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

Davide Dell'Anna

Università di Torino, Dipartimento di Informatica.
dellanna@di.unito.it

Introduction. Automated planning is a central area of Artificial Intelligence which aims to design a powerful deliberation layer for autonomous intelligent systems. Autonomy of intelligent systems doesn't concern only planning and deliberation but also acting. These two aspects are not completely disjoint: actors may deliberate or plan both before and during acting in order to perform intelligent executions, and deliberation may be strictly influenced by acting details ([1]). The gap between planning and execution is one of the main problems to face in building an autonomous system and in order to successfully do it, planning should capture important features of real world domains ([2]). In particular, when problems involve teams of (possibly heterogeneous) agents which must cooperate, relevant features to take into account are cooperation, consumable resources, continuous numeric change, as well as concurrency, time and temporal constraints. In recent years, automatic planning languages have been extended with primitives allowing to express numerical and temporal aspects of problems (e.g. PDDL 2.1 and PDDL 2.2 ([3] and [4])). In this way, relevant aspects for execution can be taken into account already at planning time. Many alternative approaches to action-based planning have been developed. In particular timeline-based approaches (e.g. [5], [6] or the mission planning frameworks [7], [8]) or MILP approaches (e.g. [9], [10]).

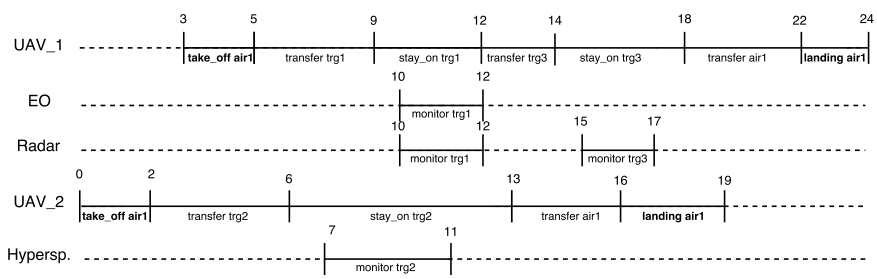
Defining the problem. In this work we aim at showing how complex real-world multi-agent[1] problems involving consumable resources, continuous numeric change, time and multi-agent coordination can be faced with action-based approaches such as numerical and temporal planning by employing state-of-art general purpose planners.

We developed a complex software architecture oriented to a centralized off-line planning system. This approach to multi-agent planning is justified by the many interactions required among agents, by the expensive coordination activities in heterogeneous teams and by optimization reasons (i.e. maximizing the number of tasks assigned to the agents and minimizing the number of agents employed). These factors make more appropriate a centralized approach than a distributed one. However, we left the low-level controls (e.g. sensor pointing,

[1] We focused our attention on multi-UAV mission, leveraging the experience and knowledge acquired on UAVs (Unmanned Aerial Vehicles), and in particular on MALE (Medium Altitude Long Endurance) and MAME (Medium Altitude Medium Endurance) UAVs, by participating to industrial research project SMAT [11], coordinated by Alenia Aermacchi. However the solutions found can be easily adapted for other robotic domains and the results that can be achieved are similar.

agents' movements, etc.) to single agents which have specific on-line acting and deliberative capabilities (intelligent monitoring, re-planning or continual planning, [12]).

We addressed a class of real-world aerospace planning problems involving teams of heterogeneous UAVs whose objective is to observe a set of targets (POINT-targets or LOC-target, i.e. polygonal chains defined by a sequence of vertices, such as portion of a river or of a motorway). Targets observation must meet a series of user requirements which specify the type of sensor to use and temporal constraints. UAVs configuration depends on logistic information which express both the suite of sensors and the temporal windows of availability of the vehicles. Therefore, w.r.t. a specific temporal window, each UAV is able to observe a subset of the set of mission's targets according to its configuration and the sensors required for the observation. Since targets observations are not a priori assigned to agents, the planning system must also autonomously decide which agents employ to perform observations, possibly minimizing the number of agents employed and the global duration of the mission. The following figure reports a graphical example of plan automatically synthesized by the planning system and involving two UAVs 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 nontrivial class of problems introduced above can be defined as $Class\_U = \langle UAV, OR, C \rangle$, where UAV is a set of UAVs a priori configured with suites of sensors, OR is a set of user requests of observations of targets (specifying the minimum duration of observation of a target and the sensors to use) and C is a set of constraints related to the mission (e.g. temporal constraints on global mission duration, UAVs resource constraints and initial position, etc.). We extended this class with two additional types of temporal goals: a set of constraints M among different targets observations (e.g. `trg1` must be observed `BEFORE` `trg2`) and a set of constraints W which specifies the time windows of observability of targets (e.g. `trg1` must be observed between `9 a.m.` and `12 a.m.`).

Encoding and decoding. Temporal aspects and numerical resources play a central role in this class of problems. In this work we studied how to encode the described class of problems by adopting two different PDDL-based approaches: a temporal approach, employing durative actions and timed-initial-literals and exploiting the temporal planners capability to automatically handle temporal aspects of planning (e.g. actions' temporal location and duration and concurrency), and a numerical approach, in which time is treated as a numerical fluent and therefore time passing and concurrency must be simulated (an approach similar to [13]).

Encoding isn't a trivial process and, despite the many knowledge engineering solutions proposed over the years ([14]), it is not always possible a direct translation from the user requirements into PDDL. Problem constraints and requirements in fact deeply influence the set of necessary fluents and the preconditions and effects of actions. For instance the encoding of a request of observation of a *Target T2, between 9 a.m. and 12 a.m., with an EO sensor, for at least 60 sec., after the observation of Target T1* impacts on initial state of the problem (initializing fluents, timed-initial-literals and predicates stating the minimum duration of observation, the sensor required, the target's observability, etc.), on actions schema definition (preconditions and effects of actions, and definition of additional actions, e.g. actions enabling the observation of targets) and also on goals (requiring the observation of the target). Furthermore the encoding phase must also introduce, supported by an internal knowledge base, all the information that are necessary to correctly define the domain and the problem and that are beyond the user interest and knowledge (e.g. the UAVs' initial position, cruise speed, consumption rate, the suite of sensors loaded on board of UAVs, the target geometries, their location in the environment, etc.).

It is worth noting that in addition to encoding, a significant decoding phase is essential to provide exhaustive and meaningful information to the user, especially in numerical (but also in temporal) planning. Numerical planners, in fact, don't provide any relevant temporal information about the scheduled actions. Therefore it is necessary a complex decoding procedure which, for each agent, simulates the execution of the actions.[2].

Experimental results. We performed our tests on both synthetic problems and real-world multi-UAV multi-target planning scenarios. We automatically generated a dataset of 600 different problems encoded in both numerical and temporal formalism. We defined three main classes of examples based on the dimensionality of problems in terms of number of UAVs and targets involved: **Class 2U6T** involving two UAVs and six targets, **Class 3U8T** and **Class 4U10T**. Problems were then fed to four numeric planners (COLIN [16], POPF2 [17], Metric-FF [18] and LPG [19]) and three temporal planners (Colin, POPF2 and TFD [20]) with a timeout of 180 seconds for every problem[3].

Preliminary results shown very different degrees of scalability, therefore we considered more accurate and different classes of problems.[4] The system easily handles the increase of the number of UAVs and target involved in problems when few (or no) temporal constraints are involved. See for instance column **U** for numerical planning or column **U+M** for temporal planning. Conversely,

[2] A more extensive description of encoding and decoding solutions, as well as a more detailed experimental validation, are available in [15]

[3] The machine employed for the experiments was equipped with SO Linux Mint 12 64bit, Intel Core i3-2367M CPU@ 1.40GHz x 4, 4GB di RAM.

[4] Results reported concerns only the planner COLIN which behaved on average better than others and is able to perform both numerical and temporal planning.

the introduction of the full set of temporal constraints (**U+M+W**), which are closer to real-world multi-UAV scenarios, causes difficulties to planners, especially temporal ones. The two planning models complementarily react to the extensions of the class of problems with different constraints. In particular temporal constraints between target observations (**M**) have a stronger impact on numerical model, while with a temporal model it is more difficult to solve problems involving temporal windows for targets observation (**W**).

	Temporal model				Numerical model			
	U	U+M	U+W	U+M+W	U	U+M	U+W	U+M+W
2U6T	80,0%	90.0%	60,0%	40,0%	100%	97.5%	100%	92,5%
3U8T	50,0%	95,0%	10,0%	45,0%	100%	62,5%	80,0%	62,5%
4U10T	40,0%	80,0%	0,0%	25,0%	100%	5,0%	60,0%	17,5%
Total	60,0%	88,3%	23,0%	36,6%	100%	55,0%	80,0%	57,5%

Table 1. Every cell reports the rate of problems with a certain set of constraints solved by the planner COLIN in *numerical* and *temporal planning*.

We also tested the system by considering six real-world scenarios (missions very similar to the ones used in SMAT ([11]) as a test bed). These scenarios are very demanding since they involve two UAVs and up to nine target observations (both of Point and LOC types). Moreover, the mission requests contain very strict temporal constraints and inter-target constraints. This is very hard test for automatic planner. In fact, only the two simplest problems were solved by a temporal planner, while the numeric approach allowed to find a plan for all the 6 missions (within a timeout of 10 min).

Conclusions. The work shows that PDDL numerical and temporal approaches can be successfully employed (with some work of knowledge engineering and a nontrivial encoding phase) to efficiently model and solve a great number of real life complex problems involving cooperative heterogeneous robotic agents in which numerical resources, time and continuous effects are mandatory. The main difficulty of these approaches is not expressiveness, but rather scalability, which is mainly due to temporal constraints. In particular constraints between different agents are challenging for numerical models, due to the simulation of concurrency, while temporal windows of observability of targets are more demanding for a temporal model than a numerical one which treats them as numerical constraints, ignoring time concept. Results shown that the adoption of a numerical model, despite the necessity of much more complex encoding and decoding phases, is advantageous in real-world scenarios, since it reacts better than temporal model. However, there is no clear winner between the two models: each one is better for a certain type of problems. It is easy to integrate the two approaches within the same architecture and decide which model to use, according to the types of requirements the user expressed. Action-based approaches therefore have proven to be competitive with other state-of-art proposals to multi-agent planning (e.g. our results are comparable to [9]) and, even if still limited and sometimes nontrivial to adopt, their capabilities (as also shown in recent works on new numeric planning heuristics, e.g. [21]) seem to be able to provide opportunities of further progress.

References

1. Malik Ghallab, Dana S. Nau, Paolo Traverso: The actor's view of automated planning and acting: A position paper. Artif. Intell. 208: 1-17 (2014)
2. Enrico Scala, Pietro Torasso: Proactive and Reactive Reconfiguration for the Robust Execution of Multi Modality Plans. ECAI 2014: 783-788
3. Maria Fox, Derek Long: PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains. J. Artif. Intell. Res. (JAIR) 20: 61-124 (2003)
4. S Edelkamp, J Hoffmann: PDDL2. 2: The language for the classical part of the 4th international planning competition. IPC04, at ICAPS04
5. Marta Cialdea Mayer, Andrea Orlandini, Alessandro Umbrico: A Formal Account of Planning with Flexible Timelines. TIME 2014: 37-46
6. Malik Ghallab, Herv Laruelle: Representation and Control in IxTeT, a Temporal Planner. AIPS 1994: 61-67
7. Donati, A., Policella, N., Cesta, A., Fratini, S., Oddi, A. Cortellessa, G., Pecora, F., Schulster, J., Rabenau, E., Niezette, M., Steel, R.: Science Operations Pre-Planning & Optimization using AI constraint-resolution - the APSI Case Study 1. In: Proc. of SpaceOps-08.
8. Barreiro, J., Boyce, M., Do, M., Frank, J., Iatauro, M., Kichkaylo, T., Morris, P., Ong, J., Remolina, E., Smith, T., Smith, D.: EUROPA: A Platform for AI Planning, Scheduling, Constraint Programming, and Optimization. In: ICKEPS 2012
9. A Richards, J Bellingham, M Tillerson, J How: Coordination and control of multiple UAVs. AIAA guidance, navigation, and control conference, Monterey, CA
10. JS Bellmgham, M Tillerson, M Alighanbari, JP How: Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environments. Proc. of IEEE, 2816-2822 (2002).
11. M. Boccalatte, F. Brogi, F. Catalfamo, S. Maddaluno, M. Martino, V. Mellano, P. R. Prin, F. Solitro, P. Torasso, G. Torta: A Multi-UAS Cooperative Mission Over Non-Segregated Civil Areas. JIRS 70(1-4): 275-291 (2013)
12. Michael Brenner, Bernhard Nebel: Continual planning and acting in dynamic multiagent environments. Autonomous Agents and Multi-Agent Systems 19(3): 297-331 (2009)
13. Sebastiano Concetto Marco Caff, Francesco Di Mauro, Enrico Scala: A Numeric PDDL Based Approach for Temporally Constrained Journey Problems. ICTAI 2014: 99-106
14. Shah, M., et al.: "Knowledge engineering tools in planning: State-of-the-art and future challenges." Knowledge Engineering for Planning and Scheduling (2013): 53.
15. Davide Dell'Anna: Numerical and temporal planning for a multi-agent team acting in the real world. http://www.davidedellanna.com/docs/MSc_Thesis_DellAnna.pdf
16. Amanda Jane Coles, Andrew Coles, Maria Fox, Derek Long: COLIN: Planning with Continuous Linear Numeric Change. J. Artif. Intell. Res. (JAIR) 44: 1-96 (2012)
17. Amanda Jane Coles, Andrew Coles, Maria Fox, Derek Long: Forward-Chaining Partial-Order Planning. ICAPS 2010: 42-49
18. Jrg Hoffmann: The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric State Variables. J. Artif. Intell. Res. (JAIR) 20: 291-341 (2003)
19. Alfonso Gerevini, Ivan Serina: LPG: A Planner Based on Local Search for Planning Graphs with Action Costs. AIPS 2002: 13-22
20. Patrick Eyerich, Robert Mattmüller, Gabriele Röger: Using the Context-enhanced Additive Heuristic for Temporal and Numeric Planning. ICAPS 2009
21. Enrico Scala, Patrik Haslum, Sylvie Thiébaux, Miquel Ramírez: Interval-Based Relaxation for General Numeric Planning. ECAI 2016: 655-663