Concepts

Helmut Simonis

Constraint Based Production Scheduling

Licence

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

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?

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

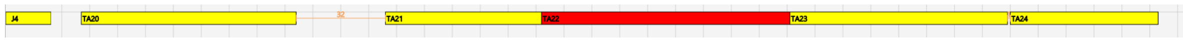
- Each *process* consists of one or more *process steps*
- A process step contains a duration formula to describe how long it lasts
- The order of *process steps* is defined by *process sequences*
- The resources needed are defined by *resource needs* (described later on)
- Tasks are created for each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

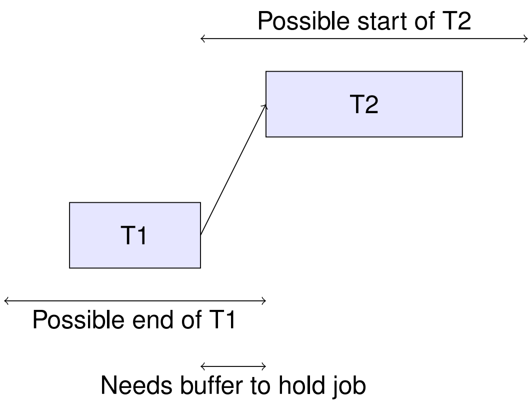
Temporal Relations

- 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



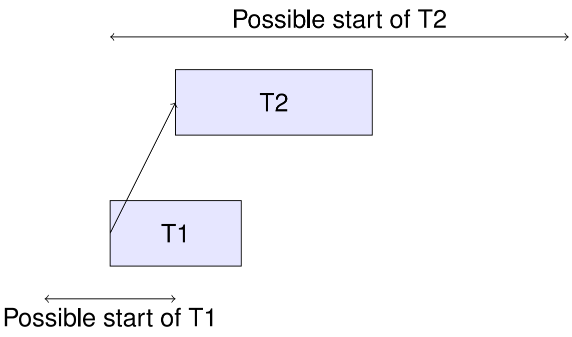
The Most Common Relation: EndBeforeStart

- 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
  - For example cooling, drying time outside a machine



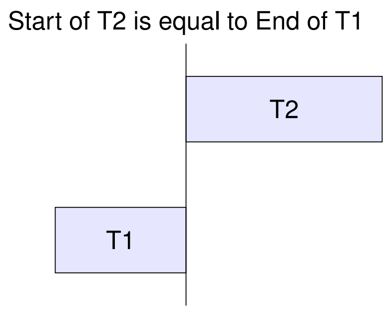
Less Common: StartBeforeStart

- 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



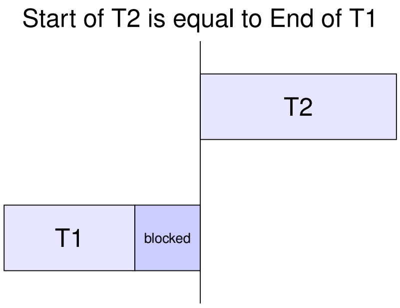
NoWait

- 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



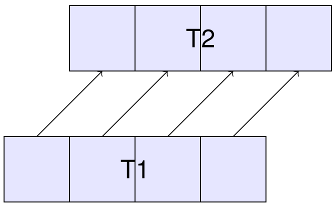
Blocking

- 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*



Special Case: Pipelining

- Sometimes, we can start on the next task while the first is still running
- Possible if one jobs consists of 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

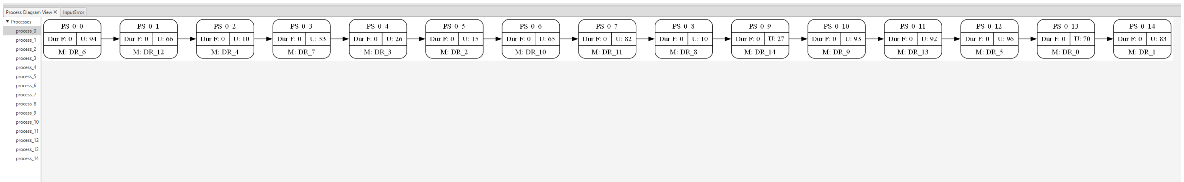
Relation	Illustration	Interpretation
$X < Y$ $Y > X$	X Y	X precedes Y Y is preceded by X
$X \mathbf{m} Y$ $Y \mathbf{mi} X$	X Y	X meets Y Y is met by X (*i* stands for *inverse*)
$X \mathbf{o} Y$ $Y \mathbf{oi} X$	X Y	X overlaps with Y Y is overlapped by X
$X \mathbf{s} Y$ $Y \mathbf{si} X$	X Y	X starts Y Y is started by X
$X \mathbf{d} Y$ $Y \mathbf{di} X$	X Y	X during Y Y contains X
$X \mathbf{f} Y$ $Y \mathbf{fi} X$	X Y	X finishes Y Y is finished by X
$X = Y$	X Y	X is equal to Y

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

Start and End of Jobs

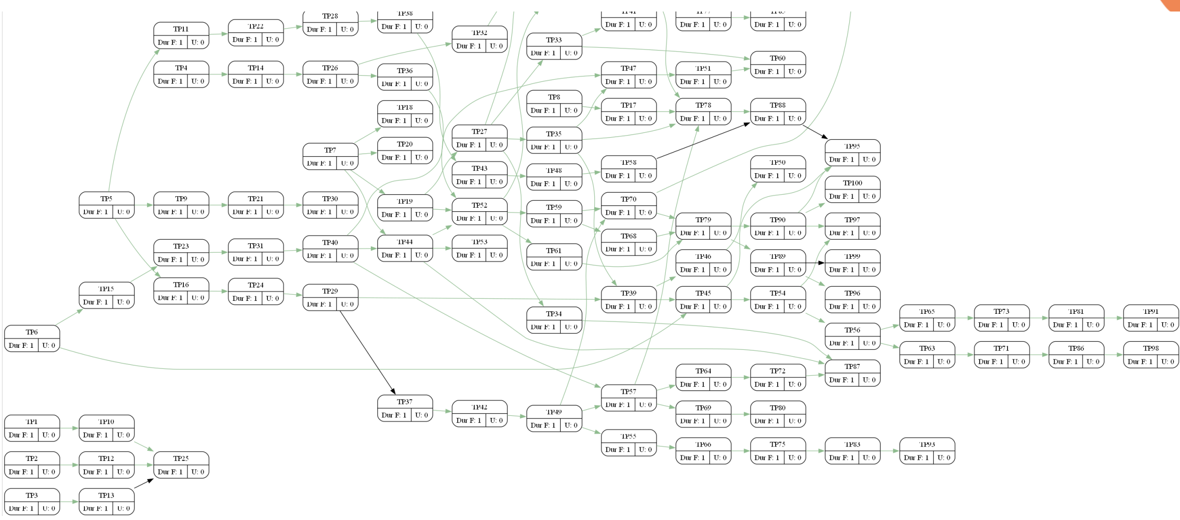
- The start of a job is equal to the start of the earliest task of the job
- The end of a job is equal to the latest end of any of its tasks
- Also called: the job *spans* its tasks
- Sometimes very simple
  - Start of job is start of first process step
  - End of job is end of last process step
  - 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

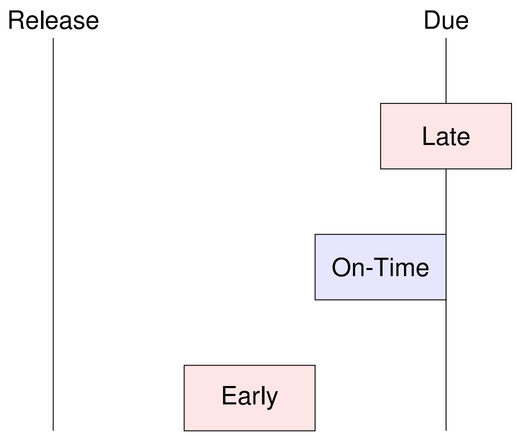
An Example of a More Complex Process



- There is no clear first or last process step

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



Relations between Jobs

- 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

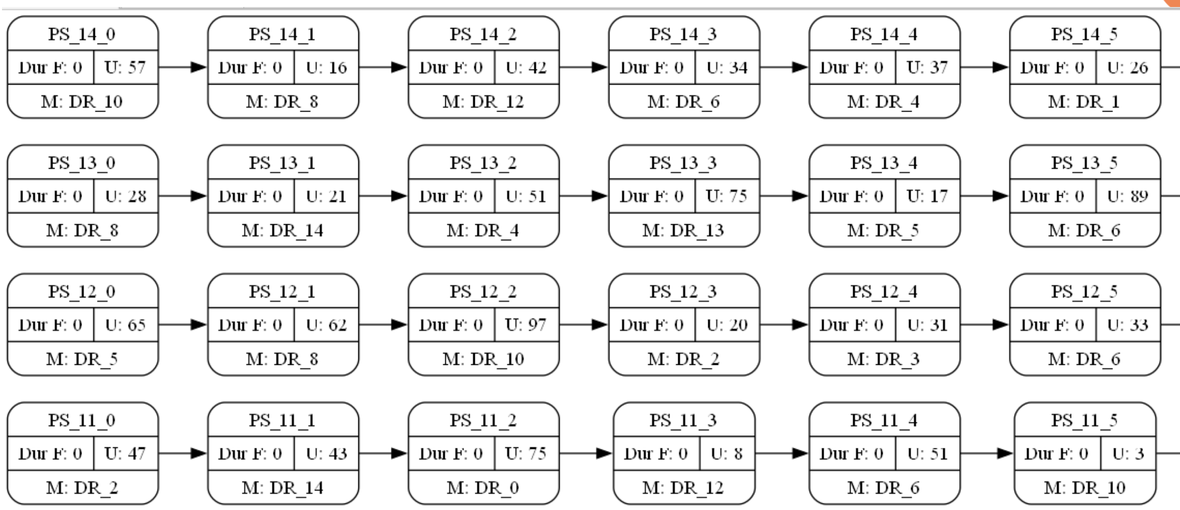
Problem Classification

- 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

Job-Shop

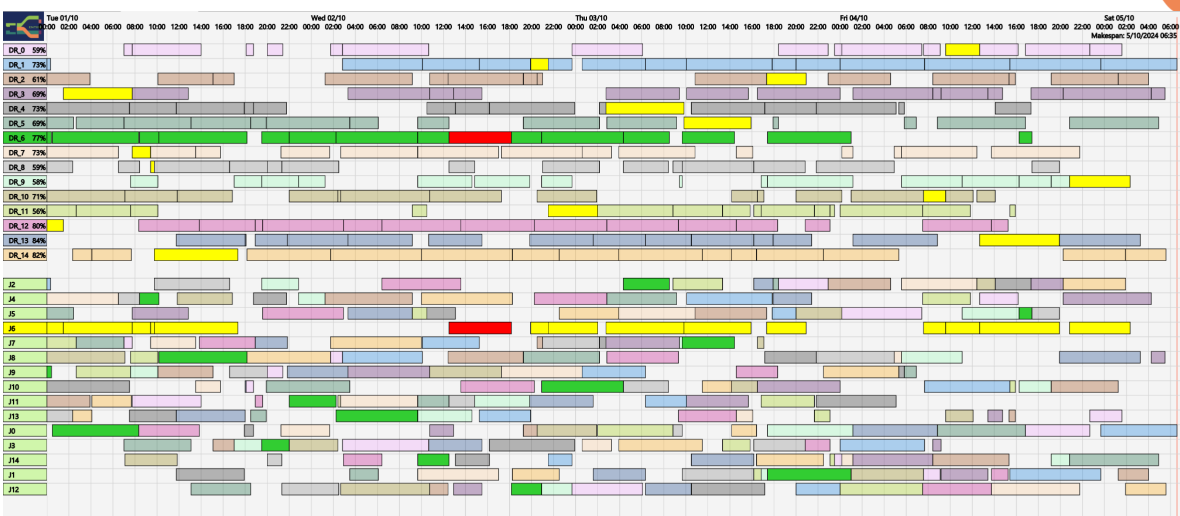
- Consists of a number of jobs and a number of machines
- Each job visits each machine, but possibly in a different order, depending on process
- Tasks of a job are linked as a precedence chain
- Objective is to minimize overall end, the *makespan*

Example Job-Shop Process



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

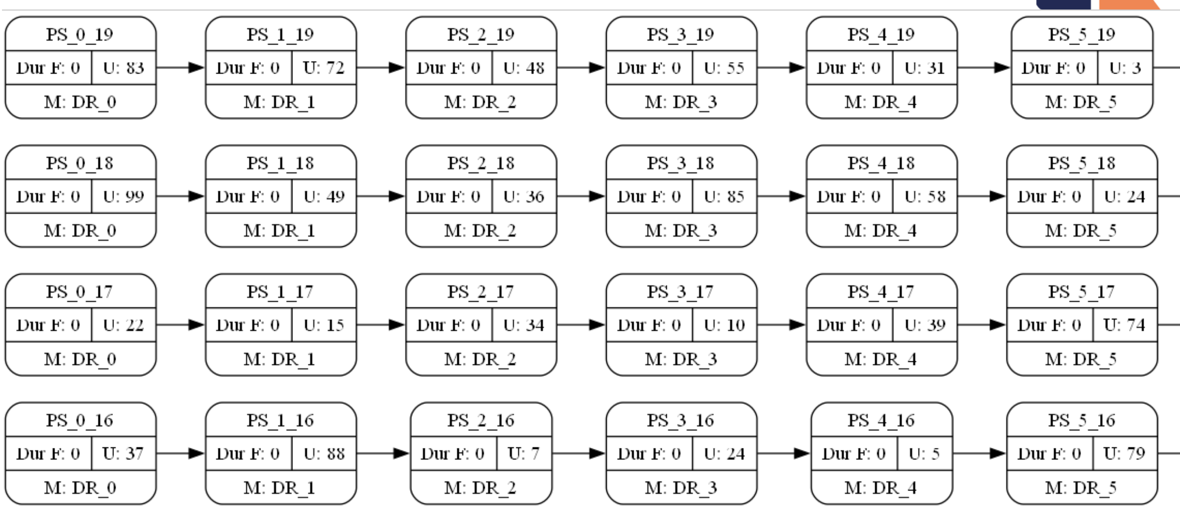
Example Job-Shop Solution



- One task is selected (in red), in both Machine and Job Gantt Chart

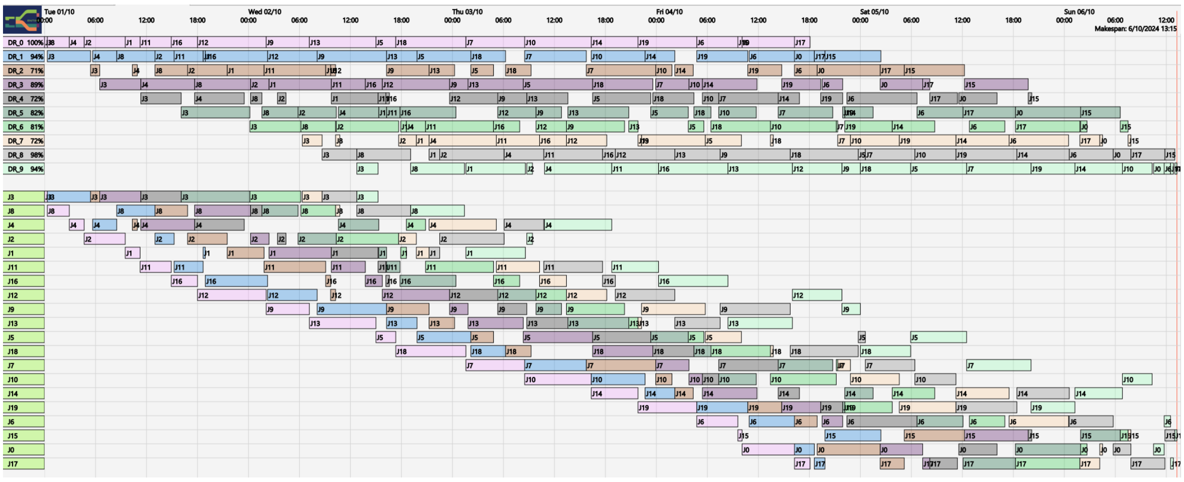
Flow-Shop

- Consists of a number of jobs and a number of machines
- Each jobs visits each machine, all jobs in the same order
- Tasks of a job a 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

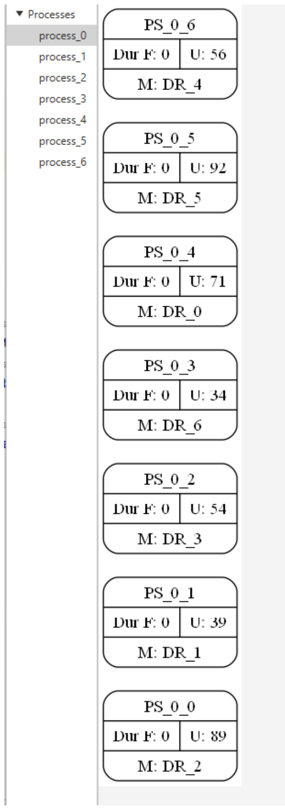


- Tasks are colored by machine, note the regular pattern in the Job Gantt Chart

Open-Shop

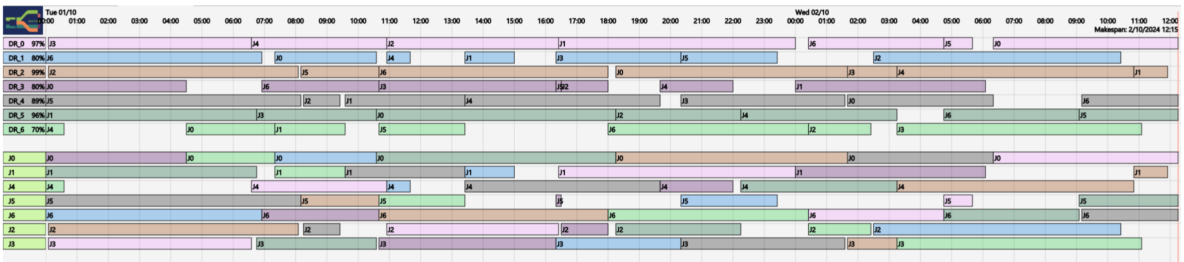
- Consists of a number of jobs and a number of machines
- Each jobs visits each machine, we have to choose the sequence individually for each order
- There are no temporal constraints between tasks, but tasks of the same job cannot overlap
- Objective is to minimize overall end, the *makespan*

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

Open-Shop Example Solution



- Example solution for 7x7 open shop example
- Order of tasks within jobs not constrained

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

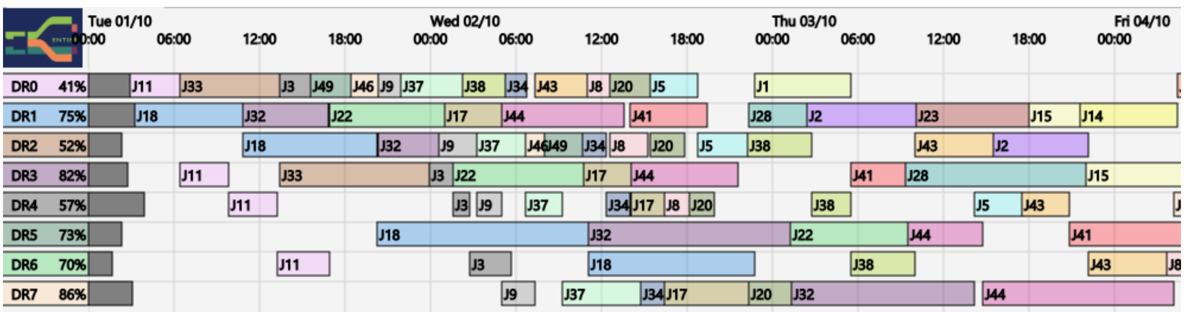
α, β, γ Notation

- The previous classes are good for research, but not very practical
- General scheme to describe problem type

Visualization

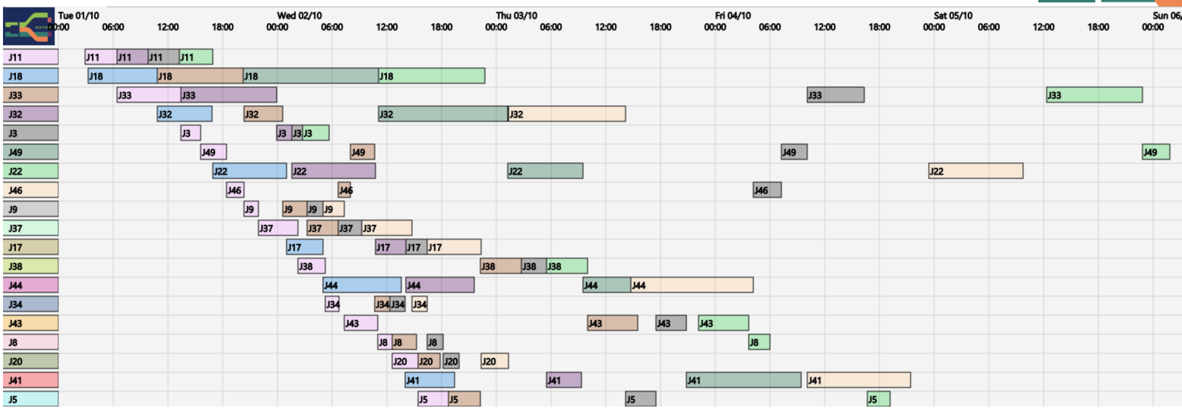
- Visualization is key to present and to understand results
- Many different ways to give an overview of schedule, and highlight problems
- Some diagrams types are used a lot, and are provided in our generic scheduling tool
- Customization is key

Machine Gantt Chart



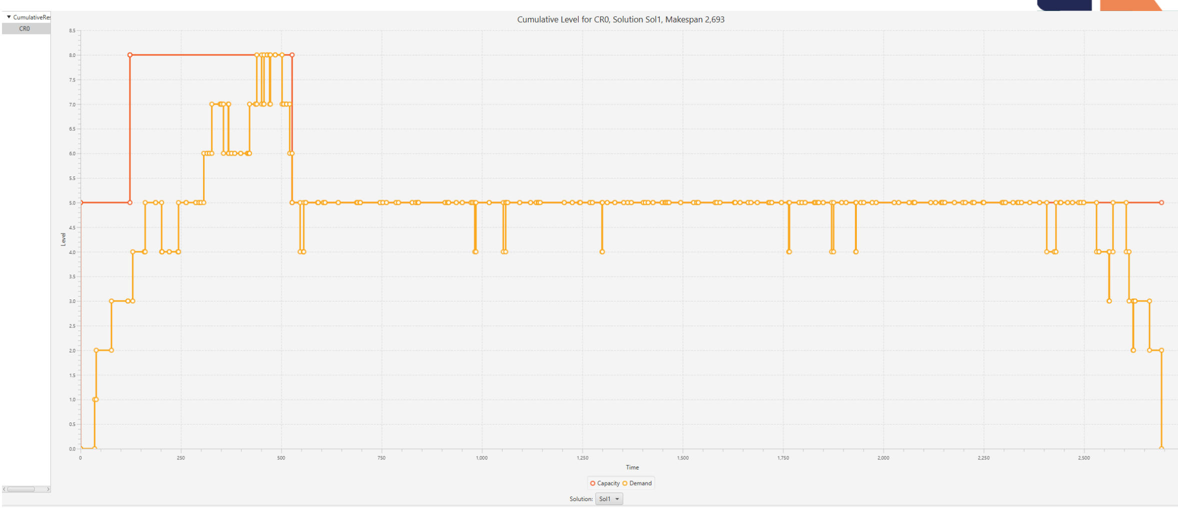
- Shows all tasks that are assigned to each machine
- Tasks should not overlap
- Also shows work in progress (WiP), down-times
- Optional display of setup and idle times

Job Gantt Chart



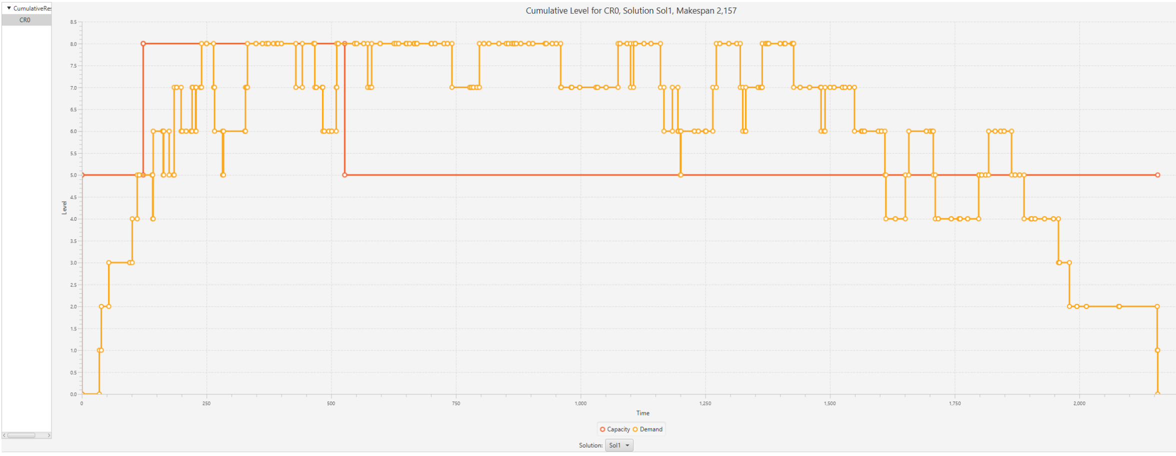
- Shows all tasks of a job in one line
- Only works for single chain of process steps
- Possible display of earliness, lateness
- Optional display of waiting and transport times

Cumulative Resource Chart

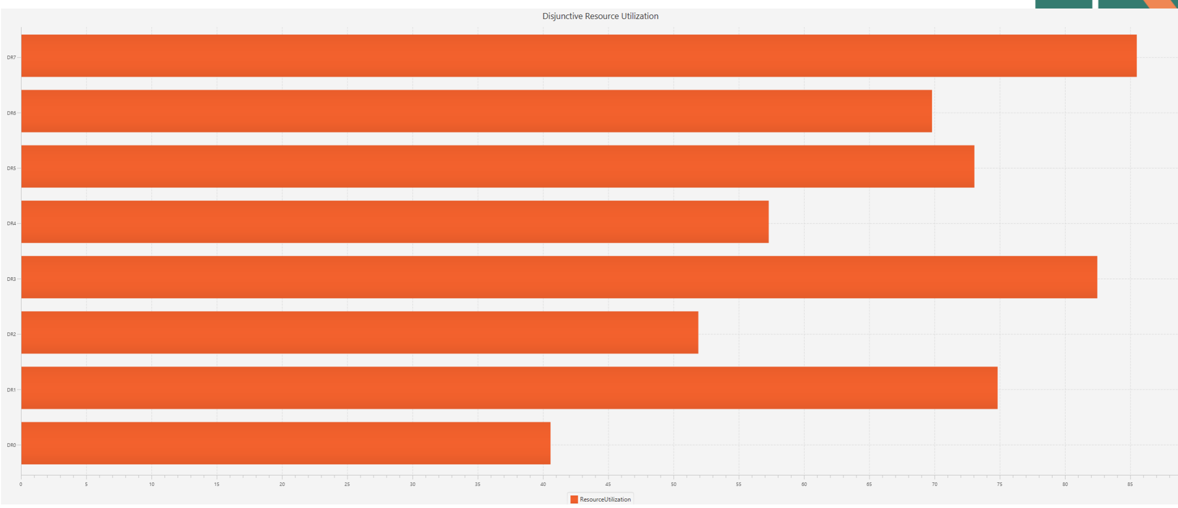


- Shows resource utilization of cumulative resource over time
- Utilization should be below capacity profile
- Unless we relax the cumulative resource constraint

Cumulative Resource Constraint Relaxed

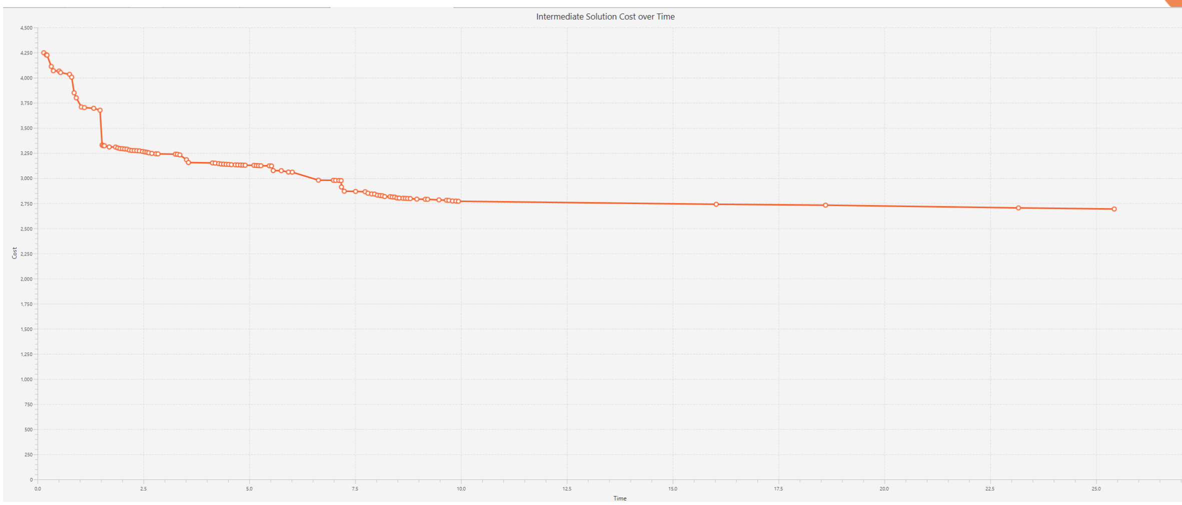


Resource Utilization



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

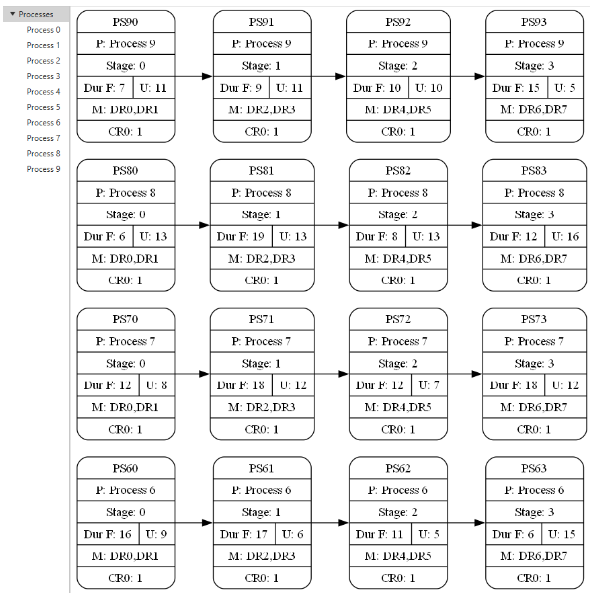
Intermediate Solutions



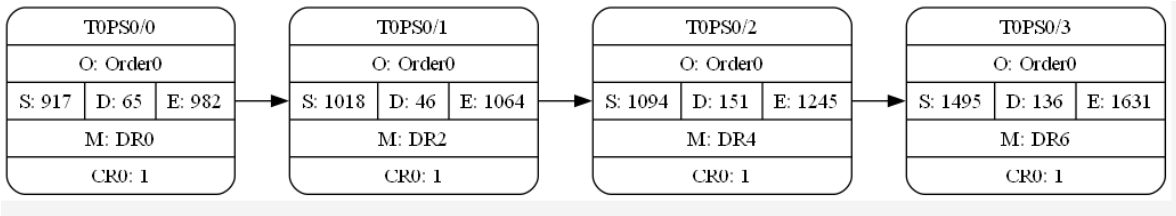
- Shows intermediate solutions found over time
- Useful to see if enough/too much time is allocated

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

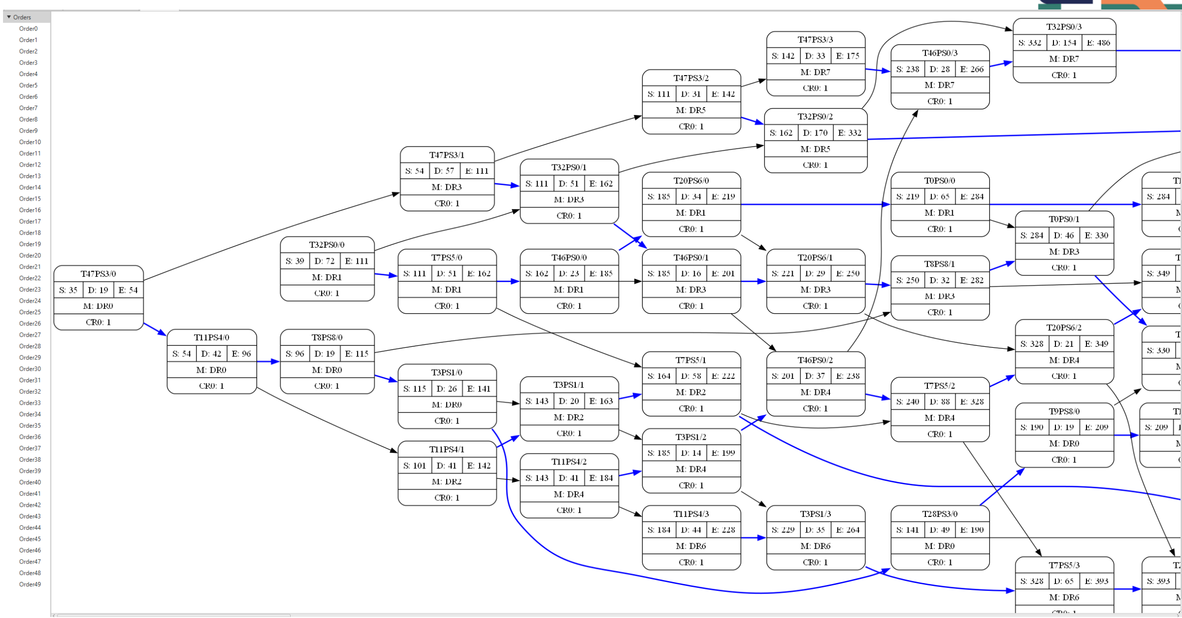


PERT Chart(Program Evaluation Review Technique)



- 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)

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