

MULTI-AGENT PATH PLANNING ALGORITHM FOR FUTURE AIR TRAFFIC MANAGEMENT

I. PROBLEM DESCRIPTION

Air traffic control is currently operating close to saturation. The amount of traffic is expected to double by 2025. New decision tools to provide Traffic Flow Managers adequate information are needed to handle the upcoming increase in traffic and diminish its environmental impact. This paradigm shift is sustained by a global effort, namely NextGen ([1], [2]) in the US and SESAR in Europe.

The main airspace allocation constraint is sector capacity. A sector is a region of airspace over which traffic is monitored by human operators. The number of aircraft per time unit in a sector is limited because of the limited workload an Air Traffic Manager can deal with. In order to make Traffic Flow Managers able to deal with more intensive workload, there is a need for efficient multi-agent path planning algorithm adapted to the specificities of civilian aircraft management (fast computation, zero risk, fuel efficiency, travel time, etc). In our intent to contribute to this work, we would like to compare the performances of several multi-agent motion planning algorithms, based on the A* algorithm, as solution to this problem.

II. RELATED WORK

This problem is at the junction of two different fields: Air Traffic Management (ATM) and Robotic Path Planning. ATM focuses on risk minimization, cognitive aspects (there is a pilot in the plane who might prefer some manoeuvres) and fairness, which discards solution where an aircraft sees the cost of its path increasing too much. Multi-agent path planning often focuses on the optimality of the set of path, the computational efficiency of the algorithm and often prefer to forget the non-holonomic constraints of the dynamic of the aircraft.

A. Air Traffic Management

In ATM, several solutions to the problem have been proposed using linear ([4], [7]) and quadratic ([3]) programming that minimize total flight time ([7]) or maximize fuel efficiency ([8]). However these solutions are computationally expensive and do not support a very large number of aircraft at a time.

B. Multi-agent path planning and A* variations

When multiple agents are needed to be directed in the environment, A* is not sufficient anymore: different variations are described in [9]. Local Repair A* adapts A* to on-demand reroute but can end up in deadlock in some maps. Cooperative A* decouples the task into single agents searches sharing information. Hierarchical A* improves a heuristic through an abstraction of the state space (we compute abstract distances). The hierarchy refers to a series of abstractions, each more general than the previous and not restricted to spatial hierarchy. Hierarchical Cooperative A* ([9]) is a decoupled planner of the Local Repair A* family that has been shown to be really efficient.

C. Direction maps

A direction map ([6]) stores information about the direction that agents have travelled in each portion of a map. Agents then use this information during planning. Moves running against the direction map receive additional penalties so that agents are encouraged to move in the direction indicated in the DM at every step. DM algorithm has

been used with A* in [6]. Our implementation of the DM algorithm will use a HCA* and a WHCA* planner instead.

III. APPROACH

Our contributions involves both algorithms adaptation and modelization of the air traffic management as a cooperative planning problem. We implemented different motion planning algorithms issued from the A* algorithm and adapted them to the constraints of ATM. In the first group of algorithms (CA*, HCA*), we only care about the informations given about a trajectory: no information about previous trajectories is used.

On the other hand, we study an algorithm observations of real aircraft trajectories highlighted the existence of similar trajectories that could be clustered ([5]), therefore we studied the possible benefits of using Direction Maps with these algorithms. As Direction Maps promotes trajectories that go with the general consensus, we expect that merging Direction Maps and multi-agents A* algorithm will improve the overall performances.

We made the following assumptions, valid for all our work:

- All planes have the same speed, which is a reasonable assumption as we consider planes flying at high altitudes.
- Planes can stop on the map. Our resolution corresponding to a squared nautical mile, this means that the plane would do a circular movement, but the cost of such a manoeuvre would be very high for the planner.

A. Algorithms implementation

We implemented four algorithms. The first one (CA*) plans the trajectories by taking agents one after the other. For each agent an A* algorithm finds the optimal path taking into account the trajectories planned for previous agents. Thus, the search has to be made in a 3D space including time space. Trajectories are shared between agents by a reservation table. The second one (HCA*) makes some improvements. Instead of using a given heuristic for the A* search, HCA* computes an abstract distance that is the length of the shortest path to the goal, ignoring the time and other agents. The shortest path is found by running a reverse A*. A drawback of these two previous algorithms is the dependence to the agent ordering, that can cause deadlocks on constrained environments. The third algorithm, WHCA*, avoids this problem by planning a path for an agent while taking into account other agents for w steps. Then, the search is launched every w or less steps after changing randomly the agents ordering. The basic used heuristic is the distance as crow flies. The last algorithm that we implemented is an attempt to combine between HCA* and distance map (DM).

DM can be used with any heuristic based algorithm. A distance map indicates for each cell of the grid a preferred direction. It is used to change the underlying costs of traversing the world. In our case, HCA* expands nodes in order of increasing cost values f , defined by $f = h + g$. When we combine HCA* with DM, the g -cost is modified to encourage cooperative movement. In other words, nodes are most likely to be expanded if the direction given by the heuristic h and the direction of the DM are the same. As planes have similar starts and goals, we expected that the DM forms a kind of highway for planes.

A DM has to be learned from paths of agents in previous planning. Once a plane as completed is trajectory, the DM is updated and is used to plan next planes and so on. There are several ways to combine

HCA* and DM. We decided to use the DM in HCA* to both calculate the abstract distance and plan next moves. In other words, the DM is used to modify both the g-cost when the abstract distance is computed and the g-cost that is used by HCA* to choose the next node to expand.

B. ATM refinements

Modelling the ATM problem as a cooperative planning problem is an important issue. In order to adapt our algorithms to the problem of ATM we added some refinement. The first problem we faced, was how to discretize the space. We chose to represent the space as an eight-connected grid map and keep the time continuous. However the reservation table uses an intrinsic discretized time representation. At each time a plane holds four cells of the grid in a square. This corresponds to a safety zone around each aircraft. We defined it as a 3 nautical miles radius around the plane (corresponding to one pixel on the grid around the plane in each direction except the rear one).

Secondly, we do not allow direct crossing either vertically, horizontally or on the diagonal. This is done by reserving future nodes in the direction we are going. This reservation can lead us to reserve more than one node in several cases but it never seemed to arm the traffic in our simulations.

Finally, we extend the cost model in order to minimize the number of manoeuvre (ie heading change) that creates discomfort for passengers. This is done by multiplying the actual cost by a constant alpha ($\alpha \sim 0.01$) with an exponent proportional to the number of manoeuvres: $Cost_{abstract} = Cost_{real} * \alpha^k$, where k is the number of manoeuvres in the path.

C. Visualization tool

Our goal is to support the Air Traffic Manager in his task and help him face increasing workload. Therefore we created a basic user interface that allow him to visualize the aircraft trajectories. As we already know the aircraft initial and final point over a day for all the trajectories we test, the tool allows to browse the entire timeline of the day. But we implemented the algorithms in order to be able to deal with aircraft on the fly, as they enter the sector. Once an aircraft is in the sector and has specified where it is going, the Air Traffic Manager can ask the software what would be the optimal trajectory for the aircraft under specified constraints (fuel efficiency, weather, passenger constraints).

IV. EVALUATION

We evaluated the performances of the algorithm for two different kind of criteria: computational performances and ATM analysis of the paths.

A. Computational performances

We made several runs (20) of identical simulations in order to judge of the performances of the algorithm. The two criteria we are interested in are the time to compute the entire set of paths for a number of agents and a given environment and the number of nodes explored by the A* algorithm for computing these paths. We study the influence of the number of aircraft in the environment and of the size of the environment for this criteria.

As we used randomly generated path for this part of the evaluation, we did not test the DM algorithm because it only makes sense to use this algorithm on trajectories that are correlated.

Size of the environment: As we can see in Figure 1, the computation time of WHCA* becomes better when the size of the environment increases but a basic CA* is performing better for small environments. However if we look at the number of nodes explored by each algorithm, it appears that CA* is really inefficient. Overall, we seen that WHCA* algorithm is a good choice for large environments.

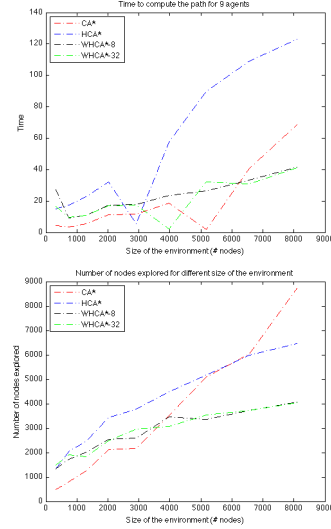


Fig. 1. Performances of the CA*, HCA* and WHCA* algorithms for different size of the environment (displayed as number of nodes).

Number of agents: As we can see in Figure 2, the WHCA* does not support well the increase in workload. Indeed, we could not extract useful data from the run of WHCA* because of the principal drawback of this algorithm: it does not offer guarantee that it will find a solution for the n agents. Even if we can suppose that some better implementation could enhance the performances facing intense workload, this result shows the need to change the algorithm itself if we want to use it for ATM purposes.

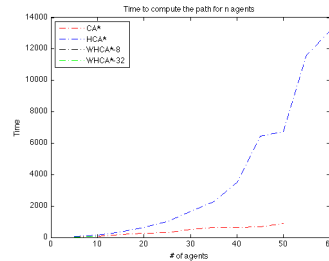


Fig. 2. Performances of the CA*, HCA* and WHCA* algorithms for increasing number of agents.

B. ATM analysis of the solution

In order to evaluate our different algorithms, we confronted them to real data corresponding to a typical day of the Cleveland sector.

The set of all the joined origin-destination pair is displayed in Figure IV-B. Finally, we escalated the constraints in order to simulate weather issues and more intense traffic (up to 10 times).

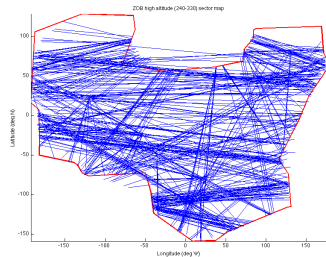


Fig. 3. Trajectories over the Cleveland sector for one day (2492 aircrafts).

As the running time for each simulation was consequent on a set of trajectories this large, we did not have time to output statistics from the simulations. We still looked at the two most important criteria for aircraft path planning: path length (fuel efficiency) and number of manoeuvres (passenger comfort).

The only algorithm that returned solutions within an admissible time on the real data was the CA*. For this algorithm we obtained good results:

- 90% of the paths on the computed solution were not more than 10% longer the path that would have been computed if the aircraft was alone.
- 80% of the paths include less than 2 manoeuvres.

Unfortunately we were not able to have a working Direction Maps algorithm on this very large algorithm. Our supposition is that because of the tendency of this algorithm to increase the number of nodes explored by the A* algorithm, the number of explored nodes in our large environment becomes to large to compute a solution in a feasible time.

C. Visualization tool and Constraints escalation

Finally, we modified our implementation of the CA* algorithm in order to allow it to deal with "on the fly" data. The planes are considered as they enter the sector and the algorithm computes a path for them in real time. This part of our work returned very good results with a capability to deal with aircraft trajectory planning in real time that far exceeded our attempts. In order to stress the environment, we tried several experiments like: increasing the number of planes present in the environment at the same time, increase the safety zone around the aircraft and add weather disturbances (by creating obstacles in the environment). In all these cases our planification tool returned good results, both for computational and ATM criteria (we went up to increase the traffic by ten times with reasonable real time computation and paths close from optimal).

V. DISCUSSION

A. Insight into the algorithm

Our approach was not successful for two reasons. The first one is that DM increase hugely the number of explored nodes. Indeed, as the

g-cost is increased by the DM, the heuristic h is still admissible but underestimates the cost of traversing the world until the goal too much. More an heuristic underestimates the g-cost of the rest of the path to the goal, more expanded nodes will be. Moreover, as HCA* expands nodes in a 3D space, the phenomenon is increased. The second reason is that, in large environments, the number of training steps to run before the DM converges is very large. These two problems make our current approach intractable. To solve this problem, we could modify our approach to use the DM only to compute the abstract distance, but because of the second reason we chose not to explore this possibility.

The complete sub-optimality of the DM algorithm as defined in [6] only appeared to us quite late. This project made us realize how important it was to consider every aspect of the algorithm we read about in a will to implement them because in our case fast and efficient path planning for small environments became extremely slow computation in a large environment.

B. Additional work

Among interesting additional work, we believe that it could be interesting to improve the WHCA* algorithm in order to make it more adapted to important workload.

Another approach for future work would be to increase the realism of the path prediction by merging an optimal path planner like CA* or WHCA* with the desired paths specified by the aircraft pilots. In this approach, a pilot would specify checkpoints for his flight while the algorithm would compute the sub-paths between every checkpoints.

REFERENCES

- [1] H. Erzberger. Transforming the nas: The next generation air traffic control system, Oct. 2004.
- [2] H. Erzberger and R. Paielli. Concept for next generation air traffic control system. In *Air Traffic Control Quarterly*, volume 10(4), pages 355–378, 2002.
- [3] E. Frazzoli. Quasi-random algorithms for real-time motion planning and coordination. *Acta Astronautica*, 54(4–10):485–495, 2003.
- [4] E. Frazzoli, Z. H. Mao, J. H. Oh, and E. Feron. Aircraft conflict resolution via semi-definite programming. *AIAA J. of Guidance, Control, and Dynamics*, 24(1):79–86, 2001.
- [5] Maxime Gariel, Ashok N. Srivastava, and Eric Feron. Trajectory clustering and an application to airspace monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1511–1524, 2011.
- [6] Renee Jansen and Nathan Sturtevant. A new approach to co-operative pathfinding. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, AAMAS '08, pages 1401–1404, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
- [7] L. Pallottino, E. Feron, and A. Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *Intelligent Transportation Systems, IEEE Transactions on*, 3(1):3–11, mar 2002.
- [8] A. Richards, J. How, T. Schouwenaars, and E. Feron. Plume avoidance maneuver planning using mixed integer linear programming. *Proceedings of AIAA Guidance Navigation and Control Conference*, 2001.
- [9] T. Standley and R. Korf. Complete algorithms for cooperative pathfinding problems. In Toby Walsh, editor, *IJCAI*, pages 668–673. IJCAI/AAAI, 2011.