EVOLUTION STRATEGIES
Poster Paper

Guenter Rudolph, chair

Co-evolution with the Bierwirth-Mattfeld hybrid scheduler

David J. John
Department of Computer Science
Wake Forest University
Winston-Salem, NC 27109, USA

Simply stated, the Job Shop Scheduling Problem (JSSP) finds a minimum time schedule given M machines and J jobs. Each job consists of a sequence of tasks, and every task requires one of the M machines for a fixed duration of time. A schedule is an assignment of all the tasks to the machines with the properties (1) the tasks for a particular job execute in the order as defined by the job, (2) no two tasks for the same job execute simultaneously and (3) no machine can execute more than one task at a time. The total time required by a schedule is called the *makespan*. The overall goal of the JSSP is to find a schedule with smallest *makespan*.

Genetic algorithm approaches to the solution of JSSP have been studied for some time. Most often the chromosome represents a permutation of the tasks according to a particular scheduling rule. Three important and well studied classes of schedulers are non-delay, active and semi-active [2]. These generate solutions that possess distinct mathematical properties. Fitness is the value of the *makespan* using the permutation with a specific scheduling scheme. The operations of mutation and crossover are specialized for the application.

Recently Bierwirth and Mattfeld [1] introduced a new scheduler with a fixed parameter δ, $0 \leq \delta \leq 1$, used to evolve good feasible schedules. The parameter provides a mechanism for the mixing of non-delay and active schedulers. This is known as the *hybrid* scheduler. Here their idea is explored further by studying the effect of co-evolving the parameter δ along with the task sequences.

Bierwirth and Mattfeld [1] experimented with different invariant values of δ for particular problems. They concluded that setting the value of δ to $\frac{1}{2}$ generally yields good solutions. Experimentally, for many job shop scheduling problems, there appear to be a number of δ values for which the hybrid scheduler does a good job. Just as important, there are many values for δ where the performance of the hybrid scheduler is poor. At this point it is not known how to compute these superior values of δ mathematically. Having a good value of δ does improve the hybrid scheduler.

A natural modification of the Bierwirth and Mattfeld hybrid scheduler in the context of a genetic algorithm permits the co-evolution of the values of δ along with the permutations of the tasks. The hybrid scheduler using the co-evolutionary genetic algorithm, *co-H*, was tested on the 188 benchmark examples found in Mattfeld [2]. This set of benchmarks consists of a mix of easy, moderate and difficult problems. For each of these examples the best known *makespan* values are indicated, as well as the method which found it. The algorithm *co-H* was run $1,000$ times on each benchmark.

The *co-H* technique was applied to all the benchmark problems that were solved previously by a collection of different algorithms. *co-H* has done reasonably well on the entire set. Just as important, *co-H* is a promising method to find reasonable *makespan* values in a relatively short number of generations. Unfortunately, this co-evolutionary technique does not escape the “large problems are much harder to solve” phenomenon observed in most genetic algorithms.

References

[1] Christian Bierwirth and Dirk C. Mattfeld. Production scheduling and rescheduling with genetic algorithms. *Evolutionary Computation*, 7(1):1–17, 1999.

[2] Dirk C. Mattfeld. *Evolutionary Search and the Job Shop*. Production and Logistics. Physica-Verlag, Heidelberg, 1996.