Research Plan

Jeroen van Riel

December 2023

1 Planning

The main feedback I received from Marko and Rik on my bachelor end project was that I could have focused a bit better on a narrower selection of topics. Therefore, in light of the discussion we had about the online problem setting, my preference would be to first study the offline case (assuming infinite look-ahead $t_a = \infty$) a bit more.

1.1 Offline Scheduling

I would like to start with investigating the generalization capabilities of a reinforcement learning based method for the offline scheduling problem, starting with the single intersection case.

At this moment, the method of Zhang et al. seems the most promising candidate here, because I stumbled upon some work in which something very similar is done. For at least two weeks, I have had a note lying on my desk with the idea to look into the forks of GitHub repository containing the code of [1]. It turns out that one of these forks belongs to Rob Claassen, which graduated at TU/e on the application of precisely this deep reinforcement learning method for scheduling [2]. The below list outlines my rough idea of what this part of the project could look like.

- adapt Zhang et al. [1] method
  - run the code as-is (bare minimal, but done)
  - understand and their code (document this for myself)
  - understand the graph neural network embedding (GIN)
  - show how to adapt the disjunctive graph for our problem
  - implement this adaptation
- adapt Tassel et al. [3] method (optionally)
  - explain that this is also some form of dispatching
- assess learning ability

  - manually define problem instance distribution with some kind of structure
    - small platoons
    - large platoons
    - rate for each lane
  - using experiments, study whether the method learns to perform better on this class of problems after learning
    - extreme experiment (sanity check): just keep using the same instance and see whether the solution quality improves
  - study learning rate
  - study generalization
    - accross problem distributions (transfer)
    - accross problem sizes

Once we have gathered hands-on experience with the above approach, a possible next step would be to try and improve the method by considering alternative algorithm designs. As we have argued before, the above approach is based on dispatching. A natural question is whether it makes sense to change this schedule construction. Furthermore, the above method does not have any *search* mechanism.

- systematic study of solution methods
  - schedule construction/search, (dispatching rule vs. branch-and-bound)
  - platoon preservation theorem
    - finish the proof of the current version
    - compare this version to the version of Limpens and try to understand why their proof seemed complicated at first sight
  - motivate study of heuristics
    - manually trying to design heuristics can provide guidance in the design of the schedule construction mechanism, i.e., in how the *search tree is constructed*
    - even more, it could potentially provide simple rules for *pruning the search tree*
    - relate to Monte Carlo tree search

1.2 General Traffic Control

As we discussed during our last meeting, dropping the assumption of knowing all future vehicle arrivals, so assuming $t_a < \infty$, opens up a lot of interesting research directions. A good first step here would be to precisely formulate the general problem we are trying to solve, to which Section **??** was meant to be a first attempt.

At this moment, I think that we can extend the method of Zhang et al. towards an online setting, while also making the regularity assumption (see Definition 3.1 in [4]) to allow the two-stage decomposition. When a new vehicle arrives, this means that the scheduler can only "insert" this vehicle somewhere in the existing schedule. A natural first step is to compare the performance of such policy based on the Zhang et al. method to the exhaustive (Algorithm 1) and gated (Algorithm 2) policies as defined by Marko and Rik [4].

1.3 Preliminaries

Although the research plan is not yet fully specified, there are some aspects that will probably find their way into the project. The topics listed below require a brief introduction in the final report.

- job shop
  - variations
  - complexity
  - disjunctive graph
  - classic methods
    - dispatching rules
    - branch-and-bound
- mixed-integer linear programming
  - branch-and-bound
  - branch-and-cut
- learning objective [5]
  - exploit problem distribution
  - generalization
- reinforcement learning
  - state, transition function, actions, policy, reward
  - value-based methods and policy gradient methods
  - function approximation
    - neural methosd in general (DRL)
    - attention-based mechanisms
    - graph neural nets

1.4 Weekly Schedule

todo, some table here

References

[1] C. Zhang, W. Song, Z. Cao, J. Zhang, P. S. Tan, and C. Xu, “Learning to Dispatch for Job Shop Scheduling via Deep Reinforcement Learning,” Oct. 2020.

[2] R. Claassen, “Application of Deep Reinforcement Learning and Graph Neural Networks to the Machine Scheduling Problem,” Master’s thesis, Oct. 2022.

[3] P. Tassel, M. Gebser, and K. Schekotihin, “A Reinforcement Learning Environment For Job-Shop Scheduling,” Apr. 2021.

[4] R. W. Timmerman and M. A. A. Boon, “Platoon forming algorithms for intelligent street intersections,” *Transportmetrica A: Transport Science*, vol. 17, pp. 278–307, Feb. 2021.

[5] Y. Bengio, A. Lodi, and A. Prouvost, “Machine Learning for Combinatorial Optimization: A Methodological Tour d’Horizon,” Mar. 2020.