

# Competitive Scheduling Scenarios

*Rico Walter*

Faculty of Mathematics and Computer Science  
Friedrich-Schiller-University Jena  
07737 Jena - Germany  
ricow@minet.uni-jena.de

February 28, 2008

Scheduling as well as game theory are two classical areas in applied mathematics of great academic interest and broad practical relevance. Till now, nearly all studies on these two areas have been done independently from each other and only a few game-theoretic approaches to scheduling are known. Therefore the goal of my work is to bridge the gap between scheduling and combinatorial game theory.

For it two rather different scenarios that are more or less closely related to this issue will be presented. At first I will introduce a generalized class of subtraction games and secondly I will deal with load balancing scenarios in presence of a random decision maker.

The first - rather abstract - scheduling scenario is a generalization of the well-known two-person game called *subtraction game*. In our modified variant we keep to the classic situation but additionally consider local payoffs for every move. We assume cyclic, integer payoffs and call this game class *subtraction games with cyclic local payoffs*. In a zero-sum sense the two players try to maximize the difference between their total payoff, which is the sum of all of their received local payoffs, and the opposing total payoff. This difference is called game-theoretic value.

A question of great interest is, for which cycle lengths and sets of legal moves the sequence of the game-theoretic values becomes periodic. My work mainly concentrates on games with cycle length two and I will give a total characterization concerning periodicity for it.

The basic situation of the second scheduling scenario is the same as in load balancing problems. Given are a finite set of  $m$  identical machines and a finite set of  $n$  jobs with positive processing times that have to be processed by any of the machines. The objective of the scheduler is to find an assignment of all jobs to the machines that minimizes the maximum load (also called makespan) over all machines, where the load

of a machine is defined by the sum of the processing times of the jobs that are assigned to this machine. In case  $m = 2$  this problem is identical to the partition problem and thus NP-complete.

By adding a random decision maker we transfer the basic, deterministic situation into a situation that can be interpreted as “a game against nature” and therefore might contain some competitive aspects. The task of the scheduler is the same as above, but now he has additionally to cope with the random decisions during the assignment. To tackle this problem we use heuristic approaches based on greedy-strategies with different evaluation functions that sound promising for the deterministic case. Experimental studies show that on average such simple strategies still yield pretty good results in presence of a random decision maker.

It’s not that surprising that partly random decisions are sometimes capable of generating better solutions or even an globally optimal one. But what is more surprising is the fact, that instances exist, for which the objective function value gained by the deterministic version is an upper bound for the worst case performance of the randomized version. So to sum up, we should preferably not talk about “a game against nature” in general, because of the benefit that can be achieved in some cases.