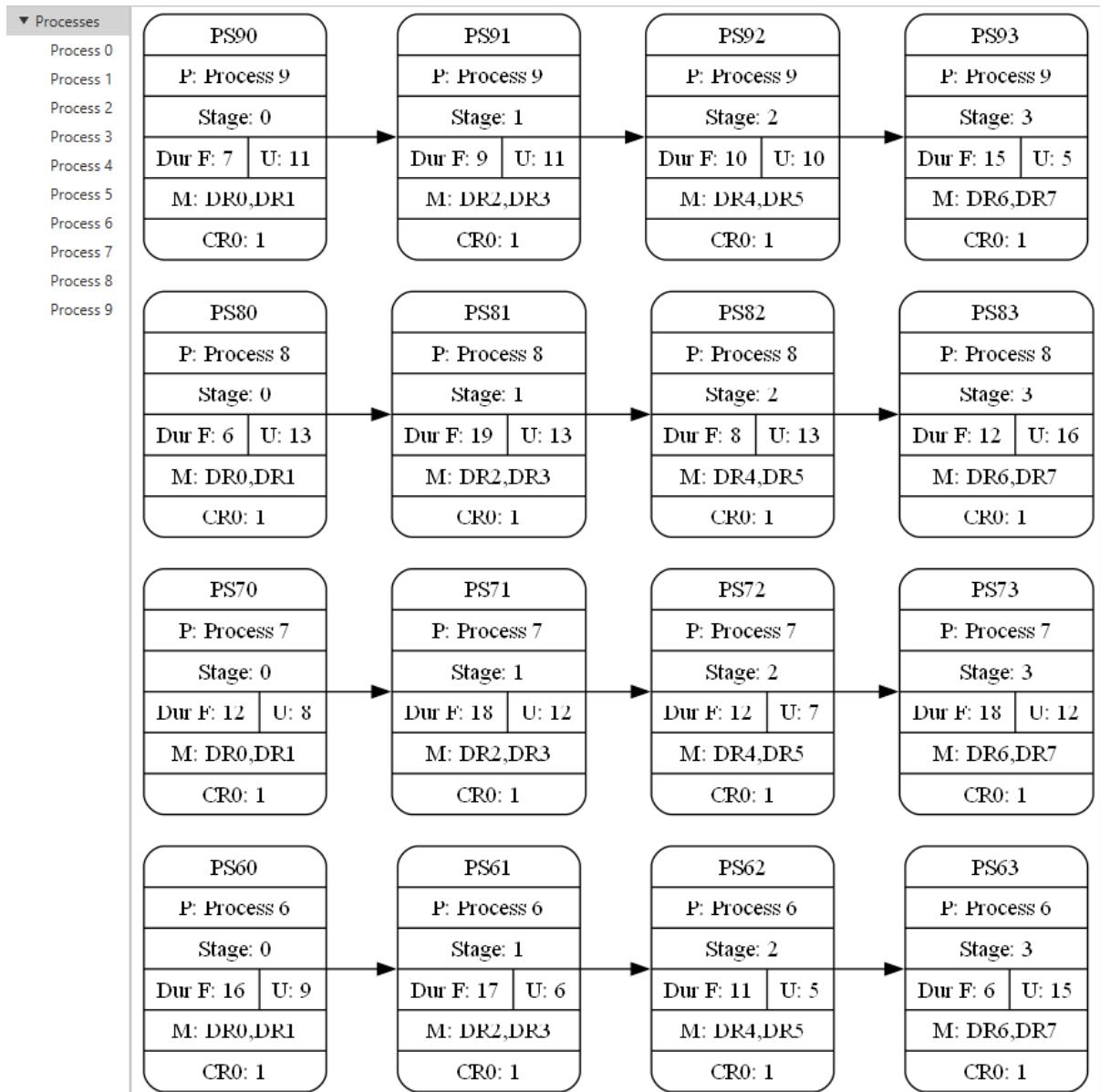
Intermediate Solutions ✓



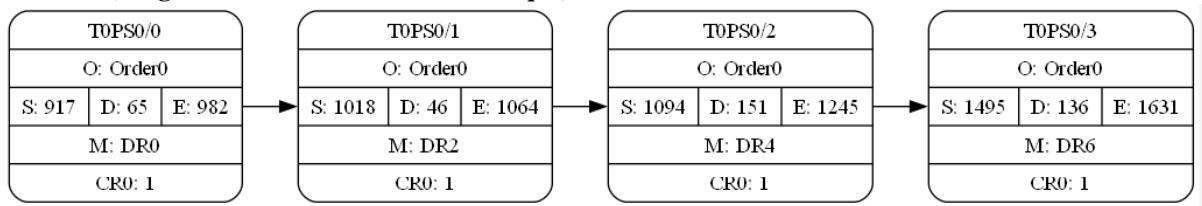
- Shows intermediate solutions found over time
- Useful to see if enough/too much time is allocated

Process Diagram ✓

- See all details of one process in one image
- Can also look at all processes in one diagram
- Options to show/hide different fields

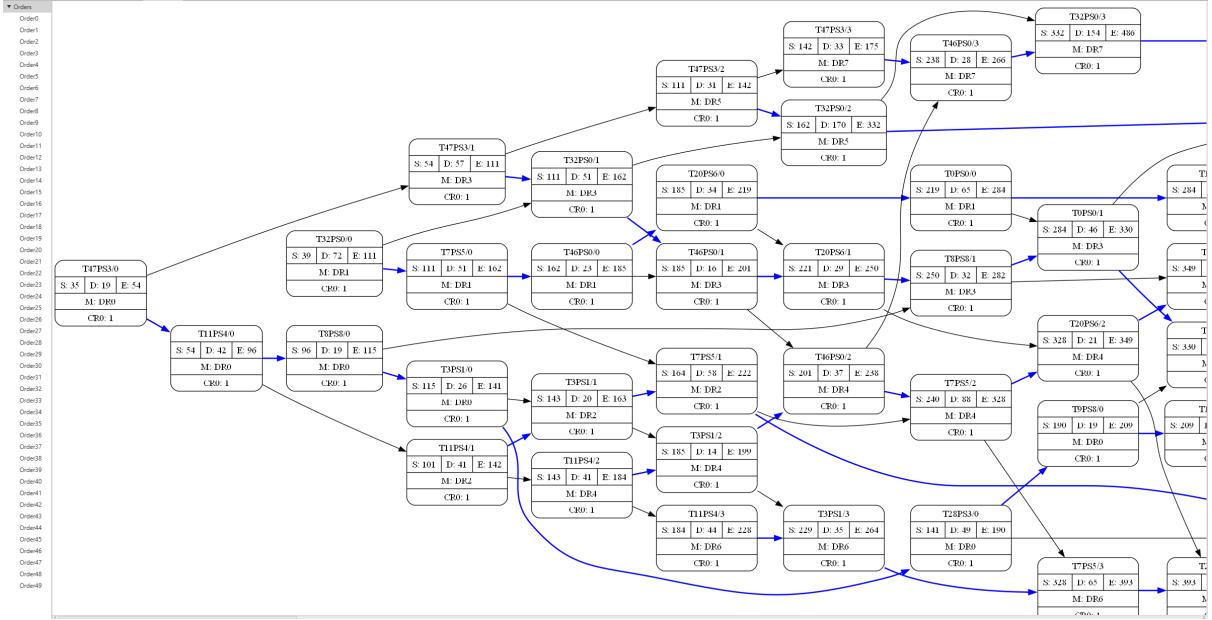


PERT Chart(Program Evaluation Review Technique) ✓



- Show details of job as a graph
- Useful if task graph is not a chain
- Often used in project management

PERT Charts become Confusing Quite Quickly



- Especially if all resource dependencies are included (in blue)

Calendars X

	Jan 22	Feb 22	Mar 22	Apr 22	May 22	Jun 22	Jul 22	Aug 22	Sep 22	Oct 22	Nov 22	Dec 22
Mon	3	3	3	3	3	3	3	3	3	3	3	3
Tue	3	3	3	3	3	3	3	3	3	3	3	3
Wed	3	3	3	3	3	3	3	3	3	3	3	3
Thu	3	3	3	3	3	3	3	3	3	3	3	3
Fri	3	3	3	3	3	3	3	3	3	3	3	3
Sat												
Sun												

- Shows weekly structure for one or more years
- Indicates public holidays, shut-downs, etc
- Indicating working days, KPI for each day

6 Summary

Summary

- We introduced the key concepts for scheduling problems
- Orders, products, processes
- Jobs and tasks
- Existing problem classifications
 - Academic
 - Limited practical usefulness
 - Used for benchmarking
- Key visualization ideas