MathSys / Complexity mini-project:
Evolving Control Rules for Complex Systems

Background: Finding good decentralized control rules to optimise the performance of complex systems in a dynamic environment is a challenging task, because the complex system exhibits emergent behaviour that cannot be captured in a simple analytic form. One way to tackle this is by simulation optimisation: Given a simulation model of the complex system, and a space of possible control rules, a search algorithm can try to find the control rule that yields the best result on the simulation model.

A powerful class of search algorithms would be Evolutionary Algorithms (EAs), i.e., heuristics inspired by the principles of natural evolution. We have already successfully applied EAs to automatically design dispatching rules that control the production process in a semiconductor factory [1].

Mini-project: In this project, the idea of automatically generating control rules by means of an evolutionary algorithm shall be applied to more complex scheduling problems. In particular, we would be interested in environments where coordination between agents or product flows is necessary. For example, if two workers need to come together to complete a task, or if a product consists of several components, and the production of the components needs to be synchronised such that once they have been completed, they can be joined to form the final product.

We have tools for simulating production and also for the evolutionary algorithm. However, some programming in Java will be required to adapt the code.

PhD prospect: The topic is part of a research collaboration with the University in Bremen (Germay) and the Victoria University Welington (New Zealand). There are many avenues to extend this project into a full PhD project, either by further increasing the complexity of the production problem or by improving the efficiency of algorithms by integrating clever sampling techniques or surrogate models to reduce the number of computationally expensive simulation runs.

Deliverables: Control rule for the problem considered.

Student's requirements: Some programming in Java is necessary.

References

[1] Pickardt, C.; Hildebrandt, T.; Branke, J.; Heger, J.; Scholz-Reiter, B.: "Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems". International Journal of Production Economics, Elsevier, 2013