

# Introduction and Motivation

**Helmut Simonis**

## Constraint Based Production Scheduling



## Licence



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

This license requires that reusers give credit to the creator. It allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, for noncommercial purposes only. If others modify or adapt the material, they must license the modified material under identical terms.



# Acknowledgments



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

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

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

## Key Points



- 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 be 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



```
1  "order": [
2    {
3      "product": "Prod0",
4      "process": "Process 0",
5      "due": 5449,
6      "releaseDate": "1/10/2024 00:00",
7      "release": 0,
8      "qty": 7,
9      "dueDate": "19/10/2024 22:05",
10     "name": "Order0",
11     "earlinessWeight": 1,
12     "latenessWeight": 1
13   },

```

ENTIRE EDIH

Production Scheduling

Slide 8

## Orders Loaded



| Order X |    |         |           |     |       |                  |         |                 |                |                 |
|---------|----|---------|-----------|-----|-------|------------------|---------|-----------------|----------------|-----------------|
| 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             |

ENTIRE EDIH

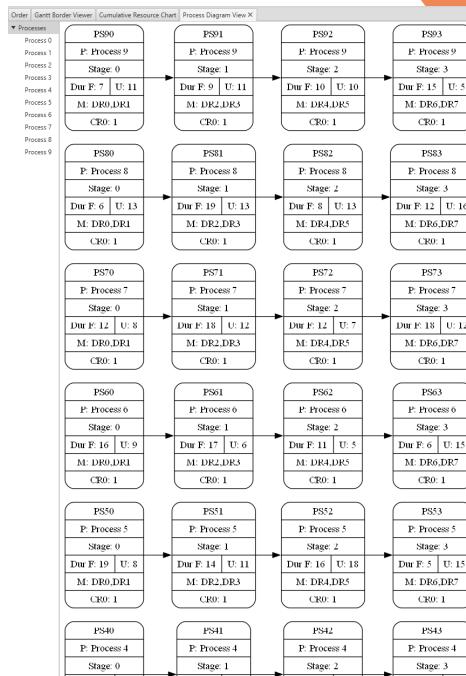
Production Scheduling

Slide 9

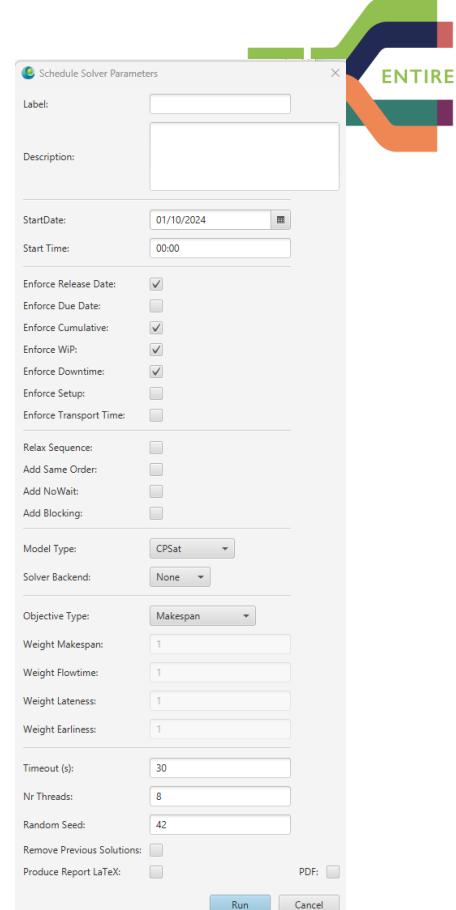
# Process Diagram



- 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,...)

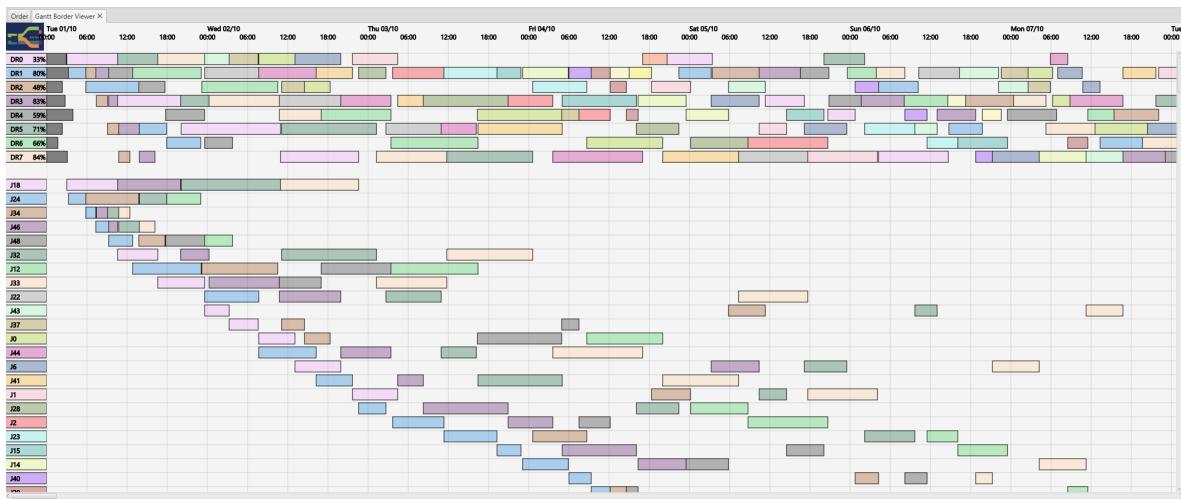


## Selecting Solver Options



- 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

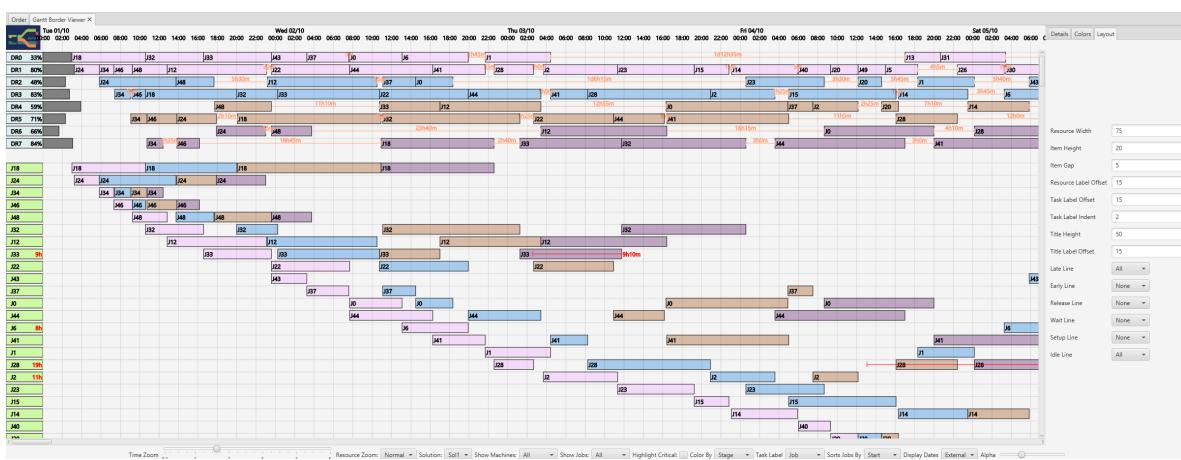


ENTIRE EDIH

Production Scheduling

Slide 12

# Adapted Gantt Chart

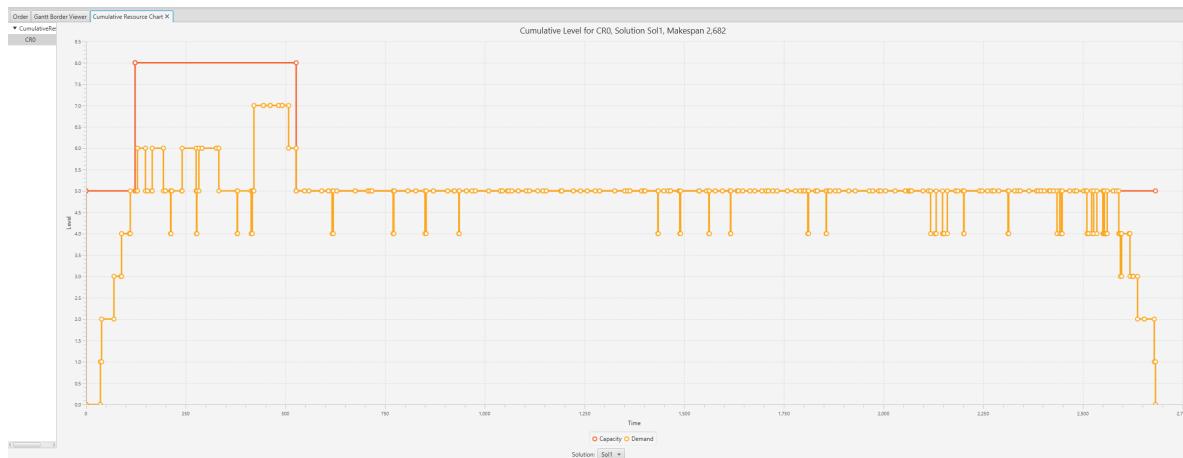


ENTIRE EDIH

Production Scheduling

Slide 13

# Cumulative Resource Chart

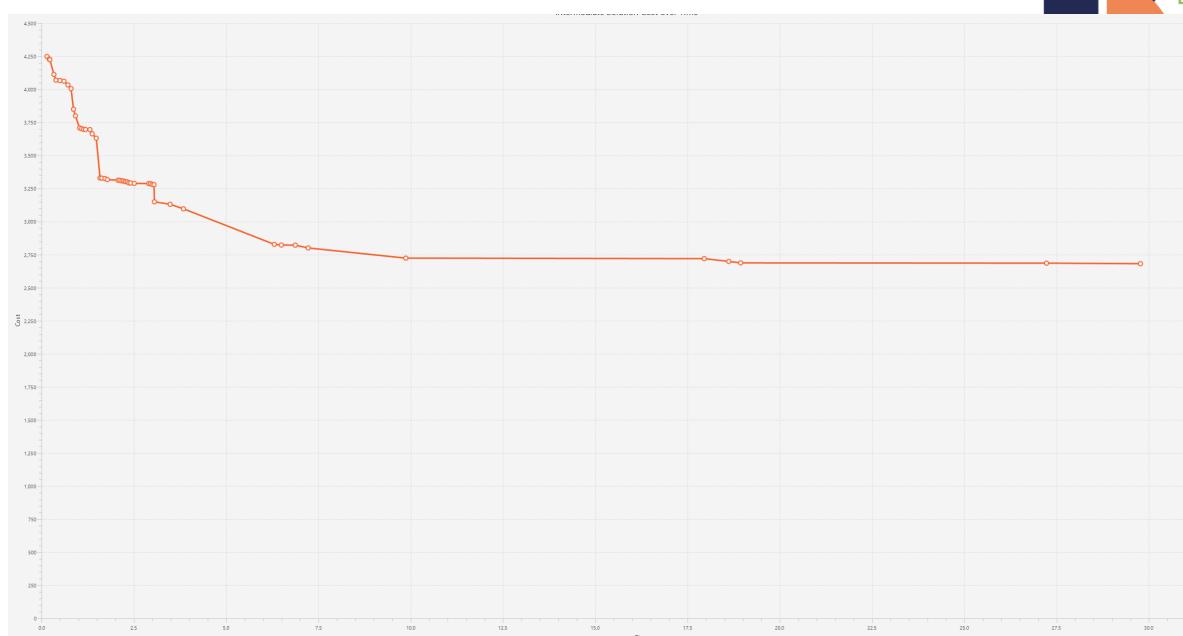


ENTIRE EDIH

Production Scheduling

Slide 14

## Intermediate Solutions Found



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

ENTIRE EDIH

Production Scheduling

Slide 15

# 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



*"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/>

ENTIRE EDIH

Production Scheduling

Slide 20

## Other Technologies



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

ENTIRE EDIH

Production Scheduling

Slide 21

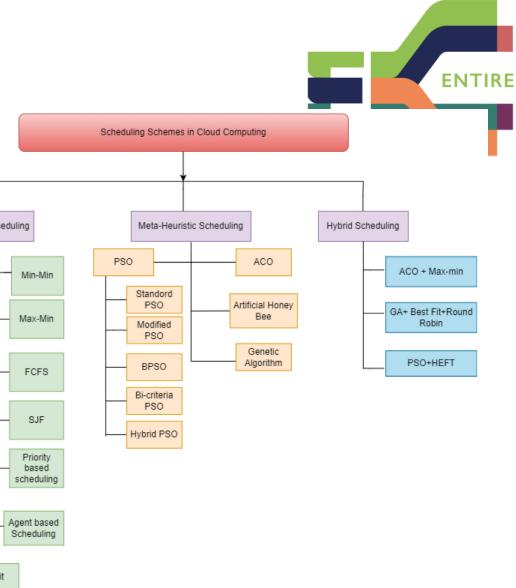
# Heuristics

- 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

ENTIRE EDIH

Production Scheduling

Slide 22



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>

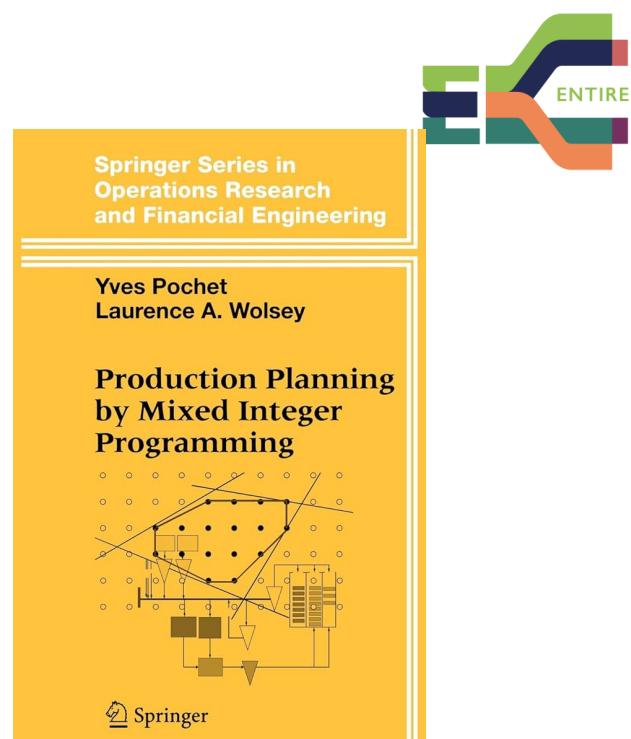
# Integer Programming

- Sub-class of constraint 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

ENTIRE EDIH

Production Scheduling

Slide 23

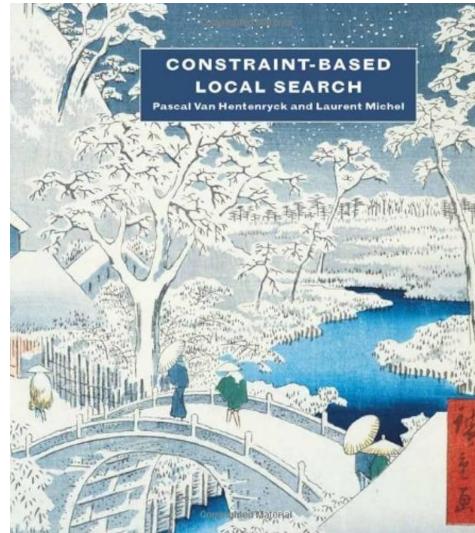


<https://link.springer.com/book/10.1007/0-387-33477-7>

# 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/](https://mitpress.mit.edu/9780262220774/constraint-based-local-search/)

ENTIRE EDIH

Production Scheduling

Slide 24

## 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               | -                           |

ENTIRE EDIH

Production Scheduling

Slide 26

# 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
- 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