

Complexity Science mini-project: Evolving Dispatching Rules for Parallel Machines

Background: Manufacturing is a complex operation. The project considers a workstation with several parallel machines with sequence-dependent setup times, i.e., the setup time required before the machine can start working on the assigned job depends on what it has produced before. For example, if we consider a workstation that colours metal parts, no setup is required to colour a new item in the same colour as the previous one, short setup is required to change from a bright to a dark colour, but extensive cleaning is necessary when changing from a dark to a bright colour. As the jobs arrive stochastically, they need to be assigned to machines. Obviously, one would like to minimize the necessary setup costs, but at the same time jobs should not wait too long and become tardy (complete past their due date).

Genetic Programming is a heuristic inspired by the principles of natural evolution that can generate arbitrarily complex rules or programmes.

Mini-project: The goal of the mini-project would be to use Genetic Programming to evolve a rule that assigns the stochastically arriving jobs to the parallel machines such that the overall tardiness (completion of jobs after their due date) is minimized. The work would follow the lines of [1], but use better terminals. We do have a ready-to-use simulator and Genetic Programming tool in our research group, if time permits, the resulting rule could be compared to an approach based on the behaviour of ant colonies [2].

PhD prospect: The idea to evolve general rules (sometimes also called policies) that describe the best action for any situation is applicable in many complex systems. We have in the past already successfully looked at various other scheduling scenarios, and further scenarios would be to consider release policies rather than scheduling policies, or to switch to hospital management rather than manufacturing.

Deliverables:

A rule that assigns jobs to machines online.

Student's requirements:

Basic knowledge in Java might be helpful, as simulator and GP are programmed in Java.

References

[1] Jakobovic, Jelenkovic and Budin, "Genetic Programming Heuristics for Multiple Machine Scheduling", In: EuroGP 2007, p. 321-330.

[2] Campos, Bonabeau, Theraulaz, Deneubourg: "Dynamic Scheduling and Division of Labor in Social Insects", Adaptive Behavior, 2001 8(2), 83-92