A SWARM ALGORITHM FOR OPTIMIZING PATH-PLANNING WITH MULTIPLE DEPOTS AND BATTERY RECHARGING USING AN INVERTED ANT COLONY ALGORITHM

Jan-Gerald Huenges

Distributed Artificial Intelligence Laboratory
Technical University Berlin
Str. des 17. Juni 135
Berlin
dominik.schroeck@campus.tu-berlin.de

Santiago Masip Gimeno

Distributed Artificial Intelligence Laboratory
Technical University Berlin
Str. des 17. Juni 135
Berlin
dominik.schroeck@campus.tu-berlin.de

Aleksandra Pranskaityte

Distributed Artificial Intelligence Laboratory
Technical University Berlin
Str. des 17. Juni 135
Berlin
dominik.schroeck@campus.tu-berlin.de

Dominik Schroeck

Distributed Artificial Intelligence Laboratory
Technical University Berlin
Str. des 17. Juni 135
Berlin
dominik.schroeck@campus.tu-berlin.de

Christopher-Eyk Hrabia, M.Sc.

Distributed Artificial Intelligence Laboratory
Technical University Berlin
Str. des 17. Juni 135
christopher-eyk.hrabia@dai-labor.de

Prof. Dr.-Ing. habil. Dr. h.c. Sahin Albayrak

Distributed Artificial Intelligence Laboratory
Technical University Berlin
Str. des 17. Juni 135
Berlin
sahin.albayrak@dai-labor.de

1 INTRODUCTION

Unmanned Aerial Vehicles (UAV) are gaining popularity in private as well as commercial sector. One often-mentioned use-case is the parcel delivery. In the near future, UAV might be used to deliver parcels to your doorstep. The logistics industry hopes for positive impacts on their costs and competitiveness. The deployment of a delivery concept is quite complex. To allow for early evaluation and reduce the number of required flight-testing, a simulation engine is required. This is why SIMULATIONNAME has been developed. It supports multiple depots and UAV as well as simple change of path-planning algorithm. UAVs move in a 3D environment The engine does not simulate the behaviour of the UAV in the aerial space itself but allows for evaluation of spatial distribution of depots or the suitability of a path-planning algorithm. It was developed considering the use of swarm algorithms in the delivery use-case and allow for inter-drone communication and autonomous organisation avoiding a central ground control.

2 RELATED WORK

Literature on simulation engines for UAV delivery is still relatively scarce. Simulation engines that were developed concentrate rather on specific components or flight behaviour. (Johnson and Fontaine 2001) developed a simulation engine for the behaviour of specific hardware components of a UAV. (Lu and Geng 2011) simulated the flight dynamic behaviour of a UAV using Matlab/Simulink. MultiUAV (Rasmussen, Mitchell, Schulz, Schumacher, and Chandler 2003), a very general simulation engine has been developed to simulate cooperative algorithms for finding targets. In 2005, an agent-based simulation engine implementing a surveillance scenario has been developed (Jang, Reddy, Tosic, Chen, and Agha 2005). UAV try to find targets without prior knowledge of the targets' locations. They implement UAVs as agents and also use the notion of obstacles and bases. The sensor concept is similar to the one used in this paper.

3 SCENARIO

NO CENTRAL ENTITY, optimizes
Before embarking the software model, the delivery scenario shall be described. Every UAV has a depot, in the engine called basestation, from which it receives items to be delivered. An item's destination is within the range of a Basestation which means that every Basestation has a limited range in the area. UAVs are assigned to one specific Basestation and only deliver items of it. After receiving an item, the UAV will begin flying to the specific destination. It scans the area and stores information on all fields it has seen en-route. If a UAV meets another one, they will exchange information on the already-explored parts of the area.
If a field on the route contains an obstacle, it will deviate from its route and try to find one around the obstacle. Obstacles can have different heights, thus UAV als
After delivering the item, it will fly back to the basestation to receive a new item. If the battery reaches a certain threshold, the UAV will fly to the nearest basestation (not neccessarily its home base) and recharge its battery. UAVs are moving in a 3D environment which means that obstacles can have different heights.

4 DESCRIPTION OF THE MODEL

Figure 4 represents the software architecture. ARCHITECTURE GRAPHIC

1. General Structure : UI, UAV Simulator, MESA, Analysis, BaseStations (depots)
2. BaseStation: In the real world, this is the depot of a logistics company. It assigns orders to UAVs and hands items over to them. Basestations are responsible for all items in a part of the map and for a subset of all drones. Also, they contain chargers for UAVs whose battery reached a low level. In the configuration, one can set how many basestations are supposed to be on the map by specifying the number of fields the basestation is responsible for (its delivery area). Basestations are located as close to the center of their area as possible, but only on obstacles (on buildings).
3. An obstacle can be any static object that obstructs a UAV on its way flying to a point (such as trees, buildings etc). Obstacles can have different heights. The simulator will load the height information from a JPEG file with different colors for different heights. The height levels represent different height corridors that UAVs are allowed to fly in.
4. The simulation is built upon the MESA modeling framework. The following
5. UI: GUI in Browser, config.ini: As per MESA standard, the User interface is realized using a web-server and a browser-based UI. The configuration is parsed from a configuration file, config.ini. It configures the setup of the grid (altitude levels, pixel width/height), UAV settings such as battery life and the sensor range. The simulation also supports different runmodes, one for running unit-tests, one for ordinary run with the server and graphical user interface as well as running without GUI which improves performance as no rendering is required.

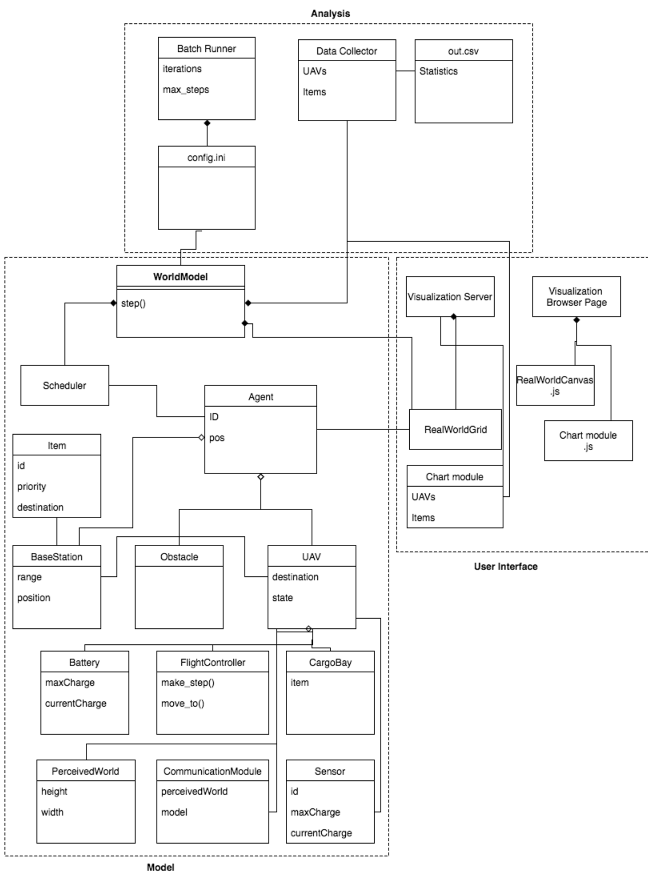

Figure 1: Architecture of simulation model

6. UAV Simulator
7. Data Analysis: CSV output of KPI, out.csv

4.1 MESA

MESA (Masad and Kazil 2015) is an open source agent-based modeling framework (ABM) developed specifically for the Python programming language. Agents are objects that have rules and states that act accordingly on every step of the simulation (Axtell 2000). ABM allows to capture the path of the simulation along with the solution and allows for analysing the dynamic history (Axtell 2000). ABM also allows to dynamically pause and resume the simulation at any given step and analyse the current results.
MESA comes with implementations of important components, such as a grid for implementing a simple 2D environment, a web-browser-based UI, a data analysis tool and an agent scheduler. It allows for easy extension or modifications to develop specific simulation engines.
Our simulation uses all core components of MESA, such as the MultiGrid, the DataCollector, the scheduler (RandomActivation), agents and the server component (Masad and Kazil 2015):

- The MultiGrid is a 2D grid with fields that allow for more than one agent on one field. This is particularly important for our simulation as agents of different height are all stored in the same grid in the model.
- The DataCollector is used for collecting relevant quantitative data to support the evaluation of an algorithm.
- The scheduler activates the agents. An agent will only do an action if the scheduler activates it. We use the RandomActivation which means that in every step the agents are activated in a random order.
- The server renders the graphical user interfaces and renders changes on the model. It also allows for pausing and resuming the simulation or progress step-wise.

In order for MESA to meet our requirements, we made some modifications to MESA.

- Which components did we remove or modify and why?
- Reprogramming of rendering engine. How and why?

4.2 GUI

1. No simulation of a ground control system! The software is built for simulating decentralized organization.
2. The GUI is implemented using MESA's standard server with some changes to improve rendering

4.3 UAV

1. most important component, tried to make a realistic simulation of a UAVs components
2. UAV do not communicate with a central entity such as a control server organizing the flight traffic. UAV communicate with each other if they are in sensor range and only rely on their own sensors.
3. Perceived World contains all cells that the UAV knows already. It stores information on obstacles. To improve performance, this is realized using dictionaries, one for each height level, mapping coordinates to information whether there is an obstacle on that field or not. For performance reasons, we decided against using MESA's grid here as iterating it reduces performance significantly. This is

especially true for the grid exchange where each UAV would have to check every field on the grid for content. A dictionary contains unique values and is easier to exchange.

4. Components (make a drawing). No sensors for physical flight behavior
   - FlightController: This component contains the path-finding/route-planning algoritm. It represents a navigation unit. Multiple algorithms could be implemented, compare section 5 for an example. The algorithm can make use of the exchanged information on the grid provided by the *CommunicationModule* component and thus supports swarm algorithms.
   - Battery: The battery has a limited battery life that decreases on each step. When the battery reaches a certain threshold (configurable), the UAV will fly to the nearest depot to recharge.
   - CargoBay: The cargoBay represents the UAV's storage unit which can contain exactly one item.
   - CommunicationModule: If the sensor found another UAV in the sensor's radius, this component will communicate with the other UAV and exchange grid information. Exchange of information on previously-unknown terrain helps when a UAV has to deliver an item to a new area of the map. This allows for precomputing optimal routes before exploring the actual area.
   - Sensor: Scanning the grid for obstacles and other agents. This component is the interface to the model
5. States

(Jang, Reddy, Tosic, Chen, and Agha 2005) can be helpful to structure this part!

1. multiple depots
2. Describe the modular and extensible character of the architecture
3. Describe how the model accomodates for the integration of self-organizing algorithms for UAVs
4. Describing changes and extensions in the mesa framework
5. drones have to recharge and thus deviate from their actual route
6. ...

4.4 Data Analysis

- KPI's
- out.csv
- DataCollector from MESA

5 IMPLEMENTATION OF A SAMPLE ALGORITHM: A STAR

To provide an example simulation, the A* algorithm has been used SOURCES!

6 EVALUATION

1. Use KPI's that we defined
2. Compare with existing

7 CONCLUSION

1. Overall summary
2. limitations of this approach
3. future development?

REFERENCES

Axtell, R. 2000. “Why agents?: on the varied motivations for agent computing in the social sciences”.

Jang, M.-W., S. Reddy, P. Tosic, L. Chen, and G. Agha. 2005. “An actor-based simulation for studying UAV coordination”. In *15th European Simulation Symposium*, pp. 323.

Johnson, E., and S. Fontaine. 2001. “Use of flight simulation to complement flight testing of low-cost UAVs”. In *AIAA Modeling and Simulation Technologies Conference*.

Lu, P., and Q. Geng. 2011. “Real-time simulation system for UAV based on Matlab/Simulink”. In *Computing, Control and Industrial Engineering (CCIE), 2011 IEEE 2nd International Conference on*, Volume 1, pp. 399–404. IEEE.

Masad, D., and J. Kazil. 2015. “Mesa: An Agent-Based Modeling Framework”.

Rasmussen, S., J. Mitchell, C. Schulz, C. Schumacher, and P. Chandler. 2003. “A multiple UAV simulation for researchers”. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, pp. 5684.

AUTHOR BIOGRAPHIES

THERESA M. K. ROEDER is an Associate Professor of Decision Sciences at San Francisco State University. She holds a PhD in Industrial Engineering and Operations Research from UC Berkeley. Her research interests lie in O.R. education and simulation modeling, especially in healthcare and higher education. Her email address is wsc15roeder@gmail.com.

PETER I. FRAZIER is an Associate Professor in the School of Operations Research and Information Engineering at Cornell University. He holds a Ph.D. in Operations Research and Financial Engineering from Princeton University. His research interests include optimal learning, sequential decision-making under uncertainty, and machine learning, focusing on applications in simulation optimization, e-commerce, and the physical sciences. His email address is wsc16frazier@gmail.com.

ROBERTO SZECHTMAN received his Ph.D. from Stanford University, and currently is Associate Professor in the Operations Research Department at the Naval Postgraduate School. His research interests include applied probability and military operations research. His email address is wsc16szechtman@gmail.com.

ENLU ZHOU is an Assistant Professor in the H. Milton Stewart School of Industrial & Systems Engineering at Georgia Institute of Technology. She holds a Ph.D. in Electrical Engineering from the University of Maryland, College Park. Her research interests include stochastic control and simulation optimization, with applications in financial engineering and revenue management. Her email address is wsc16zhou@gmail.com.

First Name Last Name 1
First Name Last Name 2

Affiliation
Address, City, Country
E-mail address

Figure 2: Example title page heading with 2 authors from the same institution.

First Name Last Name 1

Affiliation 1
Address, City, Country 1
E-mail address 2

First Name Last Name 2

Affiliation 2
Address, City, Country 1
E-mail address 2

Figure 3: Example title page heading with 2 authors from different institutions.

First Name Last Name 1

Affiliation 1
Address, City, Country 1
E-mail address 2

First Name Last Name 2

Affiliation 2
Address, City, Country 1
E-mail address 2

First Name Last Name 3

Affiliation 3
Address, City, Country
E-mail address 2

Figure 4: Alternate example title page heading with 3 authors from different institutions.

First Name Last Name 1	First Name Last Name 2
Affiliation 1 Address, City, Country 1 E-mail address 2	Affiliation 2 Address, City, Country 1 E-mail address 2
First Name Last Name 3	First Name Last Name 4
Affiliation 3 Address, City, Country 1 E-mail address 2	Affiliation 4 Address, City, Country 1 E-mail address 2

Figure 5: Example title page heading with 4 authors from different institutions.