



DART: A Machine-Learning Approach to Trajectory Prediction and Demand-Capacity Balancing

SESAR

Belgrade, Serbia November 28-30 2017

Pablo Costas



Founding Members



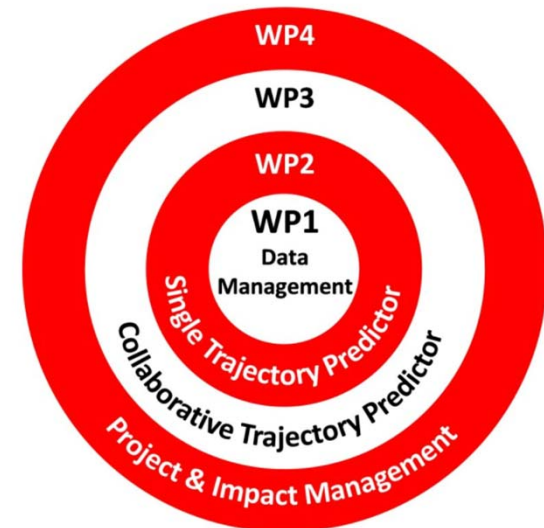
DART



DART Project



- **DART:** Data-driven Aircraft Trajectory prediction Research
 - SESAR 2020 Exploratory Research
 - *Topic ER-02-2015 - Data Science in ATM*
 - June 2016-June 2018 (currently ongoing)
- **Objective:** Address the suitability of applying big data techniques for **predicting multiple aircraft trajectories** based on **data-driven models** and accounting for **ATM network complexity** effects
- **Focus on:**
 - Single Trajectory Prediction (WP2)
 - Multiple (Collaborative) Trajectory Prediction (WP3)
- **Extended Objective:** Iterative **multi-criteria optimization process**, considering different stakeholders interests
- **Link to DatAcron H2020 project** (discrete events forecasting for moving entities)



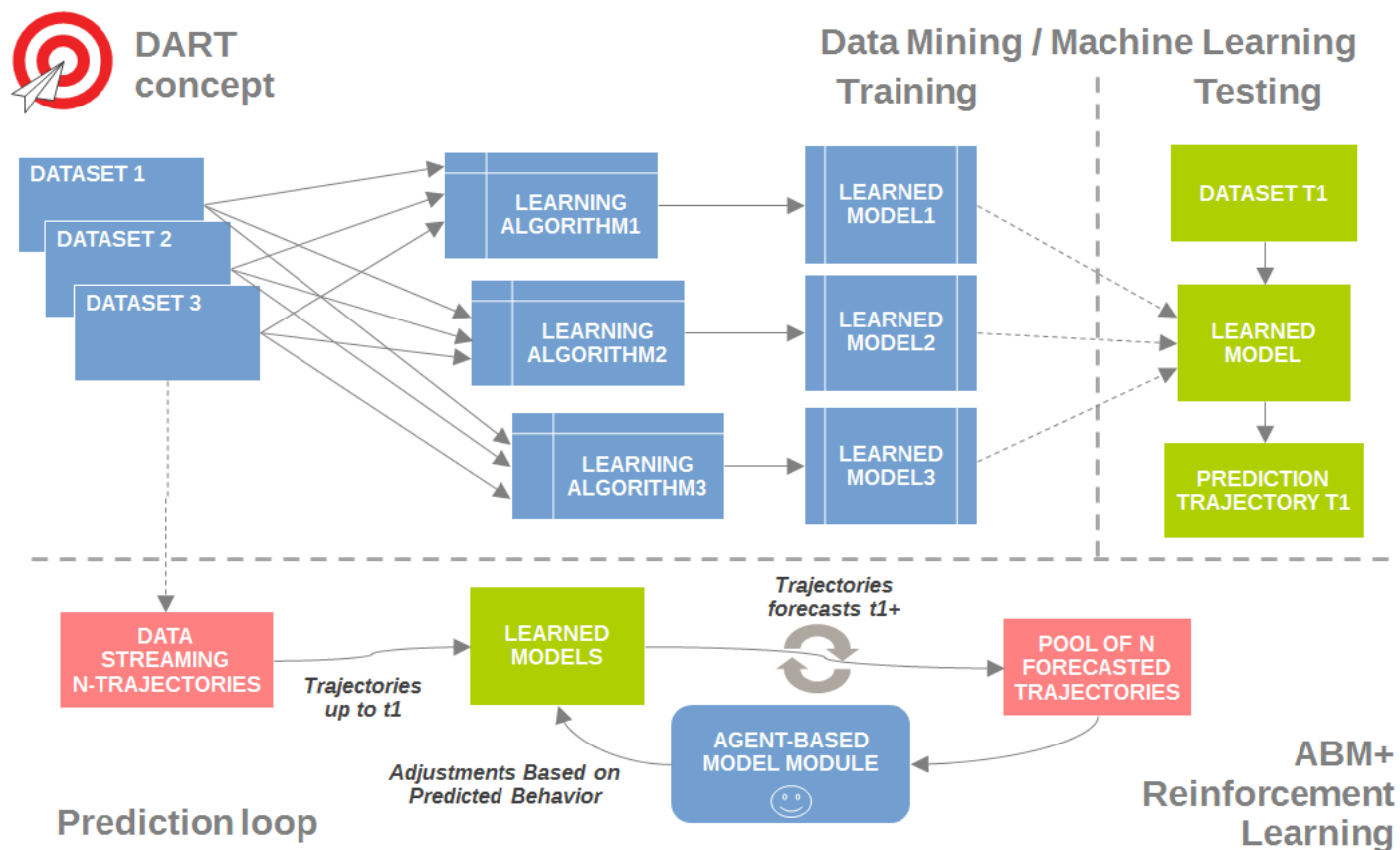
DART Concept



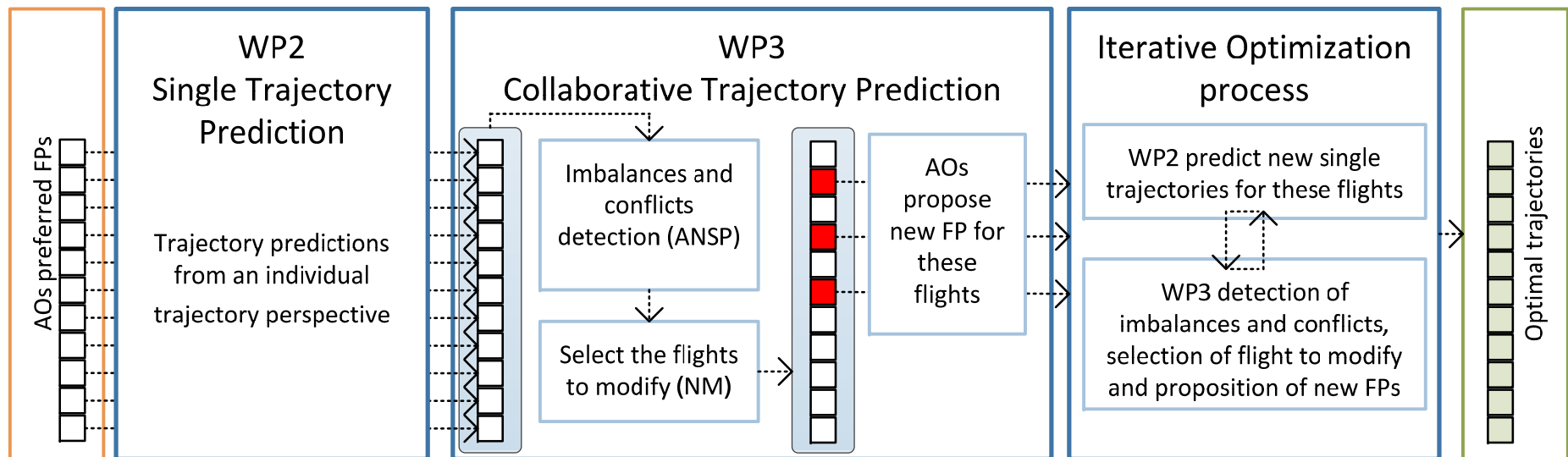
Objectives

DART will deliver **understanding on the suitability of applying data-driven and agent-based models** for enhancing our abilities to **increase predictability** of aircraft trajectories.

Increasing predictability <-> Reducing uncertainty



DART Scenarios



This scenario aims at analyzing and evaluating machine learning algorithms for trajectory predictions **from an individual trajectory perspective** (i.e. without considering traffic) from the **airspace users' point of view**.

Once detected the sectors demand-capacity **imbalances and the potential conflicts**, there will be **selected those flights to modify** in order to remove the imbalances and conflicts.

For those flights to modify: i) a new FP from AOs preferred list will be selected and ii) a new single trajectory will be predicted (WP2)

Multiobjective optimization process:

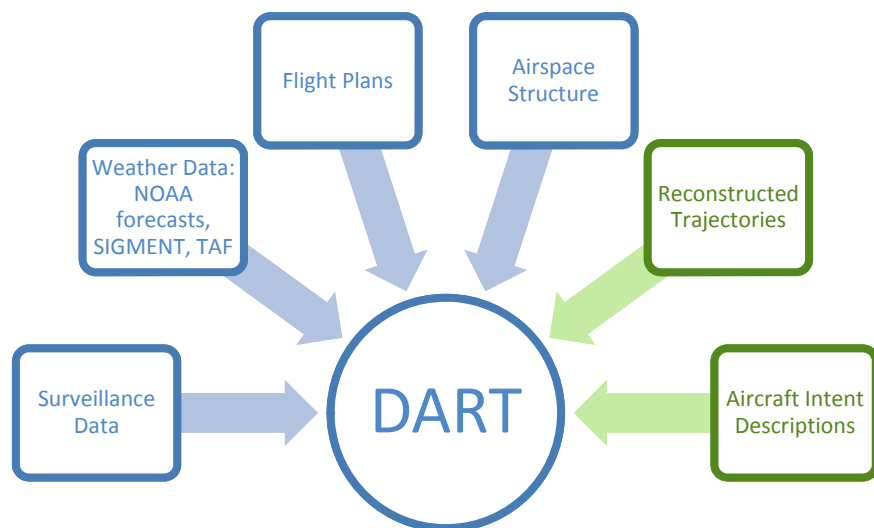
- Minimizing the sector imbalances and potential conflicts.
- Minimizing the cost thought maximizing the adherence to the airlines preferred FPs.

Single Trajectory Prediction



- A **trajectory** can be defined as the **time-evolution of the position of the aircraft's center of mass** (and other state variables)
- A **predicted trajectory** is a representation of the aircraft's future trajectory, typically given by a **chronologically ordered sequence of aircraft states**, where each state includes variables such as position (of the center of mass), speeds and weight
- When using models to predict aircraft motion, additional variables are required to predict a trajectory: **aircraft performance characteristics, atmospheric variables, aircraft intent and initial aircraft state**

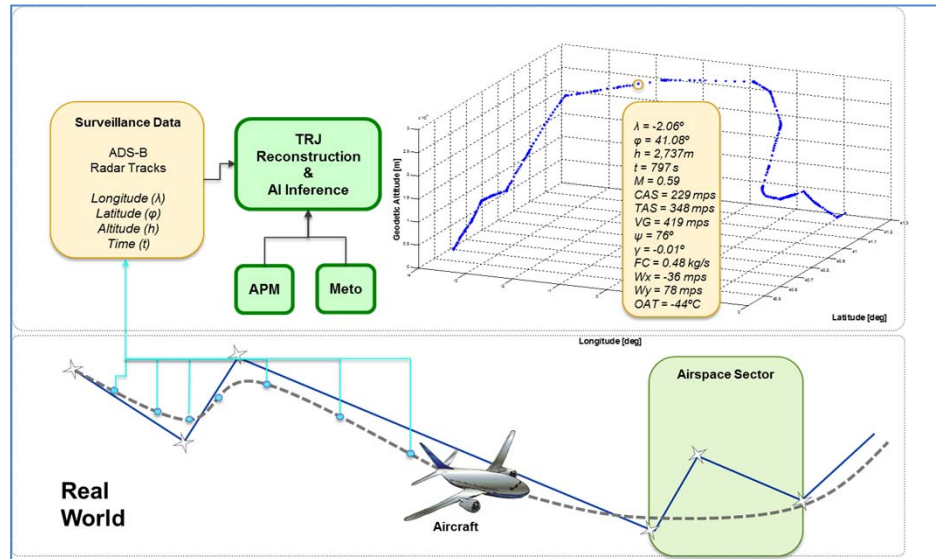
Data ingestion and feature extraction



- **Surveillance Data.** Radar tracks of the Spanish airspace controlled by EnAire, the Spanish Air Navigation Service Provider (ANSP).
- **Weather Data.** Forecasts (downloaded from NOAA in grib format), Significant Meteorological Information (SIGMET), Meteorological Aerodrome Report (METAR) and Terminal Aerodrome Forecast (TAF).
- **Flight Plans.** Standard dataset generated by Airspace Users (AU) and agreed with the ANSPs, that represents an intended flight or portion of a flight. The FPs considered within DART are those stored in the Spanish ATC operational system, and include all flight plan amendments associated to the originally filed FP (GIPV from SACTA).
- **Airspace structure.** The airspace is organized in accordance with the envisioned traffic flow and the availability of resources to manage that traffic. Includes both possible and applied sector configurations
- **Re-constructed trajectory.** Extended trajectory information that includes additional aircraft state variables that are not included in the surveillance datasets (e.g., airspeeds, mass, and the like) with higher data sampling.
- **Aircraft Intent Description.** Semantic description of a trajectory that represents the set of instructions to be executed by the aircraft in order to realize its intended trajectory, equivalent to the commands issued by the pilot or the FMS to steer the aircraft.

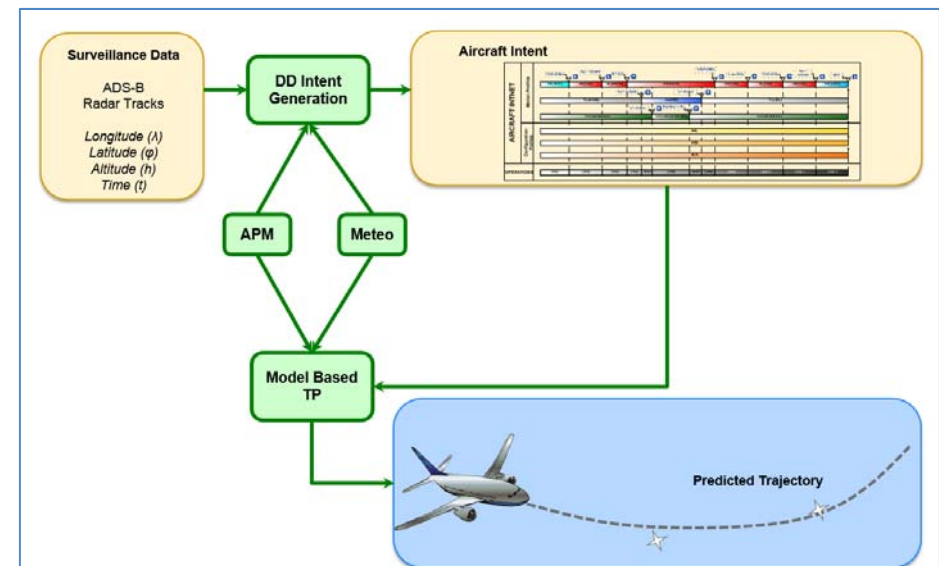
Single Trajectory Prediction

Feature Extraction

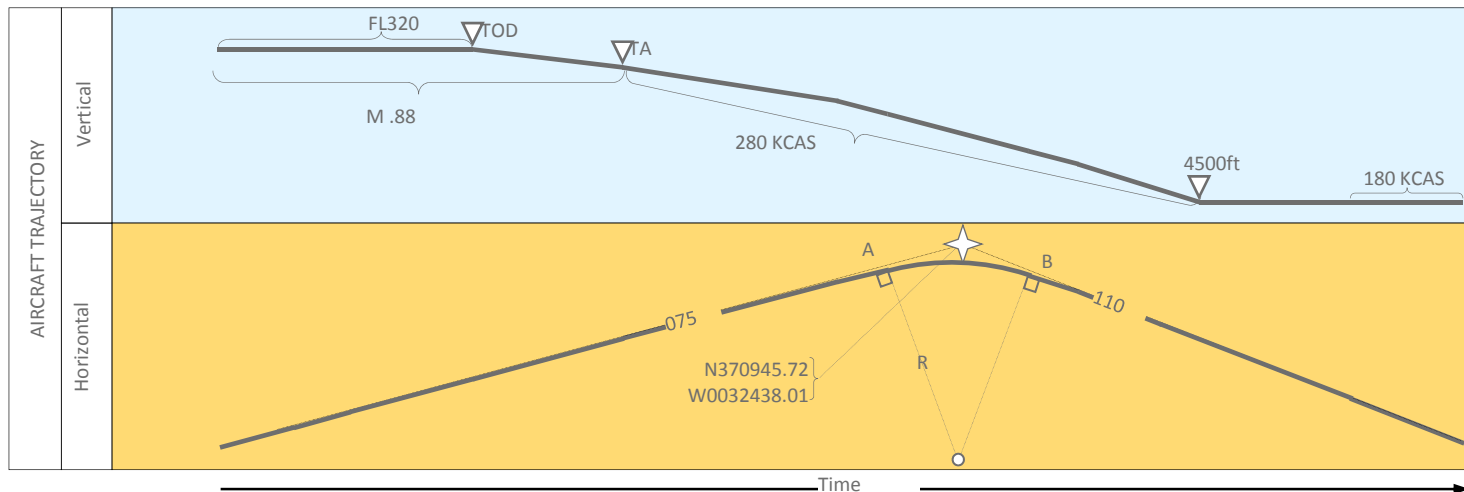
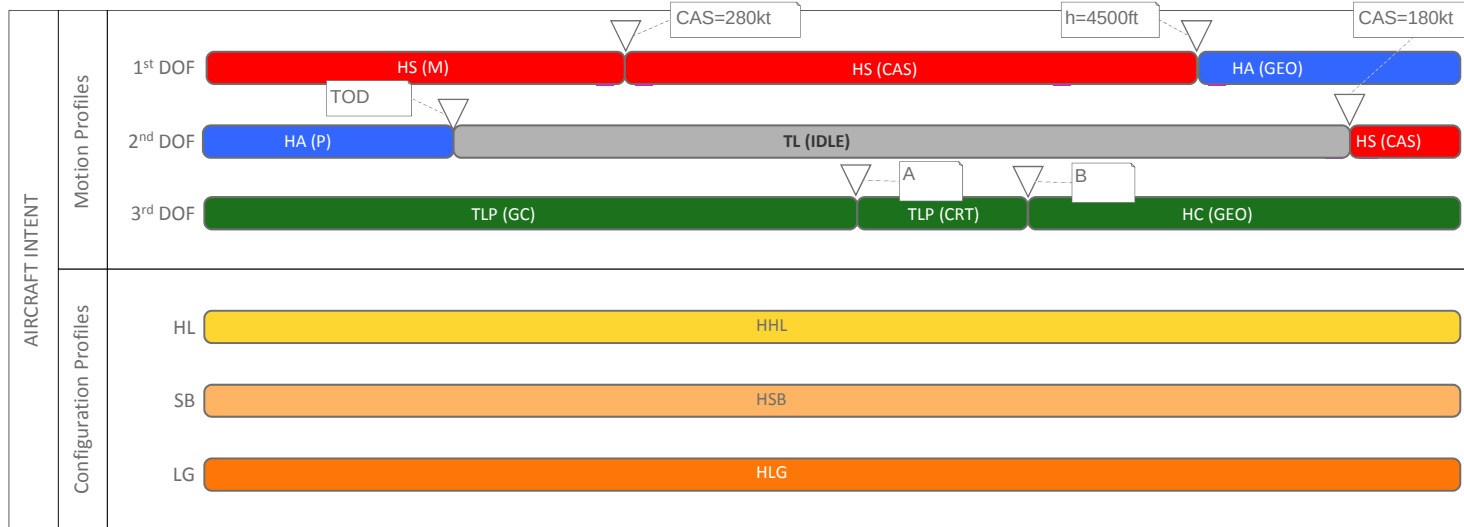


Re-constructed trajectory. Extended trajectory information that includes additional aircraft state variables that are not included in the surveillance datasets (e.g., airspeeds, mass, weather conditions, ...) with higher data sampling.

Aircraft Intent Description. Semantic description of a trajectory that represents the set of instructions to be executed by the aircraft in order to realize its intended trajectory, equivalent to the commands issued by the pilot or the FMS to steer the aircraft.



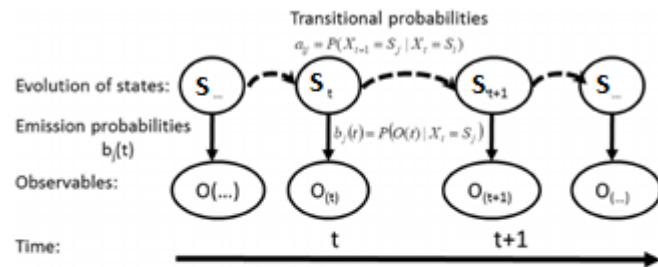
Aircraft Intent Example



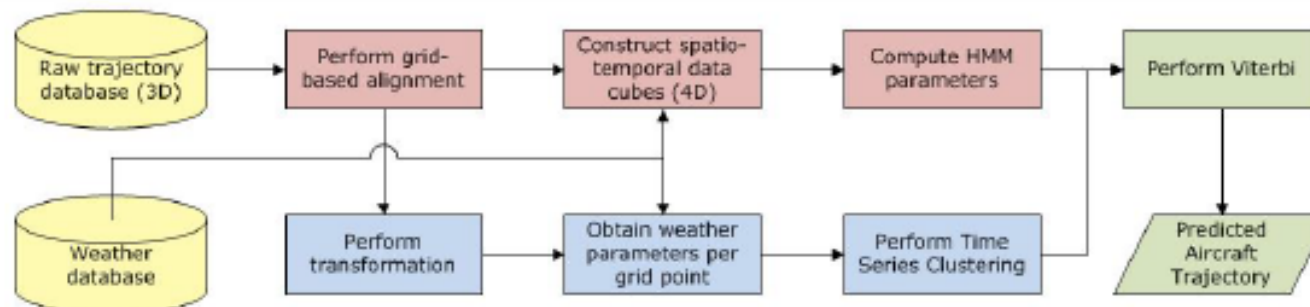
Single Trajectory Prediction

Hidden Markov Models

Given a set of historical raw or **reconstructed trajectories** for specific aircraft types along with pertinent historical weather observations, we aim at learning a model that reveals the correlation between weather conditions and aircraft positions and predicts trajectories in the form of a time series.



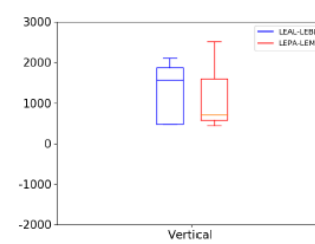
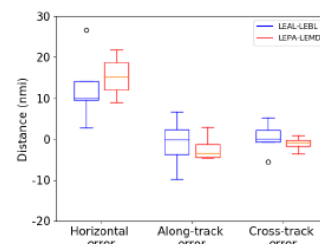
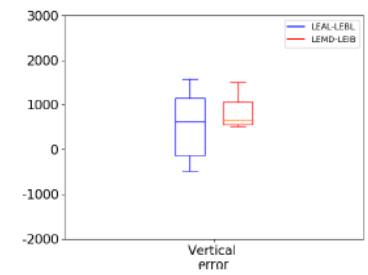
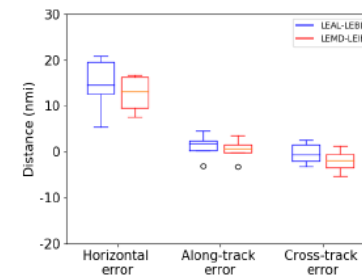
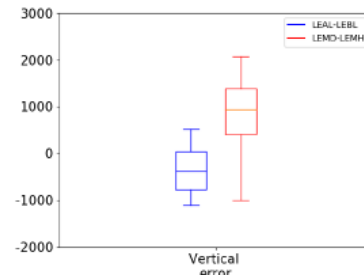
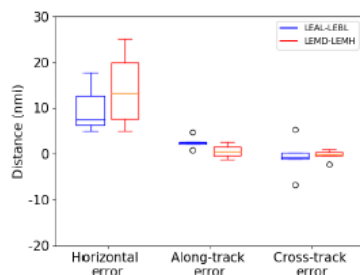
- States $S = \{S_1, S_2, \dots, S_K\}$ are represented by reference points' coordinates (*latitude, longitude, altitude*) that form aligned trajectories.
- Transition probabilities $A = \{a_{ij}\}$, $1 \leq i, j \leq K$, i.e. a_{ij} is the probability of an aircraft discretely transitioning from one state S_i to another S_j along its aligned trajectory, $\bar{T}$.
- Emission probabilities $B = \{b_i(o)\}$, $1 \leq i \leq K$ is the probability of discrete weather parameters having been observed at a specific state, S_i .
- Initial probabilities $\pi = \{\pi_i\}$, $1 \leq i \leq K$ is the probability of an aligned trajectory beginning at a specific state, S_i .



Single Trajectory Prediction Hidden Markov Models



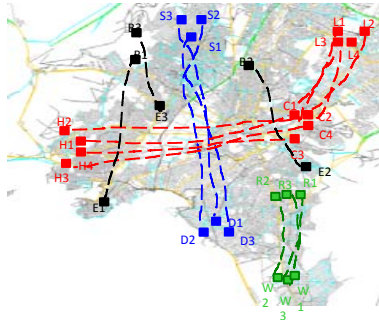
Case	Route1	TrainingSetSize		TestSetSize		Route2	TrainingSetSize		TestSetSize	
		#trjs	#pts	#trjs	#pts		#trjs	#pts	#trjs	#pts
1	LEAL-LEBL	1118	55116	200	9860	LEMD-LEIB	2572	125623	200	9769
2	LEAL-LEBL	1118	55116	19	937	LEMD-LEMH	1056	68141	19	1226
3	LEAL-LEBL	1118	55116	152	7493	LEPA-LEMD	5116	306128	152	9095
4	LEBL-LEMG	1704	127451	43	3216	LEMD-LEMH	1056	68141	43	2775
5	LEBL-LEMG	1704	127451	180	13463	LEPA-LEMD	5116	306128	180	10771
6	LEBL-LEZL	2404	183343	41	3127	LEMD-LEAM	1434	70128	41	2005
7	LEBL-LEZL	2404	183343	46	3508	LEMD-LEMH	1056	68141	46	2968
8	LEBL-LEZL	2404	183343	164	12508	LEMG-LEMD	1403	75408	164	8815
9	LEBL-LEZL	2404	183343	210	16016	LEPA-LEMD	5116	306128	210	12566
10	LEIB-LEBL	1360	53443	259	10178	LEPA-LEMD	5116	306128	259	15498
11	LEIB-LEBL	1360	53443	158	6209	LEPA-LEVC	1426	50467	158	5592
12	LEMG-LEBL	1563	114767	38	2790	LEMD-LEAM	1434	70128	38	1858
13	LEMG-LEBL	1563	114767	46	3378	LEMD-LEIB	2572	125623	46	2247
14	LEZL-LEBL	2380	186299	40	3131	LEMD-LEIB	2572	125623	40	1954



Single Trajectory Prediction

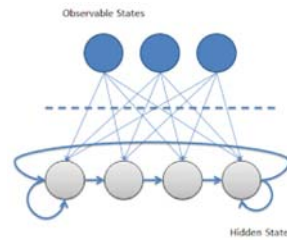
Clustering + HMM

1st Step: Clustering semantic trajectories



“Annotated” Trajectories (FP, weather,...)
Clustering with ad-hoc distance functions (not just spatio-temporal but weather, date, etc...)

2nd Step: For each cluster train a HMM

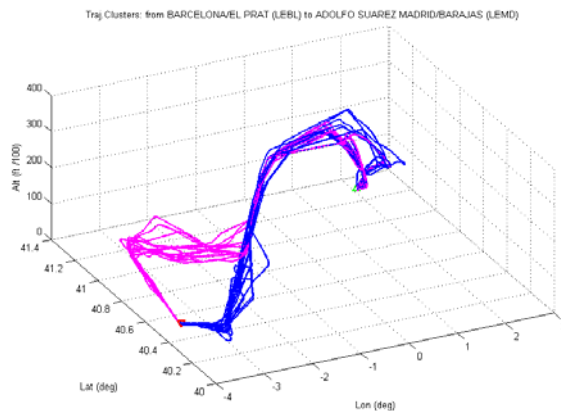


Non-uniform graph-based spatial grid
FP Waypoints are used as reference for HMM states

3rd Step: (Filter) Given a flight plan Q find top-k most probable HMM models

4th Step: (Refine) Similarity search among the semantic trajectories that belong to the top-k HMMs

Waypoint-to-waypoint matching to medoids
3-D deviation (Haversine distance)



Semantic distance metric (points):

$$D_{LS}(r_i, r_j) = \lambda \cdot \text{dist}_e(r_i, r_j) + (1 - \lambda) \cdot \text{dist}_k(r_i, r_j)$$

$$\text{dist}_e(r_i, r_j) = \frac{\sqrt{w_1(x_i - x_j)^2 + w_1(y_i - y_j)^2 + w_1(z_i - z_j)^2 + \frac{w_2}{w_1}(t_i - t_j)^2}}{\max \text{EuclideanDistance (STD)}}$$

$$\text{dist}_v(r_i, r_j) = 1 - \frac{v_i \cdot v_j}{\|v_i\|^2 + \|v_j\|^2 - v_i \cdot v_j}$$

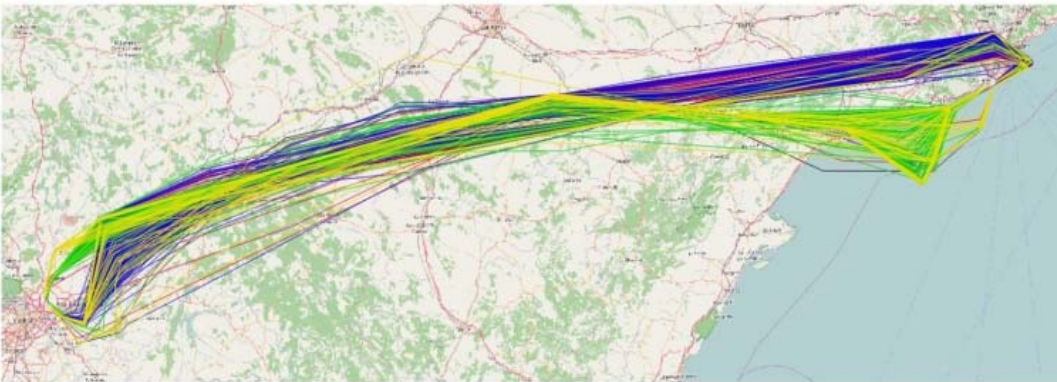
Semantic trajectory distance:

$$D_R(R_i, R_j) = \lambda \cdot D_R^{st}(R_i, R_j) + (1 - \lambda) \cdot D_R^v(R_i, R_j)$$

Single Trajectory Prediction

Clustering + HMM

Using the formulation above, this two-phase hybrid clustering/HMM approach was tested in a benchmark dataset of actual flight trajectories (around 1400 flights). One airport pair was considered from the Spain airspace (Barcelona/Madrid) and each direction was modeled separately, as it involves different flight plans and takeoff/landing approaches.



Example of four main clusters (colored) and one cluster of noise & outliers (black) produced in the clustering phase upon the RT (actual routes) using the EDR semantic-aware similarity metric.

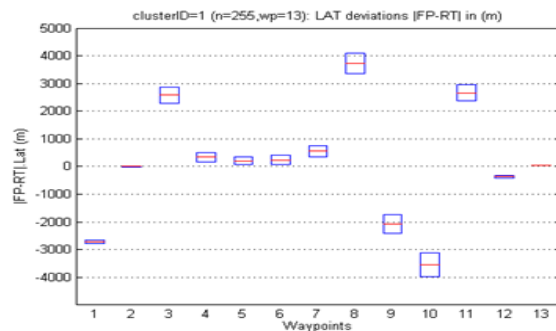
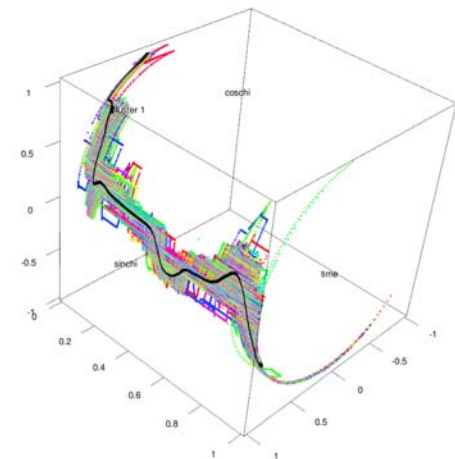
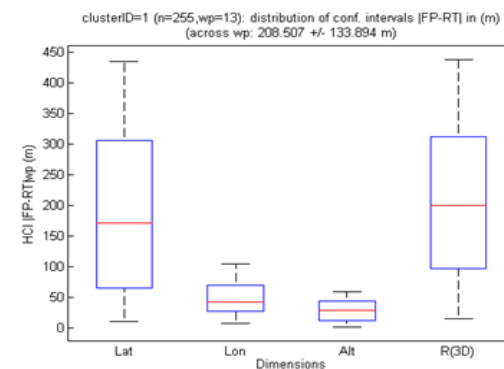


Figure illustrates the per-waypoint means and confidence intervals for Latitude in cluster 1 as described above. The height of each bounding box is directly linked to the uncertainty associated with producing the maximum-likelihood deviation from the HMM emissions in each reference waypoint, i.e., the difference between the flight plan and the aircraft actual route.



Bearing clustering, represented in t , $\cos(\chi)$, $\sin(\chi)$



The height of each box, i.e., the size two central quartiles, is directly linked to the statistical uncertainty in predicting each dimension of the pair-wise deviations between flight plans and the cluster medoid.

Collaborative Trajectory Prediction



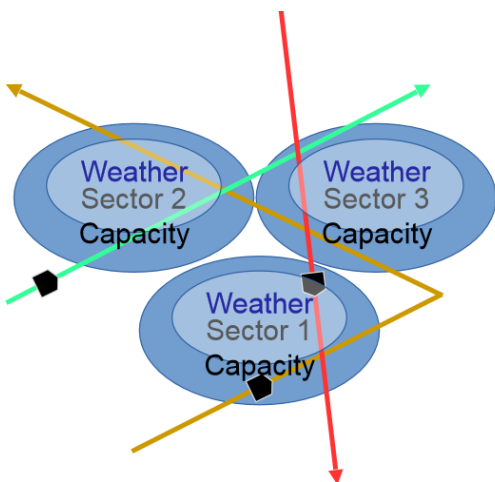
Scope: This scenario objective is to demonstrate how DART predictive analytics capability can help in trajectory forecasting when demand exceeds capacity (from a global perspective), at **planning phase** (pre-tactical).

D>C

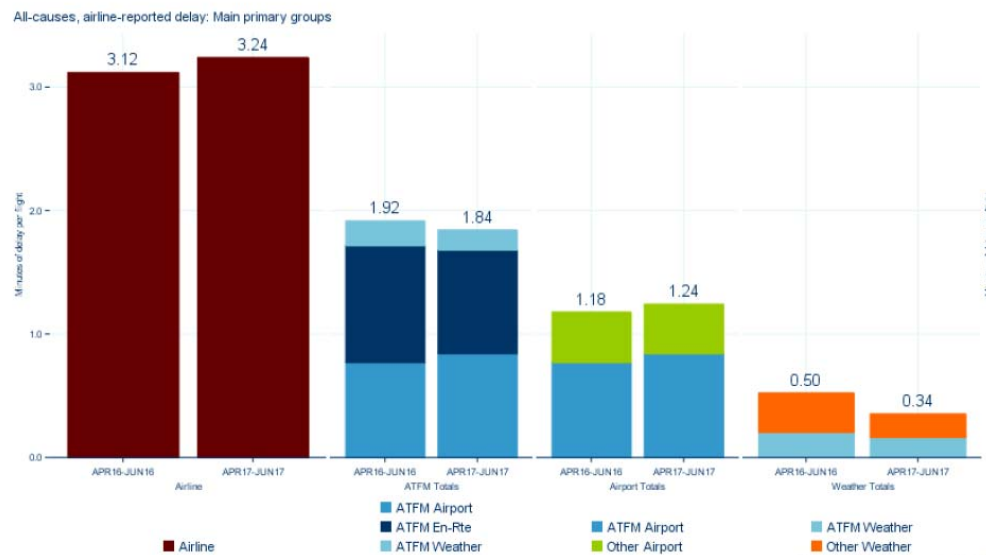
- System capacity is not enough
- Some flights must be delayed → regulation
- Delays are expensive and problematic

Measures will be applied to the WP2 trajectories due to the imbalance between demand and capacity

Goal: Improve global predictability (*relying on accurate planning information*)



Primary Delay Causes Q2 2016 vs. Q2 2017



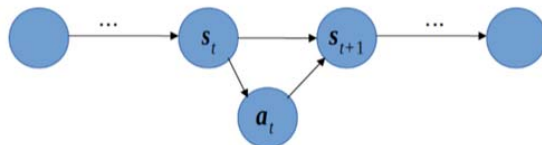
Collaborative Trajectory Prediction

Approach: Formulate a Markov Decision Process

- Solving MDP = planning
- Reinforcement Learning methods are considered appropriate
- Multi-agent RL approach inherently appropriate
- MDP requires a Reward Model
 - Reward functions take into account participation of trajectories to hotspots and delays imposed - later results consider AU's preferences in terms of strategic delay cost, as well.
- *Other options resulted in a huge state-action space (i.e. keeping locations, or action = next target location)*



- State vector:
 - $d_t \in \mathbb{N}$ Imposed delays so far (1 per flight)
 - No need to keep locations in states
 - Time is factored out
- Actions per agent: 0 or 1 (**while on ground**) UNTIL a MAX delay is reached.
 - Available options limited by flight plan
- OR:
- Actions per agent: 0 ... MAX delay
 - Available options limited by flight plan



Reward Model

- Capacity: c = capacity of sector
- Demand: $\text{demand}(l, j)$ = number of flights in time slot j
- DCB violations: $\text{hotspot}(c, l, j) = 1$ if $\text{demand}(l, j) > c$, 0 otherwise

Participation of flights in A in hotspots

Delays imposed and AU's preferences

$$Rwd_A(s_A^t, str_A^t) = \lambda_1 * C(str_A^t, s_A^t) + \lambda_2 * D(str_A^t, s_A^t) \quad (1)$$

C: A function that takes into account the number of hotspots and the "contribution" of flights in them (in terms of duration of being involved in a hotspot)

D: A function that depends only on the delays imposed to flights: Currently this is translated into strategic delay cost.

str: The strategy of agents - i.e. their chosen delay.

This function aims to reduce hotspots (via the minimization of flights contribution to delays) and delays (costs due to delays) imposed to flights

Collaborative Trajectory Prediction



Reinforcement Learning approach considered (in progress):

1. Independent Learners approach (*Ind-Colab-RL*)

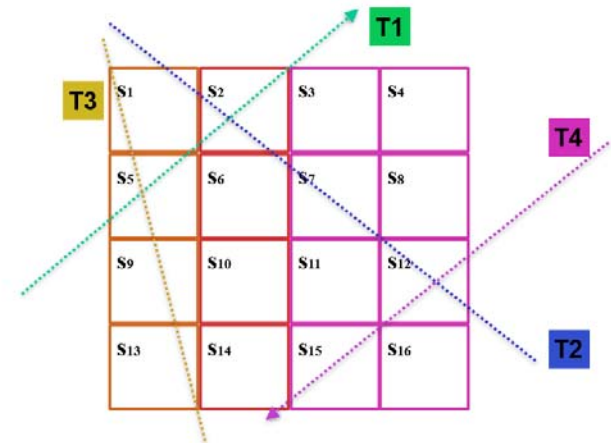
Each agent (flight) is self-interested and learns by itself to resolve the DCB problem, by taking into account own state and measuring its own reward after each decision.

2. Sparse Collaborative Q-Learning – Agent-based decomposition – Edge based update (*Ed-Colab-RL*)

This is a variant of the sparse cooperative edge-based Q-learning method. Multiple agents are jointly interacting with the environment and decide by taking into account joint state and measuring individual reward after each joint decision.

3. Sparse Collaborative Q-Learning – Agent-based decomposition – Agent based update (*Ag-Colab-RL*)

This is a variant of the agent-based update sparse cooperative edge-based Q-learning method that allows agents to share their joint reward after joint decision.



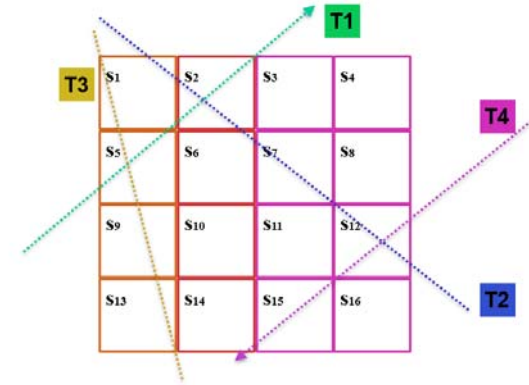
Collaborative Trajectory Prediction



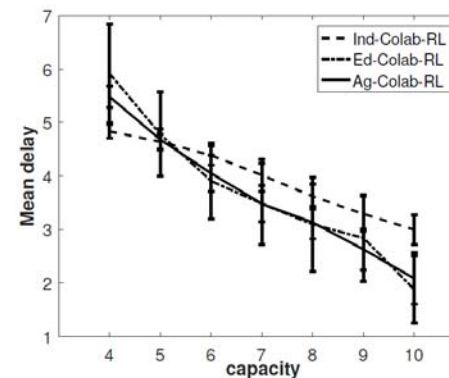
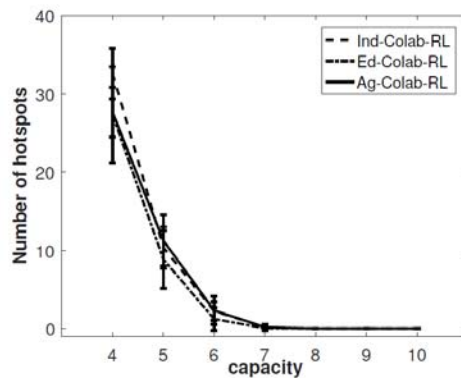
Experiments' set up

Table 1. Parameter values used during the simulated experiment

Parameter	value
grid structure of sectors	4×4
capacity of sectors, C	$\in [4, 10]$
number of planes, N	100
Duration and Step of Occupancy Counting Period	6
total time period duration H	180
maximum delay	10



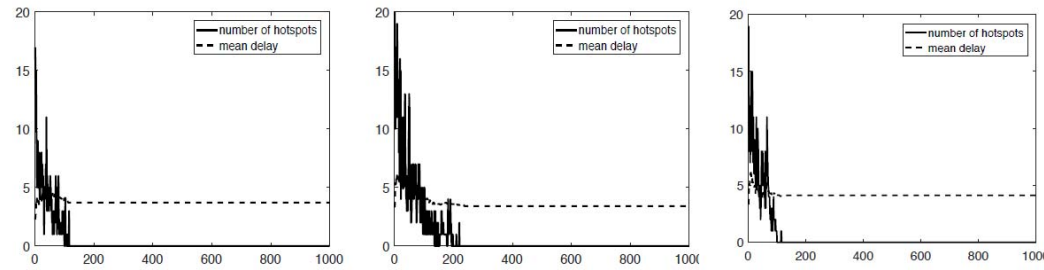
Experimental Results: Results achieved for different sectors' capacities



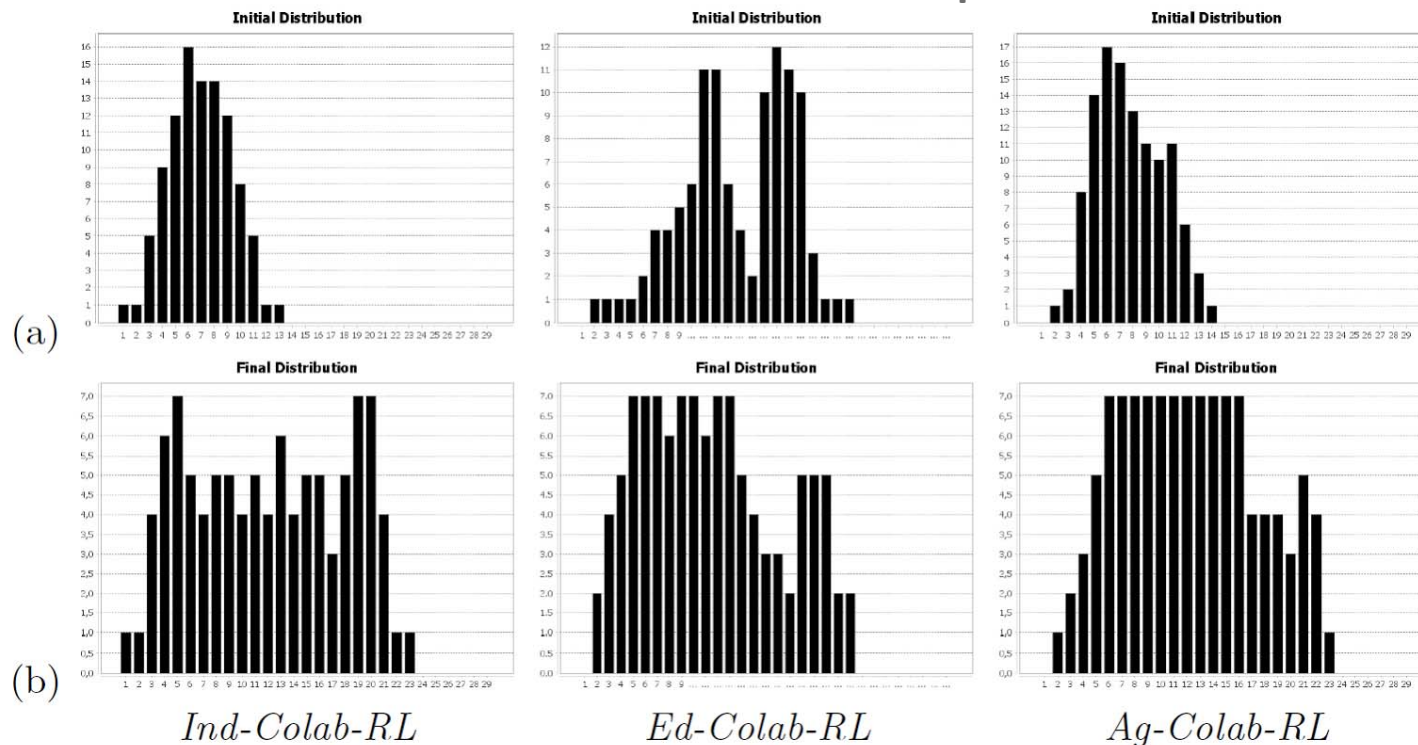
Collaborative Trajectory Prediction



Experimental Results: Learning curves showing convergence to solution.



Independent Learners Edge-Based Learners Agent-Based Learners
Experimental Results: Results shown final demand to periods in sectors



Collaborative Trajectory Prediction



Experimental Results: Highlights

- The results confirm that the proposed multi-agent formulation provides a promising framework for tackling the DCB problem.
- All methods demonstrated very similar behavior by eradicating hotspots with Edge Based Update being slightly more effective compared to others in terms of the number of hotspots and mean delay achieved, but less efficient than Agent Based Update in terms of convergence speed.

Latest Experimental Results on Real-World Scenario

- A scenario has been drafted using real world data for the 23d of November 2016.
- Delay cost of the Flights has been added to the reward function.
- New deterministic rule (+DR) that facilitates the exploration, by pruning the state space has been applied. The deterministic rule allows flights NOT participating in hotspots to get delay equal to 0, this reduced the search space and thus increased methods computational efficiency.

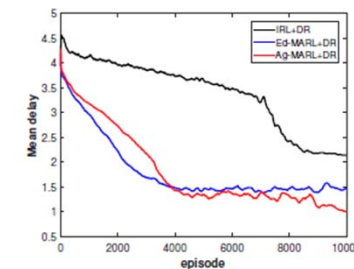
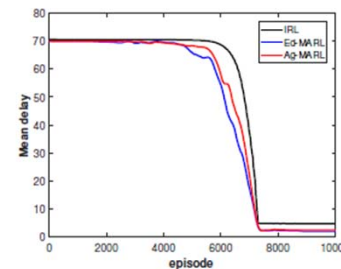
Collaborative Trajectory Prediction



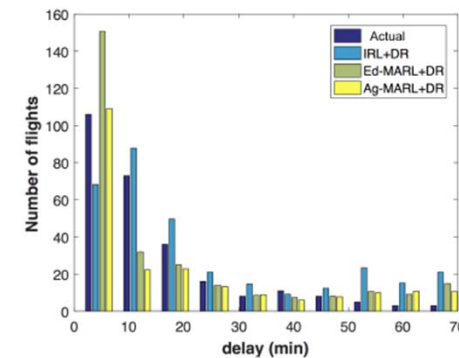
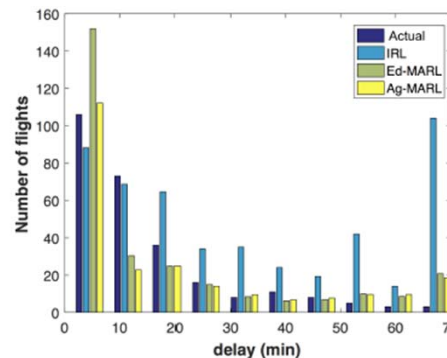
Latest Experimental Results on Real-World Scenario

Method	# flights delayed	Total delay	Mean delay (delayed flights)	Mean delay (all flights)
Ind-Colab-RL	322	7871	24	2.14
Ind-Colab-RL + DR	473	16589	34	4.51
Ed-Colab-RL	280	4668	16	1.27
Ed-Colab-RL + DR	310	6866	22	1.87
Ag-Colab-RL	160	3743	23	1.02
Ag-Colab-RL + DR	271	8284	30	2.54
Actual	269	4130	16	1.12

Learning curves
Showing convergence to
zero hotspots(left) and
mean-delay (right)



Distribution of delay(mins)
to flights achieved by RL
methods compared
to the actual ones.



Collaborative Trajectory Prediction



Latest Results from real world Scenario

- The methods manage to provide solutions to the DCB problem, imposing delays that result to zero hotspots.
- The new deterministic rule considerably increases the methods' performance.
- Agent Based Update is more effective in terms of the number of flights delayed, also compared to the actual delays.



DART: A Machine-Learning Approach to Trajectory Prediction and Demand-Capacity Balancing

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]



Founding Members



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.