Robustness in Context-Aware Route Planning

Adriaan ter Mors
Delft University of Technology
A.W.terMors@tudelft.nl

Abstract

Context-aware routing is the problem of finding the shortest route from a start location to a destination location while taking into account the planned movements of other agents. Existing approaches either integrate path planning and conflict resolution, or resolve conflicts after a tentative path has been found. For the integrated approach, optimal route planning was long believed to be computationally too expensive.

After analysis of the context-aware routing problem, we have been able to develop a much faster optimal algorithm, with which real-time, optimal planning may become a reality. The usability and robustness of any context-aware planning approach requires more research, however. Also, the robustness of a plan must be viewed in the context of a plan execution method, which may be able to solve small conflicts that occur at run-time.

Introduction

As in traditional shortest path planning, the objective in context-aware route planning is to find the shortest path (or route[1]) in time from a start location to a destination location. The difference is that a context-aware routing planning agent (usually representing a vehicle) takes into account its *context*, by which we mean the other agents active in the environment. Not in a general sense, like a network packet avoiding certain routes because of congestion, but in a very specific sense, by visiting locations only at times when they are not occupied by other agents (according to their plans).

Context-aware routing planning is relevant in domains where multiple vehicles are active on an infrastructure consisting of limited-capacity resources. Example domains include aircraft taxiing at airports, routing of Automated Guided Vehicles (AGVs) in automated manufacturing, and transport of containers by AGVs at ports. The conflicts that must be prevented by a context-aware routing algorithm are *head-on* conflicts and *catching-up* conflicts.

[1]To avoid confusion, we will use the term *path* when we describe a sequence of locations/edges/resources to be visited, and the term *route* when in addition to the sequences of locations visited, we maintain the times at which we intend to visit them.

Related work

In the literature on context-aware route planning, we can distinguish two main approaches: the first is separate conflict resolution from path planning, the second is to integrate the two. In the first approach, the first phase consists of path planning in which the actions of other agents are not taken into account. Typically, for the first phase the shortest path — or the k shortest paths — are found using a traditional path planning algorithm. In a second phase, conflicts are resolved, typically by speeding up or slowing down agents. The second approach is to take decisions on both which locations to visit, and when to visit them, to be able to avoid locations at times when they are occupied. In the literature, many algorithms of the second type are based on classical path planning algorithms, like A*, or Dijkstra's algorithm, extended to take the dimension of time into account. The following brief overview of the context-aware path planning literature was taken from our contribution to the main conference (ter Mors, Zutt, & Witteveen 2007):

One of the first papers on context-aware routing is from Broadbent et al. (Broadbent *et al.* 1985); they employ a simple shortest path algorithm to find a set of initial routes. In case of catching-up conflicts, some vehicles are slowed down; for head-on conflicts, an alternative route is found that does not make use of the road at which the conflict occurred. Broadbent's algorithm can be used both on unidirectional and bidirectional infrastructures, but in the latter case it need not find the optimal solution. The approach proposed by Hatzack and Nebel (Hatzack & Nebel 2001) also can be considered as a two-phase approach to this problem. Compared with earlier approaches, they use a more refined model of the common infrastructure by considering parts of the infrastructure (such as lane segments and intersections) as resources having a limited capacity. Once the individual routes have been chosen, conflict resolution can be modeled as a job-shop scheduling problem with blocking to ensure that the constraints imposed by the resources are not violated.

The algorithm proposed by Huang et al. (Huang, Palekar, & Kapoor 1993) is an integrated algorithm. It finds a path through the (graph of) free time windows on the resources, rather than directly through the graph of resources. Huang's algorithm is optimal both for unidirectional and

bidirectional networks, but it assumes unit capacity for all resources. Fujii et al. (Fujii, Sandoh, & Hozaki 1989) combine the search through free time windows with a heuristic that calculates the shortest path from the current resource to the destination resource, assuming no other traffic. The solution method proposed should result in an optimal, polynomial-time algorithm, but their description of the algorithm contains a number of ambiguities. Additionally, the authors do not provide any complexity analysis of the algorithm. The work of Kim and Tanchoco (Kim & Tanchoco 1991) is similar to the work of Fujii et al., but their treatment of the problem and the analysis of their algorithm is more comprehensive. Kim and Tanchoco's algorithm finds the (individually) optimal solution for both uni- and bidirectional networks, and they give an $O(n^4v^2)$ time complexity for their algorithm, where n is the number of vehicles in the system, and v is the number of resources in the infrastructure network. Due to this relatively high run-time complexity (especially given the limited computational resources of an early 90s PC), Taghaboni-Dutta and Tanchoco (Taghaboni-Dutta & Tanchoco 1995) developed an approximation algorithm that decides at every intersection to which resource to go next, based on the estimated traffic density of the resources from the current intersection to the destination. The authors show a small loss of plan quality, but they claim that the algorithm consumes significantly fewer computational resources; however, they do not quantify the run-time complexity of the approximation algorithm, nor do they present any CPU time comparisons.

We can conclude from existing research that many researcher consider optimal 'integrated' routing for single vehicles too computationally expensive for practical use. However, few papers provide comprehensive analysis of the algorithms presented within, and no benchmark problems for context-aware routing exist that allow the comparison of different algorithms. In our work we show that efficient algorithms can be found having a better worst-case performance than the complexity bound of $O(n^4v^2)$ found by Kim and Tanchoco.

Robustness and plan execution

With any planning approach to problem solving, we have to be aware that changes in the environment can invalidate the plans made. In context-aware path planning, this problem is especially pronounced for two reasons: i) a small delay in the execution of one plan may lead to a global deadlock; ii) the 'environment' consists of autonomous agents, each of which may display behaviour that is unexpected for the other agents. Hence, for context-aware plans to be usable, they must be robust.

The relation between plan robustness and plan execution is an interesting one. Suppose that a vehicle enters a road a few seconds early. This might result in a catching-up conflict, as it might follow its predecessor too closely. If the agent is unable to cope with this situation and it clobbers into the leading vehicle, then it seems the respective plans of the agents were not robust. If, on the other hand, each vehicle is equipped with sensors which with to measure the distance to a leading vehicle, then the early agent might slow down a bit, and both agents might arrive at their destination a few seconds early or late.

From the above example we can conclude that robustness is a *combination* of planning algorithm and execution method. If we take this idea one step further, then we can also consider the robustness if the planning algorithm does not solve *all* conflicts in advance. For example, in Kim and Tanchoco's algorithm (Kim & Tanchoco 1991), a lot of computation time is required to ensure that no catching-up conflicts occur. However, in case vehicle speeds are comparable, then it is not inconceivable that collision-avoidance at execution-time is sufficient to prevent such conflicts without greatly disturbing the plans of the agents. Whether this is indeed the case, or whether a small disturbance has a major cascading effect, should be further researched.

We can strive to improve robustness by improving the planning algorithms used, but also by enhancing the execution method. To improve robustness in planning, we might try (i) introducing slack in the plans of the agents, (ii) taking network congestion into account when choosing a route, or (iii) limiting the number of bidirectional resources in the network. To improve robustness in plan execution, a look-ahead function might be valuable.

Operational Setting

Airport taxi route planning is our current application domain. An important feature of this domain is that, on the taxiways, it is the pilot controlling the aircraft, and not the *auto*pilot. Naturally, this increases the chance of differences occurring between a plan and its execution, making robustness of route planning even more important.

Within the airport domain, we also consider scenarios of wintry weather conditions. From a route planning point of view, weather conditions such as snowfall are interesting for a number of reasons. First of all, taxiways and runways can become unavailable for some time while they are cleared of snow, requiring agents to reconsider their planned routes. Second, to ensure safe take-off, an aircraft has to be *de-iced* shortly before take-off. De-icing usually takes place at a remote facility called a *de-icing station*, which the aircraft must taxi to before it can go to the runway.

Contributions

In our work, we attempt to further the research field of context-aware routing, and investigate its applicability in realistic application domains. More specifically, we will make the following contributions:

Analysis of routing problem and algorithms: Analysis of the context-aware routing problem has demonstrated that an $O(nv\log(nv)+nv^2)$ algorithm exists for optimal, single-vehicle routing (where v equals the number of vehicles, and n the number of resources in the infrastructure). This improvement over the previously cited bound of $O(n^4v^2)$ means that real-time application of optimal routing algorithms are within reach for many realistic domains.

Multi-stage routing: A useful extension to routing from a start location to a destination location is to find the optimal route that visits (at least) a *fixed sequence* of locations. Although the multi-stage routing problem is computationally more expensive than 'single-stage' routing, it can still be solved optimally in polynomial time.

Robustness and applicability of routing algorithms: Unexpected changes in the environment and in the behaviour of other agents can invalidate an agent's plan. An empirical study should determine how sensitive context-aware plans are to changes, on which factors this sensitivity depends, and what the role is of plan execution techniques on the usability of plans.

Concluding Remarks

In the above text, we betrayed the origins of our research by using the word 'agent' where others may have used 'vehicle' or 'AGV' instead. Originally, context-aware routing was mostly studied by researchers with a background in manufacturing and engineering. Our background is in the field of multi-agent planning and coordination, in which the *autonomy* of individual agents is emphasized. This means that agents generally place their own interests above the greater good. For example, in our current application domain of airport taxi route planning, each airline will be concerned only with the timely departure of its own aircraft, and not care if flights of other airlines are delayed, although each airline will still heed the instructions of the Air Traffic Control tower.

Current context-aware routing algorithms, including our own (ter Mors, Zutt, & Witteveen 2007), simply assume that agents are willing to respect the reservations of other agents at resources they also need to use. In our current implementation, for example, the agents plan sequentially, each agent planning around the reservations that have been made by previous agents. To preserve the autonomy of the agents, they should be allowed to negotiate over the use of resources, or at the least over who will be first in line to plan its route. Hence, next to the development of better routing algorithms and a study of robustness proposed above, multi-agent negotiation in routing promises to be an exciting area of future work.

Acknowledgments

This research is supported by NWO (Netherlands Organization for Scientific Research), Grant No. CSI4006.

References

Broadbent, A. J.; Besant, C. B.; Premi, S. K.; and Walker, S. P. 1985. Free ranging AGV systems: promises, problems, and pathways. In *2nd International Conference on Automated Materials Handling*, 221–237.

Fujii, S.; Sandoh, H.; and Hozaki, R. 1989. A routing control method of automated guided vehicles by the shortest path with time-windows. In *Production Research: Approaching the 21st Century*, 489–495.

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

Huang, J.; Palekar, U.; and Kapoor, S. 1993. A labelling algorithm for the navigation of automated guides vehicles. *Journal of Engineering for Industry*.

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

Taghaboni-Dutta, F., and Tanchoco, J. 1995. Comparison of dynamic routing techniques for automated guided vehicle system. *International Journal of Production Research* 33(10):2653–2669.

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)*.