Operational transport planning with incidents

Jonne Zutt
Delft University of Technology
J.Zutt@tudelft.nl

Abstract

This research addresses robust incident management techniques in distributed environments with autonomous actors. Focusing on applications in operational pickup and delivery transportation planning, the distributed environment consists of a transport network, vehicles controlled by operational agents and pickup and delivery tasks with time-windows.

The two main challenges of this research are to explore and develop *context-aware* routing methods (routing methods that are aware of other traffic) and methods that are *robust* even in uncertain environments. This uncertainty in the pickup and delivery domain arises from changes (or newly arriving) transportation requests, malfunctioning transportation means or traffic jams in the transport network (e.g. due to accidents).

A multi-agent transport planning simulator (Traplas) has been developed for experimenting in the pickup and delivery domain. Where analyzing the problem does not lead to satisfying answers, Traplas is used to test and compare the different methods developed during this research.

Introduction

My research focuses on the development of efficient and robust incident management techniques in pickup and delivery transportation planning (Savelsbergh & Sol 1995). In pickup and delivery transportation, a set of transport resources (e.g. vehicles, airplanes, ships) have to execute a set of transportation requests by the customers. These requests to transport a freight from a source to a destination location have specified pickup and delivery time-windows within which the corresponding events must take place. The more a transport resources satisfies these constraints, the higher the reward received for executing the transportation tasks.

The transport network itself, which defines how the transport resources can move around, is also modeled by resources. The infrastructure resources have a distance, a speed limit and also a capacity. This capacity refers to the number of transport resources that can access the infrastructure resource simultaneously. If more transport resources plan to occupy an infrastructure resource than allowed by its capacity, we speak of a *conflict*.

Due to the limited capacities on the transport network efficient and robust *context-aware* routing methods are needed. In its most general form this problem is about finding a collectively optimal and feasible set of routes for several agents on an infrastructure with limited-capacity resources. Context awareness refers to the fact that an agent has to be aware of the consequences of the route planning by other agents, since his individually optimal route choice might be seriously affected by the route choices of other agents. Examples of applications where this route planning problem plays a role are automated guided vehicle (AGV) routing, scheduling in flexible manufacturing systems (FMS), waterway management, and airport taxiing.

Apart from efficiency, attention is paid to the *robustness* of the planning methods. *Incidents* are events that can disrupt the regular operation of a system, in this case composed of a transportation infrastructure (the network), vehicles and transportation jobs. Such incidents may occur due to, for example, sudden changes in the planning of transportation tasks, as the consequence of operational bottlenecks caused by inadequate execution of transportation jobs or due to malfunctioning of one or more individual resources (errors at the infrastructural resource or transportation resource level).

Related work

Several different approaches to context-aware routing are known in literature, which can be categorized according to the moment at which conflicts are resolved. One can try to *prevent* conflicts (or minimize the number of conflicts) by using social laws (Shoham & Tennenholtz 1995). Social laws are are certain rules, such as traffic laws, which, at the cost of some performance, can prevent conflicts by constraining the behavior of the agents. A simple example is that agents are only allowed to follow a Hamiltonian cycle.

Conflicts can be ignored until the very last moment and solved during execution. For example, agents can enter resources on a first-come-first-served basis. Or conflicts can be detected and solved in advance. Hatzack and Nebel suggested to transform the routing problem for a fleet of vehicles to Job Shop Scheduling with blocking (Hatzack & Nebel 2001), resulting in a two-phase approach where in the first phase a route is constructed for each agent and in the second phase, per agent, the entry times into the resources of the route are determined. Finally, the routing and conflict-resolution can be integrated. This is the approach taken by by (Kim & Tanchoco 1991).

The real world is highly dynamic, many disruptions and task modifications occur, leading to the necessity of incident management methods. Incident management methods can be distinguished into *pro-active* and *reactive* methods. Pro-active methods attempt to create robust schedules, while reactive methods recover from incidents at the moment they occur.

One pro-active approach to incident management is to generate robust schedules that are able to absorb a certain amount of disruptions without the need for replanning. Gao describes a technique called temporal protection (Gao 1995). For each location and vehicle historical statistics are maintained about its reliability. Then, according to this data the duration of actions such as drive, load, etc., are extended with some temporal slack. Davenport et al. developed time-window slack and focused time-window slack methods (Davenport, Gefflot, & Beck 2001). Instead of hiding the temporal slack in the durations of actions the time-window slack method modifies the problem definition a little, such that the scheduling algorithm can reason about the slack.

Examples of reactive methods can be found in the field of robot path planning in partially-known environments. The Lifelong Planning A* (LPA*) and D* variants (Stentz 1994; Koenig & Likhachev 2002) are similar to the well-known best-first search algorithm A*, but try to improve on this by not starting from scratch when a small disturbance occurs. For example, while a robot approaches its goal location it obtains new information from sensing its local environment. This data might change the travel costs (e.g. suddenly it detects a wall). In that case, a basic shortest path algorithm would have to restart from scratch. The D* or LPA* algorithms are able to locally propagate this change in costs, using stored data from the previous computations, and are potentially faster in computing the new optimal shortest path.

Approach and results

In this research multi-agent systems are applied to pickup and delivery problems. The advantages of multi-agent systems are that *(i)* they are *decentralized*, which matches the intention to create robust systems (no central point of failure), and that *(ii)* the problem can be modeled naturally (the planners for the transport resources and customers can be represented by agents).

As a baseline comparison for the operational pickup and delivery transportation problem, a set of simple traffic rules is used to solve conflicts. For example, the entry into location resources by the agents is regulated using heuristics, such as first-come-first-served, or that use information such as the reward or deadlines of the tasks they are executing, or the amount of delay they already had. We compare this to informed planning, where the agents make public reservations for their plans, so that other agents can take this into account while planning for themselves.

Subsequently, we attempt to improve on informed planning, by looking at the method of Hatzack and Nebel that transforms the problem to Job Shop Scheduling with blocking (Hatzack & Nebel 2001). Instead of creating a plan for each agent sequentially, which is the case in the informed method, a refinement is to schedule (and reserve) only the first part of a plan. In each round, a winner can be determined using the aforementioned heuristics of the baseline method. This winning agent continues with the next part of its plan. Another improvement is to integrate this scheduling process with rerouting. With a certain probability (for speed-up) the agent can consider alternative routes when determining its bid for the current round.

Finally, collaboration is considered. There are two potential ways in which the agents can improve joint performance by collaboration. First, they can solve conflicts by negotiating on who claims a resource first (by comparing plans in which the one agent goes first to plans in which the other agent goes first). Second, tasks can be re-assigned to another agent. This can be profitable if the costs for executing a task is higher than the reward that the agent obtains. Reassigning freight that is already loaded only occurs if transhipment is allowed.

To gather empirical results on the efficiency and robustness of the planning methods, an agent-based transport planning simulator called Traplas, available at `http://traplas.sf.net`, is developed and a benchmark set is constructed for pickup and delivery transport planning problems.

The main independent variables in these experiments are the planning method used, the topology and size of the transport network, the number of agents (vehicles) and the rate of incidents. The dependent variables are total reward (as defined per task), total costs, tardiness of task execution, agent and network utilization and the cpu-time needed to perform the simulations.

With these experiments the robustness of the methods can be verified empirically by increasing the rate of incidents. Among the replanning heuristics is a slack inserting heuristic; this heuristic performs somewhat inferior without incidents, but experiments have to show how it competes when the rate of incidents increases. Also, the other planning methods are fast, which means it could be possible to start planning from scratch if an incident occurs. If this turns out to be too slow, our simulation tool also includes some variants of LPA*.

The first results have shown that the methods described above are indeed listed in non-decreasing order of performance, and they are also listed in non-decreasing order of cpu-time consumption. The uninformed baseline methods, for any of the used traffic rules, produces bad results. Without incidents the later mentioned methods do not have significantly better performance. Hence, the additional cpu-time required does not pay out here. However, for an increasing rate of incidents, the situation changes and the later mentioned methods are more robust.

Results on the context-aware routing methods can be found in our conference paper (ter Mors, Zutt, & Witteveen 2007). For experiments on the performance and robustness of transportation routing methods, see (Zutt & Witteveen 2006). In the future, experiments will be done on the influence of the topology (e.g., lattice, smallworld, random) and size of the transport network.

Relevance

This research addresses context-aware routing and scheduling methods that are as efficient as possible. Different methods have been developed and are compared to each other. In these experiments the transportation request workload, the number of transport resources, and the topology and size of the network were varied. The aim of this comparison is to obtain information on how much and what type of information should be communicated between agents in order to achieve high performance of the total system.

Furthermore, this research aims to develop methods that are robust. Therefore, the robustness (sensitivity analysis) of different methods under increasing density of disruptions, such as malfunctioning resources, is compared. Finally, empirical data on the interaction effects with the transport network topology are investigated. This will hopefully lead to a better understanding of the relation between performance, robustness and the transport network structure.

Acknowledgments

This research has been supported by the TNO-TRAIL project (16) on *Fault detection and recovery in multi-modal transportation networks with autonomous mobile actors* and by the Dutch Ministry of Economic affairs under the SENTER TSIT program on *Cybernetic Incident Management* (TSIT2021).

References

Davenport, A. J.; Gefflot, C.; and Beck, J. C. 2001. Slack-based techniques for robust schedules. In *Proceedings of the Sixth European Conference on Planning (ECP-2001)*.

Gao, H. 1995. Building robust schedules using temportal protection - an empirical study of constraint based scheduling under machine failure uncertainty. Master's thesis, University of Toronto, Toronto, Ontario, Canada.

Hatzack, W., and Nebel, B. 2001. The operational traffic problem: Computational complexity and solutions. In Cesta, A., ed., *Proceedings of the 6th European Conference on Planning (ECP'01)*.

Kim, C. W., and Tanchoco, J. 1991. Conflict-free shortest-time bidirectional AGV routeing. *International Journal of Production Research* 29(1):2377–2391.

Koenig, S., and Likhachev, M. 2002. Improved fast replanning for robot navigation in unknown terrain. Technical report, Georgia Institute of Technology and Georgia Institute of Technology.

Savelsbergh, M. W. P., and Sol, M. 1995. The general pickup and delivery problem. *Transportation Science* 29(1):17–29.

Shoham, Y., and Tennenholtz, M. 1995. On social laws for artificial agent societies: Off-line design. 73(1–2):231–252.

Stentz, A. T. 1994. Optimal and efficient path planning for partially-known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)*, volume 4, 3310 – 3317.

ter Mors, A. W.; Zutt, J.; and Witteveen, C. 2007. Context-aware logistic routing and scheduling. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling (to appear)*.

Zutt, J., and Witteveen, C. 2006. Operational transport planning with incidents, experiments with traplas. In van Zuylen, H. J., ed., *Proceedings of the 9th TRAIL Congress (TRAIL'06)*.