TransportEditor – Creating and Visualising Transportation Problems and Plans

Ondrej Škopek and Roman Barták
Charles University, Faculty of Mathematics and Physics
Malostranské nám. 25, Praha 1, Czech Republic

Abstract
Transportation planning became a popular domain for automated planning systems and many domains in International Planning Competitions involve some form of a transportation problem. This paper introduces TransportEditor – a Java program for creating various variants of transportation domain models, for editing and visualising specific problem instances, for generating a plan by calling external or internal planners, for validating the plans for example by VAL, and finally for visualising and tracing the plans.

Introduction
Automated planning struggles from the lack of tools for editing and visualising planning domain models and plans. This is partly due to generality of the concept of automated planning, that makes it hard to develop general editors while covering peculiarities of a specific domain, and partly due to less interest in knowledge engineering aspects of planning. There exists only a few generally applicable tools such as itSimple (Vaquero et al. 2013), GIPO (Simpson et al. 2007), and Planning Domains (Muise 2016), PDDL editors such as PDDL Studio (Plch et al. 2012) and VIZ (Vodrážka and Chrpá 2010) and plan visualisers such as iGantt (Barták and Skalíký 2009) and VisPlan (Glinský and Barták 2013).

In this system demo we introduce TransportEditor – a Java program for creating various transportation domain models and plans that appear in various forms in International Planning Competitions (IPC).

Transport Domain
Transport domain has been introduced in IPC 2008 and it deals with transporting packages between locations using trucks. There is a directed graph defining locations and connections between them. Edges might be weighted to model distance, time or fuel consumption for moving between locations. Packages are initially waiting at some locations (nodes) and the goal is to deliver them to their destination locations using trucks. The trucks are also initially waiting at locations on the graph, so they may need to drive from one location to another location and it may be parameterised by length and fuel consumed when driving between the locations. Locations and connections define a directed graph that is visualised by the software. The user has the freedom to place the nodes (locations) anywhere in the 2D area, the
system tries (on demand) to display the graph in a visually pleasant way.

The other two types of objects in the domain are trucks and packages. For each package the user specifies its current location and its destination location and also its size to be used if trucks have limited capacity. For each truck, it is possible to specify its current and destination locations, as well as capacity (current and maximal) for packages and for fuel.

Figure 1 shows the TransportEditor when displaying a particular problem instance on the left panel. Domain models and problem instances can be saved in PDDL so they can be used by any PDDL-compliant planner.

Planning

The TransportEditor has its own built-in planners (based on DFS and A* with ad-hoc heuristics and control knowledge) that can be applied to currently opened problem instance. Moreover, the user can use any installed domain-independent planner that accepts domain models and problem instances in PDDL. It is only necessary to specify the executable command to call the planner.

Plan Visualisation, Validation, and Tracing

The final step in using the TransportEditor is visualising the plans, their validation, and dynamic tracing. The system can open the plan produced by an external planner and show it as a (time annotated) sequence of actions. It is also possible to display the plan in the form of a Gantt chart that makes it easier to spot concurrent actions for temporal planning. The plan can also be validated, for example by calling VAL (Howey and Long 2003). Again, this is done by specifying the executable command to call the validator. In addition to static visualisation of the plan, it is possible to trace it and to observe locations of trucks and packages in any step of the plan (see the right panel of Figure 1).

Summary

Transportation planning is an important real-life problem and a challenging benchmark for automated planners. TransportEditor attempts to simplify development of planning domain models with various constraints by visualising particular problem instances and plans and integrating planners and validators to simplify the workflow for human developers. The tool is implemented in Java so it is accessible for various computer platforms.

Acknowledgments

Roman Bartáš is supported by the Czech Science Foundation under the project P103-15-19877S.

References


