

Multi-Hoist Cyclic Hoist Scheduling

Gergely Márton Szathmári

Supervisor: Markó Horváth

June 4, 2026

Single-hoist CHSP

- Studied during the previous semester
- MIP formulation implemented
- Solver and visualization created

Current work: Multi-hoist CHSP

- Multiple hoists operate on the same line
- Significantly more difficult scheduling problem
- Hoist collisions must be avoided
- Objective: maximize throughput

- Hoists move carriers from tank to tank
- Empty and loaded travel times are given
- Carriers are soaked in the tanks
- Minimum and maximum soaking times are given
- Hoists repeat the same actions every cycle

Goal

- Find a feasible collision-free schedule
- Minimize cycle length
- Equivalent to maximizing throughput

Most papers solve the problem using a Mixed Integer Program.

The formulations determine:

- Start times of loaded moves
- Hoist assignment of moves
- Collision-free execution

However:

Missing information

The empty movements of the hoists are not explicitly specified, therefore the complete hoist trajectories cannot be reconstructed.

Collision Avoidance

Key idea of existing formulations:

- Restrict feasible positions of hoists
- Prevent overtaking and collisions
- Guarantee existence of collision-free trajectories

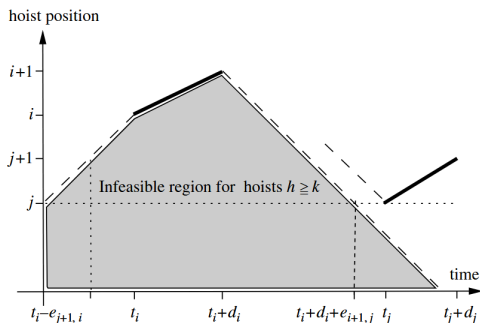
Example constraint:

$$t_i + d_i + e_{i+1,j} - t_j \leq M \left(3 - y_{ij} - z_i^k - \sum_{h=k}^K z_j^h \right)$$

Term	Meaning
t_i	Start times of move i
y_{ij}	Binary indicator: 1 if move j starts after move i
z_i^k	Assignment: 1 if move i is done by hoist k
d_i	Duration of loaded move i
$e_{i+1,j}$	Empty travel time from destination of move i to start of move j
M	A large constant

Collision Avoidance

$$t_i + d_i + e_{i+1,j} - t_j \leq M \left(3 - y_{ij} - z_i^k - \sum_{h=k}^K z_j^h \right)$$



The shaded forbidden region on the time-space diagram cannot be occupied by larger indexed hoists.

Goal

Construct the exact trajectory of every hoist.

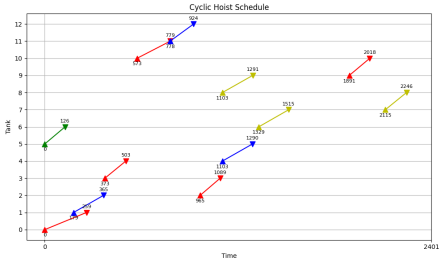
For two consecutive moves of a hoist:

- 1 Wait after completing the previous move
- 2 Travel directly to the next tank
- 3 Wait until the next move starts

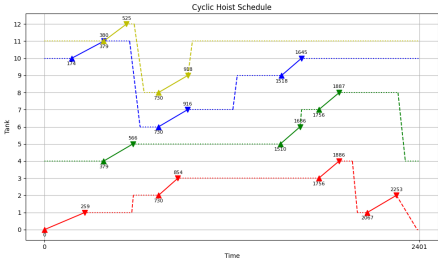
Additional assumptions:

- No evasion maneuvers
- No movements through the cycle boundary

Example



Original optimal schedule
(only loaded moves determined)



Complete solution
(generated by our model)

New Variables

Variable	Meaning
a_i	arrival time to move i
b_i	departure time after move i
x_{ij}^k	Binary indicator: 1 if move j directly follows move i on hoist k

The variables x_{ij}^k define the exact sequence of moves executed by each hoist.

Trajectory Constraints

The exact formulation can be found in my report.

- First the x_{ij}^k variables are defined properly
- Travel constraints:
$$a_j = b_i + e_{i+1,j}$$
whenever move j immediately follows move i

These constraints force direct travel between consecutive tasks.

- The beginning and the end of the cycle requires many more technical constraints to avoid collisions

Tested on benchmark instances from the literature.





Observations:

- Complete trajectories obtained
- Collision-free schedules constructed
- In most cases the optimal cycle time remained unchanged

Summary

- Reviewed multi-hoist CHSP literature
- Studied existing MIP formulations
- Identified missing trajectory information
- Introduced a trajectory-construction model
- Generated complete executable schedules

References

-  Phillips, L. W., and P. S. Unger. "Mathematical Programming Solution of a Hoist Scheduling Program." *AIIE Transactions* 8 (2): 219–25. (1976).
-  Emna Laajili. Modélisation et algorithmes pour le dimensionnement et l'ordonnancement cyclique d'atelier de traitement de surface. *Automatique / Robotique*. Université Bourgogne Franche-Comté, (2021).
-  Leung, Janny MY, et al. "Optimal cyclic multi-hoist scheduling: A mixed integer programming approach." *Operations Research* 52.6 (2004): 965-976.
-  Che, Ada, et al. "An improved mixed integer programming approach for multi-hoist cyclic scheduling problem." *IEEE Transactions on Automation Science and Engineering* 11.1 (2013): 302-309.

- **Text Optimization:** Refined grammar, syntax, and spell-checking.
- **Code Generation:** Developed and debugged Python scripts.
- **Visual Styling:** Assisted in formatting and structuring the presentation.

Thank you for your attention!