Numerical and temporal planning for a multi-agent team acting in the real world

Davide Dell'Anna

University of Turin

AIRO @ AI*IA 2016
November 28, 2016

Introduction

Designing a robotic architecture: the deliberation layer

In real-world planning and acting are strictly connected (Ghallab, 2014).
Some important points of connection:

- Resources consumption
- Time
- Concurrency

Classical planning is not enough for real world problems.
Need of extensions to classical planning:

- MILP (Richards, 2002), (Bellingham, 2002)
- SMT (Shin, 2005), (Moura, 2008)
- Time Line planning (Mayer, 2014), (Ghallab, 1994), (Donati, 2008), (Barreiro, 2012)
- PDDL extensions (Fox, 2003) (Edelkamp, 2004)
- ...

Introduction

The challenges

Which approach to choose for heterogeneous multi-agent teams acting in the real world?

What are the main features of this class of problems?

How to encode them in action-based planning languages and which planning model is more appropriate?

What about scalability of the state-of-art planners for real-world numerical and temporal constraints?

Outline

The class of problems

The test-bed: SMAT (Advanced Monitoring System of the Territory, leaded by Alenia Aermacchi) (Boccalatte, 2013)

- Heterogeneous multi-UAV (MALE and MAME types) missions
- Mission temporal bounds and global duration constraints
- User-defined requests of observation of targets in specific temporal windows and via specific sensor-suite
- Temporal constraints between different observations
- Central Control Station: a centralized approach to high-level multi-UAV planning

Starting from this test-bed formal definition and generalization of a class of multi-agents problems.

The class of problems

The UAV class of problems (Class of problems UAV)

The main problem

UAVs must continuously perform some task.

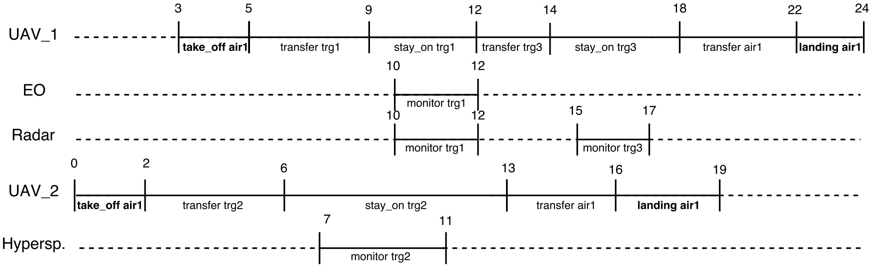


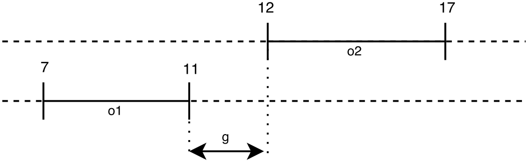
Figure : A graphical example of plan for a multi-UAV problem involving five agents (*UAV_1*, *UAV_2* and *EO*, *Hyperspectral* and *Radar* sensors) and requiring four observations of three different targets. For target *trg1* an observation involving two sensors together is required in order to perform a data fusion operation.

The class of problems

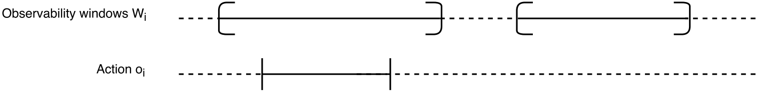
Temporal constraints

Temporal constraints between different targets observations:

- *Before*(o_1, o_2)



Targets observation in user-defined temporal windows:



Outline

The system

Input

Observation Requests[1] and time windows constraints

OR	Date	Target	Type	Sensor	Obs. Win.	MinDur
1	03/08/16	Centrale Trino	Point	EO	8.00 - 13.00	5m
2	03/08/16	Centrale Trino	Point	Radar	8.00 - 13.00	5m
3	03/08/16	TangTO	LOC	EO	7.40 - 9.00	NA
4	03/08/16	A4 (TO-NO)	LOC	EO	7.40 - 9.00	NA
5	03/08/16	A21(TO-Tortona)	LOC	EO	7.40 - 9.00	NA

MultiObs constraints

Equal(1, 2): observation requests $1 - 2$ must be observed together (request of data fusion).

Mission temporal constraints

The mission must be performed between the 6 a.m. and 12 a.m. The maximum duration of the mission is 12000 seconds.

[1] Geographical information has been omitted for the sake of simplicity .

The system

Output[2]

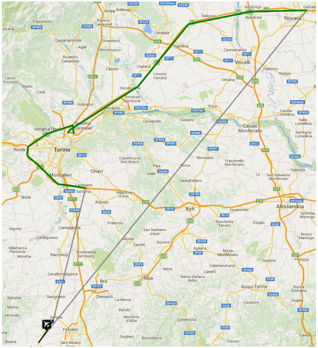


Figure : MAME N4.

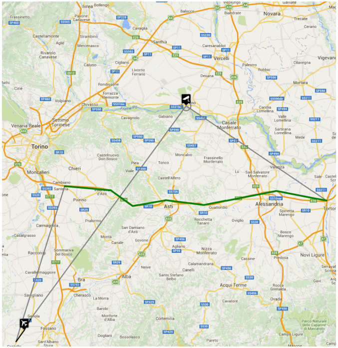


Figure : MALE N7

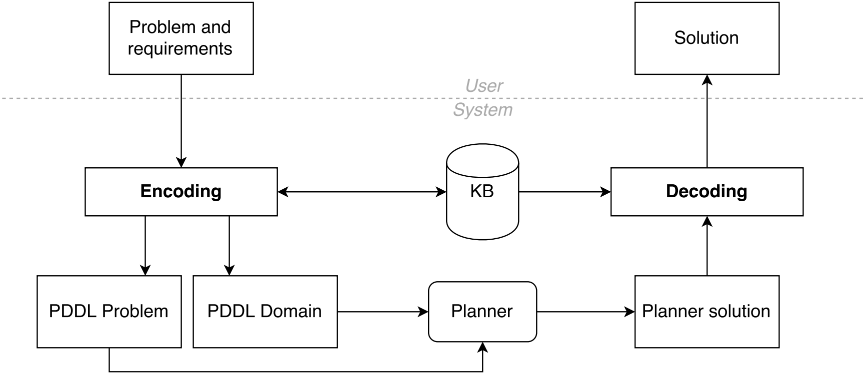
Assignments

OR 3 and 4 (TangTO and A4(To-No)) to UAV MAME N4

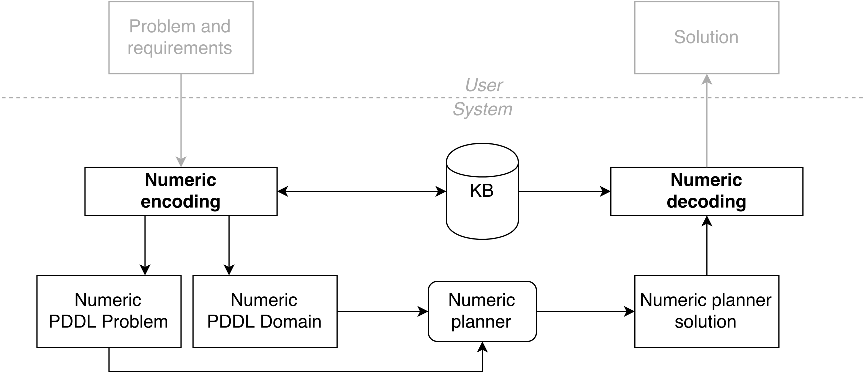
OR 1, 2 and 5 ((CentraleTrino) and A21(To-Tortona)) to UAV MALE N7.

[2] The output was obtained by using COLIN planner in 3 sec. on a machine equipped with SO Linux Mint 12 64bit, Intel Core i3-2367M CPU@ 1.40GHz x 4, 4GB RAM

The system

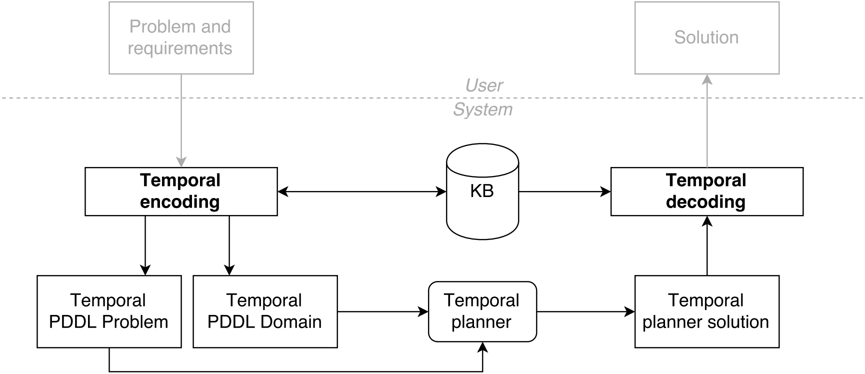

Problem and requirements
Solution
User
System
Encoding
KB
Decoding
PDDL Problem
PDDL Domain
Planner
Planner solution

The system



Problem and requirements
Solution
User
System
Numeric encoding
KB
Numeric decoding
Numeric PDDL Problem
Numeric PDDL Domain
Numeric planner
Numeric planner solution

The system



Problem and requirements
Solution
User
System
Temporal encoding
KB
Temporal decoding
Temporal PDDL Problem
Temporal PDDL Domain
Temporal planner
Temporal planner solution

The system

Numerical and temporal planning models

	Numerical	Temporal
Time	**simulated and discretized** via numerical fluents + waiting actions	**continuous and automatically handled** (timed initial literals + durative actions)
Concurrency	**simulated**	**automatically handled**
Actions contiguity	**guaranteed** (easy encoding)	**forced** (non-sequential plans)
Time windows	expressed via numerical fluents	timed initial literals + propositional fluents
ObsReq constraints	require additional support actions	easily expressed via durative actions
Temporal metric	numerical fluents	only total-time

Table : A comparison between the encoding capabilities of numerical and temporal planning models.

Outline

Experimental evaluation

Goal

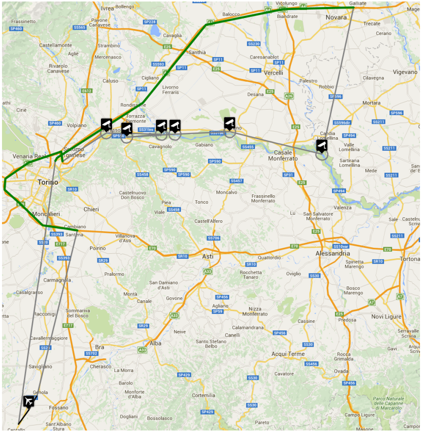
Comparison between numerical and temporal models, in terms of:

- Planners performances
    - Coverage
    - Plan quality
    - Solving time
- Complexity of class of problems
    - Model behavior with different types of constraints

Models tested on both realistic scenarios emerged from the SMAT project and on synthetic problems automatically generated.

Experimental evaluation

Real-world scenarios - A complex single agent mission



- 1 UAV
- 8 targets (Point and LOC)
- 9 observations requests
- 8 time windows of interest
- 2 before constraints
- 1 data-fusion request

Numerical model: first solution (0.23 sec), optimal solution (1 sec.)
Temporal model: no solution (timeout 180 sec.)

Experimental evaluation

Classes of problems

Three main classes of examples based on the dimensionality of problems in terms of number of UAVs and targets involved:

- **Class 2U6T**: two UAVs and six targets.
- **Class 3U8T**: three UAVs and eight targets.
- **Class 4U10T**: four UAVs and ten targets.

Each class is characterized by three features:

- **Assignments**: observations are a priori assigned to UAVs or not
- **Multiobs**: temporal constraints between different observations are specified (up to 4 different constraints).
- **Windows**: end-user requests that observations are performed within specific temporal windows.

Experimental evaluation

Dataset

10 randomly generated different problems for each possible combination of values of the features of the classes of examples.

Dataset of **600** different problems.

4 numerical planners: COLIN, POPF2, Metric-FF and LPG.
3 temporal planners: COLIN, POPF2 and TFD.

Timeout of 180 seconds for every problem[3].

4200 different results: 2400 of numerical problems and 1800 temporal ones.

[3] All planning was executed on a machine equipped with SO Linux Mint 12 64bit, Intel Core i3-2367M CPU@ 1.40GHz x 4, 4GB RAM.

Experimental evaluation

Models comparison

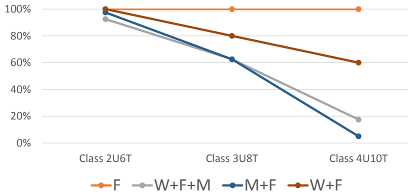


Figure : Numerical planning.

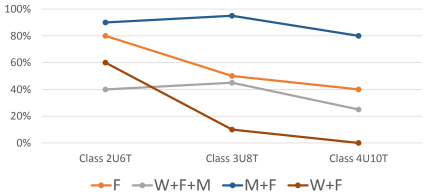


Figure : Temporal planning.

Two line charts displaying the rate of problems solved by COLIN w.r.t. the set of constraints involved.

Outline

Conclusions

Contribution

- Formal definition of a typical real-world multi-agent class of problems.
- Modeling of problems in both numerical and temporal planning formalisms.
- Analysis of the impact of different types of constraints and features on problems complexity and planning models.
- Evaluation of state-of-art planners.

Conclusions

Conclusions and future works

- Numerical model performed better, but no clear winner
- Complementarity of numerical and temporal planning models
- Action-based approach promising in many relevant real life problems with few (and not strict) temporal constraints

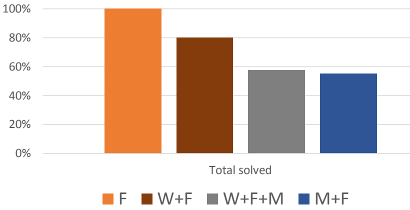


Figure : Numerical planning.

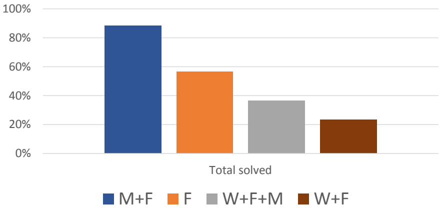


Figure : Temporal planning.

Future works:

- Analysis of behaviors of the two planning models on different domains.
- Deeper comparison with different state-of-art planning approaches

Thank you for your attention.