6. Robustness considerations

6.1 Introduction to flexibility and robustness in scheduling

Scheduling is generally seen as a function with known inputs. For instance, the set of available machines is supposed to be known and the processing times of operations are supposed to be fixed. The model used for solving the problem is supposed to be the most suitable model – even if it is often a model corresponding to a simplified version of the problem ([McKay et al., 1998]). However, it is well known that *real-word scheduling problems usually are very different from the mathematical models studied by researchers in academia* ([Pinedo, 1995]).

Sometimes, the scheduler does not take care of the real application of its schedule, because this schedule is only used for simulating reasons. But in a real-world context, jobs arrive continuously, machines can break down, operators may be absent, critical tools may already been used, raw materials deliveries can be delayed, preferences of operators are not taken into account, processing times are not perfectly known, etc. When the schedule has to be applied the probability to process this schedule exactly as planned is very low.

In such a context, it is clear that decision makers have to react in real time to modify the proposed schedule, in order to always have a feasible solution and the notion of "quality" of a schedule can be discussed. The quality of a schedule is valid before the schedule becomes on line, but on line, the problem is to maintain a feasible solution, without blocking problems and if possible with a good quality. We can notice that an optimal schedule can be modified very quickly in a real time context, and can lead finally to a very bad solution, if this apparently optimal schedule was very sensitive to the perturbations that occur. On the other hand, a schedule with a "not so bad" value of the objective function may lead to a "not so bad" solution even after some unexpected events, if it was not too sensitive to the perturbations. This is the reason why searching for a compromise between the quality and the robustness of a schedule takes all the sense.

When the scheduler does not take uncertainty into account when building a solution, the proposed solution is called a *predictive* schedule and the solution approach a *predictive* approach. In order to deal with uncertainty, [Davenport and Beck, 2000] separate solution approaches into two categories: *proactive* approaches that take account of some knowledge of uncertainty, and *reactive* approaches for which the schedule is revised in real time, each time an unexpected event occurs. [Herroelen and Leus, 2005] distinguish five approaches in project scheduling, considering also stochastic scheduling, scheduling under fuziness and sensitivity analysis as possible approaches.
The aim of proactive scheduling is to make the schedule more *robust*. Several definitions ([Davenport and Beck, 2000]) have been proposed for robustness in the literature. Among others, [Billaut et al., 2005] state that *a schedule is robust if its quality is little sensitive to data uncertainties and to unexpected events*, and for [Leon et al., 1994] *a robust schedule is one that is likely valid under a wide variety of disturbances.* [Davenport and Beck, 2000] conclude that when dealing with uncertainty, *it is very likely to employ both proactive and reactive techniques.*
Robustness is related to *flexibility*, that can be seen as a freedom given in real time to the decision maker, allowing him to repair the schedule if an unexpected event or a non modeled constraint makes it infeasible. Flexibility can take several aspects ([Billaut et al., 2005]). The *temporal flexibility* allows a decision maker to start an operation earlier or latter, the *sequencing flexibility* allows the decision maker to modify or to define its own sequence of operations on a machine, the *assignment flexibility* allows to modify the assignment of an operation to another resource and finally the *mode flexibility* allows to modify the execution mode of an operation (overlapping, preemption, setup considerations, etc.) in real time.

When considering robust scheduling problems, one difficulty is to define a measure of the robustness or of the flexibility that is proposed in real time. Some approaches in the literature associate two measures to a schedule: a measure for the robustness or the flexibility, that has to be maximized and a measure for the quality of a schedule. The measure of the quality is generally a classical objective function in scheduling like makespan or maximum lateness. We focus in this chapter on the approaches in the literature dealing with robustness and consider more than one criterion. Other approaches dealing with a single criterion concerning robustness, flexibility, or stability are not presented here. The interested reader can refer to the more recent surveys of [Aytug et al., 2005] and [Herroelen and Leus, 2005].

Some of the approaches presented in this chapter propose sequential flexibility by characterizing a set of solutions. Since these approaches does not explicitly make any assumption on which uncertainties are considered and how, they are not really "proactive methods". But since the aim of these methods is to