DESIGN OF SCHEDULING ALGORITHMS


*University of Groningen, The Netherlands
**CNRS, Nantes Research Institute in Communications and Cybernetics, France
***Nantes University, France
****Cyprus College, Cyprus

ABSTRACT
The academic field of production research has been growing rapidly over the last decades with researchers proposing numerous analytical and heuristic optimization methodologies for the solution of planning & scheduling problems. However, adoption by manufacturing companies is lagging behind. This paper suggests that the basic reason behind this imbalance is the inadequate representation of the planning & scheduling process when designing decision support systems. Hence, the algorithms that are designed and included in these systems might not reflect the problems that actually have to be solved in practice. In this paper we discuss the basic factors that are important for the development of planning & scheduling decision support systems. These factors will be based on insights from cognitive psychology, computer science, and operations management.

Keywords: scheduling, algorithms, cognitive science

INTRODUCTION
The accomplishment of a manufacturing company’s objectives is strongly connected to the efficient solution of complex scheduling problems that are faced in the production environment. The academic field of production research has been growing rapidly over the last decades with researchers proposing numerous analytical and heuristic optimization methodologies for the solution of scheduling problems (Slack et al., 2004). However, very few of them have been extensively adopted by manufacturing companies. The basic reason behind this imbalance is the inadequate representation of the very complex scheduling process, as this is implemented in practice.

The main aim of this chapter is to provide a theoretical discussion on the design of scheduling algorithms. In the first part of this discussion a detailed description of the general problem-solving process will be presented. The second part will concern a critical review of traditional production research algorithmic design approaches. This review will contrast the model of the scheduling environment as this is conceived by traditional production research approaches against models that specifically address human and organizational issues. The final part of this chapter builds on the discussion of the previous sections and proposes the development of a scheduling theoretical framework. This framework will help practitioners to design decision support tools that specifically address human and organizational considerations of the scheduling case considered.

PROBLEM SOLVING FUNDAMENTALS
Modeling a planning/scheduling problem and designing an algorithm to support the process of solving the problem is rather complicated. This process is sensitive to errors, assumptions, and perceptions,
which might result in unsatisfactory solutions. This section gives an overview of problem solving fundamentals in order to improve our understanding of the role of algorithms and the process of designing such algorithms. First, we will give attention to a systems view on problem solving. Next, a cognitive psychology perspective of problem solving will be presented.

From a systems perspective, problem solving can be described as a process consisting of several stages. A landmark paper in this respect is Mitroff et al. (1974). They distinguish four elements and six stages (see figure 1) and argue that many problem-solving processes differ in terms of the starting point and the subset and sequencing of stages taken into account.

It might seem that the regular way of problem solving is to start in I and follow the stages 1, 2, 3 and 4, while giving attention to the validation steps 5 and 6. However, Mitroff et al. argue that -from a systems perspective- all elements can be a suitable starting point for a fruitful problem solving process. If, for example, a standard solution is being applied that seems not appropriate anymore, a problem solver might consider the scientific model (III) that was behind the standard solution or directly criticize the conceptual model (II) that has been used. Hence, different paths can be followed in order to solve a problem.

The next fundamental issue they raise is that each element and stage requires different skills of people involved. Activities in the right hand side of the figure are said to require formal, analytic skills and people that excel in these skills are, according to Jung, denoted as Thinking types. Activities at the left hand side require intuitive thinking and human relations skills. Jung denotes people that excel in these activities as Intuitive and Feeling types. There are not that many persons that excel in both set of skills (Jung, 1976).

From a cognitive psychology perspective, a landmark in the field of problem solving is the work of Herbert A. Simon and Allen Newell, starting with their 1958 paper on human problem solving (Newell et al., 1958). Problem solving is seen as the interaction of a problem solver with the problem’s state space (denoted as task environment). The type of problems that they primarily consider are transformation problems consisting of an initial state, a goal state, and the rules of the game (available options and restrictions). They distinguish sub-goal setting and strategy selection as the main activities in problem solving, and focus on the differences in memory storage and usage in the solution process. Newell and Simon see planning, in accordance to Miller et al. (1960), as a hierarchically controlled process in which several cognitive mechanisms, such as working memory, executive control, and
knowledge representation co-operate. The working memory is used to store and retrieve plans when they are being generated or executed (1960: 207).

In order to solve a problem, Newell and Simon (1972) state that a representation of the transformation problem needs to be developed or selected. A representation includes: (1) a description of the current state, (2) operators/actions in order to change/transform the current state, including the constraints/limitations, and (3) tests in order to verify that the achieved state corresponds to the goal. The problem state space constitutes of all achievable (intermediate) states. The problem state space is generally very large, and applying operators as well as testing for verification costs time. Therefore, it requires efficient and effective procedures to search and test solutions. These procedures determine the search space, which may or may not contain the best solution available. Dynamic memory usage of such procedures as well as the way to resolve conflicting possibilities need attention.

An alternative view of problem solving is developed by Hayes-Roth and Hayes-Roth (1979). Their main criticism is that a cognitive theory of problem solving should start with the actual behavior of problem solvers. They found that humans behave opportunistically when solving a problem. While dealing with a problem, they use information and anticipate on future situations and behave accordingly. Problem solving is seen as a process in which opportunities that emerge are as important for the decisions to be made as predetermined subgoals and sequences of operations. This view of problem solving focuses more on the cognitive processes during the execution or implementation of a plan and seems to fit better in case of ill-structured problems, where at least one of the three parts of a problem representation is not completely available.

An important category of these ill-structured problems are so-called design problems (Simon, 1981). Design problems occur when creativity is required, i.e. use of pattern recognition and reuse of already developed solutions to similar problems are not sufficient to solve the problem. The problem state space is infinite in such problems, and both initial state and goal state might be difficult to represent in an unambiguous and complete description. However, decomposition of such problems and applying search strategies may still be worthwhile, as empirical research has shown (Ormerod, 2005).

**Algorithms**

Procedures for searching and testing solutions are denoted as algorithms. An algorithm for solving a problem determines how searching and testing will be done. Algorithms may differ in the efficiency and effectiveness in which they operate. Less efficient algorithms take more time or storage memory than efficient algorithms. In general, this is denoted as the computational complexity of an algorithm. As the runtime or storage requirement normally depends on the size of the problem, computational complexity is expressed as a function of a suitable input measure of the problem, e.g., the number of jobs $n$ that need to be scheduled, or the number of machines $m$ that have to be planned.

A problem is said to be polynomially solvable if an algorithm exists for which the computational complexity can be expressed as a polynomial in a suitable input measure. The idea is that if problem sizes increase, time or storage requirements will also increase, but according to a polynomial function. Less effective algorithms perform less in finding or testing satisfying solutions. Some algorithms do not even find a solution, others are not able to guarantee that the solution that has been found is of sufficient quality. The latter are often denoted as heuristics. The effectiveness of heuristics can be assessed through a worst-case analysis. Note that the effectiveness of an algorithm depends on the formulation of the goal. If the goal is to find the optimal solution (i.e. no better solution is available in terms of the objective function), the algorithm should not only find a solution in the set of optimal solutions, but also prove that there is no better solution. If the goal is rephrased to finding a solution with an objective function value larger than $z$, any solution that satisfies this criterion qualifies, and the algorithm will be much more efficient as it does not have to proof uniqueness or optimality.
This brings us to the last remark on algorithms, a remark with respect to their robustness. Scheduling problems are characterized according to a three field scheme $\alpha|\beta|\gamma$, where the $\alpha$ field describes the set of resource characteristics, the $\beta$ field the input characteristics, and the $\gamma$ field the objective function. For example, a flow shop scheduling problem with two machines (i.e. all jobs visit the machines in the same sequence), $n$ jobs and objective function “minimize the make span” is represented as $F2|n|C_{\text{max}}$. For this problem, Johnson (1954) found a polynomially solvable algorithm. However, if the problem is slightly changed to a three-machine flow shop scheduling problem with the same objective, Johnson’s algorithm no longer solves this problem optimally. Garey et al. (1976) even proved that this problem is NP complete. Some algorithms are therefore strongly connected to a single model and often not optimal for a related problem. Applying it to such a problem might result in near-optimal results, but a very disappointing result might also be possible. They may even be inapplicable to such a problem, hence providing no solution at all. Other algorithms are more robust if changes in the characteristics of the problem occur. For example, simple heuristics, such as hill-climbing or an Earliest Due Date dispatching rule, can be applied to a large set of problems, although not always generating good quality solutions, and hence are much more flexible to changes in problem characteristics than specific tailor-made algorithms. The latter might be quicker or better in the process of finding good solutions. We see a similar design problem in nature. Some insects are very specialized on a type of plant for their food consumption, while others are not sensitive for changes in the availability of specific natural resources. The design of algorithms is hence the result of a decision process in which many factors have to be taken into account.

Improving problem solving

Many attempts have been made in the past in order to improve the result of the process of problem solving, for example, problem structuring improvements (what aspects of the problem situation are relevant), modeling improvements (multi-objective models, constraint programming, combinatorial models), technical improvements (the technique used by the algorithm, i.e. the programming language or the hardware platform), skills improvements (better training of planners in using the tools), and social improvements (improve acceptance of outcome, reduce resistance). These attempts can be characterized according to the scheme of Mitroff et al. (1974) (see figure 1). Problem structuring improvements focus on stage 1. Modeling improvements focus on stage 2. Technical improvements focus on stage 3. Skills and Social improvements focus on stage 4.

Problem structuring. Many attempts to improve problem structuring were initiated after the criticism of Ackoff (1978, 1979) on the developments in the Operations Research community appeared. Flood and Jackson (1991) constructed a system of systems methodologies, that gives much attention to problem identification in terms of systems and participants, and describe various approaches, such as Operations Research, Interactive planning, Soft Systems Methodology, and Critical Systems Heuristics. Note that their grouping of these methodologies is based on the (often implicit) assumptions that the methodologies make about the problem context. Examples of such assumptions might concern the moment that information will be available for problem solving, the moment that all relevant stakeholders are identified, the presence of ambiguity in the interpretation of problem and its relevant aspects, and the role of ratio, politics, and power in defining a problem and accepting a solution. From this list it will be clear that efforts to improve problem structuring are strongly related to reconsidering traditional assumptions with respect to the problem and its context.

Model building. Improvements in model building during the last decade have shown a huge increase in new methods that better fit with other representations of a problem. Traditionally,
Operations Research has focused on optimizing single objective deterministic problems. Heuristics were considered to be of limited value, at least from a scientific point of view. However, many engineers and computer scientists have started working at other solution approaches, such as Constraint Programming (Baptiste et al., 2001), Simulated Annealing (Laarhoven and Aarts, 1987), Tabu Search, Neural Networks, Genetic Programming (Pham and Karaboga, 1998, Gen and Cheng, 2001), and Evolutionary Algorithms (Coello Coello, 2006). Characteristics of the problems for which these approaches are suitable are satisfying goals instead of optimizing, and large problem spaces suitable for mimicking search patterns that are found in nature. Other improvements have focused on multi-objective instead of single-objective problem solving. In a pluralistic or even coercive problem context, various stakeholders will have different goals and objectives. These objectives might even be contradictory. In order to take multiple objectives into account in a (mathematical) model, several possibilities have been explored. First, a weighted function of the various objectives could be used. However, this raises the problem of scaling, weight parameter setting, and interpretation of the solution. Next, the notion of dominance of solution was introduced. The concept of Pareto optimality as a dominance criterion has been proposed most frequently (Coello Coello, 2006). The idea of this concept is that a solution is not dominated until another solution has been found that improves the performance on all objectives that have to be taken into account. The set of non-dominated solutions is therefore in practice considerably large.

Finally, many improvements have been realized in the field of non-deterministic problem solving, although still a huge number of papers that is being published mainly focuses on deterministic problems. An important characteristic of non-deterministic problems is that the situation at the moment of taking the decision is not assumed to be known in advance. The future is therefore modeled as uncertain. Based on this incomplete knowledge, still a good solution has to be proposed. The question is what will be considered a good solution. Literature has introduced the notion of risk preference of the decision maker in order to cope with this issue. The goodness of fit of a solution depends on the preference for risk of the decision maker / problem owner. Some try to avoid risks, others focus on management of risks, and some even seek risks. Dorfman (2007) distinguishes four basic positions: either tolerate, treat, terminate, or transfer risks. Traditionally, models have mainly focused on risk avoidance. New directions are to handle other risk mitigation strategies as well.

Using a model should be possible not only because of the foreseeable future at the moment of decision making/selecting a solution, but also because of the time available for building a model and finding a solution. If the time available is too small to do both, one can either decrease the time for model building or for model solving. Recent developments in Advanced Planning Systems show that it is sometimes possible to reduce the time needed for model building by using intelligent software that enable the use of standard OR techniques (Pochet and Wolsey, 2006).

Model solving. Several attempts have been made in order to improve the process of solving a model (step 3 of figure 1). The efforts invested in finding algorithms with a lower time-complexity for some well-defined problems are still huge. The same holds true for the improvements in the speed of computers, programming languages, parallel processing, et cetera. These improvements are strongly related to the fields of Informatics and Computer Science.

From a cognitive perspective, Wezel and Cegarra (2007) discuss possible roles of human in solving models for multi-actor problem situations, where humans co-operate with machines (i.e. combinations of hard and software) in order to make a schedule. Various tasks need to be accomplished by both human and machine in order to make a good schedule. Some of these tasks can be performed without interaction with another actor, some need to be performed in co-operation, because of strong interdependencies and/or a high impact on the quality or acceptability of the resulting plan. This issue will be explored further in the following section.
Implementation. Finally, large investments have been made in improving skills of problem solvers/decision makers. Planners, for example, have received more training in using computerized planning systems as well as the organizational context of planning problems. Due to the introduction of participative techniques in problem structuring, stakeholders are involved earlier, which might affect the acceptance of solutions. Nevertheless, many people think that a problem is solved at the moment the model has identified a solution. Such a limited view of problem solving will stay a stumbling block on the road to improving problem solving.

DESIGNING ALGORITHMS FOR SCHEDULING PROBLEMS: A CRITICAL REVIEW
In the previous section, a rigorous description of theoretical issues related to the general problem-solving process was provided. This section takes a closer look on the procedure of designing algorithms for the solution of scheduling problems. As discussed in the previous section, the design of a scheduling algorithm is mainly related to stages 2 and 3 of Mitroff’s problem-solving model (figure 1). As it has been observed, scheduling algorithms designed through the traditional production research approach are rarely favored to ad-hoc approaches in realistic industrial environments (Berglund and Karltn, 2005). This section examines in detail the traditional approach provides a critical review of its shortcomings. It then proceeds to discuss the characteristics of the scheduling process as it happens in practice, and proposes the development of a new theoretical framework for the design of scheduling decision support systems that explicitly consider the human and organizational factors within a scheduling environment.

3.1 The scheduling environment: the traditional production research view
As it has been discussed by various researchers, the traditional production research approach to the design of scheduling algorithms has rarely made an impact in realistic industrial environments (Portougal and Robb, 2000; Fransoo and Wiers, 2005), (Berglund and Karltn, 2005). In practice, the majority of schedulers assume full control of the scheduling process by employing ad-hoc solution approaches. While custom-built IT decision support systems are occasionally used to support the schedulers’ work, only a limited number of fully automated scheduling systems exist. In order to examine the reasons for the lack of practical use for algorithms designed through the traditional production research approach, it is important to examine the nature of the scheduling process as it is assumed by such a problem-solving process. The following notion provides a basis for this discussion. In general, the traditional production research view of the scheduling environment assumes that a scheduling task defined by a rigorous mathematical model has to be implemented at a specific moment in time, or at well-defined intervals (Meredith, 2001). This view magnifies the importance of the scheduling algorithm within the scheduling environment and degrades the implications of human and organizational considerations.

In terms of the cognitive aspects of the process, the human scheduler is considered a black box in the scheduling environment. Her/his participation in the overall process is implied, but is not explicitly considered in terms of the mathematical model. In principle, the presence of a human scheduler is not even required, since there is no reference to the source of the scheduling information that is used as input data by the scheduling algorithm. As a result, the interaction between the human scheduler and the scheduling algorithm is given no special consideration. The design of the Human Computer Interaction (HCI) environment for the algorithm is based on the skills and the intuition of programmers, rather than the use of an appropriate scientific methodology.

The organizational structure of the scheduling environment is modeled as an automated flow line process. The necessary scheduling information is provided to the scheduling algorithm by unknown sources, however, this information is always considered to be in the required format as well
as timely and accurate. The algorithm generates schedules in a format that is assumed to be meaningful and understandable by all parties who receive them. Manual or automatic editing of generated schedules is not considered to be part of the decision-making process. The possible existence of organizational structures within the scheduling environment, such as a team of human schedulers, and their relationship with neighboring organizational structures such as the planning department and the shop-floor environment are also excluded from consideration.

The IT infrastructure that supports the scheduling process is considered to be simple and is based on the existence of an autonomous computing facility required for the execution of the algorithm. The technical specifications of the electronic data which are used and generated by the algorithm are assumed to be consistent with the specifications that are used by neighboring departments within the manufacturing environment.

Given Mitroff’s view of problem-solving, it can be said that the traditional production research approach to scheduling starts with the construction of the scientific model (step III), implicitly assuming that a standard solution can be applied successfully to the problem considered. Thus, the problem-solving process reduces to the development of an algorithm that attempts to optimize an objective for the rigid mathematical representation of the scheduling environment.

This critical view of the traditional process does not imply that the algorithms designed through the traditional process are not efficient. In fact, many of these algorithms incorporate elements that address specific human and organizational considerations, such as the need for flexibility in generated schedules and the existence of multiple conflicting optimization objectives. However, it is very difficult to develop accurate mathematical models of the problem environment for practical scheduling problems. More importantly, there exists a trade-off between the accuracy of the developed mathematical model and the computational complexity of the solution algorithm. Since the development of mathematical models that explicitly consider all human and organizational factors in a scheduling environment is a difficult (if not impossible) task, the implemented algorithms are rarely used in practice, at least in the manner envisaged by their designers.

The question that naturally arises from the observations of the previous paragraphs is the following: How can we improve the problem-solving process of scheduling problems in a way that will provide useful support to the human scheduler in realistic production environments? In order to answer this question it is necessary to examine the scheduling environment from a realistic perspective.

3.2 The scheduling environment: the realistic view
In contrast to the view of the traditional production research approach, in a realistic industrial environment the scheduling process is a complex interpersonal and interdepartmental process that takes place dynamically over time (figure 3). The human scheduler resides at the heart of the scheduling environment, since s/he processes or communicates the majority of scheduling information needed for the implementation of the process. The human scheduler generates schedules through a cognitive process which can be assisted by the existence of decision support tools. These tools are not necessarily in a software form. If the implementation of the scheduling process requires the cooperation of a group of schedulers, rather than the skills of a single human operator, the complexity of the overall process increases further.

From an organizational point of view, scheduling information flows in the environment from various sources (employees, departments, other IT systems) and in various forms (verbal communication, paperwork, electronic data). The scheduling algorithm is a part of the software system that provides support in the decision making process. It receives input information either from the scheduler(s) or from other software systems and database facilities. The algorithm generates schedules based on the scheduling information provided; however, these schedules are not necessarily accepted
by the human scheduler or group of schedulers. The finalized schedule might be the result of a manual editing process that takes place based on personal experience, group experience, departmental negotiations and unexpected events.

The conceptual model depicted in figure 3 provides a more realistic representation of the complexity of the scheduling process in relation to the traditional production research view. Still, it does not portray the dynamic human and organizational factors that can seriously disturb its implementation. Some of these factors are illustrated in figure 4.

Since the human scheduler plays the central role in the overall process, his/her personal skills as well as his/her physical and psychological condition are of great importance to the smooth implementation of the decision making process. Inadequate training, tiredness, personal problems, poor communicational skills, limited IT skills, inability to handle pressure and lack of motivation are just some of the factors that might have an effect on the cognitive process of generating, editing and deciding on the suitability of schedules.

This cognitive process must also take into account the existence of various objectives within the scheduling environment, which may have conflicting nature. The term ‘objective’ is used here not only as a description of a rigorously defined mathematical function that needs to be optimized, but in the more general sense. These objectives can be attributed to various sources (scheduler’s objectives, departmental objectives, managerial objectives) and they can have different time horizons (short, medium and long-term objectives). As a result, it is extremely difficult to accommodate them all within the context of a typical algorithmic design. The process of generating schedules is heavily influenced by the existence and the prioritization of scheduling objectives within the production environment. Especially the existence of contradicting high-level objectives of the departments can lead to tensions and generate mistrust on the priority of jobs that need to be scheduled.

Another factor that plays a crucial role on the successful implementation of the cognitive process is the design of the Human-Computer Interaction environment between the scheduler and the software-based decision support system. An HCI design that is not based on a rigid design methodology can lead to misunderstandings, erroneous decisions, and most importantly distrust by the human operators.

From an organizational point of view, the flow of scheduling information between all parties involved is not necessarily timely or accurate, since, as explained earlier, this information originates...
from various sources and can take different forms. The existence of scheduling data in both handwritten/printed and electronic format can generate significant problems since the human operators will have to provide this information to the decision support system manually. Erroneous data input will generate erroneous schedules and will subsequently lead to poor decision making.

These observations lead us back to the question how we can improve the problem-solving process of scheduling problems in a way that will provide useful support to the human scheduler in realistic production environments. Although there cannot be a simple or easy answer to this question, a potential approach to a realistic design methodology is described in the following section.

The way forward: a framework for designing scheduling algorithms with human and organizational considerations

The complexity of the scheduling process as described in the previous paragraphs does not imply that the design of a scheduling algorithm is an unnecessary or impossible task. It indicates though that if these algorithms are to be employed within the context of a decision support system in realistic industrial environments, their design should be based on a new approach.

The development of this new approach can benefit from the analysis of the scheduling process that was described previously. This analysis established the following facts: (1) Each scheduling problem is unique and has its own particular human and organizational characteristics, (2) The development of a mathematical model that incorporates all human and organizational characteristics of the scheduling process is most of the times not possible or even necessary (due to the mathematical complexity of the resulting model), and (3) A scheduling algorithm is only a part of an overall decision support system that will be used by a human scheduler. What has become evident from the analysis is that there is a need for a robust development process which will drive not only the design of the scheduling algorithm, but also the design of the overall decision support system that will function within the scheduling environment. We propose the development of a theoretical framework for the design of scheduling decision support systems that will explicitly consider human and organizational considerations of the scheduling case considered. The proposed framework should contain the following information:

1. A categorization of cognitive, organizational and technical attributes of production scheduling processes.
2. A categorization of cognitive, organizational and technical characteristics explicitly considered by existing scheduling algorithms.
3. Recommendations on the type of core algorithms that should be employed for the design of a production scheduling decision support system based on the cognitive, organizational and technical characteristics of the scheduling case considered.
4. Guidelines on the design of new algorithms (or the modification of existing ones) that will address cognitive, organizational, and technical considerations not currently handled by existing scheduling algorithms.
5. Guidelines on the design of interactive computing environments (graphical user interfaces) that will address the cognitive, organizational and technical considerations of the human operator while performing the required scheduling tasks.

The use of this framework implies that the design of a decision support system should start with an examination of the human, organizational, and technical characteristics of the scheduling case considered. This information can be obtained and analyzed using appropriate methodologies such as interviews, observations, context diagrams, data flow diagrams, as well as hierarchical and cognitive task analysis. The designer of the decision support system can then match the results of the analysis to the guidelines of the theoretical framework and proceed accordingly. This process might lead to the use
of an existing general-purpose algorithm with a very simple user interface, but may also lead to the
design of a new complex case-based algorithm with high-end Graphical User Interface provisions,
depending on the analysis of the scheduling case considered.

CONCLUSIONS
The design of a scheduling algorithm is a complex process that should not be solely based on the
development of a scientific model for the scheduling problem considered. Human and organizational
characteristics of the scheduling environment play a significant role on the practical adoption of a
scheduling decision support tool.

This chapter discussed in detail the process of designing algorithms for the solution of
scheduling problems. A rigid description of the general problem-solving process was initially provided
with references to Mitrov’s model as well to models from the field of psychology. This was followed
by a description of the alternative types of algorithms that exist and their corresponding characteristics.
The first part of this chapter was concluded with an analysis of the possible improvements that can be
applied to each step of the problem-solving process, as this is conceived by Mitrov.

The second part of the chapter focused on the scheduling problem-solving process. It discussed
and reviewed the traditional production research algorithmic design process using as a tool the model
of the scheduling environment employed by such processes. It then contrasted this model with the
realistic view of the scheduling environment as it happens in practice pointing out the important human
and organizational considerations which are not addressed by the traditional production research design
approach.

Based on the findings of this analysis, a possible development of a theoretical framework which
can be used on a case-by-case basis for the design of scheduling decision support tools has been
discussed. This framework will provide guidelines on both the design of scheduling algorithms and
their associated graphical user interface. The aim is to develop not just scheduling algorithms, but
integrated support tools that will address human, organizational, as well as technological considerations
of the scheduling environment considered.

As it is obvious, the development of such a framework requires the completion of large-scale
psychological, organizational and technological studies that will provide the respective framework
information. The authors of this chapter aim to work towards the development of such a framework in
the future.

REFERENCES
Baptiste, P., Le Pape, C., Nuijten, W., 2001, Constraint-Based Scheduling: Applying Constraint Programming to
Heidelberg.
Berglund, M., Karlton, J., 2005, Human, Technological and Organizational Aspects Influencing the Production Scheduling
Process, in Proceedings of the International Conference of Production Research (ICPR ’05), Salerno, Italy.
Coello Coello, C.A., 2006, Evolutionary multi-objective optimization: a historical view of the field, IEEE Computational
International Conference of Production Research (ICPR ’05), Salerno, Italy.
San Francisco.

Johnson, S.M., 1954, Optimal two-and-three-stage production schedules with set-up times included, Naval Research Logistics Quarterly, 1, 61-68.


Mitroff, I.I., Betz, F., Pondy, L.R., Sagasti, F., 1974, On managing science in the systems age: two schemas for the study of science as a whole systems phenomenon, Interfaces, 4, 3, 46-58.


Portougal, V., Robb, D.J., 2000, Production Scheduling Theory: just where is it applicable?, Interfaces 30:6, 64-76.

