DUCK: A Drone-Urban Cyber-Defense Framework Based on Pareto-Optimal Deontic Logic Agents

Tonmoay Deb, [1] **Jürgen Dix,** [4] **Mingi Jeong,** [2] **Cristian Molinaro,** [3] **Andrea Pugliese,** [3] **Alberto Quattrini Li,** [2] **Eugene Santos,** [2] **V.S. Subrahmanian,** [1*†] **Shanchieh Yang,** [5] **Youzhi Zhang,** [6]

[1]Department of Computer Science, Northwestern University, Evanston, IL, United States
[2]Department of Computer Science, Dartmouth College, Hanover, NH, United States
[3]Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, Rende, Italy
[4]Department of Informatics, Technical University of Clausthal, Clausthal, Germany
[5]Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, United States
[6]CAIR, Hong Kong Institute of Science & Innovation, Hong Kong

Abstract

Drone based terrorist attacks are increasing daily. It is not expected to be long before drones are used to carry out terror attacks in urban areas. We have developed the DUCK multi-agent testbed that security agencies can use to simulate drone-based attacks by diverse actors and develop a combination of surveillance camera, drone, and cyber defenses against them.

Introduction

76 drone-based terrorist attacks by groups like ISIS, PKK, and occurred before 2019 with a 70% success rate (Barten et al. 2022; Balkan 2019).

DUCK is an agent-based drone urban cyber-defense testbed in which a Blue team defends a city against a Red team (terrorists). Built using a deontic logic theory (Subrahmanian et al. 2000; Stroe, Subrahmanian, and Dasgupta 2005), DUCK allows defenders to use CCTVs, Blue drones, and cyber attacks, to defend against a swarm of Red drones. Defenders can simulate attacks and see the impact of those attacks. We can answer questions such as "If there are m blue drones (of different types and capabilities) and n red drones of varying types and capabilities that attack New York, what damage will be done?" Eventually, DUCK will enable defenders to answer questions such as: (i) how many blue drones are needed to save X% of New York City from k red drones? (ii) if we invest Y dollars to protect Chicago from k red drones, what percentage of Chicago will be saved?

DUCK Testbed

DUCK represents a city as a graph: nodes represent regions, edges denote adjacency relationships. Each drone is an autonomous agent. Autonomous Blue and Red headquarter (HQ) agents serve as a command centers for their team. Blue has CCTV agents that monitor nodes. HQ agents set goals for drones (e.g. protect a set of nodes), send them instructions, and/or hack them. Drone agents can move from node to node, fire at drones of the other color, and fire at a node (red drones only), send camera feeds to its HQ, and more.

Each agent can perform some actions according to constraints in a deontic logic "agent program", e.g., a blue drone bd may be permitted (but not required) to fire at red drones within their firing range as long as they have payload. bd might be forbidden from firing at city nodes (Eiter, Subrahmanian, and Pick 1999). Upper (resp. lower) case symbols denote variables (resp. constants).

$$\begin{aligned} \mathbf{P}\, fire(RD) &\leftarrow range(bd, R)\,\&\, dist(bd, RD, R')\,\&\\ &\quad R' \leq R\,\&\, payload(bd, P)\,\&\, P \geq 0.\\ \mathbf{F}\, fire(CN) &\leftarrow city\_node(CN). \end{aligned}$$

A drone's agent program imposes constraints on what the agent is permitted, forbidden, or obliged to do in different situations. Each agent autonomously chooses what actions to perform in a given state based on a set of objective functions to capture local objectives (e.g. a blue drone might minimize expected damage to only the nodes it is protecting) and global objectives (e.g. minimize expected damage to the whole city). The key computation performed by an agent is to compute a Pareto-optimal set of actions at each time step. DUCK enables us to vary all of these parameters, e.g. number and types of drones, capabilities, positioning, agent actions, agent programs, objective functions and more.

At any time t, a DUCK agent communicates with other agents during a communication phase, and then computes a Pareto-optimal set of actions to perform (e.g. hacking actions (Lin et al. 2018; Yahuza et al. 2021; Nassi et al. 2021), firing, moving to another location, etc.) leading to a new state at time $t+1$. DUCK includes new algorithms to compute such Pareto-optimal sets of actions extending (Stroe, Subrahmanian, and Dasgupta 2005) (which looked at a single objective function).

DUCK Testbed Implementation

The DUCK architecture (Figure 1) has over 10,100 lines of code in C++ and Python. DUCK uses a customized 3D

*corresponding author. Email: vss@northwestern.edu

†Video at: https://sites.northwestern.edu/nsail/projects/duck/

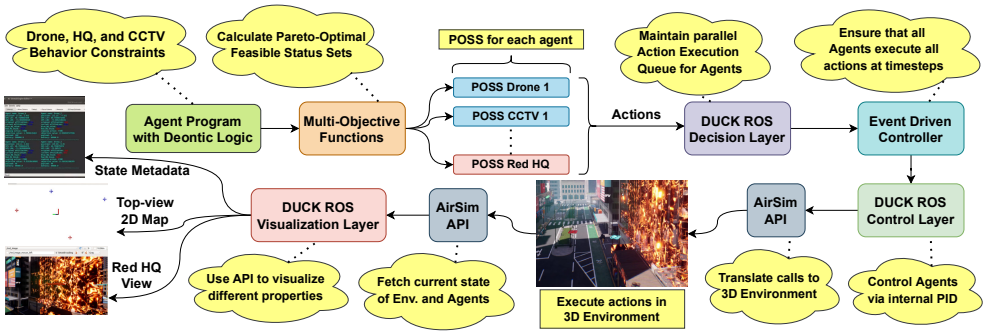

Figure 1: Simplified DUCK Testbed Architecture for a single execution cycle (timestep).

city model based on the Unreal Engine and Microsoft's AirSim (Shah et al. 2018) simulator to visualize multiple drones in the 3D environment, navigation, and real-time drone state management. We extended AirSim to include CCTVs, drone-drone attacks, destruction of regions, hacking, battery and payload constraints. We built high-level Python APIs for the new DUCK features to interact with AirSim. Key visualization, control, communications, and concurrency aspects were built within the Robot Operating System (ROS) which is connected to AirSim. The ROS component is used in DUCK's decision, control, and visualization layers. Each layer has separate nodes running in parallel for message communications. The decision layer identifies the actions each agent decides to perform at time t (by finding a Pareto-optimal set of actions for that agent that is compatible with that agent's agent program). Not all actions that agents perform will succeed. The control layer injects stochasticity into agent actions and determines which attempted actions succeed. For example, a blue drone bd may fire at a red drone rd. ROS will render this in the 3D environment but may decide that rd is not destroyed. The visualization layer shows information such as ground truth, attempted actions by agents objective values, camera images sent by agents, real-time movement on a map and more.

DUCK Demonstration

The DUCK demo[1] allows a user to set the number and capabilities (e.g. payload, firing range, battery) of the blue and red drones, and the number of CCTVs. Figure 2 shows the demonstration on 3 screens shortly after launching.

[1]Available at: https://sites.northwestern.edu/nsail/projects/duck

Left Screen. The top left window shows Red HQ's view of a red drone camera (which can be swapped to show other red drone cameras). The bottom left window shows how objective function values change as the demonstration proceeds (e.g. if an agent's actions don't work out as expected because of another agent's actions). The bottom right view shows agents' optimal actions at time t. The top right window shows a real-world map with top-view of the drones.
The Middle Screen shows the ground truth at one intersection in NYC. Three blue drones (in the yellow circle) are shown firing at a target red drone (red circle) which is trying to destroy the region.
Right Screen. The top and bottom windows of the right part show Blue HQ's view (of one blue drone camera) and video from one CCTV. The center-left window shows the camera view controller which allows users to swap HQ camera feeds. The top-left window visualizes the state of the drones, e.g., GPS coordinates, battery, payload, hack status, alive/destroyed, etc. The launch button in the bottom center handles event-driven simulation. On each click, agents execute actions at that timestep.

Conclusion

DUCK allows defenders to simulate terrorist attacks on urban areas and assess efficacy of diverse defenses (e.g. drone deployments, hacking) and thus compute how to achieve minimize damage to the city under some cost constraints. Extensions will include using game theoretic and reinforcement learning models, hacking, vision, pursuit-evasion (Agmon, Kaminka, and Kraus 2011; Amigoni and Basilico 2012), and human-agent interaction techniques.

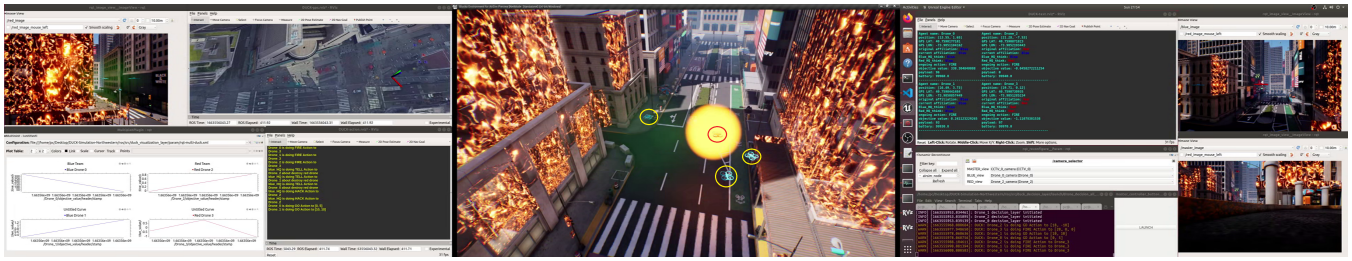

Figure 2: DUCK 3-Screen Demonstration. The middle screen visualizes GT. Left and right screens show other technical details.

References

Agmon, N.; Kaminka, G. A.; and Kraus, S. 2011. Multi-robot adversarial patrolling: facing a full-knowledge opponent. *Journal of Artificial Intelligence Research*, 42: 887–916.

Amigoni, F.; and Basilico, N. 2012. A game theoretical approach to finding optimal strategies for pursuit evasion in grid environments. In *IEEE International Conference on Robotics and Automation*, 2155–2162.

Balkan, S. 2019. A Global Battlefield? Rising Drone Capabilities of Non-State Armed Groups and Terrorist Organizations. Technical report, SETA Foundation for Political, Economic and Social Research.

Barten, D. G.; Tin, D.; De Cauwer, H.; Ciottone, R. G.; and Ciottone, G. R. 2022. A Counter-Terrorism Medicine Analysis of Drone Attacks. *Prehospital and Disaster Medicine*, 37(2): 192–196.

Eiter, T.; Subrahmanian, V.; and Pick, G. 1999. Heterogeneous active agents, I: Semantics. *Artificial Intelligence*, 108(1-2): 179–255.

Lin, C.; He, D.; Kumar, N.; Choo, K.-K. R.; Vinel, A.; and Huang, X. 2018. Security and Privacy for the Internet of Drones: Challenges and Solutions. *IEEE Communications Magazine*, 56(1): 64–69.

Nassi, B.; Bitton, R.; Masuoka, R.; Shabtai, A.; and Elovici, Y. 2021. SoK: Security and Privacy in the Age of Commercial Drones. In *IEEE Symposium on Security and Privacy*, 1434–1451.

Shah, S.; Dey, D.; Lovett, C.; and Kapoor, A. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 621–635. Springer.

Stroe, B.; Subrahmanian, V.; and Dasgupta, S. 2005. Optimal status sets of heterogeneous agent programs. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, 709–715.

Subrahmanian, V. S.; Bonatti, P.; Dix, J.; Eiter, T.; Kraus, S.; Ross, R.; Ozcan, F.; and Dix, J. 2000. *Heterogeneous Agent Systems*. MIT press.

Yahuza, M.; Idris, M. Y. I.; Ahmedy, I. B.; Wahab, A. W. A.; Nandy, T.; Noor, N. M.; and Bala, A. 2021. Internet of Drones Security and Privacy Issues: Taxonomy and Open Challenges. *IEEE Access*, 9: 57243–57270.