Trajectory-centered agent-based modeling of ATM
(to resolve DCB imbalances at planning stage)

Theocharis Kravaris, Christos Spatharis, Konstantinos Blekas,
G.A. Vouros
University of Piraeus (UPRC), Greece

Data Enhanced TBO Workshop @ ICRAT 2018

Founding Members
EUROPEAN UNION EUROCONTROL
DART
SESAR JOINT UNDERTAKING

DART Operational Context

Overall Aim :

Demonstrate how DART predictive analytics capability can **improve trajectory prediction in support of DCB processes at planning phase**, further reducing uncertainty and improving ATM operations and services provided.

Scenario 1 (AUs): aims to **compute the predicted trajectory** that an aircraft will fly during an operation day **without considering traffic**.

Scenario 2 (ANSPs): aims to **study and determine the complexity to be considered in trajectories due to the influence of the surrounding traffic**, at the planning phase, taking into account flight plans and/or individual trajectory predictions.

DART Operational Context: Multiple Trajectories

This operational scenario concerns **the planning phase during the DCB process** (i.e some days before operation).

The separation between aircrafts is guaranteed; therefore**, the scenario does not consider conflicts: Resolutions adopted by ATCO won't be part of the scope in the operational scenario WP3**.

In this case**, regulations of type C (i.e. delays)** will be applied to trajectories due to **the imbalances between demand and capacity**, so DART methods recalculate and obtain the final trajectories taken into account **surrounding traffic**.

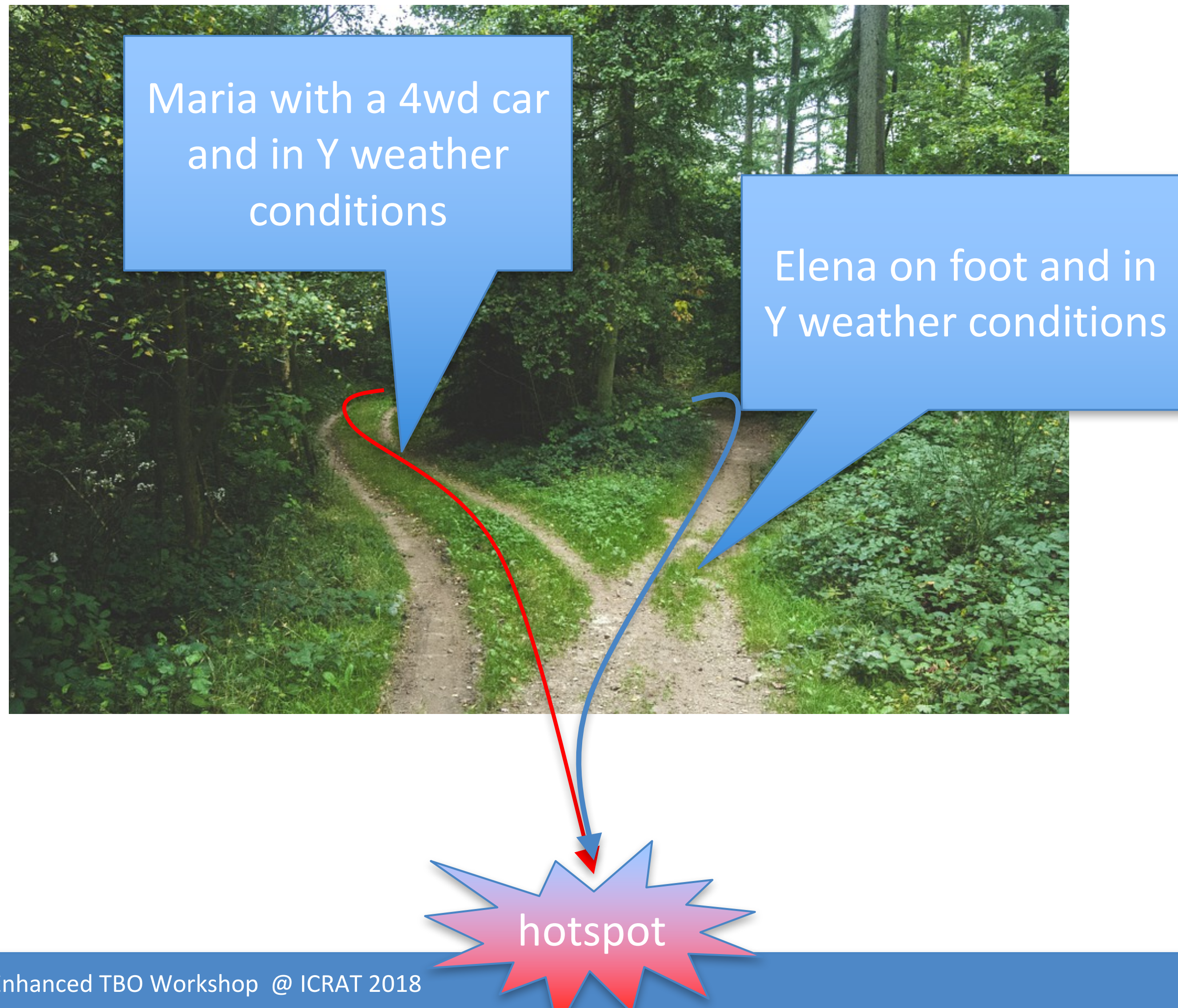
Target:

Improvement in airspace capacity management.

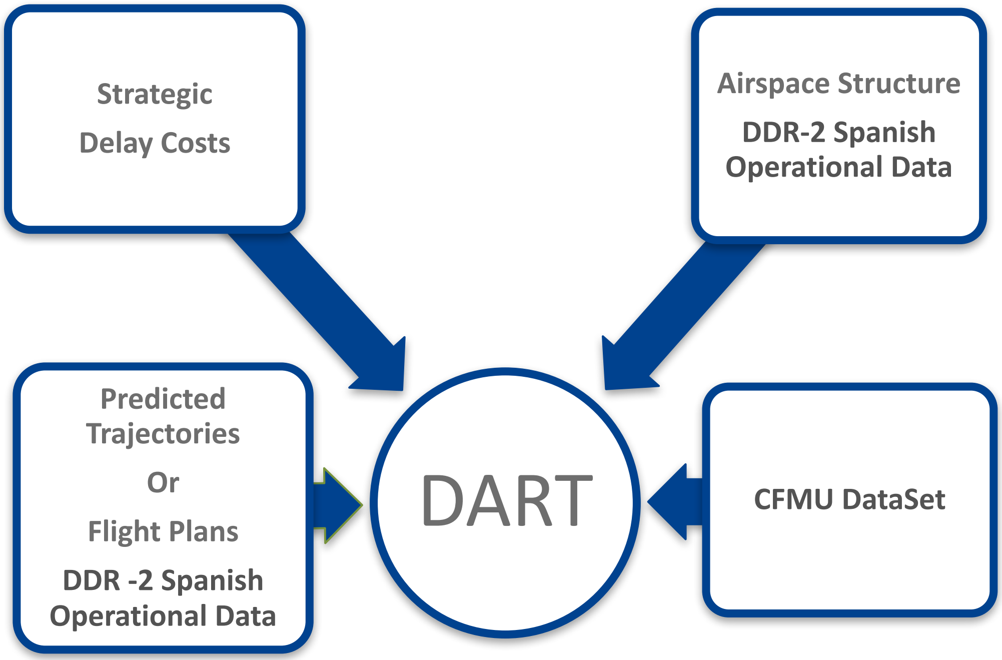
Agent – Based Collaborative Algorithms

In support for DCB process at the planning phase towards SBTs

Agent-based DCB problem resolution: Agents, trajectories & hotspots



Data Sourses



Trajectory Abstraction Model

- Each flight is situated in some sector at all times
- Abstraction of flight trajectories
  - In space and time: time series of sectors crossed **with entry/exit time**

Sufficient for DCB operations

Original: … 2 (eT,xT) 5 (eT,xT) 1 (eT,xT) 4 (eT,xT) 8 (eT,xT) ….
Delay 1: … …. …. 2 (eT’,xT’) 5 (eT’,xT’) 1 (eT’,xT’) 4 (eT’,xT’) 8 (eT’,xT’) ….

eT: Entry Time in sector S & xT: Exit Time in sector S

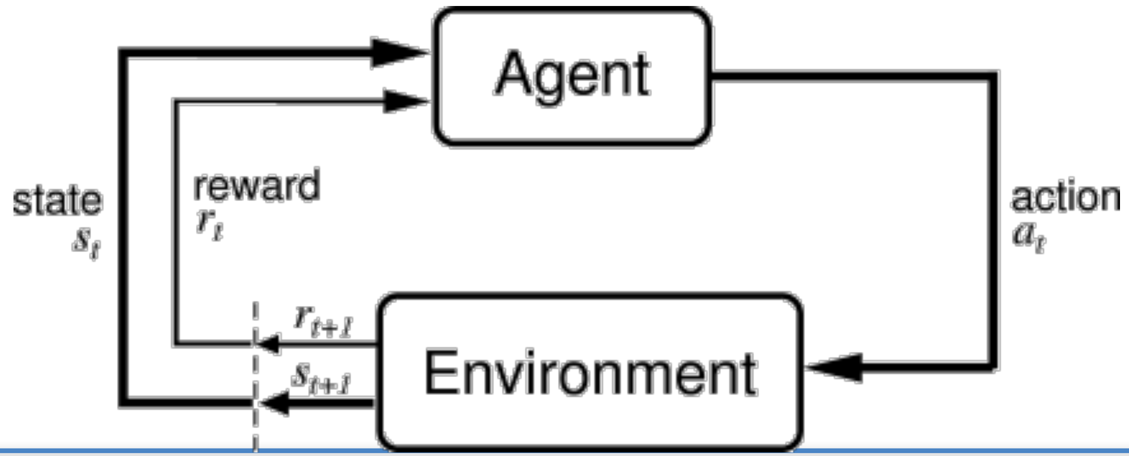
- To accommodate delays: shift the entry/exit time per sector

The Big Picture

- Solving the DCB Problem:
    - **Formulate the DCB Problem as an multi-agent Markov Decision Process (MDP)**.
    - Solving the MDP = **planning**.
    - **Multi-Agent Reinforcement Learning (RL) algorithms** inherently appropriate.

Reinforcement Learning Primer

- Agent: A particular flight executed by a specific aircraft
- **Markov Decision Process**
  - State Space (delays, hotspots)
  - Action Space (adding or not delay at each time point)
  - Transition Model (State + Action = New States)
  - Reward Model (*f* (State) = Reward)
- **Goal: Optimal Action at every state (a policy)**



Source: Reinforcement Learning: An Introduction (Sutton, Barto)

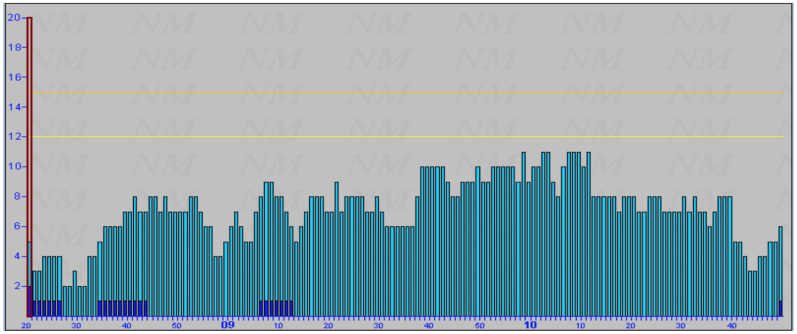
MDP model

- State vector:
  - $d_i \in \mathbb{N}$, Imposed delay so far per flight
  - Number of hotspots in which it participates (not for all methods)

- Strategy of each Agent: 0 or 1 while on ground AND max delay is not reached.

Tracking Demand Evolution (…and Hotspots)

- Measures of Demand:
  - **Hour entry Count (arrivals/hour)**
  - Occupancy Count (planes/hour)
  - Counting interval is shorter than an hour



MDP Reward Model

$$Rwd_A(s^t_A, str^t_A) = \lambda_1 * C(str^t_A, s^t_A) + \lambda_2 * D(str^t_A, s^t_A)$$

$$C(str^t_A, s^t_A) = \begin{cases} \text{-TDC*81} & \text{if TDCs} > 0 \\ \text{PositiveReward} & \text{if TDC=0} \end{cases}$$

TDC: total duration in congestions/hotspots.

$$D(str^t_A, s^t_A) = -TotalDelay * StrategicDelayCost(AircraftType)$$

Agent-based DCB problem resolution: Interacting trajectories

Interacting trajectories (agents):

Co-occurring in space and time

MARL Algorithms solving the MDP
(1st approach /4)

- **Independent Learners Approach**:
  - Each agent (flight) is **self-interested and learns by itself to resolve the DCB problem**, by measuring its own reward after each decision

State of agent i: HotSpots in which agent i is involved and own delay

Current strategy of agent i

$$Q_i(s_i, str_i) = Q_i(s_i, str_i) + \alpha[Rwd(s_{N(A_i)}, str_{N(A_i)}) + \gamma max_{str'_i} Q_i(s'_i, str'_i) - Q_i(s_i, str_i)]$$

Individual reward received from own action in the local state

Originally: global states and rewards

Factorization: local states and own rewards

MARL Algorithms solving the MDP
(2nd approach /4)

- Sparse Cooperative Q-Learning - Agent-Based Decomposition-

Edge Based Update:

$$Q_i(s_i, str_i) = \frac{1}{2} \sum_{j \in N(A_i)} Q_{ij}(s_{ij}, str_{ij})$$

Join State of agents i and j: Their delays

Joint strategy of agents i and j

$$Q_{ij}(s_{ij}, str_{ij}) = Q_{ij}(s_{ij}, str_{ij}) + \alpha[\frac{Rwd_i(s_i, str_i)}{|N(A_i)|} + \frac{Rwd_j(s_j, str_j)}{|N(A_j)|} + \gamma Q_{ij}(s'_{ij}, str'_{ij}) - Q_{ij}(s_{ij}, str_{ij})]$$

Individual reward received from own state and own action

Each agent chooses its optimal action independently

MARL Algorithms solving the MDP
(3rd approach /4)

- Sparse Cooperative Q-Learning - Agent-Based Decomposition-

Agent Based Update:

$$Q_i(s_i, str_i) = \frac{1}{2} \sum_{j \in N(A_i)} Q_{ij}(s_{ij}, str_{ij})$$

Join State of agents i and j: Both their delays

Joint strategy of agents i and j

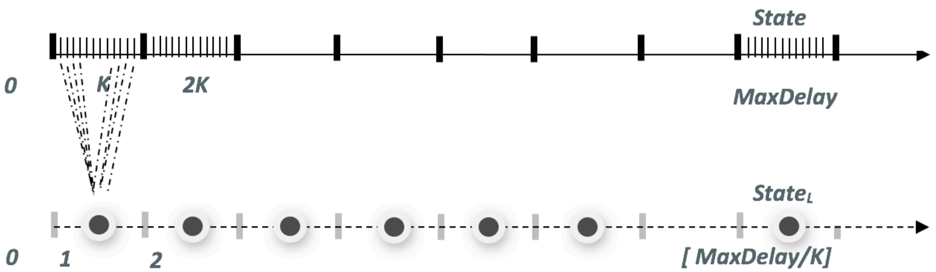
$$Q_{ij}(s_{ij}, str_{ij}) = Q_{ij}(s_{ij}, str_{ij}) +$$

$$\alpha \sum_{k \in \{i,j\}} \frac{(Rwd_{ij}(s_{ij}, str_{ij}) + \gamma Q_k(s'_k, str'_k) - Q_k(s_k, str_k))}{|N(A_k)|}$$

Individual reward received from own state and own action

Each agent chooses its optimal action independently

MARL Algorithms solving the MDP
(4th approach /4)



- Hierarchical Reinforcement Learning:
  1. Start with the **original state space**. This is the “ground” representation at state space ***State***. At this “ground” level the distance between consecutive time points is one time instant.
  2. Map State to an **abstract-feature space State*L***, where |State*L*| << |State|. This includes the abstraction of the state space so as to reduce the original space State.
  3. **Solve MDP in State*L* space**.
  4. **Map solution** from abstract space **State*L*** to ground **State** space.
  5. **Solve MDP in the original State space**.

Evaluation of multi-agent algorithms to resolve DCB problems

Evaluation Cases

10 evaluation cases of varying difficulty, by inspecting problem parameters in conjunction to the average delay **considering CFMU reported regulations**.

Each case corresponds to a specific day of 2016 above Spain

Difficulty has been determined by means of

-the **number of flights** involved,

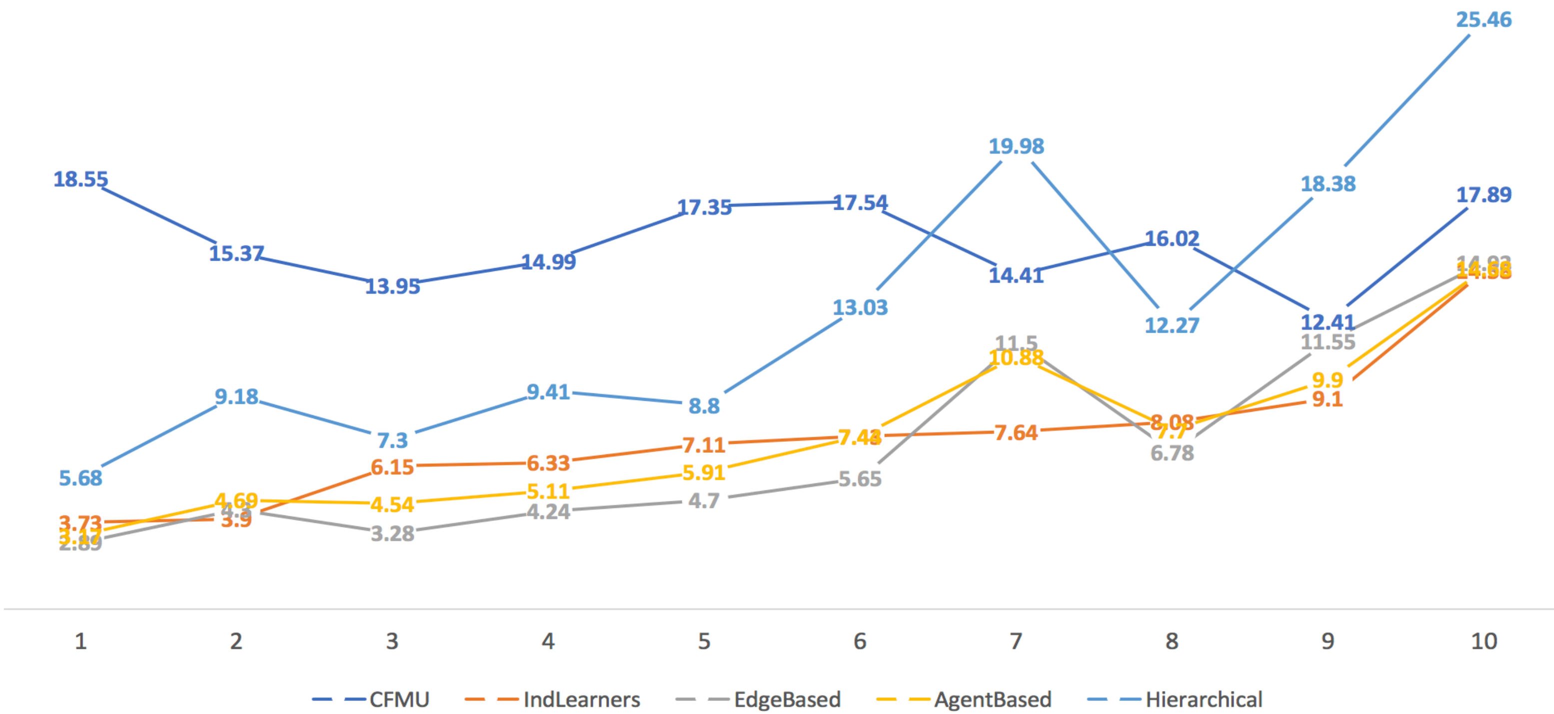
-the **average number of interacting flights per flight** (which is translated to the average degree for each agent in the coordination graph, connecting that agent with its peers),

-the **maximum delay imposed to flights** for that day to resolve DCB problems **according to CFMU data**,

-the **average delay for all regulated flights according to CFMU data**, and

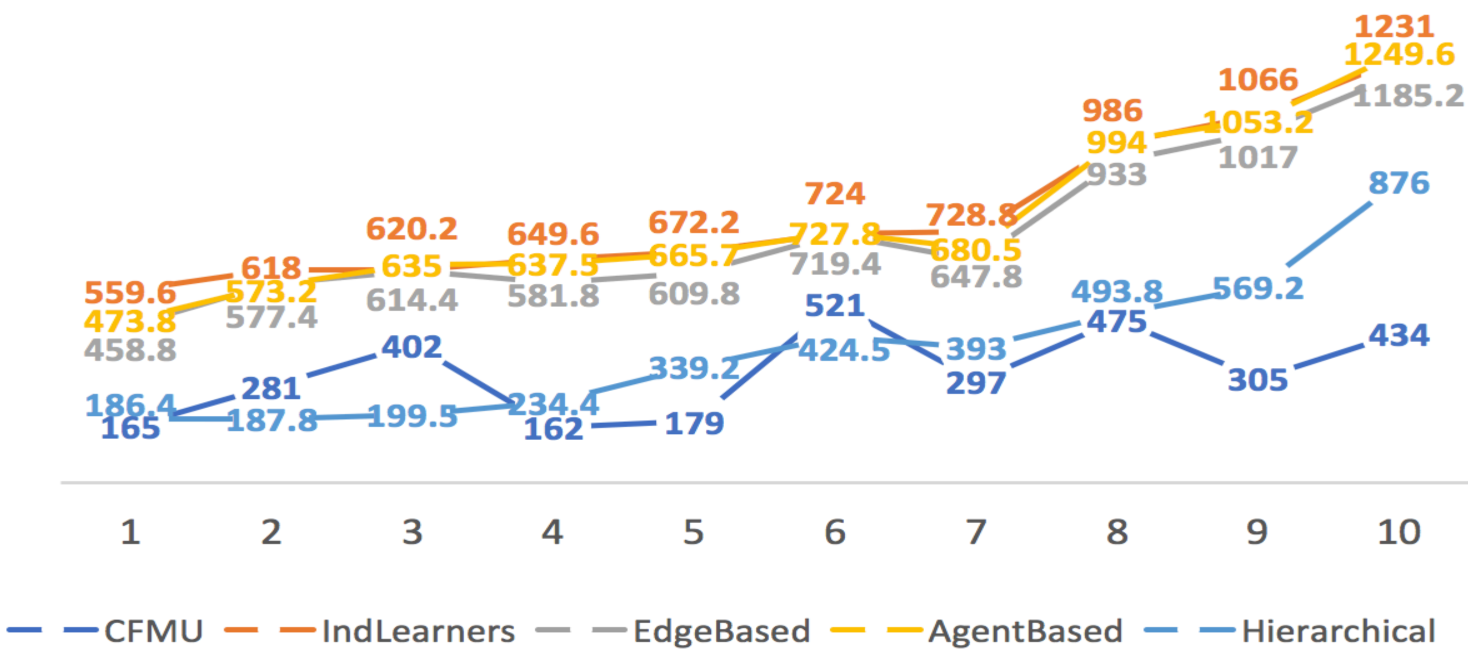
-the **number of hotspots** in relation to the **number of flights participating in hotspots**.

Experimental Results



Average delays for delayed flights.

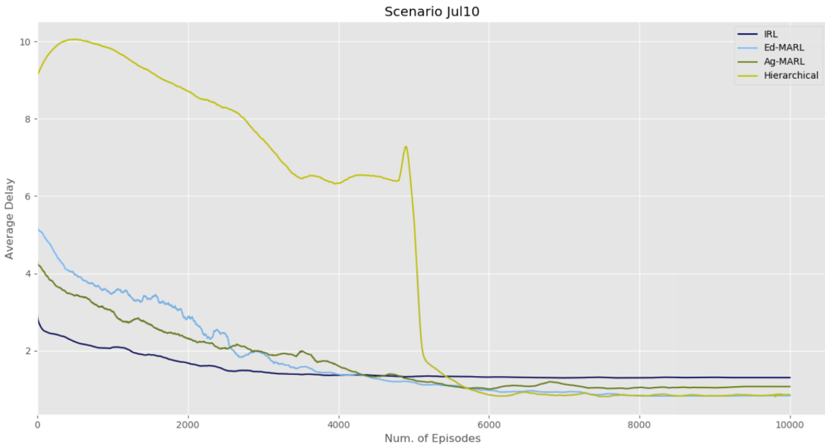
Experimental Results

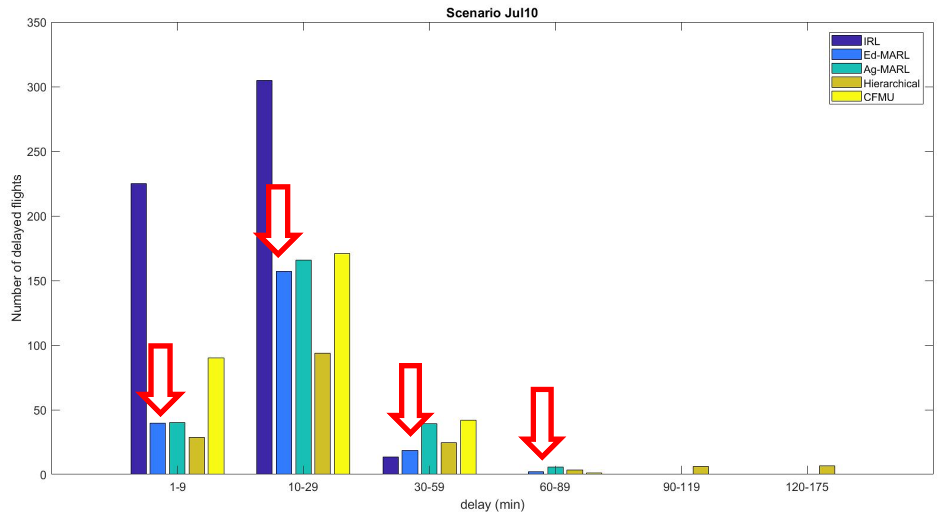


Number of delayed flights.

Experimental results

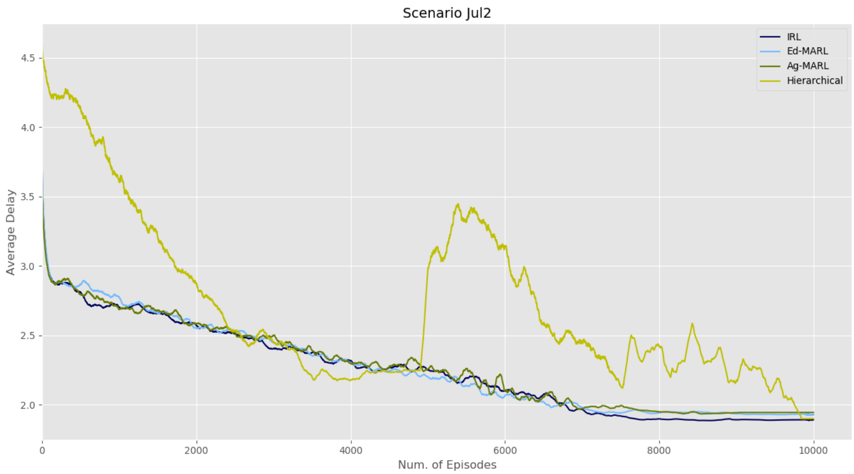
Evaluation case	Average Delay for regulated flights (according to CFMU data) (min/max)	Average Delay for regulated flights (IndLearners)	Average Delay for regulated flights (EdgeBased)	Average Delay for regulated flights (AgentBased)	Average Delay for regulated flights (Hierarchical)
Jul10	17.35	7.11	**4.7**	5.91	8.8

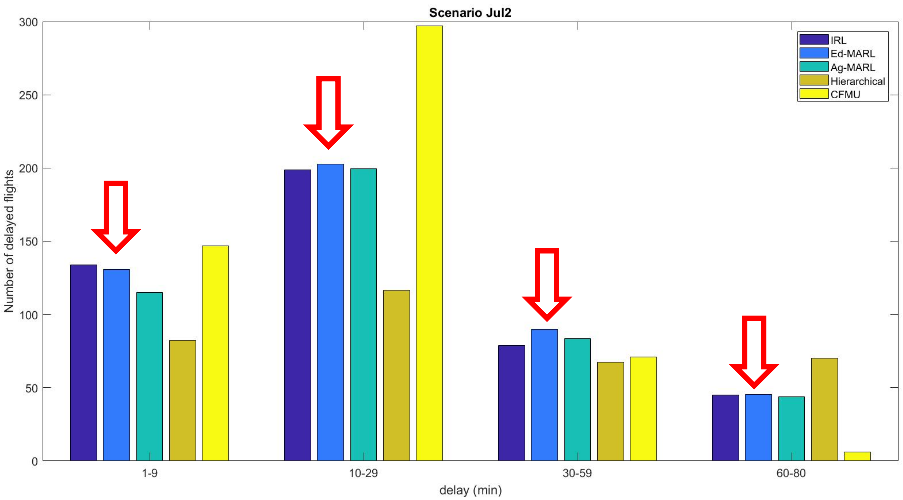




Experimental results

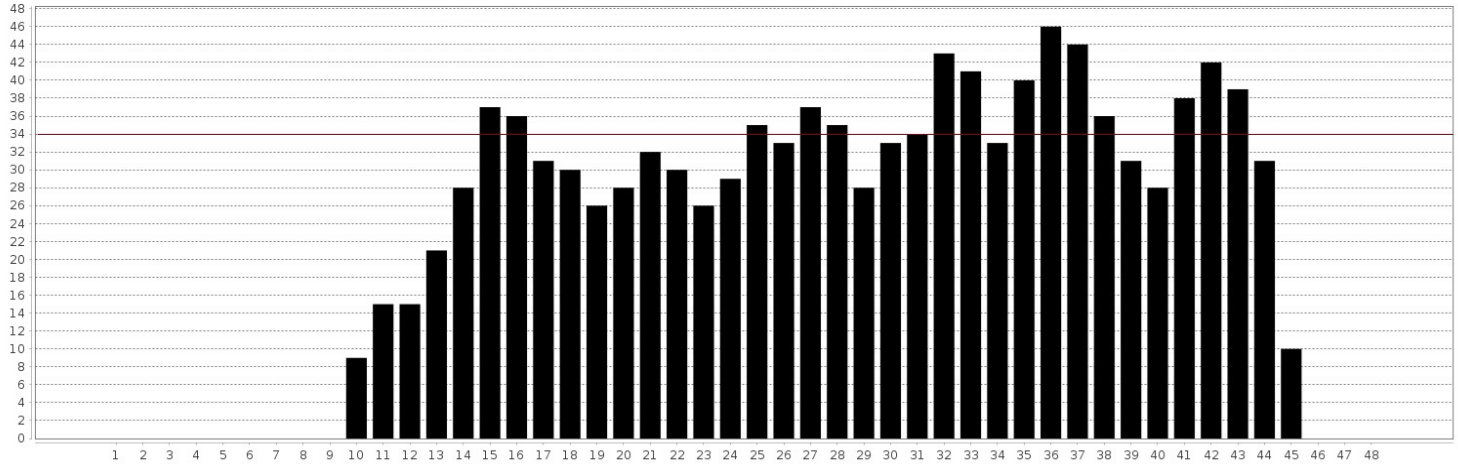
Evaluation case	Average Delay for regulated flights (according to CFMU data) (min/max)	Average Delay for regulated flights (IndLearners)	Average Delay for regulated flights (EdgeBased)	Average Delay for regulated flights (AgentBased)	Average Delay for regulated flights (Hierarchical)
Jul2	17.89	**14.58**	14.93	14.68	**25.46**



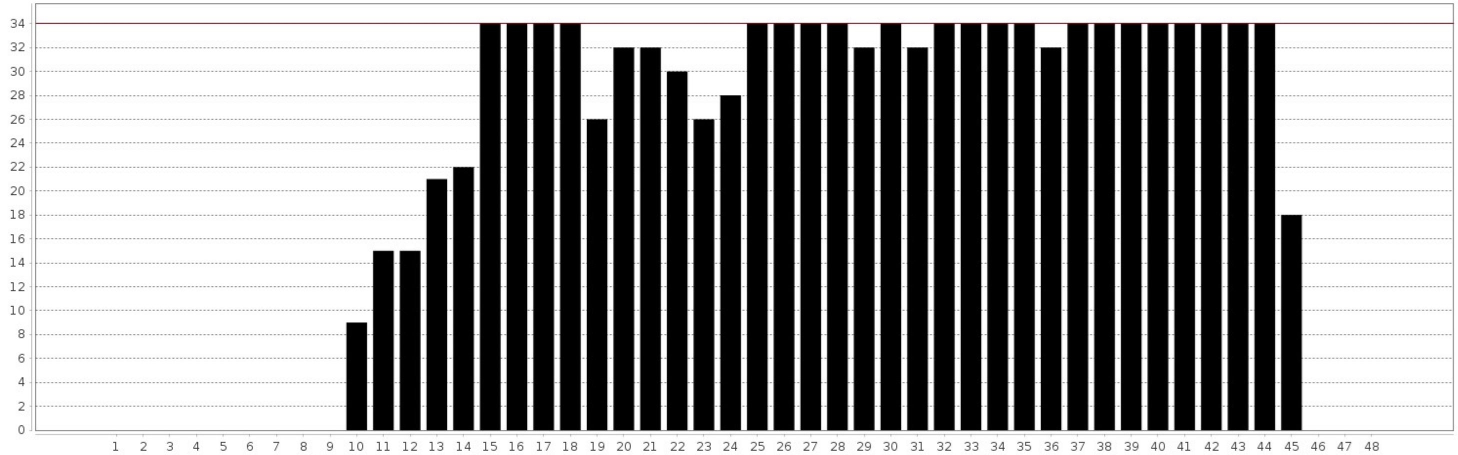


Experimental results (demand distributions at the initial and final states - IndLearners)

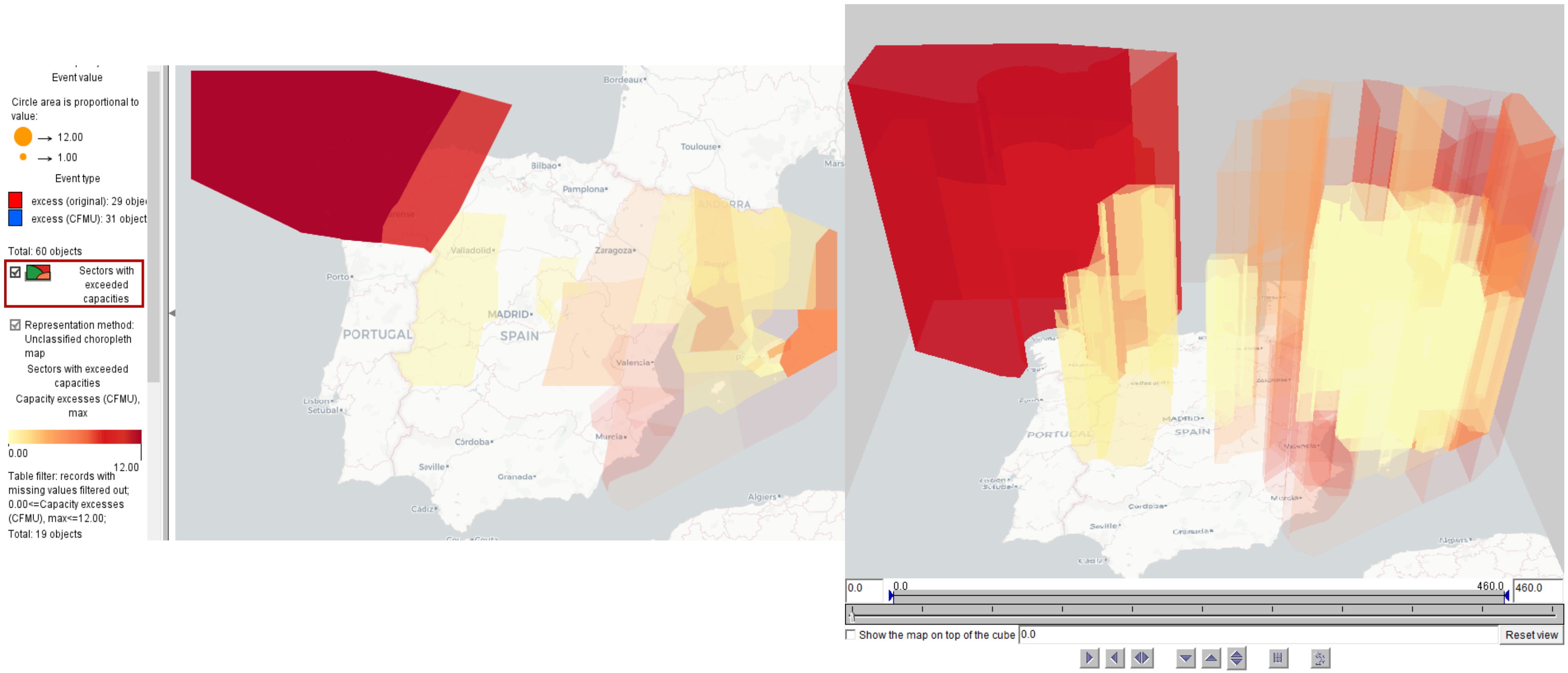
Initial Distribution Sector 105, Capacity 34



Final Distribution Sector 105, Capacity 34

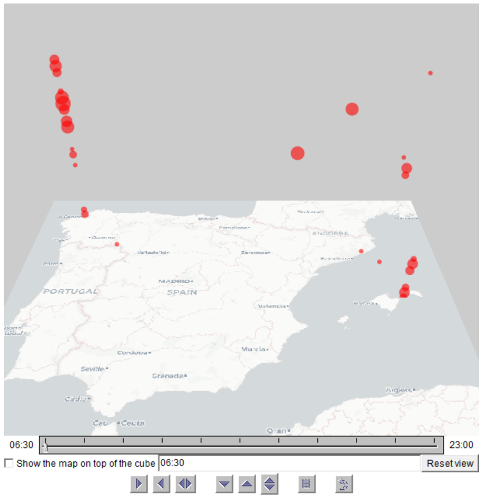


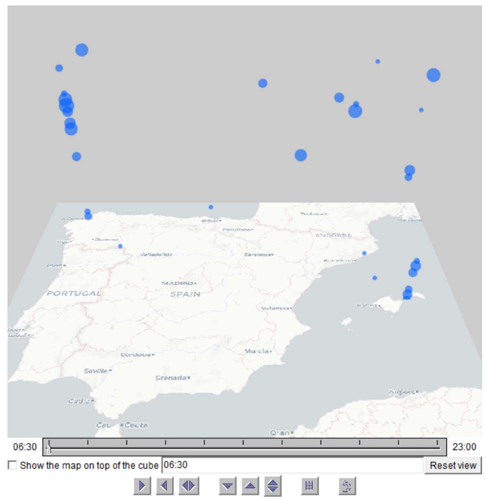
Visualizations of solutions

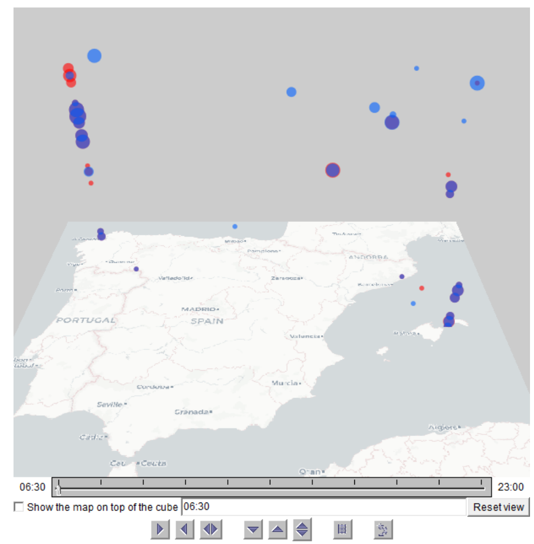


The sectors whose capacities were exceeded by the CFMU-regulated flights. The colouring from yellow to red represents the maximal capacity excess.

Visualizations of solutions

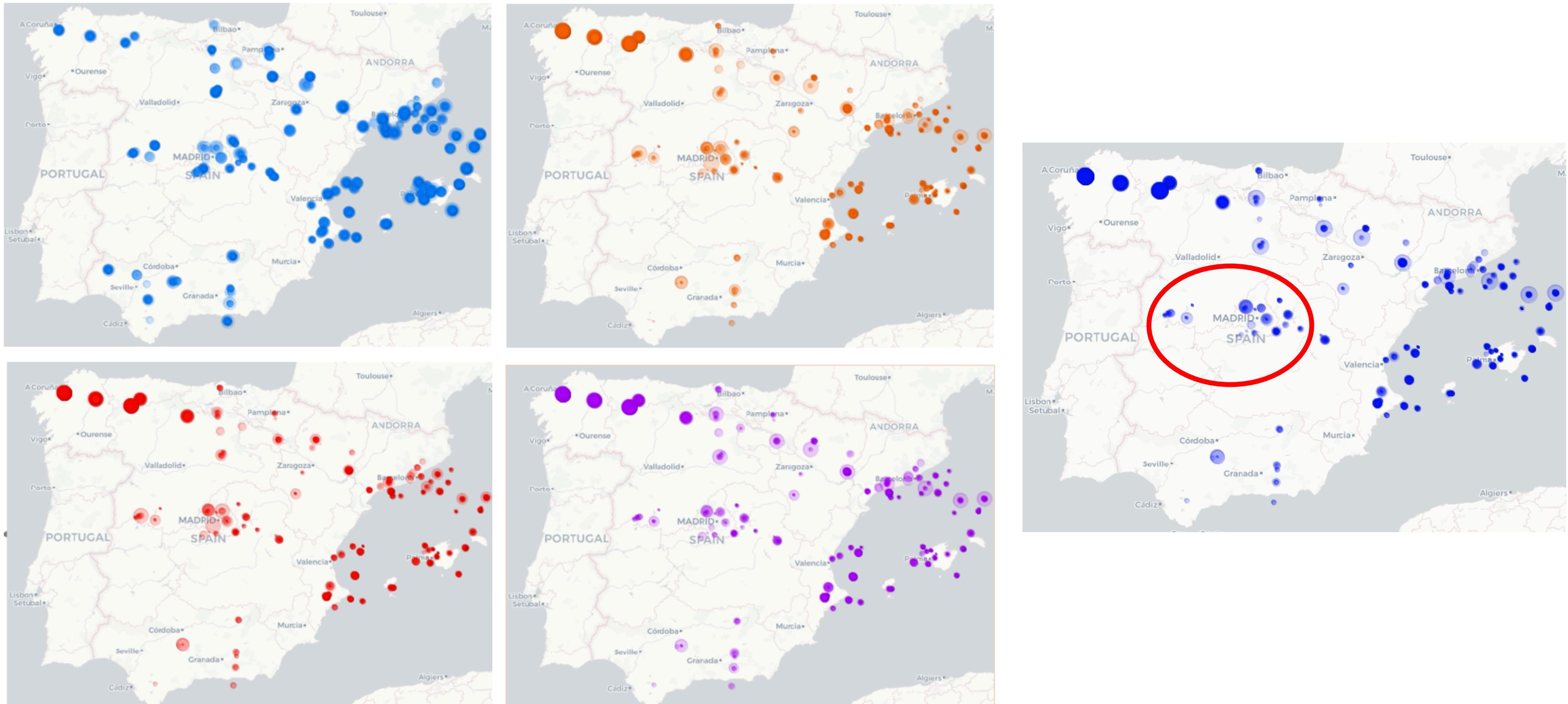






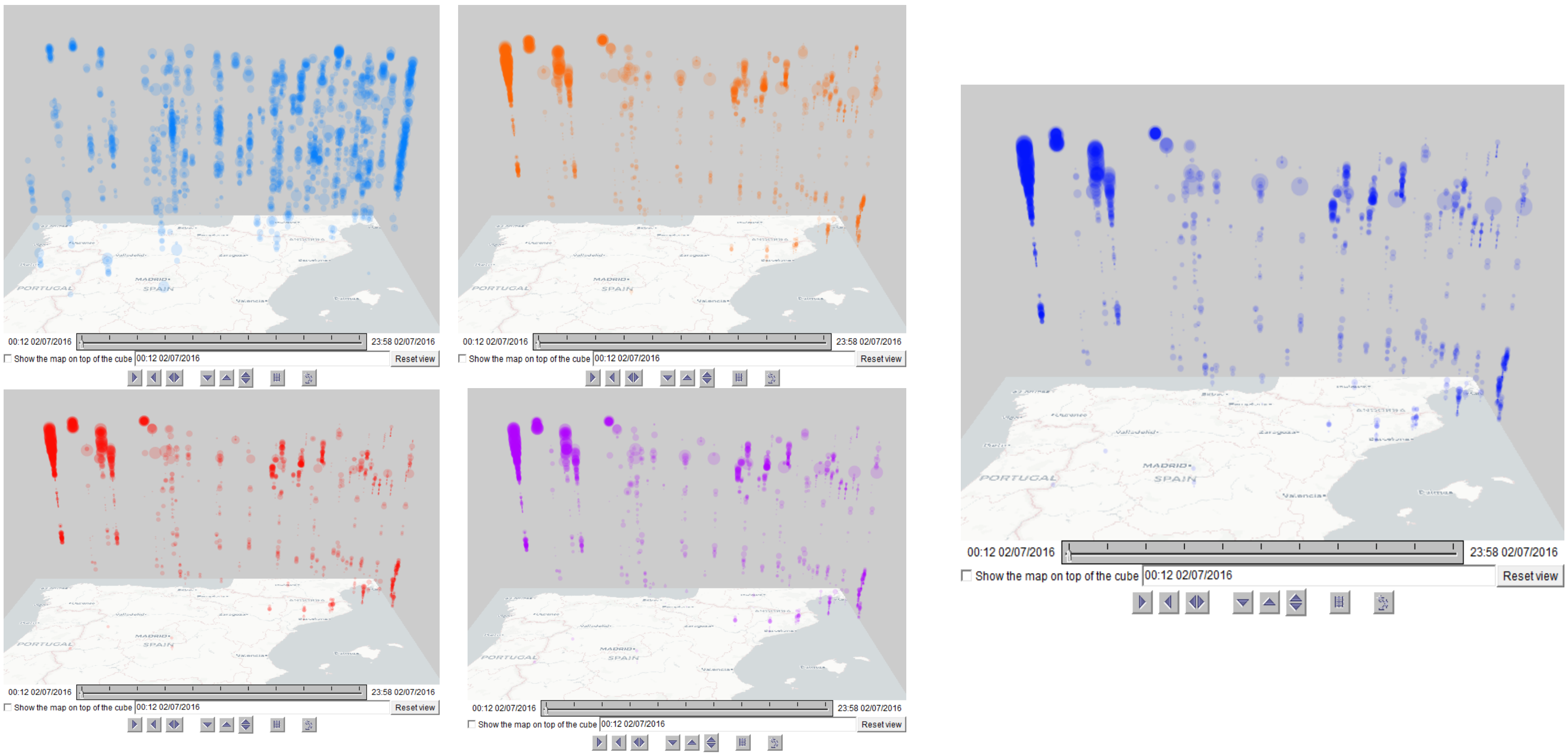
The capacity excess events are shown in a space-time cube based on the original (red) and CFMU-regulated (blue) flight data. The vertical dimension, from bottom to top, represents time.

Visualizations of solutions



Flight delays are represented by circles positioned at the sector centroids.
The sizes are proportional to the delay durations.
From top to bottom and from left to right: CFMU, AgentBased, Hierarchical, IndLearners, EdgeBased.

Visualizations of solutions



The space-time cubes show the spatio-temporal distribution of the delays. The time axis is oriented upwards. From top to bottom and from left to right: CFMU, AgentBased, Hierarchical, IndLearners, EdgeBased.

Lessons Learnt

Visualizations of results provide insights and justifications about the delays assessed, providing the necessary tools for comparing solutions and exploring “what-if” alternatives.

Comparison of results provided by the proposed methods with real-world type C regulations is not that straightforward: This has been done in a very meticulous way and one should be cautious with conclusions.

Conclusion

- **Agent-based methods have the potential for resolving DCB problems very effectively** (i.e. with small average delays for delayed flights and with zero hotspots, also considering cost indicators) and in computational **efficient ways**.
- **Agent-based methods provide** a **shift of paradigm** towards regulating flights, accounting for ATM network effects (in contrast to 1st come – 1st delayed model);
- This new paradigm inherently **enables to consider preferences regarding individual flights' delays**;
- While providing the means to **assess delays at the pre-tactical stage** to resolve all hotspots
- Contributing to **Increasing Predictability & Collaborative Decision Making.**

Data-driven AiRcraft Trajectory prediction research (DART)

Thank you very much for your attention!

This project has received funding from the SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No [number]

SESAR
JOINT UNDERTAKING

Founding Members
EUROPEAN UNION EUROCONTROL

The opinions expressed herein reflect the author's view only.
Under no circumstances shall the SESAR Joint Undertaking be responsible for any use that may be made of the information contained herein.