Production Scheduling in an Industry 4.0 Era

Joost Berkhout

Wouter Berkelmans, Eric Pauwels, Rob van der Mei & Sandjai Bhulai
Public private partnership between:

ENGIE automates plants

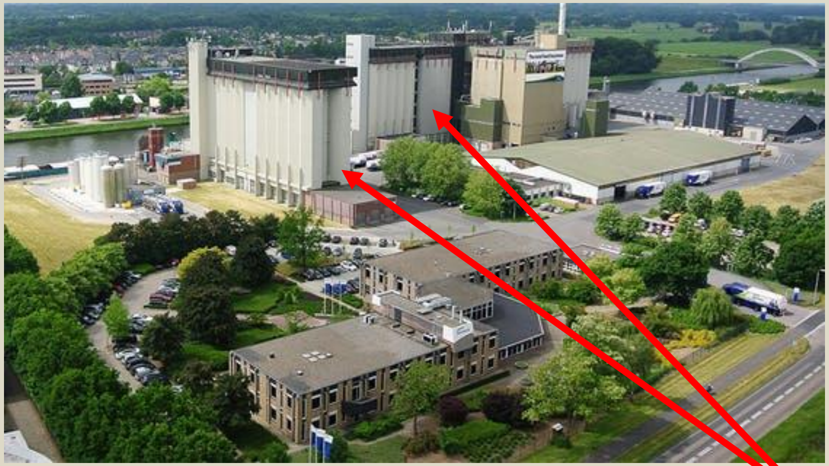
Briefly about my background

- **2008 – 2012:** MSc Econometrics & Operations Research at VU
- **2012 – 2016:** PhD at VU
- **2016 – 2019:** Post-doc at CWI with scheduling project @ ENGIE
- **Since 2019:** Fulltime AP at the A&O group in VU
- **Main research interests:** Production scheduling, simulation optimization, and theory & application of Markov chains (series expansions, Google's PageRank, social networks)

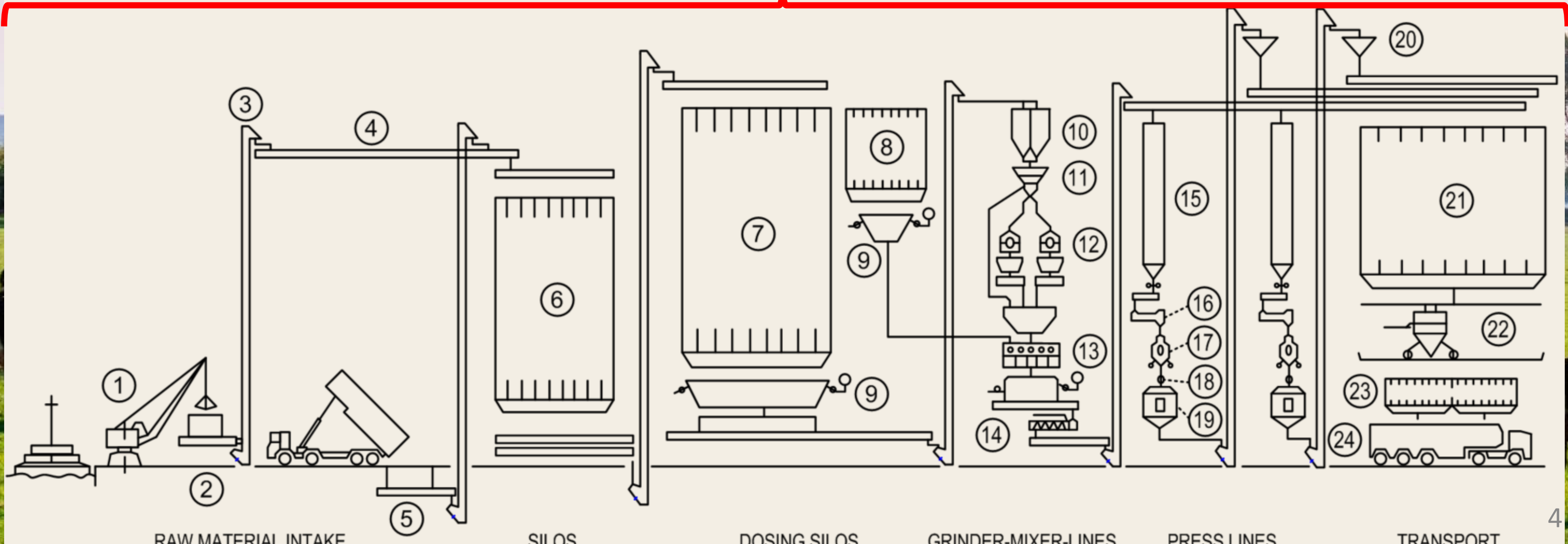
Content Presentation

- Scheduling in animal-feed plants
- Optimization approach
- Numerical experiments
- Concluding remarks

Scheduling in Animal-Feed Plants



- World-wide: $\mathbf{10^{12}}$ kg
- **120** plants in Holland
- Production aspects:
  - Customer order due dates
  - Contamination
  - Changeover times
  - …



Production Scheduling Problem

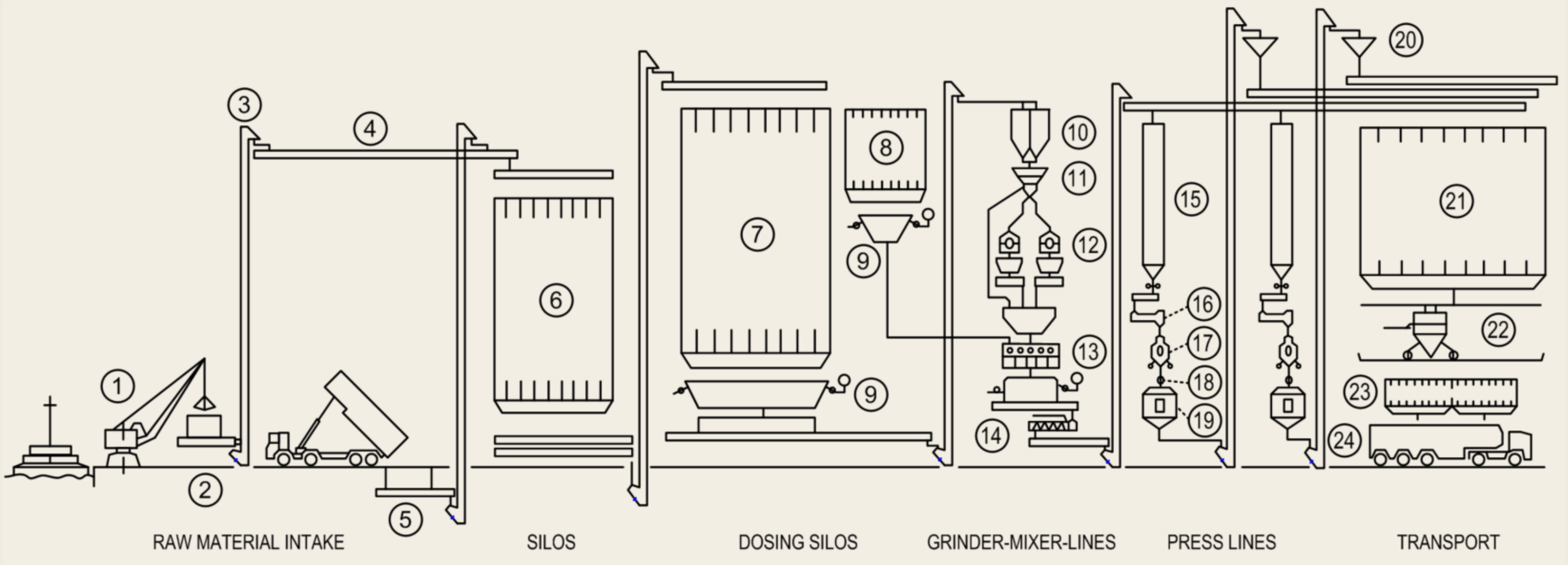
Trends: 'big data' & mass-customization (industry 4.0)

Goal: How to efficiently schedule orders to meet due dates?

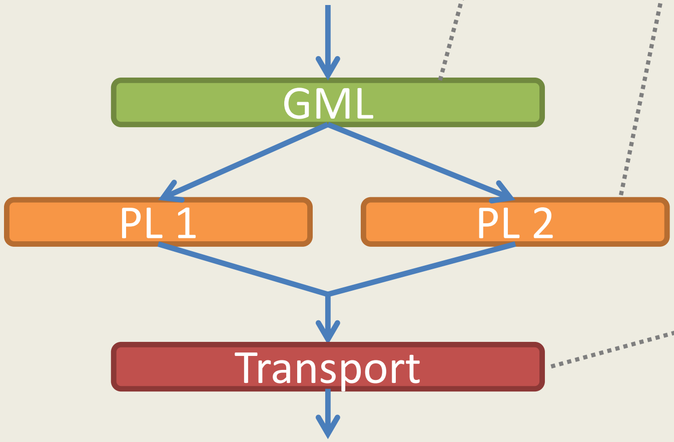
Current situation: planners 'schedule by hand' ...

As a result: time-consuming and opportunity loss (inflexible and 'big data' unused)

Production Process:

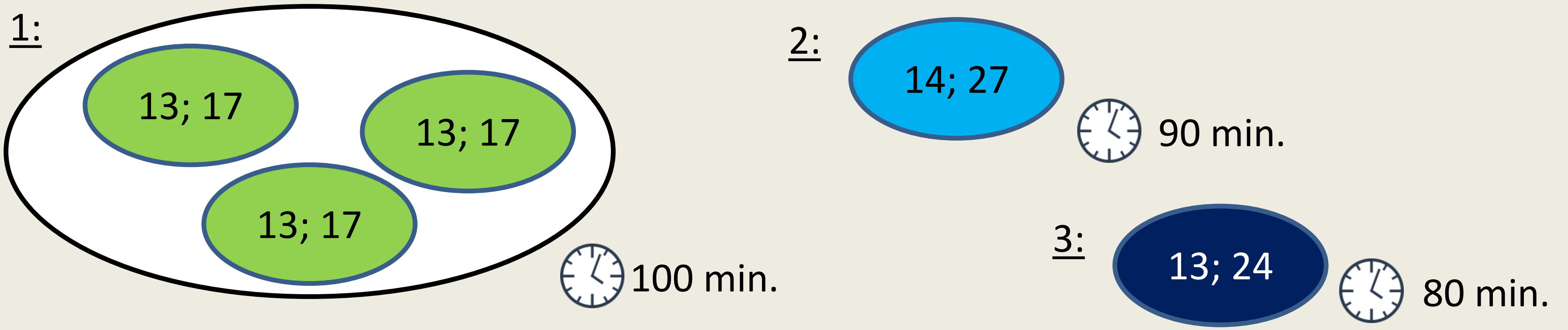


For illustration, simplify production process to:

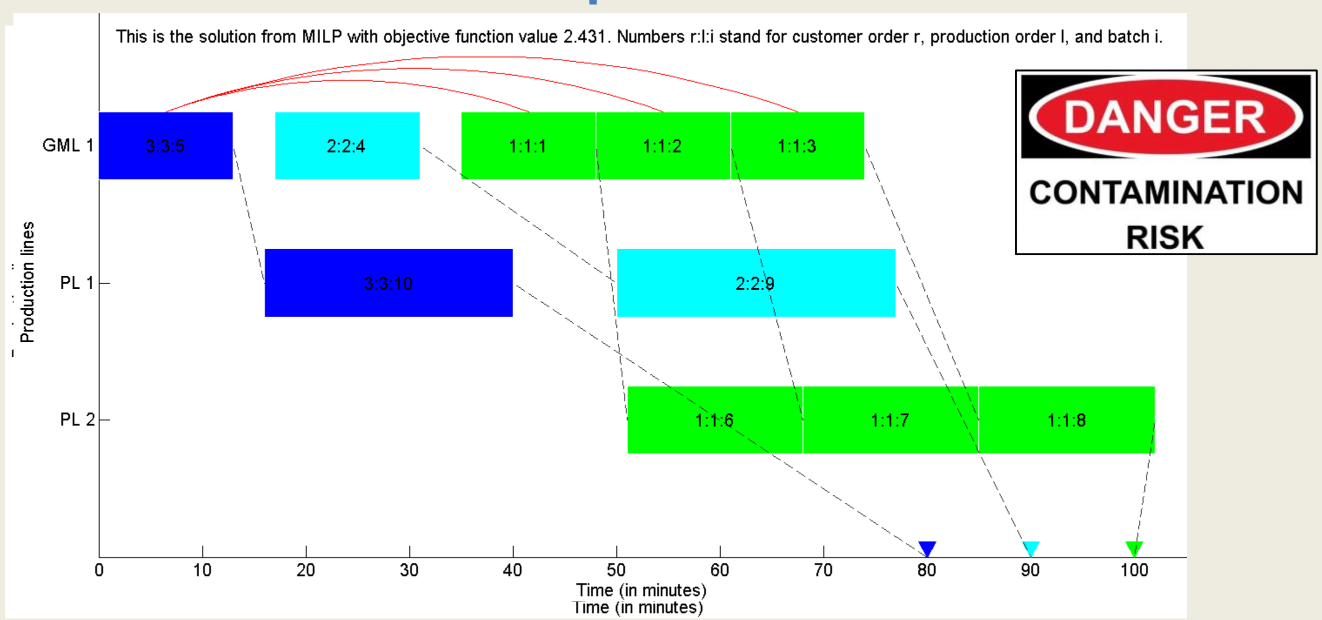


Small Example:

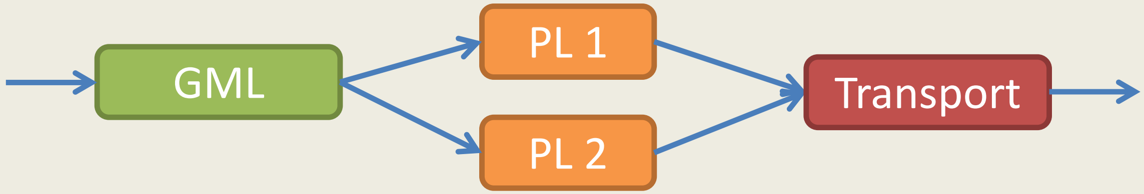
- 3 production orders, consisting of 5 batches/jobs:



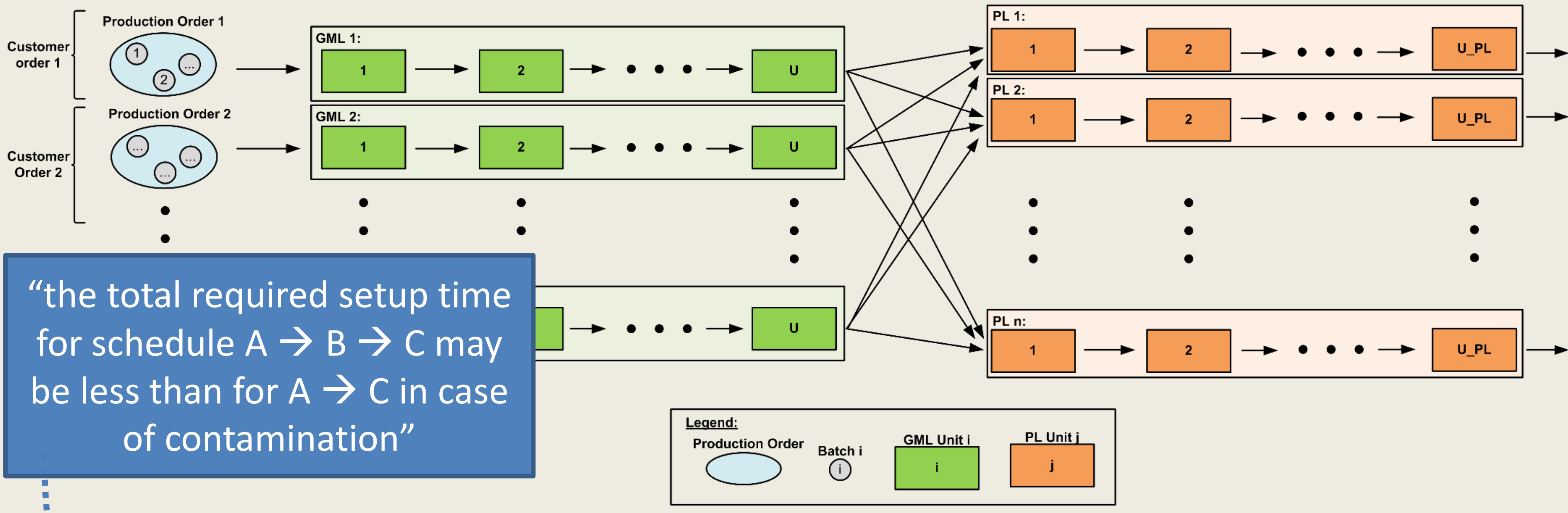
Optimal solution:



Small example:



In general, production lines consist of units:



"the total required setup time for schedule A → B → C may be less than for A → C in case of contamination"

Extended 2-stage flexible flow shop (bottleneck shifting) with non-triangular sequence-dependent setup times

Content Presentation

- Scheduling in animal-feed plants
- ➡ Optimization approach
- Numerical experiments
- Concluding remarks

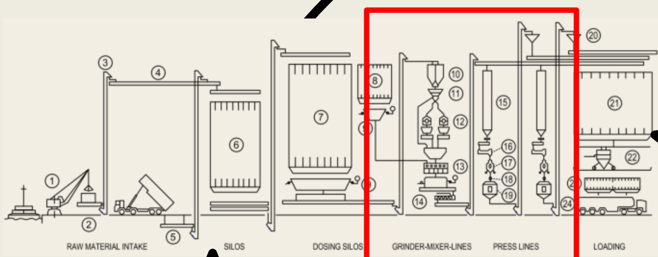
Optimization Approach:

Mixed integer linear programming (MILP):

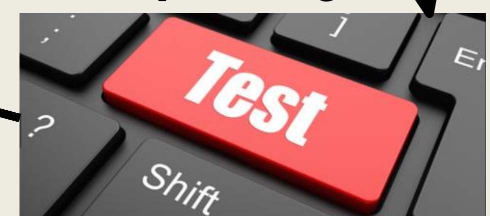
$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{y}} \quad & z \triangleq \mathbf{c}^{\mathsf{T}}\mathbf{x} + \mathbf{d}^{\mathsf{T}}\mathbf{y} \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{y} \left\{ \begin{matrix} \leq \\ = \\ \geq \end{matrix} \right\} \mathbf{b} \\
& \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max}, \quad \mathbf{y} \in \{0,1\}^{n_y}
\end{aligned}
$$

MILP implementation:

Simplification:



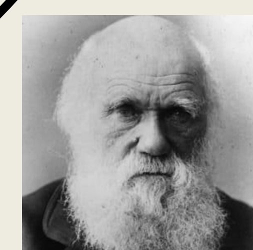
Accuracy testing:



Schedule advice:

Solve MILP:

GUROBI OPTIMIZATION +



Model-based evolutionary algorithm (MBEA)

Optimization approach

This is a model-based evolutionary algorithm

- Two steps optimization approach:
  1. Apply **Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA)** from Bosman et al. (2016)* to bottleneck production area
  2. Fix schedule on bottleneck and solve remaining problem with MILP**
- Motivation for this approach:
  - NP-hard problem (comprise between performance & running time)
  - Flexibility in black box optimization approach & MILP
  - GOMEA is state-of-the-art and MBEAs seen as most powerful EAs
- Our contributions include:
  - Extension of GOMEA for permutation problems* to parallel flow shops
  - Realizing societal impact by developing an efficient optimization approach

*Bosman, P. A. N., Luong, N. H., & Thierens, D. (2016). Expanding from discrete cartesian to permutation Gene-Pool Optimal Mixing Evolutionary Algorithms. *GECCO 2016 - Proceedings of the 2016 Genetic and Evolutionary Computation Conference*.

** Berkhout, J., Pauwels, E., van der Mei, R., Stolze, J., & Broersen, S. (2020). Short-term production scheduling with non-triangular sequence-dependent setup times and shifting production bottlenecks. *International Journal of Production Research*.

GOMEA encoding of schedules

- A schedule for J jobs and M machines is represented by real numbers ("keys") $x_1, \ldots, x_J$ all in $[1, M+1)$:
  - If $m \leq x_i < m+1$: Job i scheduled on Machine m
  - If $m \leq x_i < x_j < m+1$: Job i before j on Machine m

Example: for $J = 5$ jobs

$$x = [2.3, 1.7, 2.6, 1.4, 2.5]$$

encodes schedule

Machine 1: Jobs 4 $\longrightarrow$ 2

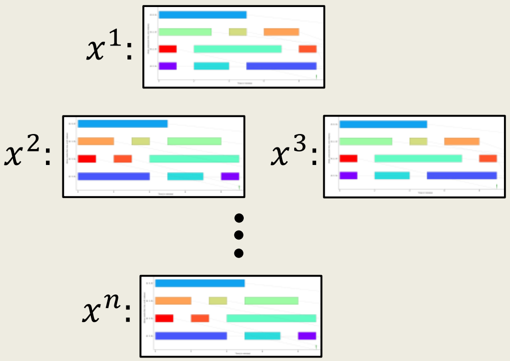
Machine 2: Jobs 1 $\longrightarrow$ 5 $\longrightarrow$ 3

- Cost $C(x)$ = "tardiness + makespan of schedule x"

Gene-pool Optimal Mixing EA (GOMEA)

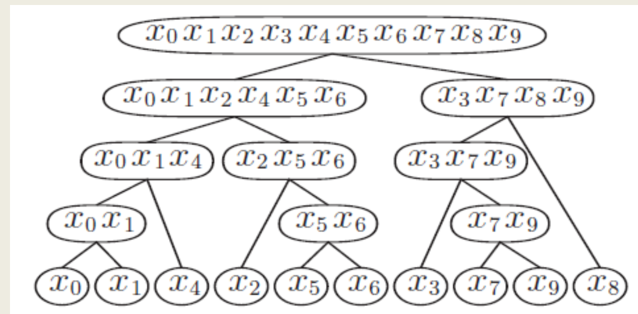
Strength GOMEA: Good subsolutions are detected and exploited during variation

Initial population of feasible schedules



(a.k.a. gene-pool)

Build linkage tree to identify subsets of jobs that contribute to the quality of solutions



Continue till a stopping rule holds

Variation of population

For each schedule in population:

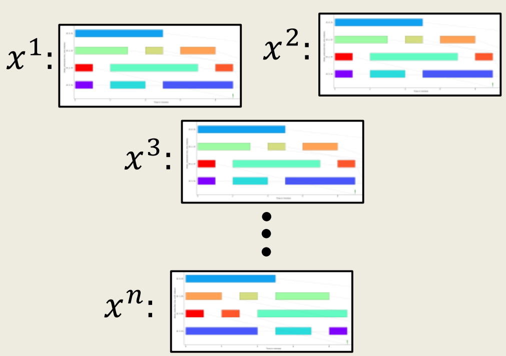
For each subset in linkage tree:

"schedule inherits encoding for subsets from random donor schedule *if it gets better*"

Mutate schedule

"optimal mixing"

New population



Example optimal mixing in GOMEA ($J = 5$ jobs)

Mixing parent schedule

$$x = [2.3, 1.7, 2.6, 1.4, 2.5]$$

Schedule:
Machine 1: Jobs 4 ⟶ 2
Machine 2: Jobs 1 ⟶ 5 ⟶ 3

with donor schedule

$$x' = [2.8, 1.2, 2.4, 1.5, 2.1]$$

Schedule:
Machine 1: Jobs 2 ⟶ 4
Machine 2: Jobs 5 ⟶ 3 ⟶ 1

for subset $\{x_1, x_3, x_5\}$, we get

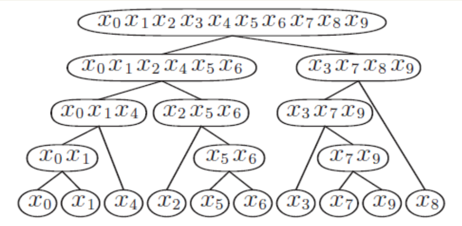
$$x^{new} = [2.8, 1.7, 2.4, 1.4, 2.1]$$

Schedule is thus:
Machine 1: Jobs 4 ⟶ 2
Machine 2: Jobs 5 ⟶ 3 ⟶ 2

which is accepted if it is better

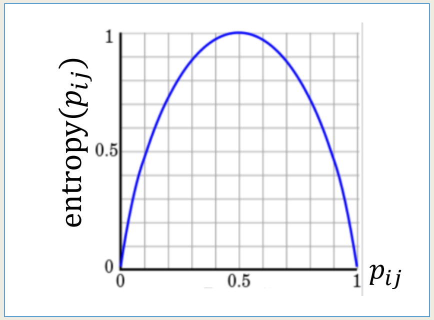
Note: Starting with a feasible population, mixing keys always leads to a feasible schedule

Building a linkage tree



- Population-based dependency quantification between Jobs i and j is:

$$\delta(i,j) = \delta(j,i) = \delta_1(i,j)\delta_2(i,j)$$



Relative-ordering information
Measures how often i is scheduled before j in the population:

$$\delta_1(i,j) = 1 - \text{entropy}(p_{ij})$$

with

$$p_{ij} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{1}\{x_i^k < x_j^k\}$$

Adjacency information
Measures the proximity of i and j in the population:

$$\delta_2(i,j) = 1 - \frac{1}{n}\sum_{k=1}^{n} (x_i^k - x_j^k)^2$$

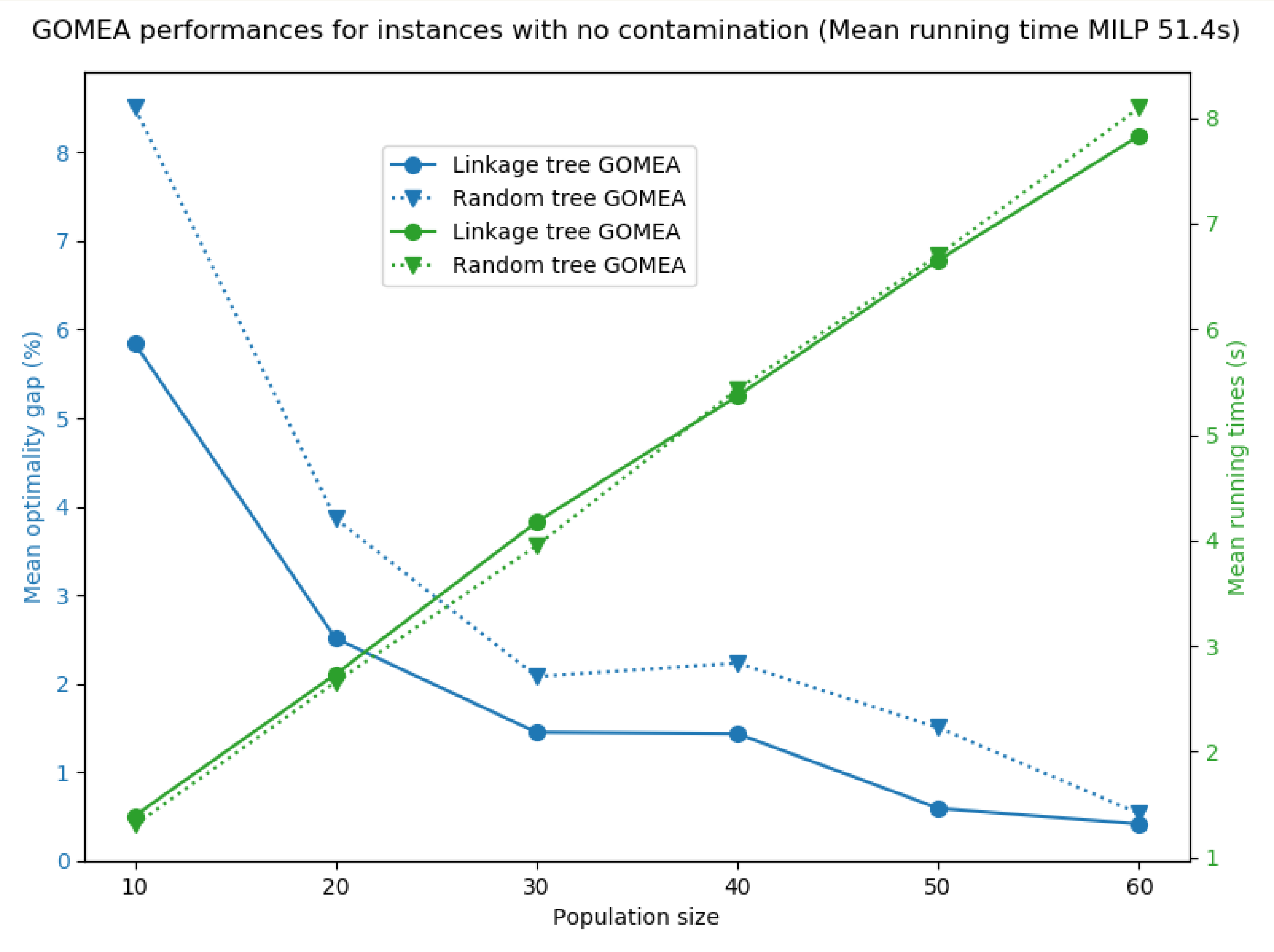
- Linkage tree built by iteratively combining the most dependent jobs (on average) in $O(J^2 n)$
- **Further research:** Exploring more advanced distance functions

Content Presentation

- Scheduling in animal-feed plants
- Optimization approach
- ➡ Numerical experiments
  - Performance GOMEA for parallel flowshops
  - Performance optimization approach in practice
- Concluding remarks

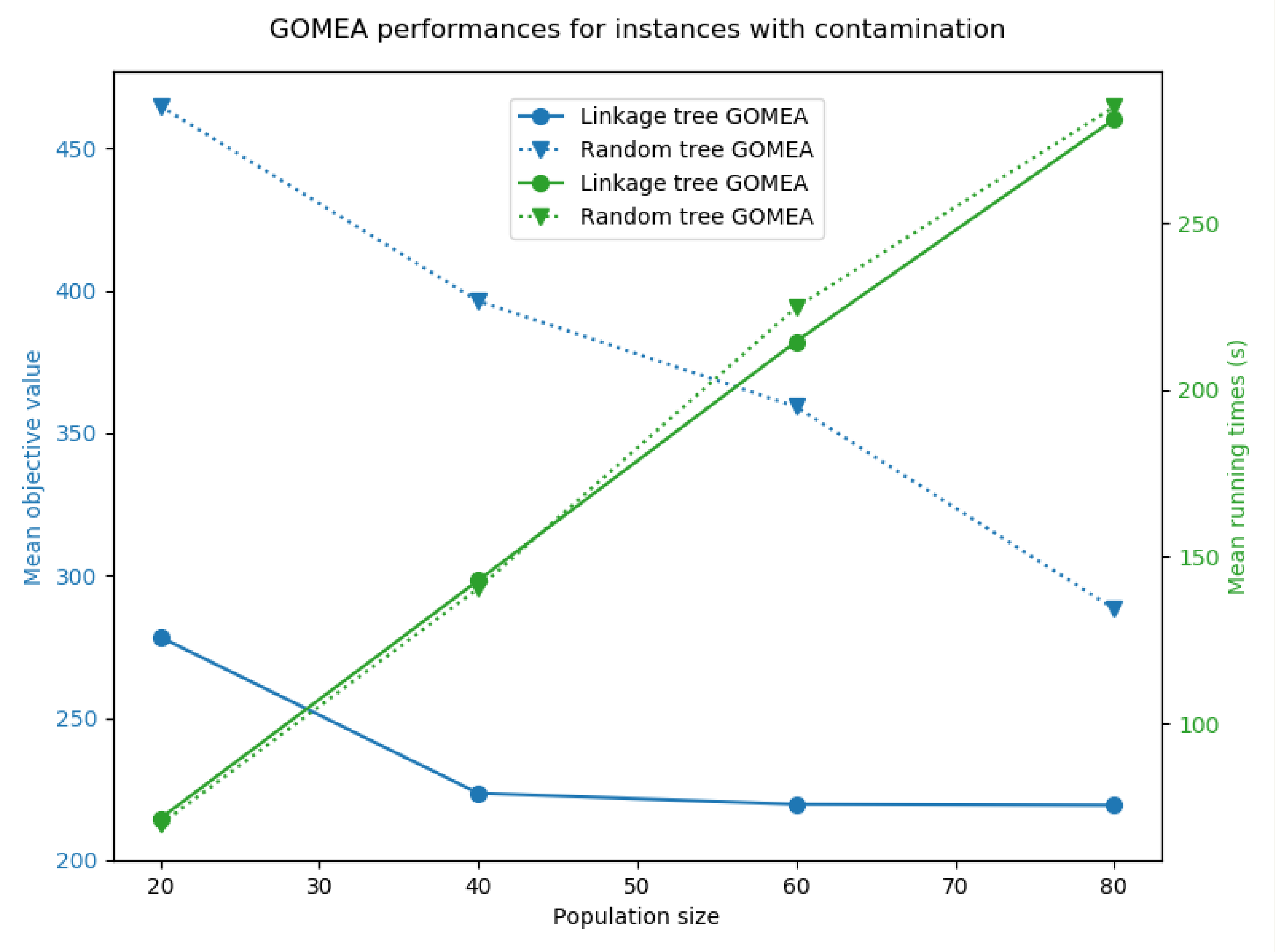
Are GOMEA solutions close to optimality?

Experiment: 4 machines of 3 units each with 11 jobs, average-results over 20 random instances ($> 1.4 \cdot 10^{10}$ schedules)



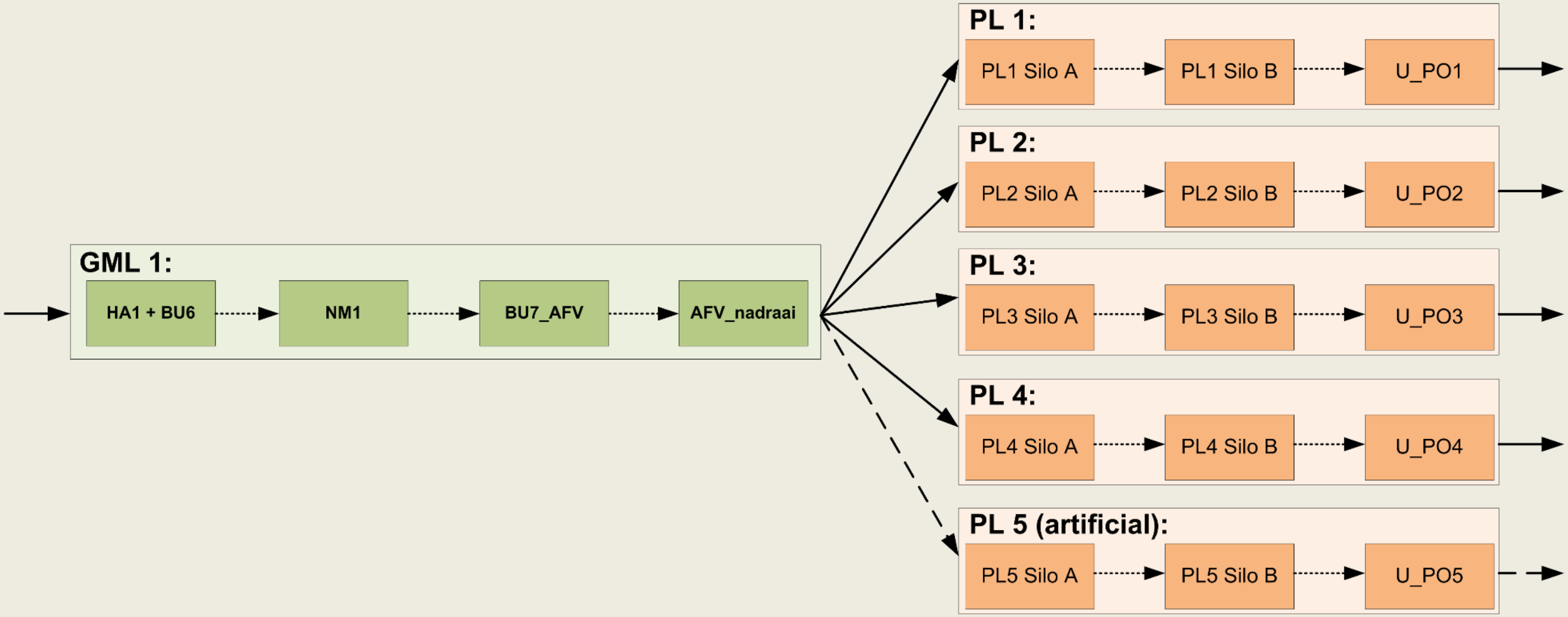
Impact of *learning a model* in EA

Experiment: 4 machines of 3 units each with **50** jobs, per experiment 20 random instances with contamination (> $7.1 \cdot 10^{68}$ schedules)



Results for a Pilot Plant:

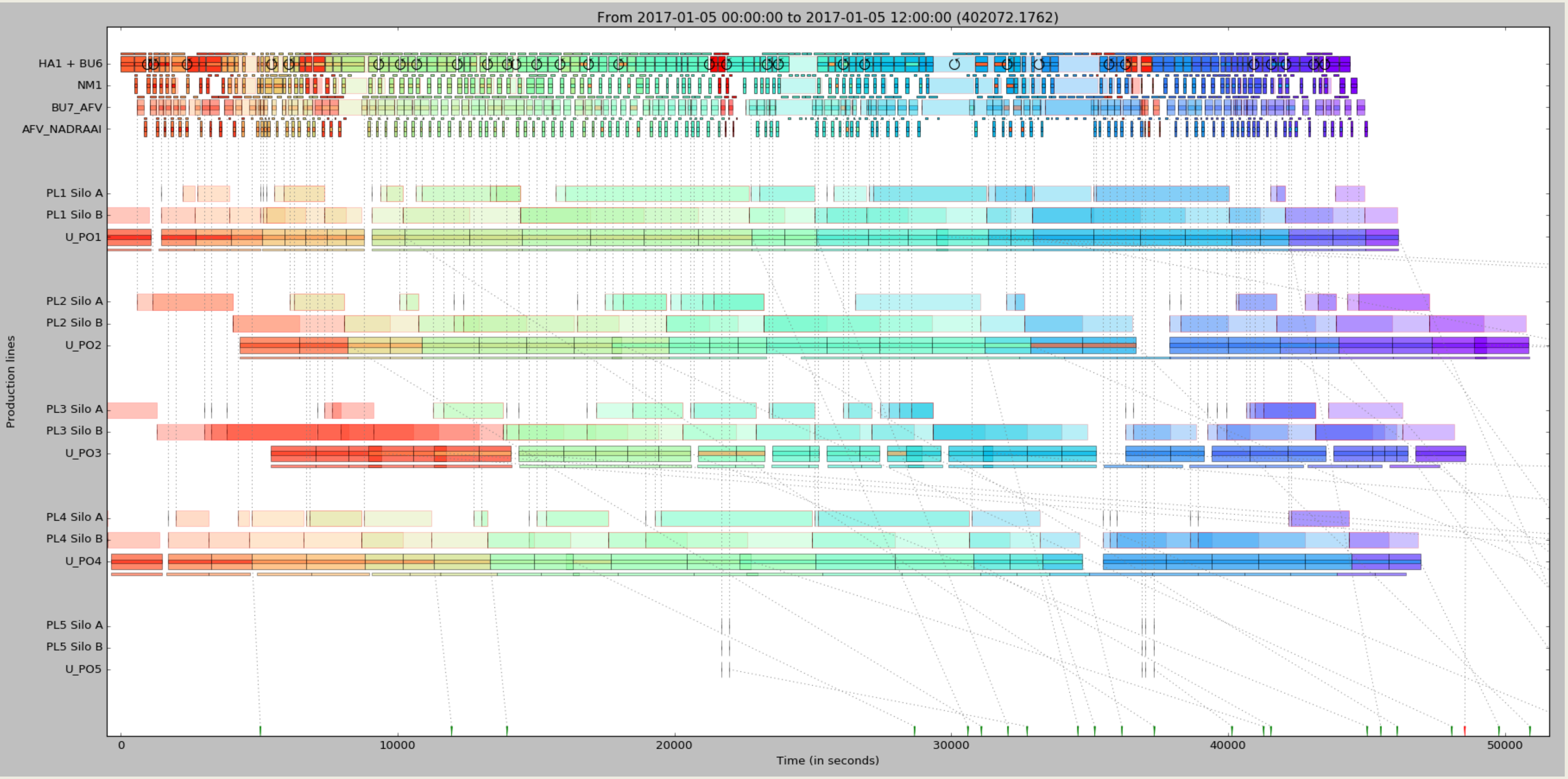
1 grinder-mixer line and 5 press lines



(Recall:) **Optimization approach is:**

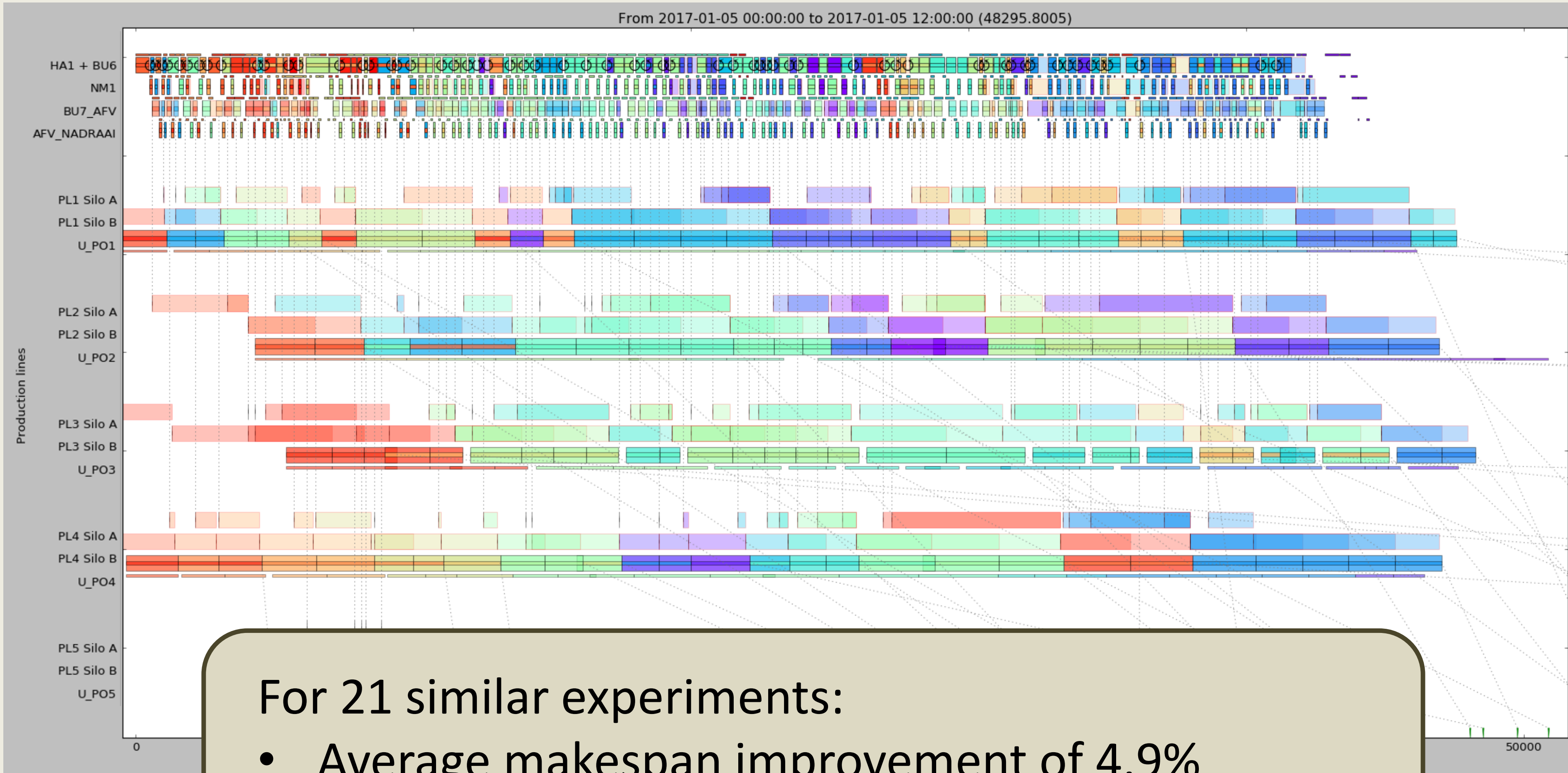
1) GOMEA on bottleneck
2) Solve MILP of complete problem

Realized schedule for 12 hour time window (120 jobs)



Optimized schedule (in 111s)

Makespan is reduced by 40 minutes (5.5%) and all deadlines are met:



For 21 similar experiments:

- Average makespan improvement of 4.9%
- Except for one, all deadlines are met

Content Presentation

- Scheduling in animal-feed plants
- Optimization approach
- Numerical experiments
- Concluding remarks

Concluding Remarks

- GOMEA efficiently detects and exploits important subsolutions in parallel flowshops
- MILP model is implemented in a pilot plant in Limburg (testing for accuracy and optimization gain)
- Further research:
  - Optimization of transport and finished product silos
  - Further development of (tailored) heuristics
  - Exploring the application of GOMEA to a routing and scheduling problem in home health care (together with René Bekker and Yoram Clapper)

Thanks for your attention!

Any questions?

Mail: j2.berkhout@vu.nl