# Routing Driverless Transport Vehicles in Car Assembly with Answer Set Programming – Extended Abstract *

M. Gebser[1], P. Obermeier[1], T. Schaub[1 2], M. Ratsch-Heitmann[3], and M. Runge[3]

[1] University of Potsdam, Germany
[2] INRIA Rennes, France
[3] Mercedes-Benz Ludwigsfelde GmbH, Germany

**Abstract.** Automated storage and retrieval systems are principal components of modern production and warehouse facilities. While low-level control systems take care of navigating automated guided vehicles along programmed routes and avoid collisions even under unforeseen circumstances, in the common case of multiple vehicles sharing the same operation area, the problem remains how to set up routes such that a collection of transport tasks is accomplished most effectively. We address this prevalent problem in the context of car assembly at Mercedes-Benz Ludwigsfelde GmbH, where routes for automated guided vehicles used in the production process have traditionally been hand-coded by human engineers. Unlike this, we propose a declarative approach based on Answer Set Programming to optimize the routes taken by automated guided vehicles for accomplishing transport tasks.

## 1 Application Domain

Automated guided vehicles play a key role in modern industries, let it be in warehouses, mines, or as in our case production facilities. Most of the time, however, they are programmed by human engineers to execute specific transport tasks. The lack of elaboration tolerance does not only lead to high expenditures, but the resulting rigid control also rules out any flexible fleet management. We thus address the challenge of devising a new control system for automated guided vehicles supplying the assembly lines at the car factory of Mercedes-Benz Ludwigsfelde GmbH, which is expected to be flexible enough to adapt to malfunctions and emerging requirements, and whose quality of operation can be measured and optimized relative to given objectives. To this end, we propose to perform task assignment and vehicle routing by means of Answer Set Programming (ASP; [2]), given its declarative and elaboration tolerant approach to combinatorial multi-objective problem solving.

To illustrate our specific application setting, consider the retouched layout of an assembly hall at the production plant of Mercedes-Benz Ludwigsfelde GmbH in Figure 1. The overall goal is to guarantee that all car components are at their designated place next to the assembly line when they are due for installation. For instance, a vehicle may first halt at some storage location to load production material, then move on to an assembly station in need of the material, and from there cart off leftover material to a recycling facility. The sketched transport task thus involves stopovers at three distinct locations, between which the vehicle must pick a route without getting blocked by others, where dedicated parking spaces are included in the layout to let vehicles make room.

* The full version [1] of this extended abstract has been presented at ICLP 2018.

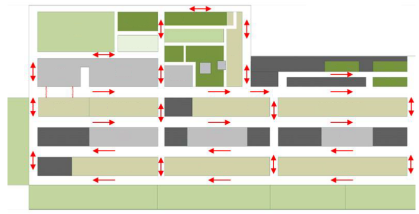

**Fig. 1.** Real-world factory layout with transport corridors and directions indicated by arrows

## 2 Implementation and Evaluation

We devised an ASP encoding (available at `http://www.cs.uni-potsdam.de/wv/projects/daimler/resources-iclp18.tar.xz`) for scheduling the accomplishment of transport tasks in the context of car production, involving four logical parts: task assignment and ordering, task completion, vehicle routing, and objective criteria. The latter include the makespan of a solution, the sum of route lengths over all vehicles, and the crossings as well as overlaps of vehicle routes in this order of significance.

A preliminary field study on 18 use cases provided by production engineers at Mercedes-Benz Ludwigsfelde GmbH compared the open-source control system software *openTCS* (see `http://www.opentcs.org`) to our ASP approach to vehicle routing utilizing *clingo* (version 5.2.2; `https://potassco.org`). While *openTCS* furnishes a generic platform for automated guided vehicle control, it does not ship a solver for hard combinatorial tasks such as the multi-objective optimization of routes off the shelf, which *clingo* is geared for and where it is envisaged to come into play as a plugin in the future.

Due to its greedy approach to assign a task and lock locations along the shortest route of a vehicle one by one, *openTCS* fails to come up with a feasible solution in 8 of the 18 use cases. Unlike that, *clingo* succeeds in obtaining provably optimal routes in all of them, taking up to ten seconds of runtime for 17 small use cases that focus on particular rerouting scenarios and about 5 hours for one use case emulating a full production cycle of roughly 20 minutes in real time. Critical factors for computational performance include, for one, the representation size, where the 20-minutes scheduling horizon leads to about one million solver-internal constraints, and for another, the high combinatorics indicated by more than 20 million search conflicts. For potentially reducing such factors and scaling up further, numerical variables as featured by constraint/integer programming and difference logic extensions [3] may enable a more compact representation of scheduling horizons. While our ASP approach aims at global optimization, decomposition methods like the decoupling of task assignment and routing investigated in [4] could also help to reduce combinatorics.

## References


1. Gebser, M., Obermeier, P., Schaub, T., Ratsch-Heitmann, M., Runge, M.: Routing driverless transport vehicles in car assembly with answer set programming. TPLP **18**(3-4) (2018) 520–534
2. Lifschitz, V.: Answer set planning. Proceedings of the International Conference on Logic Programming (ICLP 1999), MIT Press (1999) 23–37
3. Kaminski, R., Schaub, T., Wanko, P.: A tutorial on hybrid answer set solving with clingo. Proceedings of the International Reasoning Web Summer School (RW 2017), Springer (2017) 167–203
4. Gebser, M., Obermeier, P., Otto, T., Schaub, T., Sabuncu, O., Nguyen, V., Son, T.: Experimenting with robotic intra-logistics domains. TPLP **18**(3-4) (2018) 502–519