Decision-making of Online Rescheduling Procedures Using Neuroevolution of Augmenting Topologies

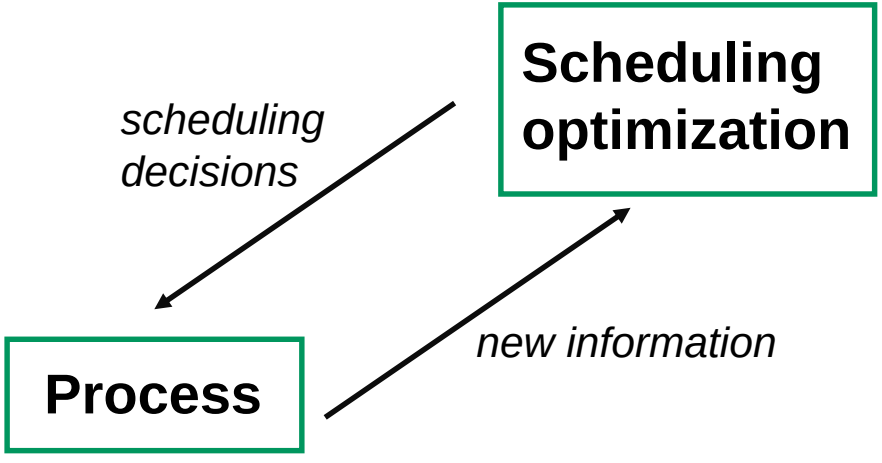
Teemu Ikonen and Iiro Harjunkoski

Aalto University
School of Chemical
Engineering

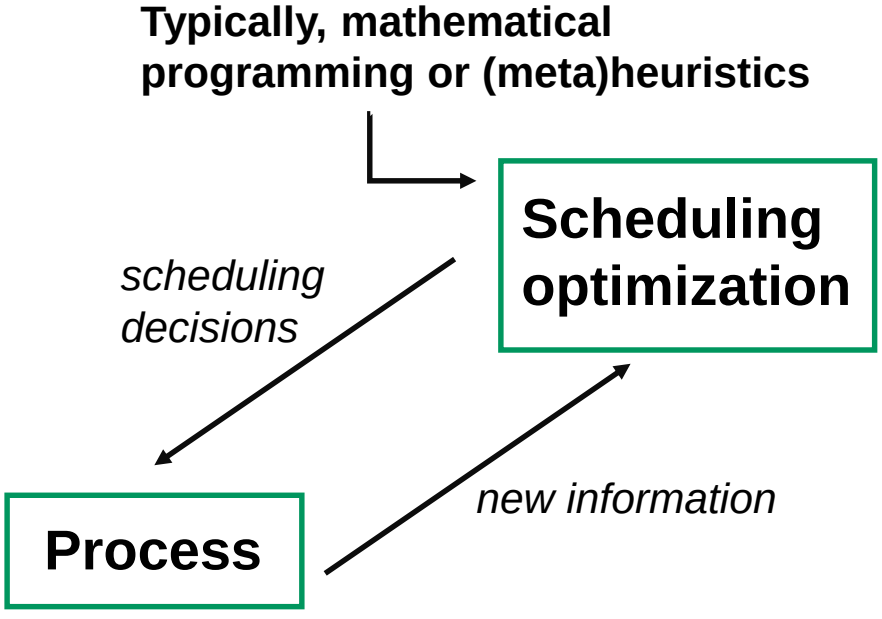
ESCAPE-29, Eindhoven, the NL

18th of June, 2019

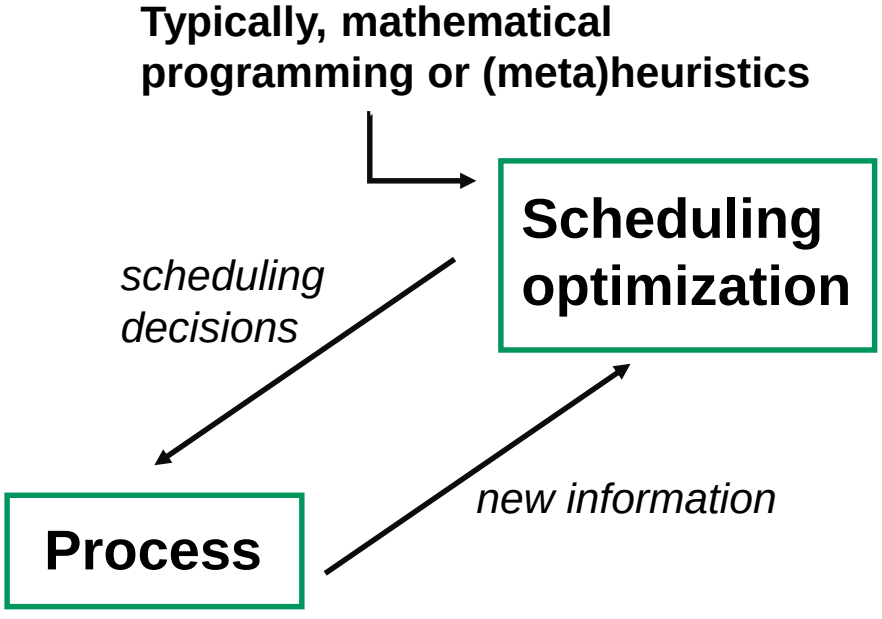
Introduction: Online Rescheduling Decisions



Introduction: Online Rescheduling Decisions



Introduction: Online Rescheduling Decisions



Questions

- **How far ahead should we schedule (i.e. horizon length)?**
- **When should we reschedule?**
- **Periodic or event-triggered?**
- **Mathematical programming or metaheuristics?**

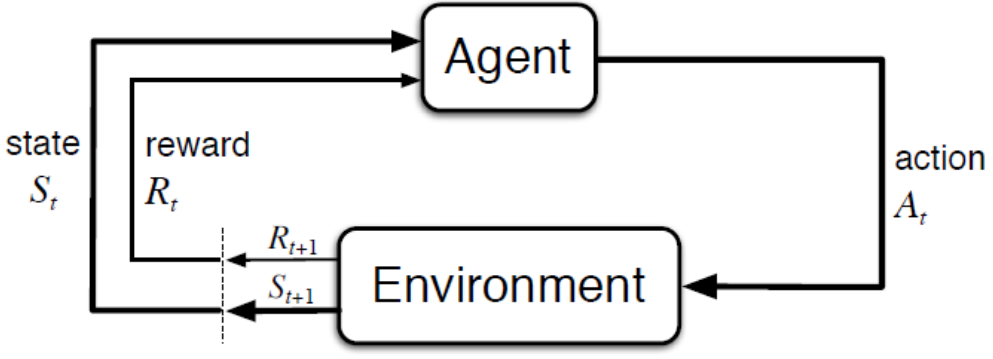
Outline

- **Reinforcement learning (RL)**
  - Introduction
  - Neuroevolution of Augmenting Topologies (NEAT)
- **Proposed approach**
- **Test case & periodic rescheduling as a reference method**
- **Conclusions**

Reinforcement Learning

Reinforcement Learning

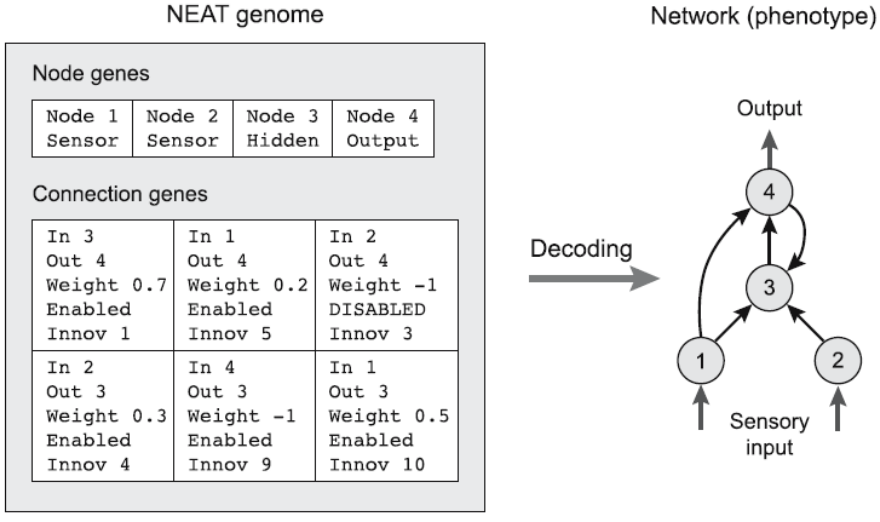
- **One of the three main branches of machine learning**
  - Along with supervised and unsupervised learning
- **A goal-seeking agent learns while interacting with an environment**
- **Exploration and exploitation**
- **A wide variety of different algorithms**
  - Q-learning, SARSA, Deep Q Network, etc.



Sutton and Barto (2018)

Neuroevolution of Augmenting Topologies (NEAT)

- **First proposed by Stanley and Miikkulainen (2002)**
- **A genetic algorithm that simultaneously evolves the topology and weighting parameters of a neural network**



Floreano et al. (2008)

Neuroevolution of Augmenting Topologies (NEAT) (cont'd)

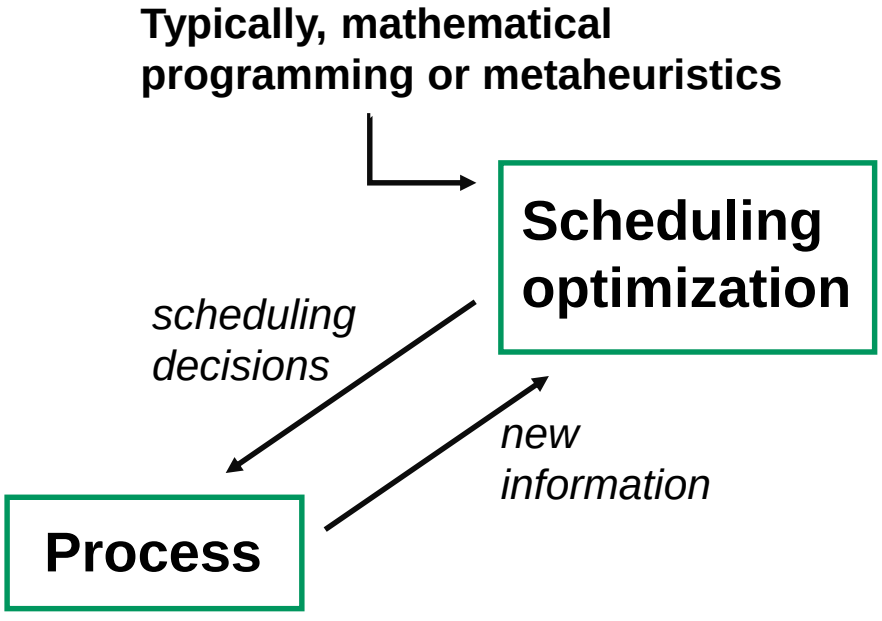
- **Complexity of neural network (NN) is minimized**
  - Initiated from very simple NNs
  - The complexity of NNs is incrementally increased during the evolution
- **The performance is reported to compare well against gradient-based backpropagation algorithms (Such et al., 2017)**
- **Hausknecht et al. (2014) applied NEAT to train a neural network to play 61 different Atari 2600 games**

RL in Scheduling

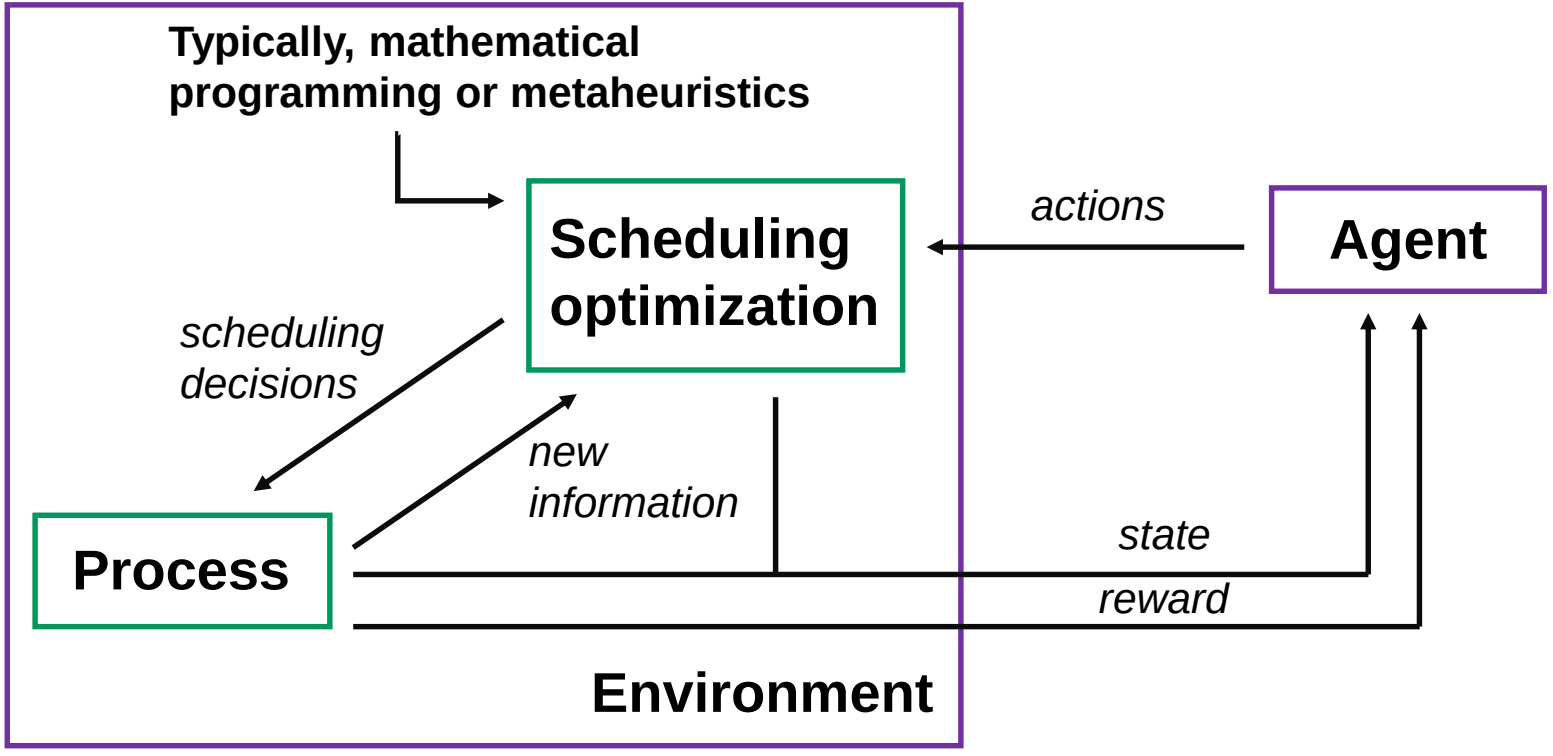
- **Learning dispatching rules from historical scheduling data**
  - Olafsson & Li (2010)
  - Ingimundardottir & Runarsson (2018)
  - …
- **Learning to make explicit scheduling decisions in a simulated environment**
  - Šemrov et al. (2016)
  - Atallah et al. (2019)
  - …

Proposed Approach

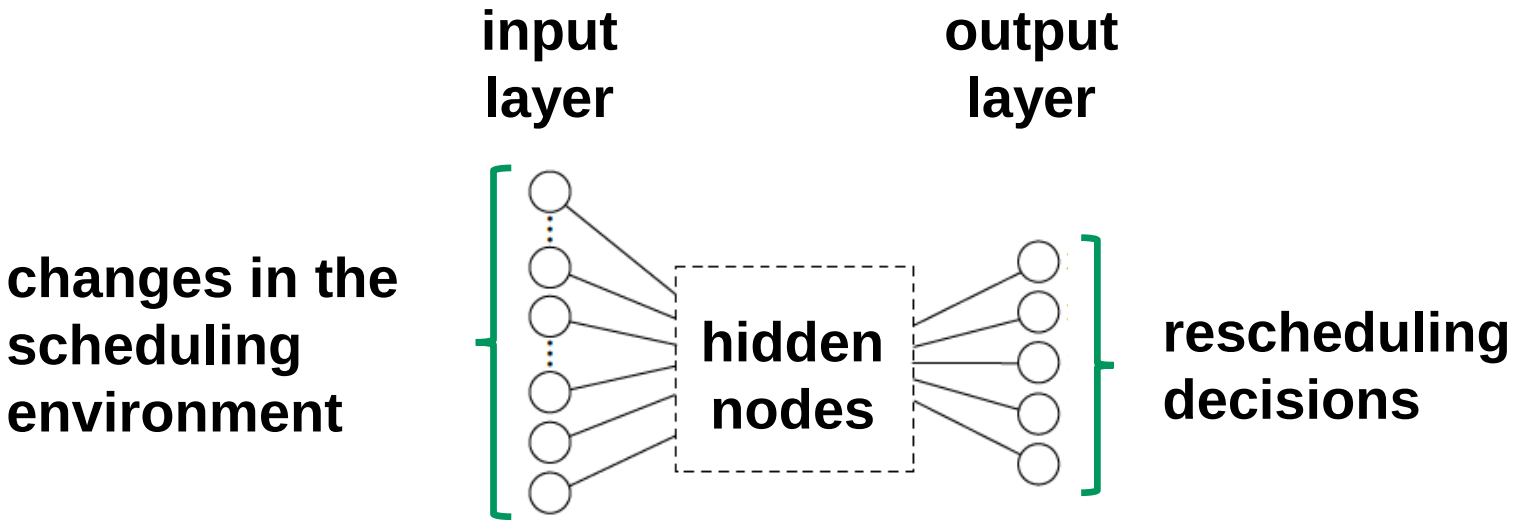
Agent & Environment



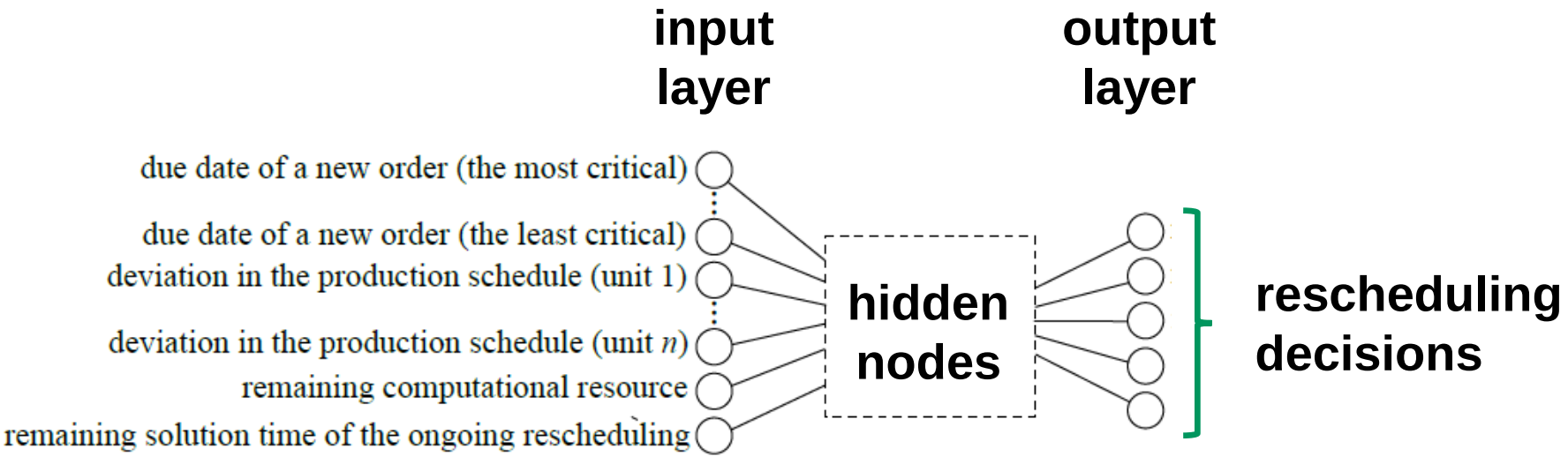
Agent & Environment



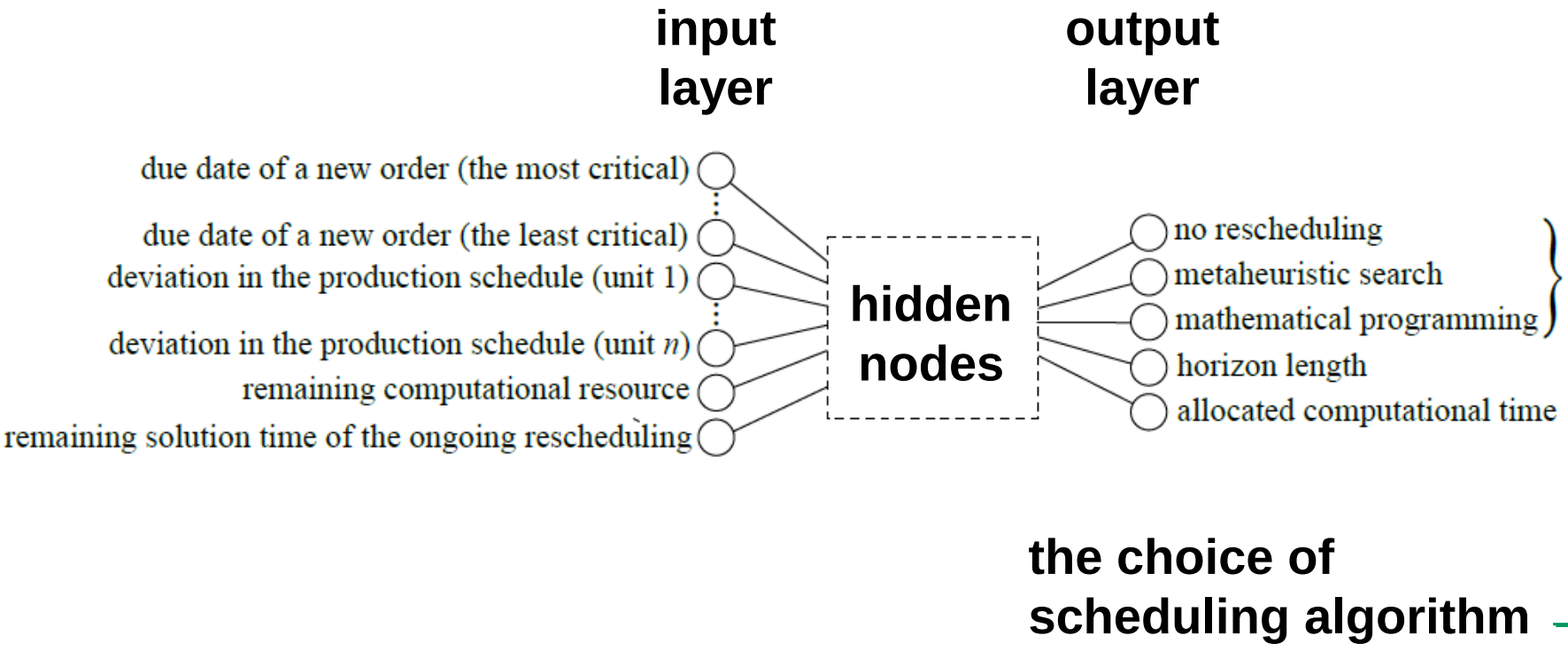
Proposed Approach



Proposed Approach



Proposed Approach



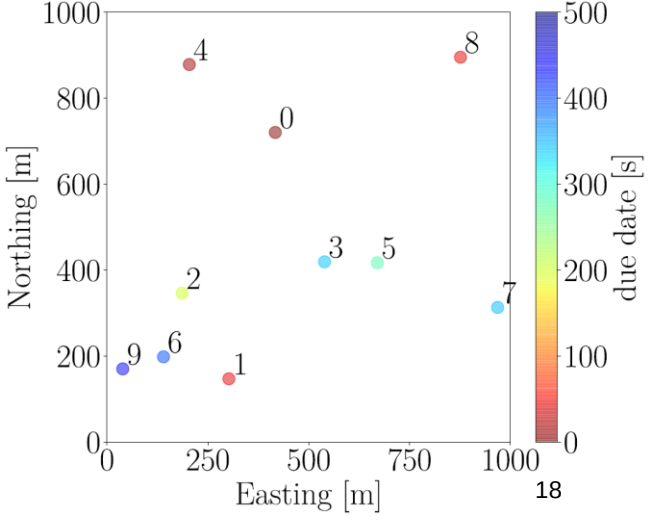
Test Case Using Periodic Rescheduling

Optimization Problem

- **A vehicle visits sites with due dates**
  - With a constant speed of 10 m/s
- **The objective is to minimize the delay sum of all visits**
- **Optimizer: ant colony optimization (ACO)**
- **New information is obtained during the process**
  - Rescheduling required
- **We use the rescheduling interval of 50 s, horizon length of 500 s, and allocate 5 s computational budget to each rescheduling**

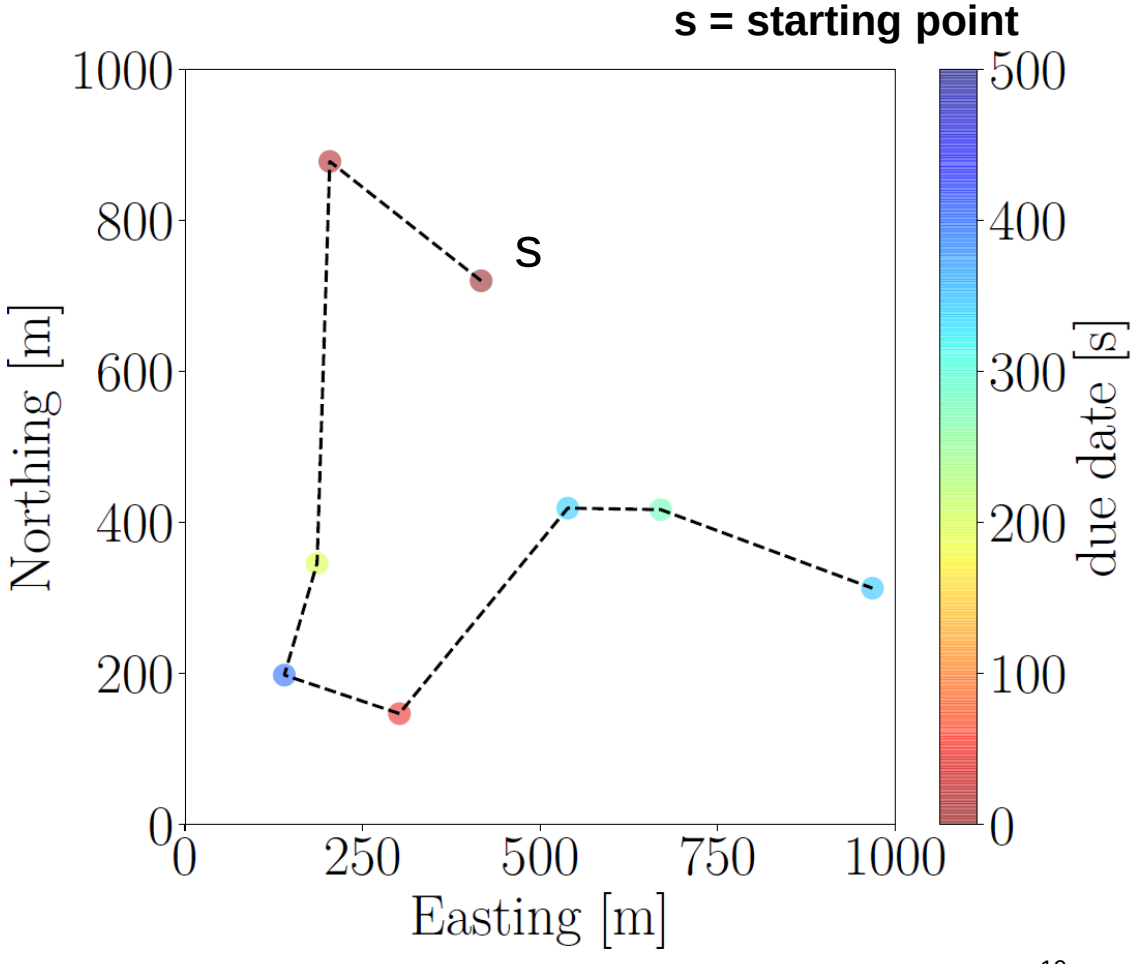
Small-scale example

site [-]	order date [s]	due date [s]
1	0	46
2	0	198
3	0	343
4	0	14
5	0	279
6	0	400
7	0	346
8	4	43
9	185	439



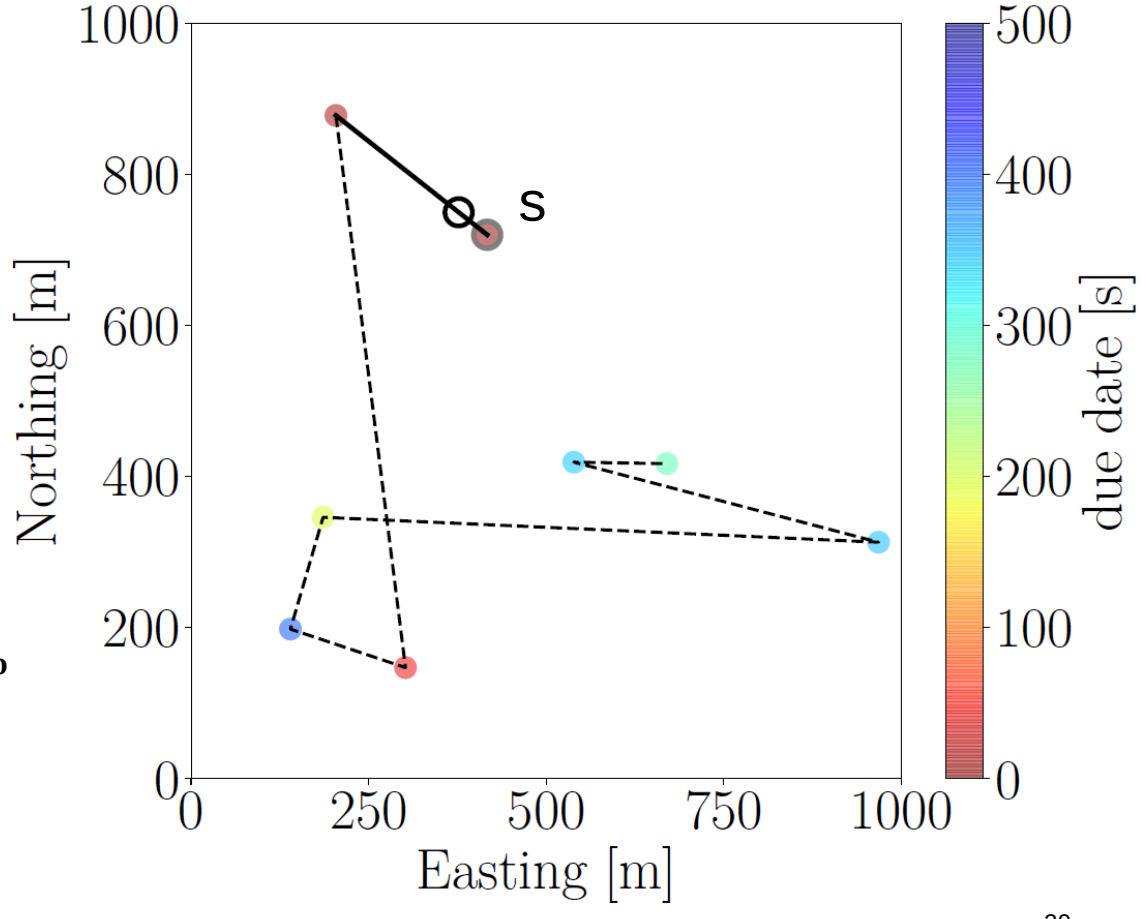
Small-scale Example: Initial Route

- $T_{\text{info}} = 0$ s
- $T_{\text{exe}} = 0$ s
- **Generated by greedy search (known due dates only)**



Small-scale Example: 1st ACO Run

- $T_{\text{info}} = 0$ s
- $T_{\text{exe}} = 5$ s
- The red point at the bottom is prioritized

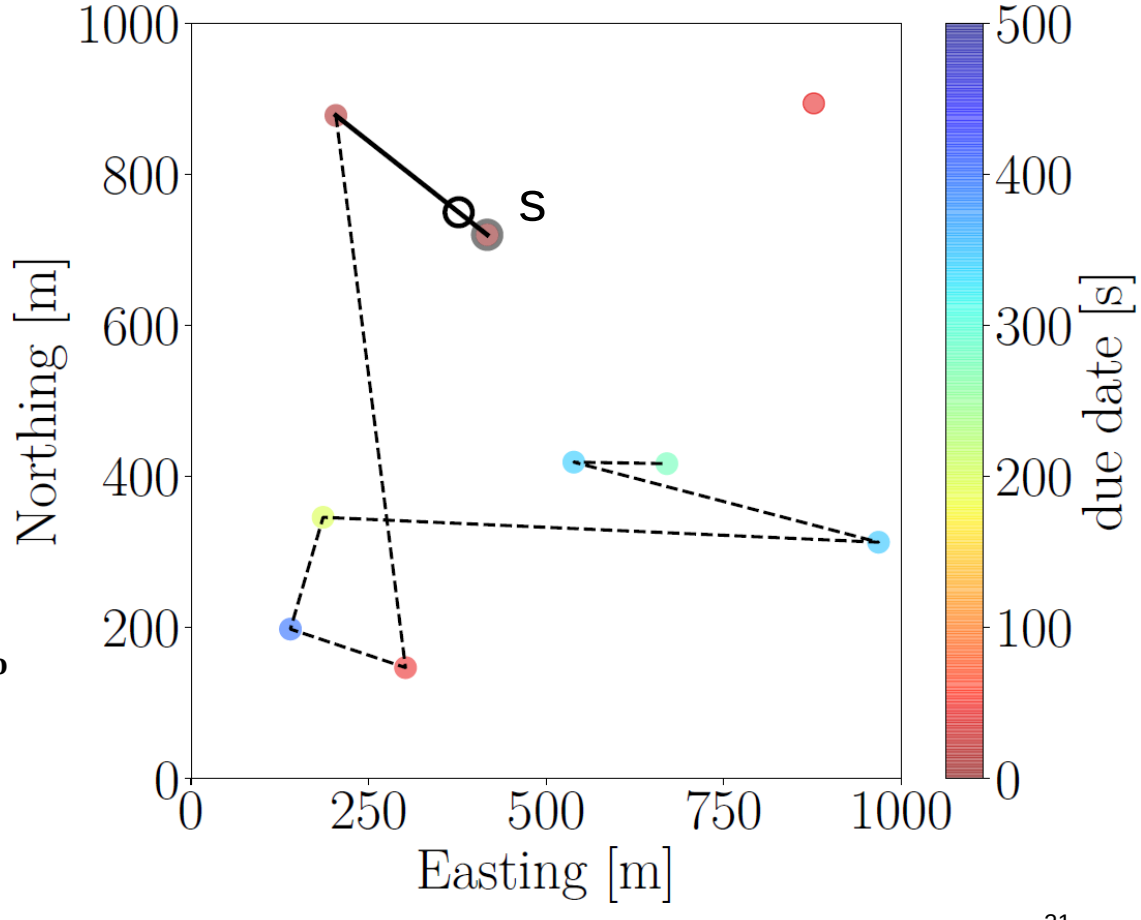


location of the vehicle at T_{info}

location of the vehicle at T_{exe}

Small-scale Example: 1st New Site

- $T_{\text{info}} = 0$ s
- $T_{\text{exe}} = 5$ s



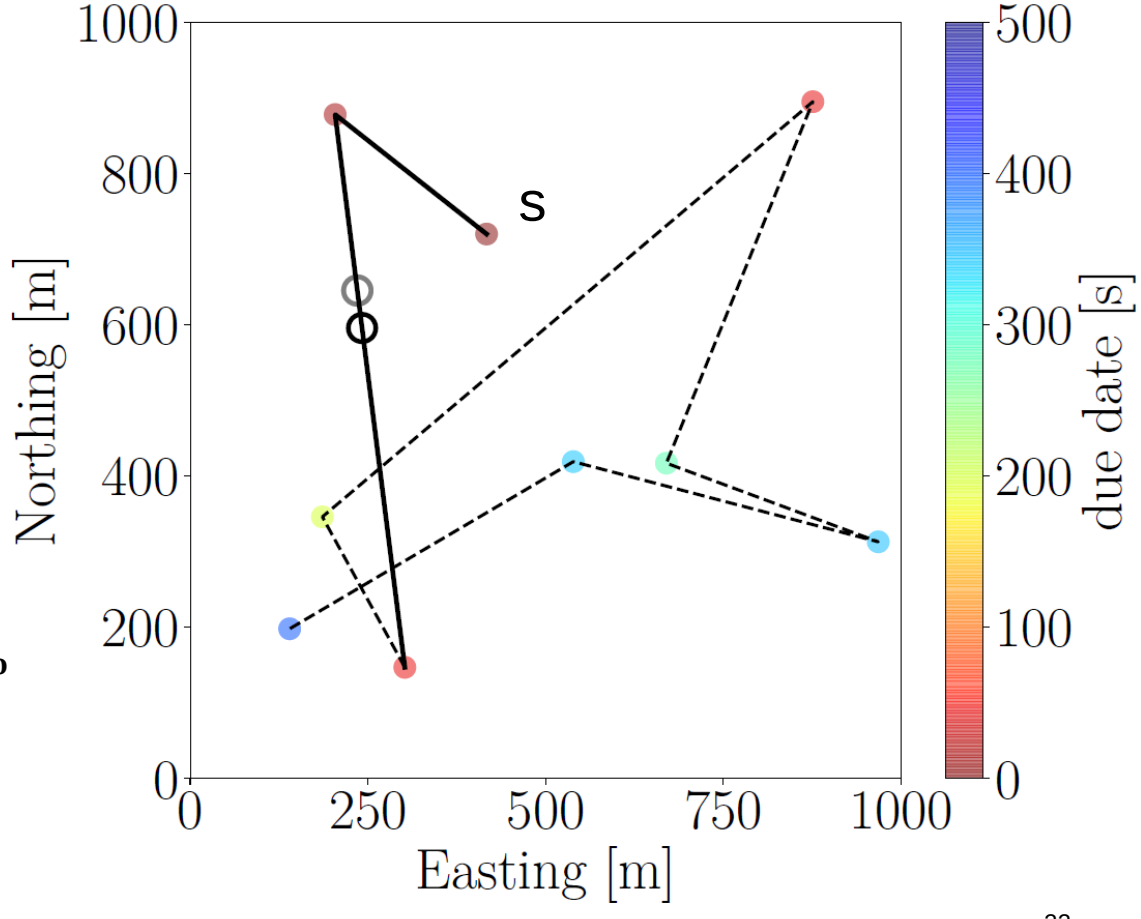
location of the vehicle at T_{info}

location of the vehicle at T_{exe}

Aalto University
School of Chemical
Engineering

Small-scale Example: 2nd ACO Run

- $T_{\text{info}} = 50$ s
- $T_{\text{exe}} = 55$ s
- A new point is included in the route
- Many changes



location of the vehicle at T_{info}

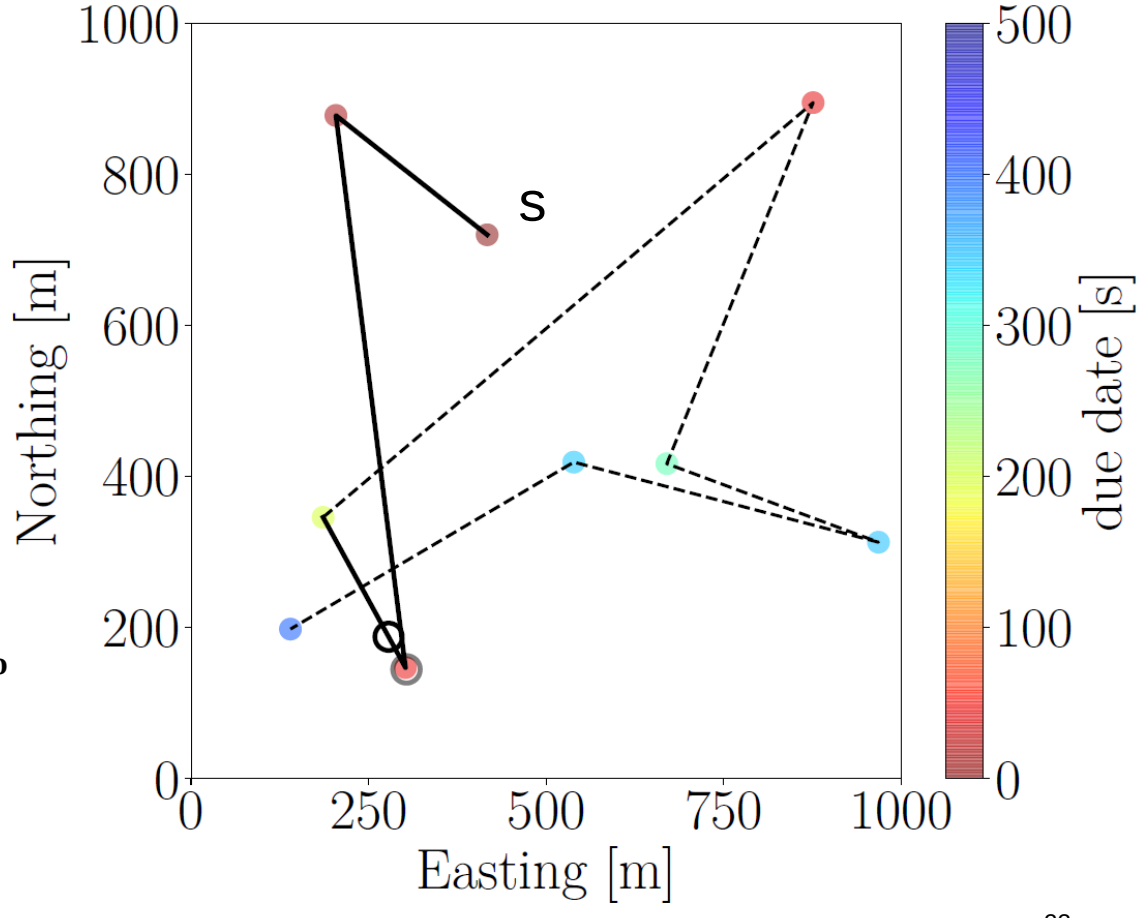
location of the vehicle at T_{exe}

- $T_{\text{info}} = 50$ s
- $T_{\text{exe}} = 55$ s
- A new point is included in the route
- Many changes

Aalto University
School of Chemical Engineering

Small-scale Example: 3rd ACO Run

- $T_{\text{info}} = 100$ s
- $T_{\text{exe}} = 105$ s



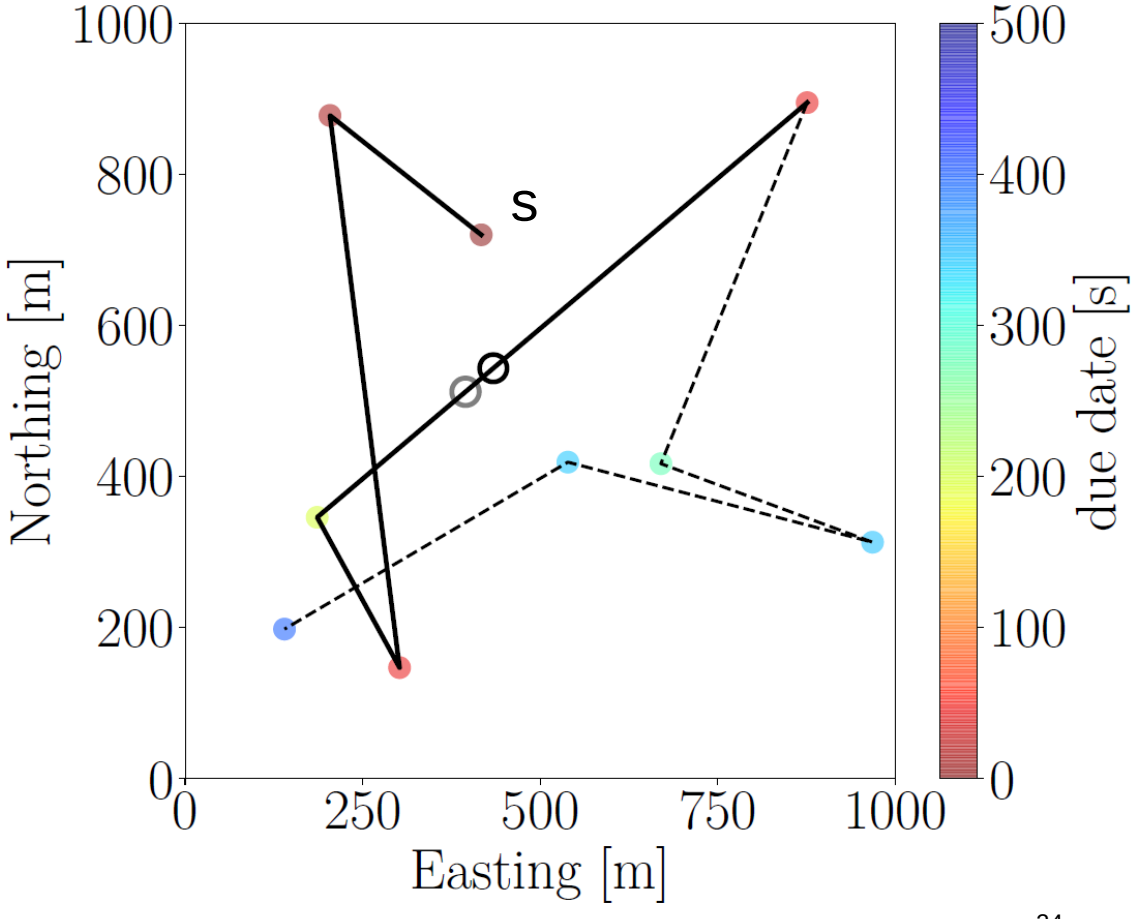
location of the vehicle at T_{info}

location of the vehicle at T_{exe}

Aalto University
School of Chemical
Engineering

Small-scale Example: 4th ACO Run

- $T_{\text{info}} = 150$ s
- $T_{\text{exe}} = 155$ s



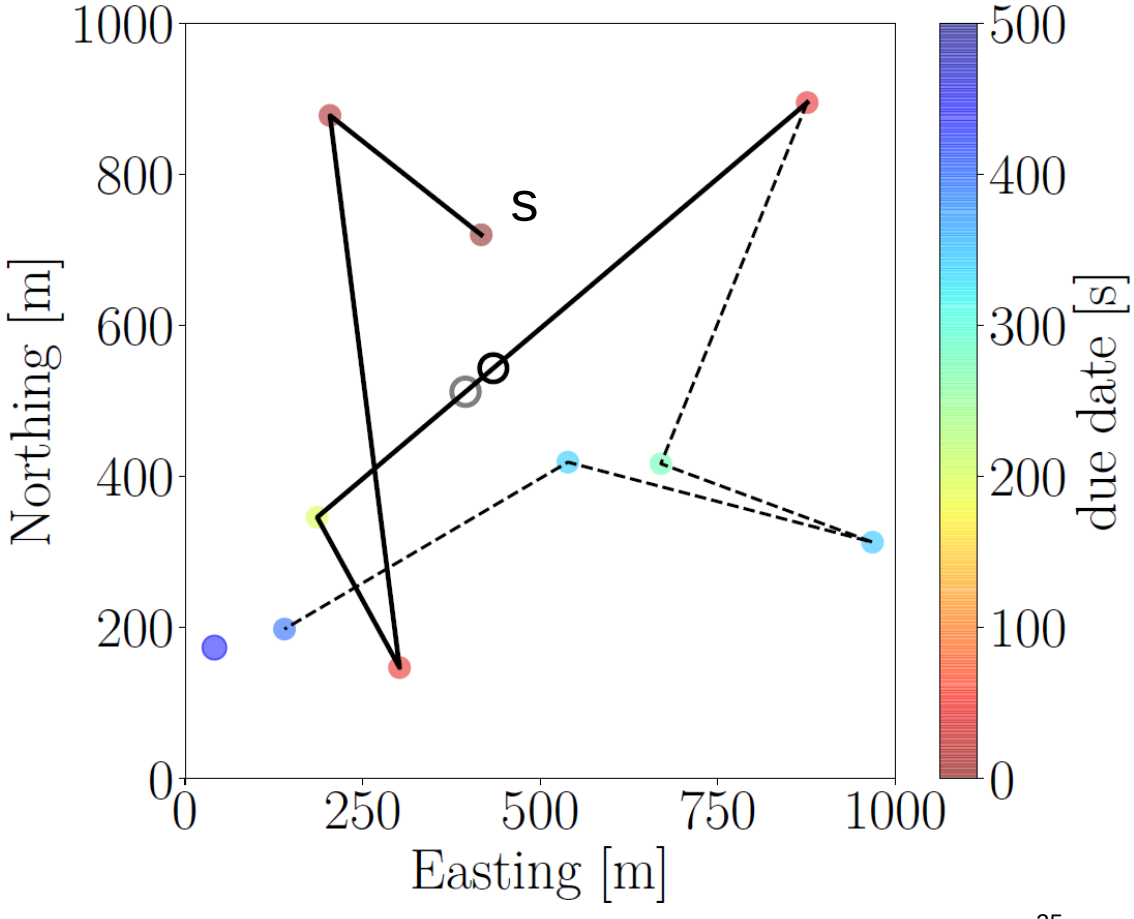
location of the vehicle at T_{info}

location of the vehicle at T_{exe}

Aalto University
School of Chemical
Engineering

Small-scale Example: 2nd New Site

- $T_{\text{info}} = 150$ s
- $T_{\text{exe}} = 155$ s



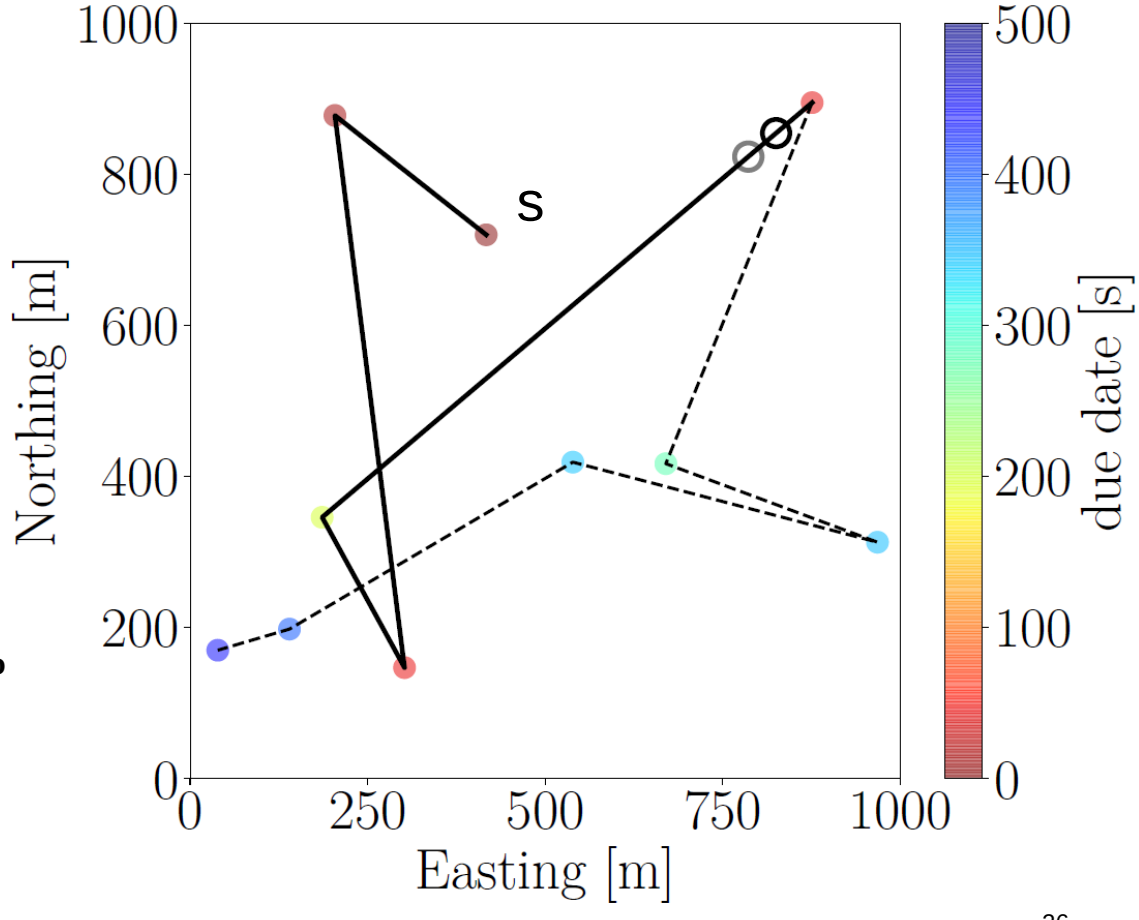
location of the vehicle at T_{info}

location of the vehicle at T_{exe}

Aalto University
School of Chemical
Engineering

Small-scale Example: 5th ACO Run

- $T_{\text{info}} = 200$ s
- $T_{\text{exe}} = 205$ s
- A new point is included in the schedule
- A minimal change in the schedule

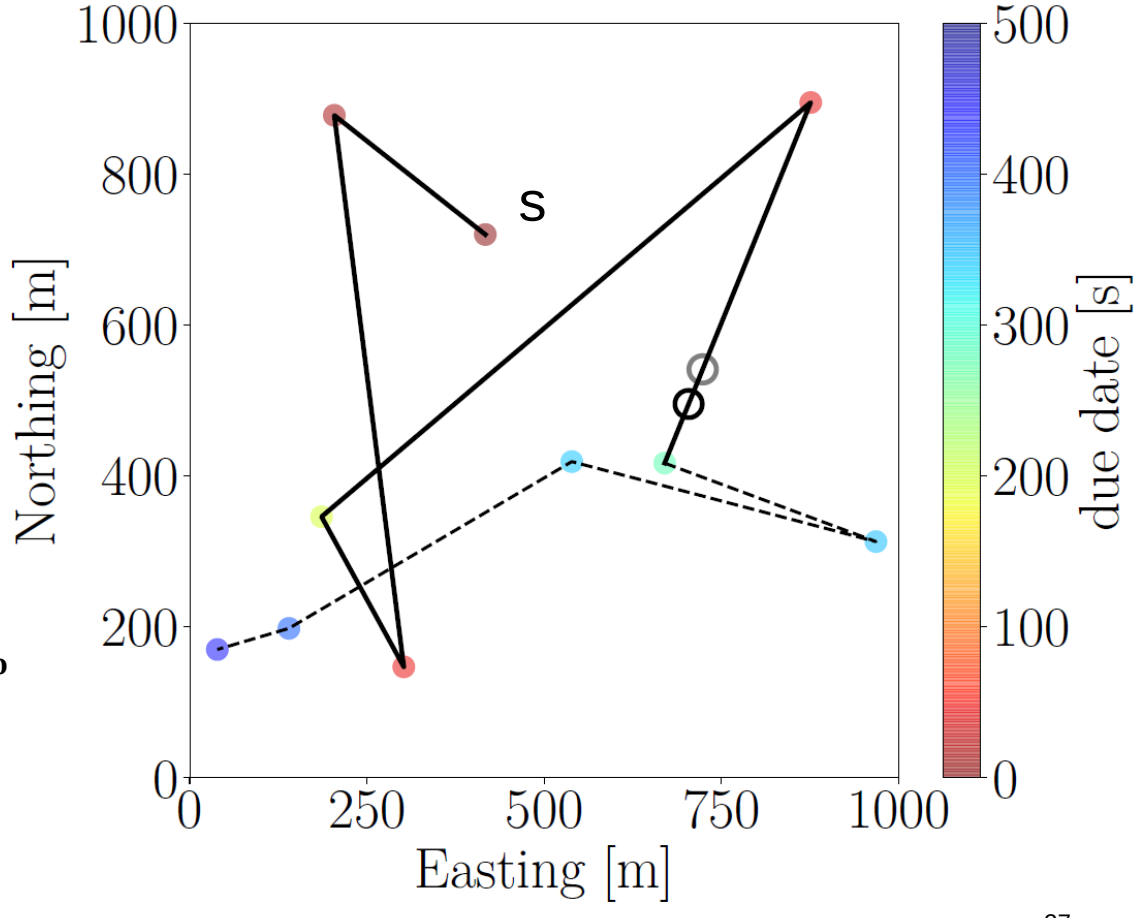


location of the vehicle at T_{info}

location of the vehicle at T_{exe}

Small-scale Example: 6th ACO Run

- $T_{\text{info}} = 250$ s
- $T_{\text{exe}} = 255$ s



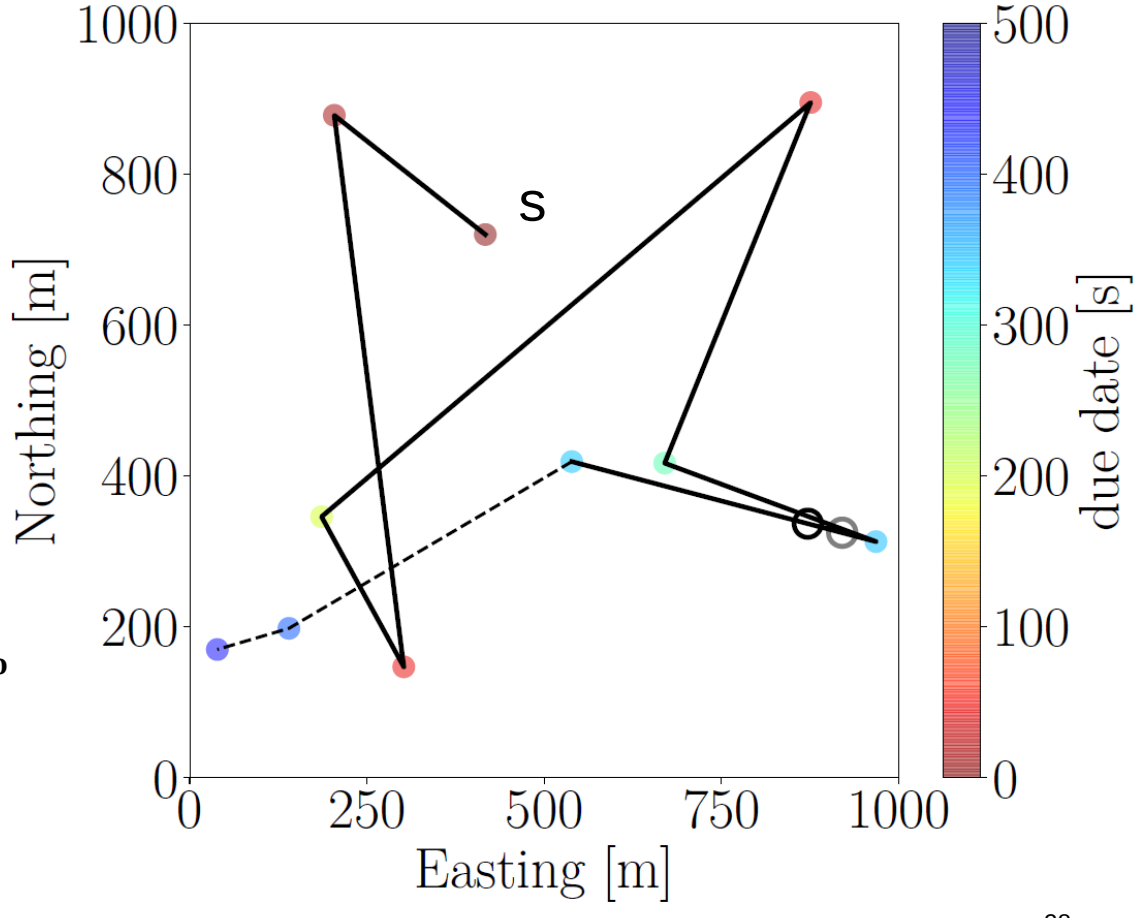
location of the vehicle at T_{info}

location of the vehicle at T_{exe}

Aalto University
School of Chemical
Engineering

Small-scale Example: 7th ACO Run

- $T_{\text{info}} = 300$ s
- $T_{\text{exe}} = 305$ s



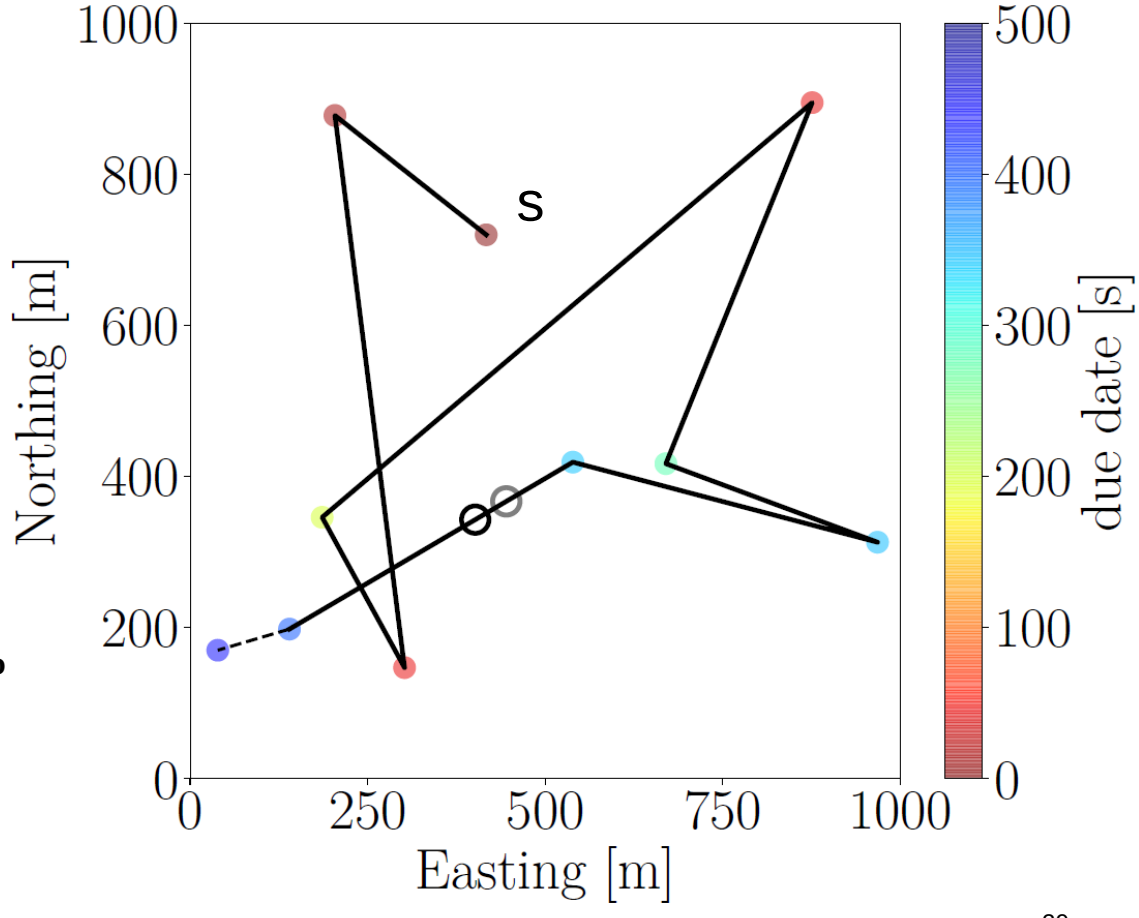
location of the vehicle at T_{info}

location of the vehicle at T_{exe}

Aalto University
School of Chemical
Engineering

Small-scale Example: 8th ACO Run

- $T_{\text{info}} = 300$ s
- $T_{\text{exe}} = 305$ s



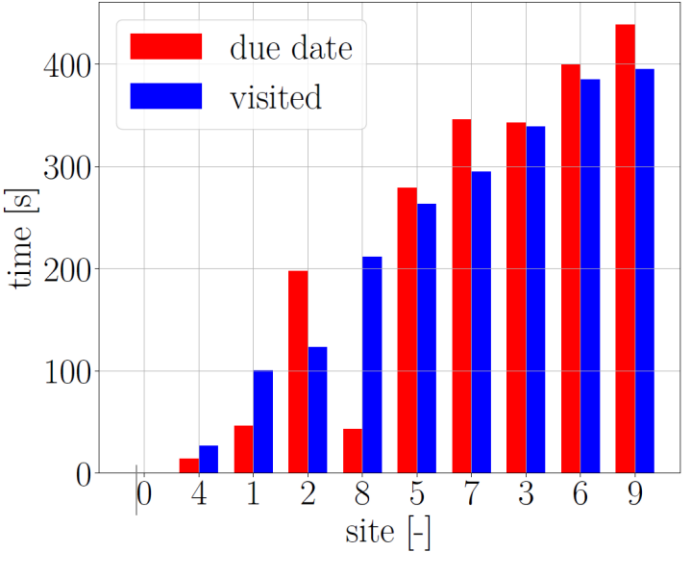
location of the vehicle at T_{info}

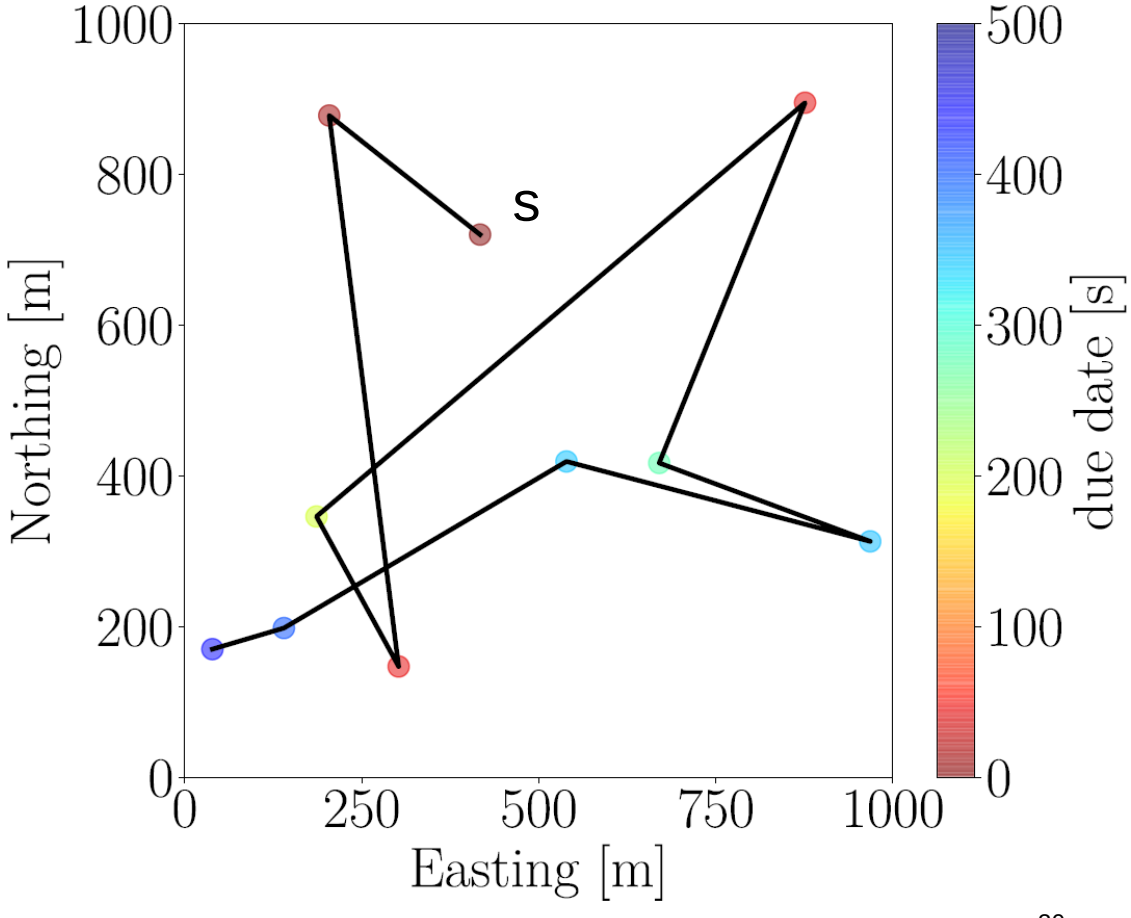
location of the vehicle at T_{exe}

Aalto University
School of Chemical
Engineering

Small-scale Example: Final Route

- **The final delay sum is 235.28 s**

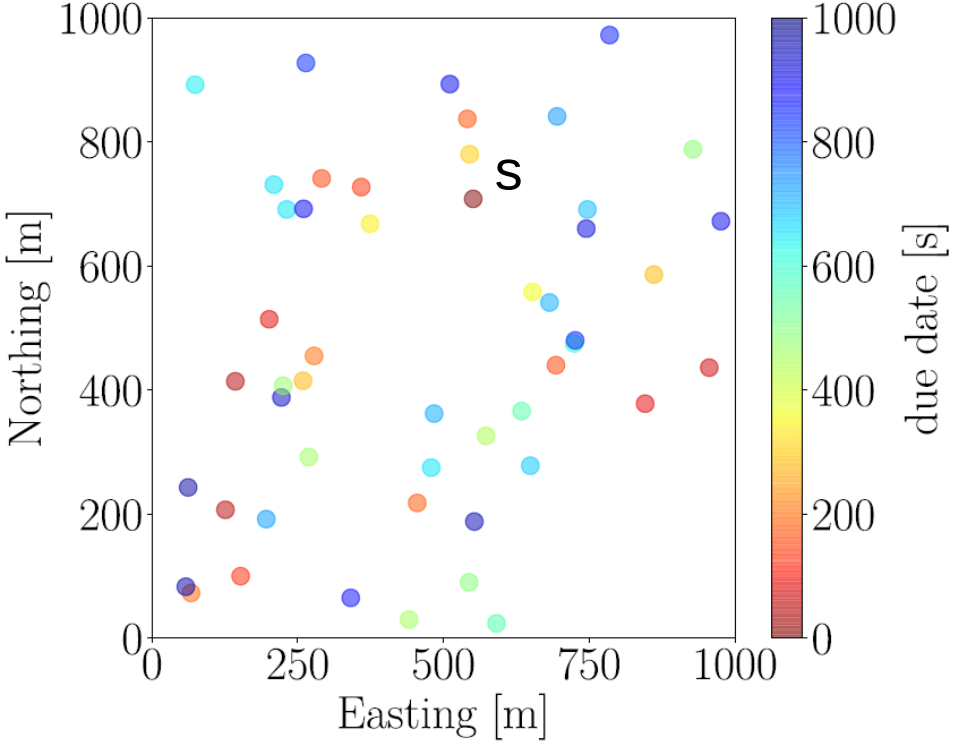




Aalto University
School of Chemical Engineering

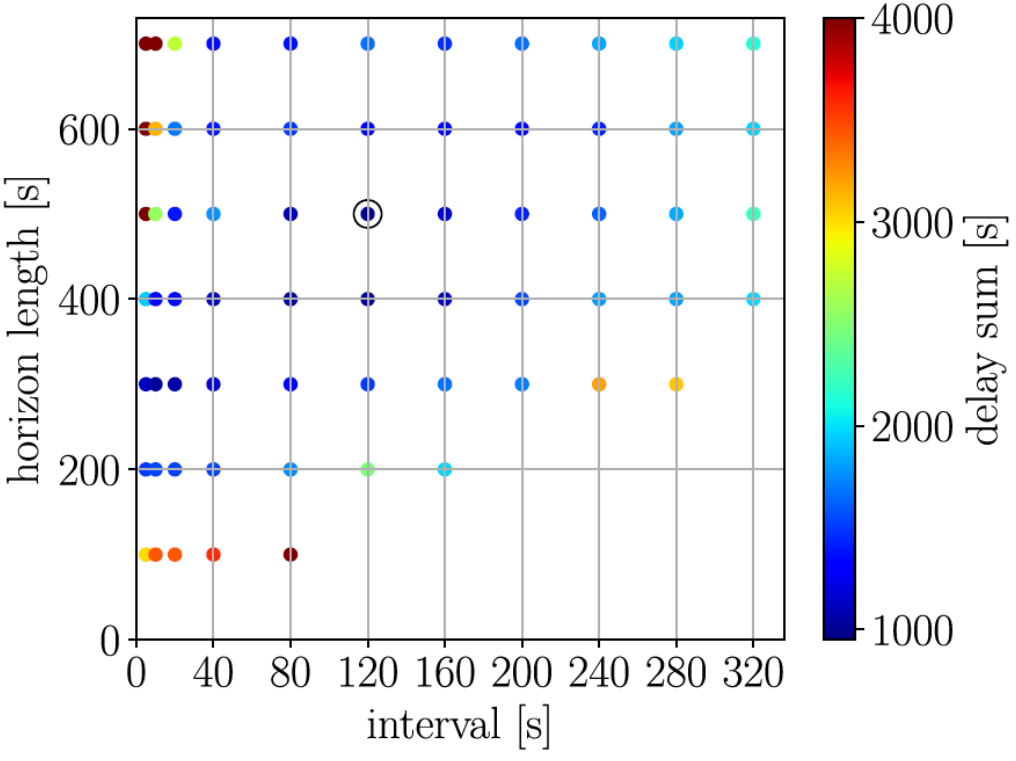
Large-scale Test Case: Optimization Problem

- **All due dates are randomly drawn from the uniform distribution of [0,1000 s]**
- **A total of 50 sites**
  - 40 sites are known at $t_{\text{info}} = 0$ s
  - The information of the remaining 10 sites arrive during the process
- **Computational budget for rescheduling is restricted to 50 s**



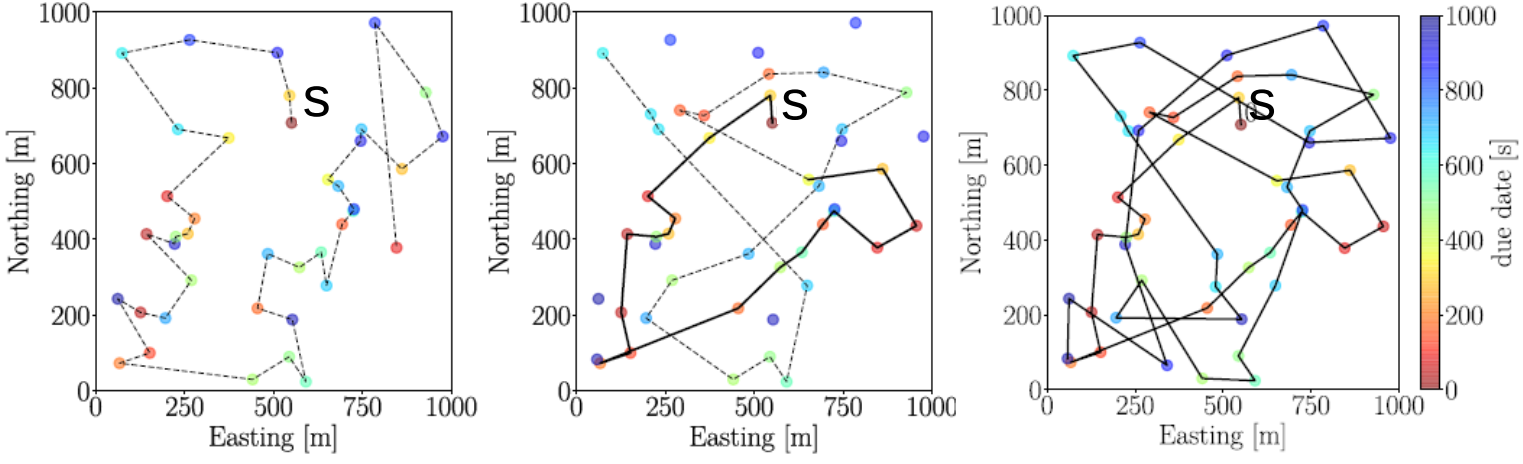
Large-scale Test Case: Parameter Tuning

- **Grid search in the space of horizon length and rescheduling interval**
- **In total 66 combinations**
- **The optimized parameters are (circled)**
  - Horizon length of 500 s
  - Rescheduling interval of 120 s



(the darkest red points exceed the scale)

Large-scale Test Case: Representative States



(a) initial route (b) $t_{\text{info}} = 240$ s, $t_{\text{exe}} = 245.56$ s (c) final realized route

- **Optimized scheduling interval yields $\lceil 1000 \text{ s} / 120 \text{ s} \rceil = 9$ rescheduling procedures**
- **The final delay sum is 992.89 s**

Conclusions

- **We propose an approach where**
  - The process and scheduling optimization together form the environment in reinforcement learning
  - An agent is trained by NEAT to make the rescheduling decisions
- **Test case**
  - New information causing minor or major recourse is demonstrated
  - We tuned the horizon length and rescheduling interval of periodic rescheduling (reference method)
- **Future work**
  - Investigate the proposed approach on the test case

References

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. *MIT press.*

Floreano, D., Dürr, P., & Mattiussi, C. (2008). Neuroevolution: from architectures to learning. *Evolutionary Intelligence*, 1(1), 47-62.

Such, F. P., Madhavan, V., Conti, E., Lehman, J., Stanley, K. O., & Clune, J. (2017). Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv*:1712.06567.

Hausknecht, M., Lehman, J., Miikkulainen, R., & Stone, P. (2014). A neuroevolution approach to general atari game playing. IEEE Transactions on Computational Intelligence and AI in Games, 6(4), 355-366.

Olafsson, S., & Li, X. (2010). Learning effective new single machine dispatching rules from optimal scheduling data. *International Journal of Production Economics*, 128(1), 118-126.

Ingimundardottir, H., & Runarsson, T. P. (2018). Discovering dispatching rules from data using imitation learning: A case study for the job-shop problem. *Journal of Scheduling*, 21(4), 413-428.

Šemrov, D., Marsetič, R., Žura, M., Todorovski, L., & Srdic, A. (2016). Reinforcement learning approach for train rescheduling on a single-track railway. *Transportation Research Part B: Methodological*, 86, 250-267.

Atallah, R. F., Assi, C. M., & Khabbaz, M. J. (2018). Scheduling the operation of a connected vehicular network using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, (99), 1-14.

Acknowledgement

The financial support from Academy of Finland, through project SINGPRO, is gratefully acknowledged.