Skills Development Program Scheduling

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.

Note

- This is a document which combines all materials from the Scheduling course
- Files as also available individually in separate directories

Insight is one of the largest data research and innovation centres in Europe...

4 Co-Lead Universities 9 partner institutions	Built on **20** years of research in Data Analytics and AI
450+ Academics, Postdocs, PhDs, RAs	**3400+** Scientific conference and journal papers
175+ Funded collaborations with industry partners	**350+** Research Awards
16 Spin out companies 72 license agreements	**135+** H2020 consortia, 500+ collaborations, 40+ countries
1,137+ school visits, 28,000 students	**276** PhDs graduated

Background

- Mathematics @ TH Darmstadt
- 1986-1990 ECRC GmbH, Munich
- 1990-2000, Technical Director, Cosytec SA, Orsay
- 2000-2005, Imperial College London, Parc Technologies Ltd
- 2013-2014, President, Association for Constraint Programming
- Best Application Paper Awards, CP 2009, CP 2013
- Program Chair, CP 2020, CPAIOR 2014
- Distinguished Service Award, ACP

Part I

Introduction

Key Points

- Introducing a running example
- AI is more than LLM
- Stochastic vs. deductive AI methods
- Constraint Based Scheduling and its alternatives
- Key advantages
  - Compositional
  - Reusable
  - Explainable
- Course structure

Developing a Generic Scheduling Tool

- No programming, configured by JSON input data
- Compositional use of different constraint types
- Different commercial or open-source back-end solvers
- Developed in Java
- Interactive JavaFX front-end
- Can by used as back-end scheduling tool/server
- Instance generator included
- Readers for multiple benchmark types included
- Release planned early 2025
- Preview during the course, hands-on experience this afternoon

Introducing a Simple Scheduling Problem

- Will be used throughout the program
- Generated by instance generator
- 50 orders for different products, release and due dates
- 4 stages, always performed in the same sequence
- Two identical machines available for each stage
- Cumulative manpower constraint
- Complete description as JSON document

Excerpt of JSON Description

```
"order": [
  {
    "product": "Prod0",
    "process": "Process 0",
    "due": 5449,
    "releaseDate": "1/10/2024 00:00",
    "release": 0,
    "qty": 7,
    "dueDate": "19/10/2024 22:05",
    "name": "Order0",
    "earlinessWeight": 1,
    "latenessWeight": 1
  },
```

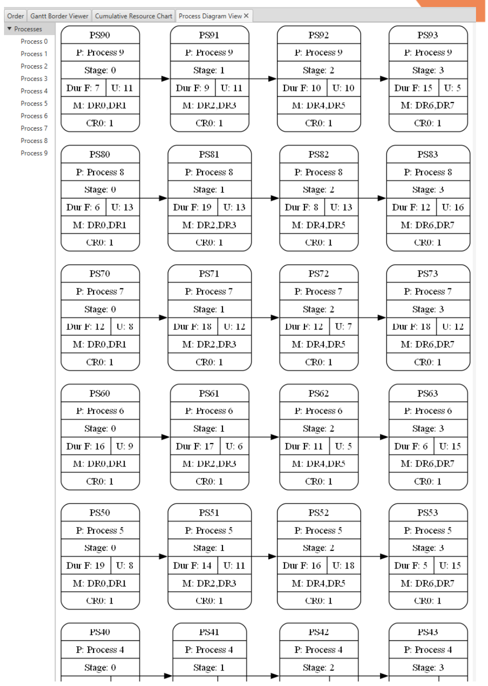

Orders Loaded

Order ✕

Name	Nr	Product	Process	Qty	Due	DueDate	Release	ReleaseDate	LatenessWeight	EarlinessWeight
Order0	0	Prod0	Process 0	7	5,449	19/10/2024 22:05	0	1/10/2024 00:00	1.0	1.0
Order1	1	Prod1	Process 1	6	2,134	8/10/2024 09:50	0	1/10/2024 00:00	1.0	1.0
Order2	2	Prod1	Process 1	7	1,266	5/10/2024 09:30	0	1/10/2024 00:00	1.0	1.0
Order3	3	Prod1	Process 1	1	1,976	7/10/2024 20:40	0	1/10/2024 00:00	1.0	1.0
Order4	4	Prod9	Process 9	5	2,866	10/10/2024 22:50	0	1/10/2024 00:00	1.0	1.0
Order5	5	Prod9	Process 9	3	3,339	12/10/2024 14:15	0	1/10/2024 00:00	1.0	1.0
Order6	6	Prod4	Process 4	9	1,676	6/10/2024 19:40	0	1/10/2024 00:00	1.0	1.0
Order7	7	Prod5	Process 5	4	5,471	19/10/2024 23:55	0	1/10/2024 00:00	1.0	1.0
Order8	8	Prod8	Process 8	1	1,966	7/10/2024 19:50	0	1/10/2024 00:00	1.0	1.0
Order9	9	Prod8	Process 8	1	4,279	15/10/2024 20:35	0	1/10/2024 00:00	1.0	1.0
Order10	10	Prod9	Process 9	6	5,733	20/10/2024 21:45	0	1/10/2024 00:00	1.0	1.0
Order11	11	Prod4	Process 4	4	3,088	11/10/2024 17:20	0	1/10/2024 00:00	1.0	1.0
Order12	12	Prod8	Process 8	9	2,569	9/10/2024 22:05	0	1/10/2024 00:00	1.0	1.0
Order13	13	Prod7	Process 7	4	2,331	9/10/2024 02:15	0	1/10/2024 00:00	1.0	1.0
Order14	14	Prod4	Process 4	9	3,290	12/10/2024 10:10	0	1/10/2024 00:00	1.0	1.0
Order15	15	Prod3	Process 3	6	1,968	7/10/2024 20:00	0	1/10/2024 00:00	1.0	1.0
Order16	16	Prod4	Process 4	8	1,579	6/10/2024 11:35	0	1/10/2024 00:00	1.0	1.0
Order17	17	Prod1	Process 1	3	4,263	15/10/2024 19:15	0	1/10/2024 00:00	1.0	1.0
Order18	18	Prod5	Process 5	9	4,491	16/10/2024 14:15	0	1/10/2024 00:00	1.0	1.0
Order19	19	Prod3	Process 3	4	613	3/10/2024 03:05	0	1/10/2024 00:00	1.0	1.0
Order20	20	Prod6	Process 6	2	5,034	18/10/2024 11:30	0	1/10/2024 00:00	1.0	1.0
Order21	21	Prod7	Process 7	4	1,797	7/10/2024 05:45	0	1/10/2024 00:00	1.0	1.0
Order22	22	Prod8	Process 8	7	4,286	15/10/2024 21:10	0	1/10/2024 00:00	1.0	1.0
Order23	23	Prod9	Process 9	8	1,970	7/10/2024 20:10	0	1/10/2024 00:00	1.0	1.0
Order24	24	Prod3	Process 3	4	1,286	5/10/2024 11:10	0	1/10/2024 00:00	1.0	1.0
Order25	25	Prod6	Process 6	6	4,170	15/10/2024 11:30	0	1/10/2024 00:00	1.0	1.0
Order26	26	Prod8	Process 8	4	5,481	20/10/2024 00:45	0	1/10/2024 00:00	1.0	1.0
Order27	27	Prod1	Process 1	4	3,255	12/10/2024 07:15	0	1/10/2024 00:00	1.0	1.0
Order28	28	Prod3	Process 3	7	1,021	4/10/2024 13:05	0	1/10/2024 00:00	1.0	1.0
Order29	29	Prod5	Process 5	4	5,315	19/10/2024 10:55	0	1/10/2024 00:00	1.0	1.0
Order30	30	Prod9	Process 9	7	5,075	18/10/2024 14:55	0	1/10/2024 00:00	1.0	1.0
Order31	31	Prod1	Process 1	6	3,089	11/10/2024 17:25	0	1/10/2024 00:00	1.0	1.0
Order32	32	Prod0	Process 0	8	3,324	12/10/2024 13:00	0	1/10/2024 00:00	1.0	1.0
Order33	33	Prod7	Process 7	9	607	3/10/2024 02:35	0	1/10/2024 00:00	1.0	1.0
Order34	34	Prod9	Process 9	1	2,914	11/10/2024 02:50	0	1/10/2024 00:00	1.0	1.0

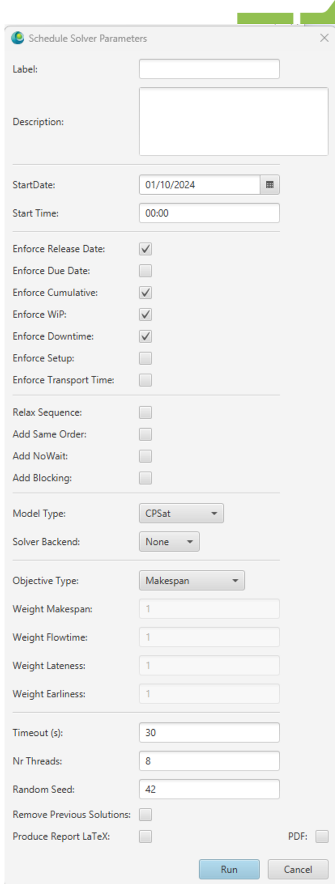
Process Diagram

- Processes describe how products are made
- Multiple process steps
- Not always in a straight sequence
- Duration formula based on quantity made
- Temporal constraints between steps
- Possible machines to run on
- Resource requirements (manpower, electricity,...)

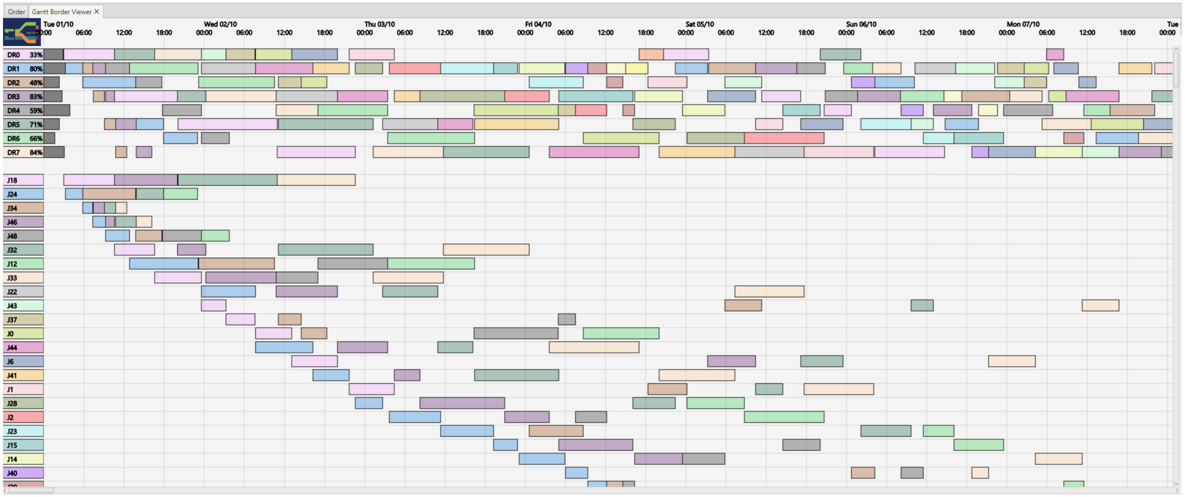


Selecting Solver Options

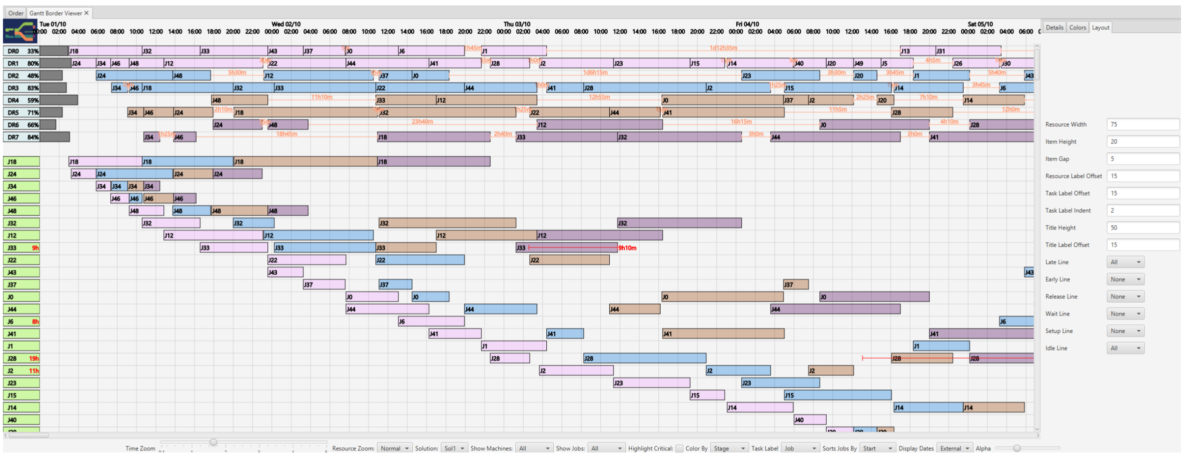
- Which constraints to enforce
  - Here: do not enforce due dates
- Additional constraints to try
- Why solver to run
  - Here: Use open-source CPSat solver
- Which objective to use
  - Here: Makespan, overall project end
- What resources to use
  - Allow 30 seconds
  - Use 8 parallel threads



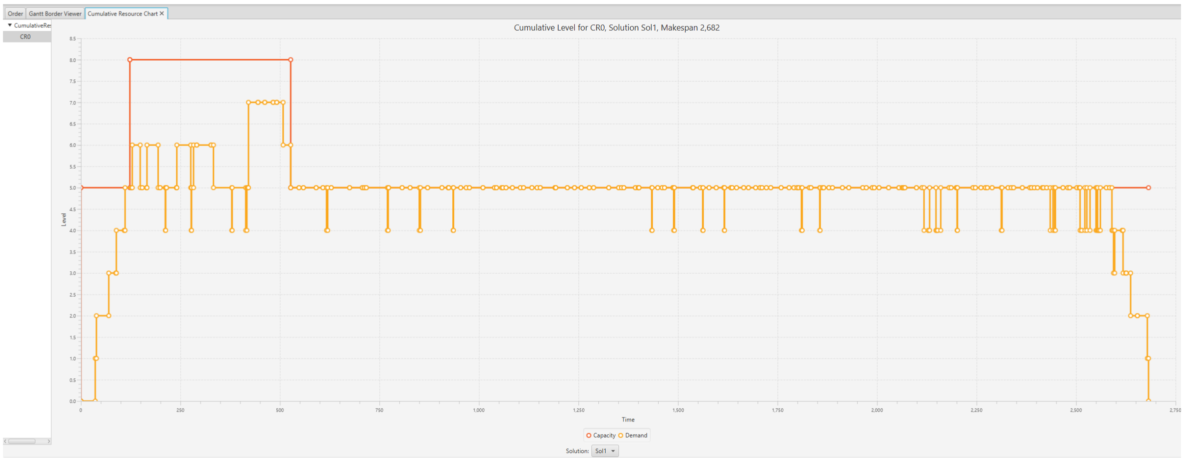
Schedule - Initial Gantt Chart



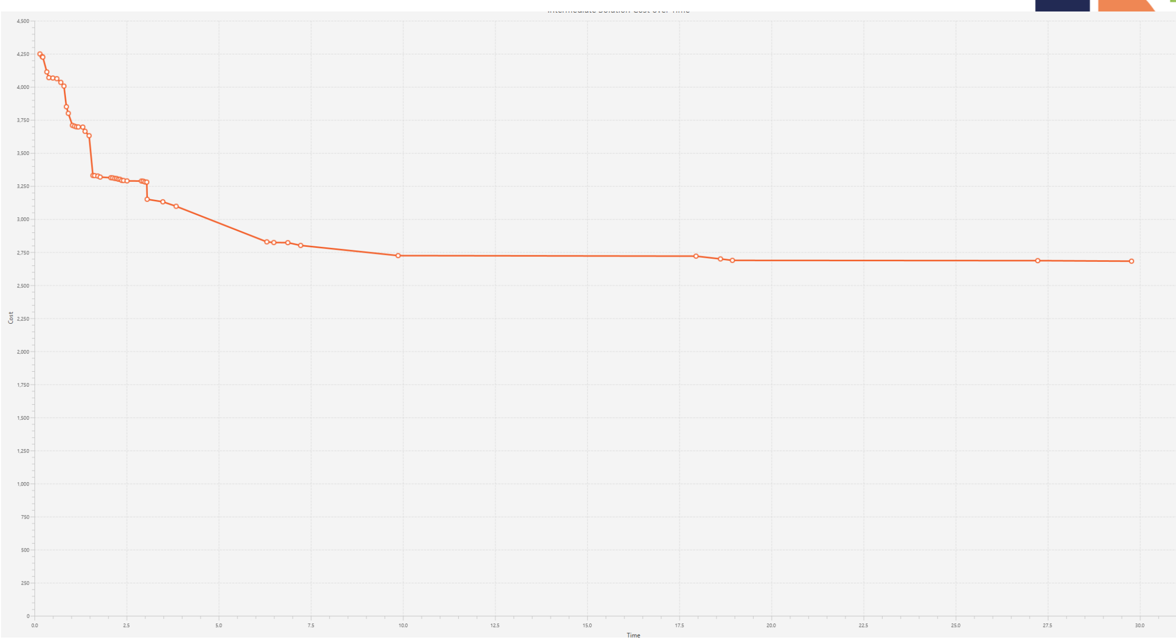
Adapted Gantt Chart



Cumulative Resource Chart



Intermediate Solutions Found



- Ongoing search for improved solutions
- Depends on time and resources, solver used

What is Artificial Intelligence?

> *Artificial intelligence, or AI, is the field that studies the synthesis and analysis of computational agents that act intelligently.*

David Poole, Alan Mackworth. Artificial Intelligence, Cambridge University Press, 3rd Edition, 2023.

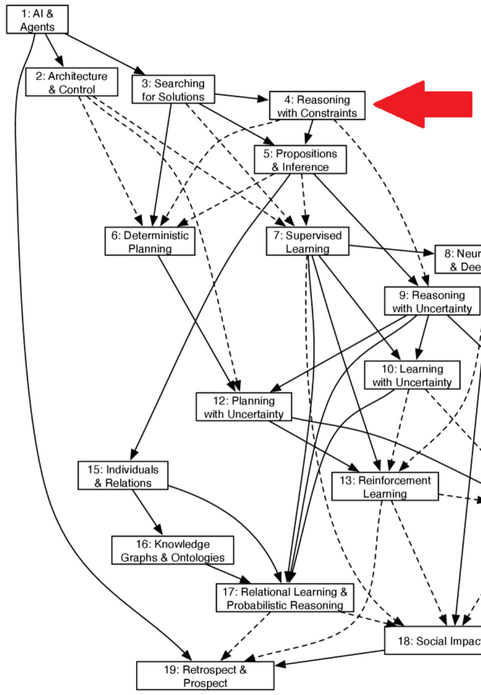
- This definition leaves a lot of questions.

The Great Divide

- Two fundamentally different approaches to AI
  - Reasoning based
  - Stochastic
- Currently, the stochastic methods get all the attention
- But they have their problems
  - Impossible to understand what is happening inside
  - Hallucinations, making up convincing false statements
  - Enormous resource requirements
  - Privacy/IP of training data
  - Really limited to a few multi-nationals

Topics in AI

- Chapter Structure of AI Book
- Shows importance of deductive/search based approaches



What is Constraint Programming?

Constraint programming technology is used to find solutions to scheduling and combinatorial optimization problems. It is based primarily on computer science fundamentals, such as logic programming and graph theory, in contrast to mathematical programming, which is based on numerical linear algebra.
Constraint programming is invaluable when dealing with the complexity of many real-world sequencing and scheduling problems.

IBM (https://ibmdecisionoptimization.github.io/docplex-doc/cp.html)

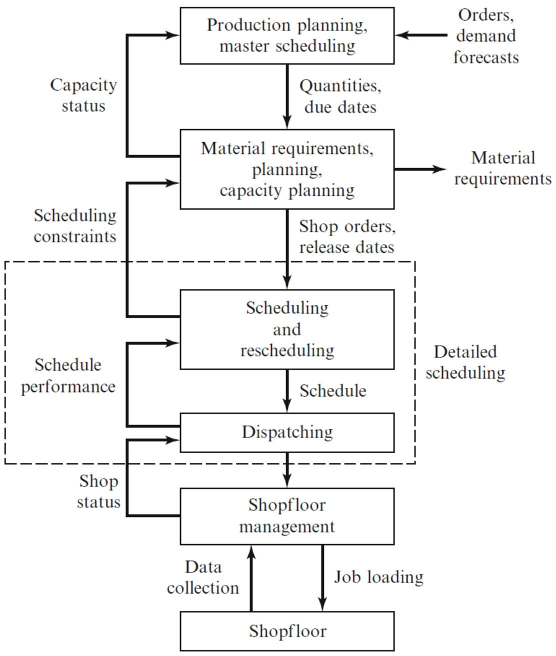
What is Scheduling?

Scheduling is a decision-making process that is used on a regular basis in many manufacturing and services industries. It deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives.

Michael Pinedo. Scheduling. Springer, 5th edition, 2016.

Information Flow Diagram in a Manufacturing System

- According to Pinedo, page 5.
- We focus on what is shown as *detailed scheduling*



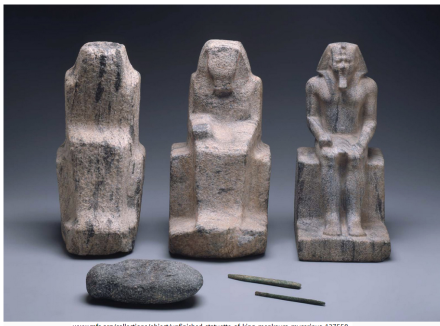
Constraint Programming - in a nutshell

- Declarative description of problems with
  - *Variables* which range over (finite) sets of values
  - *Constraints* over subsets of variables which restrict possible value combinations
  - A *solution* is a value assignment which satisfies all constraints
- Constraint propagation/reasoning
  - Removing inconsistent values for variables
  - Detect failure if constraint can not be satisfied
  - Interaction of constraints via shared variables
  - Incomplete
- Search
  - User controlled assignment of values to variables
  - Each step triggers constraint propagation
- Different domains require/allow different methods

Constraint Programming is Different

- Declarative Programming
  - Concentrate on what you want
  - Not how to get there
  - Program != Algorithm
  - Program = Model
- Applied to Combinatorial Problems
  - No complete polynomial algorithms known (exist?)
  - CP less ad-hoc than heuristics
  - Models can evolve

A Subtractive Process



www.mfa.org/collections/object/unfinished-statuette-of-king-menkaura-mycerinus-137558

“Oh, bosh, as Mr. Ruskin says. Sculpture, per se, is the simplest thing in the world. All you have to do is to take a big chunk of marble and a hammer and chisel, make up your mind what you are about to create and chip off all the marble you don’t want.”-Paris Gaulois.

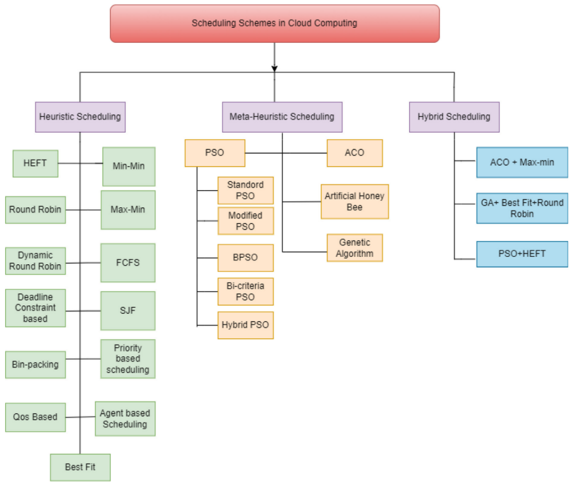
Source: `https://quoteinvestigator.com/2014/06/22/chip-away/`

Other Technologies

- Heuristics
- Integer Programming
- Local search
- Deep neural networks

Heuristics

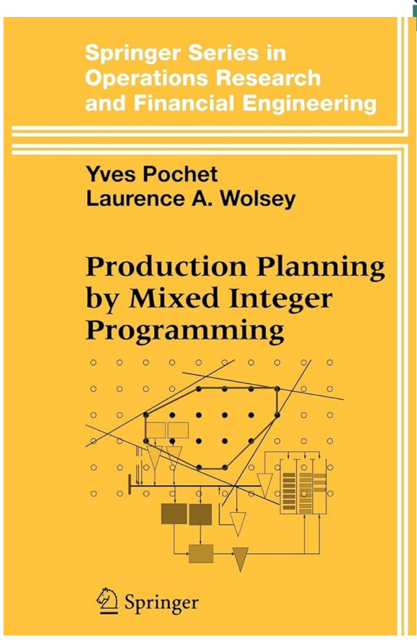
- Do not try to explore the search space
- Find a good enough solution by making greedy choices
- More general meta-heuristics schemes
- Very good heuristics exist for specific problem types
- Not compositional, added constraints may destroy existing approach
- Often not reusable code base



From: Singh, Kumar, and Singh: An empirical investigation of task scheduling and VM consolidation schemes in cloud environment, Computer Science review, 2023, `https://www.sciencedirect.com/science/article/pii/S1574013723000503`

Integer Programming

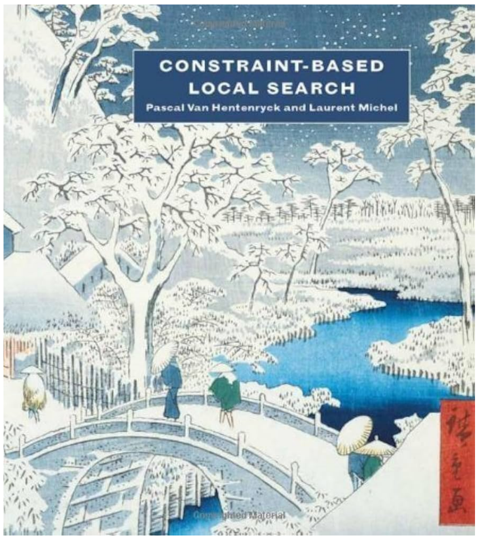
- Restrict yourself to linear constraints
- Powerful reasoning on the complete set of constraints
  - Linear Programming
  - Cut generation
- Expressing scheduling constraints can be difficult
- Scalability issues for detailed scheduling



`https://link.springer.com/book/10.1007/0-387-33477-7`

Local Search

- Start with an initial solution
- Try out changes that maintain feasibility
- Gradual improvement over time
- Not compositional
- No guarantee of solution quality
- Unifying approach: Constraint-Based Local Search



https://mitpress.mit.edu/9780262220774/constraint-based-local-search/

Course Structure

Time	Day 1	Day 2
09:00-10:30	Introduction & Motivation	Costs & Objective Functions
10:30-11:00	Coffee	Coffee
11:00-12:30	Scheduling Concepts	Advanced Concepts
12:30-14:00	Lunch	Lunch
14:00-15:30	Machine Constraints	Case Studies
15:30-16:00	Coffee	Coffee & Close
16:00-17:00	Experiments	-

What is not covered?

- How does it all work?
- How to integrate into an existing IT environment
- How to define and solve new constraints
- Interactive solving techniques

How does it all work?

- You don't really need to know this to use Constraint Programming
- Advantage of declarative, compositional formulation
- I teach an introductory course on Constraint Programming for CRT-AI
  - Ask for details if interested
- Overview of courses, books and materials at `https://arxiv.org/abs/2403.12717`

Summary

- Why use Constraint Based Scheduling?
- Compared to other AI methods
- Compared to other solution approaches

Part II

Concepts

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

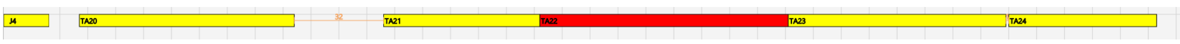
- Each *process* consists of one or more *process steps*
- A process step contains a duration formula to describe how long it lasts
- The order of *process steps* is defined by *process sequences*
- The resources needed are defined by *resource needs* (described later on)
- Tasks are created for each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

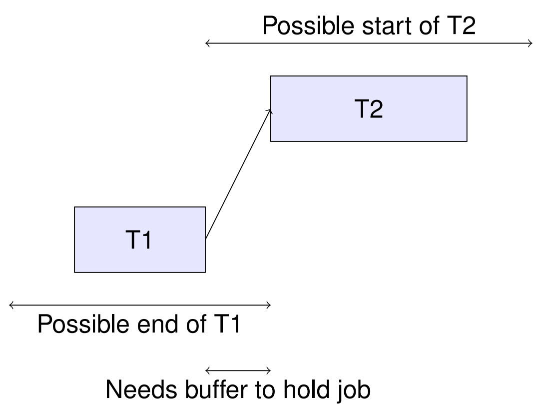


Annotations on Features

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

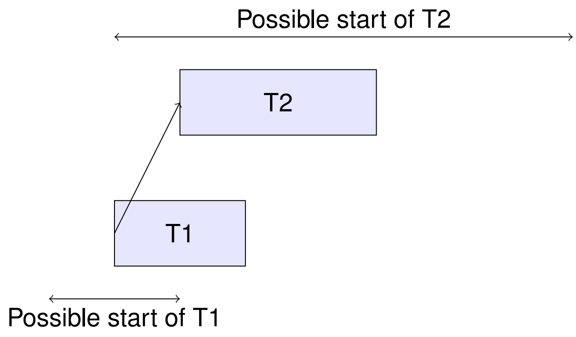
The Most Common Relation: EndBeforeStart ✓

- 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



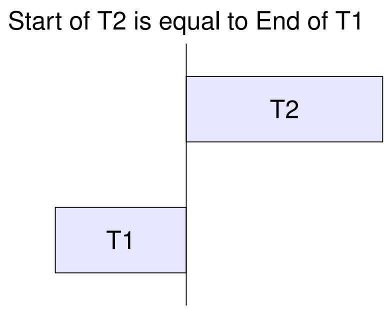
Less Common: StartBeforeStart ✓

- 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



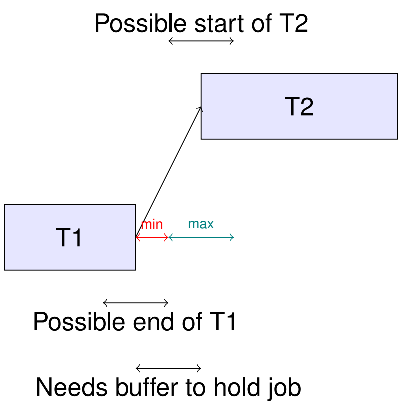
NoWait ✓

- 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



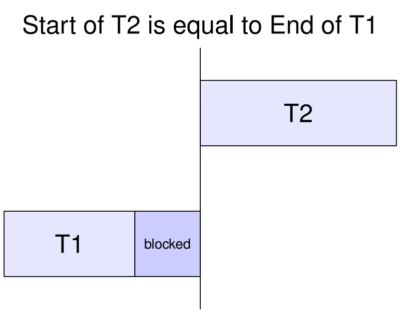
MaxWait (✓)

- 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



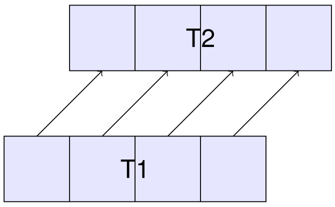
Blocking ✓

- 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*



Special Case: Pipelining ✗

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

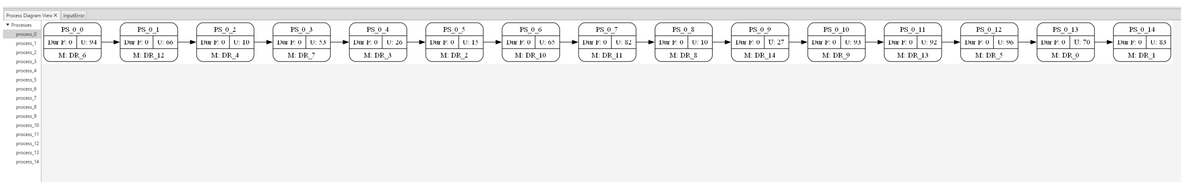
Relation	Illustration	Interpretation
$X < Y$ $Y > X$	X Y	X precedes Y Y is preceded by X
$X \mathbf{m} Y$ $Y \mathbf{mi} X$	X Y	X meets Y Y is met by X (*i* stands for *inverse*)
$X \mathbf{o} Y$ $Y \mathbf{oi} X$	X Y	X overlaps with Y Y is overlapped by X
$X \mathbf{s} Y$ $Y \mathbf{si} X$	X Y	X starts 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 ✓

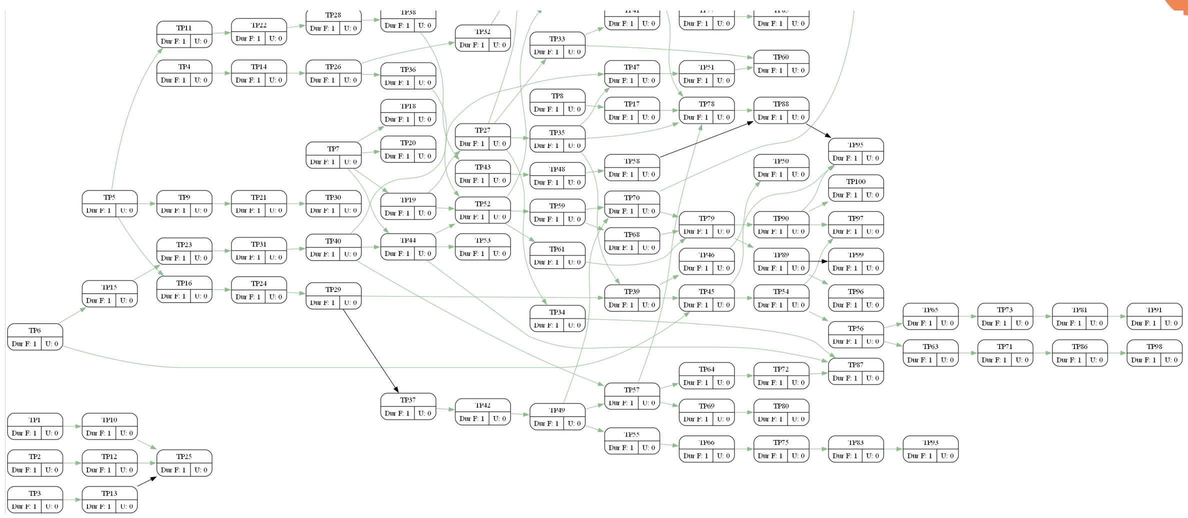
- The start of a job is equal to the start of the earliest task of the job
- The end of a job is equal to the latest end of any of its tasks
- Also called: the job *spans* its tasks
- Sometimes very simple
  - Start of job is start of first process step
  - End of job is end of last process step
  -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

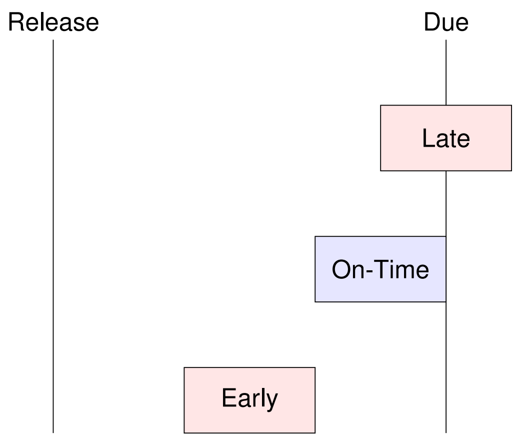
An Example of a More Complex Process



- 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

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 ✗

- 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

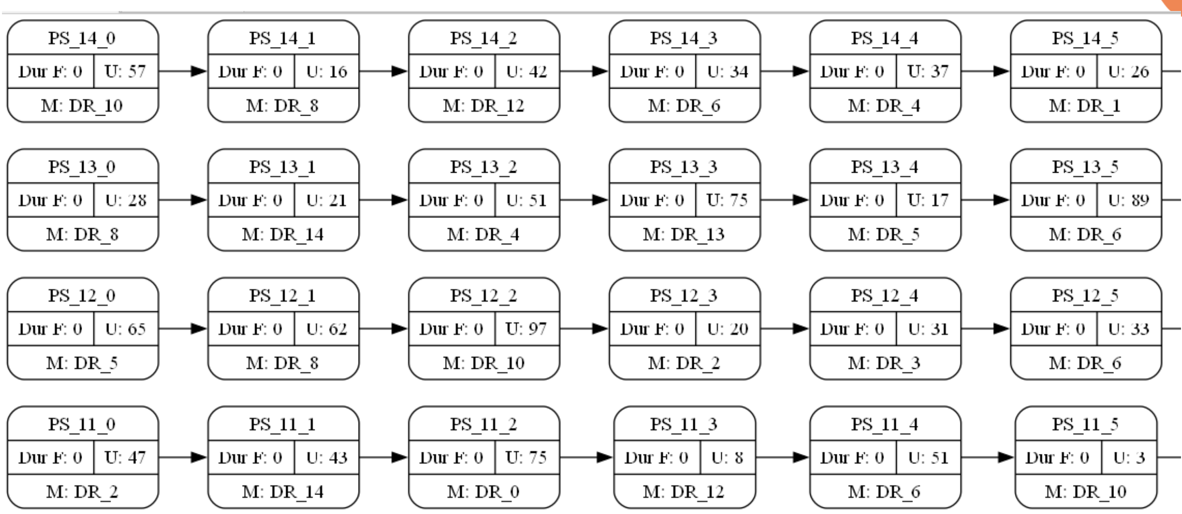
Problem Classification

- 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 ✓

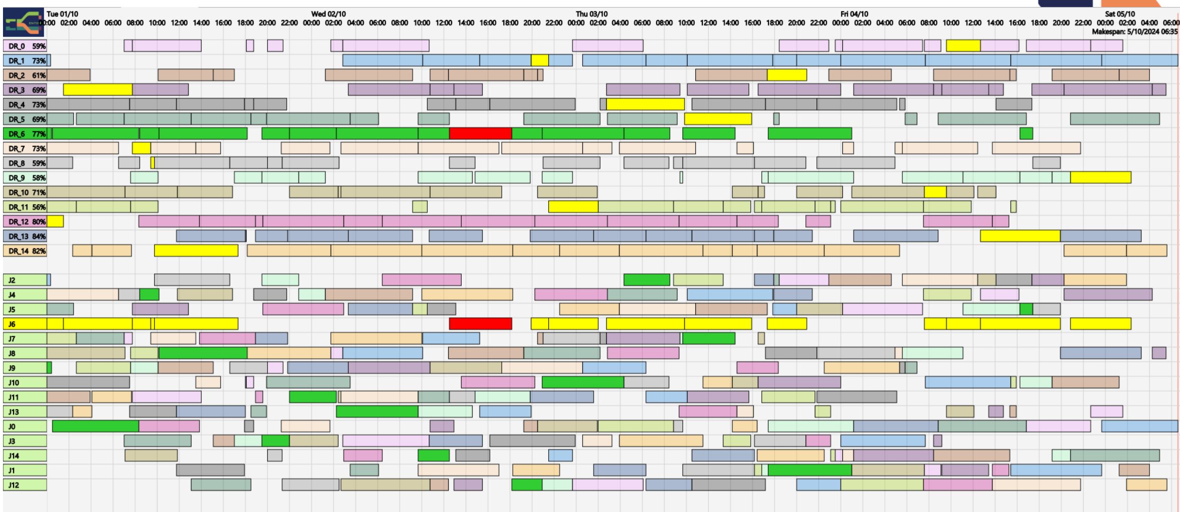
- Consists of a number of jobs and a number of machines
- Each job visits each machine, but possibly in a different order, depending on process
- Tasks of a job are linked as a precedence chain
- Objective is to minimize overall end, the *makespan*

Example Job-Shop Process



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

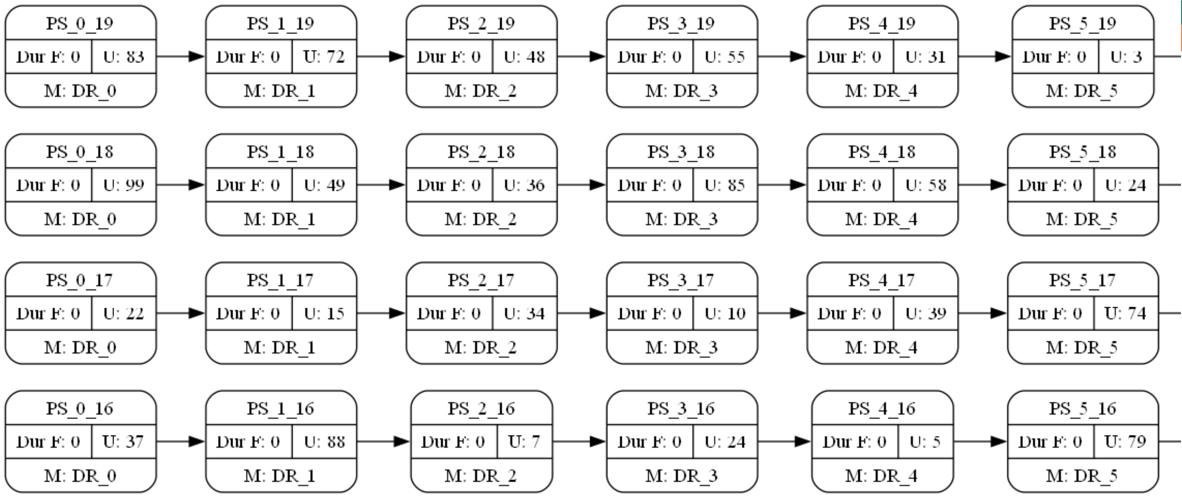
Example Job-Shop Solution



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

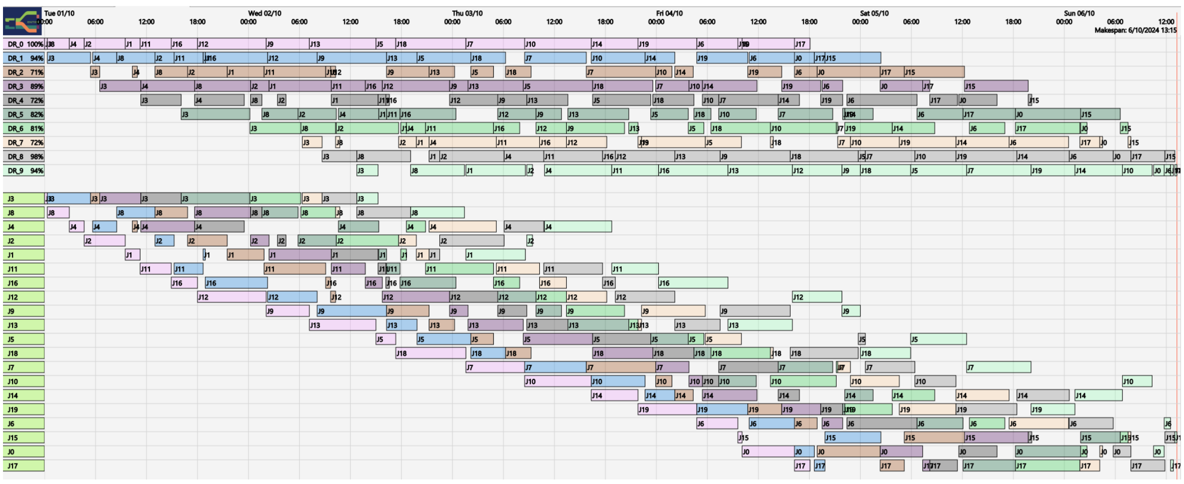
Flow-Shop ✓

- 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



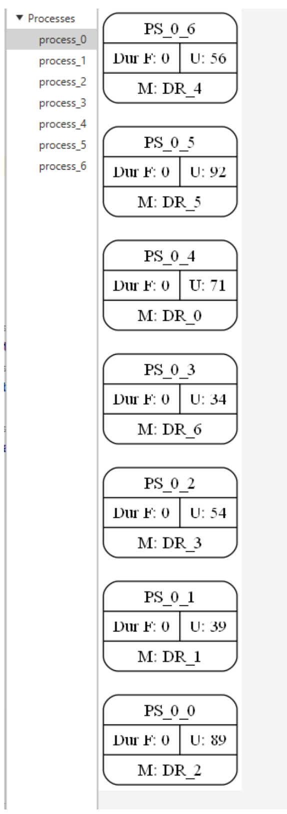
- Tasks are colored by machine, note the regular pattern in the Job Gantt Chart

Open-Shop ✓

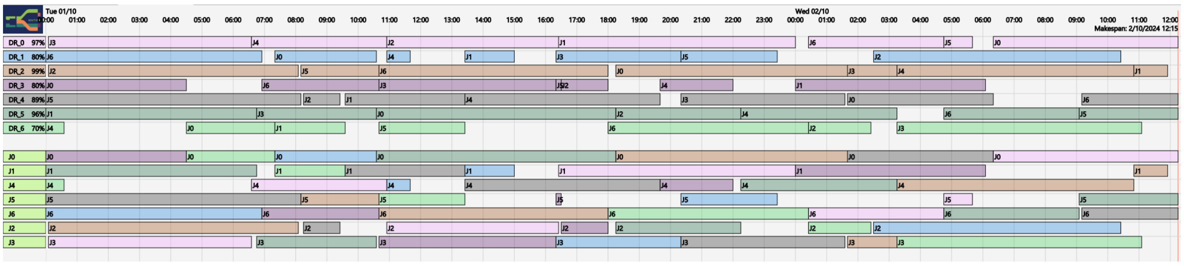
- Consists of a number of jobs and a number of machines
- Each jobs visits each machine, we have to choose the sequence individually for each order
- There are no temporal constraints between tasks, but tasks of the same job cannot overlap
- Objective is to minimize overall end, the *makespan*

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
- Note that machines are still idle in optimal solution

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

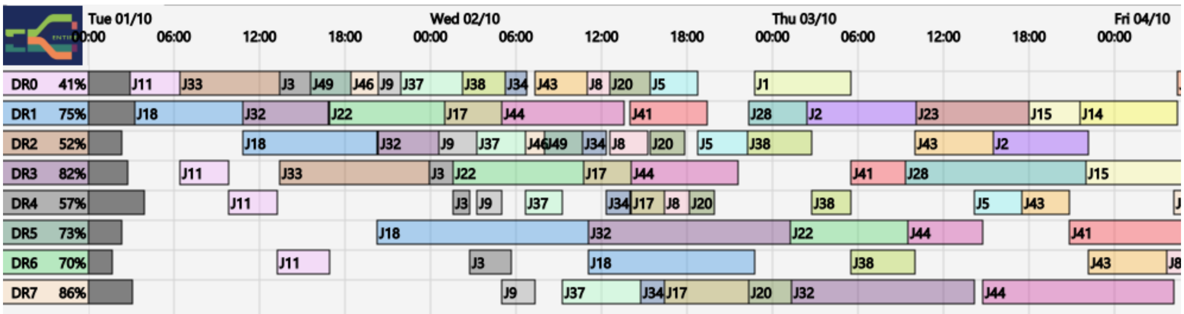
$\alpha/\beta/\gamma$ Notation

- 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`

Visualization

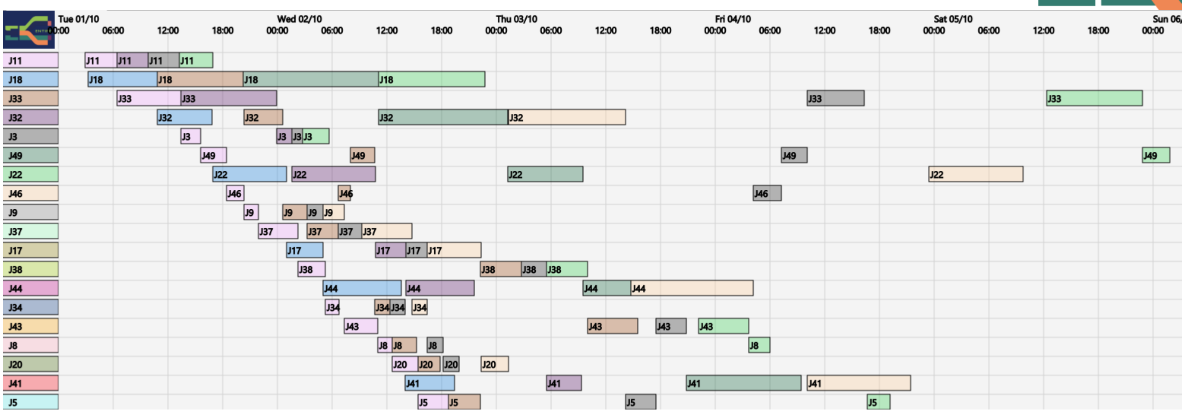
- Visualization is key to present and to understand results
- Many different ways to give an overview of schedule, and highlight problems
- Some diagrams types are used a lot, and are provided in our generic scheduling tool
- Customization is key

Machine Gantt Chart ✓



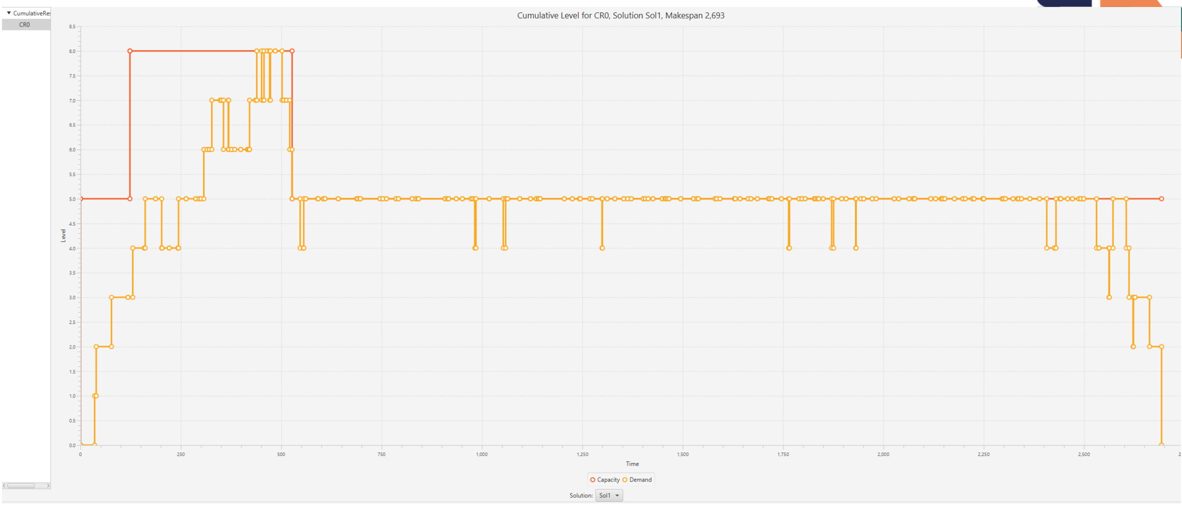
- Shows all tasks that are assigned to each machine
- Tasks should not overlap
- Also shows work in progress (WiP), down-times
- Optional display of setup and idle times

Job Gantt Chart ✓



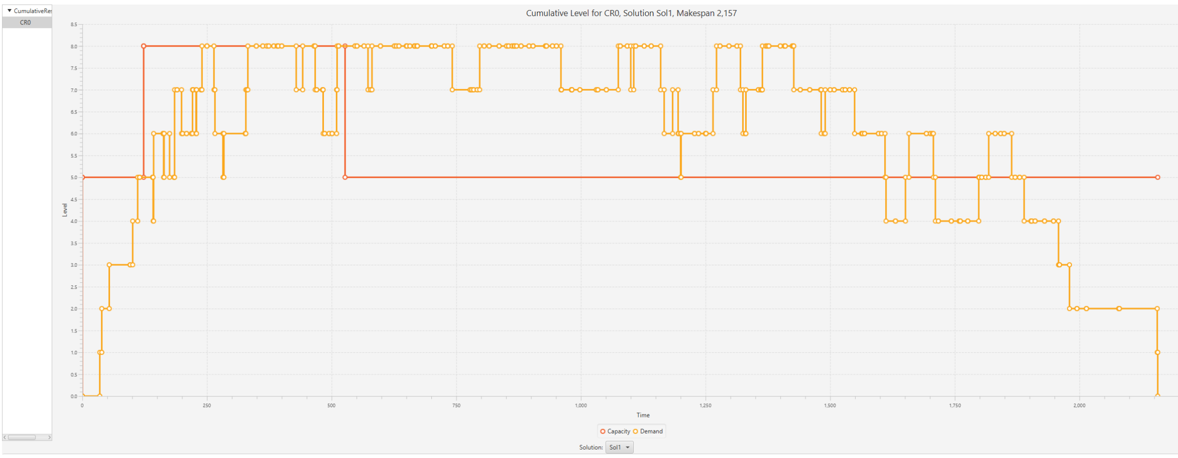
- Shows all tasks of a job in one line
- Only works for single chain of process steps
- Possible display of earliness, lateness
- Optional display of waiting and transport times

Cumulative Resource Chart ✓

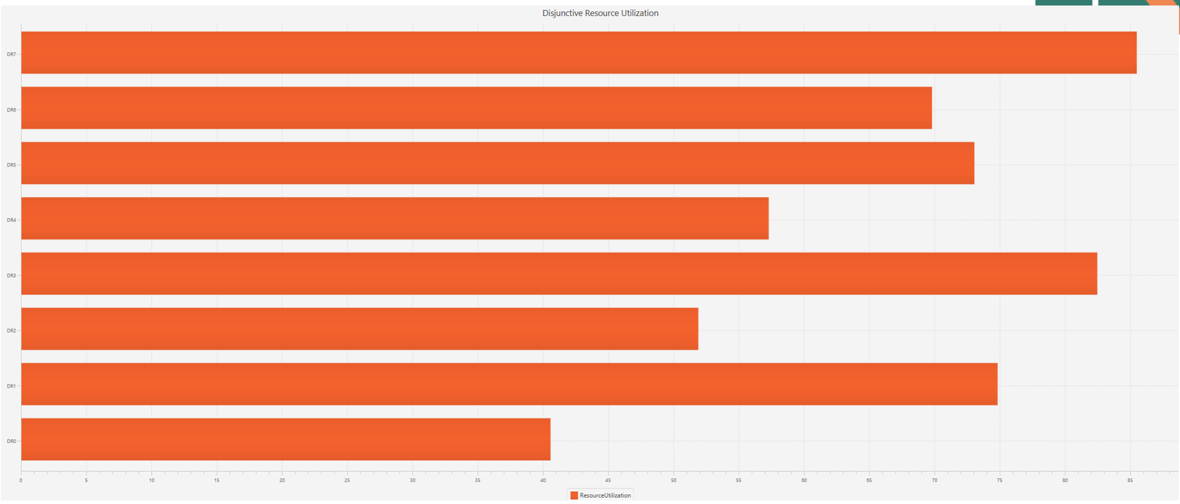


- Shows resource utilization of cumulative resource over time
- Utilization should be below capacity profile
- Unless we relax the cumulative resource constraint

Cumulative Resource Constraint Relaxed

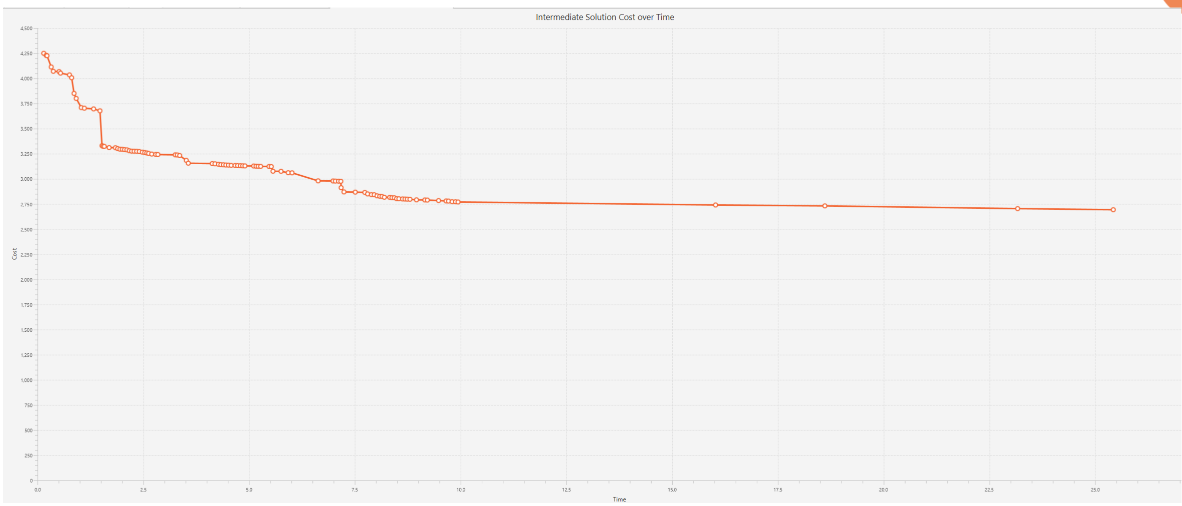


Resource Utilization ✓



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

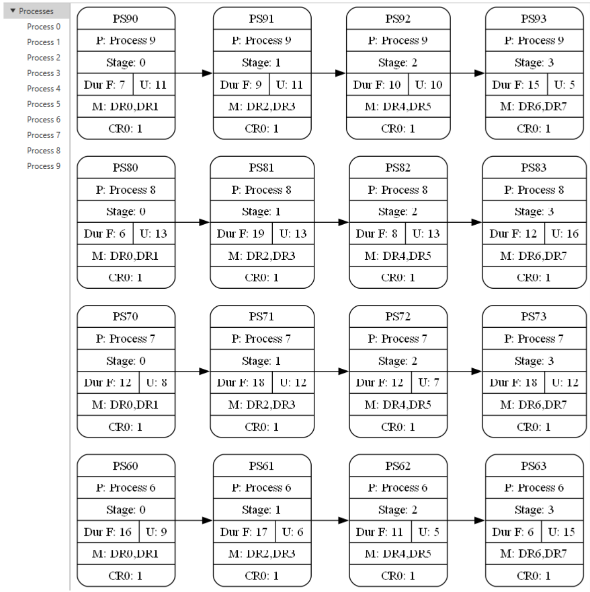
Intermediate Solutions ✓



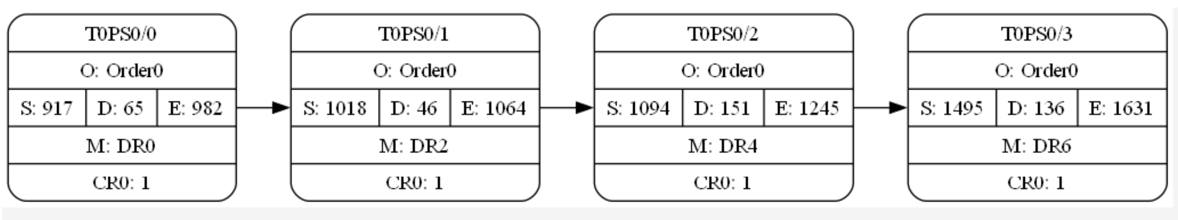
- Shows intermediate solutions found over time
- Useful to see if enough/too much time is allocated

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

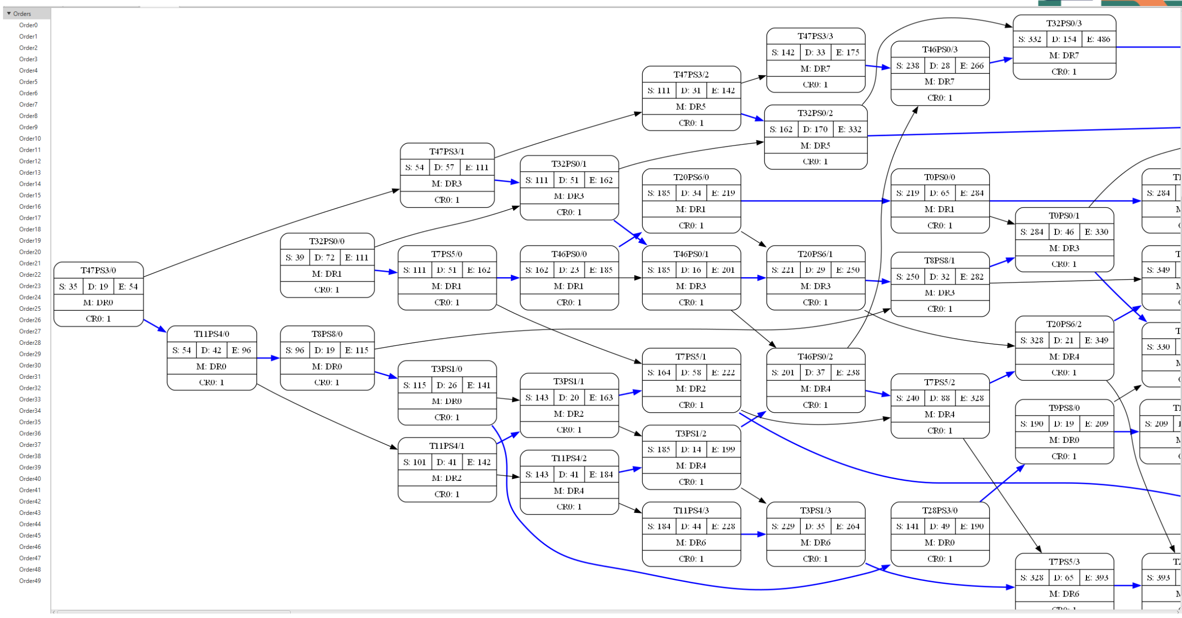


PERT Chart(Program Evaluation Review Technique) ✓



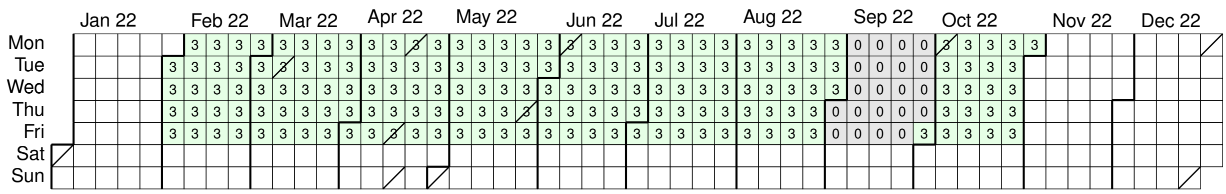
- Show details of job as a graph
- Useful if task graph is not a chain
- Often used in project management

PERT Charts become Confusing Quite Quickly



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

Calendars ✗



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

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

Part III

Machines and Resources

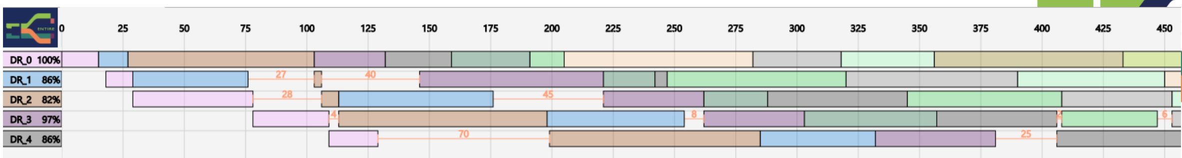
Key Points

- Introduce different types of resources
- Disjunctive resources - one task at a time
- Cumulative resources - demands and capacity
- Machine choice - Use one of multiple machines
- Work in progress and planned downtimes
- Calendars - Not working all the time

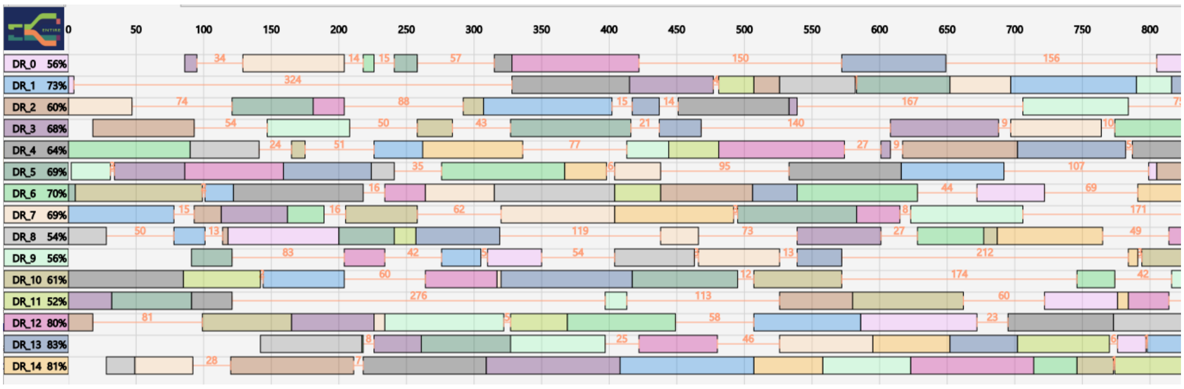
Disjunctive Resource ✓

- A *disjunctive resource* works on one task at a time
- Each task runs uninterrupted from start to end
- The machine may be *idle* between tasks
- The machine may be unused at start and end of schedule
  - Some of this may be unreachable, there is no work that can be done in these periods
  - Problem of cold start, especially for flow-shop type problems
- *Active time* is time between first and last use
- Resource utilization compares productive time to active or available time

Disjunctive Machines Examples



- Flow-Shop example, some unreachable time on later resources in process, some idle time



- Job-Shop example, a lot of idle time

Preemption ✗

- Normal constraint for disjunctive constraints is one task at a time
- Once a task is started, it runs until it is finished
- *Preemption* allows to stop a task, run a different task, then resume the previous task to the end
- Example: This is how Operating Systems run tasks inside a computer
  - This works since cost of suspending a task is relatively low
  - Also needed as tasks continuously produce output which is expected
- In manufacturing, preemption often is an exception in an emergency
- Occurs a lot in project management, e.g. construction

How to Deal with Preemption in Scheduling

1. Handle this as manual intervention for critical situations
2. Dedicated preemptive scheduling constraints
3. Allow limited number of interruptions
    - Split each task into multiple pieces of unknown length
    - Normally, schedule all parts together for total duration
    - For preemption, schedule other task after first/second part
    - All parts of task must add up to total duration

Cumulative Resources ✓

- A cumulative resource provides capacity over time, the sum of the demands at each timepoint cannot exceed the available capacity at that time
- Resource demand by one task is considered constant from start to end
    - Need to break task into smaller segments to model time variable demand
- In itself a hard problem, so full propagation not possible
    - Active research area since 1993, when the constraint was introduced in CHIP

Specifying Cumulative Resources

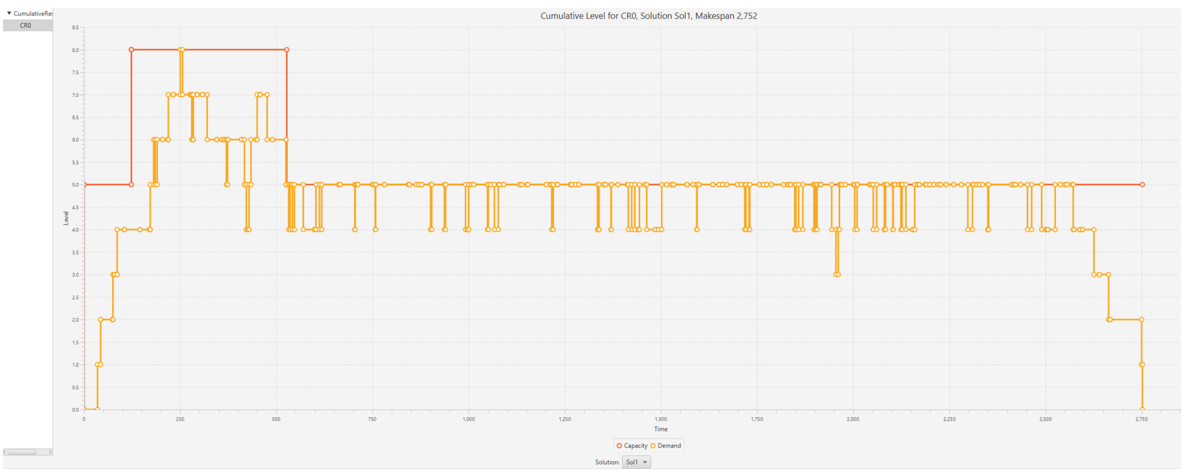
- Describing a cumulative resource
  - The resource itself
  - The capacity profile over time
  - The demands per processStep
- Each task may or may not need a specific cumulative resource
- The assumed total amount of work needed is constant
- We can calculate resource utilization by comparing demand to capacity

name	*CR0*

fromDate	*17/10/2024 07:16*
cumulativeResource	*CR0*
name	*CP00*
from	*0*
capacity	5

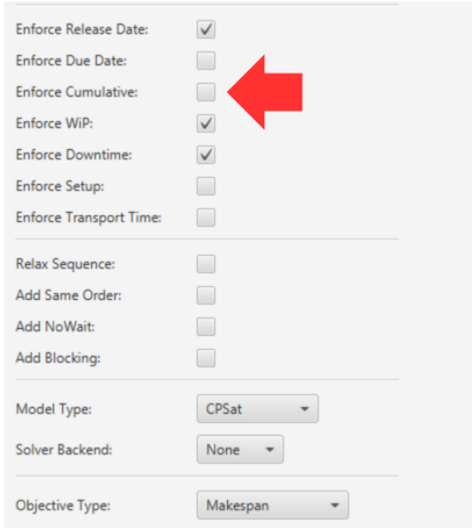
cumulativeResource	*CR0*
name	*CN0/0/CR0*
processStep	*PS0/0*
demand	1

Cumulative Resource Profile

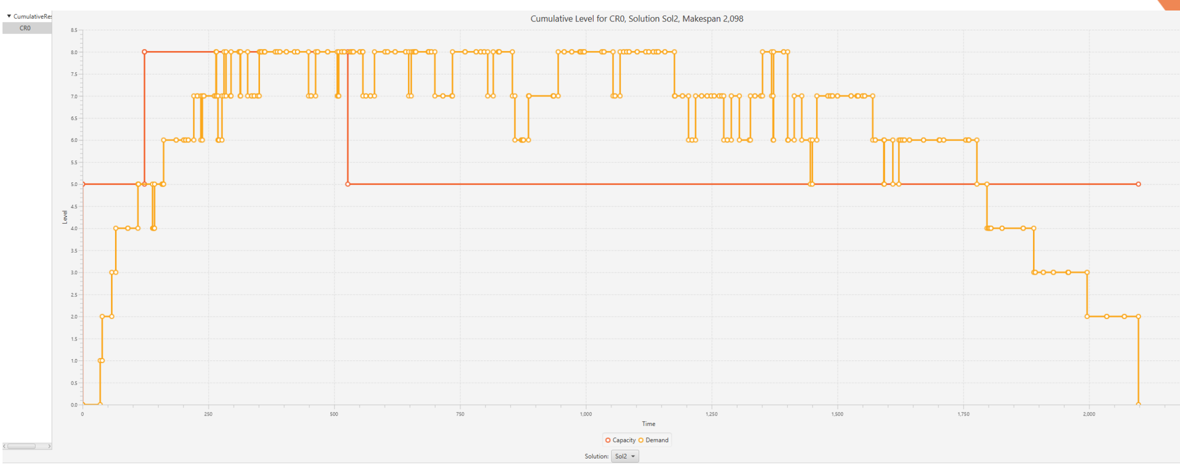


What is the Impact of the Cumulative?

- We want to understand what impact a cumulative resource has
- We can disable the constraint in the solver options
- Re-run the scheduler
- Observe the impact on the objective
- See where the capacity limit is not respected in new solution



Cumulative Profile When Constraint is Disabled



- Objective reduced from 2,752 to 2,098
- Overall resource use now reaches 8 in period where capacity is limited to 5

Variant: Resource Limit as Objective ✗

- For some scheduling problems, the duration of the schedule is fixed
- The objective is: how many resources are needed to schedule all tasks within the available time?
- Capacity is a variable, part of objective function
- Example later on for assembly line balancing
  - Number of stations on line is fixed
  - Objective is to minimize *Takt*, the cycle time allocated for one step
- Consider solving this question with multiple scenarios, instead of different objective

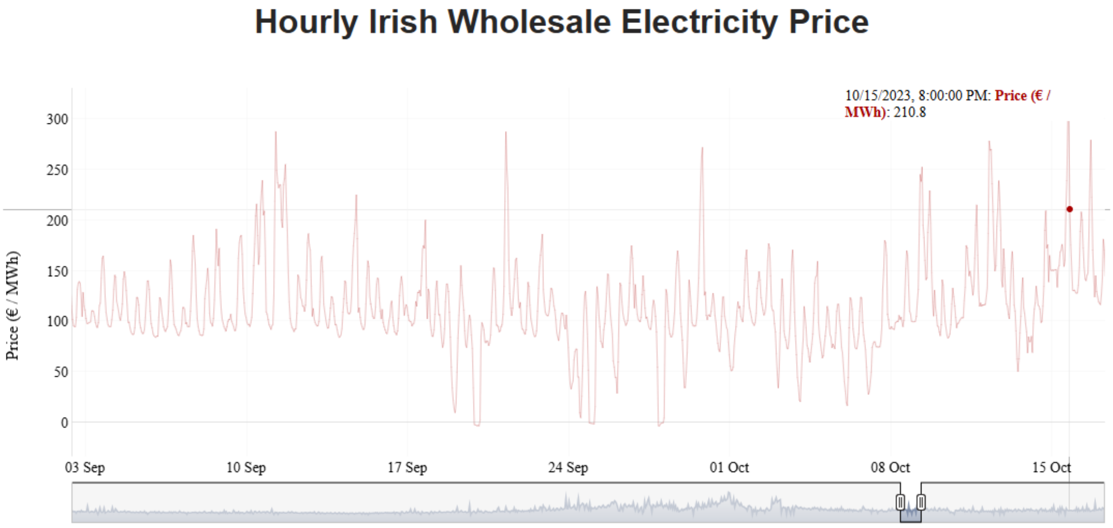
Variant: Trading Time for Capacity ✗

- In some cases, the duration of a task depends on how many resources are available
- Total amount of work (energy) is constant, higher demand (power) means lower duration
- In easiest case, fixed demand levels are assumed
  - Resources are assigned to task throughout duration
  - Example: assigning software engineers to projects
  - Remember Books's law
    - *Adding manpower to a late software project makes it later.*
- Most general case, any profile is OK, as long total demand is covered
  - Cost of reassigning resource from one task to another is considered minimal

Variation: Time Variable Resource Cost ✗

- Resource cost may vary over time
- Example: Overtime cost for working on weekend
- Example: Energy cost with time variable tariff
- Avoid periods of high cost, use areas of low cost

Example: Electricity Price in Ireland



from: `https://kilowatt.ie/wholesale-electricity-prices-ireland/`

Variant: Soft/Hard Limits ✗

- Often, some capacity is available for "free", sunk costs
- Resource use above that limit costs extra
- Example: Full-time staff/contract workers
- Example: In-house capacity/rented computing capacity
- Multiple profiles, each with its own cost per unit

Variant: Lower Utilization Limit ✗

- Sometimes, we also want to enforce a lower limit of the resource use
- We want to avoid resources being idle
- Express a lower limit on the resource use
- Can be hard to satisfy for specific demand and capacity values

Manpower Constraints ✓

- Use cumulative constraints to express manpower limits
- Some tasks may need multiple workers
- Total capacity profile is number of workers available at each time
- Profile may change with shift-pattern (regular pattern)
- Holidays/sick-leave/training reduce available manpower at specific times
- Constraint does not assign workers, only checks that enough capacity is available

Skills ✗

- Not all workers have the same qualifications
- Workers may need to be trained/certified to perform certain tasks
- Each task may require specific skill(s)
- Nested resource constraints to cover the needed skills
  1. One worker may have all required skills, only one worker is needed
  2. Multiple workers needed to cover all required skills, no worker has all skills
  3. More than one worker needed anyway, the group must cover required skills
- Training/certification program may create its own scheduling problem

Alternative: Assigned Operators ✗

- In special cases, it may be required to assign specific workers to tasks
- Each worker can work on one tasks at a time (disjunctive constraint)
- Multiple workers are qualified to perform certain jobs (machine choice, one worker is assigned)
- Multiple workers are qualified to perform certain jobs (multiple workers with that skill are needed)
- Named individual must be assigned for traceability
  1. Is there a hand-over from one shift to the next?
  2. Complete work must be performed within one shift

Fractional Manpower Needs ✗

- Some tasks may not need a full-time operator
- Different scenarios
  - Operator only needed at start/end of task (setup, cleaning)
  - Operator is needed to load/unload items into machine
  - One operator can supervise three, but not four machines
- This gets too complex/too fragile very quickly

Choosing which machine to use

- Problem with Job-shop/Flow-shop: There is only one machine per processStep
  - What happens if any of those machines stops working?
  - Do we stop production completely?
- Most plants have multiple machine for the same task
- Three fundamental alternatives
  - Multiple, identical machines
  - Multiple machines with different speeds
  - Preferences for specific machines, but viable alternatives exist
- On the other hand, sometimes identical machines are treated as different
  - Dedicated lines for major products, avoiding setup/cleaning times

Identical Machines ✓

- Easiest case, several machines of same type
- You can choose any of the available machines
- Processing time is the same on all machines
- Product quality is identical
- Define which machines are available with ResourceNeed

Machine Dependent Speed (✓)

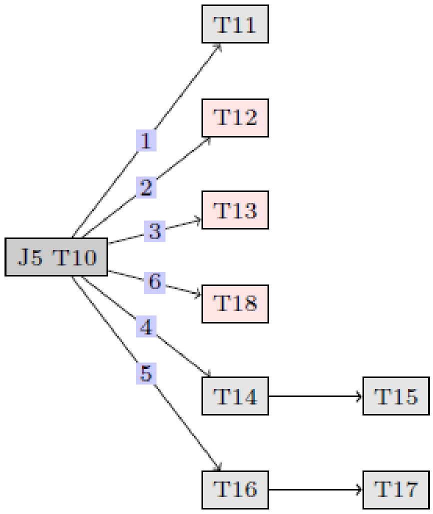
- Duration of the task depends on machine on which it is run
- Two common scenarios
  1. Some machines are faster than others (new generation)
  2. Different processes are faster/slower on some machines
- Express task duration as part of ResourceNeed
- Prefer faster machines, but balance machine use

Machine Preference (✓)

- Each process step has a preference ranking of machines, from best to worst
- Potential Causes
  - Product quality
  - Production speed
  - Production cost
  - Skill level required
  - Scrap rate
- Handle preferences as part of objective
- Enforce certain levels of preference to understand impact

Example from Siemens Energy Case Study

- Six alternatives for task T10
- Preference ranking from one (best) to six (worst)
- Some alternatives require additional tasks
- Tasks in red are outsourced



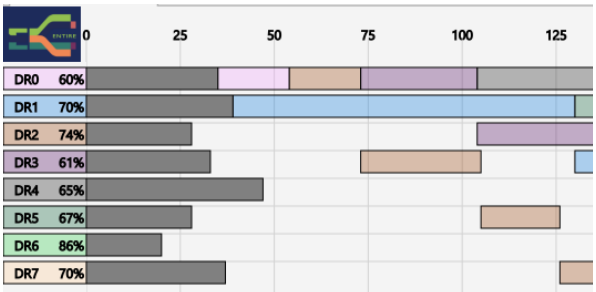
Work in Progress ✓

- Typical a plant does not start from scratch
- There is work currently running on machines
- This work must be finished before new work can be scheduled
- Called *Work in Progress (WiP)*
- Specified in input data

Name	DisjunctiveResource	Duration	Start	End	StartDate	EndDate
WDR0	DR0	35	0	35	1/10/2024 00:00	1/10/2024 02:55
WDR1	DR1	39	0	39	1/10/2024 00:00	1/10/2024 03:15
WDR2	DR2	28	0	28	1/10/2024 00:00	1/10/2024 02:20
WDR3	DR3	33	0	33	1/10/2024 00:00	1/10/2024 02:45
WDR4	DR4	47	0	47	1/10/2024 00:00	1/10/2024 03:55
WDR5	DR5	28	0	28	1/10/2024 00:00	1/10/2024 02:20
WDR6	DR6	20	0	20	1/10/2024 00:00	1/10/2024 01:40
WDR7	DR7	37	0	37	1/10/2024 00:00	1/10/2024 03:05

Work in Progress ✓

- Typical a plant does not start from scratch
- There is work currently running on machines
- This work must be finished before new work can be scheduled
- Called *Work in Progress (WiP)*
- Specified in input data, shown in gray
- Part of the disjunctive constraints



Planned Downtimes ✓

- Sometimes, a machine is unavailable for a period of time
- Maintenance, upgrade
- Planned activity with fixed start and end
- This should be considered in schedule
- Given as input data
- Part of the disjunctive constraints
- Gaps may lead to loss of productivity

Name	DisjunctiveResource	Duration	Start	End	StartDate	EndDate
DDR1	DR1	51	3,749	3,800	14/10/2024 00:25	14/10/2024 04:40
DDR2	DR2	66	5,137	5,203	18/10/2024 20:05	19/10/2024 01:35
DDR4	DR4	52	2,888	2,940	11/10/2024 00:40	11/10/2024 05:00
DDR6	DR6	57	4,412	4,469	16/10/2024 07:40	16/10/2024 12:25

Variant: Scheduled Downtime ✗

- Sometimes, we can decide when the downtime should occur (within reason)
- We can schedule it like any other task
- Avoid unproductive gaps in schedule
- More complex case for regular, scheduled downtimes
  - Maintain the correct time gap between maintenance checks
- How is in control in scheduling these events?

Unplanned Downtime ✗

- A machine breaks down unexpectedly
- This is not reflected in current schedule (unplanned)
- How to react?
  - Extend current task until finished (if task continues after breakdown)
  - Create new task to complete work later on (if task is partially finished)
  - Scrap task, reintroduce order in next schedule (if task is scrapped by breakdown)

Calendars

- A plant may not run 24/7, but shut down for regular/irregular periods
  - Overnight
  - Weekend
  - Public holidays/holidays/Christmas
- Some parts of plant may operate on different calendars
  - Office/lab may be working office hours only
- Considering multi-site problems, plants may be working in different time-zones
  - Common example: data centres around the world

Important Questions

- Which time points/time periods are expressed in working time, which in wall time?
- Examples
  - Release/due dates typically expressed in wall time
  - Task duration expressed in working time
  - Min/max waiting time expressed in wall time

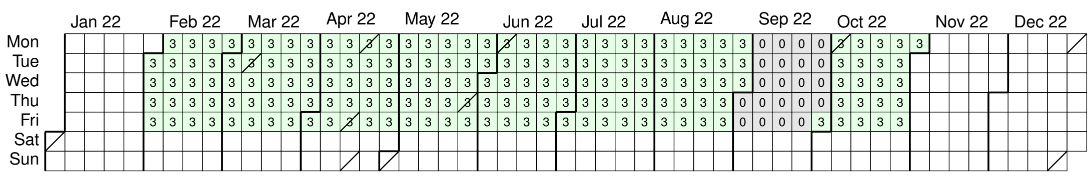
Single, Factory-wide Calendar

- Three shift operation common
  - 06:00 - 14:00
  - 14:00 - 22:00
  - 22:00 - 06:00
- Start/end of weekend not obvious
- Handling of public holidays plant specific
- Lots of input data

Shift Pattern Definition ✗

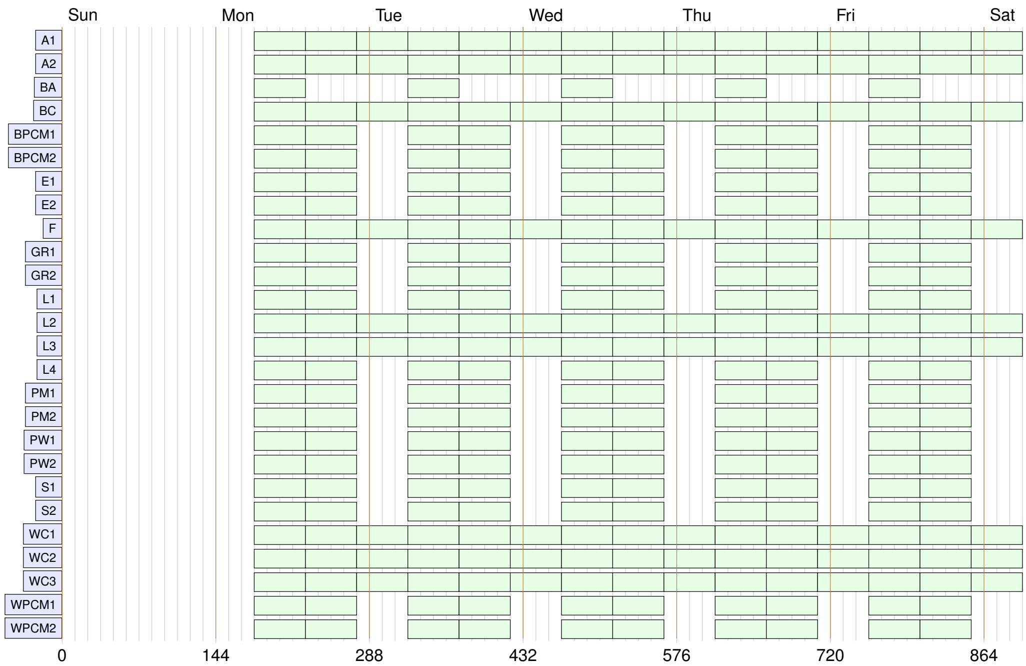
Name	Shift Model	Percentage	Start Date	End Date	Start Time	End Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
AV12	Shift 15 1	0.80	01/02/2022	01/11/2022	06:00	14:00	x	x	x	x	x		
AV13	Shift 15 2	0.80	01/02/2022	01/11/2022	14:00	22:00	x	x	x	x	x		
AV14	Shift 15 3	0.80	01/02/2022	01/11/2022	22:00	06:00	x	x	x	x	x		
UV4	Shift 15 1	0.00	01/09/2022	30/09/2022	06:00	14:00	x	x	x	x	x		
UV5	Shift 15 2	0.00	01/09/2022	30/09/2022	14:00	22:00	x	x	x	x	x		
UV6	Shift 15 3	0.00	01/09/2022	30/09/2022	22:00	06:00	x	x	x	x	x		

- Definition of three shifts for Mon-Fri, shut-down in September
- Plant does not shut-down for Bank holidays (marked /)



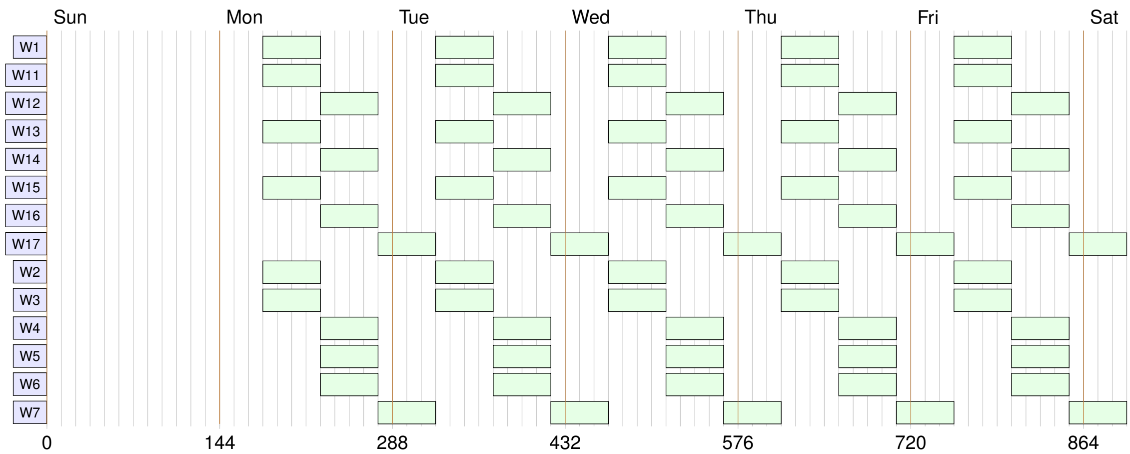
- Resulting shift calendar

Weekly Machine Dependent Calendar



- Note machines running one shift, two shifts, or three shifts

ShiftPattern for Workers



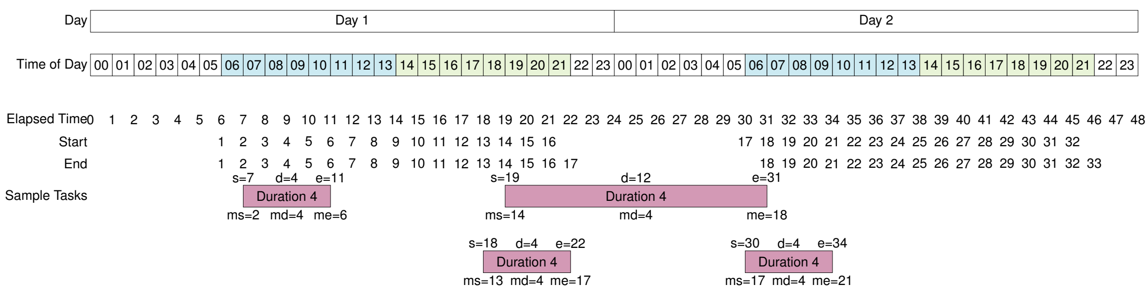
- Note different resource levels for morning, afternoon and night shift

Calendar Dependent Duration ✗

- In some factories, tasks have different duration depending on when they are run
- During the night-shift fewer workers are available, tasks like cleaning take longer
- During holidays, most expert operators are absent, tasks take longer due to less experienced operators
- For every working period, define a utilization factor to define nominal capacity (0-100%)
  - Tasks take longer if work capacity is lower
  - Only supported in few solvers (CPO)

Tasks Stretching over Shutdown ✗

- When a machine does not run continuously, the duration of tasks in wall time may change



- Task starting at 07:00 has duration of 4 hours
- Same task starting at 19:00 stretches over nightly shutdown (22:00-06:00), extending its duration to 16 hours

Summary

- Introduced different resource types
  - Disjunctive resources
  - Cumulative resources
  - Machine choice
- Identifying resources is a key element of defining scheduling problem
- Many problem specific variants exist, also impacting the constraint reasoning
- Keep as simple as possible - as complex as required
- Not all described variants already in our generic tool

Part IV

Experiments

Key Points

- This section describes the scheduling tool
  - This is a *preview* of the current state, not released yet!
- How to load/create data
  - From files
  - By instance generator
  - From benchmark problems
- How to run the solvers
  - Which solvers are supported
  - What to expect in terms of performance
- Experiments to try
  - Limited time
  - Possible "test before invest" continuation

The Scheduling Tool

- We create the tool as basis for experiments
- To test ideas and solvers
- As a teaching tool
- Slightly higher standard than usual academic prototypes
  - This is a *preview*, not released yet
- Not a commercial tool
  - But can use commercial solvers
  - Also open-source solvers
- Written in Java, JavaFX
- Can also be used as a back-end scheduling server
- Uses our Java application framework generator
- Will become available in early 2025

Back-end solvers

- Provide both open-source and commercial solver interfaces
- Allow experimentation without having to buy commercial tools straightaway
- Gives a level playing field to compare solvers and models
- Provides out-of-the-box, generic performance

Google OR-Tools CPSat Solver

- Open-Source tool provided by Google
- Available at `https://developers.google.com/optimization/cp/cp_solver`
- Probably best open-source CP solver for scheduling
- This solver is packaged with scheduler

Example Problem

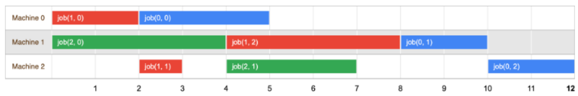
Below is a simple example of a job shop problem, in which each task is labeled by a pair of numbers (m, p) where m is the number of the machine the task must be processed on and p is the *processing time* of the task — the amount of time it requires. (The numbering of jobs and machines starts at 0.)

- job 0 = [(0, 3), (1, 2), (2, 2)]
- job 1 = [(0, 2), (2, 1), (1, 4)]
- job 2 = [(1, 4), (2, 3)]

In the example, job 0 has three tasks. The first, (0, 3), must be processed on machine 0 in 3 units of time. The second, (1, 2), must be processed on machine 1 in 2 units of time, and so on. Altogether, there are eight tasks.

A solution for the problem

A solution to the job shop problem is an assignment of a start time for each task, which meets the constraints given above. The diagram below shows one possible solution for the problem:



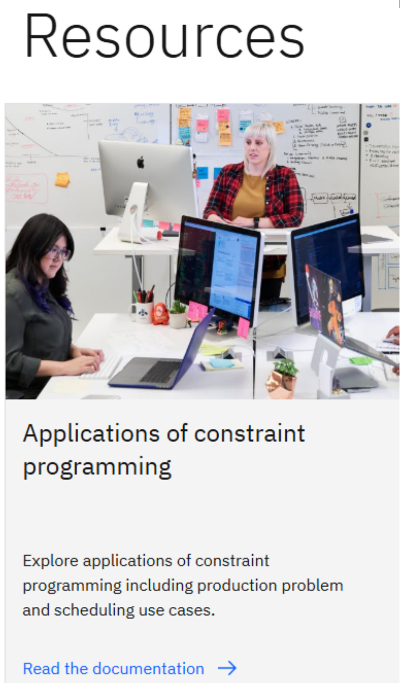
You can check that the tasks for each job are scheduled at non-overlapping time intervals, in the order given by the problem.

The length of this solution is 12, which is the first time when all three jobs are complete. However, as you will see below, this is not the optimal solution to the problem.

(from OR-Tools website)

CP Optimizer from IBM

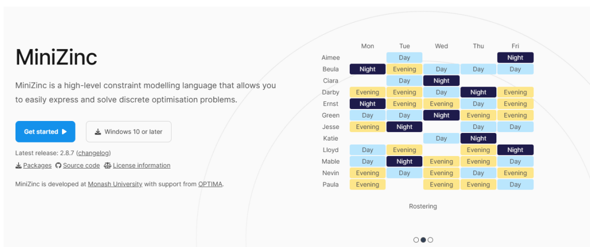
- Commercial tool of IBM
- `https://www.ibm.com/products/ilog-cplex-optimization cplex-cp-optimizer`
- Part of optimization suite with Cplex, OPL
- We do **not** provide this solver, we allow to interface with it
- Academic licenses available
- Well-known for capabilities for scheduling



(from CPOptimizer website)

MiniZinc from Monash University

- Modelling language and backend tools from Monash University in Melbourne, Australia
- Available from `https://www.minizinc.org/`
- Widely used for teaching
- Allows different backend solver to run from same model
- Generic CP tool, not optimized for scheduling
- Requires separate installation, open-source



(from MiniZinc Website)

Which Solver is Better?

- We present results on a few benchmark types
- Fair comparison between solvers
  - Same hardware, Windows 11 laptop
  - CPU i7-10875H @ 2.3GHz, 64GB, four cores
  - Same timeout (600 s)
- Not a fair comparison to state-of-the-art
  - Uses out-of-the-box model
  - Significant improvements possible
  - More specific models
  - Parameter tuning
  - Unlimited runtime

Taillard Job-Shop Benchmarks

		All Instances				Optimal Only		Non Optimal Only			
		Optimal (% of All Instances)				Time (% of VB)		Cost (% of VB)		Bound (% of VB)	
Group	Nr	Both	CPO	CPSat	None	CPO	CPSat	CPO	CPSat	CPO	CPSat
15/15	10	90.00	0.00	0.00	10.00	105.19	141.18	100.00	100.00	97.17	100.00
20/15	10	20.00	0.00	0.00	80.00	267.27	263.20	100.99	100.05	98.50	99.93
20/20	10	0.00	0.00	0.00	100.00	n/a	n/a	100.74	100.06	97.96	100.00
30/15	10	10.00	0.00	10.00	80.00	174.32	100.00	100.18	100.49	99.87	100.00
30/20	10	0.00	0.00	0.00	100.00	n/a	n/a	100.30	101.30	99.40	100.00
50/15	10	100.00	0.00	0.00	0.00	100.00	685.09	n/a	n/a	n/a	n/a
50/20	10	10.00	60.00	0.00	30.00	100.00	381.38	100.00	101.60	100.00	100.00
100/20	10	10.00	90.00	0.00	0.00	100.00	416.13	100.00	101.73	100.00	66.81

- Significant number of problems solved to optimality in 600s
- In terms of quality, solvers are quite similar
- CPO wins in terms of solution times for larger instances

Results for Hybrid Flexible Flow-Shop

		All Instances				Optimal Only		Non Optimal Only			
		Optimal (% of All Instances)				Time (% of VB)		Cost (% of VB)		Bound (% of VB)	
Group	Nr	Both	CPO	CPSat	None	CPO	CPSat	CPO	CPSat	CPO	CPSat
20	25	76.00	0.00	20.00	4.00	100.00	580.71	100.00	100.00	96.52	100.00
25	25	80.00	0.00	8.00	12.00	101.65	238.02	100.00	100.37	97.67	100.00
30	25	60.00	0.00	4.00	36.00	100.35	264.69	100.18	101.05	100.00	100.00
40	25	4.00	16.00	0.00	80.00	100.00	2554.03	100.00	104.68	100.00	100.00
50	25	0.00	4.00	0.00	96.00	n/a	n/a	100.00	107.87	100.00	100.00
100	25	0.00	0.00	0.00	100.00	n/a	n/a	100.00	120.43	100.00	100.00
200	25	0.00	0.00	0.00	100.00	n/a	n/a	100.00	188.60	100.00	100.00
300	24	0.00	0.00	0.00	100.00	n/a	n/a	100.00	263.22	100.00	100.00
400	25	0.00	0.00	0.00	100.00	n/a	n/a	100.00	246.34	100.00	100.00

- Only smaller/medium instances solved to optimality
- For those problems, both solvers perform well
- CPO significantly better on large instances

General Recommendations

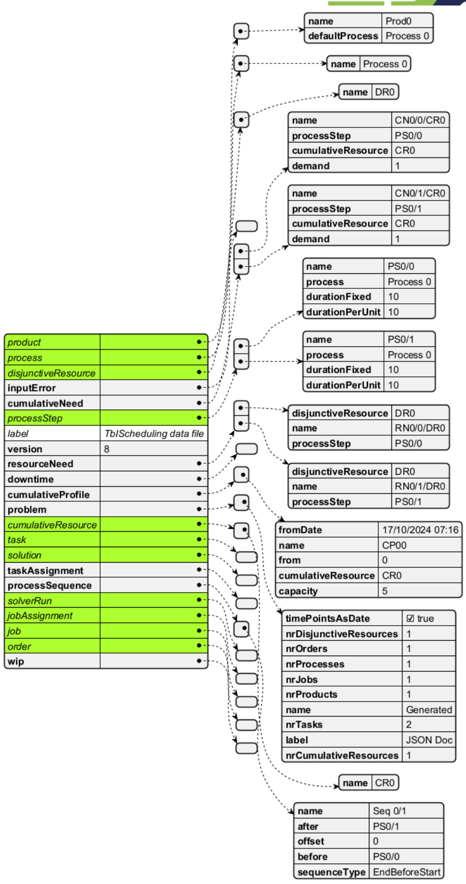
- If you already have access to CPO, use it!
- For new problem types, do an evaluation with CPSat first
- Out of the box, CPO performs more consistently
- May be easier to extend CPSat with your own research
- Use multiple cores and memory to your advantage

Input Data

- We have defined a specific JSON data format to describe scheduling problems
- This is different from the native/XML data format of the application (do not use)
- Load with menu `File - Load DataFile...`
- Save with menu `File - Save DataFile...`
- The format is described in a document

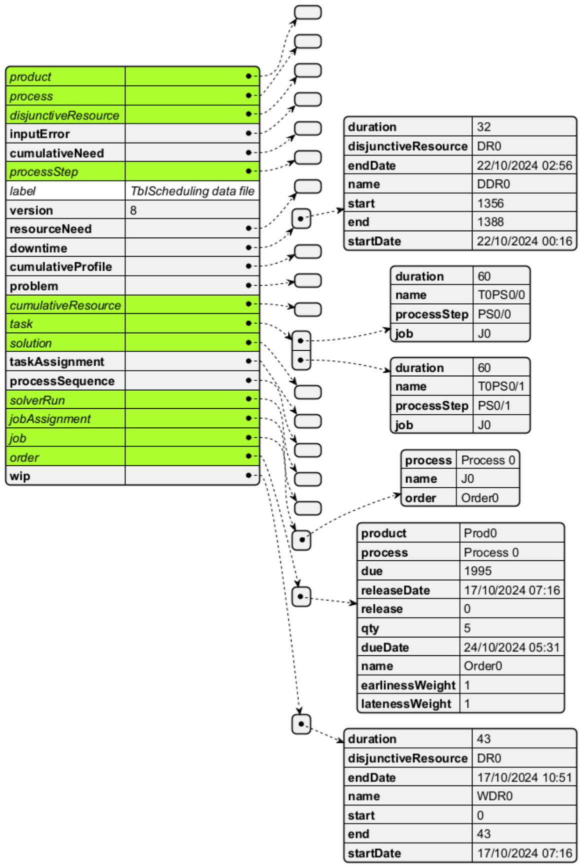
Base Data

- Description of
  - Product
  - Process
  - DisjunctiveResource
  - CumulativeNeed
  - ProcessStep
  - ResourceNeed
  - CumulativeProfile
  - Problem
  - CumulativeResource
  - ProcessSequence



Schedule Input Data

- Description of
  - Downtime
  - Task (x2)
  - Job
  - Order
  - WiP

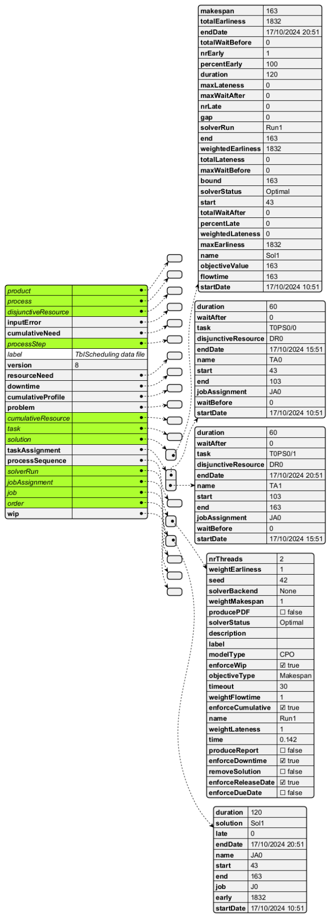


Result Data

- We use the same JSON format to describe the results of the schedule
- Added field types for SolverRun, Solution, assigned Jobs and Tasks

Sample Results

- Description of
  - Solution
  - SolverRun
  - Job Assignment
  - Task Assignment

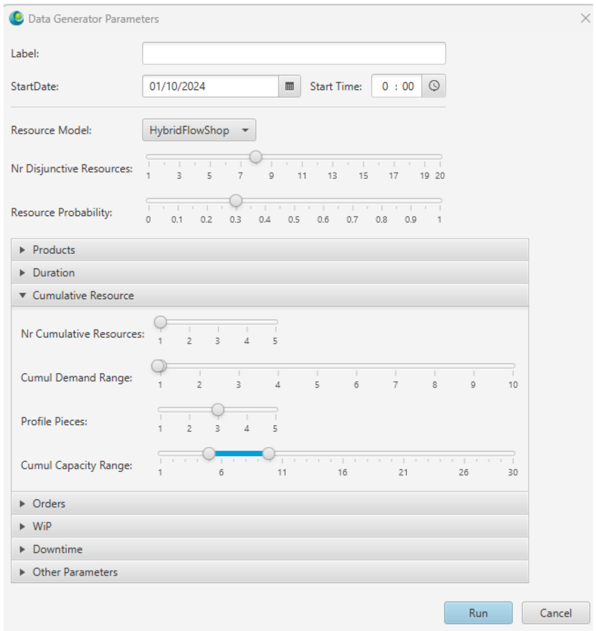


Instance Generator

- Application allows to generate different types of test problems
- Different types of resource models
- Different numbers of orders, resources, WiP, downtime
- Useful to generate more life-like examples combining different constraint types

Instance Generator Dialog

- Resource Model
  - Select a resource model defining the overall structure of problem
- Nr Disjunctive Resources
  - Describe how many disjunctive resources are generated
- Resource Probability
  - The probability that a resource is compatible with a task
  - Only for some resource models

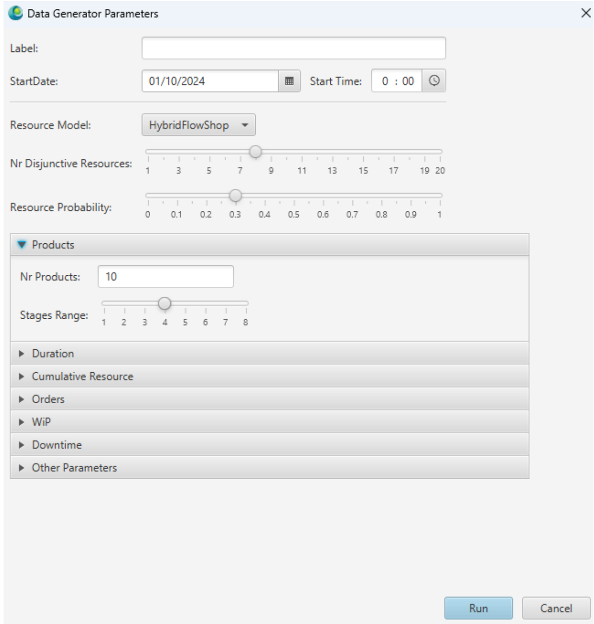


Resource Models

- Flow-Shop
  - Multiple stages, all jobs use machines in same order
- Job-Shop
  - Multiple stages, jobs use machines in different order
- Open-Shop
  - Multiple stages, no predefined order of machines
- Hybrid Flow-Shop (default)
- Hybrid Job-Shop
- Hybrid Open-Shop
  - Like x-shop, but with multiple machines per stage
- Random
  - Multiple stages, each stage using a random subset of machines
- All
  - Multiple stages, each stage allowing all machines

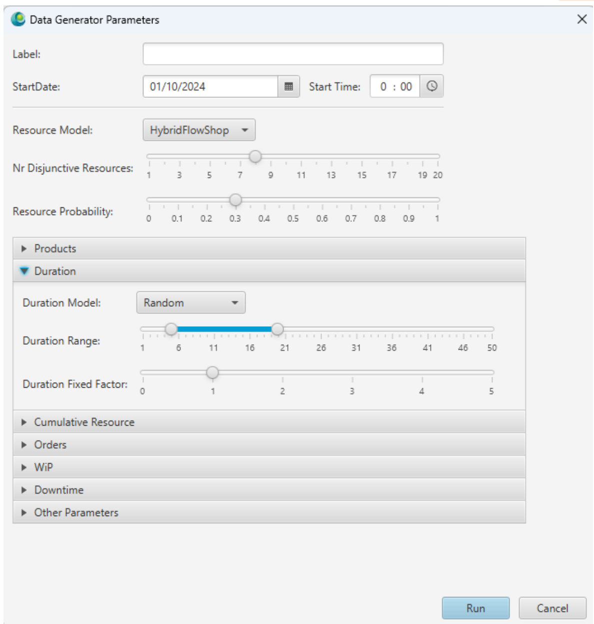
Instance Generator - Products

- Nr Products
  - Number of products to be generated
  - Products may be reused by multiple orders
- Stages Range
  - Range slider, sets lower and upper bound on number of stages



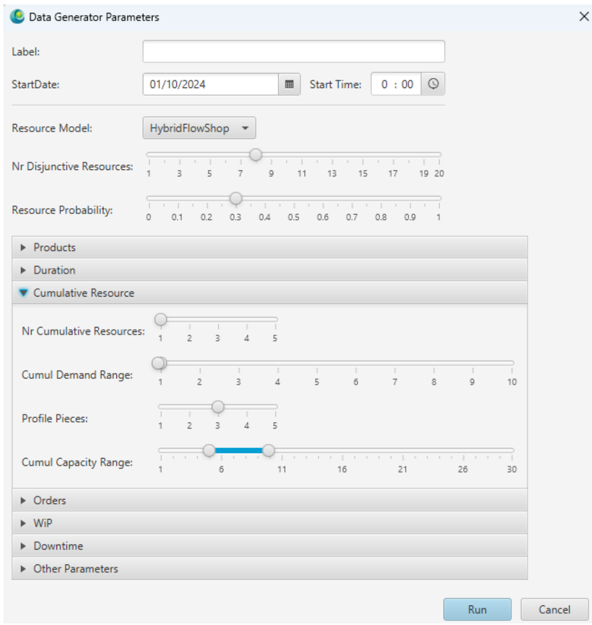
Instance Generator - Duration

- Duration Model
  - Different ways to link duration of processSteps
- Duration Range
  - Range slider to set lower and upper bounds on perUnit duration
- Duration Fixed Factor
  - How fixed and perUnit duration values are linked



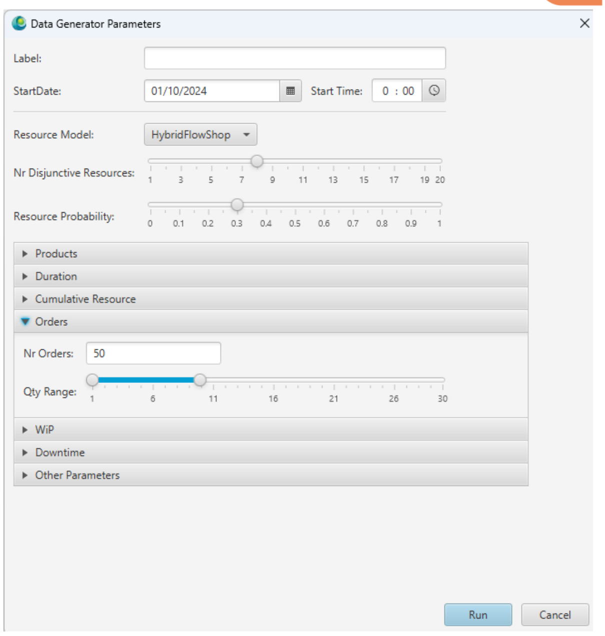
Instance Generator - Cumulative

- Nr Cumulative Resources
  - Number of cumulative resources generated
- Cumul Demand Range
  - Range slider to select lower and upper bound on cumulativeResource-Need demands
- Profile Pieces
  - Number of segments of CumulativeProfile generated for each resource
- Cumul Capacity Range
  - Range slider to select lower and upper bounds on cumulative profile capacity values



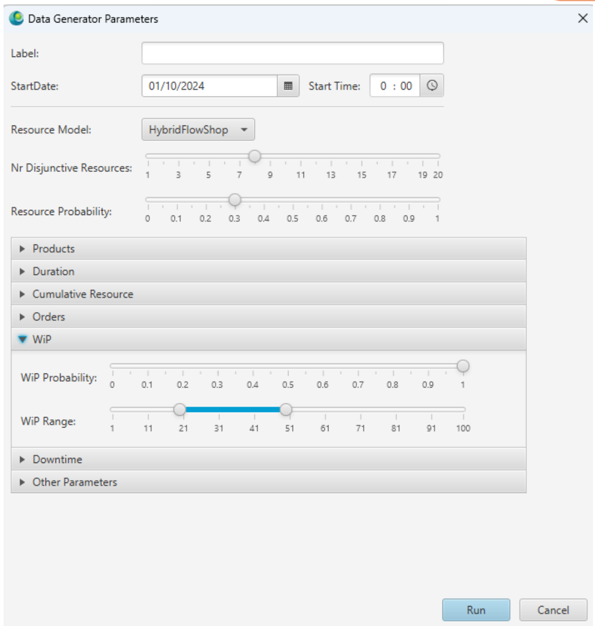
Instance Generator - Orders

- Nr Orders
  - Number of orders generated, each order is assigned a random product/process
- Qty Range
  - Range slider to select lower and upper bounds on quantity for each order



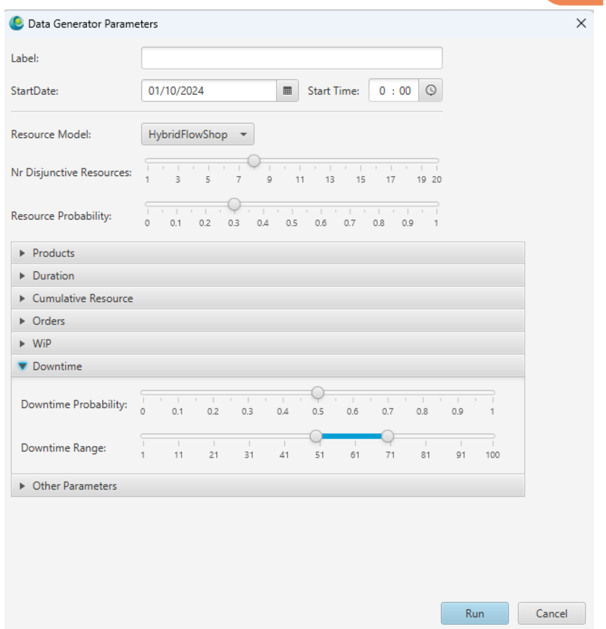
Instance Generator - WiP (Work in Progress)

- WiP Probability
  - Probability of generating a WiP for a disjunctive resource
- WiP Range
  - Range slider to set lower and upper bound on WiP duration



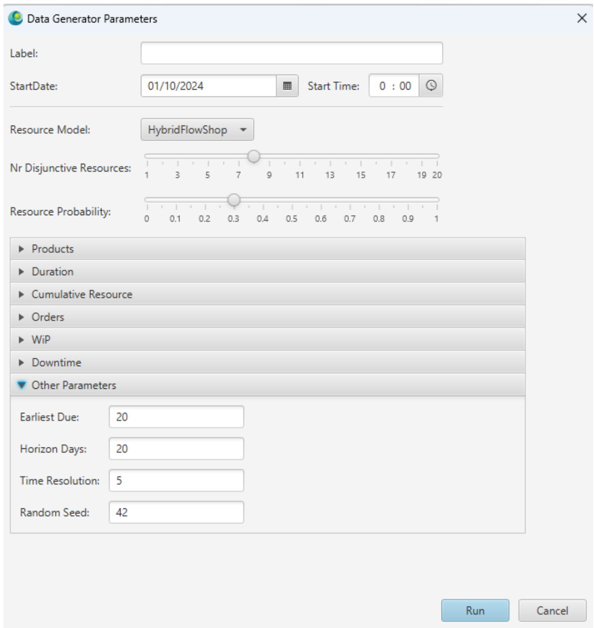
Instance Generator - Downtime

- Downtime Probability
  - Probability of generating a downtime for a disjunctive resource
- Downtime Range
  - Range slider to select lower and upper bounds on downtime duration



Instance Generator - Other Parameters

- Earliest Due
  - Smallest allowed value for a due date
- Horizon Days
  - What planning horizon to consider (in days)
- Time Resolution
  - In minutes, links internal and external time presentation
- Random Seed
  - Random seed to make reproducible random choices



Taillard Scheduling Benchmarks

- Three datasets of different sizes
  - Job-shop
  - Flow-shop
  - Open-shop
- Load with menu `File - Load DataFile... - Taillard -`
- Larger instances need more solver time to reach good solutions (600 s)

Simple Assembly Line Balancing Problem (SALBP)

- Will be discussed on more details as case study
- Design an assembly line setup by solving a scheduling problem
- Balance a set of operations across a number of stations of an assembly line
- Precedence graph is not a chain, can be very complex
- Specialized problem normally solved with specialized tools
- Load with menu `File - Load SALBP Problem...`

Test Scheduling Benchmark set from ABB

- Will be discussed in more details as case study
- Schedule a set of tests on a number of machines, minimizing total duration
- Single stage tests, possibly large number of resources
- Closely related to bin-packing
- Load with menu `File - Load Test Scheduling Problem...`

Experiment 1

- Start the application
  - Our running example will be automatically generated
- Look at the process diagram `Window-Product-Process Diagram`
- Run the solver `Scenario - Run ScheduleJobs Solver`
- Observe the results in Gantt Chart
- Customize display
- Look at Cumulative Resource Chart `Window-Solution-Cumulative Resource Chart`

Experiment 2

- Re-run solver disabling cumulative constraint
- Observe result in Gantt chart
- See impact on Cumulative Resource chart
- Switch between solutions in charts

Experiment 3

- Check Gantt chart display for delayed tasks, enabling lateness display
- Re-run solver, enforcing due-date constraints
- What impact does this have on objective

Experiment 4

- Change objective to on-time delivery
- Results are very different, why?
- More explanations on this tomorrow

Experiment 5

- Load one of the other example types
- For example, Taillard Job-shop 15x15
- Understand process diagram
- Run solver
- Look at intermediate solutions found

Summary

- We presented an overview of our generic scheduling tool
- Discussed available solvers, both commercial and open-source
- Described the JSON data format for input and output
- Gave an overview of the instance generator provided
- Shows example problems included with tool
- Suggested some experiments to run

Part V

Objectives

Key Points

- Why we search for good, but not always optimal solutions
- The different objectives provided in scheduling tool
- More complex optimization schemes involving multiple objectives
- Other criteria that might guide which solution we prefer
- An interesting research direction

Why have an Objective?

- For most scheduling problem, we define some form objective
- A mathematical formula that we evaluate on a schedule to compare it
- It is not always clear whether that formula represents some direct business benefit
- But, there are far more bad solutions than good solutions!
- The objective tells us if the solution is more "good" or "bad"
- Different stakeholders will have different views what makes a solution "good" or even "acceptable"

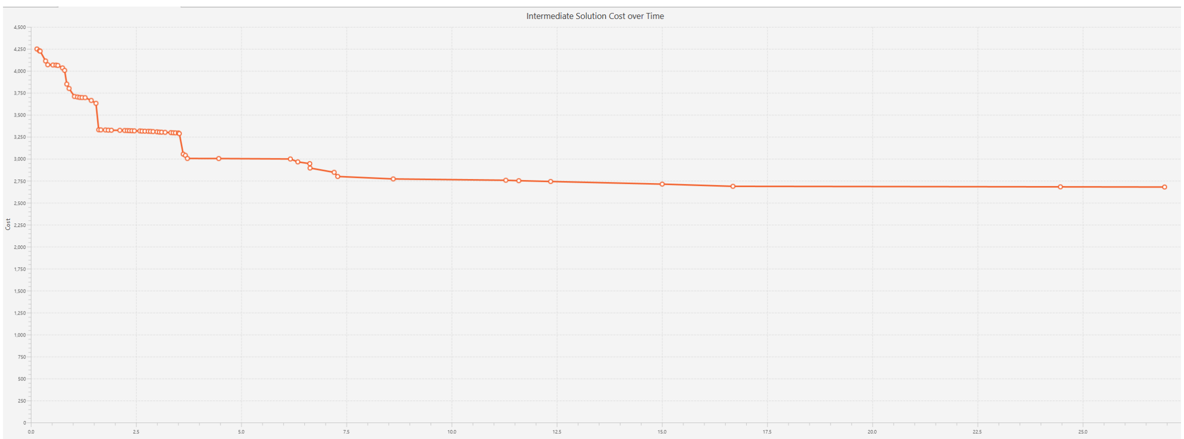
Minimizing Cost vs Maximizing Profit

- A lot of objectives aim to reduce cost of production
- This is not always a good thing
  - Doing nothing costs nothing
- But defining the profit obtained by a schedule is not easy
- Many intangible factors weigh in
  - Happiness of the customers (which customers are unhappy, does it matter?)
  - Happiness of personnel (finding and retaining skilled personnel is critical)
  - Happiness of stakeholders (sales, production, inventory, management)

Timeliness

- How quickly do we need a solution?
  - Sometimes we need a solution right now
  - We may also have time to wait a bit, or even more
  - Waiting five minutes, having a short break for a coffee, will often be acceptable
  - For some problems, running a scheduler overnight is possible
  - Do we need the ultimate in solution quality, or an acceptable solution right now?
- Benchmarks are often run with unlimited resources
  - "We used four years of computer time to solve these problems"

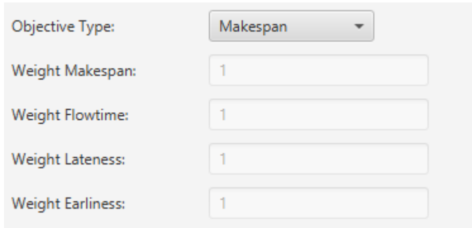
Diminishing Returns Running a Solver



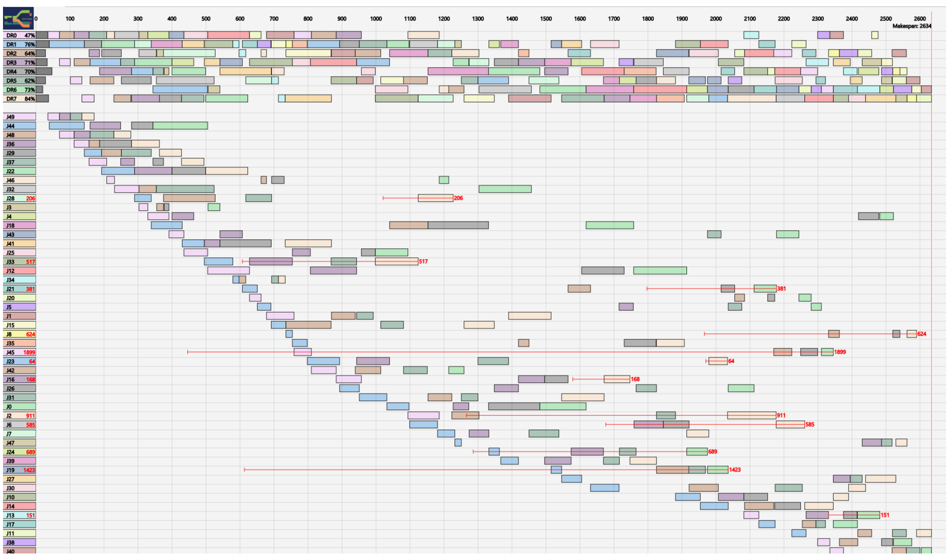
- Which compromise between quality and speed are we looking for?

Setting the Objective

- We can select a predefined objective in solver dialog
- There are weight factors to give more impact to some cost terms in on-time and hybrid objectives

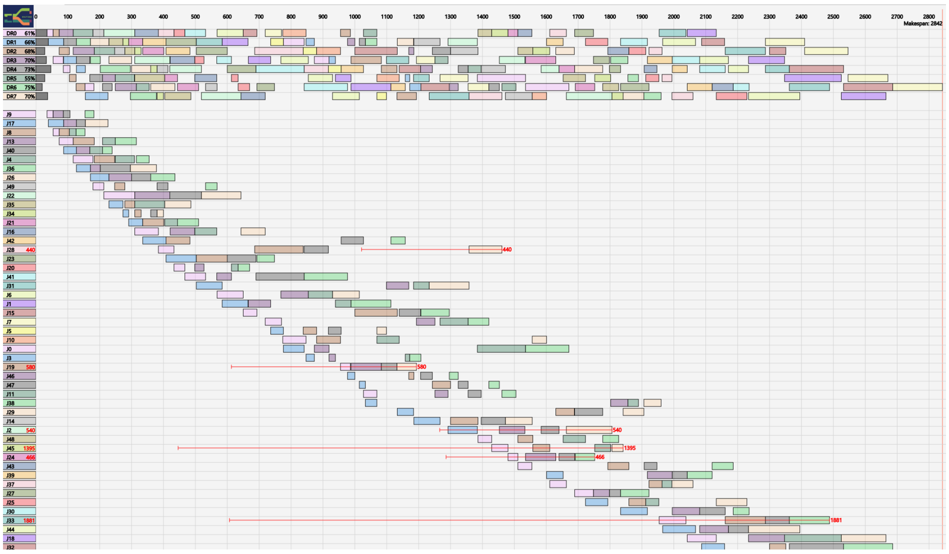


Makespan ✓



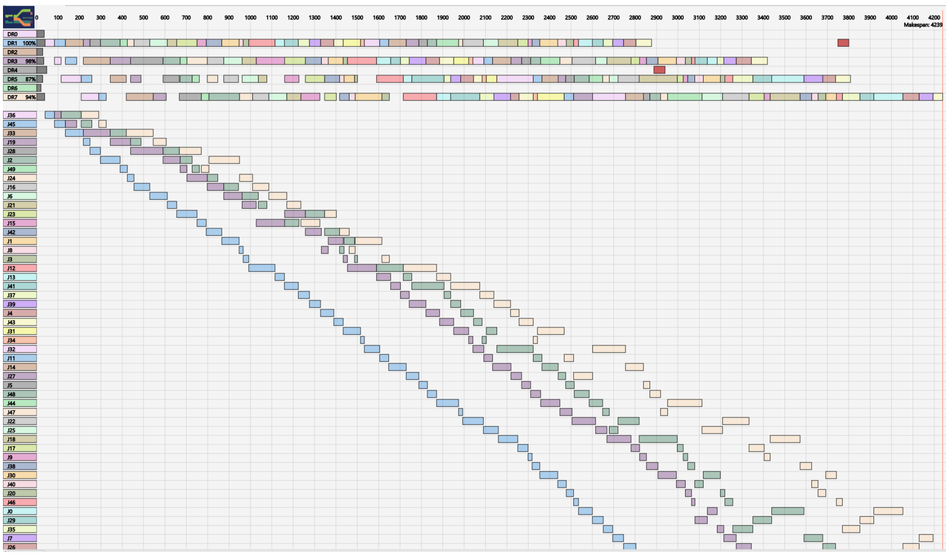
- Minimize the overall project end
- Very traditional objective in scheduling
    - Justified in project scheduling
    - Not so clearly justified in manufacturing
- A number of jobs are significantly late

Flowtime ✓



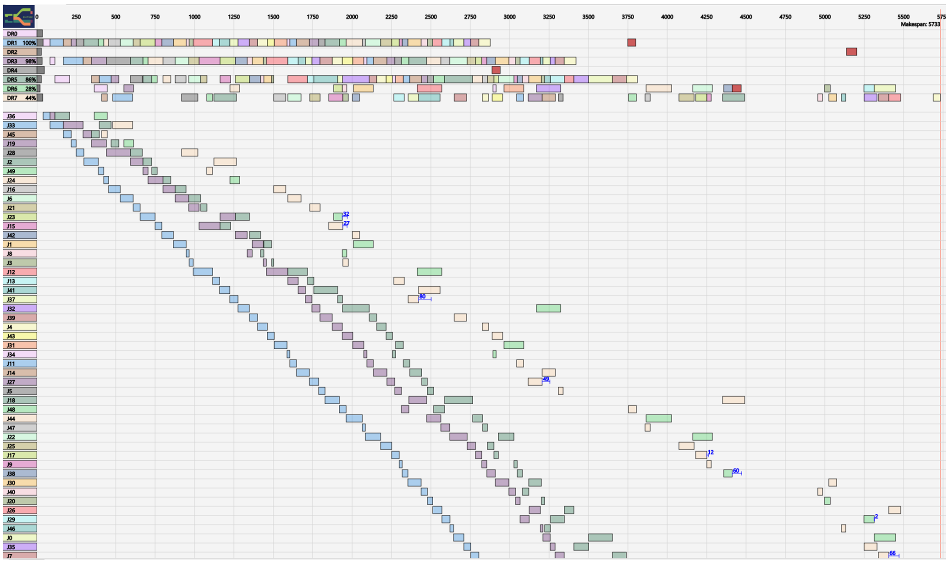
- Minimize the sum of job ends
- Prefer any machine to end early
- Not always easy to find good solutions

Total Lateness ✓



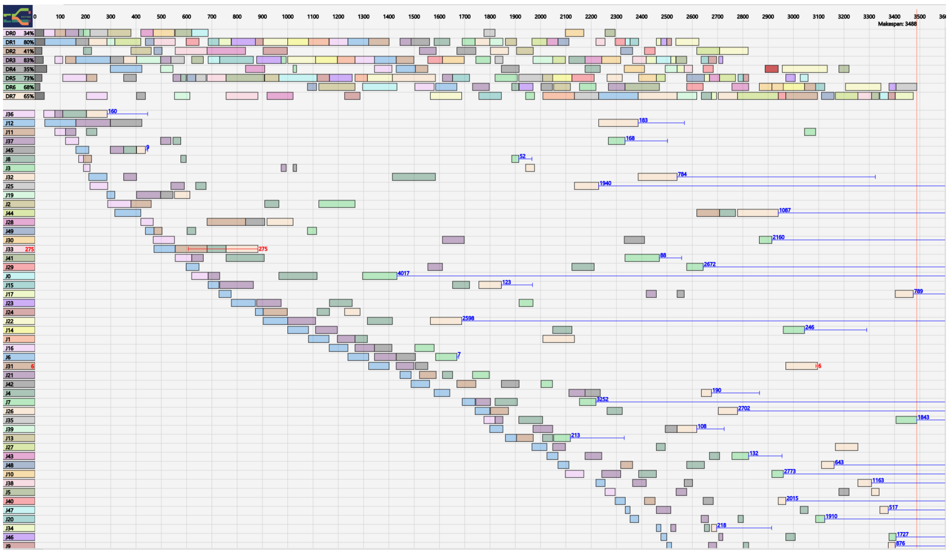
- Able to remove all delays on jobs
- Does not care about makespan or earliness

Maximizing On-Time Delivery ✓



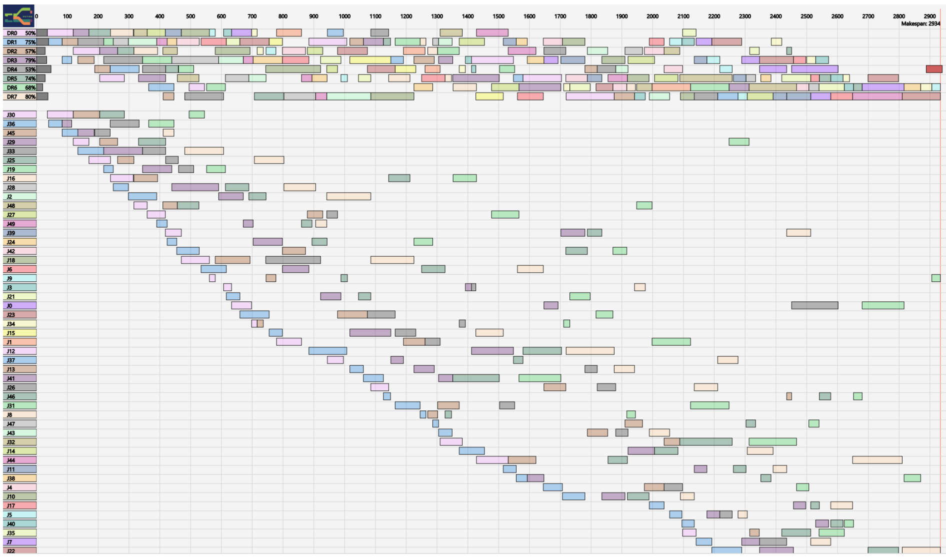
- Weight 100 for lateness, weight 1 for earliness
- Removes all delays, very little earliness
- Makespan increased dramatically

Hybrid Objective ✓



- Weights makespan:1000, flowtime:0, lateness:10, earliness:1
- Does not remove lateness completely
- Probably needs more time to improve

Hybrid Objective (Enforce Duedate)✓



- Sometimes enforcing a constraint is more powerful
- Here require that due dates are respected
- Leads to overall better solution

Comparing Solutions with Different Objectives

SolverRun	ObjectiveType	ObjectiveValue	SolverStatus	Bound	GapPercent	Makespan	Flowtime	TotalLateness	MaxLateness	NrLate	WeightedLateness	TotalEarliness	MaxEarliness	NrEarly	WeightedEarliness	PercentEarly	PercentLate
Run1	Makespan	2,634	Solution	1,050.00	60.14	2,634	86,339	7,618	1,899	12	7,618.00	76,688	4,887	38	76,688.00	76.00	24.00
Run2	Flowtime	66,356	Solution	39,248.00	40.85	2,842	66,356	5,575	1,881	7	5,575.00	94,628	5,045	43	94,628.00	86.00	14.00
Run3	TotalLateness	0	Optimal	0.00	NaN	4,239	119,745	0	0	0	0.00	35,664	1,494	50	35,664.00	100.00	0.00
Run4	OnTime	328	Optimal	328.00	0.00	5,733	155,081	0	0	0	0.00	328	80	8	328.00	16.00	0.00
Run5	Hybrid	3,554,610	Solution	1,150,697.00	67.63	3,488	117,180	281	275	2	281.00	38,510	4,017	34	38,510.00	68.00	4.00
Run6	Hybrid	2,992,627	Solution	1,155,981.00	61.37	2,934	96,782	0	0	0	0.00	58,627	4,530	43	58,627.00	86.00	0.00

- System tries to reduce the objective
- May mean other aspect of solution is poor
    - *Total Lateness* bad if just reducing *Makespan*
    - *Makespan* bad if just reducing *Total Lateness*
- Hybrid objectives can find better compromises
- Using constraints to restrict search can help
- Needs more work on lower bounds

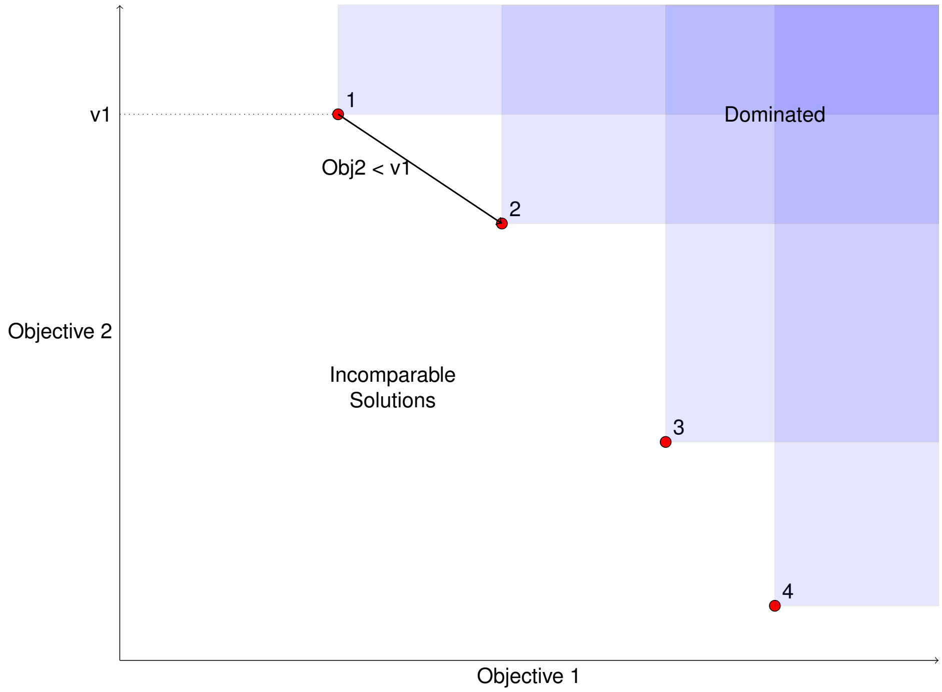
Optimizing Resource Levels ✗

- We have already discussed this in the Resources section
- Sometimes we aim to optimize resource use, not time or delay
- Typical is minimizing
  - The number of disjunctive machines needed
  - A cumulative resource capacity
  - The manpower required to perform all tasks
- We may do this for understanding the problem
- The optimized schedules will be brittle
  - Any resource breakdown will cause an issue
  - Spare capacity is a good thing (if it is not too expensive)

Multi-level Objectives ✗

- In some situations, a hybrid objective combining different aspects is not enough
- We need to find the best compromises between the different objective types
  - Without an a-priori weight to state which is more important
- A solution *dominates* another solution, if for all objective types, it is better than the other
- Two solutions are *incomparable* if for some objective type one solution is better, but for some other objective, the other solution is better
- *Pareto frontier*: Set of all non-dominated, incomparable solutions

Pareto Frontier for Two Incomparable Objectives



- Finding Pareto frontier by repeated optimization of objective1

Other Quality Vectors

- There are other scales on which we may measure whether a solution is "good"
  - Fairness
  - Robustness
  - Product Quality
  - Customer Satisfaction
  - Diversity

Fairness

- Typically involves humans
- If we assign operators, do we
  - Treat all operators in a fair way?
  - Give effective workers more work
  - Provide opportunities for training and skill development
  - De-risk dependency on key personnel
- Also, use multiple machines of same type consistently
  - Balanced
  - Not balanced

Robustness

- By scheduling, we create a plan
- Often, reality does not follow the plan
  - Unforeseen events, machine breakdowns, sick-leave
  - Delays in raw material delivery, inventory problems
  - Rush orders
  - Small variations in plan execution
- Can we protect the plan against certain types of unplanned events?
- Is the plan still useful when things change?
- Or, can we update the plan quickly enough to adapt to changes

Product Quality

- The tighter the schedule, the more risk there is of cutting corners
- If we minimize curing times to speed up production, quality may be affected
- The fastest machine is not always the best in terms of quality, cost

Customer Satisfaction

- Our objectives for minimizing lateness are lacking context
- Some customers are more important than others
- Some orders are more important to the customer than others
- A phone call by a human can capture more detail than an electronic order form
- We can adjust our schedule if we know what is important and what is not
  - But where do we get this information?
  - How do we avoid that a customer says "all my orders are critical"

Diversity

- Is it sometimes useful to present different solutions to a user to choose from
- These solutions should be substantially different to make choice meaningful
- Unfortunately, solvers often find very similar solutions
- Typically, there are far too many solution to enumerate them all
- We can add constraints to ask for the next solution to be quite different from the previous ones
- Needs good definition to define similarity
- Hamming distance on machine order a good starting point

Key Performance Indicators (KPI)

- Performance indicators can be computed from a given schedule, and allow to compare different schedules to each other
- Often, these are business oriented, not process driven
- There is a difference between an objective and a performance indicator
  - The objective drives the search for a solution
  - The KPI evaluates the quality of a solution, can be totally unrelated to objective
- Ideally, the KPI are expressed in such a way that solutions for different problems can be compared
  - Number of late orders, allows comparison of two solutions of the same problem
  - Percentage of late orders, allows comparison of two different schedules

KPIs for Sample Solutions

- Comparing different solutions of running example with enabling/disabling some constraints
- Compare *Makespan* to *On-time Delivery* objective
- There is no *Setup Time* constraint specified for this problem

Makespan	Flowtime	TotalLateness	MaxLateness	NrLate	WeightedLateness	TotalEarliness	MaxEarliness	NrEarly	WeightedEarliness	PercentEarly	PercentLate	Duration	Start	End
2,688	83,425	10,083	1,959	11	10,083.00	82,067	4,938	39	82,067.00	78.00	22.00	2,653	35	2,688
2,690	85,051	0	0	0	0.00	70,358	4,133	50	70,358.00	100.00	0.00	2,655	35	2,690
2,136	58,403	0	0	0	0.00	97,006	4,956	50	97,006.00	100.00	0.00	2,101	35	2,136
2,324	62,494	0	0	0	0.00	92,915	4,751	50	92,915.00	100.00	0.00	2,289	35	2,324
5,733	154,918	0	0	0	0.00	491	122	10	491.00	20.00	0.00	5,538	195	5,733

TotalWaitBefore	TotalWaitAfter	MaxWaitBefore	MaxWaitAfter	TotalIdleBefore	TotalIdleAfter	MaxIdleBefore	MaxIdleAfter	TotalSetupBefore	TotalSetupAfter	MaxSetupBefore	MaxSetupAfter	TotalActiveTime	TotalProductionTime	ActiveUtilization	SetupPercent	IdlePercent
23,297	23,297	1,943	1,943	6,823	6,823	435	435	0	0	0	0	19,917	13,094	65.74	0.00	34.26
24,903	24,903	1,611	1,611	5,901	5,901	342	342	0	0	0	0	18,995	13,094	68.93	0.00	31.07
12,081	12,081	449	449	785	785	80	80	0	0	0	0	13,879	13,094	94.34	0.00	5.66
0	0	0	0	4,211	4,211	111	111	0	0	0	0	17,305	13,094	75.67	0.00	24.33
0	0	0	0	28,641	28,641	773	773	0	0	0	0	41,735	13,094	31.37	0.00	68.63

KPIs Already Defined ✓

Makespan Max of job ends

Flowtime Sum of job ends

Total Lateness Sum of job lateness (tardiness)

Max Lateness Max of job lateness

NrLate Number of late jobs

WeightedLateness Weighted sum of job lateness

PercentLate percentage of late jobs

...Earliness same indicators, but for earliness

Duration Difference between overall start and overall end

Start start of earliest job

End end of last job

KPIs Already Defined (cont'd) ✓

TotalWait Sum of Wait time before/after a task of a job

MaxWait Max wait time before/after a task of a job

TotalIdle Sum of Idle times of disjunctive machines

MaxIdle Max Idle Time on a disjunctive machine

TotalSetup Total setup times

MaxSetup Max setup time

TotalActiveTime Total active time between first and last use of a machine

TotalProductionTime Sum of all task duration

ActiveUtilization Percentage of production time compared to active time

SetupPercent Percentage of setup time compared to active time

IdlePercent Percentage of idle time compared to active time

KPI Ranking ✗

- If we have multiple solutions, we want to rank them based on a comparison of different KPIs
- Different stakeholders will rank different KPIs in very different way
- This seems to require some customization of the formulas used
- We can also try to infer a ranking method based on some comparison queries asked to users
  - Do you prefer this or that solution?
  - With enough answers, we can postulate a ranking method

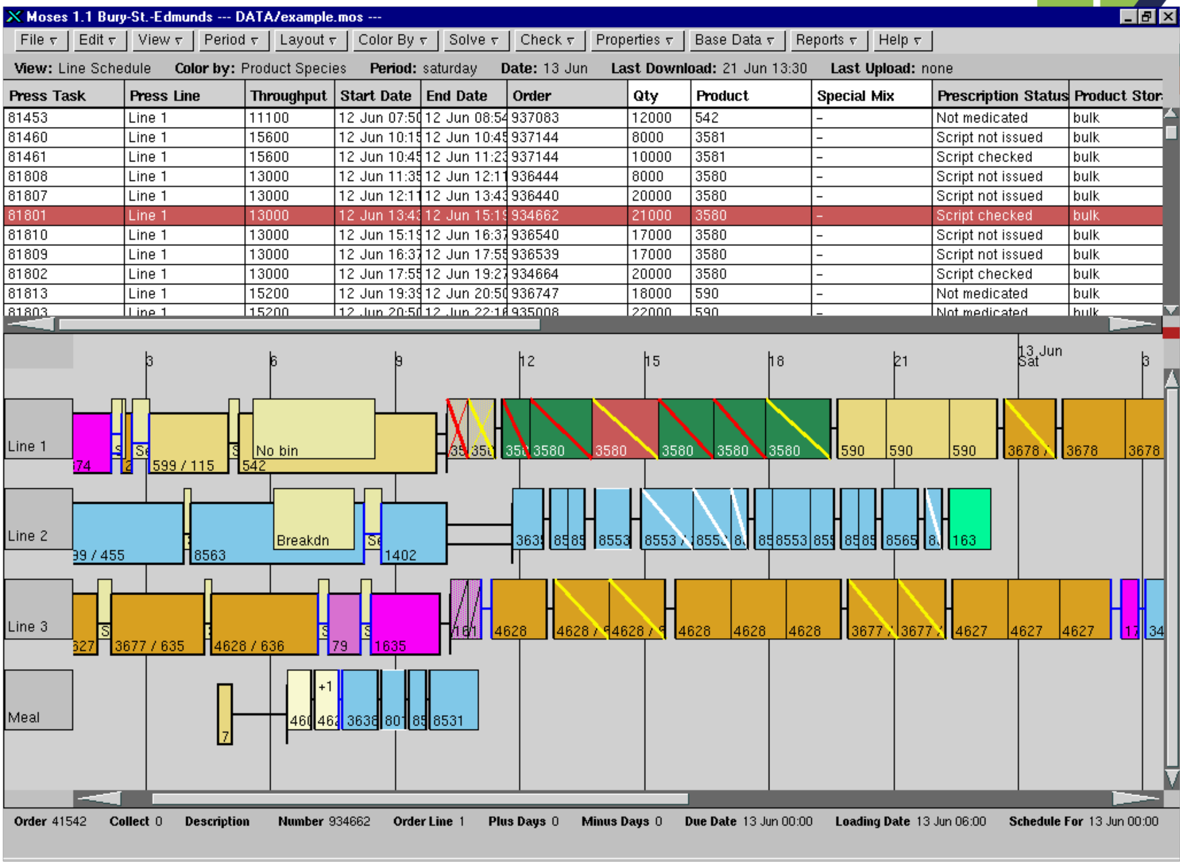
Interactive Scheduling ✗

- Some human schedulers are happy to accept a produced plan
  - Perhaps change some constraints, or weights
- Other human schedulers want to modify the plan by hand
  - This is not always easy to do
  - How can a scheduling tool handle this?
  - How much control is given to the user, who checks the constraints?
  - Do we allow the user to create invalid schedules?

Example: Moses System

- Scheduling application for animal feed mills in the UK
- Produces overnight schedule for delivery on next day
- Operator updates the schedule whenever a task is finished
- Change duration of task if it is delayed
- Move tasks by hand, changing sequence of tasks to be performed
  - System updates constraints, and warns if constraint is violated
- User can protect part of schedule from modification by system
  - Freeze all tasks up to the selected task
  - Unfreeze the schedule after the selected task
- Related to explainability

Screenshot of Moses Application



Summary

- Describe the need and role of objectives
- Presented different objectives available in the scheduling tool
- Discussed some more advanced possibilities for handling objectives
- Important to keep user on control of system

Part VI

Advanced Concepts

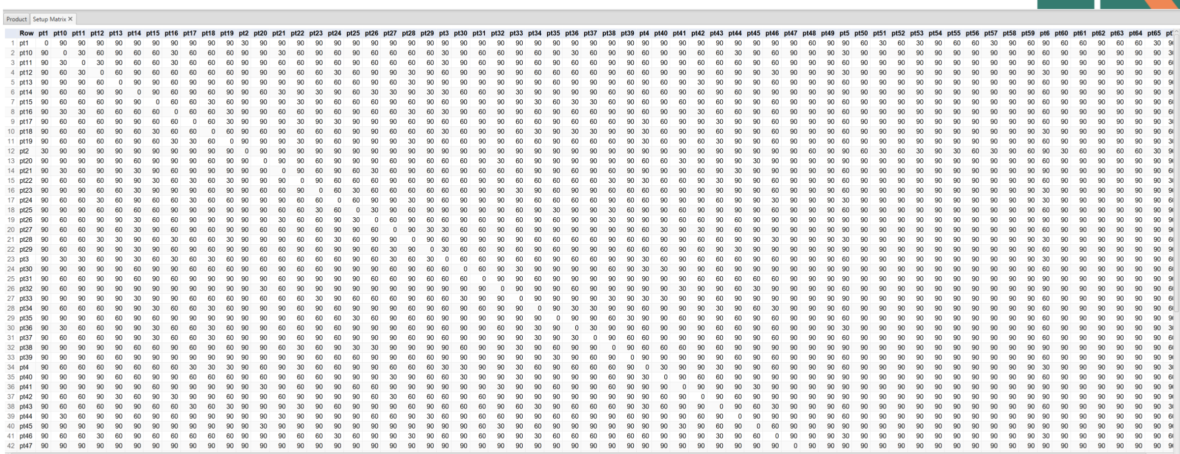
Key Points

- We present some more advanced concepts in scheduling
- These occur in more specialized problem areas
- Typically require more work on modelling
- Solver support may be limited

Sequence Dependent Setup-Time ✓

- Our usual disjunctive resource model assumes we can change easily from one task to the next
- There might be a cleaning/setup time required
  - This is part of the fixed duration part of a processStep description
- In some cases it is more complex
  - On some machines there is a setup-time required which depends on both the previous and the next product
  - This time varies significantly between product combinations
  - Typically, the time depends on some properties of the products
- The setup time is non-productive, and should be avoided when possible

Computed Setup-Time Matrix



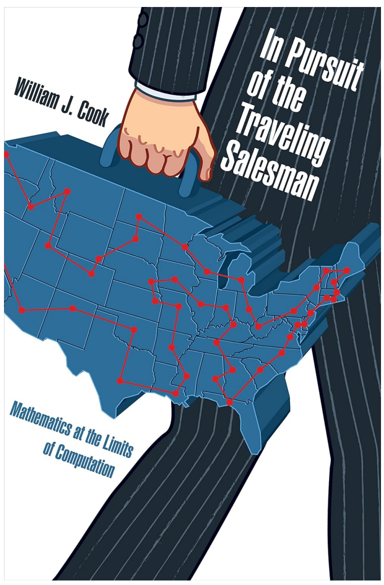
- This needs to be computed from first principles, not maintained by hand!
- Available as input data in JSON format

Relation to TSP

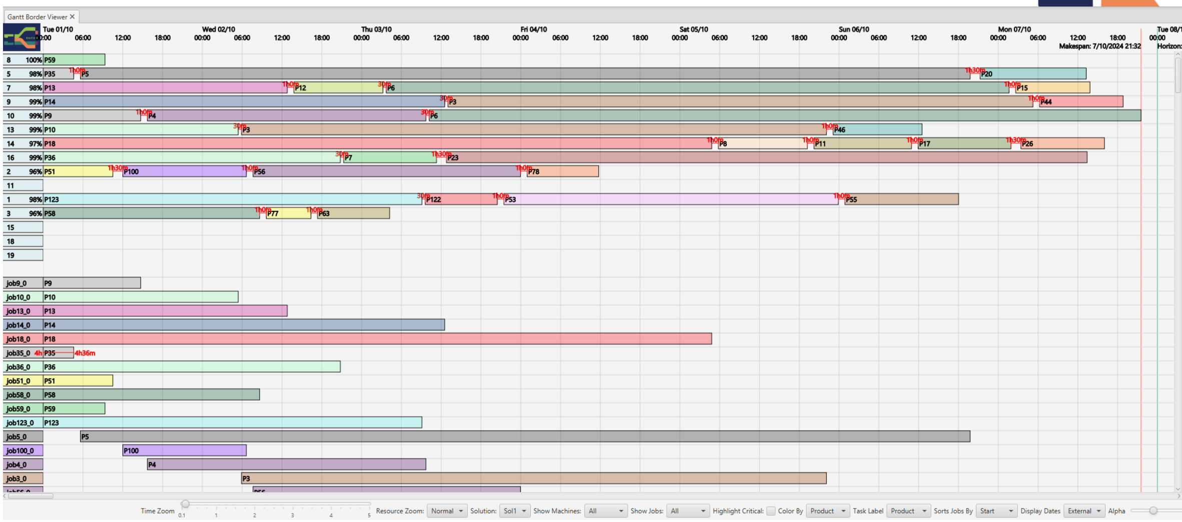
- Computing the optimal sequence of setup times is a variant of the *Travelling Salesman Problem (TSP)*
- Another of the classical hard combinatorial problems
- Due to the structure of the data, setup-time problem often are simpler to solve
  - Changing between very similar products needs no setup-time
  - Using a simple rule about product compatibility produces best results
  - Example: dark-chocolate → milk chocolate → white chocolate → milk chocolate → draw chocolate
- Problems get more difficult when release/due dates need to be respected
- This is the equivalent to the *VRPTW (Vehicle Routing Problem with Time Windows)*

Xmas Shopping Hint

- W. Cook. In Pursuit of the Travelling Salesman. Princeton University Press, 2011
- Entertaining general science presentation of the TSP and related issues



Setup Times Constraints can be Included in Model



- Shown in Machine Gantt chart, enable display in `Layout` tab
- So far, only in CPO, not in CPSat model

Related Problem: Forbidden Transitions ✗

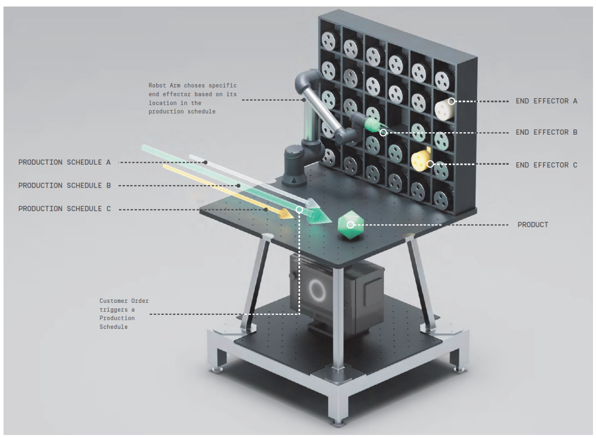
- For safety reasons, it may be forbidden to change from some product to some specific other products
- Contamination risk is considered too high
- Examples
    - In food production: Is this product peanut free?
    - In food production: Directly changing from dark to white chocolate is not allowed
    - In chemical plants: Contamination may lead to explosions
- These transitions are called *forbidden*, and must be avoided
- Careful, it is easy to paint yourself into a corner!

Dealing with Transportation Times

- Really two different problems
  - In one, the resources are in fixed locations, and we transport the jobs between the locations
  - In the other, the tasks are in fixed locations, and we transport the resources between them

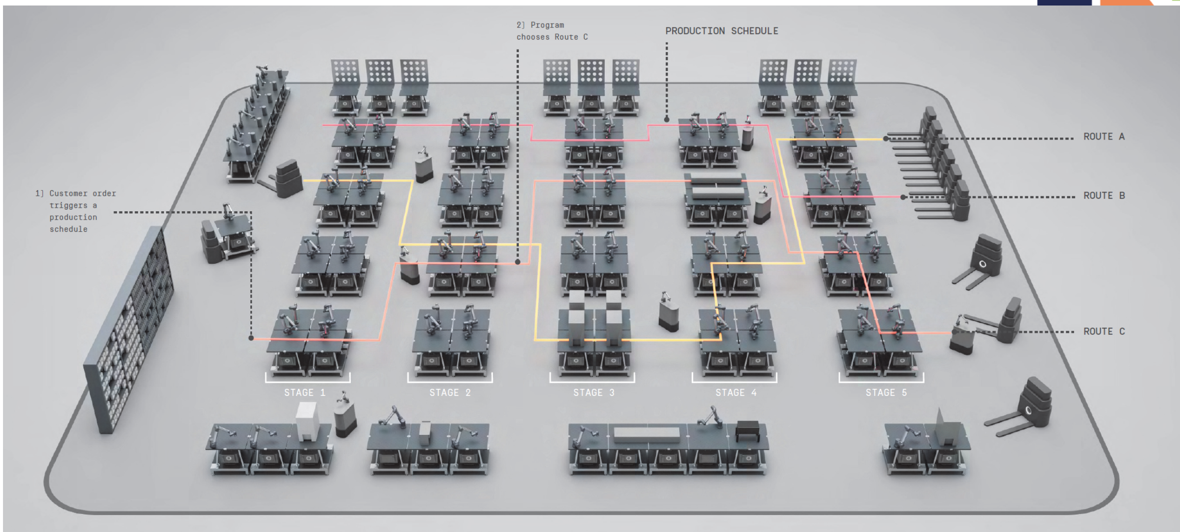
Transportation of Jobs

- Example from a project with J&J in Limerick
- Considering a *factory of the future* based on agile machines
- Robots that can be configured to perform many different tasks
- These robots may be inside one or more factories
- How to arrange them to minimize impact of transport on production



from J&J

Layout of Factor in Matrix Form



- Materials are transported between stations by moving robots
- Layout of factory determines delay caused by transport

Inclusion in Model (✓)

	Row	M0_0	M0_1	M0_2	M0_3	M0_4	M0_5	M0_6	M0_7	M0_8	M0_9	M1_0	M1_1	M1_2	M1_3	M1_4	M1_5	M1_6	M1_7	M1_8	M1_9	M2_0	M2_1	M2_2	M2_3	M2_4	M2_5	M2_6	M2_7	M2_8	M2_9	M3_0	M3_1	M3_2	M3_3	M3_4	M3_5	M3_6	M3_7	M3_8	M3_9
1	M0_0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	M0_1	0	0	0	0	0	0	0	0	0	0	2	1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	M0_2	0	0	0	0	0	0	0	0	0	0	3	2	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	M0_3	0	0	0	0	0	0	0	0	0	0	4	3	2	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	M0_4	0	0	0	0	0	0	0	0	0	0	5	4	3	2	1	2	3	4	5	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	M0_5	0	0	0	0	0	0	0	0	0	0	6	5	4	3	2	1	2	3	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	M0_6	0	0	0	0	0	0	0	0	0	0	7	6	5	4	3	2	1	2	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	M0_7	0	0	0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	1	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	M0_8	0	0	0	0	0	0	0	0	0	0	9	8	7	6	5	4	3	2	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	M0_9	0	0	0	0	0	0	0	0	0	0	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	M1_0	1	2	3	4	5	6	7	8	9	10	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10	0	0	0	0	0	0	0	0	0	0
12	M1_1	2	1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	0	2	1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	0
13	M1_2	3	2	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	0	0	3	2	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	0	0
14	M1_3	4	3	2	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	0	0	4	3	2	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	0	0
15	M1_4	5	4	3	2	1	2	3	4	5	6	0	0	0	0	0	0	0	0	0	0	5	4	3	2	1	2	3	4	5	6	0	0	0	0	0	0	0	0	0	0
16	M1_5	6	5	4	3	2	1	2	3	4	5	0	0	0	0	0	0	0	0	0	0	6	5	4	3	2	1	2	3	4	5	0	0	0	0	0	0	0	0	0	0
17	M1_6	7	6	5	4	3	2	1	2	3	4	0	0	0	0	0	0	0	0	0	0	7	6	5	4	3	2	1	2	3	4	0	0	0	0	0	0	0	0	0	0
18	M1_7	8	7	6	5	4	3	2	1	2	3	0	0	0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	1	2	3	0	0	0	0	0	0	0	0	0	0
19	M1_8	9	8	7	6	5	4	3	2	1	2	0	0	0	0	0	0	0	0	0	0	9	8	7	6	5	4	3	2	1	2	0	0	0	0	0	0	0	0	0	0
20	M1_9	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
21	M2_0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10
22	M2_1	0	0	0	0	0	0	0	0	0	0	2	1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	0	2	1	2	3	4	5	6	7	8	9
23	M2_2	0	0	0	0	0	0	0	0	0	0	3	2	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	0	0	3	2	1	2	3	4	5	6	7	8
24	M2_3	0	0	0	0	0	0	0	0	0	0	4	3	2	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	0	0	4	3	2	1	2	3	4	5	6	7
25	M2_4	0	0	0	0	0	0	0	0	0	0	5	4	3	2	1	2	3	4	5	6	0	0	0	0	0	0	0	0	0	0	5	4	3	2	1	2	3	4	5	6
26	M2_5	0	0	0	0	0	0	0	0	0	0	6	5	4	3	2	1	2	3	4	5	0	0	0	0	0	0	0	0	0	0	6	5	4	3	2	1	2	3	4	5
27	M2_6	0	0	0	0	0	0	0	0	0	0	7	6	5	4	3	2	1	2	3	4	0	0	0	0	0	0	0	0	0	0	7	6	5	4	3	2	1	2	3	4
28	M2_7	0	0	0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	1	2	3	0	0	0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	1	2	3
29	M2_8	0	0	0	0	0	0	0	0	0	0	9	8	7	6	5	4	3	2	1	2	0	0	0	0	0	0	0	0	0	0	9	8	7	6	5	4	3	2	1	2
30	M2_9	0	0	0	0	0	0	0	0	0	0	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	10	9	8	7	6	5	4	3	2	1
31	M3_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	10	0	0	0	0	0	0	0	0	0	0
32	M3_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	0
33	M3_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	1	2	3	4	5	6	7	8	0	0	0	0	0	0	0	0	0	0
34	M3_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3	2	1	2	3	4	5	6	7	0	0	0	0	0	0	0	0	0	0
35	M3_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	4	3	2	1	2	3	4	5	6	0	0	0	0	0	0	0	0	0	0
36	M3_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	5	4	3	2	1	2	3	4	5	0	0	0	0	0	0	0	0	0	0
37	M3_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	6	5	4	3	2	1	2	3	4	0	0	0	0	0	0	0	0	0	0
38	M3_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	1	2	3	0	0	0	0	0	0	0	0	0	0
39	M3_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	8	7	6	5	4	3	2	1	2	0	0	0	0	0	0	0	0	0	0
40	M3_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0
41	M4_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8	9	10
42	M4_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	3	4	5	6	7	8	9	10	2	1	2	3	4	5	6	7	8	9
43	M4_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3	2	3	4	5	6	7	8	9	3	2	1	2	3	4	5	6	7	8
44	M4_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	4	3	2	3	4	5	6	7	8	4	3	2	1	2	3	4	5	6	7
45	M4_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	5	4	3	2	3	4	5	6	7	5	4	3	2	1	2	3	4	5	6
46	M4_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	6	5	4	3	2	3	4	5	6	6	5	4	3	2	1	2	3	4	5
47	M4_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	7	6	5	4	3	2	3	4	5	7	6	5	4	3	2	1	2	3	4
48	M4_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	8	7	6	5	4	3	2	3	4	8	7	6	5	4	3	2	1	2	3
49	M4_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	9	8	7	6	5	4	3	2	3	9	8	7	6	5	4	3	2	1	2
50	M4_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	10	9	8	7	6	5	4	3	2	10	9	8	7	6	5	4	3	2	1

- Add location attribute to each resource
- Include transport time as element in temporal constraints

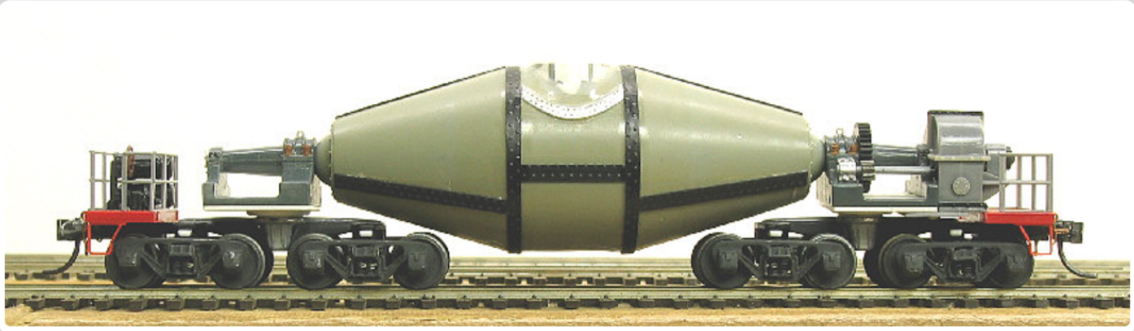
More Complex Variant

- Schedule the moving robots as well
- Assume that an empty robot travels much faster than a loaded one
- We can treat the robots as a machine choice resource for the transportation tasks

Even More Complex Variant

- Schedule the moving robots as well
- They move at the same speed empty and loaded
- We can to bring them from the end of one transport task to the start of the next one
- This is a vehicle routine problem
- In some industries, this is the harder problem compared to scheduling the plant itself
  - Torpedo scheduling in steel plant: rail cars holding molten steel, quantities limited

Torpedo Scheduling (CP 2016 Challenge)

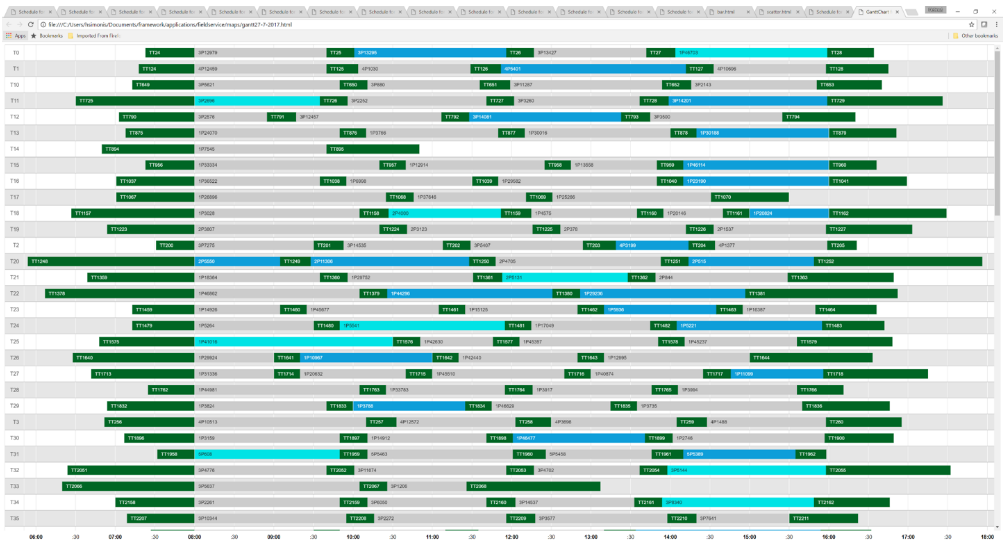


(from ACP Website `http://cp2016.a4cp.org/program/acp-challenge/`)

Scheduling Service Visits

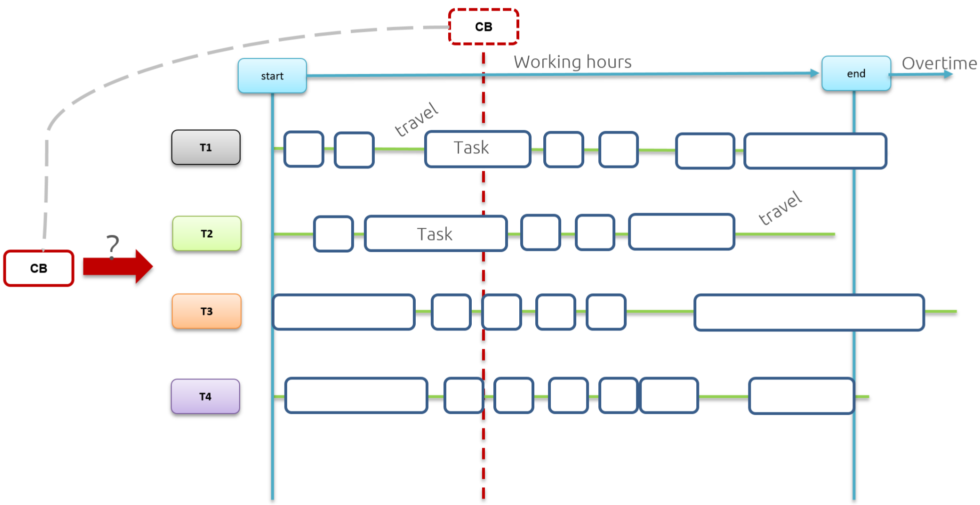
- Based on a project with UTRC-I, UTRC, OTIS
- Schedule visits to maintain equipment installed in customer premises
- Resources are the service engineers
- They have to travel between locations and perform work there
- The tasks are the maintenance operations required to keep equipment working
- Also called *Traveling Repair-person Problem*

Planning Maintenance Visits for Service Personnel



- Include single day trips, multi-day tours
- Most of the time spent at customer locations

Re-scheduling Problem



- How to react when a customer is trapped in an elevator
- All you engineers are on service calls
- Who you gonna call?

Advertisement

- This will be described in more detail in a new course
- AI Fundamentals: Skill Development Program on Transportation Optimization
- Arriving in 2025 at this location

Summary

- We presented some more advanced topics
  - Sequence dependent setup
  - Transportation time
- Not available in every solver
- Useful concepts when dealing with specific scheduling problems
- Leading to another *Skills Development Program*

Part VII

Case Studies

Key Points

- We provide a number of scheduling case studies
- Use the methodology developed to describe problems
- Use scheduling tool to provide solutions
- Generic tool provides good, but not always best solutions
- Two case studies are not handled by scheduling tool (yet)

Case Studies Overview

- Production Planning and Detailed Scheduling
  - How to use detailed scheduling in a wider context
- Assembly Line Balancing
  - Scheduling to plan design of an assembly line
- Test Scheduling
  - Scheduling tests on resources
- Factory Design
  - Location of resources affects scheduling outcome
- Oven Scheduling
  - Solving one detailed scheduling problem is not enough
- Blades and Vanes
  - Capacity and production planning over a multi-year period

Summary

- See how the methodology can be applied to solve real-world problems
- Generic tool provides immediate solution of good quality
- Visualization of results is also provided
- Tool will be available in a few weeks time

Part VIII

Production Planning Case Study

Key Points

- Case study from industry
- Production planning and detailed scheduling
- Based on project with medical devices company in Cork
    - Real problem
    - Realistic data
- Solved in two stages
    - Production planning based on run-out days and safety stock levels
    - Scheduling using our generic scheduling tool

Problem

This is a case study from industry, combining production planning and detailed scheduling. It is based on a project that some of my colleagues were working on for a local medical devices company.

The overall problem is to decide which products to make in which quantities over the planning horizon, so that we have enough stock to satisfy any customer demand, and make sure that we have some safety margin if the demand suddenly increases. At the same time we do not want to create inventory in products that we will not sell in the near future, as this increases our inventory carrying cost.

The company uses two main concepts for production planning: The run-out days for each product state how long the current stock will last, given a projected customer demand profile. We try to achieve the same run-out days value for all products, this works well for fast and slow moving products.

The safety stock values says how much stock we should have for each product. This gives us more control over the stock levels, this works better if the demand cannot be predicted as accurately as we would like, but it is more difficult to compare the stock levels for different products.

The production planning part of the application decides how much to produce for which product, but this is based on an estimate of the production capacity for the planning period. We use the detailed scheduling part of the application to validate the plan generated, and make sure that we can really produce the required capacities in the given planning period.

Product List

Name	ShortName	Nr	DailySales	InventoryAtStart	CalcDaysCover	LotSize	CycleTime	LotDuration	Machine	ProductType	SafetyStock	SafetyAlert
P1	P1	1	3.20	877	274.06	163	1.33	217	8	pt1	66	253.44
P2	P2	2	11.40	1,011	88.68	240	1.20	288	8	pt2	774	20.79
P3	P3	3	796.20	26,204	32.91	420	2.10	882	5,7,9,10,13,14,16	pt3	12,108	17.70
P4	P4	4	233.80	7,877	33.69	420	2.00	840	5,7,9,10,13,14,16	pt4	3,358	19.33
P5	P5	5	267.30	7,152	26.76	350	2.30	805	5,7,9,10,13,14,16	pt5	3,906	12.14
P6	P6	6	606.20	18,654	30.77	350	2.30	805	5,7,9,10,13,14,16	pt6	9,293	15.44
P7	P7	7	137.30	4,939	35.97	420	2.00	840	5,7,9,10,13,14,16	pt7	1,979	21.56
P8	P8	8	88.30	3,152	35.70	350	2.30	805	5,7,9,10,13,14,16	pt8	1,342	20.50
P9	P9	9	77.20	2,688	34.82	420	2.10	882	5,7,9,10,13,14,16	pt9	1,082	20.80
P10	P10	10	165.60	5,971	36.06	420	2.10	882	5,7,9,10,13,14,16	pt10	2,649	20.06
P11	P11	11	60.70	2,310	38.06	420	2.10	882	5,7,9,10,13,14,16	pt11	877	23.61
P12	P12	12	51.80	1,928	37.22	350	2.30	805	5,7,9,10,13,14,16	pt12	883	20.17
P13	P13	13	79.00	2,231	28.24	320	2.30	736	5,7,9,10,13,14,16	pt13	1,193	13.14
P14	P14	14	271.20	8,951	33.01	432	2.10	908	5,7,9,10,13,14,16	pt14	3,732	19.24
P15	P15	15	86.60	3,244	37.46	336	2.00	672	5,7,9,10,13,14,16	pt15	1,454	20.67
P16	P16	16	42.40	2,110	49.76	420	2.10	882	5,7,9,10,13,14,16	pt16	875	29.13
P17	P17	17	17.60	681	38.69	420	2.00	840	5,7,9,10,13,14,16	pt17	290	22.22
P18	P18	18	217.50	5,710	26.25	336	2.00	672	5,7,9,10,13,14,16	pt18	2,814	13.31
P19	P19	19	56.30	2,450	43.52	420	2.00	840	5,7,9,10,13,14,16	pt19	804	29.24
P20	P20	20	13.60	506	37.21	480	2.00	960	5,7,9,10,13,14,16	pt20	272	17.21
P21	P21	21	10.80	977	90.46	360	2.10	756	5,7,9,10,13,14,16	pt21	293	63.33
P22	P22	22	21.80	1,538	70.55	420	2.00	840	5,7,9,10,13,14,16	pt22	349	54.54
P23	P23	23	189.10	5,195	27.47	360	2.30	828	5,7,9,10,13,14,16	pt23	2,941	11.92
P24	P24	24	9.50	886	93.26	350	2.30	805	5,7,9,10,13,14,16	pt24	191	73.16
P25	P25	25	7.50	326	43.47	120	2.30	276	5,7,9,10,13,14,16	pt25	210	15.47
P26	P26	26	11.60	418	36.03	360	2.10	756	5,7,9,10,13,14,16	pt26	187	19.91
P27	P27	27	16.50	1,388	84.12	480	2.10	1,008	5,7,9,10,13,14,16	pt27	218	70.91

Product List (Sorted by Daily Sales)

Product ✕

Name	ShortName	Nr	DailySales	InventoryAtStart	CalcDaysCover	LotSize	CycleTime	LotDuration	Machine	ProductType	SafetyStock	SafetyAlert
P3	P3	3	796.20	26,204	32.91	420	2.10	882	5,7,9,10,13,14,16	pt3	12,108	17.70
P6	P6	6	606.20	18,654	30.77	350	2.30	805	5,7,9,10,13,14,16	pt6	9,293	15.44
P14	P14	14	271.20	8,951	33.01	432	2.10	908	5,7,9,10,13,14,16	pt14	3,732	19.24
P53	P53	53	267.70	8,264	30.87	504	1.20	605	1,2,3,8	pt2	3,734	16.92
P5	P5	5	267.30	7,152	26.76	350	2.30	805	5,7,9,10,13,14,16	pt5	3,906	12.14
P124	P124	124	242.70	16,503	68.00	240	5.00	1,200	15,18,19	pt65	3,595	53.19
P4	P4	4	233.80	7,877	33.69	420	2.00	840	5,7,9,10,13,14,16	pt4	3,358	19.33
P123	P123	123	223.40	7,600	34.02	490	2.33	1,142	1,2,3,8	pt51	3,738	17.29
P18	P18	18	217.50	5,710	26.25	336	2.00	672	5,7,9,10,13,14,16	pt18	2,814	13.31
P23	P23	23	189.10	5,195	27.47	360	2.30	828	5,7,9,10,13,14,16	pt23	2,941	11.92
P56	P56	56	168.20	4,824	28.68	504	1.20	605	1,2,3,8	pt2	2,660	12.87
P10	P10	10	165.60	5,971	36.06	420	2.10	882	5,7,9,10,13,14,16	pt10	2,649	20.06
P59	P59	59	152.80	5,666	37.08	420	1.33	559	1,2,3,8	pt51	3,095	16.83
P7	P7	7	137.30	4,939	35.97	420	2.00	840	5,7,9,10,13,14,16	pt7	1,979	21.56
P57	P57	57	134.80	5,358	39.75	588	1.10	647	1,2,3,8	pt53	2,294	22.73
P36	P36	36	133.50	3,895	29.18	336	2.00	672	5,7,9,10,13,14,16	pt36	2,057	13.77
P54	P54	54	122.40	5,059	41.33	480	1.33	639	1,2,3,8	pt51	1,965	25.28
P121	P121	121	98.10	4,334	44.18	588	1.10	647	1,2,3,8	pt53	1,524	28.64
P8	P8	8	88.30	3,152	35.70	350	2.30	805	5,7,9,10,13,14,16	pt8	1,342	20.50
P125	P125	125	86.90	8,593	98.88	240	5.00	1,200	15,18,19	pt65	1,022	87.12
P15	P15	15	86.60	3,244	37.46	336	2.00	672	5,7,9,10,13,14,16	pt15	1,454	20.67
P100	P100	100	85.20	2,665	31.28	420	1.33	559	1,2,3,8	pt56	1,115	18.19
P55	P55	55	79.50	2,876	36.18	441	2.33	1,028	1,2,3,8	pt52	1,367	18.98
P13	P13	13	79.00	2,231	28.24	320	2.30	736	5,7,9,10,13,14,16	pt13	1,193	13.14
P9	P9	9	77.20	2,688	34.82	420	2.10	882	5,7,9,10,13,14,16	pt9	1,082	20.80
P47	P47	47	74.60	5,391	72.27	160	6.84	1,095	2,11	pt47	1,132	57.09
P11	P11	11	60.70	2,310	38.06	420	2.10	882	5,7,9,10,13,14,16	pt11	877	23.61
P61	P61	61	60.30	2,758	45.74	490	1.33	652	1,2,3,8	pt56	1,073	27.94
P78	P78	78	57.60	2,234	38.78	588	1.10	647	1,2,3,8	pt59	824	24.48
P19	P19	19	56.30	2,450	43.52	420	2.00	840	5,7,9,10,13,14,16	pt19	804	29.24

Product List (Sorted by Days Cover)

Product ✕

Name	ShortName	Nr	DailySales	InventoryAtStart	CalcDaysCover	LotSize	CycleTime	LotDuration	Machine	ProductType	SafetyStock	SafetyAlert
P35	P35	35	1.30	26	20.00	120	2.30	276	5,7,9,10,13,14,16	pt35	33	0.00
P18	P18	18	217.50	5,710	26.25	336	2.00	672	5,7,9,10,13,14,16	pt18	2,814	13.31
P5	P5	5	267.30	7,152	26.76	350	2.30	805	5,7,9,10,13,14,16	pt5	3,906	12.14
P23	P23	23	189.10	5,195	27.47	360	2.30	828	5,7,9,10,13,14,16	pt23	2,941	11.92
P13	P13	13	79.00	2,231	28.24	320	2.30	736	5,7,9,10,13,14,16	pt13	1,193	13.14
P56	P56	56	168.20	4,824	28.68	504	1.20	605	1,2,3,8	pt2	2,660	12.87
P58	P58	58	55.00	1,590	28.91	420	2.33	979	1,2,3,8	pt54	1,208	6.95
P36	P36	36	133.50	3,895	29.18	336	2.00	672	5,7,9,10,13,14,16	pt36	2,057	13.77
P6	P6	6	606.20	18,654	30.77	350	2.30	805	5,7,9,10,13,14,16	pt6	9,293	15.44
P53	P53	53	267.70	8,264	30.87	504	1.20	605	1,2,3,8	pt2	3,734	16.92
P100	P100	100	85.20	2,665	31.28	420	1.33	559	1,2,3,8	pt56	1,115	18.19
P122	P122	122	45.40	1,421	31.30	490	1.33	652	1,2,3,8	pt56	725	15.33
P3	P3	3	796.20	26,204	32.91	420	2.10	882	5,7,9,10,13,14,16	pt3	12,108	17.70
P14	P14	14	271.20	8,951	33.01	432	2.10	908	5,7,9,10,13,14,16	pt14	3,732	19.24
P4	P4	4	233.80	7,877	33.69	420	2.00	840	5,7,9,10,13,14,16	pt4	3,358	19.33
P123	P123	123	223.40	7,600	34.02	490	2.33	1,142	1,2,3,8	pt51	3,738	17.29
P77	P77	77	33.00	1,146	34.73	336	1.20	404	1,2,3,8	pt61	565	17.61
P9	P9	9	77.20	2,688	34.82	420	2.10	882	5,7,9,10,13,14,16	pt9	1,082	20.80
P8	P8	8	88.30	3,152	35.70	350	2.30	805	5,7,9,10,13,14,16	pt8	1,342	20.50
P7	P7	7	137.30	4,939	35.97	420	2.00	840	5,7,9,10,13,14,16	pt7	1,979	21.56
P26	P26	26	11.60	418	36.03	360	2.10	756	5,7,9,10,13,14,16	pt26	187	19.91
P10	P10	10	165.60	5,971	36.06	420	2.10	882	5,7,9,10,13,14,16	pt10	2,649	20.06
P55	P55	55	79.50	2,876	36.18	441	2.33	1,028	1,2,3,8	pt52	1,367	18.98
P63	P63	63	42.40	1,565	36.91	490	1.33	652	1,2,3,8	pt51	689	20.66
P59	P59	59	152.80	5,666	37.08	420	1.33	559	1,2,3,8	pt51	3,095	16.83
P20	P20	20	13.60	506	37.21	480	2.00	960	5,7,9,10,13,14,16	pt20	272	17.21
P12	P12	12	51.80	1,928	37.22	350	2.30	805	5,7,9,10,13,14,16	pt12	883	20.17
P44	P44	44	5.50	205	37.27	360	2.10	756	5,7,9,10,13,14,16	pt44	126	14.36
P15	P15	15	86.60	3,244	37.46	336	2.00	672	5,7,9,10,13,14,16	pt15	1,454	20.67

Product List (Sorted by Safety Alert)

Product ✕

Name	ShortName	Nr	DailySales	InventoryAtStart	CalcDaysCover	LotSize	CycleTime	LotDuration	Machine	ProductType	SafetyStock	SafetyAl...
P35	P35	35	1.30	26	20.00	120	2.30	276	5,7,9,10,13,14,16	pt35	33	0.00
P51	P51	51	5.70	405	71.05	140	4.50	630	2	pt50	381	4.21
P58	P58	58	55.00	1,590	28.91	420	2.33	979	1,2,3,8	pt54	1,208	6.95
P82	P82	82	6.10	259	42.46	441	1.33	587	1,2,3,8	pt51	189	11.48
P23	P23	23	189.10	5,195	27.47	360	2.30	828	5,7,9,10,13,14,16	pt23	2,941	11.92
P5	P5	5	267.30	7,152	26.76	350	2.30	805	5,7,9,10,13,14,16	pt5	3,906	12.14
P56	P56	56	168.20	4,824	28.68	504	1.20	605	1,2,3,8	pt2	2,660	12.87
P13	P13	13	79.00	2,231	28.24	320	2.30	736	5,7,9,10,13,14,16	pt13	1,193	13.14
P18	P18	18	217.50	5,710	26.25	336	2.00	672	5,7,9,10,13,14,16	pt18	2,814	13.31
P36	P36	36	133.50	3,895	29.18	336	2.00	672	5,7,9,10,13,14,16	pt36	2,057	13.77
P44	P44	44	5.50	205	37.27	360	2.10	756	5,7,9,10,13,14,16	pt44	126	14.36
P122	P122	122	45.40	1,421	31.30	490	1.33	652	1,2,3,8	pt56	725	15.33
P6	P6	6	606.20	18,654	30.77	350	2.30	805	5,7,9,10,13,14,16	pt6	9,293	15.44
P25	P25	25	7.50	326	43.47	120	2.30	276	5,7,9,10,13,14,16	pt25	210	15.47
P59	P59	59	152.80	5,666	37.08	420	1.33	559	1,2,3,8	pt51	3,095	16.83
P53	P53	53	267.70	8,264	30.87	504	1.20	605	1,2,3,8	pt2	3,734	16.92
P112	P112	112	3.40	134	39.41	588	1.20	706	1,2,3,8	pt2	76	17.06
P20	P20	20	13.60	506	37.21	480	2.00	960	5,7,9,10,13,14,16	pt20	272	17.21
P32	P32	32	5.40	222	41.11	480	2.00	960	5,7,9,10,13,14,16	pt32	129	17.22
P123	P123	123	223.40	7,600	34.02	490	2.33	1,142	1,2,3,8	pt51	3,738	17.29
P99	P99	99	5.70	247	43.33	96	2.00	192	1,2,3,8	pt60	148	17.37
P77	P77	77	33.00	1,146	34.73	336	1.20	404	1,2,3,8	pt61	565	17.61
P3	P3	3	796.20	26,204	32.91	420	2.10	882	5,7,9,10,13,14,16	pt3	12,108	17.70
P100	P100	100	85.20	2,665	31.28	420	1.33	559	1,2,3,8	pt56	1,115	18.19
P55	P55	55	79.50	2,876	36.18	441	2.33	1,028	1,2,3,8	pt52	1,367	18.98
P14	P14	14	271.20	8,951	33.01	432	2.10	908	5,7,9,10,13,14,16	pt14	3,732	19.24
P80	P80	80	7.20	293	40.69	420	1.33	559	1,2,3,8	pt51	154	19.31
P4	P4	4	233.80	7,877	33.69	420	2.00	840	5,7,9,10,13,14,16	pt4	3,358	19.33
P49	P49	49	50.90	2,273	44.66	378	1.00	378	2	pt48	1,260	19.90

Setup Matrix

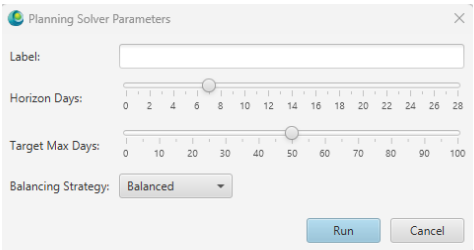
Product | Setup Matrix ✕

	Row	pt1	pt10	pt11	pt12	pt13	pt14	pt15	pt16	pt17	pt18	pt19	pt2	pt20	pt21	pt22	pt23	pt24	pt25	pt26	pt27	pt28
1	pt1	0	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	90	90	90
2	pt10	90	0	30	60	90	60	60	30	60	60	60	90	90	30	60	90	60	90	60	60	60
3	pt11	90	30	0	30	90	60	60	30	60	60	60	90	90	60	60	90	60	90	60	60	60
4	pt12	90	60	30	0	60	90	60	60	60	60	60	90	90	90	60	60	30	60	90	90	30
5	pt13	90	90	90	60	0	90	90	60	90	90	60	90	90	90	90	60	60	60	90	60	30
6	pt14	90	60	60	90	90	0	90	60	90	60	90	90	60	30	90	30	90	60	30	30	90
7	pt15	90	60	60	60	90	90	0	60	60	30	60	90	90	90	30	90	60	60	60	90	60
8	pt16	90	30	30	60	60	60	60	0	60	60	30	90	90	60	60	90	60	90	60	60	30
9	pt17	90	60	60	60	90	90	60	60	0	60	30	90	90	90	30	90	30	90	90	90	60
10	pt18	90	60	60	60	90	60	30	60	60	0	60	90	60	90	60	60	60	90	90	60	60
11	pt19	90	60	60	60	60	90	60	30	30	60	0	90	90	90	30	90	60	90	90	90	30
12	pt2	30	90	90	90	90	90	90	90	90	90	90	0	90	90	90	90	90	90	90	90	90
13	pt20	90	90	90	90	90	60	90	90	90	60	90	90	0	90	90	60	90	90	90	60	90
14	pt21	90	30	60	90	90	30	90	60	90	90	90	90	90	0	90	60	90	60	30	60	90
15	pt22	90	60	60	60	90	90	30	60	30	60	30	90	90	90	0	90	60	60	60	90	60
16	pt23	90	90	90	60	60	30	90	90	90	60	90	90	60	60	90	0	60	30	60	60	60
17	pt24	90	60	60	30	60	90	60	60	30	60	60	90	90	90	60	60	0	60	90	90	30
18	pt25	90	90	90	60	60	60	60	90	90	90	90	90	90	60	60	30	60	0	30	90	60
19	pt26	90	60	60	90	90	30	60	60	90	90	90	90	90	30	60	60	90	30	0	60	90
20	pt27	90	60	60	90	60	30	90	60	90	60	90	90	60	60	90	60	90	90	60	0	90
21	pt28	90	60	60	30	30	90	60	30	60	60	30	90	90	90	60	60	30	60	90	90	0
22	pt29	90	60	60	90	90	30	90	60	90	60	90	90	60	60	90	60	90	90	60	30	90
23	pt3	90	30	30	60	90	30	60	30	60	30	60	90	60	60	60	60	60	90	60	30	60
24	pt30	90	90	90	90	90	60	90	90	60	60	60	90	60	90	60	60	90	90	90	60	90
25	pt31	90	60	60	90	90	60	90	60	90	90	90	90	90	60	90	90	90	90	60	60	90
26	pt32	90	60	90	90	90	90	90	90	90	90	90	90	30	60	90	90	90	90	90	90	90
27	pt33	90	90	90	90	90	30	90	90	60	60	60	90	60	60	60	30	90	60	60	60	90
28	pt34	90	60	60	60	90	90	30	60	60	30	60	90	90	90	60	90	60	90	90	90	60
29	pt35	90	90	90	60	60	90	60	90	90	90	90	90	90	90	60	60	60	30	60	90	60
30	pt36	90	30	60	60	90	90	30	60	60	30	60	90	90	60	60	90	60	90	90	90	60
31	pt37	90	60	60	60	90	90	30	60	60	30	60	90	90	90	60	90	60	90	90	90	60
32	pt38	90	90	90	90	90	60	60	90	60	90	60	90	90	60	30	60	90	30	30	90	90
33	pt39	90	90	90	60	60	90	90	90	90	90	90	90	90	90	90	60	60	60	90	90	60
34	pt4	90	60	60	60	90	60	60	60	30	30	30	90	60	90	30	60	60	90	90	60	60
35	pt40	90	90	90	90	60	60	90	90	60	60	60	90	60	90	60	60	90	90	90	30	90
36	pt41	90	90	90	90	90	90	60	90	90	90	90	90	30	90	60	90	90	60	60	90	90
37	pt42	90	60	60	90	30	60	90	30	90	90	60	90	90	60	90	90	90	90	60	30	60
38	pt43	90	60	60	60	90	90	60	60	30	60	30	90	90	90	30	90	60	90	90	90	60
39	pt44	90	30	60	90	90	60	90	60	90	90	90	90	90	30	90	90	90	90	60	60	90
40	pt45	90	90	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	90	90
41	pt46	90	60	60	30	60	90	60	60	60	60	60	90	90	90	60	60	30	60	90	90	30
42	pt47	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90

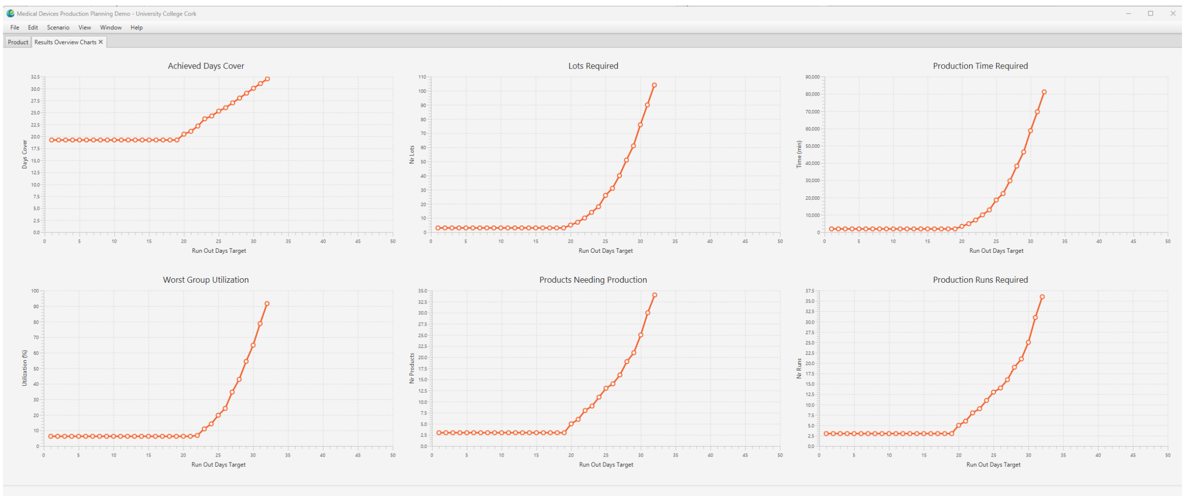
	Row	pt29	pt3	pt30	pt31	pt32	pt33	pt34	pt35	pt36	pt37	pt38	pt39	pt4	pt40	pt41	pt42	pt43	pt44	pt45	pt46	pt47	pt48
1	pt1	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	60
2	pt10	60	30	90	60	60	90	60	90	30	60	90	90	60	90	90	60	60	30	90	60	90	90
3	pt11	60	30	90	60	90	90	60	90	60	60	90	90	60	90	90	60	60	60	90	60	90	90
4	pt12	90	60	90	90	90	90	60	60	60	60	90	60	60	90	90	90	60	90	90	30	90	90
5	pt13	90	90	90	90	90	90	90	60	90	90	90	60	90	60	90	30	90	90	90	60	90	90
6	pt14	30	30	60	60	90	30	90	90	90	90	60	90	60	60	90	60	90	60	90	90	90	90
7	pt15	90	60	90	90	90	90	30	60	30	30	60	90	60	90	60	90	60	90	90	60	90	90
8	pt16	60	30	90	60	90	90	60	90	60	60	90	90	60	90	90	30	60	60	90	60	90	90
9	pt17	90	60	60	90	90	60	60	90	60	60	60	90	30	60	90	90	30	90	90	60	90	90
10	pt18	60	30	60	90	90	60	30	90	30	30	90	90	30	60	90	90	60	90	90	60	90	90
11	pt19	90	60	60	90	90	60	60	90	60	60	60	90	30	60	90	60	30	90	90	60	90	90
12	pt2	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	60
13	pt20	60	60	60	90	30	60	90	90	90	90	90	90	60	60	30	90	90	90	30	90	90	90
14	pt21	60	60	90	60	60	60	90	90	60	90	60	90	90	90	90	60	90	30	90	90	90	90
15	pt22	90	60	60	90	90	60	60	60	60	60	30	90	30	60	60	90	30	90	90	60	90	90
16	pt23	60	60	60	90	90	30	90	60	90	90	60	60	60	60	90	90	90	90	90	60	90	90
17	pt24	90	60	90	90	90	90	60	60	60	60	90	60	60	90	90	90	60	90	90	30	90	90
18	pt25	90	90	90	90	90	60	90	30	90	90	30	60	90	90	60	90	90	90	90	60	90	90
19	pt26	60	60	90	60	90	60	90	60	90	90	30	90	90	90	60	60	90	60	90	90	90	90
20	pt27	30	30	60	60	90	60	90	90	90	90	90	90	60	30	90	30	90	60	90	90	90	90
21	pt28	90	60	90	90	90	90	60	60	60	60	90	60	60	90	90	60	60	90	90	30	90	90
22	pt29	0	30	60	60	90	60	90	60	90	90	90	60	60	60	90	60	90	30	90	90	90	90
23	pt3	30	0	60	60	90	60	60	90	60	60	90	90	30	60	90	60	60	60	90	60	90	90
24	pt30	60	60	0	60	90	30	90	90	90	90	60	90	30	30	90	90	60	90	90	90	90	90
25	pt31	60	60	60	0	90	90	60	90	90	90	90	90	90	90	90	60	60	60	60	60	90	90
26	pt32	90	90	90	90	0	90	90	90	60	90	90	90	90	90	30	90	90	60	30	90	90	90
27	pt33	60	60	30	90	90	0	90	90	90	90	30	90	30	30	90	90	60	90	90	90	90	90
28	pt34	90	60	90	60	90	90	0	90	30	30	90	90	60	90	90	90	30	90	60	30	90	90
29	pt35	60	90	90	90	90	90	90	0	90	90	60	30	90	90	60	90	90	60	90	60	90	90
30	pt36	90	60	90	90	60	90	30	90	0	30	90	90	60	90	90	90	60	60	90	60	90	90
31	pt37	90	60	90	90	90	90	30	90	30	0	90	60	60	90	90	90	60	90	90	60	90	90
32	pt38	90	90	60	90	90	30	90	60	90	90	0	90	60	60	60	90	60	90	90	90	90	90
33	pt39	60	90	90	90	90	90	90	30	90	60	90	0	90	90	90	90	90	60	90	60	90	90
34	pt4	60	30	30	90	90	30	60	90	60	60	60	90	0	30	90	90	30	90	90	60	90	90
35	pt40	60	60	30	90	90	30	90	90	90	90	60	90	30	0	90	60	60	90	90	90	90	90
36	pt41	90	90	90	90	30	90	90	60	90	90	60	90	90	90	0	90	90	90	30	90	90	90
37	pt42	60	60	90	60	90	90	90	90	90	90	90	90	90	60	90	0	90	60	90	90	90	90
38	pt43	90	60	60	60	90	60	30	90	60	60	60	90	30	60	90	90	0	90	60	30	90	90
39	pt44	30	60	90	60	60	90	90	60	60	90	90	60	90	90	90	60	90	0	90	90	90	90
40	pt45	90	90	90	60	30	90	60	90	90	90	90	90	90	90	30	90	60	90	0	60	90	90
41	pt46	90	60	90	60	90	90	30	60	60	60	90	60	60	90	90	90	30	90	60	0	90	90
42	pt47	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	0	90

	Row	pt49	pt5	pt50	pt51	pt52	pt53	pt54	pt55	pt56	pt57	pt58	pt59	pt6	pt60	pt61	pt62	pt63	pt64	pt65	pt
1	pt1	90	90	60	30	60	30	90	60	60	30	90	60	90	60	60	90	60	60	30	90
2	pt10	90	30	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	30
3	pt11	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	60
4	pt12	90	30	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	60
5	pt13	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	90
6	pt14	90	90	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	90
7	pt15	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	60
8	pt16	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	60
9	pt17	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	30
10	pt18	90	60	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	60
11	pt19	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	30
12	pt2	90	90	60	30	60	30	90	30	60	30	90	60	90	30	60	90	60	60	30	90
13	pt20	90	90	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	90
14	pt21	90	60	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	60
15	pt22	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	30
16	pt23	90	60	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	90
17	pt24	90	30	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	60
18	pt25	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	90
19	pt26	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
20	pt27	90	90	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	90
21	pt28	90	30	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	60
22	pt29	90	90	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	90
23	pt3	90	60	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	60
24	pt30	90	90	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	60
25	pt31	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
26	pt32	90	60	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	60
27	pt33	90	90	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	60
28	pt34	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	60
29	pt35	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	90
30	pt36	90	30	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	30
31	pt37	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	60
32	pt38	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	60
33	pt39	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	90
34	pt4	90	60	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	30
35	pt40	90	90	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	60
36	pt41	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
37	pt42	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
38	pt43	90	60	90	90	90	90	90	90	90	90	90	90	60	90	90	90	90	90	90	30
39	pt44	90	60	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	60
40	pt45	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
41	pt46	90	30	90	90	90	90	90	90	90	90	90	90	30	90	90	90	90	90	90	60
42	pt47	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90

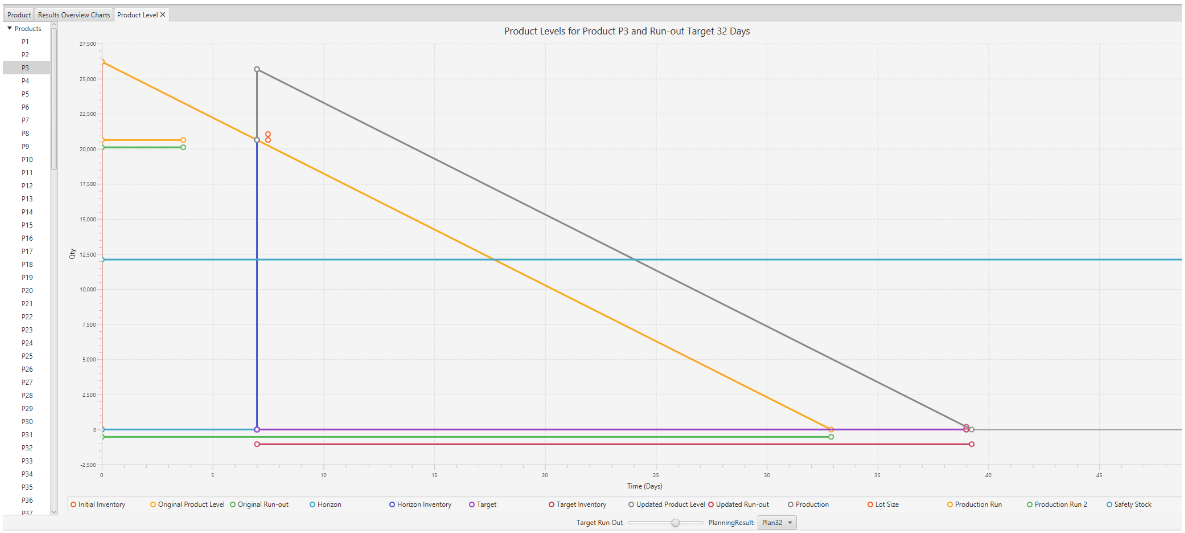
Running the Planning Solver



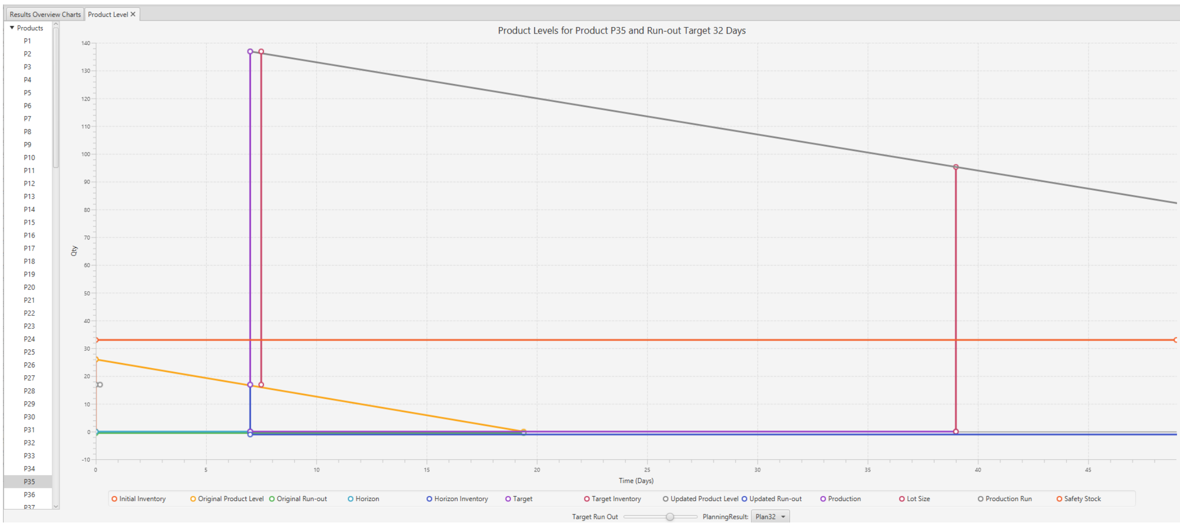
Planner Results



Product Level Chart for Product P3



Product Level Chart for Product P35

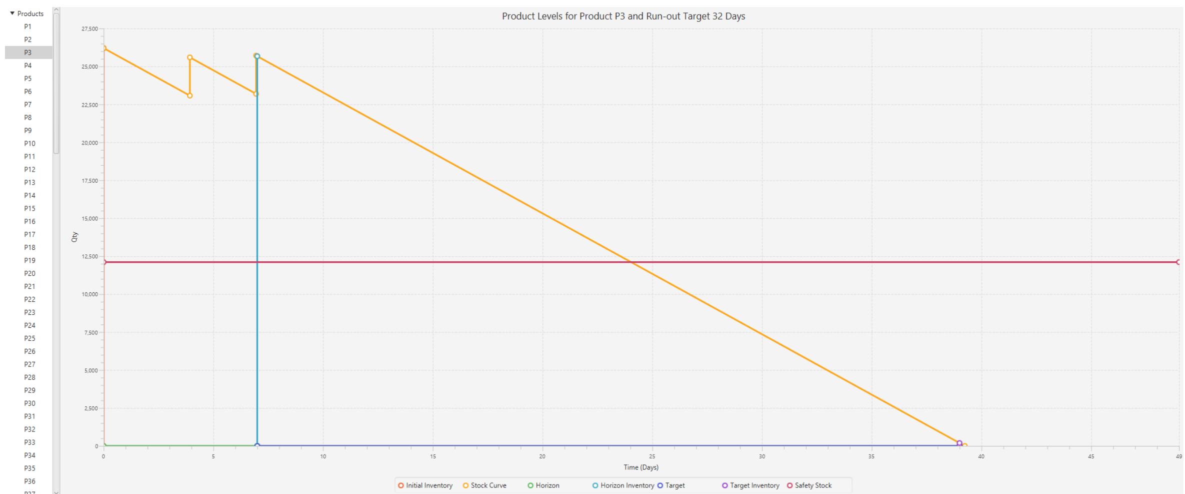


Scheduled Production Runs

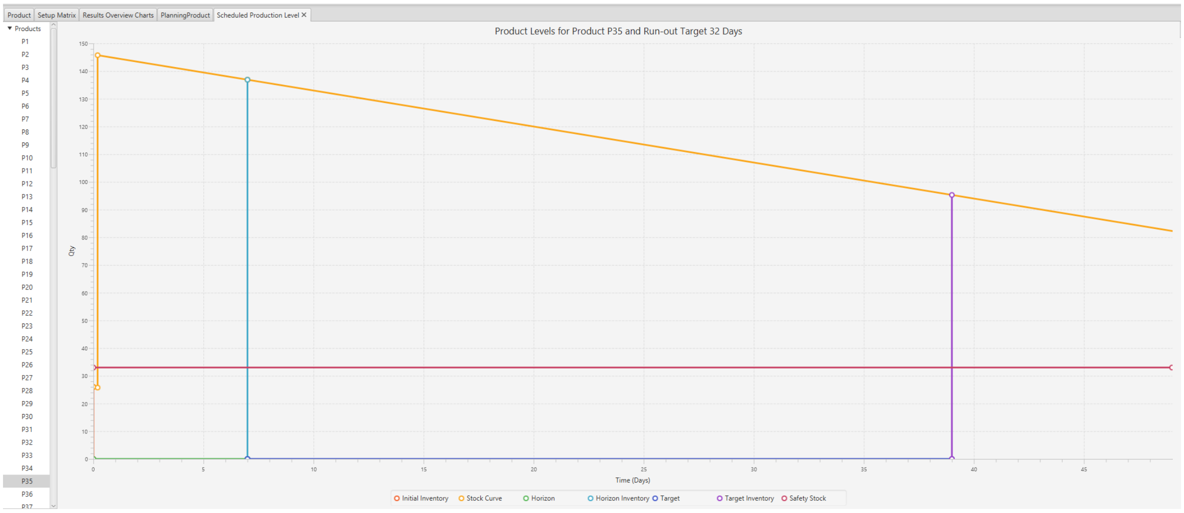
Product | Results Overview Charts | Product Level | Scheduled Production Level | ProductionRun

Name	Product	NrLots	Qty	Due	Start	End	Duration	StartDay	EndDay
job3_0	P3	6	2,520	10,080	366	5,658	5,292	0.25	3.93
job3_1	P3	6	2,520	10,080	4,712	10,004	5,292	3.27	6.95
job4_0	P4	3	1,260	10,080	0	2,520	2,520	0.00	1.75
job5_0	P5	10	3,500	10,080	1,794	9,844	8,050	1.25	6.84
job6_0	P6	7	2,450	10,080	4,224	9,859	5,635	2.93	6.85
job6_1	P6	8	2,800	10,080	0	6,440	6,440	0.00	4.47
job7_0	P7	1	420	10,080	7,442	8,282	840	5.17	5.75
job8_0	P8	1	350	10,080	816	1,621	805	0.57	1.13
job9_0	P9	1	420	10,080	3,282	4,164	882	2.28	2.89
job10_0	P10	2	840	10,080	0	1,764	1,764	0.00	1.23
job11_0	P11	1	420	10,080	6,500	7,382	882	4.51	5.13
job12_0	P12	1	350	10,080	1,651	2,456	805	1.15	1.71
job13_0	P13	3	960	10,080	0	2,208	2,208	0.00	1.53
job14_0	P14	4	1,728	10,080	0	3,632	3,632	0.00	2.52
job15_0	P15	1	336	10,080	2,580	3,252	672	1.79	2.26
job17_0	P17	1	420	10,080	5,718	6,558	840	3.97	4.55
job18_0	P18	9	3,024	10,080	3,144	9,192	6,048	2.18	6.38
job20_0	P20	1	480	10,080	3,692	4,652	960	2.56	3.23
job23_0	P23	7	2,520	10,080	2,516	8,312	5,796	1.75	5.77
job26_0	P26	1	360	10,080	0	756	756	0.00	0.53
job35_0	P35	1	120	0	0	276	276	0.00	0.19
job36_0	P36	4	1,344	10,080	6,618	9,306	2,688	4.60	6.46
job44_0	P44	1	360	10,080	2,298	3,054	756	1.60	2.12
job46_0	P46	1	350	10,080	8,372	9,177	805	5.81	6.37
job51_0	P51	1	140	6,064	0	630	630	0.00	0.44
job53_0	P53	5	2,520	10,080	707	3,732	3,025	0.49	2.59
job55_0	P55	1	441	10,080	2,580	3,608	1,028	1.79	2.51
job56_0	P56	4	2,016	10,080	7,218	9,638	2,420	5.01	6.69
job58_0	P58	2	840	10,002	3,668	5,626	1,958	2.55	3.91
job59_0	P59	1	420	10,080	464	1,023	559	0.32	0.71
job63_0	P63	1	490	10,080	0	652	652	0.00	0.45
job77_0	P77	1	336	10,080	0	404	404	0.00	0.28
job78_0	P78	1	588	10,080	0	647	647	0.00	0.45

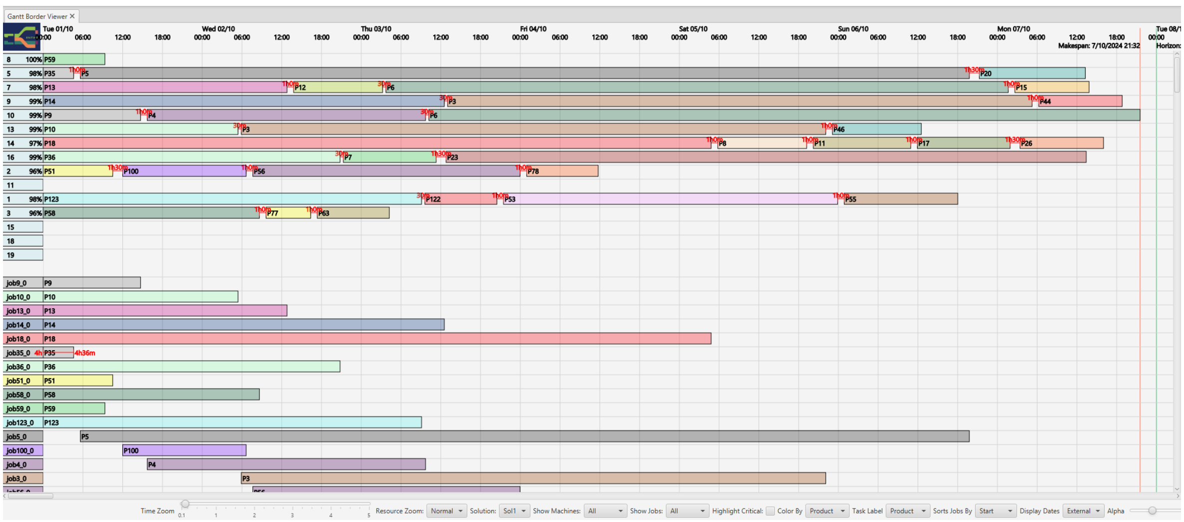
Production Level Chart for Product P3



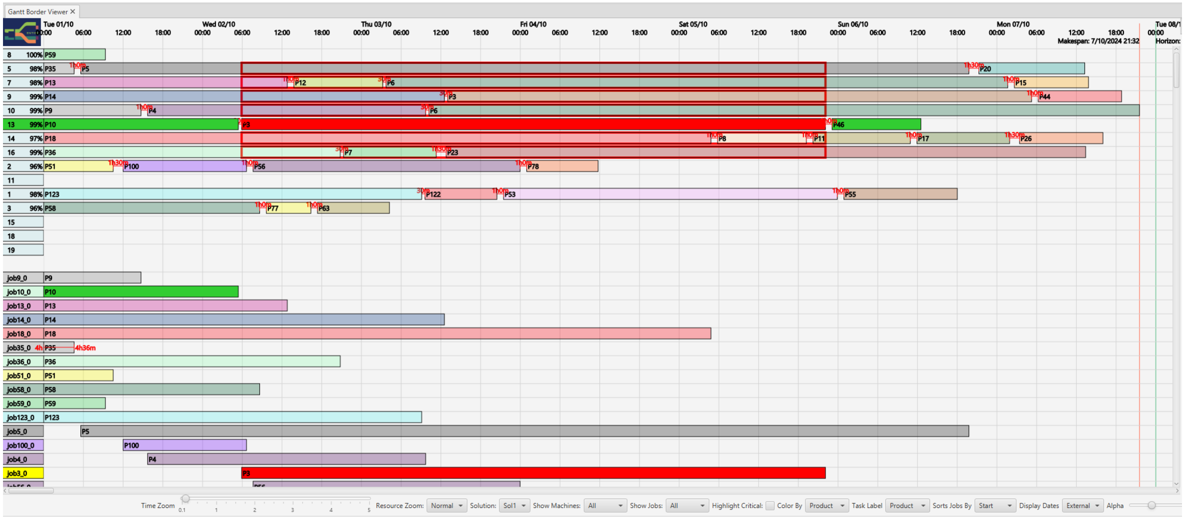
Production Level Chart for Product P35



Detailed Schedule



Showing Alternative Machines in Gantt Chart



Summary

- We demonstrated the use of our scheduling tool inside a production planning problem from industry
- Production planning decides which products to make in which quantity
    - Balance stock levels against projected demand
    - Allow for product specific safety stock levels
- Uses estimate of production capacity over planning horizon
- Use detailed scheduling to validate plan

Part IX

Oven Scheduling Case Study

Key Points

- Discusses two topics:
  - Solve a very specific industrial scheduling problem from the ASSISTANT EU project
  - Discuss the general issue of short-term scheduling vs. long-term objectives

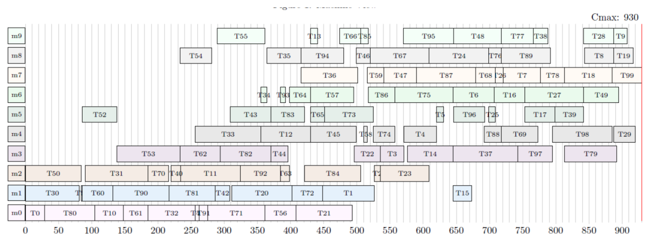
Research Challenge

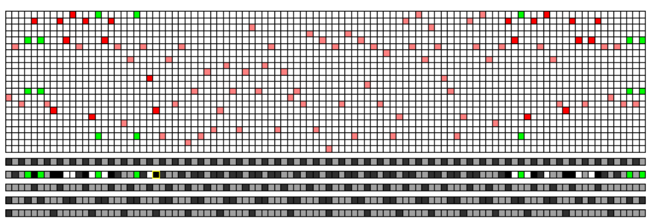
- Often the long-term business objectives are not visible in the operational decision problem
- We optimize a short-term objective without understanding the impact in the long term
- What choices should we make in short-term to improve overall result?
- Especially important when future data not yet visible
- Surprisingly, this problem is rarely discussed in literature

Examples

- Production Scheduling
- Nearly all scheduling benchmarks use c_{max} (makespan) as objective
- Why?
- Do we want to close factory as rapidly as possible?

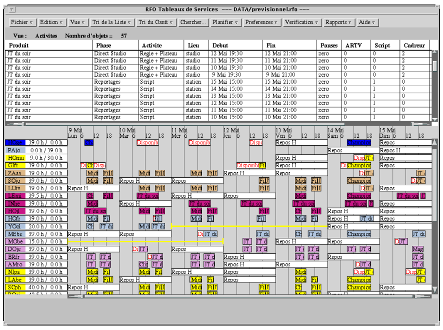
- Car Sequencing
- The best heuristics push difficult cars to the edge of schedule
- Because they are easier to schedule this way
- But: It makes it hard to schedule next day





Examples

- Personnel Rostering
- Satisfy working rules and demands for period
- But: rules apply on a rolling horizon
- Easy to over-constrain problem for next period



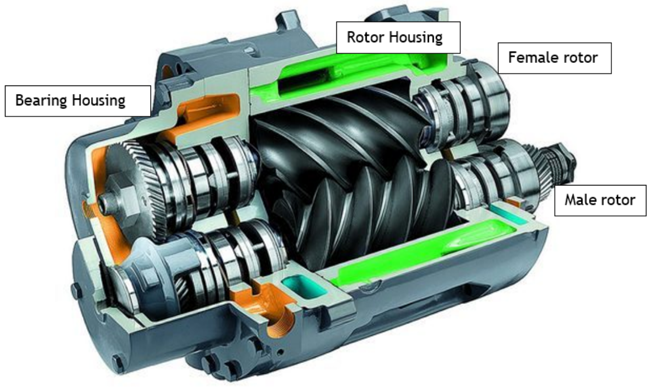
- Transportation Planning
- Build daily delivery tours, optimizing cost
- Where are your trucks at 10PM?
- Also, avoid cherry-picking at start of week

Problem Studied Here

- Example from the ASSISTANT EU project (ended last year)
- Oven schedule for one of the industrial partners
- Schedule tasks on a set of ovens
- Tasks can share oven only if they are compatible
- Conflicting objectives
  - Energy use of ovens very significant, reduce when ovens are used
  - Waiting for an oven affects quality of product
- Jobs only visible when previous process step starts
- Currently scheduled by hand, industry partner expressed strong need for change

What does this look like in the real world?

Industrial Oven



Rotors in Compressor

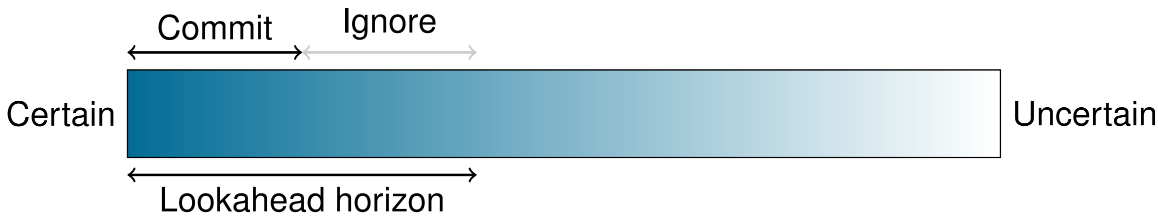
Solution Approach: Constraint Programming

- Declarative modelling approach for combinatorial problems
  - Problem expressed in terms of variables and constraints
- Global constraints
  - Combines expressive modelling abstractions and powerful reasoning
  - Examples: disjunctive, cumulative, global_cardinality
- Compositional: Add constraints as required

- Main application areas
  - Scheduling, rostering, transportation
  - Also: test generation, verification, configuration

Overall Decomposition (Standard)

- We can only see that far into future
- We do not want to take decisions now that we might regret later
- We have to make some decisions now otherwise we never do anything
- *Rolling horizon* decomposition
  - We schedule up to *lookahead horizon* units into the future
  - We commit to implement resulting schedule only to up *commitHorizon*
  - We reschedule when we receive new information, or we reach the end of commitment
  - We solve each short-term sub problem based on short-term objectives



Short-Term Schedule Modelling

- Challenge: There is no global constraint to express the oven resource constraint
- We are not able to invest a lot of time/resources to develop such a constraint
- Two choices:
  - Two traditional models with variables linking them (Lackner et al, Constraints 2023)
  - Direct model expressing conditions as disjunctions of basic constraints

The Standard Pieces

- Jobs N consisting of multiple stages Q, tasks for each stage of each job, running on machines M
- Release dates r_i of jobs given by up-stream schedule
- WiP w_k on certain machines resulting from earlier schedule
- Machine m_{ij} and start variables s_{ij} for each task
- Precedence constraints between tasks of each jobs, with total waiting time c_i when waiting for resource
- Total number of ovens used in schedule *nrOvens* by *nvalue* constraint

$$\text{nvalue}(\text{nrOvens}, [m_{ij} | i \in N, j \in Q] \mathbin{++} [k | k \in M \text{ s.t. } w_k > 0])$$

Resource Constraints

We start from the basic decomposition of the disjunctive machine choice constraint

$$\forall_{i_1, i_2 \in N} \forall_{j_1, j_2 \in Q \text{ s.t. } <i_1, j_1> \neq <i_2, j_2>} : \quad m_{i_1 j_1} \neq m_{i_2 j_2} \vee \\ s_{i_1 j_1} \geq s_{i_2 j_2} + d_{i_2 j_2} \vee \\ s_{i_2 j_2} \geq s_{i_1 j_1} + d_{i_1 j_1}$$

Express case where tasks share an oven (only when types and stages are the same)

$$\forall_{i_1, i_2 \in N \text{ s.t. } i_1 \neq i_2} \forall_{j \in Q} : \quad m_{i_1 j} \neq m_{i_2 j} \vee \\ s_{i_1 j} \geq s_{i_2 j} + d_{i_2 j} \vee \\ s_{i_2 j} \geq s_{i_1 j} + d_{i_1 j} \vee \\ (t_{i_1 j_1} = t_{i_2 j_2} \wedge m_{i_1 j} = m_{i_2 j} \wedge s_{i_1 j} = s_{i_2 j})$$

Limit stacking

Need binary variables $b_{i_1 i_2 j}$ to state that two jobs i_1 and i_2 share oven in stage j

$$\forall_{i_1, i_2 \in N \text{ s.t. } i_1 < i_2} \forall_{j \in Q} : \quad \begin{aligned}(b_{i_1 i_2 j} = 0 \wedge (m_{i_1 j} \neq m_{i_2 j} \vee \\ s_{i_1 j} \geq s_{i_2 j} + d_{i_2 j} \vee \\ s_{i_2 j} \geq s_{i_1 j} + d_{i_1 j}) \vee \\ (b_{i_1 i_2 j} = 1 \wedge t_{i_1 j_1} = t_{i_2 j_2} \wedge m_{i_1 j} = m_{i_2 j} \wedge s_{i_1 j} = s_{i_2 j})\end{aligned}$$

Count how many jobs share stage j with job i

$$\forall_{i \in N} \forall_{j \in Q} : \quad z_{ij} = \sum_{i_1 = 1}^{i-1} b_{i_1 i j} + \sum_{i2 = i+1}^{n} b_{i i_2 j}$$

Limit how many tasks can be stacked together

$$\forall_{i \in N} \forall_{j \in Q} : \quad z_{ij} < \text{maxStacked}$$

This should not work!

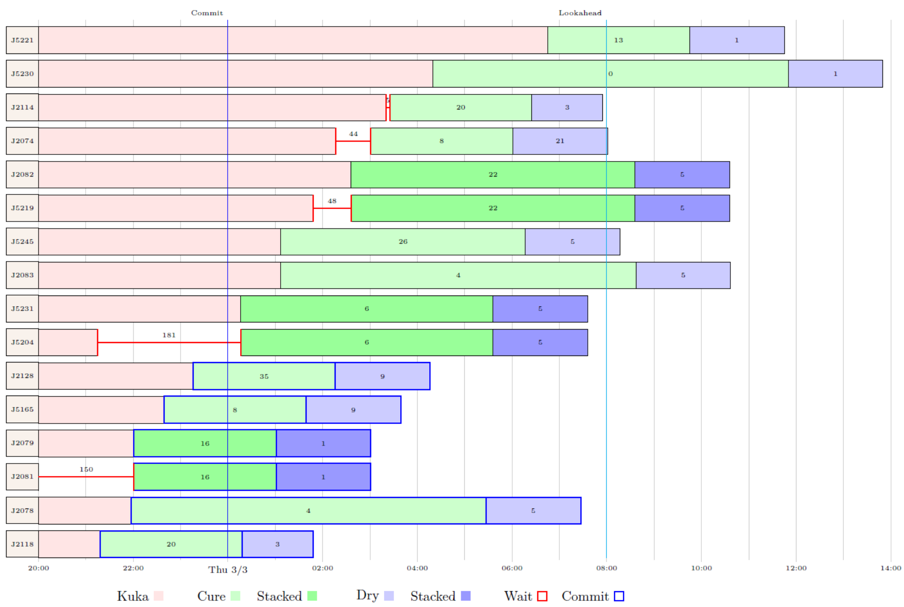
- Weakness of basic decomposition model was the reason to develop the scheduling constraints in the first place
- Does not scale well to thousands of tasks
- But model is well suited to some solvers
  - SAT based solvers, Chuffed, CP-SAT (OR-Tools)
  - MIP solvers
- This works (only) as long as problem size stays manageable

Compound Objective

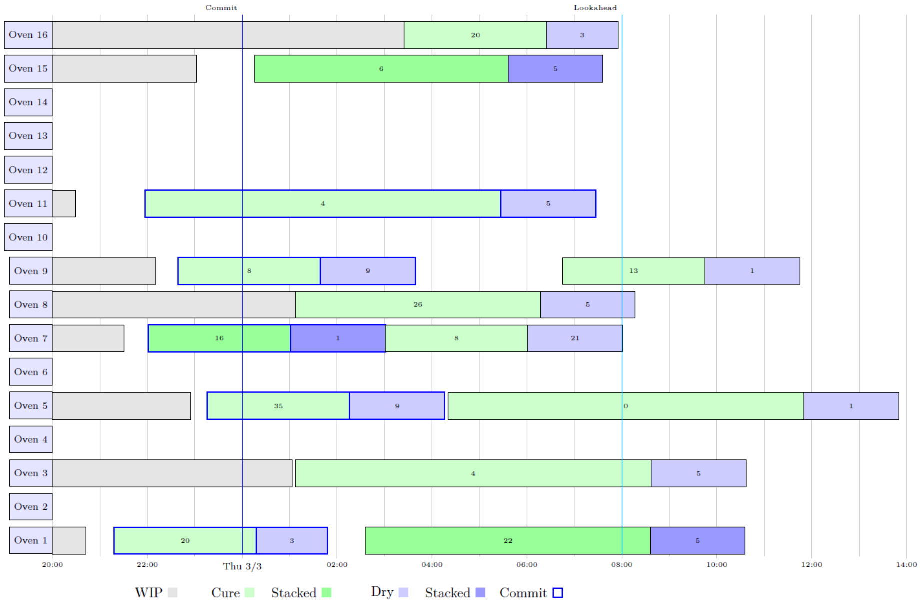
$$\min \alpha_1 \sum_{i \in N} c_i + \alpha_2 \mathrm{nrOvens} + \alpha_3 \sum_{i \in N, j \in Q} z_{ij}$$

- Three conflicting elements
  - Total waiting time for jobs
  - Number of ovens used
  - Number of tasks stacked (negative coefficient)
- Reducing waiting time requires using more ovens
- Improved stacking will require for one job to wait until second is ready

Short-Term Schedule: Job View



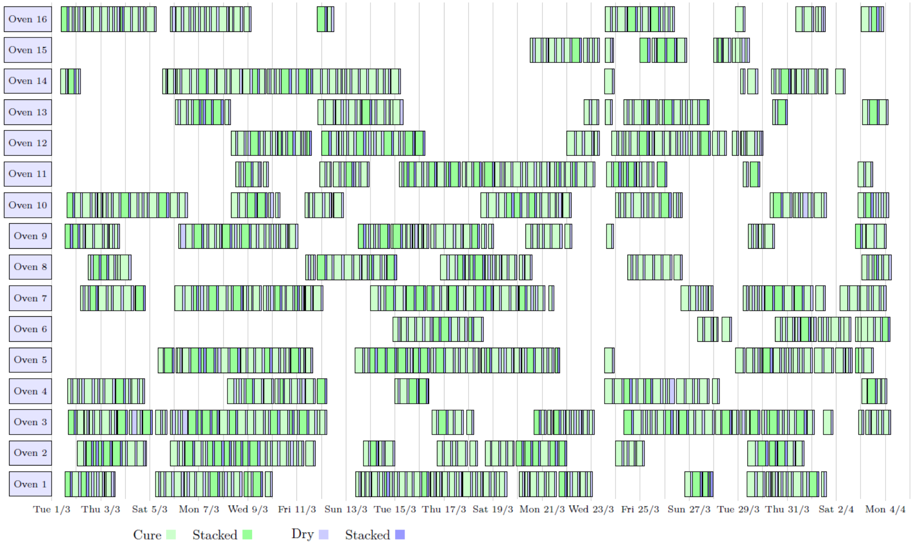
Short Term Schedule: Resource View



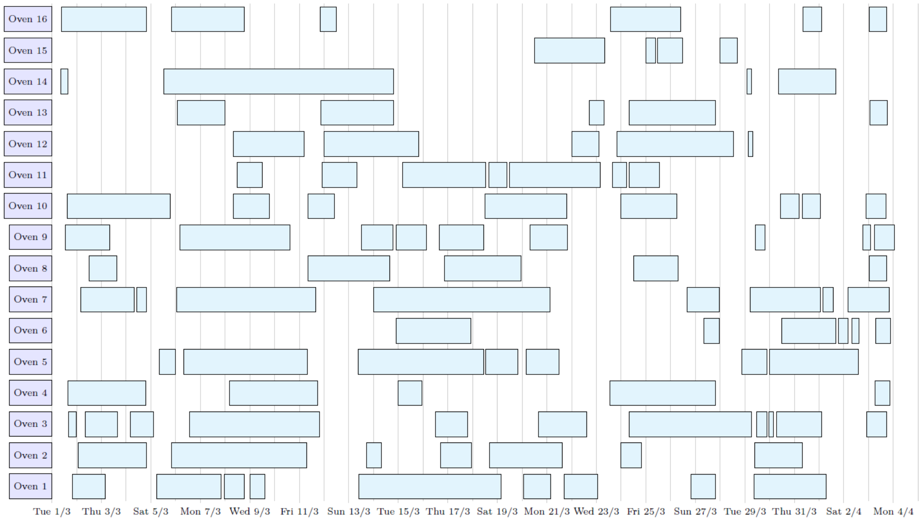
Are the short-term solutions good?

- We solve many problems to optimality, depending on solver
- Optimality gap is small, increasing search time helps a bit
- But are we optimizing the best possible objective?

Long Term Schedule: Detailed Schedule



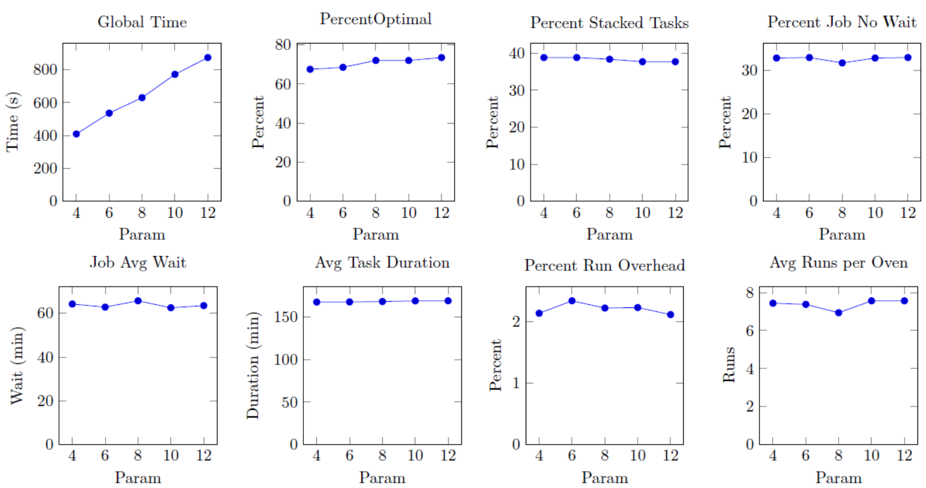
Long Term Schedule: Abstracted Oven Runs



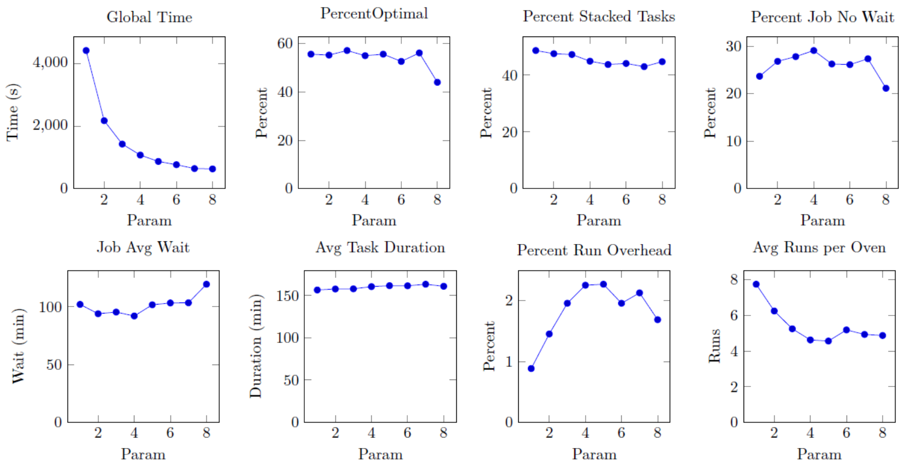
Is that a good global schedule? KPIs

Name	Unit	Explanation
Global Time	Seconds	Total time for solving all sub problems
Nr Jobs	-	Total number of jobs scheduled
Nr Tasks	-	Total number of tasks scheduled
Percent Optimal	Percentage (0-100)	How many sub problems were solved to optimality
Percent Stacked Tasks	Percentage (0-100)	Percentage of all tasks scheduled that were stacked
Percent Jobs No Wait	Percentage (0-100)	Percentage of jobs that were scheduled without any waiting time
Job Average Wait	Minutes	Average wait time over all jobs
Job Maximal Wait	Minutes	Largest waiting time for any job scheduled
Ovens Used	-	Total number of ovens used during period
Avg Task Duration	Minutes	Average tasks duration (influenced by stacking)
Oven Runs	-	Number of oven runs over total horizon
Run Overhead Percent	Percentage (0-100)	Overhead during oven runs when machine is idle
Avg Runs per Oven Used	-	Average number of oven runs per oven used

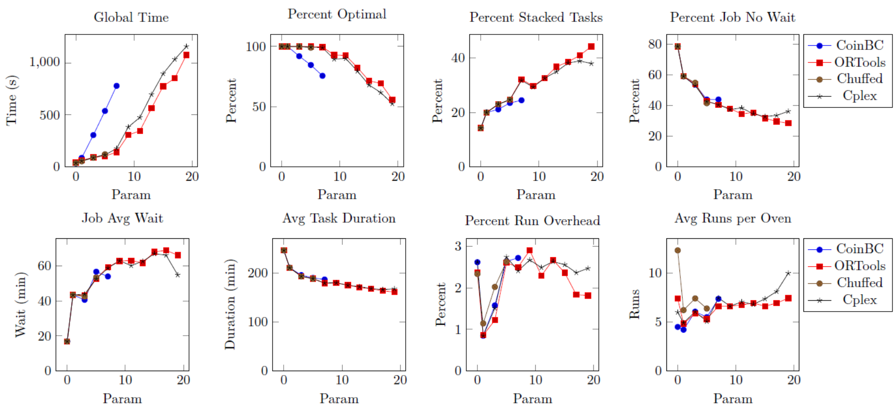
Impact of Lookahead Parameter



Impact of CommitHorizon Parameter



Comparing Different Solvers



Is the global solution really good?

- We schedule with limited information
- Hindsight is 20/20, we cannot expect best possible solution from partial information
- Process Challenge: Can we improve data visibility?
- Demand is variable over time, no steady-state solution
- Modelling Challenge: Can we define a short-term objective that produces better long-term solutions?
- Algorithm Challenge: Can we solve the global problem to optimality?
  - Assumes "a priori" visibility of data
  - This would provide a lower bound
  - But we need optimality to use as bound

Summary

- Discussed a non-standard oven scheduling problem from industry
- Models with decomposition of resource constraints
- Good/very good short-term solutions
- But is the overall schedule close to the global optimum?
- In any case, industry partner was happy with solution and analysis

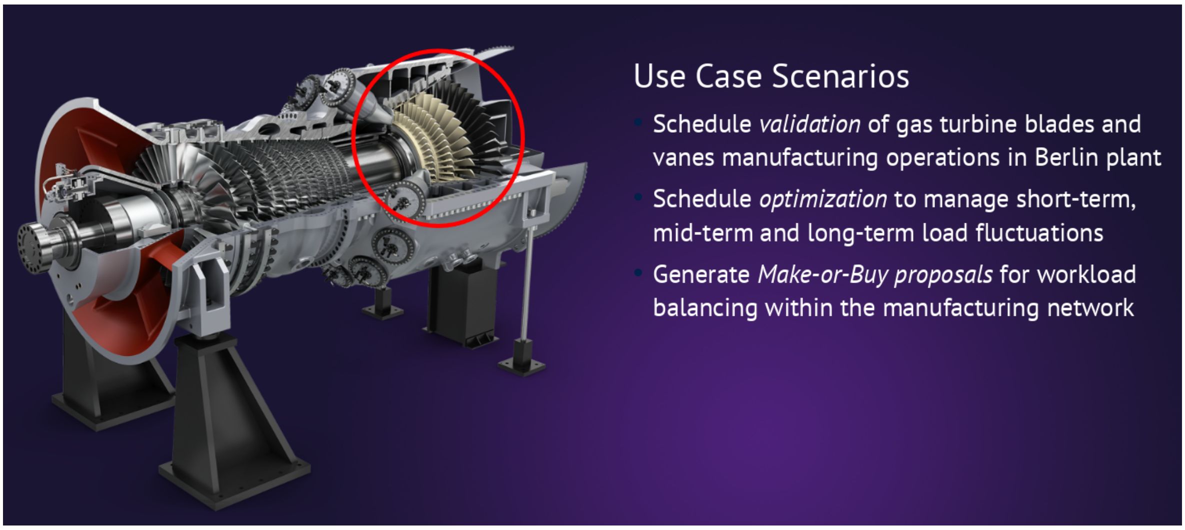
Part X

Blades and Vanes Production Case Study

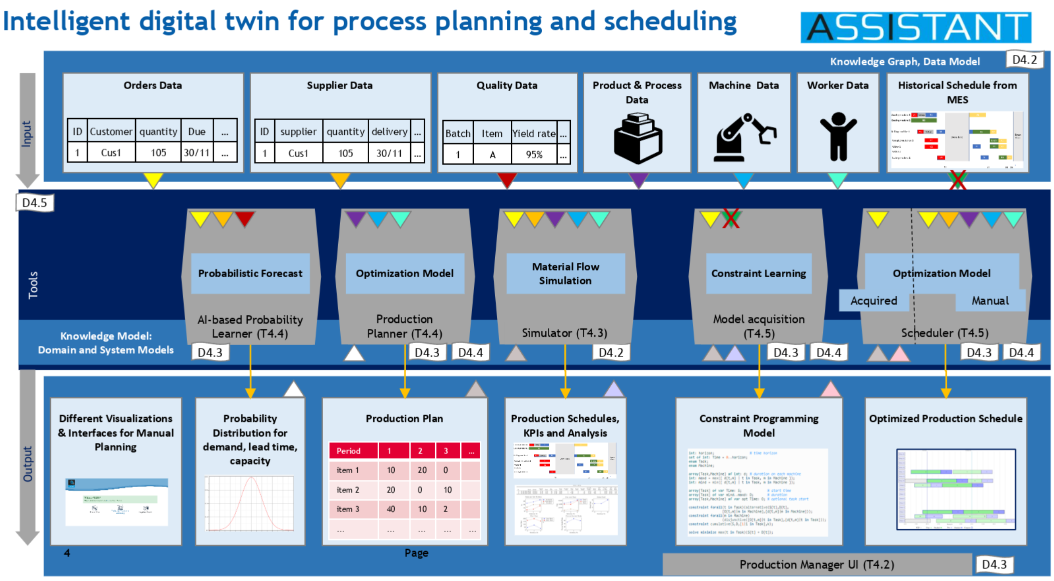
Key Points

- Scheduling/Planning tool for manufacturing industry
- Developed as part of European ASSISTANT project
- Focused on key make-or-buy decisions
- Complex manufacturing process with alternative process paths
- Outperforms both current in-house tool and commercial simulator
- Key Technology: Optimization and Constraint Programming

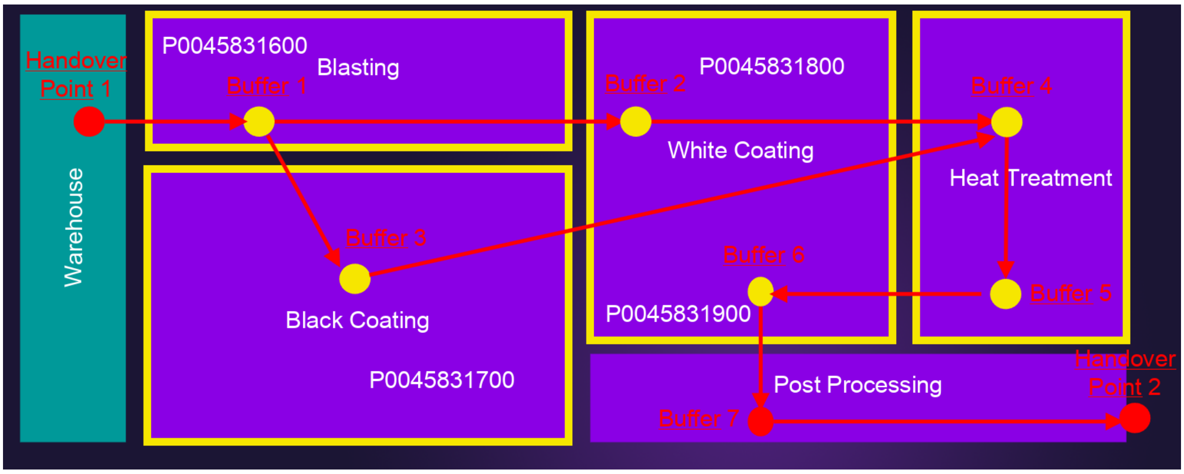
Assistant Siemens Energy Use Case



ASSISTANT Project Overview



SE Product Routing



Test Datasets

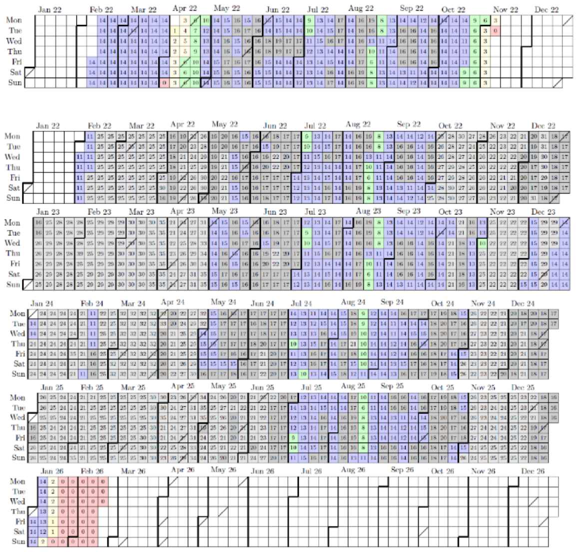
Full Scale Datasets

Berlin06: 96 orders, 9 months horizon, previous review

Berlin07: 450 orders, 4 years horizon

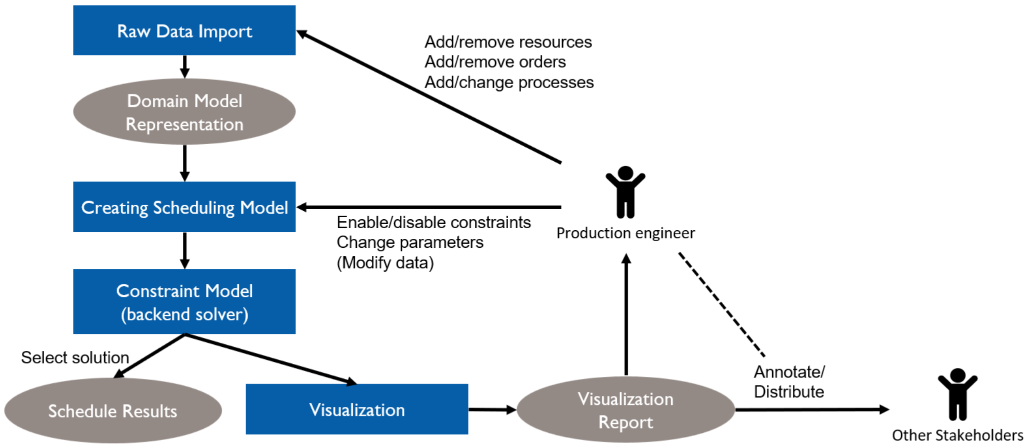
Berlin08: 559 orders, Christmas gap added

Berlin08a: 670 orders, filling gaps



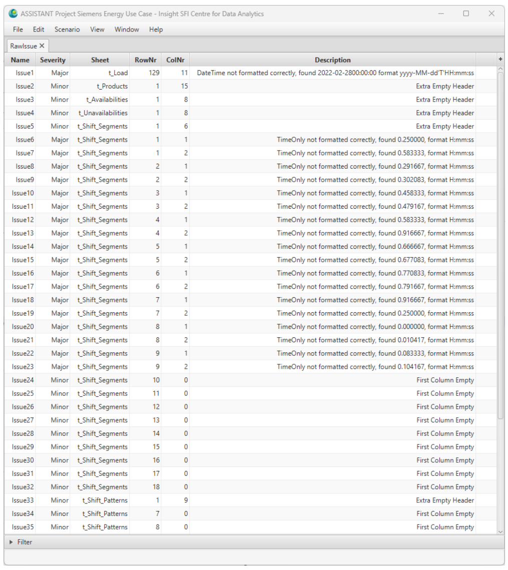
Value in cell indicates active orders
Yellow and red colors indicate low order volume

Optimizer High Level Structure



Raw Data - Manual Data Entry Causes Problems

- Raw data come from spreadsheet
  - 20 tabs
- Excel is a particularly bad input data format
- Realistic, not real data
- Created by hand/automatically from existing test scenarios
- Series of files Berlin01 - Berlin05 were too inconsistent to run
- Berlin06 still contains some errors
- Optimizer explains all issues that it finds


ASSISTANT Project Siemens Energy Use Case - Insight SFI Centre for Data Analytics

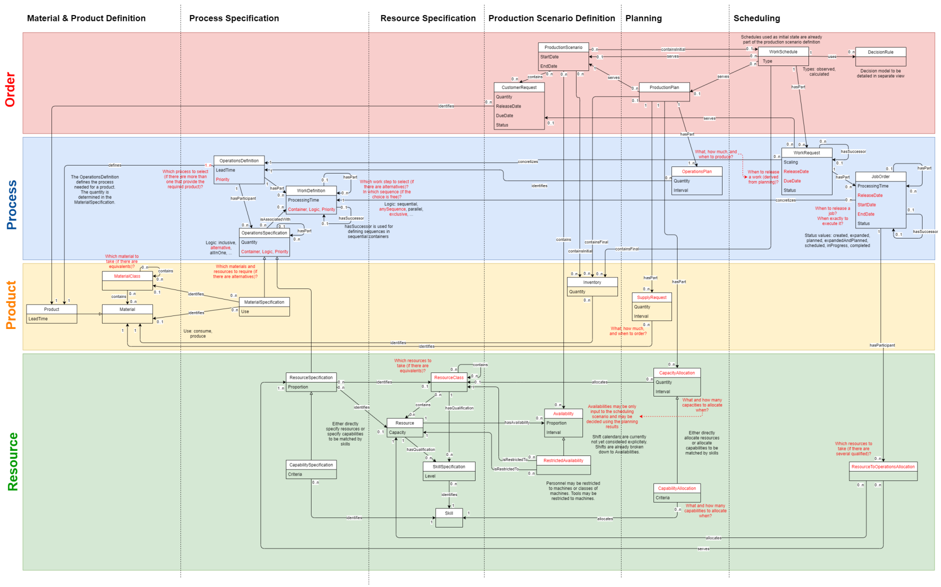
File Edit Scenario View Window Help

RawIssue

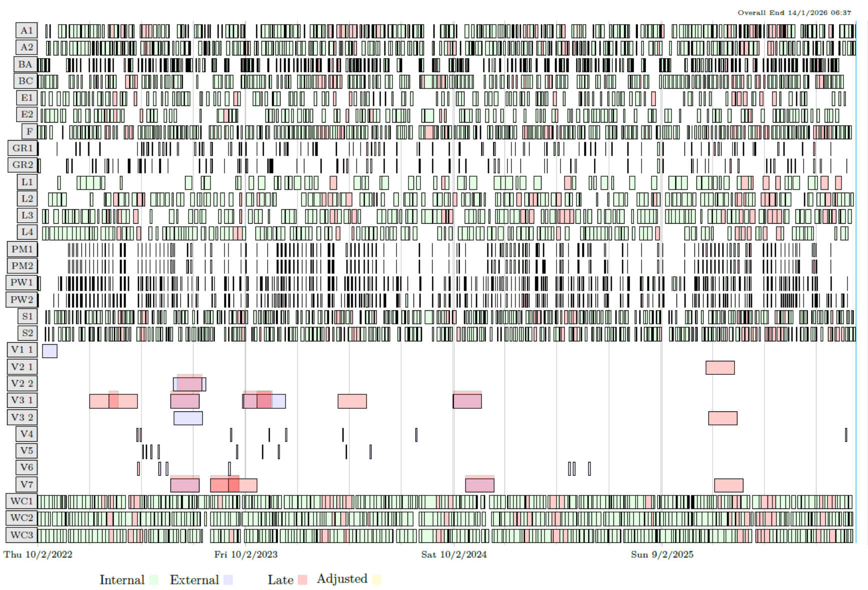
Name	Severity	Sheet	RowNr	ColNr	Description
Issue1	Major	t_Load	129	11	DateTime not formatted correctly, found 2022-02-2800:00:00 format yyyy-MM-dd'T'HH:mm:ss
Issue2	Minor	t_Products	1	15	Extra Empty Header
Issue3	Minor	t_Availabilities	1	8	Extra Empty Header
Issue4	Minor	t_Unavailabilities	1	8	Extra Empty Header
Issue5	Minor	t_Shift_Segments	1	6	Extra Empty Header
Issue6	Major	t_Shift_Segments	1	1	TimeOnly not formatted correctly, found 0.250000, format H:mm:ss
Issue7	Major	t_Shift_Segments	1	2	TimeOnly not formatted correctly, found 0.583333, format H:mm:ss
Issue8	Major	t_Shift_Segments	2	1	TimeOnly not formatted correctly, found 0.291667, format H:mm:ss
Issue9	Major	t_Shift_Segments	2	2	TimeOnly not formatted correctly, found 0.302083, format H:mm:ss
Issue10	Major	t_Shift_Segments	3	1	TimeOnly not formatted correctly, found 0.458333, format H:mm:ss
Issue11	Major	t_Shift_Segments	3	2	TimeOnly not formatted correctly, found 0.479167, format H:mm:ss
Issue12	Major	t_Shift_Segments	4	1	TimeOnly not formatted correctly, found 0.583333, format H:mm:ss
Issue13	Major	t_Shift_Segments	4	2	TimeOnly not formatted correctly, found 0.916667, format H:mm:ss
Issue14	Major	t_Shift_Segments	5	1	TimeOnly not formatted correctly, found 0.666667, format H:mm:ss
Issue15	Major	t_Shift_Segments	5	2	TimeOnly not formatted correctly, found 0.677083, format H:mm:ss
Issue16	Major	t_Shift_Segments	6	1	TimeOnly not formatted correctly, found 0.770833, format H:mm:ss
Issue17	Major	t_Shift_Segments	6	2	TimeOnly not formatted correctly, found 0.791667, format H:mm:ss
Issue18	Major	t_Shift_Segments	7	1	TimeOnly not formatted correctly, found 0.916667, format H:mm:ss
Issue19	Major	t_Shift_Segments	7	2	TimeOnly not formatted correctly, found 0.250000, format H:mm:ss
Issue20	Major	t_Shift_Segments	8	1	TimeOnly not formatted correctly, found 0.000000, format H:mm:ss
Issue21	Major	t_Shift_Segments	8	2	TimeOnly not formatted correctly, found 0.010417, format H:mm:ss
Issue22	Major	t_Shift_Segments	9	1	TimeOnly not formatted correctly, found 0.083333, format H:mm:ss
Issue23	Major	t_Shift_Segments	9	2	TimeOnly not formatted correctly, found 0.104167, format H:mm:ss
Issue24	Minor	t_Shift_Segments	10	0	First Column Empty
Issue25	Minor	t_Shift_Segments	11	0	First Column Empty
Issue26	Minor	t_Shift_Segments	12	0	First Column Empty
Issue27	Minor	t_Shift_Segments	13	0	First Column Empty
Issue28	Minor	t_Shift_Segments	14	0	First Column Empty
Issue29	Minor	t_Shift_Segments	15	0	First Column Empty
Issue30	Minor	t_Shift_Segments	16	0	First Column Empty
Issue31	Minor	t_Shift_Segments	17	0	First Column Empty
Issue32	Minor	t_Shift_Segments	18	0	First Column Empty
Issue33	Minor	t_Shift_Patterns	1	9	Extra Empty Header
Issue34	Minor	t_Shift_Patterns	7	0	First Column Empty
Issue35	Minor	t_Shift_Patterns	8	0	First Column Empty

Filter

Domain Model - Knowledge Graph



Solution for Berlin 08a - Shows Only 20% of Tasks in Model

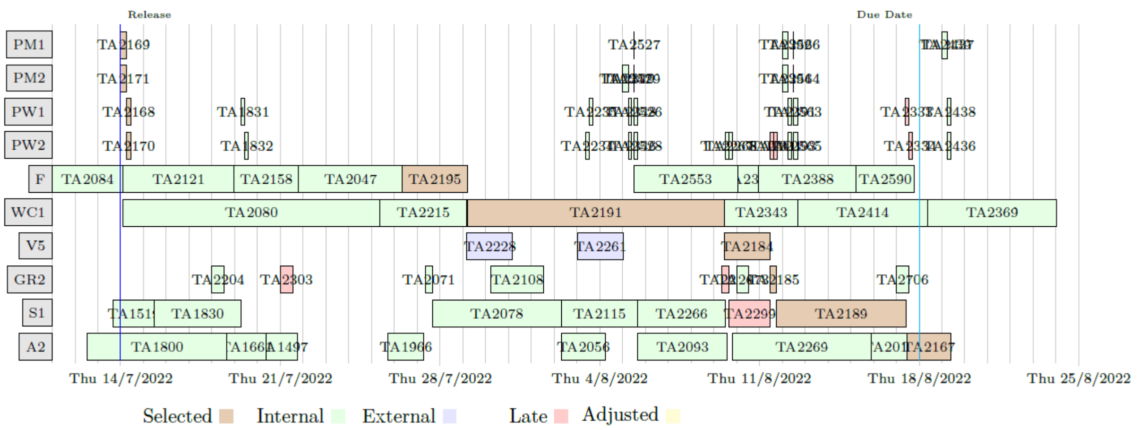


Implementation

- Requirement capture done inside project
- Data checking/cleaning most time consuming aspect
- Some specified functionality was rejected by Betriebsrat
- Built in Java
- Uses IBM's CPOptimizer back-end
- 120k LoC, 110k generated, 3k solver
- Outperforms both
  - Current in-house tool
  - Simulation based tool based on commercial simulator
- System installed at SE site, but not in daily use

Explaining Late Delivery

- Explain why some orders are delivered late
- Find root-cause, show schedule in context



Evaluation - KPIs

KPI	Baseline	Optimizer
OTD	> 80 %	92 %
Bottleneck machine utilization	99.5 %	100 %
Manufacturing defects	10-15 %	< 10 %
Scenarios in 8 hours	15-20	> 100,000

Conclusion by Siemens Energy

*"Within less than eight hours the **ASSISTANT** tools provided us thousands of manufacturing scenarios including different make-or-buy recommendations for making deliberate decisions on the way to proceed for strategic planning."*

from ASSISTANT final project review: Siemens Energy assessment

Summary

- Scheduling/Planning tool for manufacturing industry
- Developed as part of European ASSISTANT project
- Focused on key make-or-buy decisions
- Complex manufacturing process with alternative process paths
- Outperforms both current in-house tool and commercial simulator
- Key Technology: Optimization and Constraint Programming

Part XI

Where to Go from Here

Key Points

- We are working on a survey of the existing CP & Scheduling literature
- Considers over 1200 papers
- Current version of survey available at `https://hsimonis.github.io/pthg24`

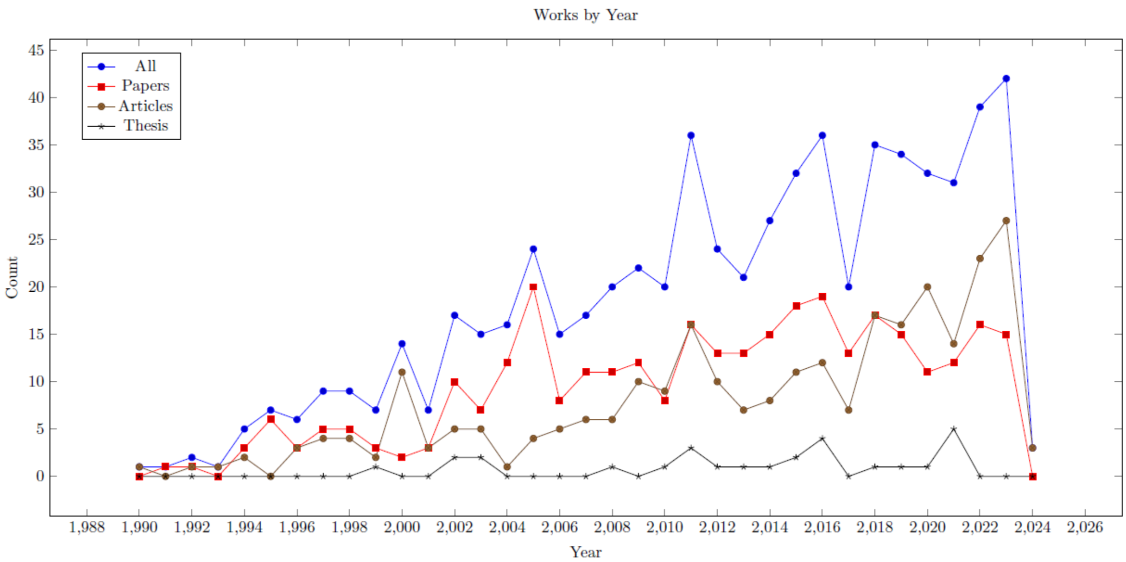
A Survey of the Existing Literature

- Joint work with Cemalettin Ozt urk, MTU
- What is out there
- Where to start
- Where to publish
- I'm interested in some specific topic, what is relevant

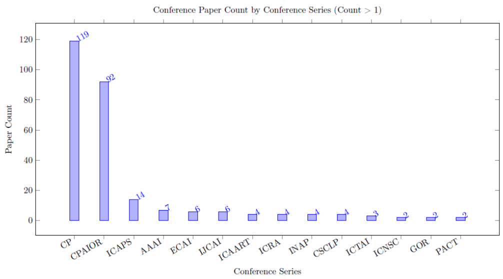
Methodology

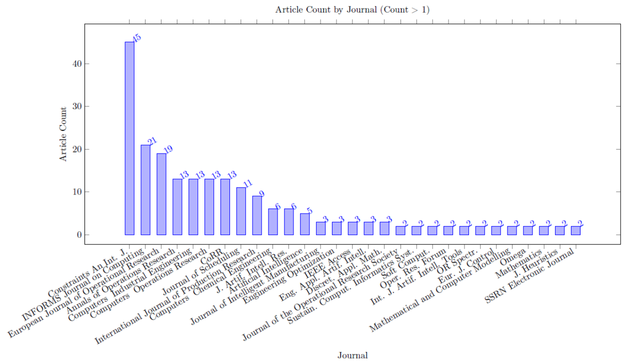
- Manually curated list of works, somewhat inclusive
- Starting with bibtex files
- Citation links through OpenCitations (open access)
- Content analysis on local copies of pdf files
- Closure of domain by analyzing missing cited and citing works
- Limited manual analysis of works (datasets, code)
- Results presented as LaTeX documents
- Open source analysis on git: `https://hsimonis.github.io/pthg24/`

Overall Analysis (Based on 671 Works)



Origin of Papers/Articles





Most Recent Articles

Table 5: Works from bibtex (Total 274)

Key	Authors	Title	LC	Cite	Year	Conference /Journal	Pages	Nr Cites	Nr Refs	b	c
ForbesHJST24 ForbesHJST24	M. Forbes, M. Harris, H. Jansen, F.A. van der Schoot, T. Taimre	Combining optimisation and simulation using logic-based Benders decomposition	Yes	[217]	2024	European Journal of Operational Research	15	0	26	1314	1496
PrataAN23 PrataAN23	Bruno A. Prata, Levi R. Abreu, Marcelo S. Nagano	Applications of constraint programming in production scheduling problems: A descriptive bibliometric analysis	Yes	[509]	2024	Results in Control and Optimization	17	0	0	1427	1497
abs-2402-00459 abs-2402-00459	S. Nguyen, Dhananjay R. Thiruvady, Y. Sun, M. Zhang	Genetic-based Constraint Programming for Resource Constrained Job Scheduling	Yes	[469]	2024	CoRR	21	0	0	1495	1498
AbreuNP23 AbreuNP23	Levi Ribeiro de Abreu, Marcelo Seido Nagano, Bruno A. Prata	A new two-stage constraint programming approach for open shop scheduling problem with machine blocking	Yes	[168]	2023	International Journal of Production Research	20	1	47	1243	1499
AbreuPNF23 AbreuPNF23	Levi R. Abreu, Bruno A. Prata, Marcelo S. Nagano, Jose M. Framinan	A constraint programming-based iterated greedy algorithm for the open shop with sequence-dependent processing times and makespan minimization	Yes	[3]	2023	Computers Operations Research	12	0	46	1244	1500
Adelgren2023 Adelgren2023	N. Adelgren, Christos T. Maravelias	On the utility of production scheduling formulations including record keeping variables	Yes	[7]	2023	Computers Industrial Engineering	12	0	43	1245	1501
AfsarVPG23 AfsarVPG23	S. Afsar, Camino R. Vela, Juan José Palacios, I. González-Rodríguez	Mathematical models and benchmarking for the fuzzy job shop scheduling problem	Yes	[8]	2023	Computers Industrial Engineering	14	0	50	1246	1502
AkramNHRSA23 AkramNHRSA23	Bilal Omar Akram, Nor Kamariah Noordin, F. Hashim, Mohd Fadlee A. Rasid, Mustafa Ismael Salman, Abdulrahman M. Abdulghani	Joint Scheduling and Routing Optimization for Deterministic Hybrid Traffic in Time-Sensitive Networks Using Constraint Programming	Yes	[13]	2023	IEEE Access	16	0	0	1248	1503
AlfieriGPS23 AlfieriGPS23	A. Alfieri, M. Garraffa, E. Pastore, F. Salassa	Permutation flowshop problems minimizing core waiting time and core idle time	Yes	[15]	2023	Computers Industrial Engineering	13	0	37	1249	1504
Caballero23 Caballero23	Jordi Coll Caballero	Scheduling through logic-based tools	Yes	[127]	2023	Constraints An Int. J.	1	0	0	1287	1505
CzerniachowskaWZ23 CzerniachowskaWZ23	K. Czerniachowska, R. Wichniarek, K. Żywicki	Constraint Programming for Flexible Flow Shop Scheduling Problem with Repeated Jobs and Repeated Operations	Yes	[159]	2023	Advances in Science and Technology Research Journal	14	0	0	1297	1506
FahimiQ23 FahimiQ23	H. Fahimi, C. Quimper	Overload-Checking and Edge-Finding for Robust Cumulative Scheduling	No	[207]	2023	INFORMS Journal on Computing	null	0	16	No	1507
Fatemi-AnarakiTFV23 Fatemi-AnarakiTFV23	S. Fatemi-Anaraki, R. Tavakkoli-Moghaddam, M. Foumani, B. Vahedi-Nouri	Scheduling of Multi-Robot Job Shop Systems in Dynamic Environments: Mixed-Integer Linear Programming and Constraint Programming Approaches	Yes	[212]	2023	Omega	15	7	60	1312	1508
GhasemiMH23 GhasemiMH23	S. Ghasemi, R. Tavakkoli-Moghaddam, M. Hamid	Operating room scheduling by emphasising human factors and dynamic decision-making styles: a constraint programming method	No	[242]	2023	International Journal of Systems Science: Operations Logistics	null	0	104	No	1509
GuoZ23 GuoZ23	P. Guo, J. Zhu	Capacity reservation for humanitarian relief: A logic-based Benders decomposition method with subgradient cut	Yes	[269]	2023	European Journal of Operational Research	29	0	112	1325	1510
GurPAE23 GurPAE23	S. Gür, M. Pinarbasi, Haci Mehmet Alakas, T. Eren	Operating room scheduling with surgical team: a new approach with constraint programming and goal programming	Yes	[270]	2023	Central Eur. J. Oper. Res.	25	1	40	1327	1511
IsikYA23 IsikYA23	Eyüp Ensar Isik, Seyda Topaloglu Yildiz, Özge Satir Akpunar	Constraint programming models for the hybrid flow shop scheduling problem and its extensions	Yes	[321]	2023	Soft Comput.	28	0	127	1350	1512
JuvinHL23a JuvinHL23a	C. Juvin, L. Houssin, P. Lopez	Logic-based Benders decomposition for the preemptive flexible job-shop scheduling problem	Yes	[331]	2023	Computers Operations Research	17	0	40	1355	1513
LacknerMMWW23 LacknerMMWW23	M. Lackner, C. Mrkvicka, N. Musliu, D. Walkiewicz, F. Winter	Exact methods for the Oven Scheduling Problem	Yes	[374]	2023	Constraints An Int. J.	42	0	32	1371	1514

Automatically Extracted Article Features

Table 6: Automatically Extracted ARTICLE Properties (Requires Local Copy)

Work	Pages	Concepts	Classification	Constraints	Prog Languages	CP Systems	Areas	Industries	Benchmarks	Algorithm	a	c
Laborie03 [369]	38	task, precedence, order, cmax, machine, job, activity, re-scheduling, setup-time, release-date, inventory, preempt, job-shop, resource, scheduling, make-span		cycle, table constraint, cumulative, disjunctive	C++	Ilog Scheduler			benchmark	edge-finding, not-last, energetic reasoning, not-first, time-tabling	1201	1731
LaborieRSV18 [372]	41	release-date, job-shop, resource, activity, precedence, sequence dependent setup, earliness, scheduling, machine, inventory, transportation, manpower, due-date, setup-time, batch process, order, tardiness, flow-shop, job, make-span, re-scheduling, task, distributed	psplib, parallel machine, RCPSP	alternative constraint, cumulative, noOverlap, disjunctive, span constraint, cycle, alwaysIn, endBeforeStart	C , Python, C++, Java	CHIP, Gecode, Ilog Solver, Cplex, Ilog Scheduler, OPL, Choco Solver, CPO	semiconductor railway, container terminal, satellite, robot, pipeline, aircraft, shipping line	chemical industry, petro-chemical industry	real-world, CSPlib, benchmark	edge-finding	1080	1610
LacknerMMWW23 [374]	42	release-date, batch process, setup-time, job, order, due-date, tardiness, scheduling, make-span, machine, task, lateness, job-shop, earliness	parallel machine, OSP, single machine	alternative constraint, disjunctive, bin-packing, noOverlap, cumulative, endBeforeStart		Chuffed, Cplex, OPL, CPO, OR-Tools, MiniZinc, Gurobi	semiconductor oven scheduling	electronics industry, steel industry, manufacturing industry	random instance, industrial partner, benchmark, instance generator, zenodo, real-life	time-tabling	984	1514
LammaMM97 [377]	15	job-shop, resource, scheduling, precedence, order, task, job, distributed, no-wait		circuit, disjunctive	C++, Prolog	ECLiPSe, OPL, CHIP	railway		real-life		1230	1760
LetortCB15 [385]	52	machine, make-span, job, precedence, resource, scheduling, task, order	psplib	cumulative, cycle, bin-packing	Java, Prolog	Choco Solver, CHIP, SICStus			generated instance, Roadef, benchmark, random instance	energetic reasoning, sweep, edge-finding	1110	1640
LiW08 [386]	18	precedence, activity, resource, completion-time, setup-time, make-span, scheduling, machine, preempt, job-shop, no preempt, job, re-scheduling, open-shop, due-date, task, order	RCPSP	disjunctive, cycle, bin-packing		Ilog Solver, OZ, Cplex, ECLiPSe, OPL, CHIP			real-world		1178	1708
LiessM08 [388]	12	preempt, resource, scheduling, machine, job, activity, precedence, job-shop, task, make-span, order, cmax	RCPSP, psplib	disjunctive, cumulative	C++	OZ			benchmark	edge-finding	1179	1709
LimtanyakulS12 [393]	32	release-date, scheduling, order, completion-time, job, resource, activity, tardiness, machine, due-date, precedence		table constraint, disjunctive, bin-packing, cumulative		OZ, Ilog Scheduler, Cplex	robot, automotive	automotive industry	random instance, real-life, generated instance, industrial partner, benchmark	not-last, energetic reasoning, not-first, edge-finding	1133	1663
LombardiM10a [402]	30	due-date, distributed, order, job, make-span, release-date, re-scheduling, task, completion-time, resource, activity, precedence, preempt, scheduling, machine	TCSP	cycle, span constraint, cumulative, disjunctive, table constraint	C	Cplex			real-world, benchmark, real-life	sweep	1160	1690

Manually Extracted Article Features

Table 4: Manually Defined PAPER Properties

Key	Title (Local Copy)	CP System	Bench	Links	Data Avail	Sol Avail	Code Avail	Related To	Classification	Constraints	a	b
AalianPG23 AalianPG23 [1]	Optimization of Short-Term Underground Mine Planning Using Constraint Programming	CP Opt	real-world	1	n		n			?	1	325
Bit-Monnot23 Bit-Monnot23 [96]	Enhancing Hybrid CP-SAT Search for Disjunctive Scheduling	ARIES CP Opt OR-Tools Mistral	real-world, github, benchmark	1	y		y	-	JSSP OSSP	-	2	371
EfthymiouY23 EfthymiouY23 [194]	Predicting the Optimal Period for Cyclic Hoist Scheduling Problems	OR-Tools	benchmark, random instance, generated instance, real-life, industrial instance	3	n		n	-	CHSP	-	3	415
JuvinHHL23 JuvinHHL23 [328]	An Efficient Constraint Programming Approach to Preemptive Job Shop Scheduling	CP Opt Mistral	supplementary material, github, benchmark	6	ref		y		PJSSP	endBeforeStart span noOverlap	4	476
JuvinHL23 JuvinHL23 [330]	Constraint Programming for the Robust Two-Machine Flow-Shop Scheduling Problem with Budgeted Uncertainty	CP Opt Cplex	real-world	0	ref		n	-	Perm FSSP	endBeforeStart noOverlap sameSequence	5	477
KameugneFND23 KameugneFND23 [336]	Horizontally Elastic Edge Finder Rule for Cumulative Constraint Based on Slack and Density	?	benchmark	5	BL PSPlib		n	-	RCPSPs	cumulative	6	480
KimCMLLP23 KimCMLLP23 [345]	Iterated Greedy Constraint Programming for Scheduling Steelmaking Continuous Casting	Gurobi OR-Tools	real-world, benchmark, zenodo	0	y		n	-	SCC	alternative noOverlap	7	485
Mehdizadeh-Somarin23 Mehdizadeh-Somarin23 [430]	A Constraint Programming Model for a Reconfigurable Job Shop Scheduling Problem with Machine Availability	CP Opt	random instance	0	n		n	-	JSSP RMS	alternative endBeforeStart noOverlap	8	529
PerezGSL23 PerezGSL23 [496]	A Constraint Programming Model for Scheduling the Unloading of Trains in Ports	custom	real-world, generated instance	0	n		n	-	SUTP	table disjunctive	9	553
PovedaAA23 PovedaAA23 [506]	Partially Preemptive Multi Skill/Mode Resource-Constrained Project Scheduling with Generalized Precedence Relations and Calendars	CP Opt MiniZinc Chuffed	real-world, github, benchmark, industrial instance, real-life	4	y		y		PP-MS-MMRCPSP/max-cal		10	557
SquillaciPR23 SquillaciPR23 [564]	Scheduling Complex Observation Requests for a Constellation of Satellites: Large Neighborhood Search Approaches	Cplex Studio	github, benchmark	2	y		n	-	EOSP	?	11	584
TardivoDFMP23 TardivoDFMP23 [575]	Constraint Propagation on GPU: A Case Study for the Cumulative Constraint	MiniCPP MiniZinc	bitbucket, github, benchmark, real-world	9	PSPLib BL Pack		y	-	RCPSP	cumulative	12	590
TasselGS23 TasselGS23 [576]	An End-to-End Reinforcement Learning Approach for Job-Shop Scheduling Problems Based on Constraint Programming	custom Choco	industrial instance, real-world, supplementary material, github, benchmark	0	ref		y	-	JSSP	noOverlap	13	591
WangB23 WangB23 [629]	Dynamic All-Different and Maximal Cliques Constraints for Fixed Job Scheduling	FaCiLe	real-world, random instance	0	(y)		n	[628]	FJS	-	14	620
YuraszeckMC23 YuraszeckMC23 [649]	A competitive constraint programming approach for the group shop scheduling problem	CP Opt	github, benchmark	0	ref		n	-	GSSP	noOverlap endBeforeStart	15	633

Extracted Features: Application Areas

Table 16: Works for Concepts of Type ApplicationAreas

Type	Keyword	High	Medium	Low
ApplicationAreas	COVID	GuoZ23 [269]	GeibingerKKMMW21 [234]	Fatemi-AnarakiTFV23 [212], Mehdizadeh-Somarin23 [430], GurPAE23 [270], JuvinHL23a [331], OujanaAYB22 [487], Lemos21 [381]
ApplicationAreas	HVAC	LimHTB16 [390], LimBTBB15 [391], GrimesIOS14 [260]		
ApplicationAreas	agriculture			AkramNHRSA23 [13], BenderWS21 [84], HamPK21 [275], Astrand21 [35], QinWSLS21 [511], Astrand0F21 [36], MejiaY20 [431]
ApplicationAreas	aircraft	PohlAK22 [502], WangB20 [628], TranDRFWOVB16 [596], Fahimi16 [205], BajestaniB13 [42], LombardiM12 [405], BajestaniB11 [41], FrankK05 [219], ArtiouchineB05 [34], Simonis99 [558]	WangB23 [629], GombolayWS18 [253], Ham18 [273], Simonis07 [559], SakkoutW00 [529], Simonis95a [556]	PrataAN23 [509], PovedaAA23 [506], Adelgren2023 [7], EtminaniesfahaniGNMS22 [202], ElciOH22 [195], ZarandiASC20 [654], HauderBRPA20 [283], abs-1902-09244 [282], Hooker19 [312], LaborieRSV18 [372], HookerH17 [314], TranAB16 [594], Lombardi10 [398], Laborie09 [370], KovacsB08 [355], KrogtLPHJ07 [608], MartinPY01 [427], SimonisCK00 [560], GruianK98 [264], Darby-DowmanLMZ97 [163], Wallace96 [625], Simonis95 [557], SimonisC95 [561]
ApplicationAreas	automotive		GuoZ23 [269], YuraszeckMPV22 [650], EmdeZD22 [199], Groleaz21 [261], LimtanyakulS12 [393], SunLYL10 [567], Lombardi10 [398], BarlattCG08 [52], SchildW00 [532]	PovedaAA23 [506], NaderiRR23 [460], CzerniachowskaWZ23 [159], NaderiBZ22 [457], NaderiBZ22a [456], AntuoriHHEN21 [22], HubnerGSV21 [318], AbreuAPNM21 [166], KoehlerBFFHPSSS21 [348], VlkHT21 [623], BarzegaranZP20 [61], GeibingerMM19 [236], abs-1911-04766 [235], BonfiettiZLM16 [113], Siala15a [552], SchnellH15 [533], AlesioNBG14 [181], HarjunkoskiMBC14 [279], BeniniBGM06 [88], KovacsV06 [360], Wallace96 [625]
ApplicationAreas	cable tree	KoehlerBFFHPSSS21 [348]		
ApplicationAreas	car manufacturing		AntuoriHHEN21 [22]	BeldiceanuC94 [78]
ApplicationAreas	container terminal	QinDCS20 [512], SacramentoSP20 [526]	LaborieRSV18 [372]	abs-2312-13682 [497], PerezGSL23 [496], TouatBT22 [592], CauwelaertDS20 [142], WallaceY20 [627], ZarandiASC20 [654], FallahiAC20 [209], Hooker19 [312], CauwelaertDMS16 [140], Dejemeppe16 [172], DejemeppeCS15 [173], NovasH12 [476], CorreaLR07 [158], LimRX04 [389]
ApplicationAreas	crew-scheduling	ZarandiASC20 [654], PourDERB18 [505]	BourreauGGLT22 [118], Zahout21 [652], GombolayWS18 [253], Mason01 [429], Touraivane95 [593]	NaderiRR23 [460], WangB23 [629], Adelgren2023 [7], EtminaniesfahaniGNMS22 [202], NaderiBZ22a [456], NaderiBZ22 [457], HeinzNVH22 [295], ElciOH22 [195], Lemos21 [381], MokhtarzadehTNF20 [443], TangLWSK18 [574], HookerH17 [314], DoulabiRP16 [190], LipovetzkyBPS14 [394], HachemiGR11 [272], MilanoW09 [441], WuBB09 [643], MilanoW06 [440], BeldiceanuC02 [79], JainG01 [323], SimonisCK00 [560]
ApplicationAreas	dairies			Bartak02 [54], Bartak02a [53]
ApplicationAreas	dairy	EscobetPQPRA19 [201]	PrataAN23 [509], HarjunkoskiMBC14 [279]	Groleaz21 [261]
ApplicationAreas	datacenter	HermenierDL11 [300]		Zahout21 [652], GalleguillosKSB19 [225], Madi-WambaLOBM17 [418], Letort13 [382], IfrimOS12 [320], LetortBC12 [383]
ApplicationAreas	datacentre		HurleyOS16 [319]	
ApplicationAreas	day-ahead market			
ApplicationAreas	deep space			HebrardALLCMR22 [285]
ApplicationAreas	drone	MontemanniD23a [446], MontemanniD23 [447], Ham18 [273]		GuoZ23 [269], JuvinHL23a [331], Adelgren2023 [7], ShaikhK23 [547], EmdeZD22 [199], Astrand21 [35], Astrand0F21 [36], AntuoriHHEN21 [22], ZarandiASC20 [654], Ham18a [274]

Prolific Authors

Table 8: Co-Authors of Articles/Papers

Author	Nr Works	Nr Cites	Entries
J. Christopher Beck	49	701	LuoB22 [416], ZhangBB22 [658], TangB20 [573], RoshanaeiBAUB20 [521], TranPZLDB18 [597], TranVNB17 [599], TranVNB17a [600], CohenHB17 [154], BoothNB16 [114], KuB16 [365], TranAB16 [594], TranWDRFOVB16 [601], LuoVLBM16 [415], TranDRFWOVB16 [596], BajestaniB15 [43], KoschB14 [353], TerekhovTDB14 [581], LouieVNB14 [412], HeinzSB13 [294], HeinzKB13 [291], BajestaniB13 [42], TranTDB13 [598], HeinzB12 [290], TerekhovDOB12 [580], TranB12 [595], ZarandiB12 [213], KovacsB11 [356], BeckFW11 [66], HeckmanB11 [289], BajestaniB11 [41], WuBB09 [643], BidotVLB09 [94], CarchraeB09 [131], WatsonB08 [632], KovacsB08 [355], BeckW07 [73], Beck07 [64], KovacsB07 [354], Beck06 [63], CarchraeBF05 [132], WuBB05 [642], BeckW05 [72], BeckW04 [71], BeckR03 [70], BeckPS03 [69], BeckF00 [68], Beck99 [62], BeckF98 [67], BeckDF97 [65]
Michela Milano	31	297	BorghesiBLMB18 [115], BonfiettiZLM16 [113], BridiBLMB16 [120], BridiLBBM16 [121], LombardiBM15 [399], BartoliniBBLM14 [60], BonfiettiLM14 [111], BonfiettiLBM14 [109], BonfiettiLM13 [110], LombardiM13 [406], LombardiMB13 [407], LombardiM12 [405], BonfiettiLBM12 [108], LombardiM12a [404], BonfiettiM12 [112], BonfiettiLBM11 [107], LombardiBMB11 [400], BeniniLMR11 [90], Milano11 [438], LombardiM10 [403], LombardiM10a [402], LombardiMRB10 [408], LombardiM09 [401], RuggieroBBMA09 [525], MilanoW09 [441], BeniniLMR08 [89], BeniniBGM06 [88], MilanoW06 [440], MilanoORT02 [439], LammaMM97 [377], BrusoniCLMMT96 [123]
Andreas Schutt	27	322	YangSS19 [644], KreterSSZ18 [364], GoldwaserS18 [251], MusliuSS18 [455], KreterSS17 [363], YoungFS17 [646], GoldwaserS17 [250], SchuttS16 [543], SzerediS16 [570], KreterSS15 [362], EvenSH15 [203], EvenSH15a [204], SchuttFSW15 [542], ThiruvadyWGS14 [585], GuSSWC14 [266], SchuttFS13 [537], SchuttFS13a [536], GuSS13 [265], SchuttFSW13 [541], ChuGNSW13 [147], SchuttCSW12 [535], SchuttFSW11 [540], Schutt11 [534], SchuttW10 [544], abs-1009-0347 [539], SchuttFSW09 [538], SchuttWS05 [545]
Michele Lombardi	25	194	BorghesiBLMB18 [115], CauwelaertLS18 [141], BonfiettiZLM16 [113], BridiBLMB16 [120], BridiLBBM16 [121], LombardiBM15 [399], BartoliniBBLM14 [60], BonfiettiLM14 [111], BonfiettiLBM14 [109], BonfiettiLM13 [110], LombardiM13 [406], LombardiMB13 [407], LombardiM12 [405], BonfiettiLBM12 [108], LombardiM12a [404], BonfiettiLBM11 [107], LombardiBMB11 [400], BeniniLMR11 [90], LombardiM10 [403], LombardiM10a [402], Lombardi10 [398], LombardiMRB10 [408], LombardiM09 [401], BeniniLMR08 [89], HoeveGSL07 [609]
Peter J. Stuckey	24	453	YangSS19 [644], DemirovicS18 [177], KreterSSZ18 [364], MusliuSS18 [455], KreterSS17 [363], SchuttS16 [543], BlomPS16 [100], KreterSS15 [362], BurtLPS15 [124], SchuttFSW15 [542], BlomBPS14 [99], LipovetzkyBPS14 [394], GuSSWC14 [266], SchuttFS13 [537], SchuttFS13a [536], GuSS13 [265], SchuttFSW13 [541], SchuttCSW12 [535], GuSW12 [267], SchuttFSW11 [540], BandaSC11 [170], abs-1009-0347 [539], SchuttFSW09 [538], OhrimenkoSC09 [483]
John N. Hooker	19	1316	ElciOH22 [195], Hooker19 [312], Hooker17 [311], HookerH17 [314], HechingH16 [288], CireCH16 [150], HarjunkoskiMBC14 [279], CireCH13 [149], CobanH11 [153], CobanH10 [152], Hooker10 [310], Hooker07 [309], Hooker06 [308], Hooker05 [306], Hooker05a [307], Hooker04 [305], HookerO03 [313], HookerY02 [315], Hooker00 [304]
Emmanuel Hebrard	17	71	JuvinHHL23 [328], HebrardALLCMR22 [285], AntuoriHHEN21 [22], ArtiguesHQT21 [32], GodetLHS20 [247], AntuoriHHEN20 [21], HebrardHJMPV16 [286], SimoninAHL15 [555], SialaAH15 [553], GrimesH15 [258], BessiereHMQW14 [93], SimoninAHL12 [554], BillautHL12 [95], GrimesH11 [257], GrimesH10 [256], GrimesHM09 [259], HebrardTW05 [287]
Pierre Lopez	17	90	JuvinHHL23 [328], JuvinHL23a [331], JuvinHL23 [330], HebrardALLCMR22 [285], JuvinHL22 [329], Polo-MejiaALB20 [503], NattafHKAL19 [466], NattafAL17 [463], NattafALR16 [464], SimoninAHL15 [555], NattafAL15 [462], SimoninAHL12 [554], BillautHL12 [95], LahimerLH11 [375], TrojetHL11 [602], LopezAKYG00 [410], TorresL00 [591]
Christian Artigues	16	203	PovedaAA23 [506], PohlAK22 [502], HebrardALLCMR22 [285], ArtiguesHQT21 [32], Polo-MejiaALB20 [503], NattafHKAL19 [466], NattafAL17 [463], NattafALR16 [464], SimoninAHL15 [555], NattafAL15 [462], SialaAH15 [553], SimoninAHL12 [554], NeronABCDD06 [481], DemasseyAM05 [176], ArtiguesBF04 [30], ArtiguesR00 [33]
Pierre Schaus	15	79	CauwelaertDS20 [142], ThomasKS20 [586], HoundjiSW19 [316], CappartTSR18 [130], CauwelaertLS18 [141], CappartS17 [129], CauwelaertDMS16 [140], DejemeppeCS15 [173], GayHLS15 [229], GayHS15 [230], GayHS15a [231], HoundjiSWD14 [317], GaySS14 [232], SchausHMCMD11 [531], SchausD08 [530]
Helmut Simonis	15	154	ArmstrongGOS22 [27], ArmstrongGOS21 [26], AntunesABD20 [20], AntunesABD18 [19], HurleyOS16 [319], GrimesIOS14 [260], IfrimOS12 [320], SimonisH11 [562], Simonis07 [559], SimonisCK00 [560], Simonis99 [558], SimonisC95 [561], Simonis95 [557], Simonis95a [556], DincbasSH90 [184]
Nicolas Beldiceanu	13	274	Madi-WambaLOBM17 [418], Madi-WambaB16 [417], LetortCB15 [385], LetortCB13 [384], LetortBC12 [383], ClercqPBJ11 [151], BeldiceanuCDP11 [80], BeldiceanuCP08 [81], PoderB08 [500], BeldiceanuP07 [82], PoderBS04 [501], BeldiceanuC02 [79], AggounB93 [9]
Luca Benini	13	146	BorghesiBLMB18 [115], BridiBLMB16 [120], BridiLBBM16 [121], BonfiettiLBM14 [109], LombardiMB13 [407], BonfiettiLBM12 [108], BonfiettiLBM11 [107], LombardiBMB11 [400], BeniniLMR11 [90], LombardiMRB10 [408], RuggieroBBMA09 [525], BeniniLMR08 [89], BeniniBGM06 [88]
Philippe Laborie	12	513	LunardiBLRV20 [413], LaborieRSV18 [372], Laborie18a [371], MelgarejoLS15 [11], VilimLS15 [621], Laborie09 [370], BidotVLB09 [94], BaptisteLPN06 [47], NeronABCDD06 [481], GodardLN05 [245], Laborie03 [369], FocacciLN00 [215]
Philippe Baptiste	11	403	BaptisteB18 [46], Baptiste09 [45], BaptisteLPN06 [47], NeronABCDD06 [481], ArtiouchineB05 [34], Baptiste02 [44], BaptistePN01 [50], BaptisteP00 [49], PapaB98 [492], BaptisteP97 [48], PapeB97 [491]
Roman Barták	11	88	SvancaraB22 [569], JelinekB16 [325], BartakV15 [59], Bartak14 [55], BartakS11 [57], BartakCS10 [56], BartakSR10 [58], VilimBC05 [620], VilimBC04 [619], Bartak02 [54], Bartak02a [53]

Limitations

- Limited coverage by OpenCitations
- Difficult to have local access to some publication types (book, incollection)
- Heavily biased towards publications in English
- More powerful NLP analysis of works possible?

Problem: Count for Most Cited Papers

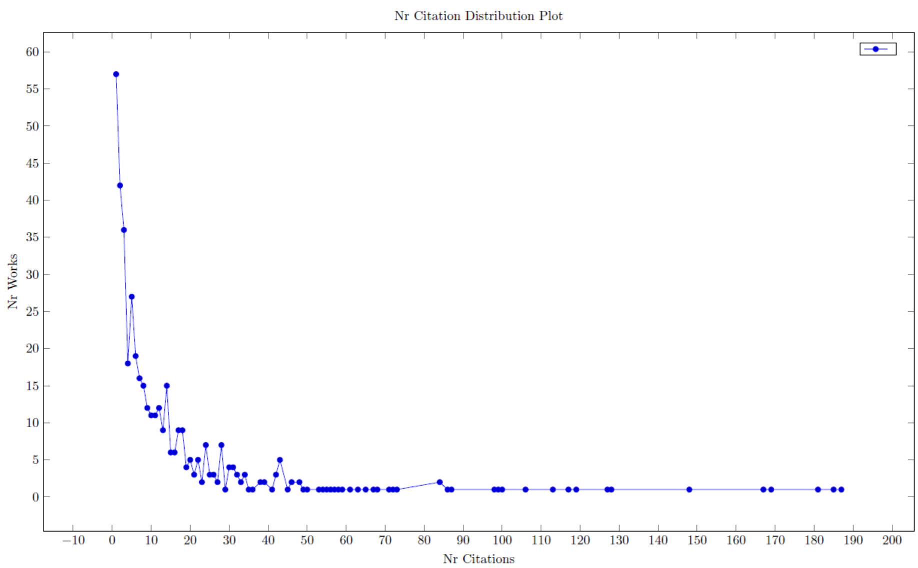
Table 9: Works from bibtex (Total 30)

Key	Authors	Title	LC	Cite	Year	Conference /Journal	Pages	Nr Cites	Nr Refs	b	c
JainM99 JainM99	A. Jain, S. Meeran	Deterministic job-shop scheduling: Past, present and future	Yes	[322]	1999	European Journal of Operational Research	45	490	150	1352	1753
HarjunkoskiMBC14 HarjunkoskiMBC14	I. Harjunkoski, Christos T. Maravelias, P. Bongers, Pedro M. Castro, S. Engell, Ignacio E. Grossmann, John N. Hooker, C. Méndez, G. Sand, J. Wassick	Scope for industrial applications of production scheduling models and solution methods	Yes	[279]	2014	Computers Chemical Engineering	33	381	176	1335	1649
BlazewiczDP96 BlazewiczDP96	J. Błażewicz, W. Domschke, E. Pesch	The job shop scheduling problem: Conventional and new solution techniques	Yes	[125]	1996	European Journal of Operational Research	33	344	127	1278	1762
HookerO03 HookerO03	John N. Hooker, G. Ottosson	Logic-based Benders decomposition	Yes	[313]	2003	Mathematical Programming	28	317	0	1347	1729
BaptistePN01 BaptistePN01	P. Baptiste, Claude Le Pape, W. Nuijten	Constraint-Based Scheduling	No	[50]	2001	Book	null	296	0	No	n/a
JainG01 JainG01	V. Jain, Ignacio E. Grossmann	Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems	Yes	[323]	2001	INFORMS Journal on Computing	19	279	23	1351	1738
AggounB93 AggounB93	A. Aggoun, N. Beldiceanu	Extending CHIP in order to solve complex scheduling and placement problems	Yes	[9]	1993	Mathematical and Computer Modelling	17	187	11	1247	1767
Hooker00 Hooker00	John N. Hooker	Logic Based Methods for Optimization: Combining Optimization and Constraint Satisfaction	No	[304]	2000	Book	null	185	0	No	n/a
Hooker07 Hooker07	John N. Hooker	Planning and Scheduling by Logic-Based Benders Decomposition	Yes	[309]	2007	Operations Research	29	181	19	1345	1715
HarjunkoskiG02 HarjunkoskiG02	I. Harjunkoski, Ignacio E. Grossmann	Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods	Yes	[278]	2002	Computers Chemical Engineering	20	169	11	1334	1733
BeldiceanuC94 BeldiceanuC94	N. Beldiceanu, E. Contejean	Introducing Global Constraints in CHIP	Yes	[78]	1994	Mathematical and Computer Modelling	27	167	8	1271	1765
LaborieRSV18 LaborieRSV18	P. Laborie, J. Rogerie, P. Shaw, P. Vilím	IBM ILOG CP optimizer for scheduling - 20+ years of scheduling with constraints at IBM/ILOG	Yes	[372]	2018	Constraints An Int. J.	41	148	35	1370	1610
Laborie03 Laborie03	P. Laborie	Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results	Yes	[369]	2003	Artificial Intelligence	38	128	10	1369	1731
OhrimenkoSC09 OhrimenkoSC09	O. Ohrimenko, Peter J. Stuckey, M. Codish	Propagation via lazy clause generation	Yes	[483]	2009	Constraints An Int. J.	35	127	15	1417	1702
KuB16 KuB16	W. Ku, J. Christopher Beck	Mixed Integer Programming models for job shop scheduling: A computational analysis	Yes	[365]	2016	Computers Operations Research	9	119	17	1367	1630
Rodriguez07 Rodriguez07	J. Rodriguez	A constraint programming model for real-time train scheduling at junctions	Yes	[520]	2007	Transportation Research Part B: Methodological	15	117	6	1430	1716
LiW08 LiW08	H. Li, K. Womer	Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm	Yes	[386]	2008	Journal of Scheduling	18	113	31	1374	1708
CorreaLR07 CorreaLR07	Ayoub Insa Corréa, A. Langevin, L. Rousseau	Scheduling and routing of automated guided vehicles: A hybrid approach	Yes	[158]	2007	Computers Operations Research	20	106	20	1296	1714
MengZRZL20 MengZRZL20	L. Meng, C. Zhang, Y. Ren, B. Zhang, C. Lv	Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem	Yes	[435]	2020	Computers Industrial Engineering	13	100	62	1393	1574
BensanaLV99 BensanaLV99	E. Bensana, M. Lemaître, G. Verfaillie	Earth Observation Satellite Management	Yes	[91]	1999	Constraints An Int. J.	7	99	0	1276	1752

OpenCitation Count Compared to Google Scholar

Key	Type	Google	OC	Ratio
JainM99	article	1116	490	2.28
HarjunkoskiMBC14	article	588	381	1.54
BlazewiczDP96	article	796	344	2.31
BaptistePN01	book	1039	296	3.51
AggounB93	article	502	187	2.68
LaborieRSV18	article	309	148	2.09
BensanaLV99	article	251	99	2.54
DincbasSH90	article	271	86	3.15
Thorsteinsson01	paper	205	67	3.06
DincbasSH88	paper	287	0	☹

Problem: Citation Count Distribution



Summary

- Use the survey to find
  - Most important works on Constraint Based Scheduling
  - Specialized papers on the constraint reasoning for scheduling
  - Works in specific application domains or specific industries