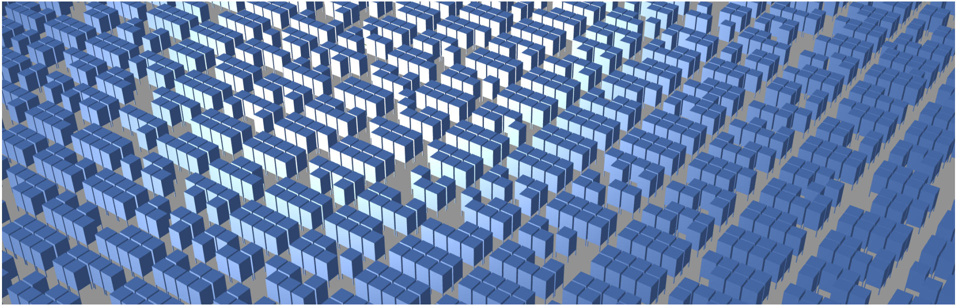
LEUPHANA
UNIVERSITY OF LÜNEBURG

Shelf Repositioning in a Robotized Warehouse

for OR 2018

Ruslan Krenzler, Lin Xie, Hanyi Li | September 2018

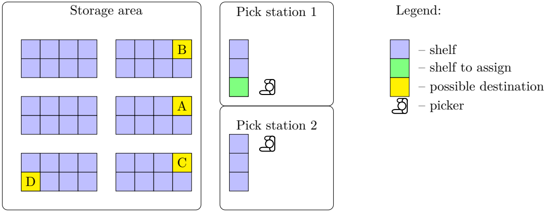
LEUPHANA UNIVERSITÄT LÜNEBURG



Warehouse video

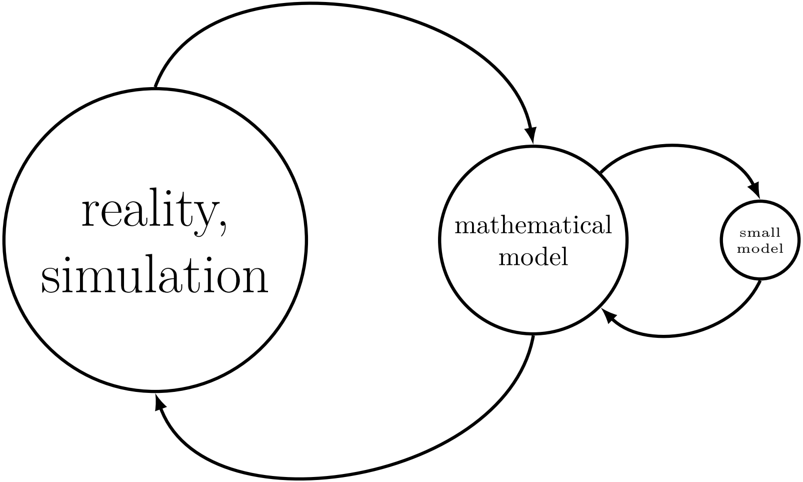
Edited Video. The original video material *is:*
DB Schenker implementing next generation e-commerce
from `https://www.youtube.com/watch?v=udr0OOxmPbc`

Where to put the shelf to?

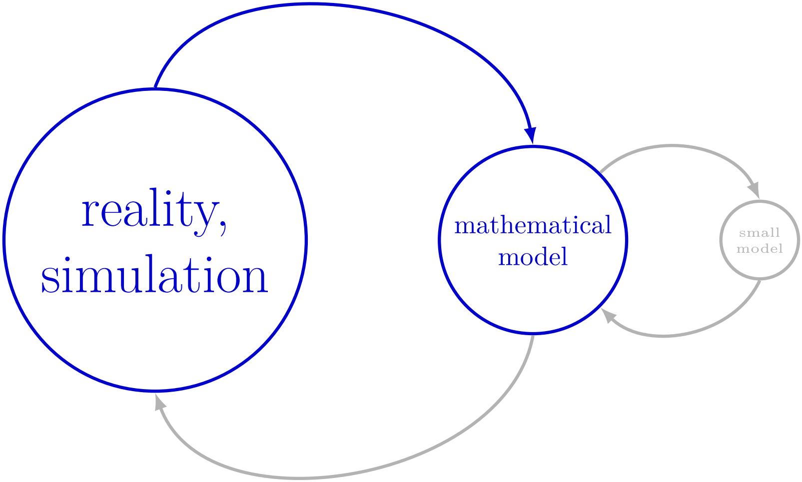


Sequence of place assignments

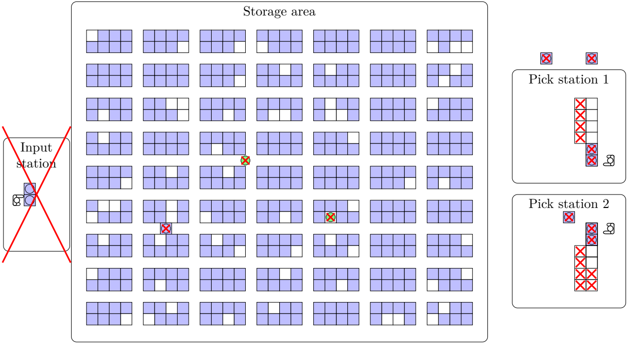
Methodology



Methodology



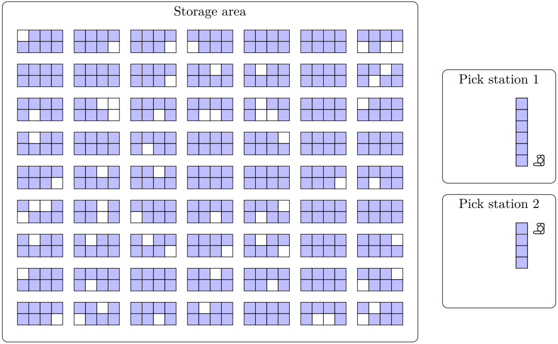
Simplification



- No input station.
- No explicit robots.
- Fixed max. queue size.
- Discrete space and time steps.
- No explicit orders. Known shelf departures. Keep queues full.

Minimize the total costs

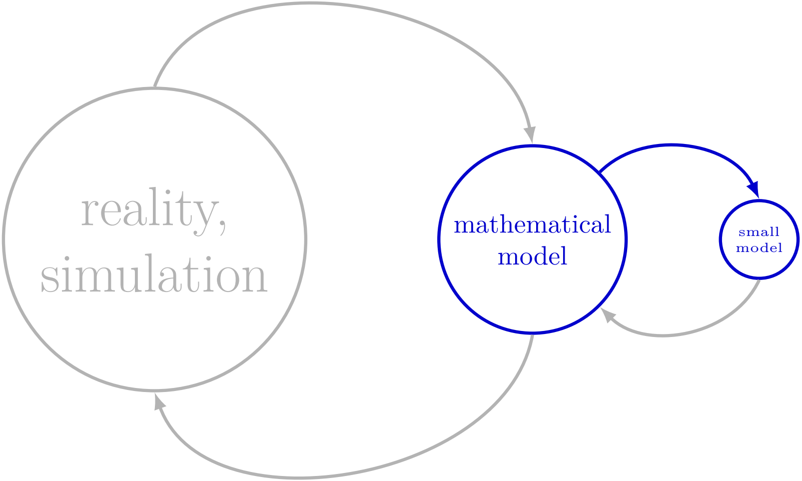
Simplification



- No input station.
- No explicit robots.
- Fixed max. queue size.
- Discrete space and time steps.
- No explicit orders. Known shelf departures. Keep queues full.

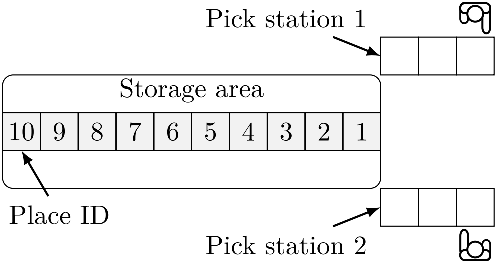
Minimize the total costs

Simple test system



Simple test system

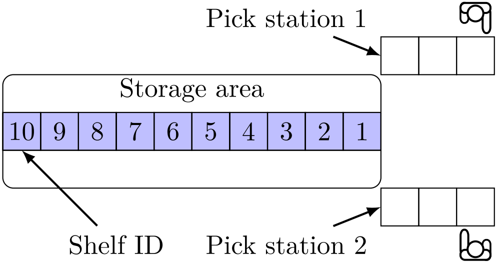
As simple as possible as complex as necessary!



- 1-dimensional storage area.
- *Two* symmetrical pick stations.
- Few places and few shelves.
- In the beginning, the shelves are presorted.
- Costs are distances.
- 1000 time steps = 1000 decisions.

Simple test system

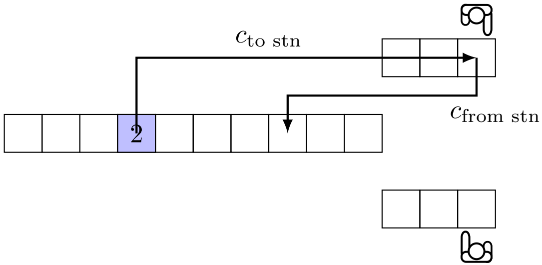
As simple as possible as complex as necessary!



- 1-dimensional storage area.
- *Two* symmetrical pick stations.
- Few places and few shelves.
- In the beginning, the shelves are presorted.
- Costs are distances.
- 1000 time steps = 1000 decisions.

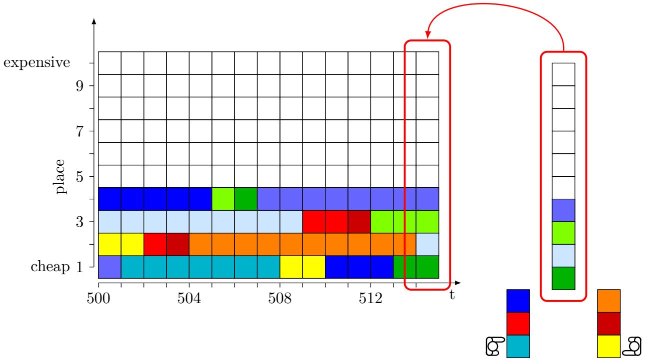
Simple test system

As simple as possible as complex as necessary!



- 1-dimensional storage area.
- *Two* symmetrical pick stations.
- Few places and few shelves.
- In the beginning, the shelves are presorted.
- Costs are distances.
- 1000 time steps = 1000 decisions.

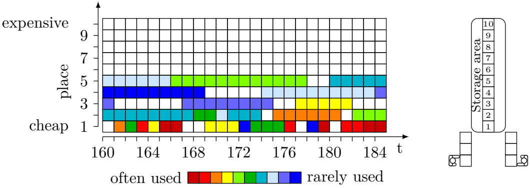
Storage chart



Algorithms we tried

- Cheapest place
- Random place
- Fixed place
- Genetic programming

Binary Integer Programming

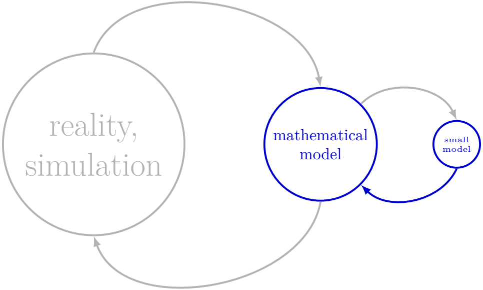


Decision variables $x_{tp} \in \{0,1\}$ t-time, p-place. Translation t-th shelf goes to place p.

1. Constraint: Assign all shelves.
2. Constraint: Start from initial order in the storage.
3. Constraint: Do not assign two shelves to one place at the same time.

- ⊕ Optimal solution.
- ⊕ Very simple formulation.
- ⊖ Not flexible.
- ⊖ Does not work for large problems.

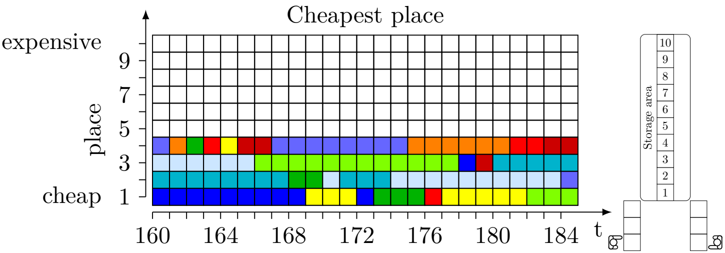
Ready for the larger model?

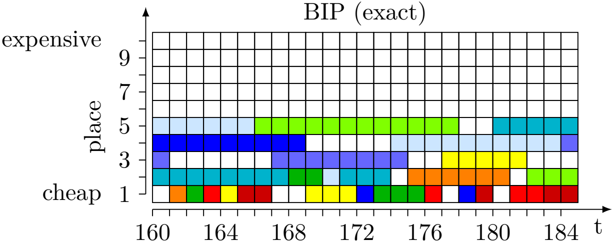


No

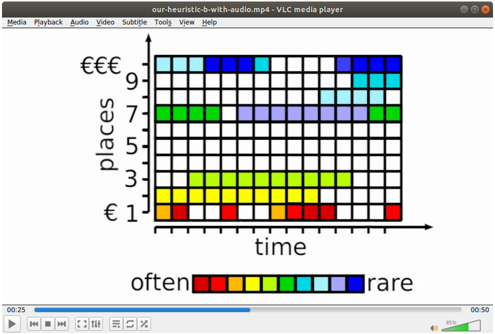
- Binary Integer Programming does not work.
- Genetic too slow.

Good heuristic for large problems





Tetris Video

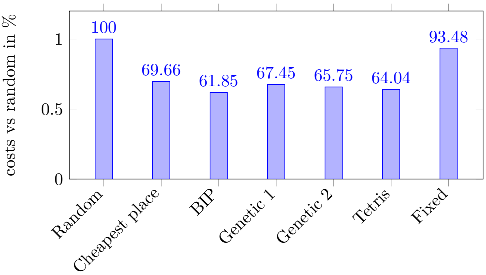


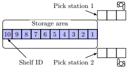
Tetris algorithm

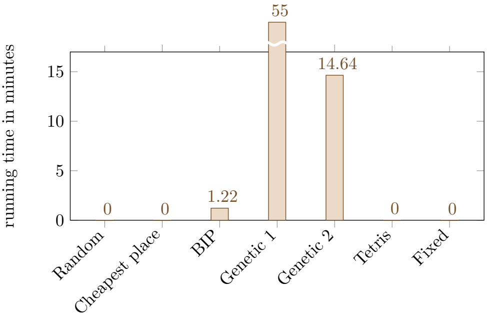
Write idea as pseudocode.

1. Create initial feasible solution with a lot of free cheap places.
2. Calculate occupation time for each shelf.
3. Sort shelves by frequency.
4. Select the most frequent shelf.
5. Assign all occurrences of the shelf to the cheapest free places.
6. Select the next most frequent shelf and go to 5, stop if there are no shelves left.

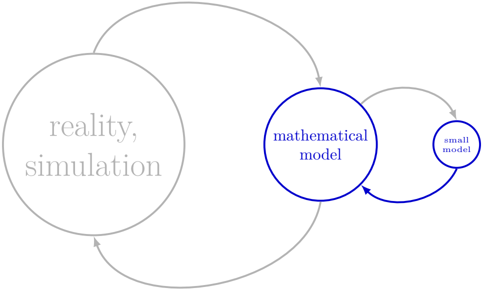
Numerical results





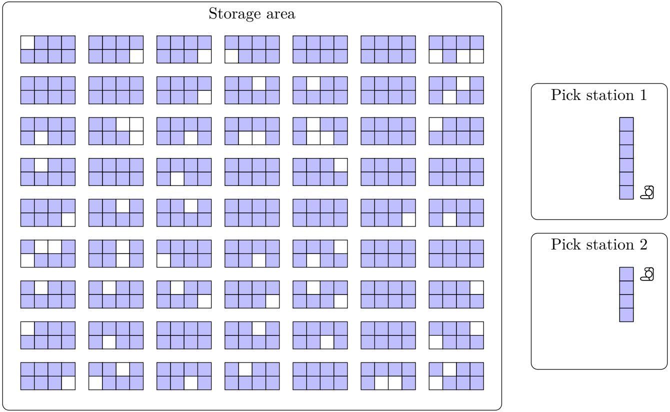


From small to more complex model



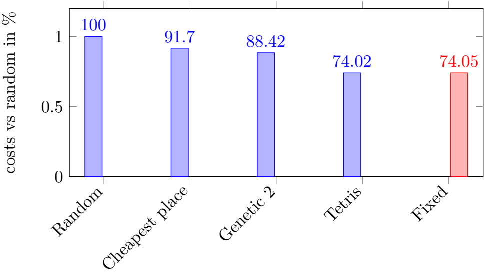
From small to more complex model

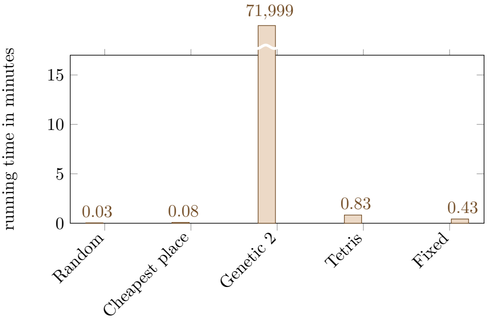
Test on



- 511 places, 411 shelves, 20 000 time steps.
- Costs are realistic distances.
- Asymmetric pick station.

Numerical results





Results

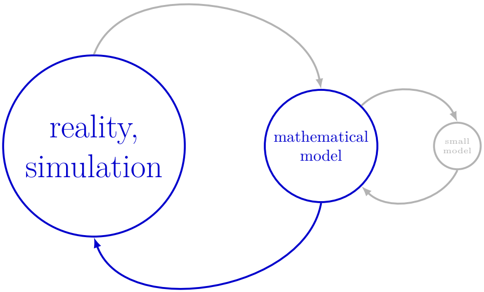
- ☑ Real-world problem.
- ☑ Mathematical model.
- ☑ Binary Integer Programming (BIP).
- ☑ Realized BIP does not work for large problems.
- ☑ Made heuristic.
- ☑ Heuristic is fast and good.

Overall results

Paper!!!

Or did we miss something?

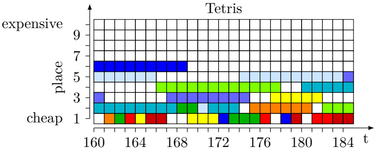
Results

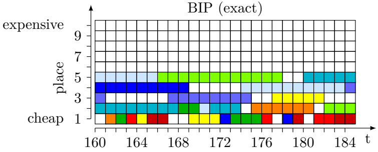


Or did we miss something?

Yes! Reality!

Reality





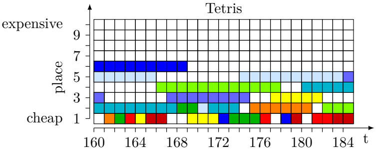
Problem:

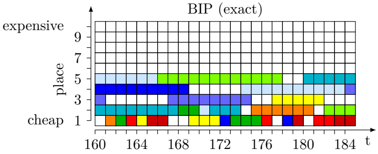
All good algorithms need precise time information,

- time when a shelf enter the storage
- time when the shelf leave the storage

But time depends on humans. Humans work randomly.
Rescaling reduces time intervals to arrival and departure order. It works only in systems with one pick station. They do not exist.

Reality





Problem:

Random times at picking stations.

Solution:

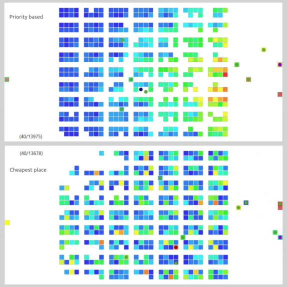
Use previous results as *concepts*:

- *Tetris* is a priority algorithm – most frequent shelves get the cheapest places. It considers future use of places.

Use Markov model and queuing theory.

Simulation video

- Realistic robot movements.
- An input station.
- But not real orders.



Open questions:

- How it will work with real orders?
- What will happen, when coupled with other optimizations (order & shelves, layout, product distribution)?

Conclusions

- Small models can be very useful.
- Why does (not) an algorithm work?
- Reality is important.

Thank you for your attention!