A real-word realization of the AntNet routing algorithm with ActivityBots

Jonas Wilfert, Niklas Paprotta, Oliver Kosak, Simon Stieber, Alexander Schiendorfer, Wolfgang Reif

Angaben zur Veröffentlichung / Publication details:

Wilfert, Jonas, Niklas Paprotta, Oliver Kosak, Simon Stieber, Alexander Schiendorfer, and Wolfgang Reif. 2021. "A real-word realization of the AntNet routing algorithm with ActivityBots." In *2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C), DC, USA, 27 September - 1 October 2021*, edited by Esam El-Araby, Vana Kalogeraki, Danilo Pianini, Frédéric Lassabe, Barry Porter, Sona Ghahremani, Ingrid Nunes, Mohamed Bakhouya, and Sven Tomforde, 289–90. Piscataway, NJ: IEEE. https://doi.org/10.1109/ACSOS-C52956.2021.00072.

A Real-Word Realization of the AntNet Routing Algorithm with ActivityBots

Jonas Wilfert*, Niklas Paprotta*, Oliver Kosak†, Simon Stieber†, Alexander Schiendorfer† and Wolfgang Reif†
*, † *Institute for Software & Systems Engineering—University of Augsburg*
*{*jonas.wilfert; niklas.paprotta*}@*uni-a.de* †{*kosak; stieber; schiendorfer; reif*}@*isse.de*

***Abstract*—To ease teaching self-organizing systems design, we implemented the AntNet routing algorithm for real-world application using educational robots called ActivityBot. Using line sensors and ultrasonic distance sensors, the robotic ants traverse a tiled graph printed on paper, collectively converging to the shortest path. In our descriptions, we address the challenges to face when employing such self-organizing systems on educational hardware and provide a video on YouTube https://youtu.be/JFduHJ0o0UM.**

***Index Terms*—AntNet, ActivityBots, Teaching**

1. Motivation

We present a real-world realization of the AntNet routing algorithm with small educational robots as agents. AntNet finds the shortest path on a graph using self-organizing (SO) ants similar to the ant colony optimization algorithm. Using such SO algorithms as a new paradigm for developing software can provide many enhancements compared to traditional approaches. These can be employed in cases where the system needs to be robust against failures, e.g., in the energy domain [1] or in major catastrophe scenarios [2]. However, functionality of SO systems are often hard to grasp. Our real-world realization helps to visualize the behavior of the individual agents. To ease the entry to this field of research for future students, we developed a simulation environment to allow testing own implementations of SO-algorithms. Crossing the gap between a simulation and reality posed some challenges we address in the following. While a corresponding video demonstrating our work can be found on *YouTube* we also provide our implementation[1].

2. Related Work

We implement the AntNet Routing Algorithm [3] which is a modification of the more general approach of Ant Colony Optimization (ACO) [4] specialized for finding the shortest path. Both algorithms are bio-inspired [5] as they mimic the way that ants in nature release pheromones while they travel to perform harvesting tasks efficiently. Similarly, ACO uses several simulated ants to traverse a graph. Once all forward-ants reach the target node, they release pheromones relative to the length of the edges they took by traversing the route backwards as backward-ants. At each node, the ant decides which edge to take next using a heuristic over the pheromone levels and the length of the next edge. Repeating this procedure several times reinforces short paths until all ants converge unto a single, approximately shortest, route. Practical usage of the AntNet algorithm can be found in the task of routing packages in communication networks [6]. One major advantage of the AntNet algorithm compared to classic algorithms like A^* is its robustness against failures of nodes and edges in the network. These can be removed while the algorithm runs and the SO ants adapt to the situation for finding another short path. We chose AntNet as an ACO derivative as the concept of forward- and backward-ants can be applied seamlessly to robotic hardware.

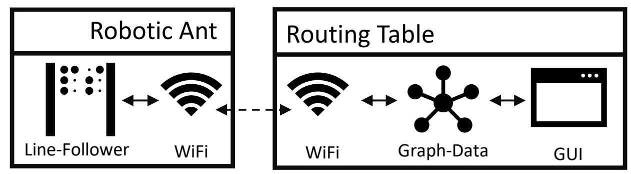

Figure 1. Overview on the system architecture.

3. Realization

We split the implementation into two main parts (cf. Fig. 1). On the hardware side, there are the ActivityBots (ants) driving on a tiled map printed on paper (cf. Section 3). Actions of the ants are controlled by the AntNet routing algorithm running on the central server (cf. Section 3) we use to emulate the release of pheromones inside a shared environment. All actors are connected to the same WiFi access point and communicate over the TCP protocol.

Real World Implementation. The robotic ants are based on the *ActivityBot* educational robot produced by *Parallax* [7]. These robots come with various supported sensors, such as an ultrasonic distance sensor and a line sensor we used. ActivityBots can be easily equipped with WiFi and are very maneuverable. This enables them to follow a graph which we designed consisting of laminated paper tiles. Inter-connectable tiles represent the edges of the graph. They show straight and curved lines the ActivityBots can follow using the line sensors on the bottom of their chassis. We print two lanes on the tiles to avoid colliding with

1. code on GitHub https://github.com/isse-augsburg/AntNetActivityBots

oncoming ants comparable to real-world road traffic. The nodes of the graph are realized as roundabouts to prevent accidents. Each entrance and exit is equipped with a unique barcode to assign which node the ant is approaching or leaving. After reading the barcode using the line sensor, the ActivityBot checks whether the roundabout is uncrowded using the ultrasonic distance sensor before entering it. Each ActivityBot monitors its distance to vehicles in front and slows down if necessary to avoid collisions. Additionally, each ActivityBot detects issues like losing the road markings or entering an unexpected roundabout entrance. If such failure happens, ActivityBots enter an error state and signal the misbehavior by blinking with the onboard LEDs until human intervention. In order to keep the code running on ActivityBots lightweight and straightforward, all complex calculations are outsourced to digital twins running on the central server, we describe in the next Section.

Central Server. A central server is needed to simulate the shared environment and synchronize the pheromones due to the lack of pheromones the robotic ants can leave behind. The server stores a digital twin of each robotic ant to allow better control over them. To stay true to the mindset of the swarm algorithm, the server is not a central authority with global knowledge but a rather a simulation of the environment. Each digital twin waits for requests by the robotic ant when reaching a roundabout and evaluates the subsequent actions based on the simulated pheromones. The new instructions are sent back to the robotic ant, which then executes the requested command while performing some error checking, as explained in the last section. This approach has major advantages compared to running computations directly on the robotic ant: First of all, it allows the ant to be visualized in a central view running on the server, which helps to understand the convergence behavior. Furthermore, it allows the parameters of the AntNet routing algorithm to be changed or replaced entirely by a different swarm algorithm without having to reprogram each robotic ant individually. Due to the physical limitations of the processor and line sensor we used, the velocity of the robotic ant is relatively slow. Estimations show that on a graph with seven nodes and six robotic ants, it would take up to 8 hours to find the shortest path finally. To decrease this period, the concept was extended with pure virtual ants. A virtual ant does not have a robotic ant as a counterpart. Therefore it is not limited to any maximum velocity, and the number of virtual ants can be freely chosen. This allows the convergence of the shortest path to speed up if desired. To synchronize the virtual with the robotic ants, a change from the original AntNet implementation was required. In the original implementation, all ants reach the target eventually. Only then they also update pheromones on the traversed path with respect to the respective path length. This approach does not work with continuous driving ants, as there is a physical limitation concerning the number of real ants driving within the target node or on the road without blocking other ants. As an alternative, we implemented a tick-based concept. With every tick, the purely virtual ants get moved by a fixed distance of a few centimeters on the graph, and the pheromones get evaporated. The evaporation factor needs to be way longer than usual, e.g., $\epsilon = 0.0005 \cdot N$ where N is the number of ants. When an ant reaches the target, it increases the pheromones on the path relative to its taken path length. Instead of following the same path back, it swaps the start and end node and returns to the original start node. The parameters used in the showcase are $\alpha = 1.0$ and $\beta = 0.1$.

Challenges. The line following sensor proved to be inferior, resulting in high variance of the individual sensors. Improvements could be achieved by a 3D-printed light shield and some adjustments to the PI controller. Additionally, two edge lines have been found to be more robust than a single following line in the middle of the road. To ensure the correctness of the barcode reading, parity bits were added to the barcodes at the roundabouts. Due to interference of the ultrasonic distance sensors, a randomized waiting time between the ultrasound impulses was added to prohibit emergent behavior.

4. Conclusion

We created a visualization of the AntNet Routing Algorithm, which uses real-world agents to illustrate the functionality. Using this algorithm, the agents' convergence towards the shortest path is very robust against failures, as seen in the complementary video. Due to the robotic ants' low velocity, we extended the real-world implementation with purely simulated ants. Both types of ants seamlessly work together to optimize a typical graph. By combining software with hardware, we provide a system that improves the teaching experience of SO systems by visualizing the behavior of multiple agents in a real-world environment while maintaining a reasonable convergence time due to the simulation environment. This enables students to implement and test their own SO algorithms in a simple but visually descriptive way.

References

[1] O. Kosak, G. Anders, F. Siefert, and W. Reif, "An approach to robust resource allocation in large-scale systems of systems," in *2015 IEEE Ninth Inter. Conf. on Self-Adaptive and Self-Organizing Systems*. Los Alamitos, Calif.: IEEE Computer Society, 2015, pp. 1–10.

[2] O. Kosak, F. Bohn, L. Eing, D. Rall, C. Wanninger, A. Hoffmann, and W. Reif, "Swarm and collective capabilities for multipotent robot ensembles," in *Leveraging Applications of Formal Methods, Verification and Validation: Engineering Principles*, T. Margaria and B. Steffen, Eds. Cham: Springer, 2020, pp. 525–540.

[3] G. Di Caro and M. Dorigo, "Antnet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.

[4] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comp. Intel. Mag.*, vol. 1, no. 4, pp. 28–39, 2006.

[5] D. Floreano and C. Mattiussi, *Bio-inspired artificial intelligence: Theories, methods, and technologies*, ser. Intelligent robotics and autonomous agents. Cambridge Mass.: MIT Press, 2008.

[6] S. Chandra, U. Shrivastava, R. Vaish, S. Dixit, and M. Rana, "Improved-antnet: Aco routing algorithm in practice," in *UKSim 2009*, D. Al-Dabass, Ed. Piscataway, NJ: IEEE, 2009, pp. 25–29.

[7] Parallax Inc., "Activitybot 360° robot kit." [Online]. Available: https://www.parallax.com/product/activitybot-360-robot-kit/