Discrete Optimization for Automatic Design of Behaviors for Khepera IV Robots

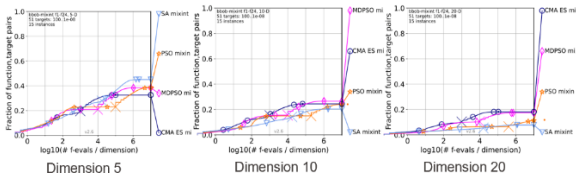
David Fontes Junqueira

Professor : Alcherio Martinoli
Assistant(s) : Cyrill Baumann

The prevalence of mobile robots in real-life scenarios has increased, requiring significant time investment from engineers to design robots with satisfactory performance. Researchers have focused on generating behaviors and controllers automatically using various machine learning and optimization methods. My project aimed to compare different optimization algorithms, for the optimization of discrete parameters in finite state machines, with the current mixed discrete particle swarm optimization (MDPSO) solution.

Being interested in nature inspired algorithms, my choice was guided towards two algorithms. The covariance matrix adaptative strategy (CMA-ES), and a classical simulated annealing algorithm (SA). A simple PSO was added to the set as a sanity check that was expected to perform equally or worse than MDPSO.

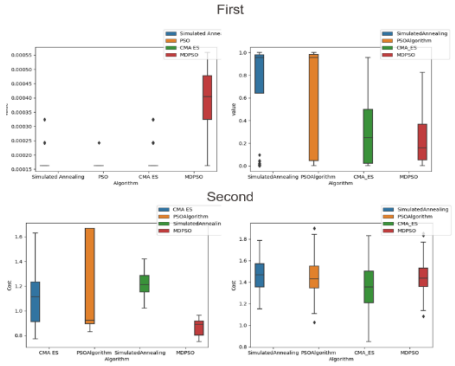
Since discrete optimization suffers a lack of available versatile implementations, I decided to use continuous optimizers and discretize inputs by rounding them to the nearest integers, along with making sure to adapt the stopping conditions accordingly. Then, I conducted a preliminary analysis of performance using the benchmarking tool COCO that presented a suite with an 80% discretized search space and optimized fully over discrete space on it, which lead to a lower general performance.



Results on the COCO benchmarks with the addition of a gaussian noise for different dimensions.

These results showed that simple implementations such as SA or PSO were struggling at higher dimensions (10,20) and under strong noise, where CMA-ES was reaching a similar performance than MDPSO.

The next steps consisted in testing the performance of those algorithms on two very noisy microscopic models that aim to reproduce a simpler version of a Webots or real-life scenario with probabilistic events. This allowed to get a better idea of the performance in scenarios closer to the intended use. A set of experiments was run for each algorithm on both models, collecting all the costs returned by the model and the best solutions provided for each experiment.



On the left: best recorded costs for 100 experiments. On the right: 100 reruns of the best solution into the model

On the model with a smaller search space, every algorithm reached the best performance, but further analysis of results demonstrated that the solutions provided by SA or PSO were less robust than the solutions provided by MDPSO or CMA-ES. Lastly, tests on the second model with a much larger search space showed that every algorithm provided a solution of comparing quality with a slight advantage for CMA-ES and MDPSO. These experiments succeeded in demonstrating mainly two key points. On one hand, MDPSO is still a very strong solution for the optimization in discrete space. On the other hand, CMA-ES is a newly appearing algorithm (2015) that is still under strong development and may be a good candidate for reaching better performances in the future with discrete-specific implementations appearing.