A Comparison Between Behavioral Arbitrator-Based and Neural Network Architectures

Hugo Birch

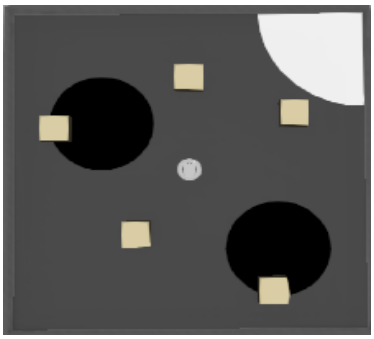
Professor: Alcherio Martinoli
Assistant: Cyrill Baumann

The 21st century, often referred to as the numerical, is at the heart of technological transformations. Robotics is the study of replacing human beings with machines for certain tasks including both physical execution and decision making.

The aim of my project is to find the most effective method to generate an automatic design between two behavior-based methods:

- Artificial Neural Network (ANN) based arbitrator
- Probabilistic Finite State Machine (PFSM) based arbitrator using Mixed-Discrete Particle Swarm Optimization (MDPSO) as optimizer

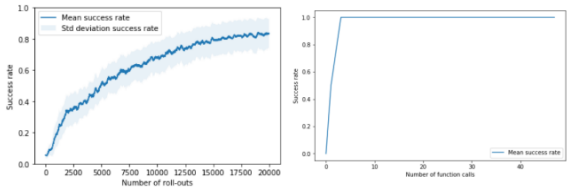
The scenario used to compare and test the different methods is a foraging task based on the previous projects. To successfully complete the task, the robot has to roam the arena until it finds a black area, corresponding to some resources. Once a black area is found, the robot must then find the white area that corresponds to its home. The task is considered completed once the robot reaches the white area.



Top view of the arena in Webots simulator

The PFSM using MDPSO as an optimizer was extensively tested. First, with simple task such as “Light following”. Then, on the foraging task and finally on a more complex foraging task to test the limits of the system. To help it cope with the complexity of the tasks, a diversity preservation and a noise reduction mechanism were introduced.

The ANN was tested using a variant of the foraging task. The robot could be initialized in four different locations at random and the obstacles could be hidden at random.



Success rate for an ANN with VPG (left) and PFSM with MDPSO (right)

The results shown for VPG underline the issue of ANN, as it can be very successful even in noisy environment, but it takes a minimum of 20’000 roll-outs on average to reach a success rate of 0.9 and it does not always converge, with some simulation never finding an optimal set of parameters.

In comparison, MDPSO can be more powerful as all simulations found a set of parameters which completed the task successfully and in less than 500 function-calls. However, MDPSO is more affected by noise which was demonstrated when an random spawn mechanism was introduced. The aim was to find an automatic solution which gives the same result as a handcrafted solution. Both solutions give, after training, the same level of success of 1, satisfying thus both our requirements. However, MDPSO reaches that level faster and converges every time making it a better solution overall.

Future works should focus on preventing premature convergence, this could be done through implementing a different version of PSO such as Guaranteed convergence PSO or by changing the parameters of MDPSO. Another aspect to investigate would be to compare the ANN-arbitrator with a full ANN.