Knowledge Representation for Intelligent and Error-Prone Execution of Robust Granular Plans. A Conceptual Study*

Sebastian Ernst and **Antoni Ligęza**
Institute of Automatics, AGH University of Science and Technology, Kraków, Poland
{ernst,ligeza}@agh.edu.pl

Abstract

Techniques currently used for vehicle route planning suffer from drawbacks, especially significant when execution of the plan is interrupted, as the algorithms tend to return to the original solution as quickly as possible, instead of trying to "understand" the situation and react accordingly. This article describes a new approach aimed at solving this problem.

Introduction and motivation

Vehicle route planning is a popular application of AI automated planning methods. In numerous applications it is supported with GPS navigation. Based on a generalized shortest-path approach it uses a directed graph as the search domain and edge weights set to match the required optimality criteria. Moreover, various additional constraints and heuristic information can be explored. (Nau, Ghallab, and Traverso 2004)

As the plan is executed, various unexpected events can occur, resulting in the need for re-planning. If all goes well, the driver is presented with updated instructions, while the new plan is of acceptable quality. First, the processing capabilities of the device are limited, so the recalculation can take a considerable amount of time, leaving the driver without any feasible directions. This can become especially apparent in urban traffic conditions, where the street grid is usually so dense and complicated that the driver reaches the next decision node *before* the new plan is ready and the entire process has to begin again.

Second, problems with imprecise determination of the driver's position may occur. In really dense street networks (including road crossings), lane-precision position is very necessary, but may be unavailable.

Third, the planner does not "understand" the situation, and instead of trying to avoid a route segment which could not have been used for some reason, it tries to return to the previous route as quickly as possible, often falling into a loop.

One such issue is the case of a blocked highway. In such case, the vehicle could be stuck in a traffic jam for a very long period of time, as the exits are quite far apart from each other. This particular example shows that *optimal* route planning is something which may stand in opposition to *robust* route planing.

The example above describes a concept introduced in some of our previous papers, called *robust planning*, or *robust plan determination* (Ernst and Ligęza 2007). A route leading through the highway is probably the quickest one, but it is less *robust* than a route using lower-grade roads, which provides more switching possibilities in case of emergency or other difficulty. *Route robustness* is therefore a measure against the risk that the solution may not be executed according to the a priori plan. The main idea behind the concept of a *robust plan* is that such a plan should consist of numerous alternative plans, represented in a concise way, and enable switching from the plan currently being executed to a new one as often as may become necessary. The degree of robustness is a qualitative factor referring to numerous execution variants.

When executing the plan, the user could be presented with alternative solutions in locations prone to, for instance, traffic jams. Especially in urban conditions, the alternative solutions are nearly as optimal as the "best" solution. Seeing the alternatives could let the driver react to observed conditions, which could not have been known when the original plan was being calculated. Knowledge compilation, based on appropriate structural representation of the route network, as well as *a priori* search, constitute the foundation for robust plan generation. Intuitively, at any *switch point*, the driver is made aware of possible alternatives and their characteristics.

This approach could enable the system, aware of the alternatives, to "switch" to the appropriate suboptimal (or, as we have established, nearly-optimal) solution and carry on providing appropriate instructions. Moreover, even if the driver is not provided with the alternatives (as is the case with current sat-nav systems), the planner, equipped with multi-variant data, could *recognize* the driver's intentions and avoid the aforementioned loop problem.

Essentially, the concept is to transform the *decision-making* system — a category represented by current satellite navigation solutions — to a *decision support* system, utilizing the developments of artificial intelligence to aid human intelligence.

This short version of the paper only highlights the most important ideas. The full version is available on demand.

*Work supported by MNiSW research grant N516 228635.

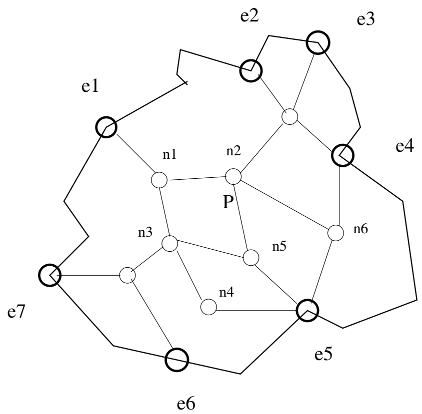

Figure 1: An example granule P with its entry/exit points ($P.e1, P.e2, \ldots, P.e7$) and the inner routes available.

Robust planning

If there are multiple paths between the start node and the goal node, they will likely intersect in several points. These intersections provide the possibility to switch between plans during execution. Therefore, we shall refer to the nodes where the paths intersect as *switch nodes*. If the initial plan can no longer be followed, the driver can switch to another solution.

As long as the plans are represented in an optimal and efficient way, the switch not only is instantaneous, but, more importantly, the information regarding the cost difference between various solutions at a switch node is readily available thanks to knowledge compilation.

Since the search domain can be rather complex and the devices only have limited processing power and memory capacity, it seems reasonable to minimize the amount of calculations needed to complete the planning process. This is especially true after the initial plan has been determined, i.e. for re-planning during plan execution. (Ernst and Ligęza 2008)

Therefore, the concept is to *compile* the basic (graph-form) *knowledge* into a more easily-processable form. The representation suggested here follows two main directions. First, it is made *granular* i.e. different levels of abstraction are introduced. Second, it is related to rule-based systems, as rule-based knowledge allows for quick processing and is reasonable in terms of size.

The principal concept is that the graph modeling the search domain is split into separate *granules* covering it. A *granule* is a knowledge representation component satisfying the following postulates:

- it covers a well-defined area, surrounded by a kind of boundary (e.g. a river, a wall, a field),
- it possesses a limited number of distinguished entry points which can be used to enter or leave the area,
- the graph of roads inside the granule is a connected one, there is at least one route from any entry point to any exit point,
- the graph of internal routes is relatively regular and dense.

Table 1: Schematic representation of an XTT decision table associated with node $n2$.

plan	node	next_node	min_cost	max_cost
p_{e1-e5}	n_2	n_5	37	49
p_{e1-e5}	n_2	n_6	39	53

An intuitive example of a general-level granule is a small city with several entry/exit roads. On a more detailed level, a granule can be a city district, surrounded by a river, a wall, a railroad or a highway without exits. A perfect example of a granule is any island connected to the main land with a limited number of bridges.

For intuition, a robust hierarchical plan is composed of a series of such granules to be traversed. This model is especially adequate for urban traffic planning within towns the districts of which are naturally separated, as noted above. When entering a, granule the driver navigates from an entry point to a desired exit point. Inside any granule, the set of routes is assumed to be relatively dense and regular. An example set of routes is presented in Figure 1.

Consider a case of navigating from the entry point $e1$ to the exit point $e5$. There are several possible linear plans for that task. We exclude plans passing through any entry/exit node different from $e1$ and $e5$; for such plans, switching to an alternative plan at the higher level may be reasonable.

After entering a granule and following a plan p, the driver attempts to arrive at $e5$. At any point, the driver must be presented with available alternatives, ordered according to preference. An example XTT decision table used for node $n2$ is shown in Table 1.

Conclusions

The innovative approach to route planning presented in the article requires efficient means of knowledge representation and compilation. There are still many open issues concerning representation (e.g. with respect to structuring of the initial graph), inference (search techniques), storage and presentation of the operational knowledge, etc. However, in complex, uncertain, overloaded environments, the proposed approach seems to offer interesting advantages over conventional planning techniques and tools.

References

Ernst, S., and Ligęza, A. 2007. Adaptive granular planning for robust plan generation under uncertain traffic conditions. In *Proceedings of the 16th international conference on Systems Science*, volume 2, 388–396.

Ernst, S., and Ligęza, A. 2008. A rule-based approach to robust granular planning. In *3rd International Symposium Advances in Artificial Intelligence and Applications (Proceedings of the International Multiconference on Computer Science and Information Technology)*, volume 3, 105–111.

Nau, D.; Ghallab, M.; and Traverso, P. 2004. *Automated Planning: Theory & Practice*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.