MAS-Planes: A Multi-Agent Simulation Environment to Investigate Decentralized Coordination for Teams of UAVs

(Demonstration)

Marc Pujol-Gonzalez
IIIA-CSIC
Campus de la UAB
E-08193 Bellaterra
mpujol@iiia.cisc.es

Jesus Cerquides
IIIA-CSIC
Campus de la UAB
E-08193 Bellaterra
cerquide@iiia.cisc.es

Pedro Meseguer
IIIA-CSIC
Campus de la UAB
E-08193 Bellaterra
pedro@iiia.cisc.es

ABSTRACT

The purpose of this demo is twofold. On the one hand, we present a novel problem that has much potential for multi-agent research. This is the problem of allowing a team of UAVs with limited communication range to autonomously coordinate to service requests so that the overall time to service requests is minimized. On the other hand, we describe the main features of MAS-Planes, a simulation environment specifically geared towards testing coordination algorithms for that coordination problem.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

Keywords

coordination; UAV; multi-agent simulation

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are an attractive technology for large-area surveillance [4]. UAVs have traditionally been controlled either remotely or by following externally-designed flight plans. Requiring human operators for each UAV implies a large, specialized and expensive human workforce. Likewise, letting UAVs follow externally prepared plans introduces a single point of failure (the planner) and requires UAVs to employ expensive (satellite) radios to maintain continuous communication with a central station. While these constraints are acceptable in some application domains, other applications require more flexible techniques.

For instance, consider a force of rangers tasked with the surveillance of a large natural park. Upon reception of an emergency notification, the rangers need to assess the situation as quickly as possible. With this aim, they could deploy a team of UAVs to be continuously flying throughout the park. Thereafter, they could issue requests to their UAVs to check certain locations. To maintain the cost-effectiveness of the approach, we assume that such UAVs cannot employ expensive communication devices. Thus, the UAVs will have limited communication ranges, significantly smaller than the park's extension. Notice that, in this setting, neither human remote control nor centralized planning is feasible due to such communication constraints.

As illustrated by our example, the only reasonable strategy to achieve UAV cooperation with limited communication range is to let the UAVs directly coordinate themselves in a decentralized manner. In this view, agents must determine their best possible actions at each point in time so that the overall time to service requests is minimized. This is the objective of the *limited-range online routing problem* (LORP), which is the focus of this work. Unfortunately, most of state-of-the-art multi-agent allocation algorithms cannot be employed for this problem, particularly because of the communication range limitation. Therefore, to the best of our knowledge, the LORP is a novel multi-agent research problem. Next we introduce a software framework, available at [5], to support the research of coordination algorithms for the LORP.

2. MAS-PLANES: A FRAMEWORK TO SUPPORT RESEARCH ON THE LORP

Most current UAV simulators are mainly designed for the development of low-level flight control and coordination techniques [1, 3]. They employ highly accurate physics and component models to produce as accurately as possible simulations. The final result are very complex simulators that require significant specific knowledge and domain expertise. Furthermore, being built for such a specialized purpose they do not emphasize the decoupling of their different components. As a consequence, it is largely impractical to adapt them to perform more specialized problem-specific experiments under varying assumptions. Additionally, they are very computationally demanding, making their usage impractical to conduct the large number of experiments required to asses the behavior of a novel coordination algorithm. Other researchers have identified these issues and developed the CoUAV [2] simulator. However, they focused on the problem of cooperative search and exploration instead of request servicing, so it was not applicable to our problem domain. Due to the aforementioned limitations, we developed MASPlanes, a simulation environment geared towards testing coordination methods for the LORP.

The MAS-Planes software framework is composed of several components:

The **problem generator** allows to randomly generate par-

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

ticular problem instances based on a wide number of parameters (e.g.: number, speed and communication range of the UAVs, frequency and spatial distribution of the incoming requests among others). Users define a scenario by fixing these parameters, and then they can generate as many problem instances of this scenario as required. A problem instance completely defines the characteristics of each participating UAV, as well as each request that will be introduced during the simulation. As a consequence, we can compare the performance of different UAV coordination algorithms on the exact same dynamic situations.

Simulation engine. As usual in the multi-agent community [6], our simulation engine is a step-based simulator. It assumes that UAVs move at a constant speed and that they fly at different heights (so there is no need to actively avoid collisions). The main loop of the simulator is straightforward: first, the kernel initializes all participating agents. Thereafter, a *step* is performed every 100 ms, where each agent is given the opportunity to receive, compute, act and send messages. Messages sent at some step are not delivered until the next one. The simulator keeps performing steps until all requests have been introduced into the system and serviced by the UAVs. Finally, it reports statistics about the requests' servicing time, distances travelled by the UAVs and total run time.

A simulation involves two main agent types: operators and UAVs. Operators are in charge of requesting UAVs to check some locations. Hence, during the initialization procedure, operator agents load the list of requests they will introduce from the problem definition. This list specifies each location that must be visited and when to introduce that request. When the time comes, operators introduce each request into the system by sending a message to one of the UAVs that lies within its communication range. If no UAV is reachable by the operator at that particular point in time, the operator keeps the request until some UAV enters its range and then delivers it. Each UAV agent maintains a list of (unserviced) requests that have been assigned to it. At each step, each UAV moves to the closer location between all the requested ones. When a requested location is reached, the request is considered fulfilled and thus eliminated from the list of pending requests of that UAV. If at some step a UAV has no requests assigned, it tries to get in range of the nearest operator to possibly acquire more requests.

LORP algorithm API. Users can extend the basic UAV operation described above by adding any number of *behaviors* to implement the coordination algorithm of their choice. To illustrate that, our framework offers the implementation of a collection of state-of-the-art coordination algorithms, both centralized (i.e. Hungarian or sequential single-item auctions) and distributed (i.e. parallel single-item auctions, max-sum, DSA). Notice that although centralized algorithms cannot be applied in practice to solve LORP instances, they can be used by relaxing some of the constraints in the problem. Their results, unattainable in practice, provide baselines for comparison with distributed algorithms, which are the ones that can actually solve LORPs. Also, MAS-Planes includes extensive documentation and a step by step tutorial that describes how to develop and evaluate an algorithm.

Evaluation tools. At run-time, the user may select any available coordination algorithm and monitor the resulting behavior of the UAVs through the simulator's GUI depicted in Figure 1. This allows the designer to observe the emergent

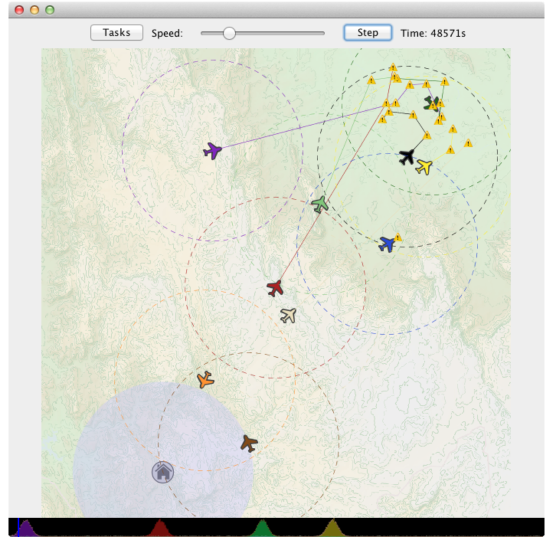

Figure 1: Simulator graphical interface.

behaviors resulting from her algorithm, providing valuable insight on how it operates in practice. Finally, the simulator can also run in *batch* mode, where it simply runs the simulation until the end and reports statistics about that particular problem instance solved with the specified coordination algorithm and settings.

To conclude, we have presented: (1) the dynamic limited-range online routing problem (LORP), a novel multi-agent research problem; and (2) MAS-Planes, a software framework that supports the research on coordination algorithms for the LORP.

Acknowledgments

Work funded by projects RECEDIT (TIN2009-13591-C02-02), AT (CSD2007-0022), COR (TIN2012-38876-C02-01), MECER (201250E053), the Generalitat of Catalunya grant 2009-SGR-1434, and the Ministry of Economy and Competitivity grant BES-2010-030466.

3. REFERENCES

[1] R. Garcia and L. Barnes. Multi-uav simulator utilizing x-plane. In *2nd International Symposium on UAVs*, pages 393–406. Springer, 2010.

[2] J. Happe and J. Berger. Couav: a multi-uav cooperative search path planning simulation environment. In *Summer Computer Simulation Conference*, pages 86–93, 2010.

[3] M. Jang, S. Reddy, P. Tosic, L. Chen, and G. Agha. An actor-based simulation for studying uav coordination. In *ESS*, pages 593–601, 2003.

[4] D. Kingston, R. W. Beard, and R. S. Holt. Decentralized perimeter surveillance using a team of uavs. *IEEE Transactions on Robotics*, 24(6):1394–1404, 2008.

[5] M. Pujol-Gonzalez, J. Cerquides, P. Meseguer, and J. A. Rodriguez-Aguilar. MASPlanes simulation toolkit. `https://github.com/MASPlanes/MASPlanes`.

[6] S. F. Railsback, S. L. Lytinen, and S. K. Jackson. Agent-based simulation platforms: Review and development recommendations. *Simulation*, 82(9):609–623, 2006.