# Scalable Robotic Intra-Logistics with Answer Set Programming

**Philipp Obermeier**
Institute of Computer Science, University of Potsdam
Germany
phil@cs.uni-potsdam.de

**Abstract**

Over time, Answer Set Programming (ASP) has gained traction as a versatile logic programming semantics with performant processing systems, used by a growing number of significant applications in academia and industry. However, this development is threatened by a lack of commonly accepted design patterns and techniques for ASP to address dynamic application on a real-world scale. To this end, we identified *robotic intra-logistics* as representative scenario, a major domain of interest in the context of the fourth industrial revolution. For this setting, we aim to provide a scalable and efficient ASP-based solutions by (1) stipulating a standardized test and benchmark framework; (2) leveraging existing ASP techniques through new design patterns; and (3) extending ASP with new functionalities. In this paper we will expand on the subject matter as well as detail our current progress and future plans.

**2012 ACM Subject Classification** Computing methodologies → Logic programming and answer set programming

**Keywords and phrases** Answer Set Programming, Logistics, Planning

**Digital Object Identifier** 10.4230/OASIcs.ICLP.2018.24

**Funding** This research is partially supported by DFG (550/9).

## 1 Introduction

Answer Set Programming (ASP; [4]) has come a long way, starting as a semantics for logic programming, over having increasingly performant systems, to a growing number of significant applications in academia and industry. In contrast to other solver paragdims, ASP offers an unprecedented degree of versatility and brevity, which is best put in perspective by solving multi-faceted problems. However, this development is threatened by a lack of commonly accepted design patterns and techniques for ASP to address dynamic application on a real-world scale. In addition, many industrial applications require the integration of multiple types of knowledge and forms of reasoning, a feature commonly neglected by existing approaches. As a first step to overcome these problems, we have identified *robotic intra-logistics* as representative scenario for our investigation. This domain is a major subject of interest in the context of the fourth industrial revolution, as witnessed by Amazon's Kiva, GreyOrange's Butler, and Swisslog's CarryPick systems.[1] All of them aim at automatizing warehouse operations (illustrated by Figure 1) by using robot vehicles that drive underneath mobile shelves and deliver them to picking stations. From there, workers pick and place the requested items in shipping boxes. For this setting, we aim to provide scalable and efficient

[1] `www.amazonrobotics.com`, `www.greyorange.com/products/butler`, `www.swisslog.com/carrypick`


Technical Communications of the 34th International Conference on Logic Programming (ICLP 2018).
Editors: Alessandro Dal Palu', Paul Tarau, Neda Saeedloei, and Paul Fodor; Article No. 24; pp. 24:1–24:5
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

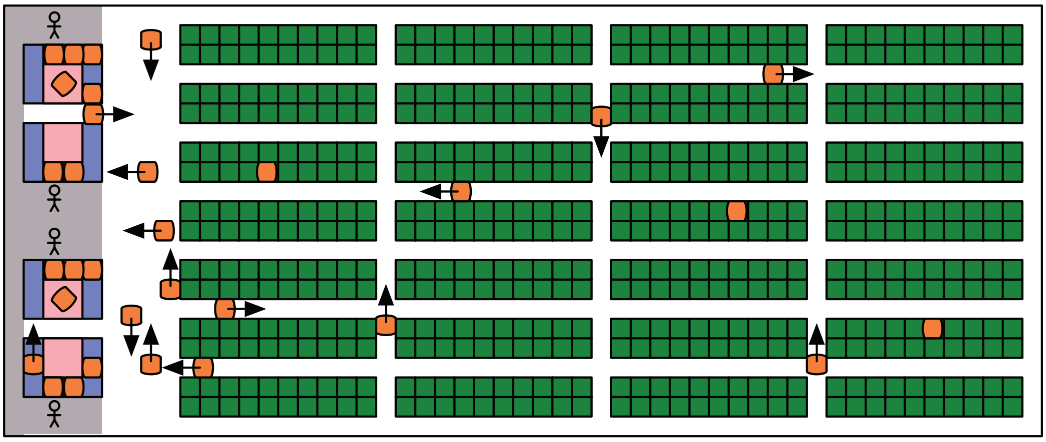

**Figure 1** Layout of an Autonomous Warehouse System [Wurman *et al.*, 2008].

ASP-based solutions by (1) stipulating a standardized test and benchmark framework; (2) leveraging existing ASP techniques through new design patterns; and (3) extending ASP with new functionalities.

## 2 Related Work

What distinguishes robotic intra-logistics from other combinatorial problems is its multidimensional nature that necessitates the integration of a great many of aspects, most notably path finding and order fulfillment.

At the core of many path finding problems lies the search for a route for an agent from an initial to a final location. The *multi-agent path finding* (MAPF) problem asks for a collision-free route for each agent such that the total makespan is minimal. MAPF is related to many real-world applications but already computationally intractable [8]. While in MAPF each agent is assigned a unique destination, its *anonymous* variant requires no assignment of agents to destinations [9]. The problem domains of *asprilo* are obviously related to multi-agent path finding. More specifically, the *asprilo* domain **M** corresponds to anonymous MAPF. Each order is uniquely associated with a destination shelf and there is no pre-assignment of a robot to an order. Robots can freely reach any destination shelf. Clearly, **M** is easily extended to cover non-anonymous MAPF by relating robots and orders.

*Task assignment and path finding* (TAPF; [5]) is a generalization of MAPF. TAPF groups agents into teams. Although teams are (non-anonymously) pre-assigned to groups of destinations, any robot in the team can be (anonymously) selected for a destination in the assigned group. G-TAPF [7] is a generalization of TAPF aiming at more realistic settings by allowing the number of tasks to be greater than the number of agents and considering deadlines, orderings, and checkpoints. That is, deadlines are associated with order lines, orders are completed in a pre-defined ordering and all lines in a single order need to be fulfilled before any line of another order is completed, and while fulfilling an order, a robot is required to go through a sequence of locations, called checkpoints. Regarding previous uses of ASP, [1] address several aspects of multi-agent path finding problems.

[6] address an online version of path finding, where not all destination tasks[2] are given initially but may arrive over time.

[2] Actually, this work also uses pick-up and delivery tasks to simulate a warehouse system.

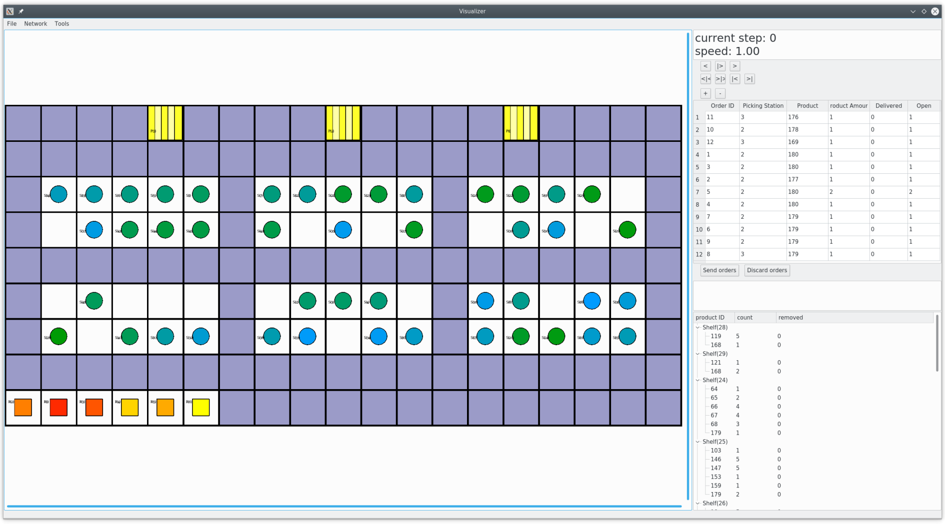

**Figure 2** Exemplary *asprilo* screenshot: The main window gives a warehouse layout; no plan is loaded. Picking stations are represented by striped yellow squares, shelves by solid circles, and robots by solid squares. Highways are brought out in purple. The side windows provide controls for plan animation, and give details about the current orders and the warehouse inventory.

## 3 Current Research Progress

So far, we introduced a scalable approach for a generalized variant of TAPF, formally laid out the problem domain, and conducted a real-life case study for car assembly:

***asprilo*: Robotic Intra-Logistics Benchmark Suite [2].** We introduce the *asprilo*[3] framework to facilitate experimental studies of approaches addressing complex dynamic applications. For this purpose, we have chosen the domain of robotic intra-logistics. This domain is not only highly relevant in the context of today's fourth industrial revolution but it moreover combines a multitude of challenging issues within a single uniform framework. This includes multi-agent planning, reasoning about action, change, resources, strategies, etc. In return, *asprilo* allows users to study alternative solutions as regards effectiveness and scalability. Although *asprilo* relies on Answer Set Programming and Python, it is readily usable by any system complying with its fact-oriented interface format. This makes it attractive for benchmarking and teaching well beyond logic programming. More precisely, *asprilo* consists of a versatile benchmark generator, solution checker and visualizer (see Figure 2) as well as a bunch of reference encodings featuring various ASP techniques. Importantly, the visualizer's animation capabilities are indispensable for complex scenarios like intra-logistics in order to inspect valid as well as invalid solution candidates. Also, it allows for graphically editing benchmark layouts that can be used as a basis for generating benchmark suites. The *asprilo* framework is freely available at `https://potassco.org/asprilo`.

**Generalized Target Assignment and Path Finding [7].** Both MAPF and TAPF models suffer from their limiting assumption that the number of agents and targets are equal. We propose the *Generalized TAPF* (*G-TAPF*) formulation that allows for (1) unequal

[3] *asprilo* stands for *Answer Set Programming for robotic intra-logistics*.

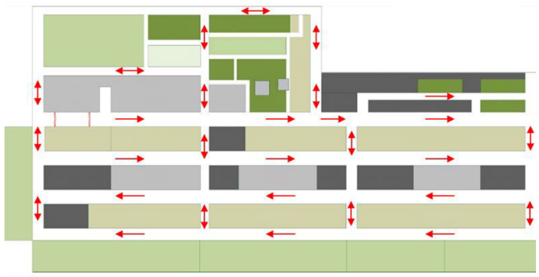

**Figure 3** Real-world factory layout with transport corridors and directions indicated by arrows.

number of agents and tasks; (2) tasks to have deadlines by which they must be completed; (3) ordering of groups of tasks to be completed; and (4) tasks that are composed of a sequence of checkpoints that must be visited in a specific order. As different G-TAPF variants may be applicable in different domains, we model them using ASP, which allows one to easily customize the desired variant by choosing appropriate combinations of rules to enforce. Our experimental results show that the popular CBM (conflict-based min-flow) algorithm is better in simple TAPF problems with few conflicts, but worse in difficult problems with more conflicts. We also show that ASP technologies can easily exploit domain-specific information to improve its scalability and efficiency. The contributions in this paper thus make a notable jump towards deploying MAPF and TAPF algorithms in practical applications.

**Routing Driverless Transport Vehicles in Car Assembly [3].** Automated storage and retrieval systems are principal components of modern production and warehouse facilities. In particular, automated guided vehicles nowadays substitute human-operated pallet trucks in transporting production materials between storage locations and assembly stations. While low-level control systems take care of navigating such driverless 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 (see Figure 3) at Mercedes-Benz Ludwigsfelde GmbH, a large-scale producer of commercial vehicles, where routes for automated guided vehicles used in the production process have traditionally been hand-coded by human engineers. Such ad-hoc methods may suffice as long as a running production process remains in place, while any change in the factory layout or production targets necessitates tedious manual reconfiguration, not to mention the missing portability between different production plants. 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. The advantages include a transparent and executable problem formalization, provable optimality of routes relative to objective criteria, as well as elaboration tolerance towards particular factory layouts and production targets. Moreover, we demonstrate that our approach is efficient enough to deal with the transport tasks evolving in realistic production processes at the car factory of Mercedes-Benz Ludwigsfelde GmbH.

## 4 Open Issues and Expected Achievements

To sum up, we expect the following major achievements through our research:

1. A standardized framework for experimental studies of dynamic systems, specifically in the intra-logistics domains
2. Novel ASP design patterns and extensions for solving various problems in dynamic systems on an industrial scale

### References


1 E. Erdem, D. Kisa, U. Öztok, and P. Schüller. A General Formal Framework for Pathfinding Problems with Multiple Agents. In M. desJardins and M. Littman, editors, *Proceedings of the Twenty-Seventh National Conference on Artificial Intelligence (AAAI'13)*, pages 290–296. AAAI Press, 2013. URL: `http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6293`.

2 M. Gebser, P. Obermeier, T. Otto, T. Schaub, O. Sabuncu, V. Nguyen, and T. Son. Experimenting with robotic intra-logistics domains. *Theory and Practice of Logic Programming*, 2018. To appear. URL: `http://arxiv.org/abs/1804.10247`.

3 M. Gebser, P. Obermeier, M. Rasch-Heitmann, and T. Schaub. Routing Driverless Transport Vehicles in Car Assembly with Answer Set Programming. *Theory and Practice of Logic Programming*, 2018. To appear. URL: `http://arxiv.org/abs/1804.10437`.

4 V. Lifschitz. Answer set planning. In D. de Schreye, editor, *Proceedings of the International Conference on Logic Programming (ICLP'99)*, pages 23–37. MIT Press, 1999.

5 H. Ma and S. Koenig. Optimal Target Assignment and Path Finding for Teams of Agents. In C. Jonker, S. Marsella, J. Thangarajah, and K. Tuyls, editors, *Proceedings of the Fifteenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'16)*, pages 1144–1152. ACM Press, 2016.

6 H. Ma, J. Li, T. Kumar, and S. Koenig. Lifelong Multi-Agent Path Finding for Online Pickup and Delivery Tasks. In *Proceedings of the Sixteenth Conference on Autonomous Agents and MultiAgent Systems (AAMAS'17)*, pages 837–845. ACM Press, 2017.

7 V. Nguyen, P. Obermeier, T. Son, T. Schaub, and W. Yeoh. Generalized Target Assignment and Path Finding Using Answer Set Programming. In C. Sierra, editor, *Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence (IJCAI'17)*, pages 1216–1223. IJCAI/AAAI Press, 2017.

8 P. Surynek. An Optimization Variant of Multi-Robot Path Planning Is Intractable. In M. Fox and D. Poole, editors, *Proceedings of the Twenty-fourth National Conference on Artificial Intelligence (AAAI'10)*, pages 1261–1263. AAAI Press, 2010. URL: `http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1768`.

9 J. Yu and S. LaValle. Multi-agent Path Planning and Network Flow. In E. Frazzoli, T. Lozano-Pérez, N. Roy, and D. Rus, editors, *Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics (WAFR'12)*, volume 86 of *Springer Tracts in Advanced Robotics*, pages 157–173. Springer-Verlag, 2012. `doi:10.1007/978-3-642-36279-8_10`.