



Funded by the  
European Union  
NextGenerationEU



European  
Digital Innovation  
Hubs Network



# Objectives

**Helmut Simonis**

## Constraint Based Production Scheduling



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



- 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

# Outline



## Optimal vs. Good Solutions

### Cost vs. Profit Based Objectives

Objective Types

Other Quality Vectors

Key Performance Indicators

Interactive Scheduling

# 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



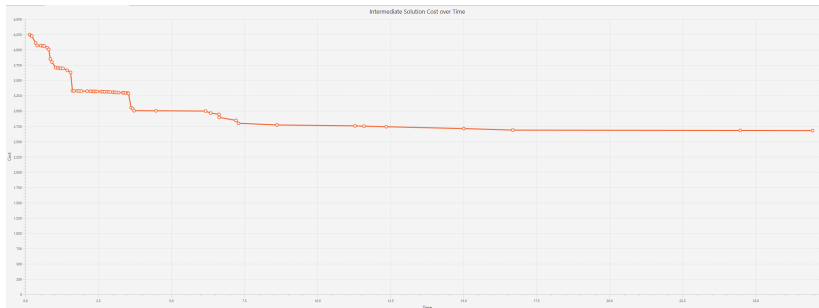
- A lot of objectives aim to reduce cost of production
- This is not always a good thing
  - Doing nothing costs nothing
- But defining the profit obtained by a schedule is not easy
- Many intangible factors weigh in
  - Happiness of the customers (which customers are unhappy, does it matter?)
  - Happiness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



- 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?

# Outline



Optimal vs. Good Solutions

## Objective Types

- Makespan

- Flowtime

- Lateness

- On-Time

- Hybrid

- Comparison

- Resource Levels

Other Quality Vectors

Key Performance Indicators

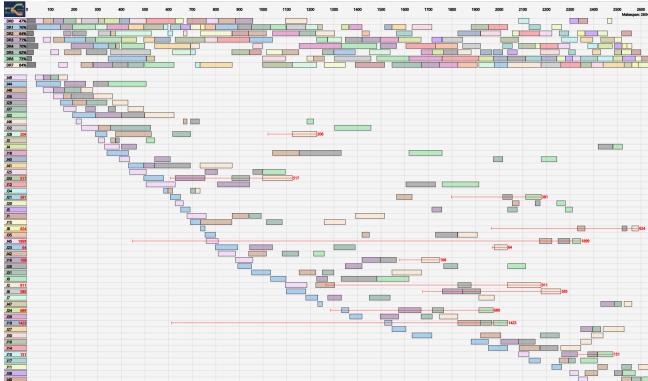
# Setting the Objective



- 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

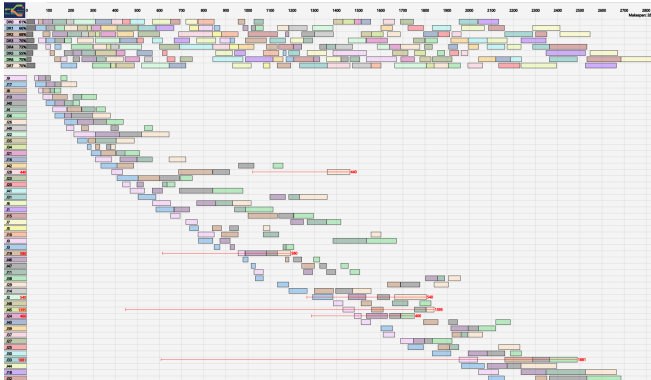
|                   |          |
|-------------------|----------|
| Objective Type:   | Makespan |
| Weight Makespan:  | 1        |
| Weight Flowtime:  | 1        |
| Weight Lateness:  | 1        |
| Weight Earliness: | 1        |

# Makespan ✓



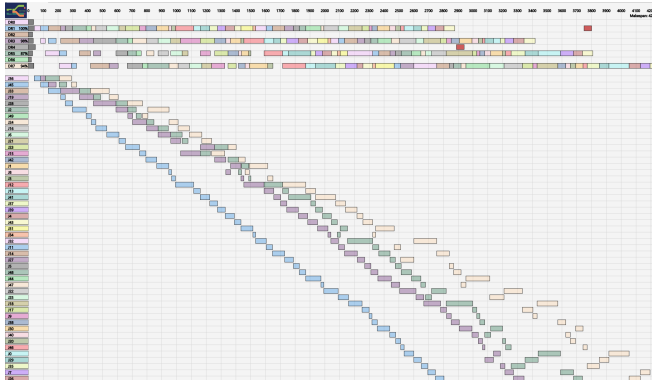
- Minimize the overall project end
- Very traditional objective in scheduling
  - Justified in project scheduling
  - Not so clearly justified in manufacturing
- A number of jobs are significantly late

# Flowtime ✓



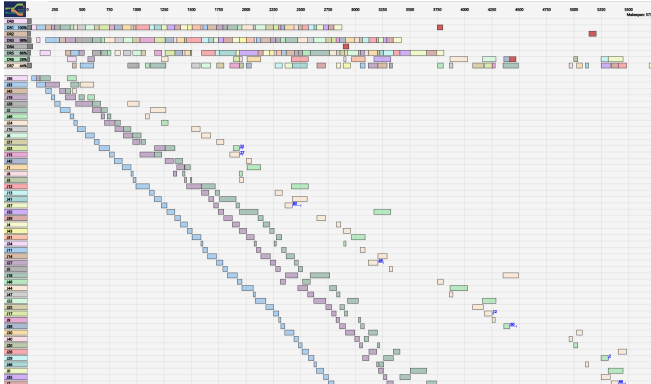
- Minimize the sum of job ends
- Prefer any machine to end early
- Not always easy to find good solutions

# Total Lateness ✓



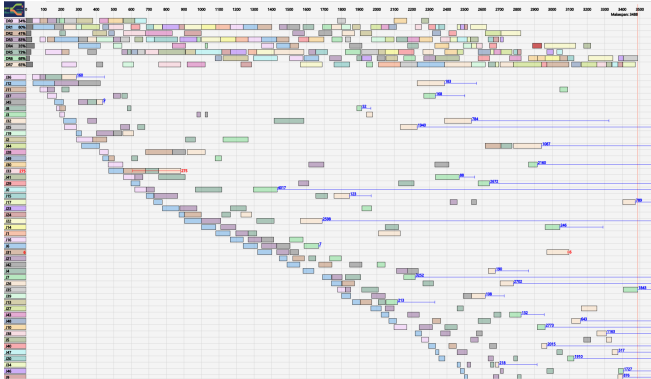
- Able to remove all delays on jobs
- Does not care about makespan or earliness

# Maximizing On-Time Delivery ✓



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

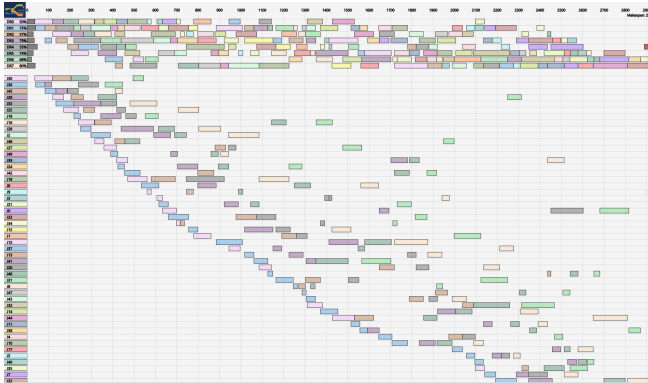
# Hybrid Objective ✓



- 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         | 1,699       | 12     | 7,618.00         | 76,688         | 4,867        | 38      | 76,688.00         | 76.00        | 24.00       |
| Run2      | Flowtime      | 66,356         | Solution     | 39,248.00    | 40.85      | 2,842    | 66,356   | 5,575         | 1,681       | 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* 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

# Multi-level Objectives X



- 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

# Outline



Optimal vs. Good Solutions

Objective Types

Other Quality Vectors

Key Performance Indicators

Interactive Scheduling

# Other Quality Vectors



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



- 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



- 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





- 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"

# Outline



Optimal vs. Good Solutions

Objective Types

Other Quality Vectors

Key Performance Indicators

Interactive Scheduling

# 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



- 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 answer, we can postulate a ranking method



# Outline



Optimal vs. Good Solutions

Objective Types

Other Quality Vectors

Key Performance Indicators

Interactive Scheduling



- Some human schedulers are happy to accept a produced plan
  - Perhaps change some constraints, or weights
- Other human scheduler 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

# 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