

Available online at www.sciencedirect.com



European Journal of Operational Research 165 (2005) 289-306



www.elsevier.com/locate/dsw

Project scheduling under uncertainty: Survey and research potentials

Willy Herroelen^{*}, Roel Leus¹

Operations Management Group, Department of Applied Economics, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium

> Received 1 May 2002; accepted 1 June 2003 Available online 1 June 2004

Abstract

The vast majority of the research efforts in project scheduling assume complete information about the scheduling problem to be solved and a static deterministic environment within which the pre-computed baseline schedule will be executed. However, in the real world, project activities are subject to considerable uncertainty, which is gradually resolved during project execution. In this survey we review the fundamental approaches for scheduling under uncertainty: reactive scheduling, stochastic project scheduling, fuzzy project scheduling, robust (proactive) scheduling and sensitivity analysis. We discuss the potentials of these approaches for scheduling under uncertainty projects with deterministic network evolution structure.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Project management and scheduling; Scheduling under uncertainty; Robustness; Schedule stability

1. Introduction

The project scheduling literature largely concentrates on the generation of a precedence and resource feasible schedule that "optimizes" the scheduling objective(s) (most often the project duration) and that should serve as a baseline schedule for executing the project. Such a *baseline* schedule (also called a predictive schedule or preschedule) serves very important functions (Aytug et al., in press; Mehta and Uzsoy, 1998). The first is to allocate resources to the different activities to optimize some measure of performance. The second, as also pointed out by Wu et al. (1993), is to serve as a basis for planning external activities such as material procurement, preventive maintenance and delivery of orders to external or internal customers. Baseline schedules serve as a basis for communication and coordination with external entities in the company's inbound and outbound supply chain. Based on the baseline schedule, commitments are made to subcontractors to deliver materials, support activities are planned (setups, supporting personnel), and due dates are set

^{*}Corresponding author. Tel.: +32-16-326970; fax: +32-16-326732.

E-mail address: willy.herroelen@econ.kuleuven.ac.be (W. Herroelen).

¹ The work was performed while the second author was Research Assistant of the Fund for Scientific Research, Flanders (Belgium) (F.W.O.).

for the delivery of project results. Moreover, from a modelling viewpoint, many real-life scheduling problems such as course scheduling, sports timetabling, railway and airline scheduling, can be modelled as variations of resource-constrained project scheduling problems. In these environments executing activities according to the preschedule is a must that is imposed by the customer: although "technically" possible, activities are not started prior to their scheduled starting time.

During project execution, however, project activities are subject to considerable uncertainty that may lead to numerous schedule disruptions. This uncertainty may stem from a number of possible sources: activities may take more or less time than originally estimated, resources may become unavailable, material may arrive behind schedule, ready times and due dates may have to be changed, new activities may have to be incorporated or activities may have to be dropped due to changes in the project scope, weather conditions may cause severe delays, etc. A disrupted schedule incurs higher costs due to missed due dates and deadlines, resource idleness, higher work-in-process inventory and increased system nervousness due to frequent rescheduling. As a result, the validity of static deterministic scheduling has been questioned and/ or heavily criticised (Goldratt, 1997).

In general, we can distinguish between five approaches to dealing with uncertainty in a scheduling environment where the evolution structure of the precedence network is deterministic: reactive scheduling, stochastic scheduling, scheduling under fuzziness, proactive (robust) scheduling, and sensitivity analysis. In this paper we will discuss these approaches mainly from a project scheduling viewpoint. In those situations where the approaches were clearly conceived in a machine scheduling context, our aim is to reveal their potentials for scheduling projects under uncertainty. Stochastic project networks that have a stochastic evolution structure and feedback (GERT networks) are not the subject of this paper. A state-of-the-art survey of GERT network scheduling can be found in Neumann (1999).

The paper is organised as follows. In the next section, we survey the research efforts in the field of reactive scheduling. In Section 3 we present a

classification scheme for schedule construction techniques under uncertainty. Stochastic project scheduling is discussed in Section 4. Section 5 is devoted to fuzzy project scheduling. In Section 6 we characterize robust baseline schedules and review various robustness/stability measures as well as methods for generating robust and stable schedules that may have potential application for scheduling projects under uncertainty. Sensitivity analysis is discussed in Section 7. A summary and suggestions for further research conclude the paper.

2. Reactive scheduling

Reactive scheduling does not try to cope with uncertainty in creating the baseline schedule but revises or re-optimizes the baseline schedule when an unexpected event occurs. Basically most efforts concentrate on "repairing" the baseline schedule (*predictive-reactive scheduling*) to take into account the unexpected events that have come up. For a review of the extensive literature in manufacturing environments we refer to Sabuncuoglu and Bayiz (2000), Szelke and Kerr (1994) and Vieira et al. (2003).

The reactive scheduling action may be based on various underlying strategies. At one extreme, the reactive effort may rely on very simple techniques aimed at a quick schedule consistency restoration. We shall refer to these approaches as *schedule repair* actions. A typical example of such a simple control rule is the well-known *right shift rule* (Sadeh et al., 1993; Smith, 1994). This rule will move forward in time all the activities that are affected by the schedule breakdown because they were executing on the resource(s) causing the breakage or because of the precedence relations. It should be clear that this strategy may lead to poor results as it does not re-sequence activities.

At the other extreme, the reactive scheduling approach may involve a full scheduling pass of that part of the project that remains to be executed at the time the reaction is initiated. Such an approach will be referred to as *(full) rescheduling* and may use any deterministic performance measure, such as the new project makespan. In a sense, schedule repair is a heuristic rescheduling pass. If our objective were to generate a new schedule that deviates from the original schedule as little as possible, we would be in the particular rescheduling case where we want to induce *ex post stability* (the ex ante case will be discussed in the next section). Such a *minimum perturbation strategy* may rely on the use of exact and suboptimal algorithms using as objective function the minimization of the sum of the (weighted) absolute differences between the start time of each activity in the repaired schedule and the original start time of that activity (El Sakkout and Wallace, 2000). Alagöz and Azizoglu (2003) study the case in which the stability measure is the number of jobs processed on different machines in the initial and the new schedule. Calhoun et al. (2002) use goal programming to revise project schedules with the initial objectives and the objective of minimizing the number of changed activities. Match-up scheduling matches up with the pre-schedule at a certain time in the future, whenever a deviation from the initial parameter values (mainly deviations from the activity duration projections) arises (Bean et al., 1991; Wu et al., 1993; Akturk and Gorgulu, 1999; Alagöz and Azizoglu, 2003).

Artigues and Roubellat (2000) study the case where, in a multi-project, multi-mode setting with ready times and due dates, it is desired to *insert a new unexpected activity into a given baseline schedule* such that the resulting impact on maximum lateness is minimized. The authors perform a clever rescheduling pass in which they restrict the solution to those schedules in which the resource allocation remains unchanged. Using a resource flow network representation they develop a stepwise procedure for generating a set of dominant 'insertion cuts' for the network. From each dominant insertion cut, they then derive the best execution mode and valid insertion arc subset for the new activity.

3. Generating a baseline schedule

The first column of Table 1 distinguishes between three basic approaches for the development of a baseline schedule. In the first approach, *no*

Table	1				
-					

Different methods fo	r schedule	generation	under	uncertainty
		8		

Baseline schedule	During project execu- tion
(i) No baseline schedule	(i) Dynamic scheduling (scheduling policies)
(ii) Baseline scheduling with no anticipation of variability	(ii) Reactive scheduling
(iii) Proactive (robust) scheduling	(iii) Management
Quality robustness	decisions
Solution robustness Flexibility	(iv) Sensitivity analysis

baseline schedule is generated. In the second scheme, a baseline schedule is developed using a deterministic scheduling method without any anticipation of variability in the input parameters that may occur during project execution. Single point estimates are used for parameters such as activity durations. The third approach is to develop a baseline schedule that *incorporates a degree* of anticipation of variability during project execution. This setting will be referred to as proactive or robust scheduling. This approach may use information about the particular variability characteristics (for example probability distributions for activity durations) and/or information about the reactive scheduling approach that will be adhered to during project execution (mostly very simple repair operations). The special case where the baseline objective is to minimize a function of the deviation between the baseline and the final schedule, focuses on ex ante stability. Often the term quality robustness is used when referring to the sensitivity of the schedule performance in terms of the objective value, while the term stability or solution robustness is used to refer to the insensitivity of the activity start times to changes in the input data. Robustness is closely related to *flexi*bility (Sevaux and Sörensen, 2002b). A schedule is called flexible if it can be easily repaired, i.e. changed into a new high quality schedule. The informal French association of researchers in scheduling GOThA (Groupe de recherche en Ordonnancement Théorique et Appliqué-http:// www-poleia.lip6.fr/~sourd/gotha/) has established a "Flexibility working group" that regularly reflects on how to define, measure and use

flexibility and also maintains a web page listing recent references (http://www.loria.fr/~aloulou/pages/biblio_gotha.html).

A second distinction can be made with regard to the way in which decisions will be taken during project execution on how to react to disruptions and when to start new activities. Three possibilities are listed in the second column of Table 1: (i) no baseline schedule is generated, but before the start of the project, a scheduling policy is chosen that will determine how to act during schedule execution; (ii) when a baseline exists, we reschedule, using any of the options that were discussed in the previous section (*predictive-reactive scheduling*); and (iii) instead of using preset scheduling policies, project management makes decisions as the project develops.

Apart from these methods for construction of the final schedule, techniques have also been devised to provide the project manager with information about *allowable* deviations in project parameters, which will aid the manager in determining which parts of the project require the most attention (the inherent assumption is that the sources of uncertainty are more or less manageable). *Sensitivity analysis*, to be discussed in Section 7, is a clear example of such an approach.

4. Stochastic project scheduling

The literature on stochastic project scheduling is rather sparse (for a detailed discussion, see Chapter 9 in Demeulemeester and Herroelen, 2002). Most efforts concentrate on the so-called *stochastic resource-constrained project scheduling problem* which will be discussed in Section 4.1. The related special case of stochastic activity interruptions, time/cost trade-off problems and stochastic multi-mode problems are the subject of Sections 4.2, 4.3 and 4.4, respectively.

4.1. Stochastic resource-constrained project scheduling

The stochastic resource-constrained project scheduling problem aims at scheduling project activities with uncertain durations in order to minimize the expected project duration subject to zero-lag finish-start precedence constraints and renewable resource constraints. The project is represented by an activity-on-the-node network G = (V, E), where the set $V = \{1, 2, \dots, n\}$ denotes the set of activities. Activity 1 and *n* are dummy activities, representing the start and end of the project. The durations of the other activities are given by a random vector $\boldsymbol{d} = (\boldsymbol{d}_2, \boldsymbol{d}_3, \dots, \boldsymbol{d}_{n-1}),$ where d_i denotes the random duration of activity *i*. We denote a particular *realization* or *sample* of *d* as $d = (d_2, d_3, \ldots, d_{n-1}) \in \mathfrak{R}^n_+$. The arcs of set E define the zero-lag finish-start precedence relations among the activities. The renewable resources $(k = 1, 2, \dots, K^{\rho})$ are available in constant integer amounts a_k^{ρ} . The non-dummy activities require an amount of $r_{ik}^{\rho} \leq a_k^{\rho}$ units of renewable resource type k. Given the presence of both resource constraints and random activity durations, schedules are generated through the application of so-called scheduling policies or scheduling strategies, and no baseline schedule is used.

According to the definitions given in Igelmund and Radermacher (1983a,b) and Möhring et al. (1984, 1985), a scheduling policy Π makes decisions at the decision points t = 0 (the start of the project) and the *completion times of activities*. A decision at time t is to start at time t a precedence and resource feasible set of activities S(t), exploiting only information that has become available up to time t. As soon as the activities have been finished, the activity durations are known yielding a realization d of activity durations. The application of policy Π leads to the creation of a schedule $\Pi(d) = (s_1, d)$ s_2, \ldots, s_n) of activity starting times and a resulting schedule makespan $C_{\max}(\Pi(d))$. The common objective considered in the literature is to create a policy that minimizes the expected project duration $E(C_{\max}(\Pi(d)))$ over a class of policies. Fernandez (1995), Fernandez et al. (1996, 1998) and Pet-Edwards et al. (1998) show how to write the corresponding optimization problem in its general form as a multi-stage stochastic programming problem.

4.1.1. Scheduling policies

A complete characterization of policies and corresponding subclasses can be found in Möhring

et al. (1984, 1985). Radermacher (1985) describes early start (ES) policies using the concept of minimal forbidden sets. Minimal forbidden sets are inclusion minimal sets of pair-wise not precedence related activities that cannot be scheduled simultaneously because they share limited resources. 'Inclusion minimal' means that each proper subset of a forbidden set can be executed simultaneously without violating any resource constraints. The number of forbidden sets may grow exponentially in the number of activities. A policy Π is an ESpolicy if for each minimal forbidden set F there exists a pair $(i, j), i, j \in F, i \neq j$, such that for each sample d of activity durations, *j* cannot be started before *i* has finished. ES-policies can easily be implemented by adding the pairs (i, j) to the original set of precedence relations and computing the earliest activity start times as $s_1 = 0$ (starting dummy) and $s_i = \max_{(i,j)\in E} (s_i + d_i), j \in V \setminus \{1\}.$

Igelmund and Radermacher (1983a,b) introduced pre-selective (PRS) policies. A policy Π is pre-selective if for each minimal forbidden set F there exists an activity $j \in F$ (the *pre-selected* or waiting activity), such that for each sample d of activity durations, j is not started before at least one activity $i \in F \setminus \{j\}$ has finished. A *selection* is a sequence of waiting activities for all minimal forbidden sets. Möhring and Stork (2000) have introduced a very useful representation of preselective policies using so-called *waiting conditions*. Waiting conditions can be modelled as AND/OR precedence constraints (Gillies and Liu, 1995; Möhring et al., 2000). A waiting condition is given by a pair $(X, j), X \subset V, j \in V \setminus X$, where activity j cannot be started before at least one activity $i \in X$ has finished. Each restriction imposed by a minimal forbidden set F and its pre-selected activity jcan be represented by the waiting condition $(F \setminus \{j\}, j)$. Obviously, each given precedence constraint $(i, j) \in E$ can be represented by the waiting condition $(\{i\}, j)$. A set W of waiting conditions induces a digraph D that contains a node for each activity as well as for each waiting condition (X, j), directed arcs (i, w) are included for each $i \in X$, with w the node representing (X, j), along with an extra arc (w, j) (Stork, 2000).

We alert the reader to the fact that pre-selective policies do have severe computational limitations.

This inspired Möhring and Stork (2000) to define linear pre-selective policies. *Linear pre-selective policies* (*LIN*) are a subclass of the class of preselective policies. The authors define a selection by a priority ordering L of the activities (respecting the original precedence constraints) in such a way that the preselected waiting activity of the minimal forbidden set F is the activity with the smallest priority, i.e., the last activity in the list L.

A policy is *job based (JBP)* if it is linear preselective (according to some ordering *L* of the activities) and if $s_i \leq s_j$ for each sample *d* and for $i \prec_L j$. For a given sample *d*, the earliest activity start times can be computed by starting each activity in the order imposed by *L* as early as possible, but not earlier than the start time of its predecessors in *L*. Clearly, the job-based policies use an "activity based" point of view and not a "resource based" view. As a result, job-based policies do not require the use of the forbidden sets. This is a very efficiency gaining characteristic since activity based policies can easily be applied to very large projects, for which the number of forbidden sets may be exorbitant.

4.1.2. Branch-and-bound

Stork (2000, 2001) has implemented branchand-bound algorithms to compute optimal *ES*-, *PRS*-, *LIN*- and *JBP*-policies, using two branching schemes, lower bound calculation and various dominance rules. He validates the algorithms on test instances generated using the problem generator *ProGen* (Kolisch and Sprecher, 1996). Preselective policies yielded the smallest expected makespan among all considered classes of policies, which is logical because the set of *PRS*-policies embraces all *LIN*- and *JBP*-policies, and clearly dominates the *ES*-policies.

4.1.3. Heuristic procedures

Research on heuristic procedures for solving the stochastic RCPSP is just emerging (Pet-Edwards, 1996; Golenko-Ginzburg and Gonik, 1997; Tsai and Gemmil, 1996, 1998). As an illustration, we briefly discuss the procedures of Golenko-Ginzburg and Gonik (1997) and the tabu search procedure of Tsai and Gemmil (1998).

Golenko-Ginzburg and Gonik (1997) consider PERT type activity-on-the-arc networks where the duration of an activity is a random variable with given density function (beta, uniform and normal distributions are used) and where a pre-given lower and upper bound on the activity duration is available. The activities require a constant amount of renewable resources during their execution. The renewable resources are available in constant amounts throughout time. The objective is to minimize the expected project duration. At each decision point, when at least one activity is ready to be scheduled, resource contention is resolved by solving a zero-one integer programming problem to maximize the total contribution of the accepted activities to the expected project duration. For each activity, this contribution is computed as the product of its average duration and the probability (determined by simulation) of it lying on the critical path. The procedure is illustrated on a numerical example.

Tsai and Gemmil (1998) report computational results for the well-known 110 Patterson test problems (Patterson, 1984) using a tabu search algorithm. They assume a beta distribution to model activity durations and use an optimistic, most likely and pessimistic time estimate to calculate the parameters of the beta distribution.

Using the expected activity durations, they compute an initial feasible solution using the minimum slack rule. The expected project duration of a feasible solution is computed as follows: (a) a duration for each activity is drawn from the beta distribution with the parameters calculated using the three time estimates, (b) given the feasible sequence and the randomly generated activity durations, the project duration is computed, (c) the calculation of the project duration is repeated 100 times and then the average project duration for the particular feasible sequence is reported as the expected project duration. It should be noted that this approach to estimate the expected project duration violates the so-called non-anticipativity constraint (Fernandez et al., 1996). The approach implicitly assumes that all uncertainty with regard to activity durations is resolved before the start of project execution ('anticipative'), which will only rarely be the case. Rather, information will normally become available only gradually as time progresses, making the use of scheduling policies, as described above, more appropriate.

4.2. Stochastic activity interruptions

Valls et al. (1999) have studied the problem of scheduling resource-constrained project activities that are either deterministic (i.e. have a known duration and cannot be interrupted) or stochastic (i.e. may be interrupted for an uncertain amount of time and resumed later). The initial processing time d_{i1} of an activity *i* that may be interrupted is assumed to be known with certainty, however, the length of the interruption w_i and the remaining processing time after the interruption d_{i2} are uncertain. An example of such a situation may be a project in which some activities are submitted to an approval process before they can be completed. The time to review and approve the work performed during the initial processing determines the length of the interruption, while the outcome of the approval process may determine the length of the final processing. Each activity has a due date δ_i and a tardiness penalty c_i . Each activity requires a constant per period amount of a renewable resource during its execution. The renewable resource types are available in a constant per-period amount. The two parts of an interrupted activity require the same number of units from each resource. The processing time of the second part d_{i2} of an interrupted activity *i* is independent of the length of the interruption w_i . The objective is to schedule the activities subject to the zero-lag finishstart precedence constraints and the resource constraints in order to minimize the expected total weighted tardiness.

The authors have developed a scenario-based approach. The scenarios are generated by specifying three time estimates both for the interruption and for the second part of each stochastic activity. The solution algorithm is a hybrid algorithm based on the scatter search methodology. The authors report on promising computational results obtained on a set of randomly generated test problems. They have extended the approach to the problem of minimizing the weighted tardiness of jobs with stochastic interruptions in a parallel machine environment (Laguna et al., 2000).

4.3. The stochastic discrete timelcost trade-off problem

The literature on the stochastic version of the discrete time/cost trade-off problem is virtually void. Wollmer (1985) discusses a stochastic version of the deterministic linear time/cost trade-off problem for activity-on-the-arc networks in which the duration of an activity can be described as $y_{ij} + \xi_{ij}$, where the decision variable y_{ij} is bounded from below by the activity crash duration l_{ii} and is bounded from above by the normal duration of the activity u_{ij} . ξ_{ij} is a bounded discrete random variable, independent of y_{ii} , with an expected value of 0. Each activity (except dummies) has an associated non-negative cost c_{ij} , which is the cost per unit decrease in y_{ij} within the range of l_{ij} and u_{ij} . The objective then is to determine activity durations y_{ii} and event realization times that minimize the expected project completion time subject to a budget constraint, or achieve a feasible fixed expected project completion time at minimum cost.

Gutjahr et al. (2000) describe a stochastic branch-and-bound procedure for solving a specific version of the stochastic discrete time/cost tradeoff problem where so-called *measures* (e.g. the use of manpower, the assignment of highly-skilled labour or the substitution of equipment) may be used to increase the probability of meeting the project due date and thus avoid penalty costs. The authors assume that the duration of an activity (i, j) in an activity-on-the-arc network is modelled by a beta distributed random variable d_{ij} . The distribution of each d_{ij} can be measured and the random variables d_{ii} are assumed to be independent. It is assumed that the distributions of the d_{ii} can be changed by certain crashing measures $m = 1, \ldots, M$. Typically, measure m reduces the expected time required for one or several activities by a certain amount. As such, the duration of activity (i, j) becomes dependent on the vector $x = (x_1, x_2, \ldots, x_M)$, where $x_m = 1$ if measure m is chosen and $x_m = 0$ otherwise. $d_{ii}(x)$ will denote the duration of activity (i, j) on the condition that a measure combination described by the vector x has

been chosen (in their experiments, the authors assign each measure randomly to an activity). Each measure *m* incurs an additional cost of c_m currency units. For each *x*, the project duration $C_{\max}(\boldsymbol{d}(x))$ can be computed on the basis of the values of $d_{ij}(x)$ using standard critical path calculations. Since $C_{\max}(\boldsymbol{d}(x))$ depends on the stochastic durations $\boldsymbol{d}(x)$, it is also a random variable.

It is assumed that *penalty costs* occur if the project is completed after its pre-specified due date. These costs are described by a loss function Λ , where $\Lambda(t)$ is the loss occurring if the project finishes at time t. The authors assume that Λ is a step function that implies that no penalty occurs if the project is completed on time. The loss $\Lambda(C_{\max}(d(x)))$ is also a random variable. The objective is to minimize the *expected* overall loss, which is equal to the crashing costs and the expected penalty costs. The authors report on promising computational results obtained on 33 random problem instances with 25, 50 and 100 nodes, beta distributed activity durations and 10, 15 or 20 crashing measures.

Scholl (2001) uses a scenario-based approach to formulate mathematical programming models for the stochastic linear time/cost trade-off problem. Using the 110 Patterson networks (Patterson, 1984) in activity-on-the-arc format as a test set, he reaches the conclusion that the so-called compensation models yield the best results for several solution robustness measures. These models assume that, given the scenario-dependent normal activity durations with corresponding crashing rates and associated crashing costs, the planned event realization times may suffer from a scenariobased delay penalized by scenario-dependent event delay costs. The objective is then to minimize the expected indirect and direct costs subject to the precedence and budget constraints.

4.4. Multi-mode trade-off problems in stochastic networks

At the time of writing, the literature on the stochastic multi-mode problem was virtually void. Jørgenson (1999) and Elmaghraby (2000) focus on a dynamic stochastic resource allocation problem in activity-on-the-arc networks where an activity a

requires total work content $W_a(k)$, a random variable, of resource k = 1, ..., K specified as renewable or nonrenewable over the entire planning horizon. An allocation of $x_a(k,t)$ units of resource k to activity a at time t costs $c_k(x_a(k,$ t), $W_a(x_a)$) per unit of time, also a random variable. The resulting activity duration is denoted by the random variable $y_k(x_a) = g_k(W_a(x_a))$. The total activity cost is then the random variable $C_k(x_a) = c_k(x_a(k,t), W_a(x_a)) \cdot g_k(W_a(x_a))$. The project is assumed to have a fixed due date δ_n and a penalty function $p(t_n - \delta_n)$, where t_n is the random variable denoting the time of realization of node *n*. The penalty function is assumed to be linear with proportionality constant p_L ; i.e. $p(t_n - \delta_n) =$ $p_L \cdot \max\{0, t_n - \delta_n\}$. The objective then is to determine the resource allocation vector X_a to all the project activities such that the total expected cost is minimized. In the case of nonrenewable resources, the objective is taken to be the minimization of the project duration.

Elmaghraby (2000) describes two dynamic programming models for solving the problem and illustrates them on a problem example. A new state space is introduced based on the concept of uniformly directed cutsets. For details, we refer the reader to Elmaghraby (2000) and Tereso (2002). Tereso et al. (2003) report on computational results obtained on a set of four projects that range in size from 5 to 18 activities. The solution times varied from a few seconds to five days on a Pentium III processor running at 650 MHz.

Jørgenson (1999) and Elmaghraby (2000) demonstrate that the dynamic resource allocation approach is superior to static optimization, which assumes certainty equivalents given by expected values. Deterministic static time/cost trade-off models underestimate the total expected project costs and neglect the value of flexibility. Updating the plans as new information becomes available by adjusting the amount of resources to be allocated may well lead to superior results. Additional computational experience in this area would be more than welcome.

Golenko-Ginzburg and Gonik (1998) assume a deterministic work content but random activity durations which are the result of performing an activity at a random speed which depends linearly on the renewable resource amounts assigned to it at its random start time. They illustrate an extension of their heuristic procedure for the fixed resource capacity case discussed above (Golenko-Ginzburg and Gonik, 1997) on a numerical problem example. Laslo (2003) describes four stochastic models for computing time/cost tradeoffs of a single activity using activity duration fractiles. The author illustrates on a small example how to determine the performance speed of a single activity, i.e. how to allocate the required budget in order to obtain the desired activity duration under both cost chance constraints and time chance constraints.

5. Fuzzy project scheduling

The advocates of the fuzzy activity duration approach argue that probability distributions for the activity durations are unknown due to the lack of historical data. As activity durations have to be estimated by human experts, often in a non-repetitive or even unique setting, project management is often confronted with judgmental statements that are vague and imprecise. In those situations, which involve imprecision rather than uncertainty, the fuzzy set scheduling literature recommends the use of fuzzy numbers for modelling activity durations, rather than stochastic variables. Instead of probability distributions, these quantities make use of *membership functions*, based on *possibility theory*.

A fuzzy set is a function that measures the degree of membership to a set. Set A in a base set X can be described by a membership function $\mu_A: X \to \{0, 1\}$ with $\mu_A(x) = 1$ if $x \in A$ and $\mu_A(x) = 0$ if $x \notin A$. If it is uncertain whether or not element x belongs to set A, the above model can be extended such that the membership function maps into the interval [0, 1]. A high value of this membership function implies a high possibility, while a low value implies a poor possibility. This leads to the definition of a fuzzy set \widetilde{A} in X as a set of ordered pairs $\widetilde{A} = \{(x, \mu_{\widetilde{A}}(x)) | x \in X\}$, where $\mu_{\widetilde{A}}(x)$, $0 \leq \mu_{\widetilde{A}}(x) \leq 1$, is called the membership function or grade of membership of x in \widetilde{A} . In the classical case where $\mu_{\widetilde{A}}(x) = 0$ or 1, \widetilde{A} is said to be a crisp set.

A fuzzy number $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\}$, where $\mu_{\tilde{A}}$ is the membership function of A, is a special kind of a fuzzy set defined as a fuzzy subset of the real line \Re that is convex, which means that $\forall a, b \in \Re, \forall c \in [a, b], \mu_{\tilde{4}}(c) \ge \min(\mu_{\tilde{4}}(a), \mu_{\tilde{4}}(b)).$ It is also required that $\exists a \in \Re : \mu_{\tilde{a}}(a) = 1$. The advocates of fuzzy scheduling admit that the precise form of a fuzzy number is difficult to describe by an expert (Hapke et al., 1999). A practical way of getting suitable membership functions of fuzzy data has been proposed by Rommelfanger (1990). He recommends that the expert express his/her optimistic and pessimistic information about parameter uncertainty on some prominent membership levels by specifying intervals on \Re : the smallest interval $[m, \bar{m}]$ for which $\mu(x) = 1$, meaning that x certainly belongs to the set of possible values; a larger interval $[\underline{m}^{\lambda}, \overline{m}^{\lambda}]$, containing $[\underline{m}, \overline{m}]$, for which it holds that values x have a good chance $\geq \lambda$ of belonging to the set of possible values; and a third interval $[\underline{m}^{\varepsilon}, \overline{m}^{\varepsilon}]$, containing the second, for which all values x have $\mu(x) < \varepsilon$. Values x with $\mu(x) < \varepsilon$ have a very small chance of belonging to the set of possible values; i.e. the expert is willing to neglect the corresponding values of x. Using a six-point representation, a fuzzy number M is then represented by the list of symbols $M = (m^{\epsilon}, m^{\epsilon})$ $m^{\lambda}, m, \bar{m}, \bar{m}^{\lambda}, \bar{m}^{\varepsilon}$) as shown in Fig. 1.

The output of a fuzzy scheduling pass will normally be a *fuzzy schedule*, which indicates fuzzy starting and ending times for the activities. Such fuzzy time instances may be interpreted as start or completion to a certain extent only. As can be conceived from, amongst others, Dorn et al.



Fig. 1. Fuzzy number \widetilde{M} in six-point representation (Hapke et al., 1999).

(1995), a fuzzy schedule assists in the explicit representation of certain degrees of freedom in the predictive schedule to represent the discretion management has to start certain jobs a little earlier or later when duly propagating certain hard and soft constraints that may be imposed. In this sense, a fuzzy schedule comprises multiple crisp schedules.

The recent volume edited by Slowinski and Hapke (2000) gathers important recent work in fuzzy scheduling. At the time of writing, the literature on fuzzy resource-constrained project scheduling was in its burn-in phase (Hapke et al., 1994, 1999; Hapke and Slowinski, 1996, 2000; Özdamar and Alanya, 2000; Wang, 1999, 2002, 2004).

The study of a fuzzy model of resource-constrained project scheduling has been initiated in Hapke et al. (1994) and Hapke and Slowinski (1996). They have extended the priority rule based serial and parallel scheduling schemes to deal with fuzzy parameters.

Hapke and Slowinski (2000) discuss the application of simulated annealing for solving the multi-objective fuzzy resource-constrained project scheduling problem. The procedure is an adaptation of the Pareto simulated annealing procedure developed by Czyzak and Jaskievicz (1996) for solving crisp multi-objective combinatorial problems. The procedure has been incorporated in an integrated software package. For details we refer to Hapke and Slowinski (2000).

Özdamar and Alanya (2000) study software development projects and offer a nonlinear mixedbinary mathematical problem formulation and accompanying solution heuristics. Their model incorporates uncertainties related to activity durations and network topology. Activities may be performed in one of different modes with a corresponding fuzzy duration. The objective function is to minimize the project duration. Özdamar and Alanya (2000) illustrate the use of four priority based heuristics: the standard minimum slack rule, the latest finish time rule, the maximum number of immediate successor rule and a minimum risk rule on a case study.

Wang (1999) has developed a fuzzy set approach to schedule product development projects

having imprecise temporal information. The project has a fuzzy ready time and fuzzy deadline and the activities are assumed to have a fuzzy duration. all described by trapezoidal fuzzy numbers. The objective is to determine a start time for each activity such that the fuzzy ready time, deadline, precedence and resource constraints are satisfied. A beam search procedure, based on the generation of groups of activities the delay of which resolves the resource conflicts (i.e. delaying alternatives (Demeulemeester and Herroelen, 2002)), that selects only the most promising nodes at each level of the search tree (the so-called beam-width) for further expansion is developed to produce a set of fuzzy start times for each activity. Then, the crisp start time of each activity is determined based on possibility theory, to maximize the satisfaction degrees of all fuzzy constraints. Wang (2002) has presented a fuzzy beam search approach for solving the problem under the objective of minimizing the schedule risk. Wang (2004) describes a genetic algorithm for solving the problem under the objective of maximizing the worst case schedule performance.

6. Proactive (robust) project scheduling

Numerous techniques for proactive (robust) scheduling have recently been published. The majority of publications are in the machine scheduling literature (Davenport and Beck, 2002).

6.1. Redundancy-based techniques

Fault tolerance has been studied in real-time pre-emptive single machine scheduling environments to ensure that faults in the system do not lead to overall system failure. Fault tolerance can be achieved through resource redundancy (multiple identical sets of resources kept in standby (Ghosh, 1996)) or time redundancy (scheduling of back-up tasks that simply reserve time for re-execution in the event of a fault (Ghosh et al., 1995)). Pure resource redundancy is rather unrealistic in a project environment: doubling the various resources would be cost prohibitive. Time redundancy may be relevant, but unfortunately a (multi-) project environment is far off from the pre-emptive polynomially solvable single machine settings studied in a real-time environment.

Temporal protection (Gao, 1995) extends the duration of activities based on the uncertainty statistics of the resources that are used for their execution. Resources that have a non-zero probability of breakdown are called breakable resources. The durations of activities requiring breakable resources are extended to provide extra time with which to cope with a breakdown. The "protected" duration of each activity equals its original duration augmented with the duration of breakdowns that are expected to occur during activity execution, based on breakdown statistics for the performing resources (mean time to failure, mean time to repair, which makes this approach less applicable in a project setting, where most resources are human beings). The baseline schedule is then obtained by solving the scheduling problem with protected durations.

Davenport et al. (2001) propose improvements of this temporal protection technique with their *time window slack* and *focused time window slack* approaches in which they do not include slack into activity durations, but explicitly compute available slack time per activity in solution schedules. In this way, they are able to utilize the same slack time for protecting more than activity, and concentrate slack in areas of the schedule that are most important or most vulnerable.

Mehta and Uzsoy (1998, 1999) insert additional idle time into the predictive schedule to absorb the impact of machine breakdowns. Mehta and Uzsoy (1999) consider the problem of minimizing total tardiness on a single machine with dynamic job arrival and random breakdowns. They compute an initial sequence by a heuristic and then insert additional idle times into the schedule. Mehta and Uzsoy (1998) study the problem of minimizing the maximum lateness in a job shop subject to machine breakdowns. Assuming the distributions of the time between breakdowns and the time to repair for the machines to be available, they generate a baseline schedule using the shifting bottleneck heuristic (Adams et al., 1988). They invoke earliness and lateness penalties whenever the last operation of a job ends sooner or later than

planned. They use two heuristics to insert idle time to minimize expected job completion time deviations. In the "linear programming based heuristic" (LPH), the idea is to develop a schedule with *expected* durations for all the activities, and minimize the summed deviation of the pre-schedule from this 'blown up schedule'.

Tavares et al. (1998) study the risk of a project as a function of the uncertainty of the duration and the cost of each activity and the adopted schedule. The adoption of an early (late) start schedule reduces (increases) the risk of an overall delay but increases (decreases) the project's discounted cost, which calls for the difficult determination of an optimal compromise. The authors suggest that the start time of each activity *i* be set equal to $s_i(\alpha) = es_i + \alpha(ls_i - es_i)$, where es_i and ls_i denote the earliest, respectively, latest start time of activity *i* given project deadline δ_n , and α , $0 \leq \alpha \leq 1$, denotes the so-called float factor. The late start (early start) schedule is obtained with $\alpha = 1$ ($\alpha = 0$). The authors prove that the use of $s_i(\alpha)$ yields a feasible schedule.

6.2. Robust machine scheduling techniques

Leon et al. (1994) describe a genetic algorithm for generating robust schedules for job shops. They define the schedule robustness of a job shop schedule S as

 $R(S) = r \times E[\boldsymbol{M}(S)] + (1-r)E[\boldsymbol{\delta}(S)],$

where M(S) is a random variable denoting the actual makespan of S in the presence of disruptions, r is a real-valued weight in the interval [0, 1], and $\delta(S) = M(S) - M_o(S)$ represents the schedule delay, defined as a random variable expressing the difference between executed and pre-schedule makespan. Since $M_o(S)$ is deterministic, the expected values of M(S) and $\delta(S)$ equate as $E[\mathbf{M}(S)] = E[\boldsymbol{\delta}(S)] + M_o(S)$. The authors assume a right-shift reactive policy that restarts the disrupted operations immediately after the disruption period. They demonstrate that schedule robustness R(S) can be computed directly for a schedule with a single disruption. When there is more than one disruption, the authors have tested three surrogate robustness measures.

Sevaux and Sörensen (2002a,b) study the single machine scheduling problem with ready times under the objective of minimizing the weighted number of late jobs. They rely on a genetic algorithm for generating quality robust schedules, i.e., schedules whose quality does not change when the input data (i.e., the ready times) change.

The approach of Sevaux and Sörensen is closely related to the one proposed by Jensen (2001), who provides extensive theoretical and experimental results for a job shop environment. Whereas the foregoing studies use the expected value objective, Jensen also studies minimization of worst case performance and of worst case deviation performance (absolute and relative). These correspond with minimax and minimax regret objectives in decision analysis. The minimax objective minimizes the consequences of the worst case scenario and tends to generate very conservative schedules. Minimax regret techniques associate schedule robustness with the schedule with the best worstcase regret performance over all potential realizations of job processing times, with 'regret' for a particular scenario measured either as absolute difference or as percentage difference between the resulting cost and the cost that would have resulted from perfect information for that scenario. As explained by Kouvelis and Yu (1997), minimax regret objectives will yield less conservative schedules, since they take into account the magnitude of missed opportunities of a decision by benchmarking its performance with the performance of the optimal 'ex post' solution.

Daniels and Kouvelis (1995) study the single machine problem under the total flow time objective. For a given schedule and a set of processing times for the single machine problem, the regret is measured as the absolute difference between the total flow time of the schedule for that scenario and the flow time obtained using the (optimal) shortest processing time rule. Kouvelis et al. (2000) focus on the two-machine flow shop environment, in which the deviation is computed between the makespan of the schedule for a scenario and the makespan of the (optimal) Johnson schedule for that scenario. The authors develop branch-andbound algorithms and heuristics for determining robust schedules. 300

As a continuation of the research on minimax regret objectives, Daniels and Carrillo (1997) investigate a combination of average system performance and performance variability in determining the optimal schedule. Focusing on a single machine environment and a set of activity processing time scenarios, their scheduling objective is to determine a β -robust schedule, i.e., the schedule with the maximum likelihood of achieving flow time performance no greater than a particular target level. Having established NP-hardness of the problem, the authors offer a branch-andbound procedure and a heuristic. They also extend the analysis to those situations where a single resource, available in limited supply, can be applied to individual jobs to linearly decrease the associated processing time variance. Computational experience indicated that β -robust schedules provide effective hedges against processing time uncertainty while maintaining near-optimal performance with respect to expected flow time.

6.3. Robust project scheduling

6.3.1. Abstraction of resource usage

Herroelen and Leus (2004) develop mathematical programming models for the generation of stable baseline schedules in a project environment. The authors make abstraction of resource usage, assuming that a proper allocation of resources has been performed. They use the concept of pair-wise *float*, $F_{ii}(S) = s_i(S) - f_i(S)$, defined as the difference between the start time of activity *j* and the finish time of activity *i* in a schedule S. The pair-wise float is only defined for activities $(i, j) \in TA$, where TA denotes the transitive closure of A, meaning that $(i, j) \in TA$ if and only if a path from *i* to *j* exists in the activity-on-the-node project network G = (N, A). The authors assign a project deadline δ_n and a probability of disruption p_i to every activity *i* (i = 1, 2, ..., n), with $\sum_{i=1}^{n} p_i = 1$. The dummy end node has disruption probability $p_n = 0$, while p_1 denotes the probability that the dummy start node, i.e. the entire project, starts later than initially anticipated. They use a random variable L_i to denote the disturbance length of activity *i* if it is disturbed, and a non-negative cost c_i per unit time overrun on the start time of activity *i*.

The authors propose to use as stability measure the expected weighted deviation in start times in the realized schedule from those in the pre-schedule. In other words, the expression they wish to minimize is $\sum_{j=1}^{n} c_j(E[s_j] - s_j(S))$, with E the expectation operator, $s_i(S)$ the start time of activity *j* in the pre-schedule *S*, and s_i a random variable representing the actually achieved start time of activity *j* (after project execution). If for all arcs $(i, j) \in TA$, MSPF_{ij} denotes the minimal sum of pairwise floats of all edges on any path leading from *i* to *j*, then $E[s_i]$ can be computed as $s_i(S)$ + $\sum_{i \in \pi^{T}(j)} p_{i}E(\max\{0; L_{i} - \text{MSPF}_{ij}\} | i \text{ disturbed}),$ where $\pi^{T}(j)$ is the set of all immediate and tran $p_i E(\max\{0; L_i - \text{MSPF}_{ij}\}|i \text{ disturbed}),$ sitive predecessors of *j*. Hence, the objective can be rewritten as min $\sum_{(i,j)\in\mathrm{TA}} c_j p_i E(\max\{0; L_i MSPF_{ij}$ |*i* disturbed). Assuming a single disruption and all L_i to be discrete with probability mass function $g_i(\cdot)$ which associates nonzero probability with positive values l_{ik} that correspond with the elements k in D_i , the set of disturbance scenarios for activity *i*, the authors solve the following linear programming model:

$$\min \quad \sum_{(i,j)\in TA} \sum_{k\in D_i} c_j p_i g_i(l_{ik}) \Delta_{ijk} \tag{1}$$

subject to

$$s_i + d_i + F_{ij} = s_j \quad \forall (i,j) \in A,$$

$$\tag{2}$$

$$s_n \leqslant \delta_n,$$
 (3)

$$l_{ik} - \mathbf{MSPF}_{ij} \leq \Delta_{ijk} \quad \forall (i,j) \in TA, \forall k \in D_i,$$
(4)

$$s_i + d_i + \lambda_{ij} + \mathbf{MSPF}_{ij} = s_j \forall (i, j) \in TA,$$
(5)

all
$$\Delta_{ijk}, s_i, F_{ij}, \text{MSPF}_{ij} \ge 0,$$
 (6)

where Δ_{ijk} is the delay in the start time of activity *j* due to a disturbance according to scenario *k* of activity *i*, and λ_{ij} is the length of the path from *i* to *j* (not including *i* and *j*) for which MSPF_{*ij*} is achieved. This linear program can be rewritten as the dual of a minimum cost network flow problem. The authors have extended the model to cope with multiple disturbances. They report on very promising computational results obtained on a set of randomly generated test instances.

Herroelen and Leus (2004) have adapted the float factor model of Tavares et al. (1998) and the model of Mehta and Uzsoy (1998, 1999) discussed in Section 6.1 to a project environment. Results obtained on a dataset consisting of 300 instances generated using the problem generator *RanGen* (Demeulemeester et al., 2003) demonstrate that both models are clearly outperformed by the model given earlier in Eqs. (1)–(6).

6.3.2. Restricted resources and robust resource allocation

If the unrestricted resource availability assumption is dropped from the analysis, Leus and Herroelen (2004) use a so-called *resource flow network* to represent the flow of resources across the activities of the project network (the concept of a resource flow network has been presented by Naegler and Schoenherr (1989), Bowers (1995) and Artigues and Roubellat (2000)). An interesting question is whether we can find a feasible resource allocation corresponding with a given feasible input schedule *S* such that

$$\sum_{(i,j)\in \mathrm{TA}\cup F}\sum_{k\in D_i}c_jp_ig_i(l_{ik})\Delta_{ijk}\leqslant U,$$

where F is the set of extra resource links. Leus (2003) has shown that this decision problem is NP-complete even when all activities have a single disruption scenario, by establishing that the parallel machine problem with weighted completion time objective (Bruno et al., 1974) can be reduced to it.

Leus and Herroelen (2004) have studied the problem of generating a robust resource allocation under the assumption that a feasible baseline schedule exists and that some advance knowledge about the probability distribution of the activity durations is available. The authors explore the fact that checking the feasibility of a resource allocation can easily be done using maximal flow computations in the resource flow network. As such, the search for an optimal allocation is reduced to the search for an associated resource flow network with desirable robustness characteristics. The authors propose a branch-and-bound algorithm that solves the robust resource allocation problem in exact and approximate formulations. The procedure heavily relies on constraint propagation during its search. The authors report on promising results obtained on a set of problem instances generated using the problem generator *RanGen* (Demeulemeester et al., 2003).

6.4. Multiple schedules (contingent scheduling)

The contingent scheduling approach is based on the generation of multiple baseline schedules (or baseline schedule fragments) before and/or during project execution that optimally respond to anticipated disruptive events, or are equivalent in performance. Responding to anticipated or even unexpected events during schedule execution is then simply done by switching to the schedule (fragment) that corresponds to the events that have occurred. This approach focuses on flexibility, rather than robustness, and is especially valuable for time-critical reactive scheduling.

Billaut and Roubellat (1996a) suggest to generate for every resource a so-called *group sequence*, i.e. a totally or partially ordered set of groups of operations, and to consider all the schedules obtained by an arbitrary choice of the ordering of the operations inside each group. Mauguière et al. (2002) and Aloulou et al. (2002) explore this sequence flexibility idea in the context of single machine scheduling.

The gist of the approach can be sketched using the 4 job–2 machine example borrowed from Billaut and Roubellat (1996a). The four jobs are subject to ready times $\rho_1 = 1$, $\rho_2 = \rho_3 = \rho_4 = 0$

Table 2Data for the 4 job-2 resource problem

	(job/oper	ation)							
	(1,1)	(1,2)	(2,1)	(2,2)	(3,1)	(3,2)	(4,1)	(4,2)	
Machine	1	2	1	2	2	1	2	1	
Processing time	1	1	1	1	1	1	1	1	

and due dates $\delta_2 = 4$, $\delta_1 = \delta_3 = \delta_4 = 5$. Additional data are shown in Table 2. The notation (i, j) refers to operation *j* of job *i*. Consider the following group sequence:

Resource 1: group 1: $\{(1,1),(2,1)\}$ group 2: $\{(3,2),(4,2)\}$ Resource 2: group 1: $\{(3,1),(4,1)\}$ group 2: $\{(1,2),(2,2)\}$

Table 3 enumerates the 16 schedules that can be generated from this group sequence by choosing an arbitrary processing order for the operations inside each group ($a \prec b$ means *a* strictly precedes *b*). All 16 schedules are feasible. In this way the decision maker is not just provided with one feasible schedule but with several ones. The hope is that during the real-time execution of the schedule, it becomes possible to switch from one solution to the other in the presence of a disruption without any loss of performance.

Billaut and Roubellat (1996a,b) extend the group sequence concept to the multiple renewable resource case by adding the condition that the operations in a group should use the same amount of a resource type, and the operations in a group are assigned to the same subset of units of the resource type. Briand et al. (2002) extend the methodology used by Billaut and Roubellat

 Table 3

 Set of schedules (Billaut and Roubellat, 1996a)

Resource 1	Resource 2
$(1,1) \prec (2,1) \prec (3,2) \prec (4,2)$	$(3,1) \prec (4,1) \prec (1,2) \prec (2,2)$
$(2,1) \prec (1,1) \prec (3,2) \prec (4,2)$	$(3,1) \prec (4,1) \prec (1,2) \prec (2,2)$
$(1,1) \prec (2,1) \prec (4,2) \prec (3,2)$	$(3,1) \prec (4,1) \prec (1,2) \prec (2,2)$
$(2,1) \prec (1,1) \prec (4,2) \prec (3,2)$	$(3,1) \prec (4,1) \prec (1,2) \prec (2,2)$
$(1,1) \prec (2,1) \prec (3,2) \prec (4,2)$	$(4,1) \prec (3,1) \prec (1,2) \prec (2,2)$
$(2,1) \prec (1,1) \prec (3,2) \prec (4,2)$	$(4,1) \prec (3,1) \prec (1,2) \prec (2,2)$
$(1,1) \prec (2,1) \prec (4,2) \prec (3,2)$	$(4,1) \prec (3,1) \prec (1,2) \prec (2,2)$
$(2,1) \prec (1,1) \prec (4,2) \prec (3,2)$	$(4,1) \prec (3,1) \prec (1,2) \prec (2,2)$
$(1,1) \prec (2,1) \prec (3,2) \prec (4,2)$	$(3,1) \prec (4,1) \prec (2,2) \prec (1,2)$
$(2,1) \prec (1,1) \prec (3,2) \prec (4,2)$	$(3,1) \prec (4,1) \prec (2,2) \prec (1,2)$
$(1,1) \prec (2,1) \prec (4,2) \prec (3,2)$	$(3,1) \prec (4,1) \prec (2,2) \prec (1,2)$
$(2,1) \prec (1,1) \prec (4,2) \prec (3,2)$	$(3,1) \prec (4,1) \prec (2,2) \prec (1,2)$
$(1,1) \prec (2,1) \prec (3,2) \prec (4,2)$	$(4,1) \prec (3,1) \prec (2,2) \prec (1,2)$
$(2,1) \prec (1,1) \prec (3,2) \prec (4,2)$	$(4,1) \prec (3,1) \prec (2,2) \prec (1,2)$
$(1,1) \prec (2,1) \prec (4,2) \prec (3,2)$	$(4,1) \prec (3,1) \prec (2,2) \prec (1,2)$
$(2,1) \prec (1,1) \prec (4,2) \prec (3,2)$	$(4,1) \prec (3,1) \prec (2,2) \prec (1,2)$

(1996b) to the case of multi-mode scheduling with minimal and maximal time-lags.

Artigues et al. (1999) study multi-mode project scheduling problems where the projects have a release date and a due date. They propose a generation procedure for finding group sequences based on a new priority rule. They also propose and test an efficient local search procedure to improve the feasibility of a group sequence. The procedures are integrated in a commercial realtime scheduling package (ORDO[®]).

7. Sensitivity analysis

A number of recent research efforts focus on the sensitivity analysis of machine scheduling problems (Hall and Posner, 2000a,b). Sensitivity analysis addresses "What if...?" types of questions that arise from parameter changes. The authors study polynomially solvable and intractable machine scheduling problems and try to provide answers to a number of fundamental questions such as (a) what are the limits to the change of a parameter such that the solution remains optimal? (b) Given a specific change of a parameter, what is the new optimal cost? (c) Given a specific change of a parameter, what is a new optimal solution? (d) When does a baseline schedule remain optimal? (e) When does the objective function value remain optimal? (f) What types of sensitivity analysis are useful to evaluate the robustness of optimal solutions? (g) What types of sensitivity analysis can be performed without using the full details of the solution? etc. An interesting area of future research is to pose and answer similar questions in a project scheduling setting. An additional interesting and as yet unexplored research topic is to determine what parameter changes are allowed to guarantee full rescheduling optimality by means of a 'simple' repair action (e.g. right shift).

Penz et al. (2001) determine the *sensitivity* guarantee of off-line scheduling algorithms for single and parallel machine scheduling problems where the actual duration of a task *i* is equal to $(1 + \varepsilon_i)d_i$, with $\varepsilon_i \in]-1, +\infty[$ representing the percentage of confidence we have on the corresponding estimated duration. Values $1 + \varepsilon_i$ are the

303

components of the perturbation vector $\vec{\epsilon}$. The sensitivity guarantee of an off-line algorithm ALG is a function $s_{ALG}(\varepsilon)$ such that for any off-line instance \Im and any ε -perturbation $\vec{\varepsilon}$, $s_{ALG}(\varepsilon)$ is the smallest real value verifying $\rho_{ALG}^{\vec{\epsilon}}(\mathfrak{I}) \leqslant s_{ALG}(\epsilon) \times$ $\rho_{ALG}(\mathfrak{I})$, with $\|\overline{\varepsilon}\| \leq \varepsilon$. In this expression, $\rho_{ALG}(\mathfrak{I}) = f_{ALG}(\mathfrak{I})/f_{OPT}(\mathfrak{I})$ denotes the *theoreti*cal or off-line performance ratio of algorithm ALG, for which $f_{ALG}(\mathfrak{I})$ denotes the objective value achieved by algorithm ALG on \Im and $f_{\text{OPT}}(\mathfrak{I})$ denotes the optimal objective value for the instance. $\rho_{ALG}^{\vec{\epsilon}}(\mathfrak{I}) = f_{ALG}^{\vec{\epsilon}}(\mathfrak{I})/f_{OPT}^{\vec{\epsilon}}(\mathfrak{I})$ denotes the effective performance ratio, i.e. obtained after execution. The numerator and denominator in the right-hand side of the expression represent the objective value of the ALG schedule for \mathfrak{I} , applied to \mathfrak{I} perturbed by $\vec{\epsilon}$, and the optimal value *ex post*, with perfect knowledge, respectively.

8. Summary and suggestions for further research

The majority of research efforts in project scheduling assume complete information about the scheduling problem to be solved and assume a static deterministic environment. Basically the research efforts aim at the generation of feasible baseline schedules that 'satisfice' or optimize single or multiple objective functions. The literature on project scheduling under risk and uncertainty is rather sparse. In this paper we offer a review of the major approaches to deal with scheduling risk and uncertainty, many of which have been mainly or solely studied in a machine scheduling environment.

The methodologies for *stochastic project scheduling* basically view the project scheduling problem as a multi-stage decision process. Scheduling policies are used that define which activities are to be started at random decision points through time, based on the observed past and the a priori knowledge about the processing time distributions. As such they share the disadvantage that they do not explicitly generate a pre-schedule that can be used as the baseline plan for making advance commitments to both subcontractors and customers. The dynamic programming approaches developed to tackle the stochastic multi-mode

problem determine the resource allocation vectors for the project activities in order to minimize total expected cost and rely on the assumption that the uncertainty resides in the work content of the activities and not in their duration.

The *fuzzy project scheduling* approach rejects the use of probability distributions for the activity durations but relies on membership functions that may be as difficult to generate. As such uncertainty is captured by the notion of "belonging" rather than in terms of "frequency" of occurrence. The literature is still in its burn-in phase.

Research in proactive (robust) scheduling has widely prospered in the field of machine scheduling. Redundancy-based techniques have already found their way to the field of project scheduling. The buffer insertion approach, the fundamental ingredient of Goldratt's critical chain methodology (Goldratt, 1997), is gaining increasing popularity among project management practitioners. While this methodology has acted as an important eye-opener, its pitfalls, mainly due to severe oversimplifications, have been revealed recently. The generation of robust multi-resource baseline schedules in combination with efficient and effective reactive schedule repair mechanisms constitutes a viable area of future research. Whereas numerous reactive scheduling mechanisms have been developed and tested in real-time machine scheduling environments, the field is in need for further research aimed at their implementation and validation in a project scheduling environment.

Research on *sensitivity analysis* has just emerged in the area of machine scheduling. Efforts to seek answers to the various types of "what if …" questions in a project setting still need to be initiated, and would offer useful information to project management.

References

- Adams, J., Balas, E., Zawack, D., 1988. The shifting bottleneck procedure for job shop scheduling. Management Science 34, 391–401.
- Akturk, M.S., Gorgulu, E., 1999. Match-up scheduling under a machine breakdown. European Journal of Operational Research 112, 81–97.

- Alagöz, O., Azizoglu, M., 2003. Rescheduling of identical parallel machines under machine eligibility constraints. European Journal of Operational Research 149, 523–532.
- Aloulou, M.A., Portmann, M.-C., Vignier, A., 2002. Predictivereactive scheduling for the single machine problem. Paper presented at the 8th Workshop on Project Management and Scheduling, Valencia, 3–5 April.
- Artigues, C., Roubellat, F., 2000. A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes. European Journal of Operational Research 127, 297–316.
- Artigues, C., Roubellat, F., Billaut, J.-C., 1999. Characterization of a set of schedules in a resource-constrained multiproject scheduling problem with multiple modes. International Journal of Industrial Engineering—Theory, Applications and Practice 6 (2), 112–122.
- Aytug, H., Lawley, M.A., McKay, K., Mohan, S., Uzsoy, R., in press. Executing production schedules in the face of uncertainties: A review and some future directions. European Journal of Operational Research.
- Bean, J.C., Birge, J.R., Mittenthal, J., Noon, C.E., 1991. Match-up scheduling with multiple resources, release dates and disruptions. Operations Research 39 (3), 470–483.
- Billaut, J.C., Roubellat, F., 1996a. A new method for workshop real time scheduling. International Journal of Production Research 34 (6), 1555–1579.
- Billaut, J.C., Roubellat, F., 1996b. Characterization of a set of schedules in a multiple resource context. Journal of Decision Systems 5 (1–2), 95–109.
- Bowers, J.A., 1995. Criticality in resource constrained networks. Journal of the Operational Research Society 46, 80– 91.
- Briand, C., Despontin, E., Roubellat, F., 2002. Scheduling with time lags and preferences: A heuristic. Paper presented at the 8th Workshop on Project Management and Scheduling, Valencia, 3–5 April.
- Bruno, J., Coffman E.G., Jr., Sethi, R., 1974. Scheduling independent tasks to reduce mean finishing time. Communications of the ACM 17 (7), 382–387.
- Calhoun, K.M., Deckro, R.F., Moore, J.T., Chrissis, J.W., Van Hove, J.C., 2002. Planning and re-planning in project and production planning. Omega 30, 155–170.
- Czyzak, P., Jaskievicz, A., 1996. Metaheuristic technique for solving multiobjective investment planning problem. Control and Cybernetics 25, 177–187.
- Daniels, R.L., Carrillo, J.E., 1997. β-robust scheduling for single-machine systems with uncertain processing times. IIE Transactions 29, 977–985.
- Daniels, R.L., Kouvelis, P., 1995. Robust scheduling to hedge against processing time uncertainty in single-stage production. Management Science 41 (2), 363–376.
- Davenport, A.J., Beck, J.C., 2002. A survey of techniques for scheduling with uncertainty, Unpublished manuscript. Available from http://www.eil.utoronto.ca/profiles/chris/ gz/uncertainty-survey.ps>.
- Davenport, A.J., Gefflot, C., Beck, J.C., 2001. Slack-based techniques for robust schedules. Paper presented at the

Constraints and Uncertainty Workshop, Seventh International Conference on Principles and Practice of Constraint Programming, November 26–December 1, Paphos, Cyprus.

- Demeulemeester, E., Vanhoucke, M., Herroelen, W., 2003. RanGen: A random generator for activity-on-the-node networks. Journal of Scheduling 6, 17–38.
- Demeulemeester, E.L., Herroelen, W.S., 2002. Project Scheduling—A Research Handbook. Kluwer Academic Publishers, Boston.
- Dorn, J., Kerr, R., Thalhammer, G., 1995. Reactive scheduling: improving robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning. International Journal of Human–Computer Studies 42, 687–704.
- Elmaghraby, S.E.E., 2000. Optimal Resource Allocation and Budget Estimation in Multimodal Activity Networks, Research Paper. North Carolina State University at Raleigh, North Carolina, USA.
- El Sakkout, H., Wallace, M., 2000. Probe backtrack search for minimal perturbation in dynamic scheduling. Constraints 5 (4), 359–388.
- Fernandez, A.A., 1995. The optimal solution to the resourceconstrained project scheduling problem with stochastic task durations. Unpublished Doctoral Dissertation. University of Central Florida.
- Fernandez, A.A., Armacost, R.L., Pet-Edwards, J., 1996. The role of the non-anticipativity constraint in commercial software for stochastic project scheduling. Computers and Industrial Engineering 31, 233–236.
- Fernandez, A.A., Armacost, R.L., Pet-Edwards, J., 1998. Understanding simulation solutions to resource-constrained project scheduling problems with stochastic task durations. Engineering Management Journal 10, 5–13.
- Gao, H., 1995. Building robust schedules using temporal protection—an empirical study of constraint based scheduling under machine failure uncertainty. Master's Thesis. Department of Industrial Engineering, University of Toronto.
- Ghosh, S., 1996. Guaranteeing fault tolerance through scheduling in real-time systems. Ph.D. thesis. University of Pittsburgh.
- Ghosh, S., Melhem, R., Mossé, D., 1995. Enhancing Real-Time Schedules to Tolerate Transient Faults, Real-Time Systems Symposium.
- Gillies, D.W., Liu, J.W.S., 1995. Scheduling tasks with and/or precedence constraints. SIAM Journal on Computing 24, 797–810.
- Goldratt, E., 1997. Critical Chain. The North River Press.
- Golenko-Ginzburg, D., Gonik, A., 1997. Stochastic network project scheduling with non-consumable limited resources. International Journal of Production Economics 48, 29– 37.
- Golenko-Ginzburg, D., Gonik, A., 1998. A heuristic for network project scheduling with random activity durations depending on the resource allocation. International Journal on Production Economics 55, 149–162.

- Gutjahr, W.J., Strauss, C., Wagner, E., 2000. A stochastic branch-and-bound approach to activity crashing in project management. INFORMS Journal on Computing 12 (2), 125–135.
- Hall, N., Posner, M., 2000a. Sensitivity analysis for intractable scheduling problems, Research paper. The Ohio State University.
- Hall, N., Posner, M., 2000b. Sensitivity analysis for efficiently solvable scheduling problems, Research paper. The Ohio State University.
- Hapke, M., Jaskievicz, A., Slowinski, R., 1994. Fuzzy project scheduling system for software development. Fuzzy Sets and Systems 21, 101–117.
- Hapke, M., Jaskievicz, A., Slowinski, R., 1999. Fuzzy multimode resource-constrained project scheduling with multiple objectives. In: Weglarz, J. (Ed.), Project Scheduling— Recent Models, Algorithms and Applications. Kluwer Academic Publishers, pp. 355–382 (Chapter 16).
- Hapke, M., Slowinski, R., 1996. Fuzzy priority heuristics for project scheduling. Fuzzy Sets and Systems 83, 291– 299.
- Hapke, M., Slowinski, R., 2000. Fuzzy set approach to multiobjective and multi-mode project scheduling under uncertainty. In: Slowinski, R., Hapke, M. (Eds.), Scheduling Under Fuzziness. Physica-Verlag, Heidelberg, pp. 197–221 (Chapter 9).
- Herroelen, W., Leus, R., 2004. The construction of stable project baseline schedules. European Journal of Operational Research 156, 550–565.
- Igelmund, G., Radermacher, F.J., 1983a. Preselective strategies for the optimization of stochastic project networks under resource constraints. Networks 13, 1–28.
- Igelmund, G., Radermacher, F.J., 1983b. Algorithmic approaches to preselective strategies for stochastic scheduling problems. Networks 13, 29–48.
- Jensen, M.T., 2001. Robust and flexible scheduling with evolutionary computation. Ph.D. thesis. Department of Computer Science, University of Aarhus, Denmark.
- Jørgenson, T., 1999. Project scheduling—a stochastic dynamic decision problem. Doctoral Dissertation. Norwegian University of Science and Technology, Trondheim, Norway.
- Kolisch, R., Sprecher, A., 1996. PSPLIB—A project scheduling library. European Journal of Operational Research 96, 205– 216.
- Kouvelis, P., Daniels, R.L., Vairaktarakis, G., 2000. Robust scheduling of a two-machine flow shop with uncertain processing times. IIE Transactions 32, 421–432.
- Kouvelis, P., Yu, G., 1997. Robust Discrete Optimization and its Applications. Kluwer Academic Publishers, Boston.
- Laguna, M., Lino, P., Pérez, A., Quintanilla, S., Valls, V., 2000. Minimizing weighted tardiness of jobs with stochastic interruptions in parallel machines. European Journal of Operational Research 127, 444–457.
- Laslo, Z., 2003. Activity time-cost tradeoffs under time and cost chance constraints. Computers and Industrial Engineering 44, 365–384.

- Leon, V.J., Wu, S.D., Storer, R.H., 1994. Robustness measures and robust scheduling for job shops. IIE Transactions 26 (5), 32–43.
- Leus, R., 2003. The generation of stable project planscomplexity and exact algorithms, Ph.D. Thesis, K.U. Leuven.
- Leus, R., Herroelen, W., 2004. Stability and resource allocation in project planning. IIE Transactions 36, 667–682.
- Mauguière, P., Billaut, J.-C., Artigues, C., 2002. Grouping jobs on a single machine with heads and tails to represent a family of dominant schedules. Paper presented at the 8th Workshop on Project Management and Scheduling, Valencia, 3–5 April.
- Mehta, S.V., Uzsoy, R.M., 1998. Predictable scheduling of a job shop subject to breakdowns. IEEE Transactions on Robotics and Automation 14 (3), 365–378.
- Mehta, S.V., Uzsoy, R., 1999. Predictive scheduling of a single machine subject to breakdowns. International Journal of Computer Integrated Manufacturing 12 (1), 15– 38.
- Möhring, R.H., Radermacher, F.J., Weiss, G., 1984. Stochastic scheduling problems. I. General strategies. ZOR—Zeitschrift für Operations Research 28, 193–260.
- Möhring, R.H., Radermacher, F.J., Weiss, G., 1985. Stochastic scheduling problems. II. Set strategies. ZOR—Zeitschrift für Operations Research 29, 65–104.
- Möhring, R.H., Skutella, M., Stork, F., 2000. Scheduling with AND/OR Precedence Constraints, Technical Report 689/ 2000. Department of Mathematics, Technische Universität Berlin, Germany.
- Möhring, R.H., Stork, F., 2000. Linear preselective policies for stochastic project scheduling. Mathematical Methods of Operations Research 52, 501–515.
- Naegler, G., Schoenherr, S.S., 1989. Resource allocation in a network model—the Leinet system. In: Slowinski, R., Weglarz, J. (Eds.), Advances in Project Scheduling. Elsevier.
- Neumann, K., 1999. Scheduling of projects with stochastic evolution structure. In: Weglarz, J. (Ed.), Project Scheduling—Recent Models, Algorithms and Applications. Kluwer Academic publishers, Boston, pp. 309–332 (Chapter 14).
- Özdamar, L., Alanya, E., 2000. Uncertainty modelling in software development projects (with case study). Annals of Operations Research 102, 157–178.
- Patterson, J.H., 1984. A comparison of exact procedures for solving the multiple constrained resource project scheduling problem. Management Science 30, 854–867.
- Penz, B., Rapine, C., Trystram, D., 2001. Sensitivity analysis of scheduling algorithms. European Journal of Operational Research 134, 606–615.
- Pet-Edwards, J., 1996. A simulation and genetic algorithm approach to stochastic resource-constrained project scheduling. Southcon Conference Record 1996. IEEE, Pascataway, NJ, pp. 333–338.
- Pet-Edwards, J., Selim, B., Armacost, R.L., Fernandez, A., 1998. Minimizing risk in stochastic resource-constrained

project scheduling. Paper presented at the INFORMS Fall Meeting, Seattle, October 25–28.

- Radermacher, F.J., 1985. Scheduling of project networks. Annals of Operations Research 4, 227–252.
- Rommelfanger, H., 1990. FULPAL: An Interactive Method for Solving (Multiobjective) Fuzzy Linear Programming Problems. In: Slowinski, R., Teghem, J. (Eds.), Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming Under Uncertainty. Kluwer Academic Publishers, Dordrecht, pp. 279–299 (Section 5).
- Sabuncuoglu, I., Bayiz, M., 2000. Analysis of reactive scheduling problems in a job shop environment. European Journal of Operational Research 126, 567–586.
- Sadeh, N., Otsuka, S., Schelback, R., 1993. Predictive and reactive scheduling with the microboss production scheduling and control system. In: Proceedings of the IJCAI-93 Workshop on Knowledge-Based Production Planning, Scheduling and Control, pp. 293–306.
- Scholl, A., 2001. Robuste Planung und Optimierung—Grundlagen, Konzepte und Methoden, Experimentelle Untersuchungen. Physica-Verlag, Heidelberg.
- Sevaux, M., Sörensen, K., 2002a. A genetic algorithm for robust schedules. Paper presented at the 8th International Workshop on Project Management and Scheduling, Valencia, April 3–5.
- Sevaux, M., Sörensen, K., 2002b. A genetic algorithm for robust schedules in a just-in-time environment, Research Report LAMIH/SP-2003-1. University of Valenciennes, France.
- Slowinski, R., Hapke, M. (Eds.), 2000. Scheduling under Fuzziness. Physica-Verlag, Heidelberg.
- Smith, S.S., 1994. Reactive scheduling systems. In: Brown, D.E., Scherer, W.T. (Eds.), Intelligent Scheduling Systems. Kluwer.
- Stork, F., 2000. Branch-and-bound algorithms for stochastic resource-constrained project scheduling, Research Report No. 702/2000. Technische Universität Berlin.
- Stork, F., 2001. Stochastic resource-constrained project scheduling. Ph.D. Thesis. Technische Universität Berlin.
- Szelke, E., Kerr, R.M., 1994. Knowledge-based reactive scheduling. Production Planning and Control 5 (2), 124–145.

- Tavares, L.V., Ferreira, J.A.A., Coelho, J.S., 1998. On the optimal management of project risk. European Journal of Operational Research 107, 451–469.
- Tereso, A.P., 2002. Project management—adaptive resource allocation in multimodal activity networks. PhD Thesis. Universidade do Minho, Portugal.
- Tereso, A.P., Araújo, M.M.T., Elmaghraby, S.E., 2003. Experimental results of an adaptive resource allocation technique to stochastic multimodal projects. In: Proceedings of the International Conference on Industrial Engineering and Production Management, Porto, June 26– 28.
- Tsai, Y.W., Gemmil, D.D., 1996. Using a simulated annealing algorithm to schedule activities of resource-constrained projects, Working Paper No. 96-124. Iowa State University.
- Tsai, Y.W., Gemmil, D.D., 1998. Using tabu search to schedule activities of stochastic resource-constrained projects. European Journal of Operational Research 111, 129–141.
- Valls, V., Laguna, M., Lino, P., Pérez, A., Quintanilla, S., 1999. Project scheduling with stochastic activity interruptions. In: Weglarz, J. (Ed.), Project Scheduling—Recent Models, Algorithms and Applications. Kluwer Academic Publishers, pp. 333–353 (Chapter 15).
- Vieira, G.E., Herrmann, J.W., Lin, E., 2003. Rescheduling manufacturing systems: A framework of strategies, policies and methods. Journal of Scheduling 6, 39–62.
- Wang, J.R., 1999. A fuzzy set approach to activity scheduling for product development. Journal of the Operational Research Society 50, 1217–1228.
- Wang, J., 2002. A fuzzy project scheduling approach to minimize schedule risk for product development. Fuzzy Sets and Systems 127 (2), 99–116.
- Wang, J., 2004. A fuzzy robust scheduling approach for product development projects. European Journal of Operational Research 152, 180–194.
- Wollmer, R.D., 1985. Critical path planning under uncertainty. Mathematical Programming Study 25, 164–171.
- Wu, S.D., Storer, R.H., Chang, P.C., 1993. One machine rescheduling heuristics with efficiency and stability as criteria. Computers and Operations Research 20, 1–14.

306