Applying Multi-Agent Path Finding to the Train Unit Shunting Problem

Issa Hanou, Jesse Mulderij, **Mathijs de Weerdt**
Delft University of Technology

M.M.deWeerdt@tudelft.nl

TU Delft

Our “AI for Rails” focus

Main questions in our lab:

1. Which *AI planning* techniques can benefit the operational planning of a hub (station & yards)? [Train Unit Shunting Problem with Service and Scheduling (TUSPwSS)]
2. How can we improve these techniques to make them most effective?

TU Delft

How to combine, service, park?

Image: David van Keulen

Image: Siebe Swart

How to combine, service, park?

How, where and when shall trains split and combine, and being parked, considering arrivals, departures, train types, planned circulation, required service jobs, track lengths, and safety norms?

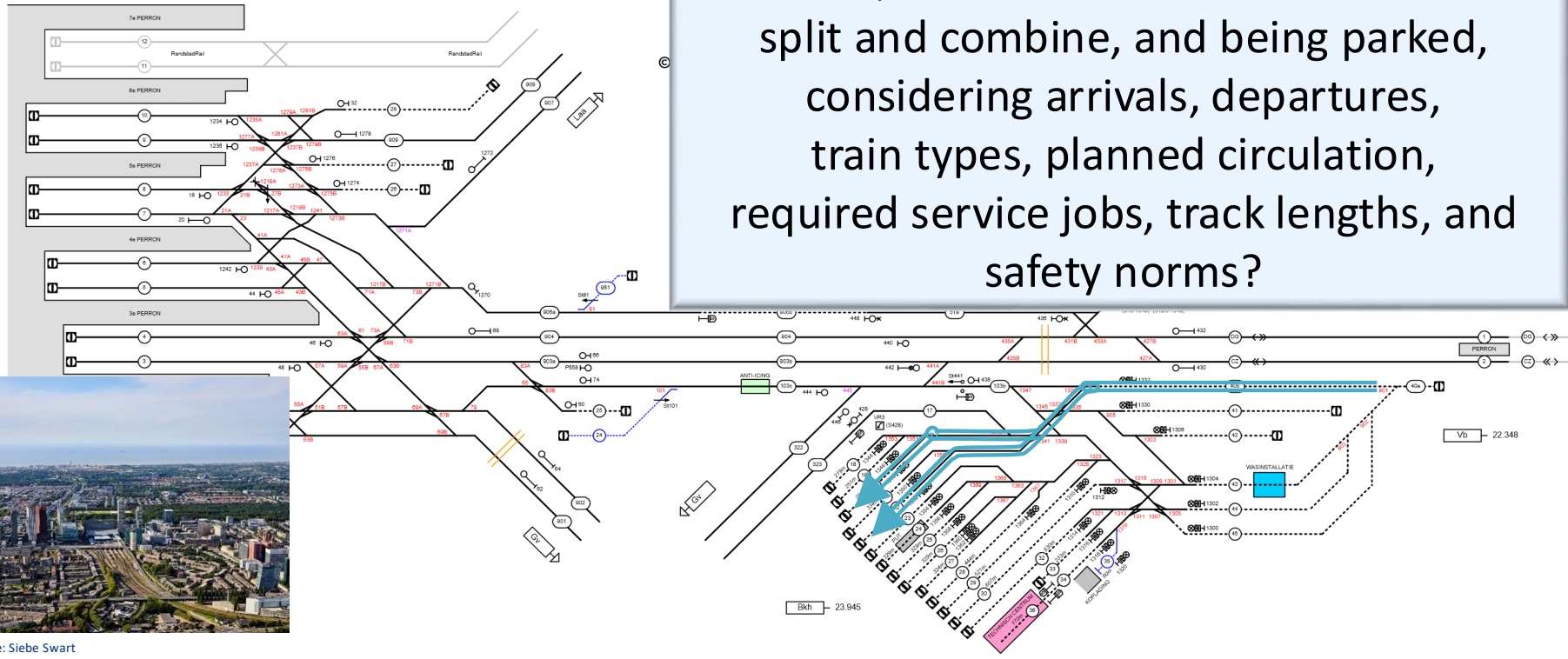


Image: Siebe Swart

Only when everything fits, we have a feasible plan!

Image: Siebe Swart

a difficulty for existing methods (e.g. MILP) seems to be the conflict-free routing

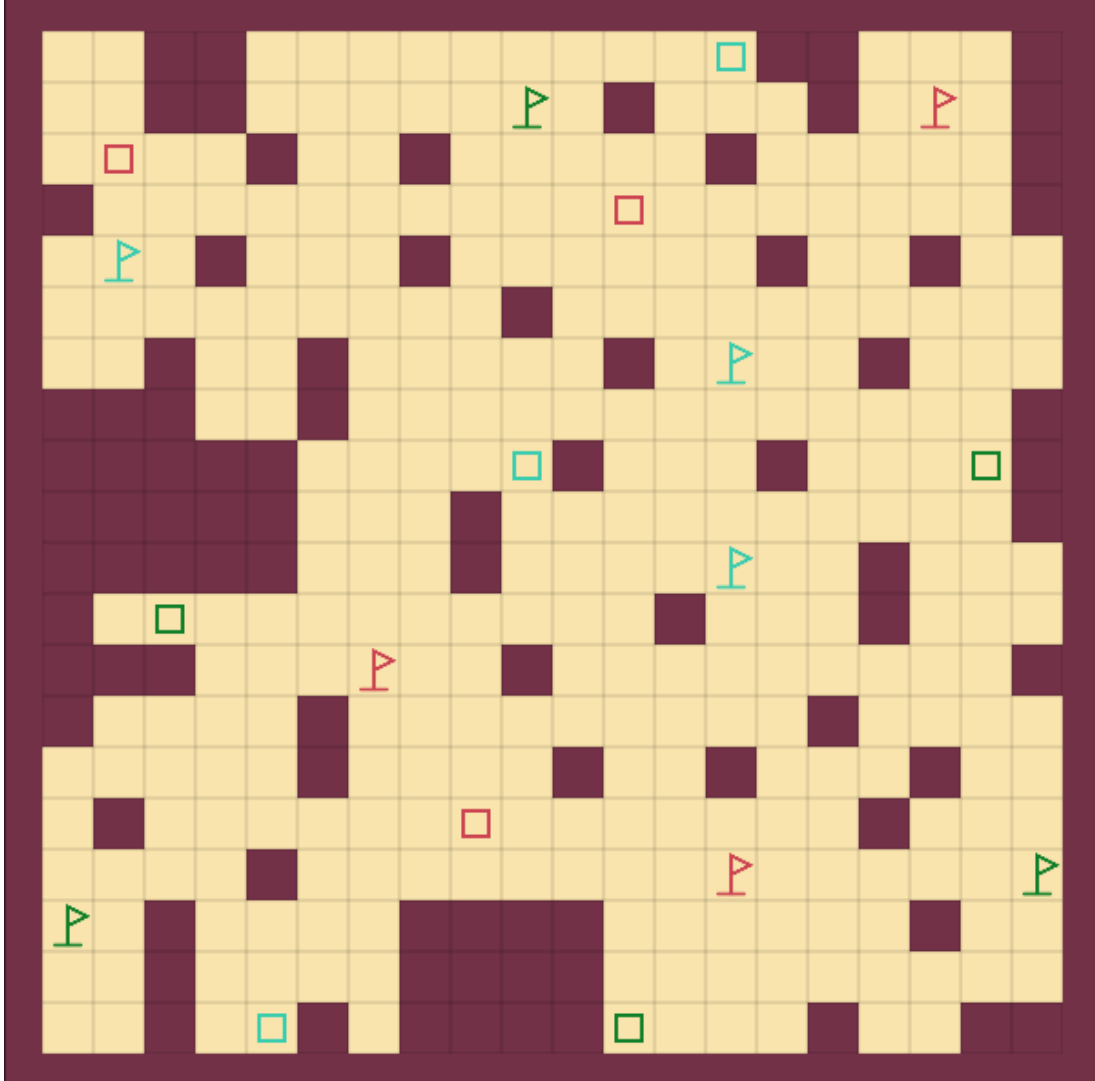
Our “AI for Rails” focus

in this talk:

1. Which *multiagent path finding* techniques can benefit the Train Unit Shunting Problem?

So no service scheduling & also ignoring train lengths for this talk.

Multi-agent path finding



TU Delft

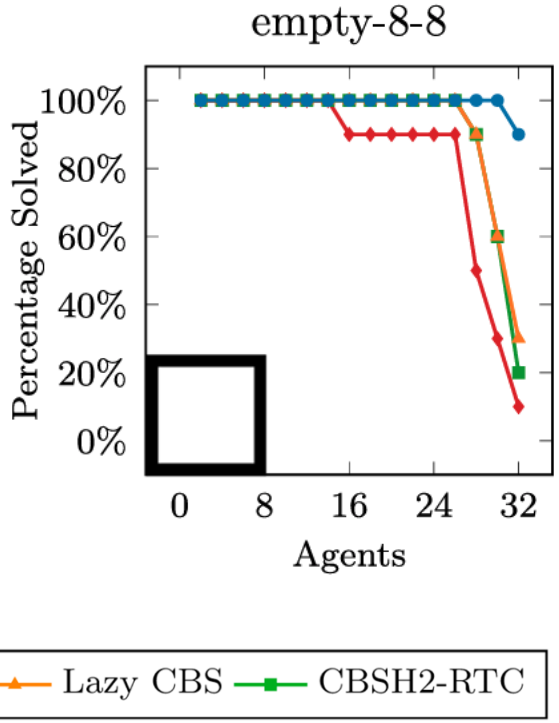
Multi-agent path finding (MAPF)

Motivation:

- MAPF is all about routing
- large and active community (over a 100 papers, competitions) http://mapf.info/
- success for up to 100s of robots, e.g. on maps representing warehouses

Branch-Cut & Price

- master-pricer setup (column generation)
- label setting for paths (in pricer)
- master problem is LP
- 11 problem-specific cuts/separators



Lam, Edward, Pierre Le Bodic, Daniel Harabor, and Peter J. Stuckey. “Branch-and-Cut-and-Price for Multi-Agent Path Finding.” *Computers & Operations Research* 144 (2022).

Table 1
Number of solved instances and average run time after removing each improvement.

Configuration	Instances solved		Average time	
Full algorithm	1188		149	
Wait-edge conflicts	1187	(−1)	149	(+0)
Two-agent wait-edge conflicts	1185	(−3)	149	(+1)
Corridor conflicts	1184	(−4)	150	(+1)
Wait-corridor conflicts	1183	(−5)	150	(+1)
Wait-delay conflicts	1188	(−0)	149	(+1)
Exit-entry conflicts	1126	(−62)	156	(+7)
Two-edge conflicts	1171	(−17)	151	(+2)
Wait-two-edge conflicts	1178	(−10)	151	(+2)
Rectangle conflicts	896	(−292)	184	(+36)
Step-aside conflicts	1187	(−1)	149	(+0)
Goal conflicts	1000	(−188)	176	(+27)
A* caching	1165	(−23)	151	(+2)
Path length branching	889	(−299)	184	(+35)

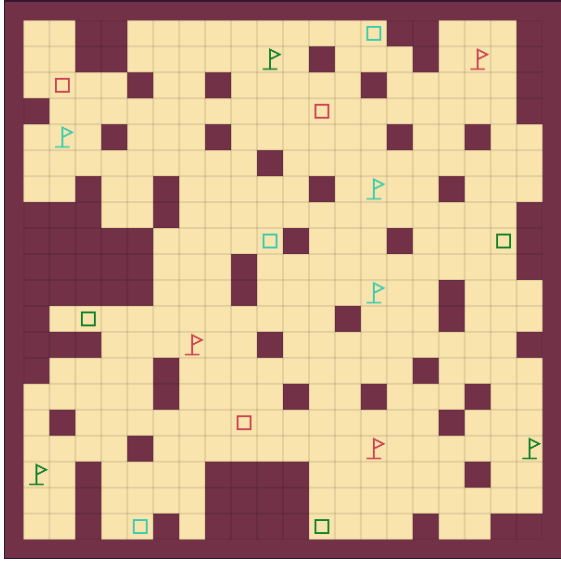
cuts

other techniques

TU Delft

Lam, Edward, Pierre Le Bodic, Daniel Harabor, and Peter J. Stuckey. “Branch-and-Cut-and-Price for Multi-Agent Path Finding.” *Computers & Operations Research* 144 (2022).

Multi-agent path finding



For TUSPwSS, extend MAPF with:

- shunting yard layouts (graphs, no grids)
- train types
- matching arriving trains to departing ones

- [train lengths]
- [including deadlines (e.g. departure times)]

TU Delft

Mulderij, Huisman, Tönissen, van der Linden, & **MdW**. (2020). Train unit shunting and servicing: a real-life application of multi-agent path finding. *arXiv preprint arXiv:2006.10422*.

Table 1
Number of solved instances and average run time after removing each improvement.

Configuration	Instances solved		Average time	
Full algorithm	1188		149	
Wait-edge conflicts	1187	(−1)	149	(+0)
Two-agent wait-edge conflicts	1185	(−3)	149	(+1)
Corridor conflicts	1184	(−4)	150	(+1)
Wait-corridor conflicts	1183	(−5)	150	(+1)
Wait-delay conflicts	1188	(−0)	149	(+1)
Exit-entry conflicts	1126	(−62)	156	(+7)
Two-edge conflicts	1171	(−17)	151	(+2)
Wait-two-edge conflicts	1178	(−10)	151	(+2)
Rectangle conflicts	896	(−292)	184	(+36)
Step-aside conflicts	1187	(−1)	149	(+0)
Goal conflicts	1000	(−188)	176	(+27)
A* caching	1165	(−23)	151	(+2)
Path length branching	889	(−299)	184	(+35)

cuts

other techniques

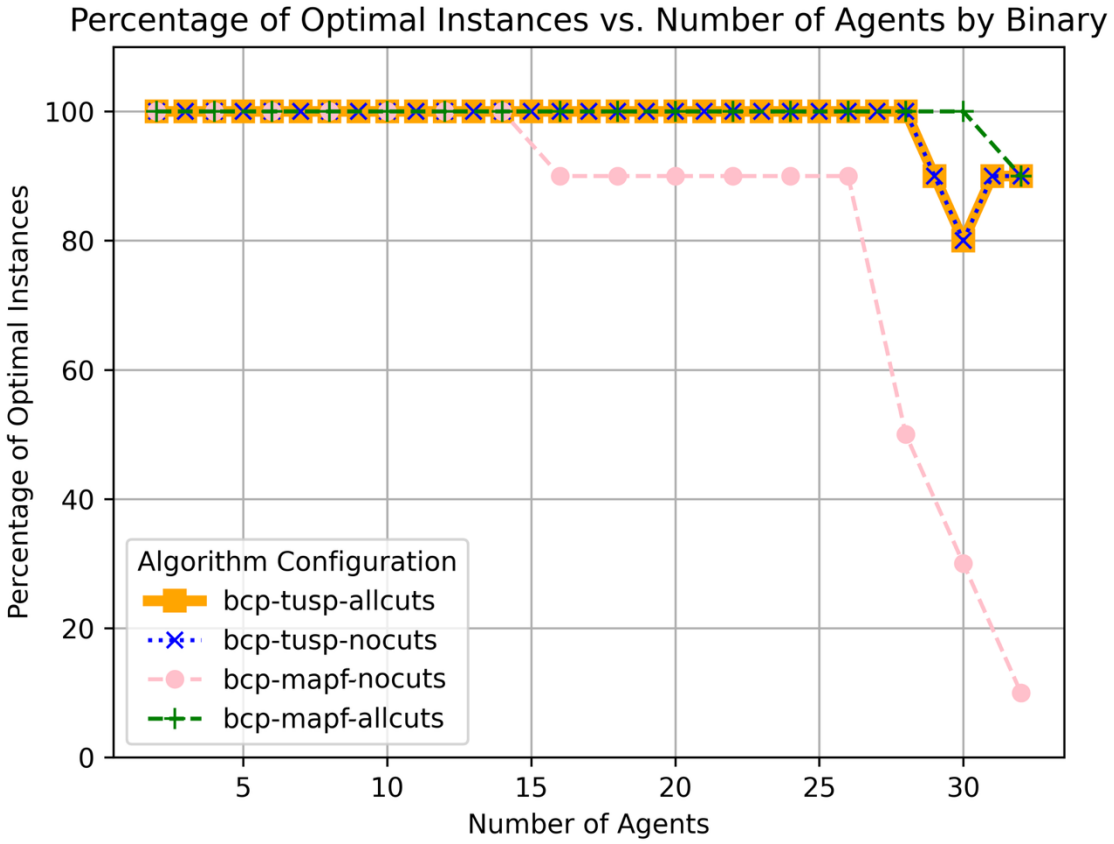
TU Delft

Lam, Edward, Pierre Le Bodic, Daniel Harabor, and Peter J. Stuckey. “Branch-and-Cut-and-Price for Multi-Agent Path Finding.” *Computers & Operations Research* 144 (2022).

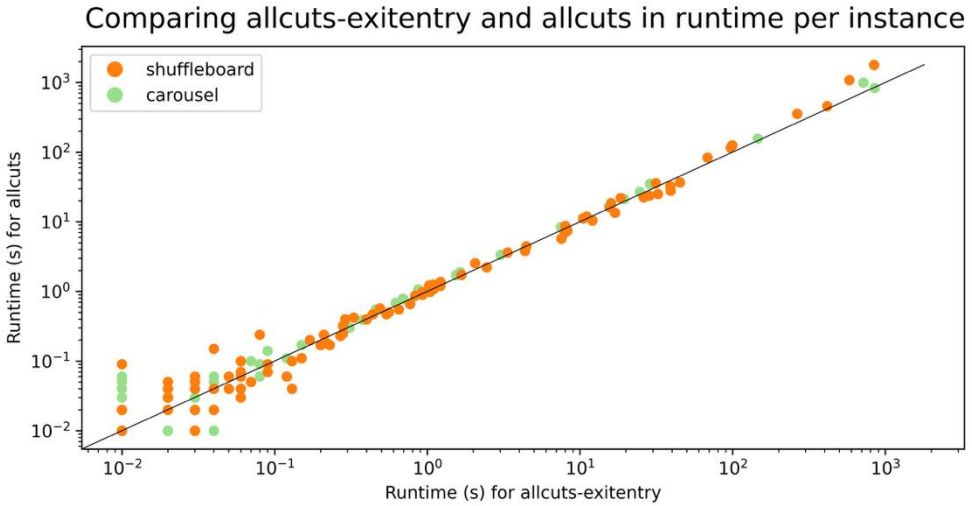
Research question

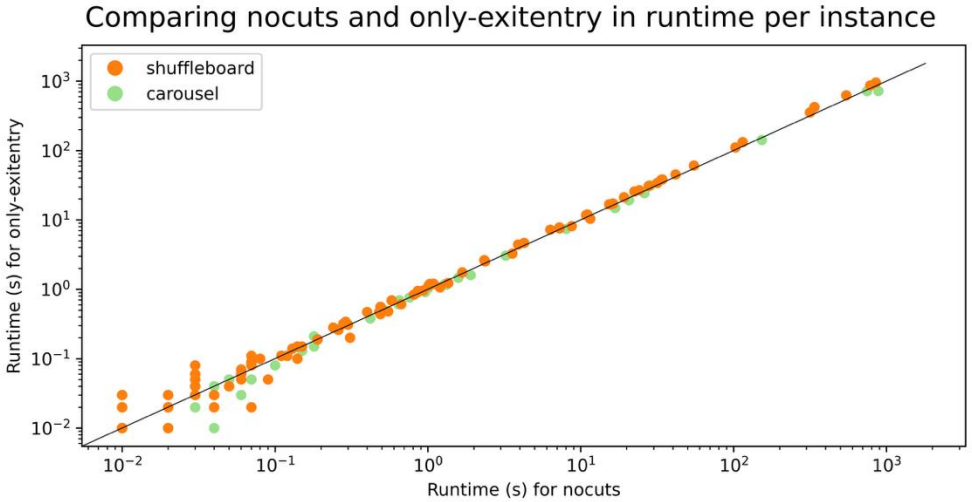
- How much do the 3 best-performing cuts for MAPF improve solving TUSP instances?
  - exit-entry conflicts
  - two-edge conflicts
  - goal conflicts

Graphs instead of grids: 8x8



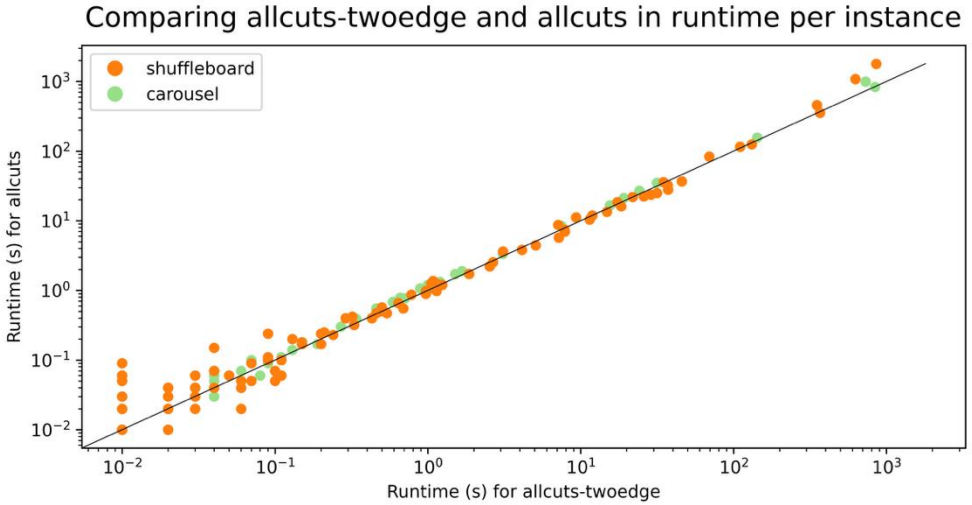
Exit-entry

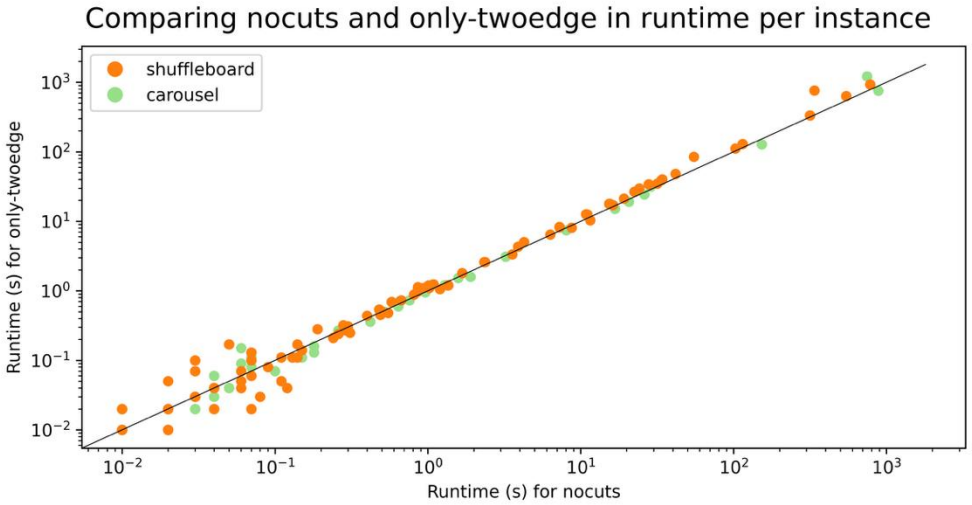




TU Delft

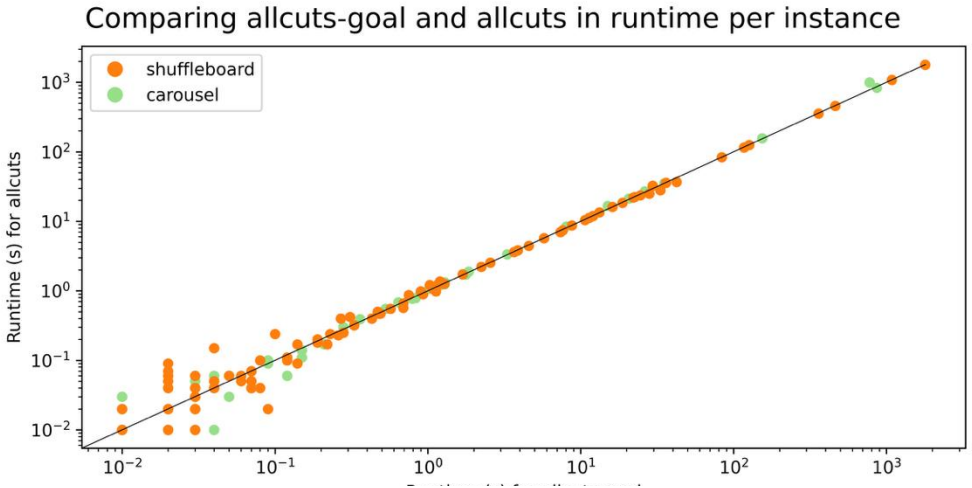
Two-edge

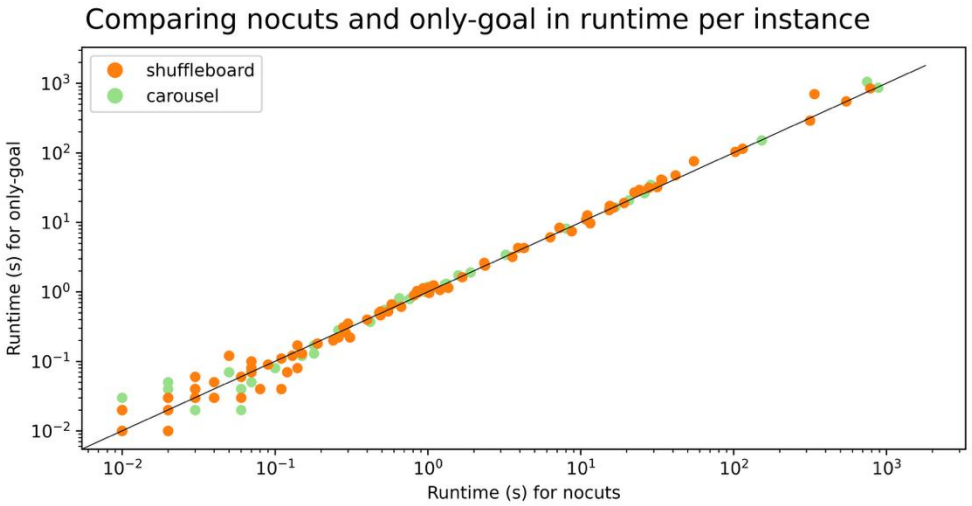




TU Delft

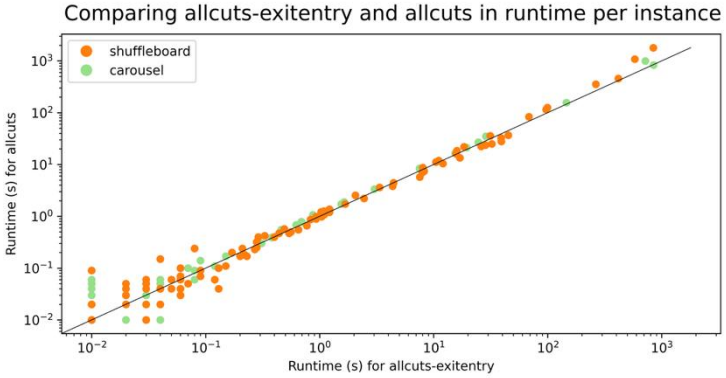
Goal

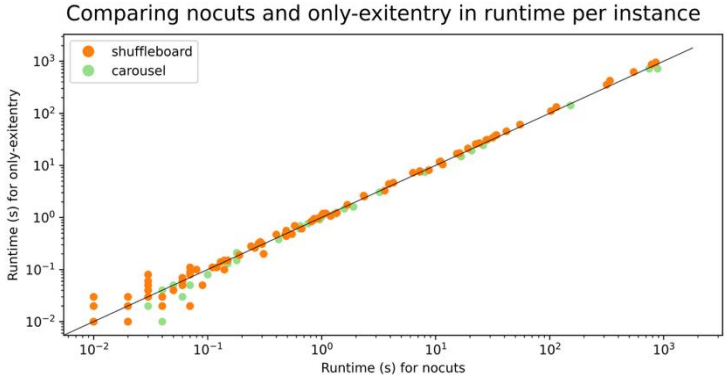


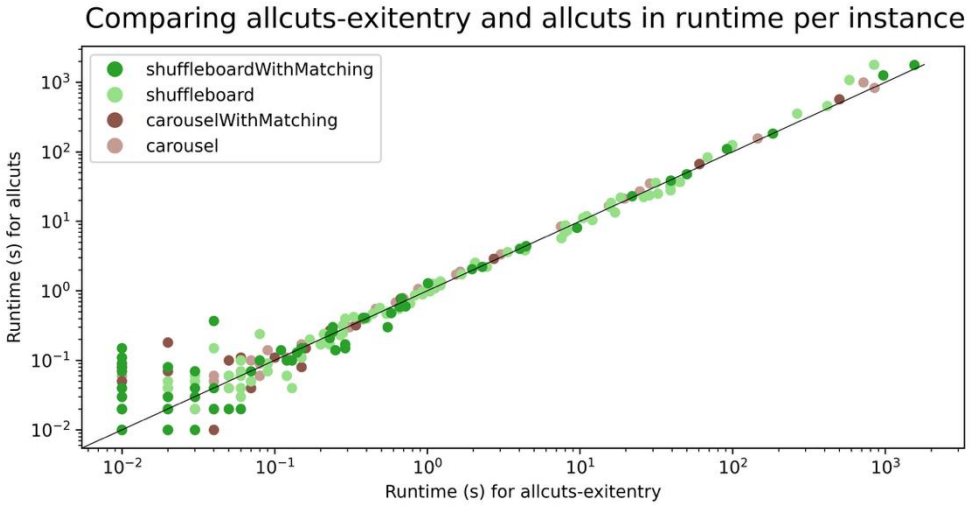


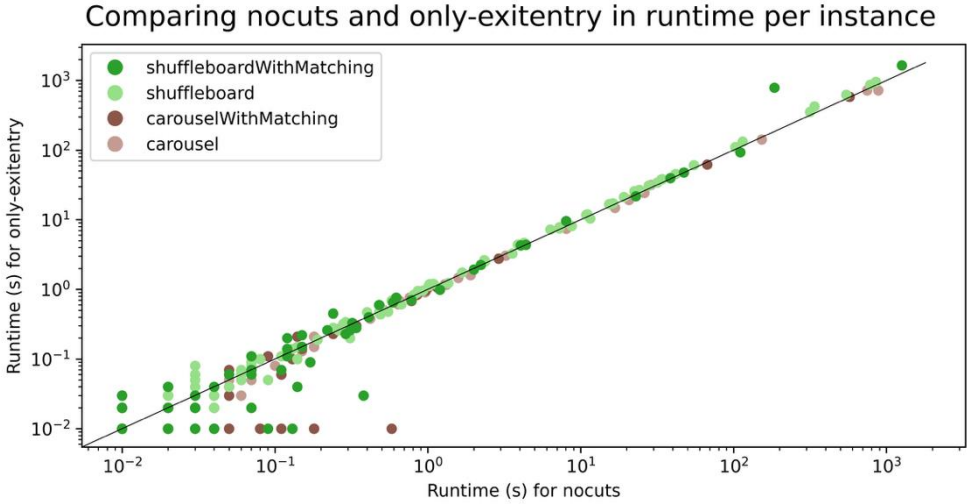
TU Delft

Exit-entry with matching

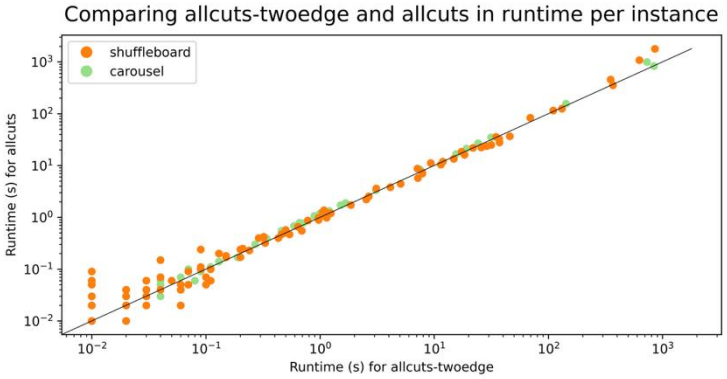


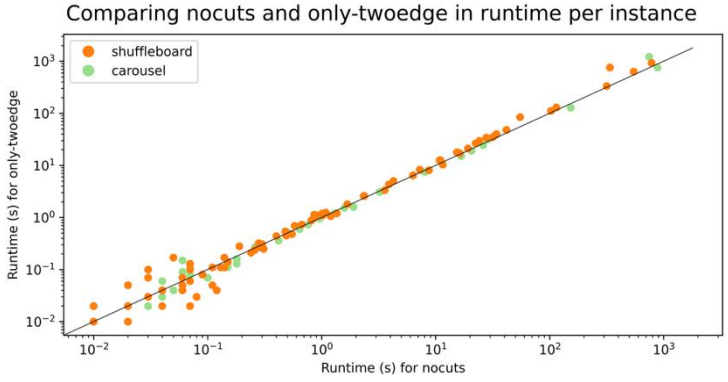


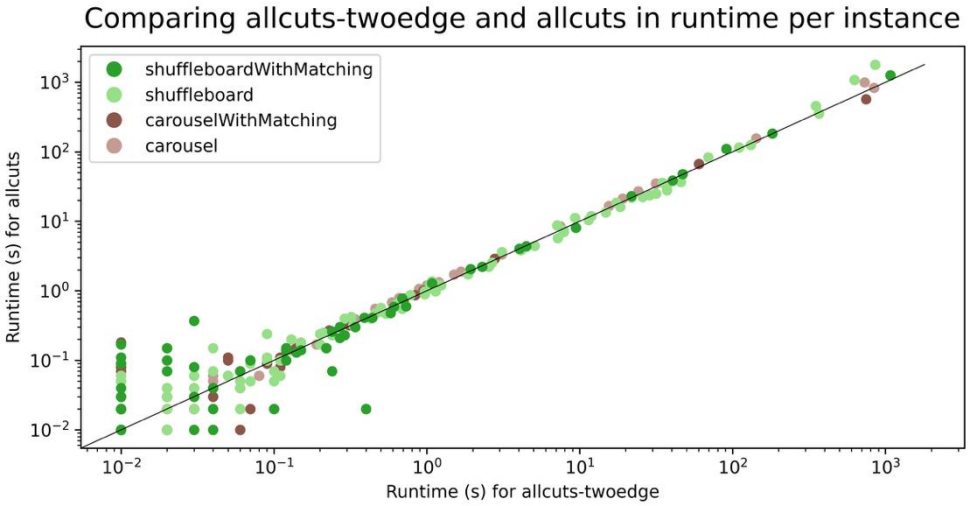


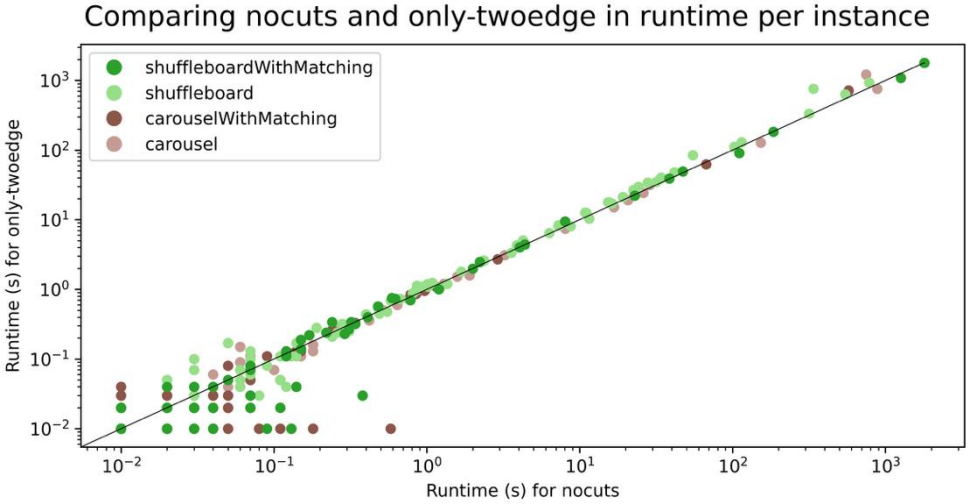


Two-edge with matching

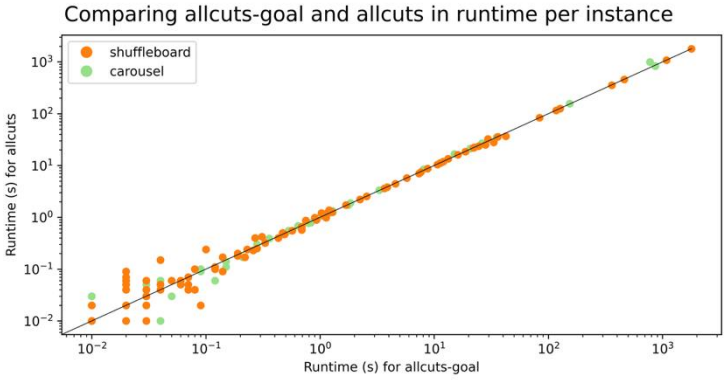


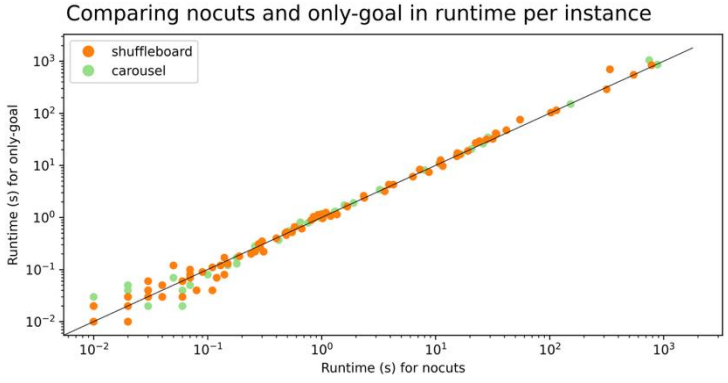


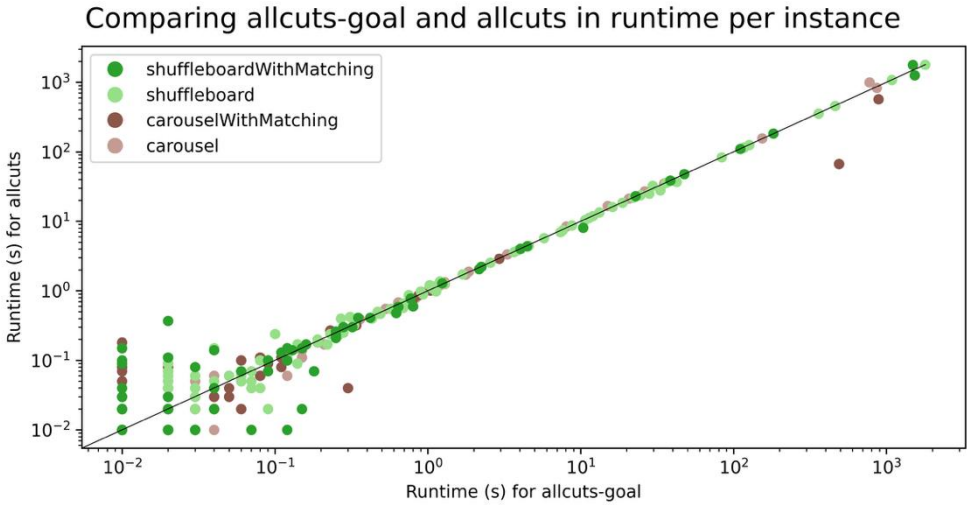


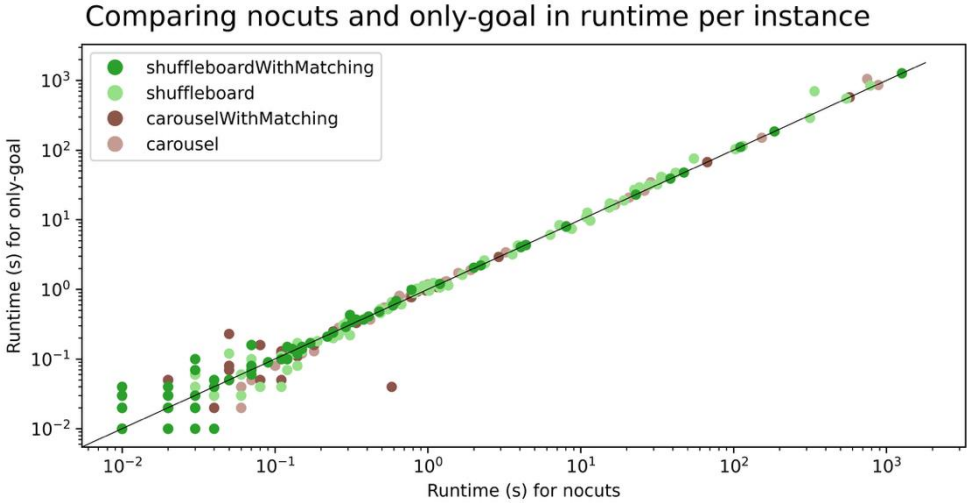


Goal with matching









Conclusions & future work

- Conclusion: Not much help from these 3 cuts, but:
  - most benefit seems to come from *A* caching* and *path length branching*
  - these 3 cuts mainly effective when there are many feasible solutions (grids & carrousel with matching)
- Future work:

Systematic comparison of related studies on the TPP.

Publication	Problem	Locomotive operation	Flexible train schedule	Modeling granularity	Safety requirement	Objective	Solution approach
Kroon et al. (1997)	TPP	No	No	Path-based	Headway	Max number of scheduled train	Branch and cut
Billionnet (2003)	TPP	No	No	Path-based	Headway	Feasible solution	Commercial solver
Freling et al. (2005)	TUSP	No	No	Route-based	Headway	Min shunting movements	MILP and column generation
Rodriguez (2007)	TSP	No	Yes	Section-based	RLSR	Min total delay	Commercial simulator
Corman et al. (2009)	CDR	No	Yes	Section-based	RLRR	Min maximum secondary delay	Branch and bound
Caprara et al. (2011)	TPP	No	Yes	Path-based	Headway	Min platform cost and train priority	Branch and price and cut
Lusby et al. (2011b)	TRP	No	No	Section-based	RLSR	Max profit	Branch and price
Sels et al. (2014)	TPP	No	No	Route-based	Headway	Min cancellation cost	Commercial solver
Pellegrini et al. (2014)	rtRTMP	No	Yes	Section-based	RLSR	Min maximum secondary delay	Rolling horizon
Petering et al. (2016)	TTP+TPP	No	Yes	Path-based	Headway	Min cycle length and train travel times	Commercial solver
Zhang et al. (2020)	TTP+TPP	No	Yes	Route-based	Headway	Min cancellation cost and train travel times	ADMM
Lu et al. (2022)	TPRP	No	Yes	Route-based	RLSR	Min train deviation	Rolling horizon
van den Broek et al. (2022)	TUSPwSS	No	No	–	–	Feasible shunting schedule	Local search
Xu and Dessouky (2022)	TSPwSS	No	No	Route-based	Headway	Min shunting cost	LR
This study	exTPP	Yes	Yes	Section-based	RLSR	Min train deviation	ADMM

Note: TPP — Train Platforming Problem; TUSP — Train Unit Shunting Problem; TSP — Train Scheduling Problem; CDR — Conflict Detection and Resolution; TRP — Train Routing Problem; rtRTMP — real-time Railway Traffic Management Problem; TTP — Train Timetabling Problem; TPRP — Train Platforming and Rescheduling Problem; TUSPwSS — Train Unit Shunting Problem with Service Scheduling; TSPwSS — Train Shunting Problem with Service Scheduling; LR — Lagrangian Relaxation; ADMM — Alternating Direction Method of Multipliers.

from: Zhang, Zhang, D'Ariano, Bosi, Lu, and Peng. "Optimal Platforming, Routing, and Scheduling of Trains and Locomotives in a Rail Passenger Station Yard." *Transportation Research Part C: Emerging Technologies* 152 (July 1, 2023)

Future work

- other cuts from MAPF: wait two-edge, wait-corridor, corridor conflicts, Rectangle Knapsack conflicts, or from train platforming (TPP): complete paths (including time)
- other ideas from MAPF: Conflict-Based Search (CBS)

- learning landmarks (Issa)
- use of constraint programming (with colleagues)
- shunting environment for benchmarking TUSPwSS
- dealing with disruptions (colleagues from University Utrecht)

Contact me if you want to know more: M.M.deWeerdt@tudelft.nl

TU Delft