Multi-Agent Pathfinding for Railway Routing

Issa Hanou and Mathijs de Weerdt
Delft University of Technology
Poster ID 119

Papers&Code:
issahanou.github.io

i.k.hanou@tudelft.nl

Takeaways

- Train shunting problem: main aspect is routing
- Current approaches do this as afterthought
- Natural routing problem formulation: **Multi-Agent Pathfinding problem** (see below)
- Model **Train Unit Shunting Problem with Service Scheduling** with Multi-Agent Pathfinding
- Use AI models and approaches to tackle shunting

Talk to me about:
→ Bridging AI Planning with transportation problems
→ Considering Railway Hub Planning problem as a whole
→ Reusing knowledge from previous plans
→ Creating recognizable plans for human planners
→ Interest in our open-source shunting yard solver

Train Unit Shunting Problem with Service Scheduling (TUSPwSS)

- Arriving train unit compositions on a gateway track
- Requested departing train compositions on a gateway track
- Service tasks to be performed on a subset of train units
- Matching the arriving train unit compositions with requested departing compositions

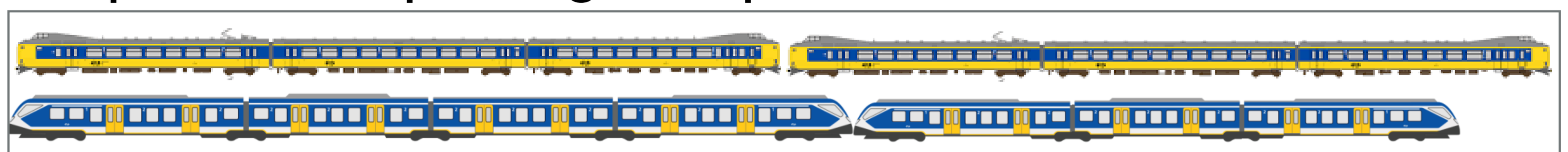

Figure 1: Two train unit compositions used in NL
Top: ICM3+ICM3 Bottom: SNG4+SNG3

Multi-Agent Pathfinding (MAPF)

- Agents with a start and goal location
- Find non-conflicting paths for all agents
- Group agents into teams with shared goals:
  - Target Assignment and Pathfinding problem
- Optimize for earliest finish time of all agents or shortest combined path length
- Complex problem: increases with number of agents
- Several approaches that solve to optimality

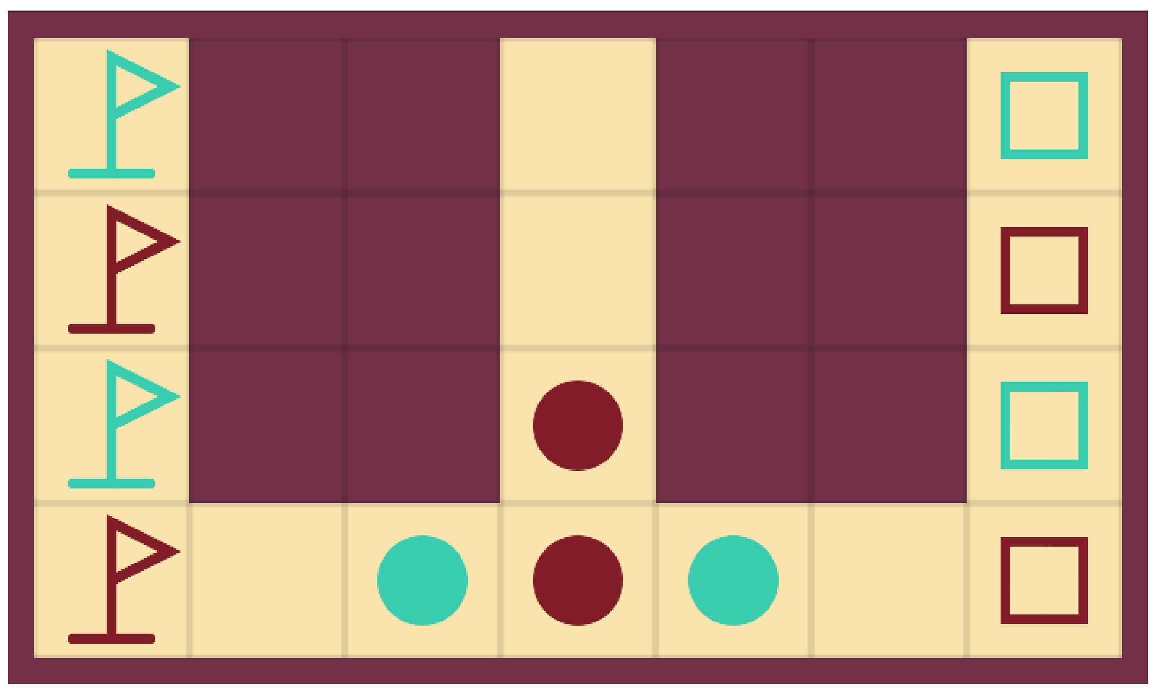

Figure 2: MAPF example with four agents (balls) in two teams (colors) with starting positions (squares) and goals (flags). In the current problem state, the brown agent has to move out of the way for the other agents to pass.

Current work

Extending MAPF model to fit TUSPwSS - *collab SINTEF*
1. Railway track layout graph structure (Fig. 3)
2. Different types of agents: length and speed (Fig. 1)
3. Model service tasks with intermediate goals (Fig. 3)
4. Matching problem with agent teams (Fig. 2)

Performance evaluation
- Adapt current AI search algorithm for MAPF to extensions
  - Maintain set of paths and detect conflicts
  - Store conflicts in constraint tree
  - Each node has constraints on individual paths
  - Resolve conflicts at high level by adding nodes to the tree with more constraints
  - In the low-level find new shortest paths for agent
- Compare this performance to **local-search approach**
  - Currently implementing open-source local search solver with simulator and instance generator

Extend model from shunting yard to **complete railway hub**
- Include station area and traffic through station

Future work

1. Finding similarities between railway hub plans
   - Partial Order Schedule as tree structure
   - Find similar tree shapes in other plans
   - Answer Set Programming
2. Finding high-level semantics of short action sequences to explain plan behaviour
   - Use hierarchical planning to label high-level actions and attach purpose
   - “Why move train A first and then train B?”
3. Finding must-reach subgoals to help reach the goal of a feasible solution
   - Landmarks are intermediate states
   - Provide interpretable milestones
4. Determining when to let a delayed train wait and when to let a different train pass - *MSc student Eric Kemmeren*
   - Use Safe Interval Path Planning to find safe traversals
   - Find breaking point when to let other train pass first

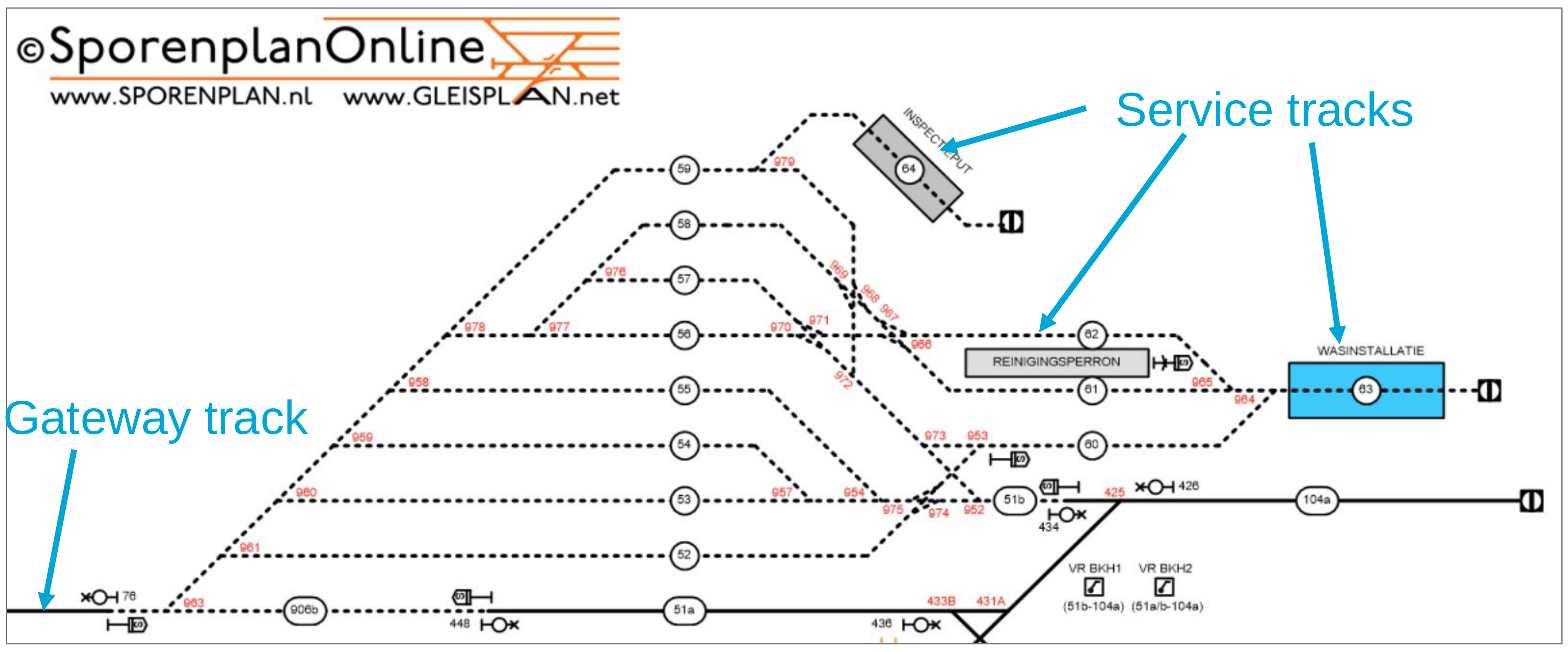

Figure 3: Shunting yard *Kleine Binckhorst* in The Hague, The Netherlands. MAPF application: each track is a location, route trains from start to goal.