



**RÉPUBLIQUE
FRANÇAISE**

*Liberté
Égalité
Fraternité*



Flight allocation in flight-centric air traffic control: Hierarchical clustering and Simulated Annealing approach

Andréas Guitart and Daniel Delahaye
SESAR Innovation Days 2025, Bled, 04/12/2025

Agenda

- 1 Context
- 2 Mathematical model and solution approach
- 3 Results
- 4 Discussion and conclusion

Agenda

- 1 Context
- 2 Mathematical model and solution approach
- 3 Results
- 4 Discussion and conclusion

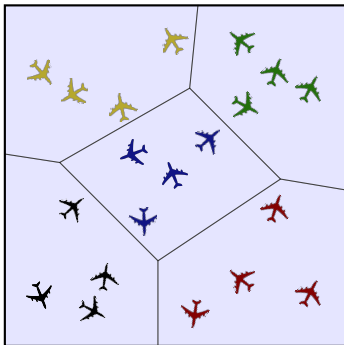
HYPER Solver project

- A re-designed ATM Layered Planning
- Moving from sector-based to flight-centric
- Moving to Single Controller Operation (SCO)
- Considering a single controller-AI teaming
- Issuing only closed-loop instructions and synchronised ATM

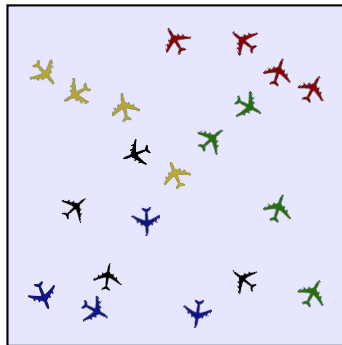
HYPER Solver

Grant Agreement No 101114820

Comparison sector and flight centric concepts

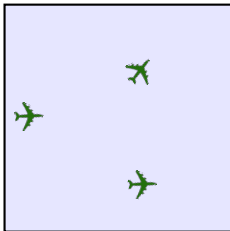


(a) Conventional air traffic control: each controller is assigned to a given sector.



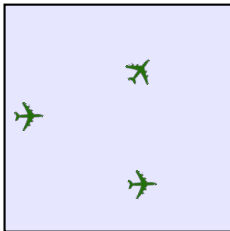
(b) Flight centric concept: each controller manages a group of flights.

Flight allocation

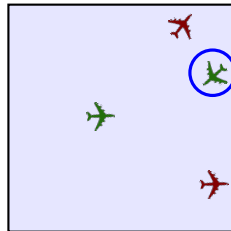


(a) Traffic situation at time t

Flight allocation

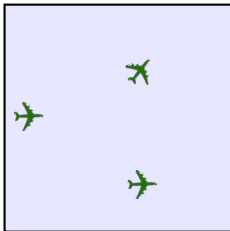


(a) Traffic situation at time t

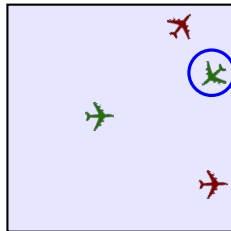


(b) Traffic situation at time $t + 1$

Flight allocation



(a) Traffic situation at time t



(b) Traffic situation at time $t + 1$

How do we automatically assign flights in a meaningful, safe, and balanced way?

Objectives

Motivations

- Model flight allocation problem
- Automatically generate flight allocation

Objectives

Motivations

- Model flight allocation problem
- Automatically generate flight allocation

Previous work

- Two MILP models
- Difficult to have a good balance of workload and a relevant allocation
- A little bit slow for real time application

Andreas Guitart, Daniel Delahaye. *Flight allocation in flight-centric air traffic control: A MILP model approach*. International Air Transportation Research and Development Symposium, Jun 2025, Prague, Czech Republic

Agenda

- 1 Context
- 2 Mathematical model and solution approach
- 3 Results
- 4 Discussion and conclusion

Algorithm architecture

Algorithm architecture

Inputs

- F the set of flights
- C the set of available controllers

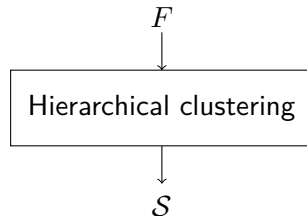
Algorithm architecture

Inputs

- F the set of flights
- C the set of available controllers

Objectives

- Group flights in F into a set of clusters S whose interaction exceeds a predefined threshold with an Hierarchical clustering



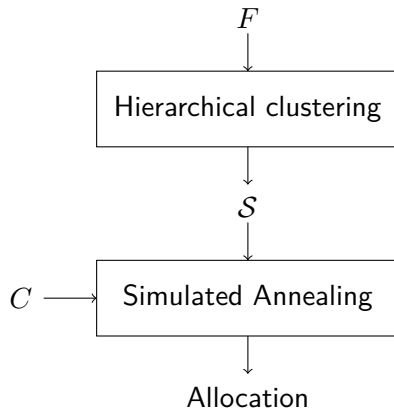
Algorithm architecture

Inputs

- F the set of flights
- C the set of available controllers

Objectives

- Group flights in F into a set of clusters S whose interaction exceeds a predefined threshold with an Hierarchical clustering
- Assign these clusters to the controllers to balance the workload

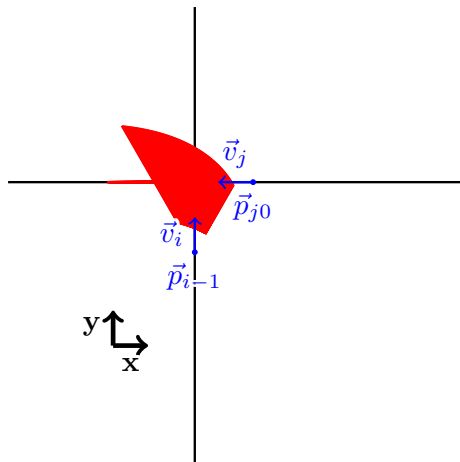


Hierarchical clustering

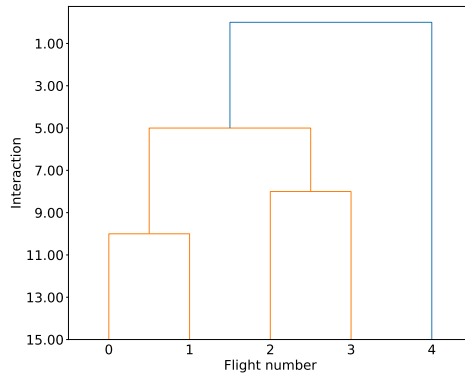
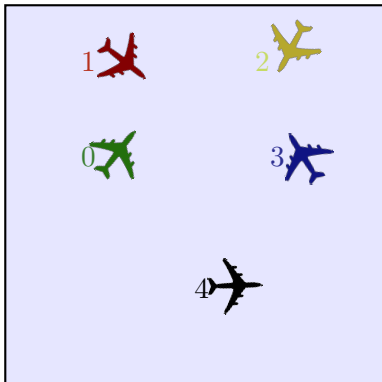
- Unsupervised learning technique that aims to build a hierarchy of nested groups based on the similarity between elements
- Create a set of flight clusters in which all flights are a high level of interaction
- Define an interaction measure
- Define a threshold of interaction τ

Conflict duration for heading maneuvers (CDH)

- Evaluate the duration of the potential conflicts
- Flight interactions are recomputed every twenty minutes
- Interactions are evaluated every 10 seconds by measuring the duration in seconds of any potential conflict over the following minute
- The final interaction score between two flights is defined as the maximum value observed across all these evaluations (60 secondes)



Hierarchical clustering



Balancing workload

- We have created a a set of flight clusters
- We want to assign these clusters to controller and balance the workload

Optimization problem

Optimization problem

Decision variables

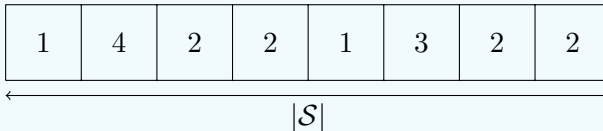
$$x_{c,s} \in \{0, 1\}, \forall c \in C, \forall s \in \mathcal{S}. \quad (1)$$

Optimization problem

Decision variables

$$x_{c,s} \in \{0, 1\}, \forall c \in C, \forall s \in \mathcal{S}. \quad (1)$$

State space representation



Optimization problem

Workload definition

Optimization problem

Workload definition

Workload of a cluster:

$$w_s = n_s + \sum_{(f_1, f_2) \in s^2} g(i_{f_1, f_2}), \quad (2)$$

- n_s the number of flights in the cluster s
- g a scoring function that maps interaction values to integer workload contributions

Optimization problem

Workload definition

Workload of a cluster:

$$w_s = n_s + \sum_{(f_1, f_2) \in s^2} g(i_{f_1, f_2}), \quad (2)$$

- n_s the number of flights in the cluster s
- g a scoring function that maps interaction values to integer workload contributions

Workload of a controller:

$$w_c = \sum_{s \in \mathcal{S}} w_s \cdot x_{c, s}. \quad (3)$$

Optimization problem

Objective function

$$\sum_{(c_1, c_2) \in C^2} |w_{c_1} - w_{c_2}| + \alpha \sum_{s_1 \in \mathcal{S}} \sum_{s_2 \in \mathcal{S} \setminus \{s_1\}} \sum_{f_1 \in s_1} \sum_{f_2 \in s_2} d_{f_1, f_2} \cdot x_{c, s_1} \cdot x_{c, s_2}, \quad (4)$$

- d_{f_1, f_2} is the mean distance between f_1 and f_2
- $\alpha \geq 0$ is a weighting parameter that governs the trade-off between balancing workload and maintaining spatial coherence in the controller assignment

Resolution approach: Simulated Annealing

Very efficient to solve large-scale problems

Resolution approach: Simulated Annealing

Very efficient to solve large-scale problems

Neighborhood

1	4	2	2	1	3	2	2
---	---	---	---	---	---	---	---



1	4	2	4	1	3	2	2
---	---	---	---	---	---	---	---

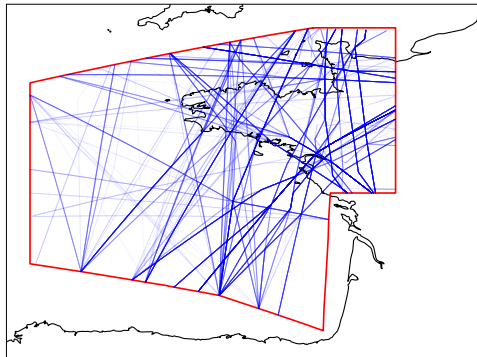
Agenda

- 1 Context
- 2 Mathematical model and solution approach
- 3 Results
- 4 Discussion and conclusion

Case study

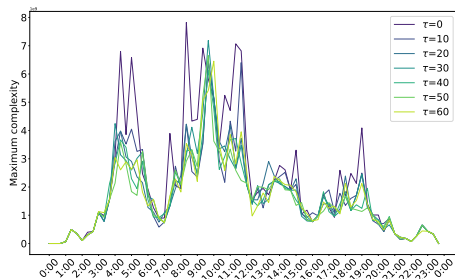
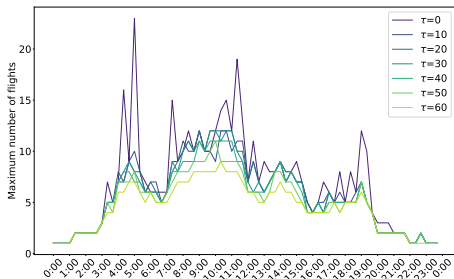
Input data

- Brest control center on June 1, 2023
- 1634 trajectories
- Allocation is updated every 20 minutes
- 15 controllers per time window



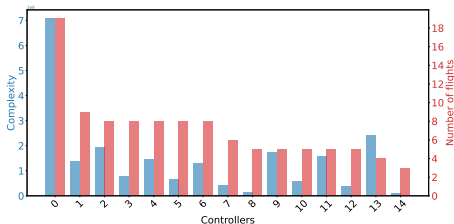
Impact of the hierarchical threshold τ

- $\tau < 20$: very high controller workload and complexity.
- As τ increases: more selective merging and better workload balance.
- $\tau = 60$: interactions ignored; only flight count balancing remains.

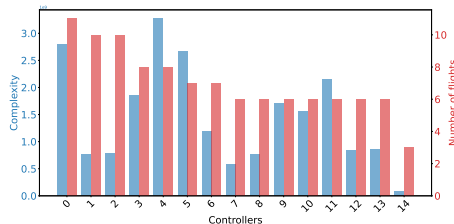


Impact of the hierarchical threshold τ

- Total imbalance of complexity between controllers



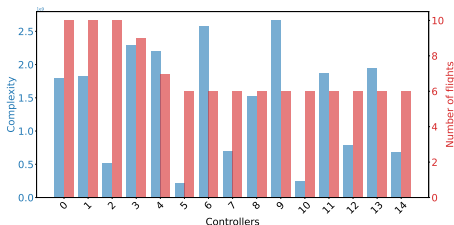
(a) $\tau = 0$



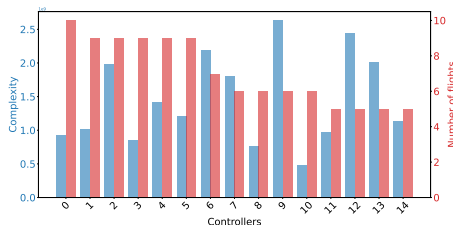
(b) $\tau = 10$

Impact of the hierarchical threshold τ

- Good balance between complexity, number of flights and consistency in terms of flight interactions.



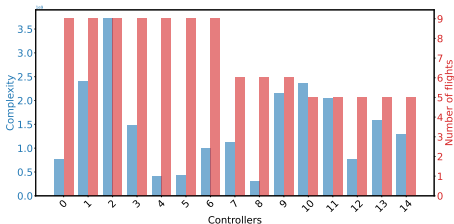
(a) $\tau = 20$



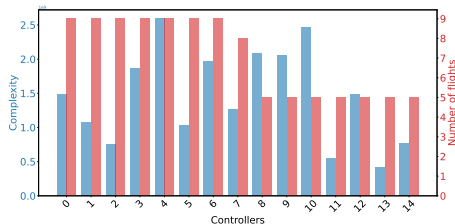
(b) $\tau = 30$

Impact of the hierarchical threshold τ

- Better balance of complexity but inconsistent in terms of flight interactions.



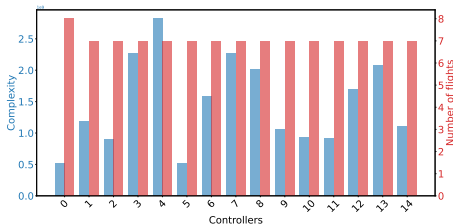
(a) $\tau = 40$



(b) $\tau = 50$

Impact of the hierarchical threshold τ

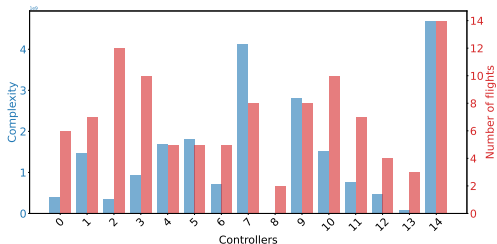
- Total balance in complexity but total inconsistency in terms of flight interactions.



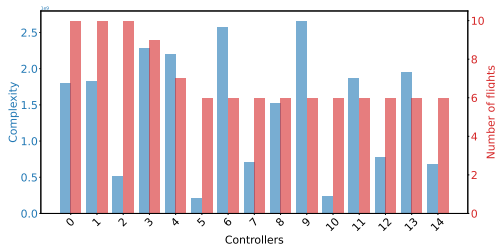
(a) $\tau = 60$

Comparison with MILP approach

- HCSA is up to $10\times$ faster: 3 s (worst case) vs 30 s for the MILP
- Better to balance the workload



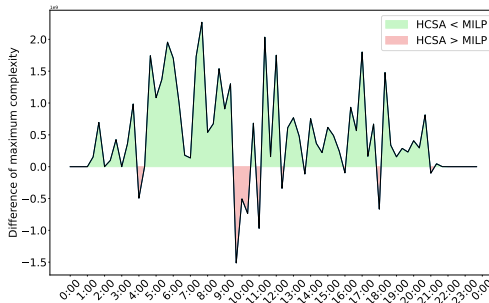
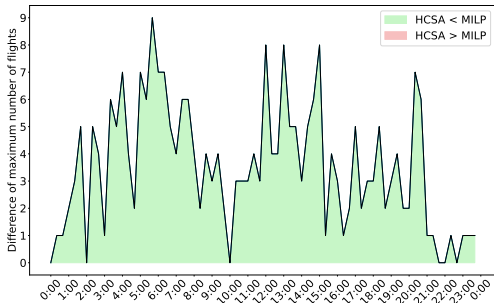
MILP



HCSA

Comparison with MILP approach

- The maximum number of flights per controller is always lower for the HCSA.
- Most of the time, the maximum complexity per controller is always lower for HCSA.



Agenda

- 1 Context
- 2 Mathematical model and solution approach
- 3 Results
- 4 Discussion and conclusion

Discussion and conclusion

Discussion and conclusion

Conclusion

- Development a new method to automatically assign the flights to controllers
- We have shown that the HCSA is significantly better than MILP approach

Discussion and conclusion

Conclusion

- Development a new method to automatically assign the flights to controllers
- We have shown that the HCSA is significantly better than MILP approach

Future works

- Test on other center airspace (Reims, Maastricht, ...)
- Compare the results with other interaction measures
- Integrate controller actions into the optimization process
- Incorporate a conflict detection and resolution module

Thank you for your attention.
Questions?