ECAI 2025
I. Lynce et al. (Eds.)

doi:10.3233/FAIA251436

An Operator-Centric Trustable Decision-Making Tool for Planning Ground Logistic Operations of Beluga Aircraft

Rebecca Eifler[a,*], Nika Beriachvili[a], Arthur Bit-Monnot[a], Dillon Z. Chen[a], Jan Eisenhut[b], Jörg Hoffmann[b], Sylvie Thiébaux[a] and Florent Teichteil-Königsbuch[c]

[a]LAAS-CNRS, Toulouse, France
[b]Saarland University, Saarbrücken, Germany
[c]Airbus AI Research, Toulouse, France

Abstract. This paper presents the demonstrator developed in the TUPLES European Union research project for assisting human operators at Airbus to plan Beluga cargo ground logistic operations. The demonstrator features techniques providing robust, explainable, and safe decisions, which all contribute to making our decision-support system trusted by the operators. We have also worked on various planning methods to scale up to the size of the real industrial problem, including hybrid machine learning and symbolic algorithms. We demonstrate the software that was tested by Airbus operators during a user study in Finkenwerder's production site in May 2025.

1 Introduction

As AI decision-making technologies become sufficiently mature for solving real use cases, there is a growing need from users to make these technologies trustworthy. The TUPLES Horizon Europe project [1] especially developed novel methods and tools for four important aspects of decision-making trustworthiness: *scalability* to real-size and diverse problems, *robustness* to uncertainties impacting the execution of the decision plans, *explainability* and *safety* of the optimized decisions. Those technologies have been demonstrated and evaluated on different real use cases provided by various European industrial partners, from aircraft manufacturing to sport squad management via public energy and waste management. Among them, Airbus provided a logistic planning use case consisting in optimizing the on-site ground management of parts transported by Beluga aircraft up to their delivery to the factory.

Assembling complex structures such as commercial aircraft relies on a network of production plants located in different sites. Assembly parts of growing complexity and size are built and transferred from plants to plants until reaching a final plant where the final product is fully assembled. For a manufacturer like Airbus, parts transit via trucks, sea vessels and so-called Beluga aircraft (see Figure 1). They are mounted on jigs that hold frames—such as the one supporting the wings in Figure 1.a—which slide from the interior of the Beluga onto a movable rack, as shown in the picture. Jigs must return to their original site after the part they hold has been delivered to the delivery site. Once arrived at the delivery site, the jigs must be unloaded and placed on intermediate storage racks. They are then waiting in the racks to be sent afterwards to the production plant when the latter is ready to accept them for production. When a part is picked up from the rack system and sent to the plant, the jig that held that part must return to the rack system and wait to be returned to its original production site by some outgoing Beluga flight. Two kinds of decision-making problems are involved: first, optimizing the flow of jigs and parts across the sites in order to respect the production calendar; second, planning the ground operations of loading and unloading Beluga aircraft and managing the storage of the parts in the rack system. This paper focuses on the second problem, i.e. planning of Beluga's ground operations, which is a complex dynamic storage management problem specific to Airbus operations.

Planning for the logistic operations of loading, unloading and storing jigs in the rack system is especially complex because the jigs must be stored in the racks in such a way as to be delivered to the factory (for production) or to the Beluga (for returning empty jigs to their original site) in the right order. If the order in a given rack is different from the delivery order on factory or Beluga side, then a costly 30 minutes swap task is required, which consists in 3 actions: (1) picking up a part at either end of the rack, (2) storing it in another rack, and (3) delivering the part which is now at the end of the rack to the factory or the Beluga. As the rack system occupation increases, the risk of requiring swap tasks, or even of blocking the system, significantly increases. Replanning the system upon new Beluga flight arrival or change in the production schedule is currently manually done by experienced engineers, and can take up to 2 hours in complex situations. In the context of the TUPLES European Union research project [1], we developed a prototype of an explainable automated planning system, which helps the engineers plan the grounding logistic operations of the Beluga, with a particular focus on interactively explaining reasons for problem infeasibility and trade-offs between conflicting objectives. This planning system has been used for a user study in May 2025 at Airbus' Hamburg production site with real planning engineers of the rack system, using the software we demonstrate in this paper.

2 Planning problem

Aircraft parts are held on jigs which can slide and be stored on the racks. Each jig has a type, defined by its loaded and empty size, and by the specific aircraft parts (e.g. wings) it is designed to hold. As an example, a wing jig can be seen in Figure 1.a, which is holding two wings in staggered rows. The jig is sliding on a mobile rack inside the hangar as it is unloaded from (or loaded to) the Beluga. When exiting

* Corresponding Author. Email: rebecca.eifler@laas.fr.

(a) Unloading/Loading of a wing from/to a Beluga aircraft. The part stands on a jig which goes itself on a mobile rack to transfer the unloaded/loaded part to/from the outside rack system.

(b) Beluga hangar and outside rack system (top left corner of the picture) where aircraft jigs and their held parts are temporarily stored before being sent to the factory.

Figure 1. Transportation and storage of aircraft parts with Beluga aircraft. Pictures are under Airbus copyright.

the Beluga hangar, the jig is loaded onto a trailer, which transports it between the hangar and the fixed racks outside. The jig is then stored on the rack system, shown in the top-left corner of Figure 1.b. When the aircraft parts are sent to the production lines, they transit through craning hangars where cranes remove the parts from the jigs. The parts are then sent to production and the jigs return empty to the rack system. Empty jigs must then return to some outgoing Beluga flights in order to refill the set of jigs needed to hold future parts produced at their original production site. We only know the types of the jigs that must be returned to their original site and the number of jigs of each type that must be returned. The racks can contain several jigs in sequence, but only the jigs which are at the edges of the racks, either factory side or Beluga side, can be pulled out from the racks. This might require swapping jigs located at the rack edges to other racks in order to free the path to jigs which are strictly inside the racks (i.e. not at their edges).

When the Beluga lands on the production site, two high-level tasks must be performed:

- unloading the parts (held on jigs) from the Beluga and storing them in the rack system;
- unstoring empty jigs from the rack system and loading them into the Beluga.

Loading parts for a given flight can be done in parallel with unloading the preceding flight by using two mobile racks which can be operated in parallel inside the Beluga hangar. Between two Beluga flights, three high-level tasks must be considered, possibly interleaved:

- unstoring parts held on jigs from the rack system and sending them to the production lines via the craning hangars;
- sending back empty jigs from the craning hangars to the rack system;
- optionally swapping the jigs which are at the edges of the racks (either factory side or Beluga side) from one rack to another, in order to free the path to jigs which are strictly inside the racks.

A planning problem consists in deciding of the sequence of such actions for a given sequence of Beluga flights and a given order of parts to be sent to the factory. The goal of the problem is to deliver all the parts to the factory in the right order, and to load back empty jigs of the correct types onto Beluga aircraft in the right order. We can easily see that the problem is NP-hard by reduction from bin-packing, and therefore it is especially difficult to solve even, in practice, for small-size instances with less than 10 jigs and 5 racks. Therefore, we split it into sequential planning sub-problems consisting in handling one Beluga flight at a time. The Beluga flight of each subproblem must be unloaded and loaded, and the Beluga-side and factory-side actions to move the jigs and parts between the racks and the Beluga or (where possible) factory hangars must be also planned within the same time frame.

While we are focusing on finding a satisfiable plan, users are actually interested in exploring various solution plans and in comparing them according to different metrics: length of the plan; number of free racks in the final state; number of swap actions, where a swap is defined as the sequence of picking up a jig from a rack and putting it back on another rack. Apart from plan length, users prefer reasoning with threshold constraints (e.g. "fewer than 2 swaps") which we add to the problem's goal. As a consequence, finding a plan which satisfies the goal condition amounts to finding a plan which satisfies trade-off constraints between different metrics.

3 Demonstrated technologies

Our demonstrator software has been developed for the TUPLES project [1] which aims at implementing novel research methods and tools for trustworthy decision-support systems. As shown in Figure 3, the demonstrator features algorithms developed in TUPLES that contribute to 3 major areas in decision-making trustworthiness: scalability, explainability and safety. Please note that robustness has been also investigated for this use case, but probabilistic methods are currently not sufficiently scalable to be part of this demonstrator involving real users on real problems of our industrial partner. Figure 3 depicts the architecture of the demonstrator software which can be globally split in 4 user interaction phases when read from left to right:

1. split the x-flights problem, $x > 0$, into x problems of 1 flight each, and solve it sequentially (otherwise it would be too large to be solved at once);
2. either solve the problem manually or automatically with different AI planning and machine learning algorithms;
3. interactively explain the decisions to the user, focusing on infeasibility restoration and conflicting trade-offs exploration;
4. inspect the resulting plan and test its optimality gap, especially for manual plans and deep learning action policies.

We now describe the main trustworthiness methods developed in the TUPLES project that are part of the demonstrator.

Scalability. The problem being especially hard to solve, we have implemented different model-based and data-driven algorithms, ranging from a domain-specific heuristic algorithm (SSBP) to domain-independent hierarchical planning (Aries), via generalised policy learning (ASNets) and heuristic search guided by a learnt

Figure 2. Conflict resolution: cannot produce part 12 from plant pl0 while maintaining trailer FT1 and imposing no swap and 1 empty rack

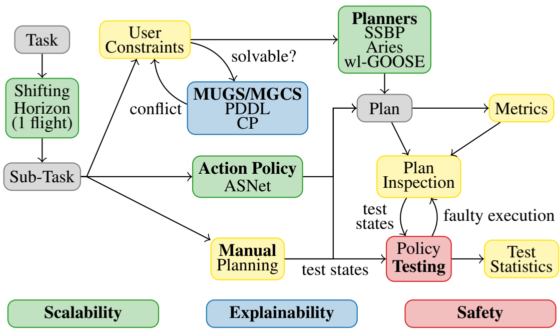

Figure 3. Architecture of the demonstrator software

heuristic (WL-GOOSE). In more detail, the Subgoal Sampling Beluga Planner (SSBP) is an efficient but incomplete domain-specific heuristic approach that performs a number of length-bounded trials aiming at reaching the goal from the initial state. Each trial repeatedly identifies a useful subgoal to achieve next in order to make progress towards the overall goal (e.g. delivering the next jig to the factory, getting a jig onto a trailer etc), and samples an action to be executed next among those likely to lead to achieving this subgoal. On the other hand, Aries models the problem as a scheduling problem with optional actions using the unified-planning library [9], and solves it with the Aries automated planner [2]. Internally, the planning problem is converted into a series of CSPs with increasing numbers of swaps, each of which is solved with a specialized hybrid CP-SAT solver. Moving to data-driven approaches, Action Schema Networks (ASNets) is a neural network architecture that exploits the relational structure of a PDDL domain – here the Beluga. It uses imitation learning to produce a generalised reactive policy that can quickly solve much larger instances from the domain than those it trained on [13, 14]. Finally, WL-GOOSE learns heuristics and state rankings to guide greedy best-first search [3]. WL-GOOSE applies statistical machine learning techniques to relational Weisfeiler-Leman features generated from a graph representation of the lifted planning problem. Similarly to ASNets, the learnt search guidance applies to much larger problems than trained on.

Explainability. Users need to trust the decisions provided by the algorithms in order to avoid blocking the system and consequently the production, which could cost millions of euros in delay penalties. They especially want to explore different trade-offs as threshold constraints on rack emptiness and number of swaps, and maintenance constraints on trailers and racks which prevent their use. If there is no plan satisfying those constraints, users would like to understand how to relax thresholds or to delay some production parts or rack maintenance in order to make the plan feasible (see Figure 2).

To provide insights into the dependencies between the different constraints we provide explanations based on minimal conflicts given by minimal unsolvable goal subsets [4]. The minimal conflicts lead explanations such as "It is not possible to use fewer than two swaps while keeping one rack empty and one trailer is in maintenance.". To compute the minimal conflicts we consider different approaches. QUICKXPLAIN [7] using the SSBP heuristic to test for solvability, provides a single minimal conflict quickly. MARCO [8] relying on Aries [2] as solvability check focuses on enumerating all conflicts.

Safety. Due to the complexity of our problem, none of the algorithms can solve it optimally in reasonable time. Therefore, given a policy π that solves the problem, we apply policy testing methods which can evaluate the optimality gap of π on a subset of states, i.e., provide lower bounds for how much π is suboptimal. Action policy testing as in [11, 5, 6] is organized as a two-step procedure. First, it involves the generation of a pool of test states t, which can be user-provided or obtained using random walks from the initial state of the problem. Second, a test oracle attempts to identify that π is suboptimal on these t, leveraging (and combining) different methods such as metamorphic testing [5] or the plan improvement tool Aras [10].

4 Conclusion

This paper presented an overview of the trustworthy planning system that was developed in the TUPLES European Union project for planning Beluga ground logistic planning operations at Airbus. The demonstrator, which was partly tested by operators in a user study in Airbus' Finkenwerder production site in May 2025, especially features hybrid machine learning and symbolic planning algorithms, explainable conflict resolution and policy testing methods developed in TUPLES. The users appreciated the ability of the tool to provide valuable and explainable decision advice in critical time-constrained situations, reducing their stress while keeping them in full control of the final solution. In the future, we plan to refine the problem representativeness and also to include probabilistic, explainable planning algorithms that we have investigated in TUPLES to handle probabilistic production demands [12]. However, they require significant additional engineering work on the demonstrator to handle and interactively simulate probabilistic events and disruptions.

Acknowledgements

This work has received funding from the European Union's Horizon Europe Research and Innovation program under the grant agreement TUPLES No 101070149. We thank the Airbus Beluga Logistic Planning team in Finkenwerder for having shared their use case and long-standing experience with us. Please note that the data used in the demonstrator are representative samples of the real operations.

References

[1] The TUPLES European Union Research Project: TrUstworthy Planning and scheduling with Learning and ExplanationS. https://tuples.ai/project/. Accessed in May 2025.

[2] A. Bit-Monnot. Experimenting with Lifted Plan-Space Planning as Scheduling: Aries in the 2023 IPC. In *Proc. 2023 International Planning Competition*, Prague, Czech Republic, July 2023. URL https://hal.science/hal-04174737.

[3] D. Z. Chen, F. Trevizan, and S. Thiébaux. Return to tradition: Learning reliable heuristics with classical machine learning. In *Proc. 34th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 68–76, 2024.

[4] R. Eifler, M. Cashmore, J. Hoffmann, D. Magazzeni, and M. Steinmetz. A new approach to plan-space explanation: Analyzing plan-property dependencies in oversubscription planning. In *Proc. 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9818–9826, 2020.

[5] J. Eisenhut, Á. Torralba, M. Christakis, and J. Hoffmann. Automatic metamorphic test oracles for action-policy testing. In *Proceedings of the 33rd International Conference on Automated Planning and Scheduling (ICAPS'23)*, pages 109–117. AAAI Press, 2023.

[6] J. Eisenhut, X. Schuler, D. Fišer, D. Höller, M. Christakis, and J. Hoffmann. New fuzzing biases for action policy testing. In *Proceedings of the 34th International Conference on Automated Planning and Scheduling (ICAPS'24)*, pages 162–167. AAAI Press, 2024.

[7] U. Junker. QUICKXPLAIN: preferred explanations and relaxations for over-constrained problems. In D. L. McGuinness and G. Ferguson, editors, *Proc. 19th AAAI Conference on Artificial Intelligence (AAAI)*, pages 167–172, 2004.

[8] M. H. Liffiton, A. Previti, A. Malik, and J. Marques-Silva. Fast, flexible MUS enumeration. *Constraints An Int. J.*, 21(2):223–250, 2016.

[9] A. Micheli, A. Bit-Monnot, G. Röger, E. Scala, A. Valentini, L. Framba, A. Rovetta, A. Trapasso, L. Bonassi, A. E. Gerevini, L. Iocchi, F. Ingrand, U. Köckemann, F. Patrizi, A. Saetti, I. Serina, and S. Stock. Unified planning: Modeling, manipulating and solving ai planning problems in python. *SoftwareX*, 29:102012, 2025.

[10] H. Nakhost and M. Müller. Action elimination and plan neighborhood graph search: Two algorithms for plan improvement. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS'10)*, pages 121–128. AAAI press, 2010.

[11] M. Steinmetz, D. Fišer, H. F. Enişer, P. Ferber, T. Gros, P. Heim, D. Höller, X. Schuler, V. Wüstholz, M. Christakis, and J. Hoffmann. Debugging a policy: Automatic action-policy testing in AI planning. In *Proceedings of the 32nd International Conference on Automated Planning and Scheduling (ICAPS'22)*, pages 353–361. AAAI Press, 2022.

[12] M. Steinmetz, S. Thiébaux, D. Höller, and F. Teichteil-Königsbuch. Explaining the Space of SSP Policies via Policy-Property Dependencies: Complexity, Algorithms, and Relation to Multi-Objective Planning. *Proceedings of the International Conference on Automated Planning and Scheduling*, pages 555–564, May 2024.

[13] S. Toyer, S. Thiébaux, F. Trevizan, and L. Xie. Asnets: Deep learning for generalised planning. *J. Artif. Intell. Res.*, 68:1–68, 2020.

[14] R. X. Wang and S. Thiébaux. Learning generalised policies for numeric planning. In *Proc. 34th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 633–642, 2024.