Optimizing Dynamic Airlift Operations: Winning Strategies in the AFRL Airlift Challenge

John Kolen
johnfkolen@gmail.com

Abstract

The Air Force Research Laboratory (AFRL) has sponsored the Airlift Challenge over the past two years, aimed at addressing the dynamic airlift problem. The dynamic nature of the challenge included the random disappearance of graph edges to simulate adverse weather conditions and the spontaneous appearance of cargo requiring delivery. This poster presents the systems that won both the 2023 and 2024 challenges. The initial approach focused on intelligent solutions for subtasks, or 'build-smart'. It soon became clear that the optimization of the scoring rate, points per second, was more important than single instance metric performance. In the subsequent competition, a 'build-fast' strategy was adopted due to this observation. This paper discusses the impact of iteration on algorithm selection for optimization problems and suggests considerations for structuring scoring processes in future competitions.

Introduction

The dynamic airlift cargo problem involves efficiently moving large amounts of cargo between airports while dealing with constraints such as fuel limits, airport capacity, and dynamic environmental changes that may cause delays or route unavailability. This problem is challenging due to the need for quick recalculations of routes to avoid delivery delays.

We begin by introducing the airlift competition and examining the operational environment, evaluation process, and scoring criteria. The discussion then shifts to the top strategies for this year and the previous one, which can be categorized as "build-smart" and "build-fast". Finally, we delve into some observations regarding competition scoring and make recommendations for future competitions.

Airlift Competition

In this section, we first examine the operating environment for the airlift competition. Then the evaluation process and scoring methods are examined. Additional details can be found in (Delanovic et al. 2023).

In 2022, a team at the Air Force Research Laboratory (AFRL) created an open-source simulation environment in Python to aid researchers in developing air cargo planning algorithms (Beckus et al. 2022). In this environment, each airplane (agent) can perform actions such as loading cargo, unloading cargo, traveling to another node, or taking no action. Agents receive observations at each time step, which include information about the airplane's status, route availability, location, cargo locations, and new cargo generation.

The environment's reward system penalizes late or missed deliveries and unnecessary movements between airports, encouraging agents to choose the most efficient routes. The evaluation of algorithms in this context involves a series of tests with increasing difficulty based on static parameters (number of agents, cargo per episode, and airports) and dynamic parameters (airport capacity, likelihood of route unavailability).

In the simulation, there are "hard deadlines" after which cargo is considered completely missed, and "soft deadlines" where no lateness penalty is applied if delivery is made by this time. The scoring system for each episode is based on missed deliveries, late deliveries, and total flight cost. The scoring considers the actual delivery time, the cost of the flight, and the number of flights over a route. Missed deliveries, lateness beyond the soft deadline, and flight costs are all factored into the score.

To encourage uniformity across scenarios, lateness and flight costs are scaled. The episode score is calculated by applying different weights to missed deliveries, lateness, and flight costs, with the objective being to minimize this score.

An overall score for a submission is calculated by linearly normalizing against baseline scores from a random agent (zero) and a "shortest path" baseline algorithm (one). Scores greater than one suggest better performance than the baselines, while negative scores indicate worse performance than the random agent. The overall score is the sum of normalized scores across all tests and levels.

First Year Approach

The initial first year strategy was build-smart: solve instances in order to optimize the scoring function. Agent details can be found in (Delanovic et al. 2023). The winning submission for the airlift problem employed a multiphase architecture. The approach involved assigning goal stacks to cargo items based on route constraints, generating flight plans to move cargo towards their goals, scheduling flights to reduce airport congestion, and determining cargo loading and unloading actions based on airport conditions and flight plans. This multilayered strategy allowed for flexible responses to changing route conditions and optimization opportunities.

Cargo was assigned a stack of goals, with intermediate goals computed from a meta-map connecting airports shared between the aircraft type maps with load and unload edges. Due to route malfunctions, the shortest path calculations considered a sum of malfunction time and travel time to determine the most efficient routes.

The agent's controller used a multilayered approach. In the first phase, cargo items were assigned goal stacks based on shortest paths, considering transitions in aircraft type. In the second phase, flight plans were generated to satisfy cargo goals, with behaviors like loading local cargo, delivering cargo, picking up assigned or unassigned cargo, staging for secondary goals, and refueling. The third phase focused on reducing airport congestion by selecting flight plans that avoided exceeding airport capacity. Actions for each aircraft were determined based on the flight plan and current state, with a greedy approach to loading. A scoring rate metric (points per second) was used for parameter tuning to improve performance under time constraints. Shortest path calculation used malxtime, malfunction time times 1000 plus the edge time, instead of edge time alone. Additionally, sending active aircraft to the hangar upon cargo delivery significantly reduced the time spent processing flight plans. Aircraft could be reactivated if new cargo appeared.

Second Year Approach

The second year strategy was entirely build-fast: solve instances as quickly as possible while maximizing scoring rate.

The scheduler starts with all aircraft in the reserved state. When new cargo becomes available, the flight plans of active aircraft are checked to see if cargo can be picked up and delivered along the existing planned path. If so, then the aircraft owns the cargo and is responsible for pickup and delivery to the next destination. While unassigned cargo still exists, the nearest reserved aircraft will be promoted to transport one cargo item along the current leg as well as any other pieces of cargo along that path. Minimizing the number of aircraft in flight helps reduce congestion at airports with limited processing capabilities. Such congestion at bottleneck airports will prevent timely delivery of cargo.

Map management changed significantly between tht two competitions. During the first phase of this year's competition, the routes were limited to a single aircraft type, thus making the metamap unnecessary. Maltime was stored in the route map graph presented to the scheduler. The second phase required two aircraft types to successfully transport cargo from origin to destination. For these scenarios, the metamap from above was still used, however, it was never updated after it was constructed. During development it was discovered that both maintaining this map and doing high level reactive planning due to bad edges negatively impacted scoring rate. More points were scored by just waiting for the route to clear that to compute alternative routes.

Observations

Scoring for the competitions had two serious issues which made it difficult to compare approaches. First, the agents received score for a test only when all seventeen levels were completed. While the early tests involving small maps can be solved in one or two seconds, the later tests on very large maps with many cargo and aircraft can take minutes to solve a single level. This process causes large discrete jumps in scoring as an agent improves over time. It also supports a speed first approach in that completing seventeen levels quickly but poorly yields a higher score than optimally solving fewer levels in that test. Future competitions should accumulate scores for each level and uses the tests for complexity grouping only.

Second, the normalization process can introduce very large level scores and skew the results. Unfortunately, these normalized scores tell us nothing about how much room for improvement exists for each level. During development, an additional metric called "zero score", which presumes that all cargo instantaneously arrives at it's destination on the first simulation step, indicated that additional optimizations were unnecessary when the scores were close. An alternative approach of normalizing between a simple algorithm and zero would eliminate the scoring disparity as well as ground the score with a possible optimum, zero, instead of two arbitrary algorithms.

Conclusion

This paper described two approaches used by the winning agents of the Airlift Competition sponsored by the AFRL. The first year's agent was initially designed as a smart problem solver. This strategy changed in the second year to one of optimizing the agent's scoring rate. It was surprising that map updates and rerouting due to bad routes negatively impacted the scoring rate and could be safely removed from the process. Some difficulties with the evaluation process were discussed, both test completion and normalization introduced artifacts into the scores. These artifacts can be avoided by accumulating each level score and normalizing the raw scores using a baseline algorithm and zero.

References

Beckus, A.; Delanovic, A.; Platts, J.; Loy, A.; and Chiu, C. 2022. A methodology for flattening the command and control problem space. In *Proc. SPIE 12113, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*.

Delanovic, A.; Chiu, C.; Kolen, J.; Gnanasekaran, A.; Surana, A.; Srivastava, K.; Zhu, H. A.; Lin, Y.; Bukingolts, N.; Willis, D.; Arustamyan, N.; Sardouk, A.; Huynh, M.; camps, D. G.-D.; Smith, K.; Bragg, D.; and Beckus, A. 2023. Results of the airlift challenge: a multi-agent AI planning competition. In Solomon, L., and Schwartz, P. J., eds., *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications V*, volume 12538, 125381E. International Society for Optics and Photonics.