KATHOLIEKE
UNIVERSITEIT
LEUVEN

DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

RESEARCH REPORT 0220

ON THE CONSTRUCTION OF STABLE PROJECT BASELINE SCHEDULES

by
W. HERROELEN
R. LEUS

D/2002/2376/20

On the Construction of Stable Project Baseline Schedules

Willy Herroelen and Roel Leus

March 2002

Operations Management Group
Department of Applied Economics
Katholieke Universiteit Leuven
Naamsestraat 69, B-3000 Leuven (Belgium)
Phones +32-16-32 69 67 and +32-16-32 69 70
Fax +32-16-32 67 32
e-mail: <first name>.<name>@econ.kuleuven.ac.be

On the Construction of Stable Project Baseline Schedules

Willy Herroelen and Roel Leus[§]

ABSTRACT

The vast majority of project scheduling efforts 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. In reality, however, project activities are subject to considerable uncertainty, which generally leads to numerous schedule disruptions. It is of interest to develop pre-schedules that can absorb disruptions in activity durations without affecting the planning of other activities, such that co-ordination of resources and material procurement for each of the activities can be performed as smoothly as possible. The objective of this paper is to develop and evaluate various approaches for constructing a *stable* pre-schedule, which is unlikely to undergo major changes when it needs to be repaired as a reaction to minor activity duration disruptions.

Keywords: *project management and scheduling; risk analysis; stability.*

[§]Research Assistant of the Fund for Scientific Research – Flanders (Belgium) (F.W.O.)

1. Introduction

It is a well-known fact that project activities are subject to considerable uncertainty, which may lead to multiple schedule disruptions during project execution. As a result, the random nature of activity durations has been the subject of numerous research efforts since the introduction of the initial PERT model (Malcolm et al., 1959; Adlakha and Kulkarni, 1989; Valls et al., 1998; Stork, 2001). The issues of project management under uncertainty and risk management have recently received growing attention (Meredith and Mantel, 2000; Goldratt, 1997; Chapman and Ward, 1997). Nevertheless, the development of a pre-computed baseline schedule (pre-schedule) with the objective of creating stability in the start times of the activities, rather than the minimization of the expected project duration or some other regular objective function, has been mostly overlooked so far.

In a multi-project environment, it may be necessary to make advance bookings of key staff or equipment to guarantee their availability (Bowers, 1995). Hence, the ability of the pre-schedule to absorb disruptions may be very important in such settings. Other sources of the need for such stability can be hard delivery dates of suppliers or subcontractors, or in a larger sense, a hard due date for intermediate or final deliverables (e.g. milestones), in other words any time restriction that is external to the project itself. This paper will be concerned with the development of a pre-schedule that can absorb disruptions in activity durations without affecting the planning of other activities, such that co-ordination of resources and material procurement for each of the activities can be performed as smoothly as possible. This objective was termed 'just-in-case' scheduling by Akturk and Gorgulu (1999).

The literature on stable project baseline scheduling is virtually void. Our use of the term 'stability' should not be confused with 'stable project scheduling', as defined by Neumann et al. (2000), who investigate the effect of shifts of sets of activities on schedules for resource-constrained project scheduling problems with minimum and maximum time-lags and define stable schedules as schedules that do not allow so-called oppositely directed shifts.

Mehta and Uzsoy (1998) study the development of stable pre-schedules in a job shop environment, where uncertainty results from machine breakdowns rather than activity duration variability. They state that a predictive schedule or pre-schedule serves two important functions. The first is to allocate resources to the different jobs to optimise some measure of (shop) 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 committing to shipping dates to customers. The authors present a linear programming based heuristic that minimizes the deviation of the starting times compared with a schedule that contains ample slack, while respecting a deadline for the project.

Tavares et al. (1998) study project risk as a function of the uncertainty in both the duration and cost of each project activity. The authors argue that the use of the earliest (latest) start time for each activity reduces (increases) the risk of an overall delay but increases (decreases) the project's discounted cost and therefore an optimal compromise has to be achieved. They present a scheduling approach that trades off the discounted project cost and the risk of not meeting the project completion time by augmenting the earliest possible start time of each project activity by the same fraction (the so-called float factor) of the total float of that activity.

A number of other studies in the machine scheduling literature (Bean et al. (1991), Wu et al. (1993), Akturk and Gorgulu (1999), Alagoz and Azizoglu (2001)) study reactive scheduling policies, which prescribe how to react to schedule disruptions. In effect, various algorithms are proposed to 'match-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.

Bolat (2000) studies the problem of assigning arriving aircraft to available gates at an airport for a given flight schedule. He states that, to take into account the dynamic nature of the problem, one wishes to make the assignments insensitive to variations in flight schedules. For a departure, protection of successive assignments for a given gate amounts to the idle time after the departure. Since the total scheduled ground time of flights and the total time available at the gates are constant, the author states that this goal can only be achieved if the idle times (free floats) are spread evenly among the gates. He then looks at the minimization of the variance of the idle times. A first thing to note, however, is that in many situations, be they aircraft assignment or a general project setting, information will be available about the plausibility of disruptions: activities are all not equally likely to be disturbed, and will not have the same expected disturbance length. Also, some activity start times may need to be better protected than others, for instance because of varying difficulties to release the required resources at later times, or for reasons of co-ordination with external parties, or simply because of the value to the customer of the projected dates being met. Another problem that can occur when the objective is to spread out free float evenly, is that propagation of a disturbance throughout the network is not taken into account: an activity can not only be disturbed by delays in its immediate predecessors, but also because of disruptions of its transitive predecessors that could not be completely absorbed before reaching the activity.

Two mathematical programming models are developed in this paper. The first model aims at minimizing the expected weighted deviation of the actual from the planned activity start times when exactly one activity duration disruption is anticipated, while the second anticipates two disturbances. Three additional models are developed to serve as benchmarks. The first model is the maximization of the weighted sum of buffer sizes, regardless of how large the buffers become. This model will reveal some links with early papers in network flow theory. The second model adapts the aforementioned linear programming based heuristic, originally developed by Mehta and Uzsoy (1998) for job shop scheduling. The third model is an adapted version of the above mentioned float factor model developed by Tavares et al. (1998). All the models make abstraction of resource usage, assuming proper allocation of resources has been performed. In such cases, reactive scheduling is trivial and we can devote our attention to the development of the pre-schedule.

The organisation of this paper is as follows. The necessary notation and definitions are provided in the next section. Our approaches for creating stable pre-schedules are described in Section 3, and some adaptations to existing models that will serve as benchmark heuristics are presented in Section 4. The computational results obtained with the various models on a set of randomly generated problem instances are described in Section 5. Finally, in Section 6, we provide overall conclusions and suggestions for future research.

2. Notation and definitions

We assume that a project is represented in activity-on-the-node format by a directed acyclic graph $G(N,A)$, where N is the set of nodes, representing the project activities, and A is the set of arcs, representing the finish-start precedence relations with timelag 0 that hold between pairs of activities. By $P(i,j)$, we denote any path from i to j in $G(N,A)$. TA is the transitive closure of A, meaning that $(i,j)\in TA$ if and only if some $P(i,j)$ exists. Activity 0 denotes the dummy start node of the network and $n=|N|-1$ denotes the dummy end node. We define set $\pi(i)$ to be the set of all immediate predecessors of activity i in A, and $\sigma(i)$ to be the set of all immediate successors of i in A. Obviously, the sets $\pi(0)$ and $\sigma(n)$ are empty. π^T and σ^T are similar mappings from N to 2^N, based on TA rather than A. We assume A to be minimal, that is $\neg(\exists(i,j)\in A,\ k\in N:\ k\in\sigma(i)\wedge j\in\sigma^T(k))$.

A schedule S is completely defined by a set of activity start times s_i, $i\in N$. The finish time f_i of an activity i occurs d_i time units after its start, where $d_i\in\mathbb{N}$ denotes the duration of activity i. By a 'pre-schedule' or 'baseline schedule', we refer to a schedule that is generated prior to project execution. This pre-schedule will invariably undergo changes during execution, due to the uncertain character of the initially projected activity durations.

Extra definitions of schedule based activity float values will enable us to capture extra information about a pre-schedule. For a given schedule S and $(i,j)\in TA$, pairwise float $F_{ij}(S)$ is defined as $s_j(S)-f_i(S)$, the time margin that exists between i and j in schedule S. $F_{ij}(S)$ is undefined for $(i,j)\notin TA$. We will define total float and free float values for every activity i in a schedule S. The free float $FF_i(S)$ of activity i is defined as $\min_{j\in\sigma(i)}F_{ij}(S)$, $\forall i\in N$, which can be seen to represent the amount of time activity i can be delayed without affecting the start time of another activity in the schedule. Given a project deadline $\omega\geq f_n(S)$, we set the latest allowable finish time of the dummy end activity $lf_n:=\omega$ and define the latest allowable start time, ls_i, and the latest allowable finish time, lf_i, of activity i according to classical CPM backward calculations. The schedule with all activities scheduled at their latest allowable start times will be referred to as LSS (latest start schedule). We can now derive the total float of activity i in schedule S with deadline ω as $TF_i(S,\omega):=lf_i-f_i(S)$, $\forall i\in N$. It is clear that schedule S is protected against expansion of its projected makespan ω due to inflation of the duration of activity i up to an amount $TF_i(S,\omega)$, disregarding other disturbances. The classical float definitions (cfr. Wiest and Levy, 1977) are obtained when S=ESS, where ESS is the CPM earliest start schedule. In the remainder of this paper, we will most often omit the argument indicating the schedule on referring to float values and other quantities; there will be little danger of confusion.

It can be seen that free float is a characteristic of one activity, while total float is rather attributable to an entire path. If one activity consumes its total float, this will mostly imply also the disappearance of (part of) the total float of its successors: total float represents the available slack if every activity scheduled earlier than the one under consideration is executed according to the pre-schedule S, and all immediate and transitive successors start at their latest allowable start. Free float on the other hand is cumulative, in the sense that, if for a particular activity the free float is exhausted and we evolve through the network towards project completion, there may be free float left for the successors; free float is the maximally allowed disturbance with zero impact on successor start times when both predecessors and successors adhere to S. Logically, to affect the projected makespan by a disturbance

in activity i, all activities on a path between i and n will have to be affected, so TF_i is the minimal sum of the pairwise float values of the edges of $G(N,A)$ over all paths leading from that activity i to n, if we choose $\omega=f_n$.

Figure 1 represents the total float and free float values for the earliest start schedule of a 9-activity project with deadline $\omega=10$.

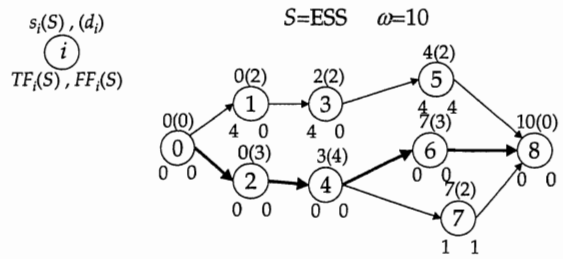

Figure 1. Earliest start schedule with float calculations.

The critical path 0-2-4-6-8 has been indicated in bold. Activities on path 1-3-5 have the same total float due to the absence of intermediate merging paths. Because of the ESS character of the schedule, the finish time of one activity on path 1-3-5 is the start of the next activity on the path. Therefore, $FF_1=F_{13}=FF_3=F_{35}=0$, while $FF_5=F_{58}=4$, which equals the total float for each activity on the path, because the single successor of activity 5 is the dummy end activity 8, and $\omega=s_8=10$.

3. Scheduling for stability

In this section we develop two mathematical programming models for the generation of stable pre-schedules. We minimize the expected weighted deviation of the sum of starting times of the project activities, when exactly one and two activity duration disruptions are known to take place during project execution.

3.1 *Analytic determination of the effects of a single disruption*

We will develop a model that assumes that *one* anticipated disturbance will occur in the network, due to an increase in the duration of a single activity. This setting should not be seen as one where always exactly one disturbance will occur. The underlying idea is that disturbances are sufficiently sparse and sufficiently spread over time and throughout the project network so that we can assume that the effect of one disturbance will not interact with the effects of another. With this approach, we follow Leon et al. (1994), whose objectives entail the minimization of pre-schedule makespan and minimization of the expected makespan delay. The authors argue that the single disruption case provides important insights, and can serve as a basis for treatment of the more general case.

For all pairs $(i,j)\in TA$, let $MSPF_{ij}$ equal the minimal sum of pairwise floats of all edges on any path $P(i,j)$ in $G(N,A)$ (such a path can always be found). The path $P(i,j)$ for which $MSPF_{ij}$ is achieved, will be denoted as $P^*(i,j)$ (when multiple paths are minimal, we just choose one). $MSPF_{ij}$ is undefined $\forall(i,j)\in(N\times N)\setminus TA$. Remark that matrix $MSPF$ resembles the matrix of shortest paths between all pairs of nodes for an activity-on-the-arc project network. $MSPF$ is a function of the pre-schedule S at

hand: for given values of $F_{ij}(S)$ for all $(i,j)\in A$, we could in principle obtain the relevant entries in the *MSPF* matrix by the following linear program.

$$\text{(SUM)} \quad \max \sum_{(i,j)\in TA} MSPF_{ij} \tag{1}$$

subject to

$$MSPF_{ii} = 0 \qquad \forall\, i \in N \tag{2}$$

$$MSPF_{ij} \le F_{ik} + MSPF_{kj} \qquad \forall (i,j)\in TA,\ \forall k \in \sigma(i)\cap(\pi^T(j)\cup\{j\}) \tag{3}$$

(SUM) is similar to the conceptual linear model given by Lawler (1976), which exploits the all-pairs shortest path optimality conditions associated with an activity-on-the-arc network (Lawler (1976) mentions (3) for *all* (i,j,k) triples – we have limited the constraints to only those corresponding with (backward) single pair shortest path optimality conditions). Eqs. (2) set boundary values that help to derive upper bounds on the minimal sum of pairwise floats between precedence related activities i and j in Eqs. (3).

We assign a probability of disruption p_i to each activity i, with $\sum_{i=0}^{n-1} p_i = 1$ and $p_n=0$; p_0 is the probability that the entire project starts later than initially anticipated. $G_i(\cdot)$ is the cumulative distribution function (cdf) of the disturbance length L_i of activity i *if* i is disturbed. Let $c_i\in\mathbb{N}$ denote the nonnegative cost per unit time overrun on the start time of activity i; $c_0=0$. The pre-schedule stability measure we propose is *the expected weighted deviation in start times* in the *realized* schedule from those in the pre-schedule. The expression we wish to minimize is $\sum_{j=1}^{n} c_j(ES_j - s_j)$, with E the expectation operator and S_j a random variable representing the actually achieved start time of activity j (after project execution). We can compute ES_j as $s_j + \sum_{i\in\pi^T(j)} p_i E(\max\{0; L_i - MSPF_{ij}\} \mid i \text{ disturbed})$. Hence, the objective can be rewritten as

$$\min \sum_{(i,j)\in TA} c_j p_i E(\max\{0; L_i - MSPF_{ij}\} \mid i \text{ disturbed}),$$

which makes clear that we deal with a (separable) non-linear programming problem, where the link between a schedule and the objective is solely via *MSPF*. To render solution less troublesome, we shall assume all L_i to be discrete, with probability mass function (pmf) $g_i(\cdot)$ that associates nonzero probability with positive values l_{ik} which correspond with the elements k in E_i, the set of disturbance scenarios for activity i. Our choice for discrete scenarios is somewhat comparable with Kouvelis and Yu (1997); the present paper considers the expected value objective, however, while Kouvelis and Yu opt for minimax and minimax regret objectives (cfr. Rosenhead et al., 1972). The problem can now be cast into a linear programming formulation:

$$\text{(EWD1)} \quad \min \sum_{(i,j)\in TA} \sum_{k\in E_i} c_j p_i g_i(l_{ik}) \Delta_{ijk} \tag{4}$$

subject to

$$s_i + d_i + F_{ij} = s_j \qquad \forall\, (i,j) \in A \tag{5}$$

$$s_n \le \omega \tag{6}$$

$$l_{ik} - MSPF_{ij} \le \Delta_{ijk} \qquad \forall\, (i,j) \in TA,\ \forall k \in E_i \tag{7}$$

(2)-(3)

all $\Delta_{ijk}, s_i, F_{ij}, MSPF_{ij} \geq 0$

Δ_{ijk} is the delay in the start time of activity j due to a disturbance according to scenario k of activity i. All separate *MSPF* entries will be assigned a large enough value because of their relation with the objective function (4), a property that is achieved in (SUM) by explicitly including the sum of all matrix elements in the objective. Protection against makespan perturbation is incorporated by means of cost coefficient c_n. Eq. (6) imposes a deadline of ω on the project completion; we assume that $\omega \geq s_n$(ESS) such that a feasible solution exists.

For an activity pair $(i,j) \in TA$, the model as formulated above will re-identify the shortest path of floats $P^*(i,j)$ for every schedule encountered in the search process. This is unnecessary, however: for any feasible choice of starting times s_i and s_j, $P^*(i,j)$ will be the path with largest sum of activity durations, because for every path from i to j, the sum of floats (corresponding with edges) plus the sum of activity durations (corresponding with nodes, excluding i and j) equals $F_{ij}=s_j-s_i-d_i$. In conclusion, equation (5°) holds, with λ_{ij} the length of $P^*(i,j)$, not including i and j. (EWD1) can be re-written by dropping constraints (2) and (3) and replacing (5) by (5°).

$$s_i + d_i + \lambda_{ij} + MSPF_{ij} = s_j \qquad \forall\ (i,j) \in TA \qquad (5°)$$

As a side note, this allows us to see that $TF_i=MSPF_{in} + (\omega - s_n)$, because λ_{in} is equal to $lf_n-lf_i = \omega-lf_i$. Inequality in (5°) would yield the same results, but we can rewrite (EWD1) and substitute for $MSPF_{ij}$ in (7), hereby eliminating the variable completely. This yields the following formulation:

$$\min \quad \sum_{(i,j)\in TA} \sum_{k\in E_i} \alpha_{ijk}\Delta_{ijk} \qquad (4')$$

subject to

$$s_j - s_i \geq d_i \qquad \forall\ (i,j) \in A \qquad (5')$$

$$s_0 - s_n \geq -\omega \qquad (6')$$

$$\Delta_{ijk} + s_j - s_i \geq \beta_{ijk} \qquad \forall\ (i,j) \in TA, \forall k \in E_i \qquad (7')$$

all $\Delta_{ijk} \geq 0$; all s_i unrestricted in sign

We have $\alpha_{ijk}:=c_j p_i g_i(l_{ik})$ and $\beta_{ijk}:=l_{ik}+d_i+\lambda_{ij}$, for all relevant triples (i,j,k). The model focuses on the relative position of activities in time, rather than absolute values of activity starting times. We can return to absolute values by shifting a solution to $s_0:=0$. If we assign nonnegative multipliers x_{ij}, v and y_{ijk} to the constraints (5′), (6′) and (7′) respectively, the dual of this formulation can be written as follows:

$$\max \quad \sum_{(i,j)\in A} d_i x_{ij} - \omega v + \sum_{\substack{(i,j)\in TA \\ k\in E_i}} \beta_{ijk} y_{ijk} \qquad (8)$$

subject to

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} + \sum_{\substack{(i,j)\in TA \\ k\in E_i}} y_{ijk} - \sum_{\substack{(j,i)\in TA \\ k\in E_j}} y_{jik} = \begin{cases} 0 & \forall i \in N, i \neq 0,n \\ v & i=0 \\ -v & i=n \end{cases} \qquad (9)$$

$$y_{ijk} \leq \alpha_{ijk} \qquad \forall\ (i,j) \in TA, \forall k \in E_i \qquad (10)$$

This can be seen to be a minimum cost network flow problem (MCNFP) on the network $G(N,TA)$ with extra arc $(n,0)$, where each arc (i,j) in the network, except for $(n,0)$, is in fact a multi-arc, representing either $|E_i|+1$ arcs (if $(i,j)\in A$) or $|E_i|$ arcs (if $(i,j)\in TA\setminus A$), with different profits (for the maximization objective) and flow capacities, but identical head and tail node. If we order the scenarios in increasing l_{ik}, then arcs y_{ij1} to $y_{ij|E_i|}$ have increasing profits ($\beta_{ij,k+1}>\beta_{ijk}$, $k=1,\ldots,|E_i|-1$) and are capacitated (Eqs. 10); arc x_{ij} is less profitable than any corresponding y_{ijk} ($\beta_{ijk}>d_i$), and is uncapacitated. Remark that α_{ijk} need not monotonely decrease (or increase) in k, and that all nodes are transshipment nodes (zero supply and demand). We also notice that there are no infinite capacity cycles with positive benefit, as all cycles pass through $(n,0)$ and infinite capacity corresponds only with x_{ij}-arcs (with benefit d_i), and ω is at least as large as the CPM makespan; hence the solution is not unbounded.

Without loss of generality, we choose $s_0:=0$, and the remaining optimal starting times are readily determined from the dual optimum. An arc that carries flow at a value strictly between its lower and upper bound is a 'free arc'. At most one arc corresponding with each multi-arc will be free, because of the structure of the profit coefficients. Complementary slackness conditions tell us that when an optimal solution is obtained in the dual, the free arcs in the dual require the corresponding primal constraints to be satisfied as an equality. We also notice that variable Δ_{ijk} is the primal multiplier associated with dual constraint (10). Complementary slackness implies that this variable be zero whenever the corresponding y_{ijk} is free. In this way, the scenario up to which every second activity of the activity pairs in TA is protected, can be conveniently read from the dual optimum (arcs with no flow), and the primal optimum can thus be constructed. This computation will often even be superfluous, as competitive solution algorithms for MCNFPs typically generate optimal 'node potentials', starting times in our case (or the negative of, using (9)), as well (Ahuja et al., 1993).

We have applied (EWD1) to the example problem shown in Figure 1, assigning all activities i equal cost $c_i=1$ (apart from $c_0=0$) and probability $p_i=1/n$ (apart from $p_n=0$), and considering for each activity a single disturbance scenario with $l_{i1}=1$, and deadline $\omega=10=s_n$(ESS). The corresponding MCNFP has 34 arcs ($|TA|=23$, $|A|=10$, plus return arc $(n,0)$). Compared with ESS, the activities on the critical path remain unchanged, which is logical knowing that the deadline equals the critical path length. However, appropriate buffers are inserted by $F_{01}=F_{13}=F_{35}=F_{47}=1$, sufficient to totally undo a unit length disturbance. It is preferred to protect the start of activity 7 from disruption in activities 0, 2 and 4 rather than to protect activity 8 from activity 7 (a 3-to-1 trade-off for F_{47} against F_{78}). With all costs equal to 1, we obtain an optimal objective value of $11/n$.

3.2 *The case of two disruptions*

When exactly two activities are disturbed, optimization can be performed as follows. We refer to the two (distinct) disturbed activities as a_1 and a_2, which are selected without replacement out of N, with probability of selection of activity i each time proportional to p_i. Activity i thus has a chance of p_i to be elected as the first activity a_1, and a chance of $p_i/(1-p_{a_1})$ that $a_2=i$, if $a_1\neq i$. The expected weighted deviation can be written as

$$\sum_{j\in N} c_j \sum_{i\in N} p_i \sum_{\substack{z\in N\\ z\neq i}} \left(\tfrac{p_z}{1-p_i}\right) E\left(S_j - s_j \mid a_1 = i \wedge a_2 = z\right) = \sum_{j\in N} c_j \sum_{\substack{\{i,z\}\subseteq N\\ z\neq i}} p_i p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) E\left(S_j - s_j \mid a_1 = i \wedge a_2 = z\right).$$

We can rewrite this expression, recognizing the separate cases in which either 0, 1 or both of the activities i and z have a possible impact on j, and obtain

$$\sum_{j\in N} c_j \sum_{\substack{i\in\pi^T(j)\\ z\notin\pi^T(j)}} p_i p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) E\left(S_j - s_j \mid a_1 = i \wedge a_2 \notin \pi^T(j)\right) +$$

$$\sum_{j\in N} c_j \sum_{\substack{\{i,z\}\subseteq\pi^T(j)\\ i\neq z}} p_i p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) E\left(S_j - s_j \mid a_1 = i \wedge a_2 = z\right).$$

For a pair $(i,j)\in TA$, let us now define the following subsets of N: $C_1(j) := N\setminus\pi^T(j)$, $C_2(i,j) := \pi^T(j) \cap \sigma^T(i)$, $C_3(i,j) := \pi^T(i)$ and $C_4(i,j) := \pi^T(j) \setminus (\pi^T(i)\cup\{i\}\cup\sigma^T(i))$ ($C_3(i,j)$ is only a function of i, but we maintain the double argument). We see that these sets are mutually exclusive and their union with $\{i\}$ is N. If $z\in C_1(j)$, it will never affect the starting time of j; this includes the case $z=j$. $z\in C_2(i,j)$ means that z is on a path from i to j, and so the disruption lengths of i and z might (completely or partially) accumulate to disrupt the starting time of j. $z\in C_3(i,j)$ corresponds with the cases where z is on a path from 0 to i, with similar consequences. Finally, $z\in C_4(i,j)$ means that disruption of z might affect j, but disruptions of i and z will not be accumulated. Since $z\in C_4(i,j)\Leftrightarrow i\in C_4(z,j)$ and $z\in C_2(i,j)\Leftrightarrow i\in C_3(z,j)$, we have to watch out for double counting. This is accomplished by omitting $z\in C_2(i,j)$, and by inserting coefficient $\left(\frac{1}{2}\right)$ when $z\in C_4(i,j)$. The expected weighted deviation reduces to

$$\sum_{(i,j)\in TA} p_i c_j \left(\sum_{z\in C_1(j)} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right)\right) E\left(S_j - s_j \mid a_1 = i \wedge a_2 \notin \pi^T(j)\right) +$$

$$\sum_{(i,j)\in TA} p_i c_j \sum_{z\in C_3(i,j)} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) E\left(S_j - s_j \mid a_1 = i \wedge a_2 = z\right) +$$

$$\sum_{(i,j)\in TA} p_i c_j \sum_{z\in C_4(i,j)} \tfrac{1}{2} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) E\left(S_j - s_j \mid a_1 = i \wedge a_2 = z\right).$$

When $(i,j)\in TA$, we have

- $E\left(S_j - s_j \mid a_1 = i \wedge a_2 \in C_1(j)\right) = E(\max\{0 ; L_i - MSPF_{ij}\} \mid \ldots)$, (11a)
  similar to (EWD1).
- $E\left(S_j - s_j \mid a_1 = i \wedge a_2 = z \wedge z \in C_3(i,j)\right) =$
  $E(\max\{0 ; L_z - MSPF_{zj} ; \max\{0 ; L_z - MSPF_{zi}\} + L_i - MSPF_{ij}\} \mid \ldots) =$
  $E(\max\{0 ; L_z - MSPF_{zj} ; L_i - MSPF_{ij} ; L_z - MSPF_{zi} + L_i - MSPF_{ij}\} \mid \ldots)$. (11b)
  The computation for $z\in C_2(i,j)$ is analogous, but not required here.
- $E\left(S_j - s_j \mid a_1 = i \wedge a_2 = z \wedge z \in C_4(i,j)\right) =$
  $E(\max\{\max\{0 ; L_i - MSPF_{ij}\} ; \max\{0 ; L_z - MSPF_{zj}\}\} \mid \ldots) =$
  $E(\max\{0 ; L_i - MSPF_{ij} ; L_z - MSPF_{zj}\} \mid \ldots)$ (11c)

The expected value operator applies to L_i and L_z in (11b) and (11c). To limit the number of scenarios to be considered in the model (normally the Cartesian product of scenarios of i and z), we exchange expectation with respect to L_z and the max operator. By Jensen's inequality, this is an underestimation of the disruption length (if $|E_z| = 1$, the result is exact). The approximation implies that we do not explicitly consider all scenarios of z separately, but rather use only EL_z. For any pair of activities i and z such that $i,z\in\pi^T(j)$ and $z\in C_4(i,j)$ for some activity j, both z and i are approximated by their expected disruption length one out of the two times the joint

effect of i and z on the start time of j is examined. Because each effect is examined twice, we use coefficient $\left(\frac{1}{2}\right)$. When $i,z \in \pi^T(j)$ and $z \in C_3(i,j)$, we approximate L_z by EL_z and cover the effect of $a_1=z$ by the same quantity (so no coefficient $\left(\frac{1}{2}\right)$ is needed).

Based on the foregoing, we will minimize (the approximation of) the expected weighted deviation by the following model. Notice that we implicitly compare with Δ_{ijk} three times. We do not reuse Δ_{ijk} itself, however, for this would result in problems to rewrite (EWD2) as the dual of a MCNFP. For the same reason, we have separated out the nested max operators in (11b).

(EWD2) $$\min \sum_{(i,j)\in TA} \sum_{k\in E_i} \alpha_{ijk} \left(P_{ij}^{[1]} \Delta_{ijk} + \sum_{z\in C_3(i,j)} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) D_{ijkz}^{[3]} + \sum_{z\in C_4(i,j)} \tfrac{1}{2} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) D_{ijkz}^{[4]} \right)$$

subject to

$$s_i + d_i \le s_j \qquad \forall\ (i,j) \in A$$

$$s_n \le \omega$$

$$l_{ik} - MSPF_{ij} \le \Delta_{ijk} \qquad \forall\ (i,j) \in TA,\ \forall k \in E_i$$

$$l_{ik} - MSPF_{ij} \le D_{ijkz}^{[c]} \qquad c=3,4,\ \forall (i,j)\in TA,\ \forall k\in E_i,\ \forall z\in C_c(i,j)$$

$$EL_z - MSPF_{zj} \le D_{ijkz}^{[c]} \qquad c=3,4,\ \forall (i,j)\in TA,\ \forall k\in E_i,\ \forall z\in C_c(i,j)$$

$$EL_z + l_{ik} - MSPF_{zi} - MSPF_{ij} \le D_{ijkz}^{[3]} \qquad \forall (i,j)\in TA,\ \forall k\in E_i,\ \forall z\in C_3(i,j)$$

the $MSPF$ values are exact

all Δ_{ijk}, $D_{ijkz}^{[c]}$, $s_i \ge 0$

with $P_{ij}^{[1]} := \sum_{z\in C_1(j)} p_z \left(\frac{1}{1-p_i} + \frac{1}{1-p_z}\right)$, for $(i,j)\in TA$. The constraints can be rewritten as follows (we again make use of equation (5°)):

(x_{ij})	$s_j - s_i \ge d_i$	$\forall\ (i,j) \in A$
(v)	$s_0 - s_n \ge -\omega$	
(y_{ijk})	$\Delta_{ijk} + s_j - s_i \ge \beta_{ijk}$	$\forall\ (i,j) \in TA,\ \forall k \in E_i$
$(w_{ijkz}^{[c]})$	$D_{ijkz}^{[c]} + s_j - s_i \ge \beta_{ijk}$	$c=3,4,\ \forall (i,j)\in TA,\ \forall k\in E_i,\ \forall z\in C_c(i,j)$
$(u_{ijkz}^{[c]})$	$D_{ijkz}^{[c]} + s_j - s_z \ge \gamma_{zj}$	$c=3,4,\ \forall (i,j)\in TA,\ \forall k\in E_i,\ \forall z\in C_c(i,j)$
(q_{ijkz})	$D_{ijkz}^{[3]} + s_j - s_z \ge \gamma_{zi} + \beta_{ijk}$	$\forall (i,j)\in TA,\ \forall k\in E_i,\ \forall z\in C_3(i,j)$

all Δ_{ijk}, $D_{ijkz}^{[c]} \ge 0$; all s_i unrestricted in sign

with $\gamma_{zj} := EL_z + d_z + \lambda_{zj}$. We indicate between parentheses the nonnegative dual multipliers assigned to each set of constraints. The dual of (EWD2) has constraints

$$(s_i) \qquad \sum_{j\in\sigma(i)} x_{ij} - \sum_{j\in\pi(i)} x_{ji} + \sum_{\substack{j\in\sigma^T(i)\\ k\in E_j}} \left(y_{ijk} + \sum_{c=3,4} \sum_{z\in C_c(i,j)} w_{ijkz}^{[c]} \right) - \sum_{\substack{j\in\pi^T(i)\\ k\in E_j}} \left(y_{jik} + \sum_{c=3,4} \sum_{z\in C_c(j,i)} w_{jikz}^{[c]} \right)$$

$$+ \sum_{j\in\sigma^T(i)} \left(\sum_{\substack{z:i\in C_3(z,j)\\ k\in E_z}} q_{zjki} + \sum_{c=3,4} \sum_{\substack{z:i\in C_c(z,j)\\ k\in E_z}} u_{zjki}^{[c]} \right) - \sum_{j\in\pi^T(i)} \left(\sum_{\substack{z\in C_3(j,i)\\ k\in E_j}} q_{jikz} + \sum_{c=3,4} \sum_{\substack{z\in C_c(j,i)\\ k\in E_j}} u_{jikz}^{[c]} \right) = \begin{cases} 0 & \forall i\in N, i\ne 0,n \\ v & i=0 \\ -v & i=n \end{cases}$$

$$(\Delta_{ijk}) \qquad y_{ijk} \le \alpha_{ijk} P_{ij}^{[1]} \qquad \forall (i,j) \in TA,\ \forall k \in E_i$$

$$(D_{ijkz}^{[3]}) \qquad w_{ijkz}^{[3]} + u_{ijkz}^{[3]} + q_{ijkz} \le \alpha_{ijk} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) \qquad \forall (i,j) \in TA,\ \forall k \in E_i,\ \forall z \in C_3(i,j)$$

$$(D_{ijkz}^{[4]}) \qquad w_{ijkz}^{[4]} + u_{ijkz}^{[4]} \le \tfrac{1}{2} \alpha_{ijk} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) \qquad \forall (i,j) \in TA,\ \forall k \in E_i,\ \forall z \in C_4(i,j)$$

subject to which

$$\sum_{\substack{(i,j)\in TA \\ k\in E_i}} \left[\beta_{ijk} \left(y_{ijk} + \sum_{c=3,4} \sum_{z\in C_c(i,j)} w_{ijkz}^{[c]} \right) + \sum_{c=3,4} \sum_{z\in C_c(i,j)} \gamma_{zj} u_{ijkz}^{[c]} + \sum_{z\in C_3(i,j)} (\gamma_{zi} + \beta_{ijk}) q_{ijkz} \right] + \sum_{(i,j)\in A} d_i x_{ij} - \omega v$$

has to be maximized. We now rewrite the dual constraints:

$$(s_i) \qquad \sum_{j\in\sigma(i)} x_{ij} - \sum_{j\in\pi(i)} x_{ji} + \sum_{\substack{j\in\sigma^T(i) \\ k\in E_j}} \left(y_{ijk} + \sum_{c=3,4} \sum_{z\in C_c(i,j)} w_{ijkz}^{[c]} \right) - \sum_{\substack{j\in\pi^T(i) \\ k\in E_j}} \left(y_{jik} + \sum_{c=3,4} \sum_{z\in C_c(j,i)} f_{jikz}^{[c]} \right)$$

$$+ \sum_{j\in\sigma^T(i)} \left(\sum_{\substack{z:i\in C_3(z,j) \\ k\in E_z}} q_{zjki} + \sum_{c=3,4} \sum_{\substack{z:i\in C_c(z,j) \\ k\in E_z}} u_{zjki}^{[c]} \right) = \begin{cases} 0 & \forall i \in N, i \ne 0, n \\ v & i = 0 \\ -v & i = n \end{cases}$$

$$(\Delta_{ijk}) \qquad y_{ijk} \le \alpha_{ijk} P_{ij}^{[1]} \qquad \forall (i,j) \in TA,\ \forall k \in E_i$$

$$(F_{ijkz}^{[3]}) \qquad w_{ijkz}^{[3]} + u_{ijkz}^{[3]} + q_{ijkz} = f_{ijkz}^{[3]} \qquad \forall (i,j) \in TA,\ \forall k \in E_i,\ \forall z \in C_3(i,j)$$

$$(D_{ijkz}^{[3]}) \qquad f_{ijkz}^{[3]} \le \alpha_{ijk} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) \qquad \forall (i,j) \in TA,\ \forall k \in E_i,\ \forall z \in C_3(i,j)$$

$$(F_{ijkz}^{[4]}) \qquad w_{ijkz}^{[4]} + u_{ijkz}^{[4]} = f_{ijkz}^{[4]} \qquad \forall (i,j) \in TA,\ \forall k \in E_i,\ \forall z \in C_4(i,j)$$

$$(D_{ijkz}^{[4]}) \qquad f_{ijkz}^{[4]} \le \tfrac{1}{2} \alpha_{ijk} p_z \left(\tfrac{1}{1-p_i} + \tfrac{1}{1-p_z}\right) \qquad \forall (i,j) \in TA,\ \forall k \in E_i,\ \forall z \in C_4(i,j)$$

with $f_{ijkz}^{[c]}$ additional nonnegative variables, defined for appropriate (c,i,j,k,z)-tuples. The dual problem formulated in this way can be solved as a MCNFP on an extended network which contains all nodes in N, and additionally $\forall (i,j) \in TA$, $\forall k \in E_i$, extra node $n_{ijkz}^{[c]}$, $\forall z \in C_c(i,j)$, c=3,4. The network thus has $1+n+\sum_{(i,j)\in TA} |E_i| \sum_{c=3,4} |C_c(i,j)|$ nodes and $1+|A|+\sum_{(i,j)\in TA} |E_i| \left(1 + 4|C_3(i,j)| + 3|C_4(i,j)|\right)$ arcs. The transformation effected by rewriting the dual constraints is illustrated in Figure 2 for c=3 (a similar transformation is performed for $n_{ijkz}^{[4]}$, without arc q_{ijkz}).

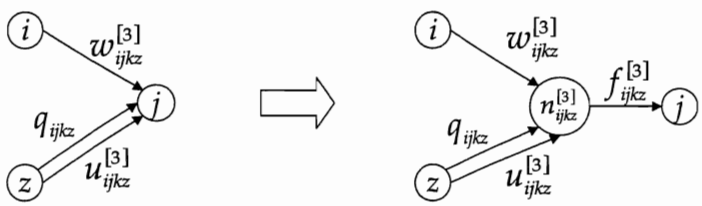

Figure 2. Illustration of the network transformation. $i,j,z \in N$, $k \in E_i$ and $z \in C_3(i,j)$.

To show that this formulation is indeed equivalent to (EWD2), we 'un-dualize' the formulation with the multipliers indicated between parentheses (nonnegative for $\le$-constraints, unrestricted in sign for =-constraints). This yields:

(UND) $\min \sum_{(i,j)\in TA} \sum_{k\in E_i} \alpha_{ijk} \left(P_{ij}^{[1]} \Delta_{ijk} + \sum_{z\in C_3(i,j)} p_z \left(\frac{1}{1-p_i} + \frac{1}{1-p_z}\right) D_{ijkz}^{[3]} + \sum_{z\in C_4(i,j)} \frac{1}{2} p_z \left(\frac{1}{1-p_i} + \frac{1}{1-p_z}\right) D_{ijkz}^{[4]} \right)$

subject to

(x_{ij})	$s_j - s_i \geq d_i$	$\forall\ (i,j) \in A$
(v)	$s_0 - s_n \geq -\omega$	
(y_{ijk})	$\Delta_{ijk} + s_j - s_i \geq \beta_{ijk}$	$\forall\ (i,j) \in TA,\ \forall k \in E_i$
$(w_{ijkz}^{[c]})$	$F_{ijkz}^{[c]} - s_i \geq \beta_{ijk}$	c=3,4,$\forall (i,j)\in TA, \forall k\in E_i, \forall z\in C_c(i,j)$
$(u_{ijkz}^{[c]})$	$F_{ijkz}^{[c]} - s_z \geq \gamma_{zj}$	c=3,4,$\forall (i,j)\in TA, \forall k\in E_i, \forall z\in C_c(i,j)$
$(f_{ijkz}^{[c]})$	$-F_{ijkz}^{[c]} + D_{ijkz}^{[c]} + s_j \geq 0$	c=3,4,$\forall (i,j)\in TA, \forall k\in E_i, \forall z\in C_c(i,j)$
(q_{ijkz})	$F_{ijkz}^{[3]} - s_z \geq \gamma_{zi} + \beta_{ijk}$	$\forall (i,j)\in TA,\ \forall k\in E_i,\ \forall z\in C_3(i,j)$

all Δ_{ijk}, $D_{ijkz}^{[c]} \geq 0$; all s_i, $F_{ijkz}^{[c]}$ unrestricted in sign

We easily see that a starting time vector that is optimal to (UND) is also optimal to (EWD2). Thus, the optimal node potentials for the original nodes (elements of N) in the transformed dual correspond with optimal starting times for the original primal. Given the shape of the network corresponding with the MCNFP, infinite capacity cycles can again only use x_{ij} arcs and so for appropriate ω values, we see that the optimal solution cannot be unbounded.

As mentioned, in order to limit the size of the resulting network flow problems, we do not explicitly take up every scenario of activity z in the model. It is possible, however, to include all $z\in C_2(i,j)$ instead of $z\in C_3(i,j)$ into (EWD2). We shall examine the relative performance of these two options in our computational experiments in Section 5. The associated MCNFP will have a comparable number of nodes and arcs, though not necessarily the same ($\sum_{(i,j)\in TA} |C_2(i,j)| = \sum_{(i,j)\in TA} |C_3(i,j)|$, but the number of scenarios may differ between the activities). Also, q_{ijkz} now represents a flow from i to j rather than from z to j.

Application of (EWD2) to the example problem shown in Figure 1, with the same deadline, costs and probabilities as before, yields the same schedule as (EWD1). The number of arcs in the corresponding MCNFP is 216 (part of which are the 34 in (EWD1)), and the number of nodes amounts to 61.

3.4 *Accounting for multiple disruptions*

Depending on the degree of 'manageability' of the project environment, interaction of disturbances will occur to a lesser or larger degree. If activities can be sped up rather easily by introducing overtime, or if the occurrence of disturbances can be virtually eliminated if the activity in question is perceived to be critical because its predecessor(s) suffered delays, the one-disturbance model is a good approximation of reality. If disturbances mainly result from random fluctuations, as it occurs in shop environments, all activity durations are more or less independent, and disturbances will regularly interfere with one another. In such cases, we can anticipate that (EWD2) will yield better results.

To further step up allowance for multiple disturbances, we will add to our model compensation for cumulation of disturbances along paths. We substitute for *MSPF* by determining values λ_{ml} based on activity durations $d_k+\gamma p_k EL_k$ for all activities k on any $P(m,l)$, where γ is parameter that controls the degree in which we

compensate the *MSPF* values for possible cumulation of disruptions. This more or less corresponds with the situation where that starting times become irrelevant and disruptions can be cushioned also towards time zero (e.g. when the inner max operator would be eliminated in (11b)). We neglect extra 'parallel' disruptions (like the ones covered by $D^{[4]}_{ijkz}$ for two disturbed activities) and extra disruptions in $\pi^T(m)$. We use γm rather than γ to make the parameter independent of the number of activities in the project (average probability is $1/n$). This extension is easily applicable to (EWD1) as well as (EWD2). The benefits of this addition to the basic models will be studied in Section 5.

4. Benchmark heuristics

4.1 *Maximizing the weighted sum of pairwise floats*

We will compare the performance of the output schedules of the proposed model with pre-schedules maximizing the *weighted sum of pairwise floats*. This amounts to maximization of the weighted sum of buffer sizes, regardless of how large the buffers become: contrary to (EWD1), no knowledge of scenarios is taken into account. Propagation of floats is considered only by summing over TA in the objective.

(WPF) $$\max \quad \sum_{(i,j)\in TA} c_j p_i F_{ij} \qquad (12)$$

subject to

$$s_i + d_i + F_{ij} \le s_j \qquad \forall\ (i,j) \in TA \qquad (13)$$

$$s_n \le \omega \qquad (14)$$

all $s_i, F_{ij} \ge 0$

We show in the Appendix that when (WPF) is appropriately formulated, its dual can also be seen to be a MCNFP. (WPF) with Eqs. (13) defined for A instead of TA and with minimization rather than maximization objective was studied by Levner and Nemirovski (1994) and Ahuja et al. (1993) in the context of so-called 'just-in-time' scheduling. In fact, very similar models can be found in Fulkerson (1961) and Kelly (1961), where the durations of the activities of an activity-on-the-arc project network are the decision variables, in the context of time/cost or time/utility trade-offs. Contrary to those references, however, model (WPF) does not impose an upper bound on (profit from) the 'buffer size' F_{ij}, which would correspond to eliminating all but the longest scenario per activity. Rather, (WPF) can be seen to be a special case of (EWD1) when each activity has only one disruption scenario, with l_{i1} larger than or equal to the maximally achievable $MSPF_{ij}$ for all pairs (i,j). Put differently, the objective is purely linear in the buffer sizes, and information about a *necessary* magnitude of protection is not entered into the model.

For the example problem shown in Figure 1, the number of arcs of the MCNFP corresponding with (WPF) is equal to 24. It can be easily seen that in the optimum, F_{47}=1 and F_{01}=F_{58}=0. Activity 3 can be positioned anywhere from s_3=2 and s_3=6. The optimal objective function equals $3/n$ (arcs incoming in 7) plus $4/n$ (F_{08})

plus $2.(4/n)$ $(F_{18}+F_{05})$ plus $2.(4/n)$ $(F_{03}+F_{13}+F_{35}+F_{38})$ plus a constant because the pairwise floats sometimes have a nonzero lower bound.

4.2 *The linear programming based heuristic*

In studying predictive scheduling for job shops, Mehta and Uzsoy (1998) insert additional idle time into the schedule to absorb the impact of breakdowns, and invoke earliness or lateness penalties whenever the last operation of a job ends sooner or later than planned. In a project scheduling context, *every* activity should be protected, and no earliness penalties will be imposed. In the 'linear programming based heuristic' (LPH) of Mehta and Uzsoy (1998), the idea then 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. We will translate the ideas of the authors into our framework, as follows. Define schedule $\Omega(\gamma)$ to be the earliest start schedule when duration of activity i is set equal to $d_i'(\gamma) = d_i + \gamma m p_i E L_i$; as above, γ measures the degree in which expected values of disruptions are propagated throughout the network. The pre-schedule according to (LPH) is the output of the following linear programming model.

(LPH) $\min \quad \sum_{i \in N} c_i \Delta_i$ (15)

subject to

$s_i + d_i \leq s_j \qquad \forall\ (i,j) \in A$ (16)

$s_n \leq \omega$ (17)

$s_i(\Omega(\gamma)) = s_i + \Delta_i \qquad \forall\ i \in N$ (18)

all $\Delta_i, s_i \geq 0$

(LPH) is activity-based, rather than arc-based. The objective only considers weights c_i, while information about disturbance characteristics (p_i and L_i) is comprised in $\Omega(\gamma)$ based on expected values. Δ_i is the quantity by which the starting time of activity i in the generously protected schedule $\Omega(\gamma)$ exceeds the pre-schedule. Equality rather than 'no larger than' in constraints (18) ensures that activities start no later than they did in $\Omega(\gamma)$; if this were not guaranteed, LSS would always be optimal. When we choose γ:=0, the optimal objective value is 0 and the optimal schedule is ESS; more generally, we can perfectly mimic $\Omega(\gamma)$ as long as $\omega \geq f_n(\Omega(\gamma))$. (LPH) can be rearranged to be the dual of a MCNFP, as shown in the Appendix.

Let us illustrate (LPH) on our example project, with ω=10. We see that $d_i'(\gamma) = d_i + \gamma$, $\forall i \in N \setminus \{n\}$. The values $d_i'(\gamma)$ and $s_i(\Omega(\gamma))$ and the optimal primal solutions s_i are indicated in Table 1 for γ=0.5 and 1.0.

Table 1. (LPH) results for the example project.

i	0	1	2	3	4	5	6	7	8
d_i', γ=0.5	0.5	2.5	3.5	2.5	4.5	2.5	3.5	2.5	0
$s_i(\Omega(0.5))$	0	0.5	0.5	3	4	5.5	8.5	8.5	12
s_i	0	0.5	0	3	3	5.5	7	8	10
d_i', γ=1	1	3	4	3	5	3	4	3	0
$s_i(\Omega(1))$	0	1	1	4	5	7	10	10	14
s_i	0	1	0	4	3	7	7	8	10

For γ=0.5, the model introduces nonzero buffers $F_{01}=F_{13}=F_{35}=0.5$, $F_{47}=1$ and $F_{58}=2.5$: all activities either start at the corresponding starting time in $\Omega(\gamma)$ or are blocked by a successor from starting any later. The optimal objective value equals 5.5. We notice that in (LPH), choice of s_i larger than $s_i(\Omega(\gamma))$ is not penalized, but not encouraged either. Hence, if protection beyond fraction γn of the expected value of disturbances is possible, it is not guaranteed that this will be chosen by the model. Chain 1-3-5 for instance is scheduled as in $\Omega(0.5)$ with buffer sizes 0.5, but the single disturbance scenario has length 1 for activities 0,1 and 3. Choice of γ=1 yields better results in this case. We shall examine the performance of (LPH) as a function of γ in the computational experiments of Section 5.

4.3 *The activity-dependent float factor model*

Tavares et al. (1998) introduce the concept of a *float factor* as a means to synthesize the large number of decision variables a project manager has to cope with. The authors correctly state that "the adoption of the earliest (latest) starting time for each activity reduces (increases) the risk of an overall delay but increases (decreases) the project's discounted cost and therefore an optimal compromise has to be achieved". In this paper, we shall disregard project cost; on the other hand, we wish to avoid delay in the start time of any activity and not only n (which is meant by 'overall delay'). In this setting, the activities in the front of the project network shall be required to start as early as possible, such that their successors are appropriately protected, whereas the activities at the end of the project shall be started as late as possible, protecting them from disruptions in their predecessors. We only use the information that (WPF) also takes into account.

We see that for any schedule S, $s_i(\text{ESS}) \le s_i(S) \le s_i(\text{LSS})$, $\forall i \in N$, when $\omega \ge f_n(S)$. Tavares et al. (1998) prove that the choice $s_i(S) := s_i(\text{ESS}) + \alpha(s_i(\text{LSS}) - s_i(\text{ESS}))$, $\forall i \in N$, yields a feasible schedule S when $\alpha \in [0;1]$. α is called the float factor; a fraction α of $TF_i(\text{ESS})$ is added to the earliest possible start time $s_i(\text{ESS})$ of activity i. Nevertheless, we shall always choose $s_0 := 0$ en $s_n := \omega$, because of their interpretation as start and finish of the entire project and because given the evaluation criterion, this is always a dominant choice. In Section 5, a comparison is made of the evolution of expected weighted deviation as a function of α. For our purposes, however, we would like the float factor to vary for the different activities $i \in N$. We define quantities $\beta(i)$ and $\phi(i)$ associated with each activity $i \in N$, as follows: $\beta(i) := \sum_{(k,l) \in A: \atop l \in \sigma^T(i) \cup \{i\}} p_k c_l$, the sum of weights of all A-arcs that are before i in the network, and $\phi(i) := \sum_{(k,l) \in A: \atop k \in \sigma^T(i) \cup \{i\}} p_k c_l$, the sum of weights of the arcs in A that follow i. We see that $\beta(i) \le \beta(j)$ and $\phi(i) \ge \phi(j)$ if $(i,j) \in TA$. We can now define an *activity-dependent float factor* $\delta(i)$, $\forall i \in N$, as follows:

$$\delta(i) = \frac{\beta(i)}{\beta(i)+\phi(i)}.$$

Logically, $\delta(0)=0$ and $\delta(n)=1$. If $\beta(i)=\phi(i)=0$, we choose $\delta(i)=0.5$ (except for $i=0$ or n). Otherwise, we see that $\delta(i)=(1+\phi(i)/\beta(i))^{-1}$, such that $\delta(i)\leq\delta(j)$ if $(i,j)\in TA$, so the resulting schedule will be precedence feasible. As $s_i+d_i\leq s_j$, $\forall(i,j)\in TA$, and $d_i\in\mathbb{N}$, we also have $[s_i]+d_i \leq [s_j]$, where $[x]$ means the integer nearest to x, such that integer starting times can be easily found if desired. Application to our example project yields the results represented in Table 2.

Table 2. Application of the activity-dependent float factor.

i	$\beta(i)$	$\phi(i)$	$\delta(i)$	s_i(ESS)	s_i(LSS)	s_i	$[s_i]$
0	0	$10/n$	0	0	0	0	0
1	$1/n$	$3/n$	0.25	0	4	1	1
2	$1/n$	$5/n$	0.16667	0	0	0	0
3	$2/n$	$2/n$	0.5	2	6	4	4
4	$2/n$	$4/n$	0.33333	3	3	3	3
5	$3/n$	$1/n$	0.75	4	8	7	7
6	$3/n$	$1/n$	0.75	7	7	7	7
7	$3/n$	$1/n$	0.75	7	8	7.75	8
8	$10/n$	0	1	10	10	10	10

5. Computational results

To compare the performance of the models we developed in Section 3, a series of computational experiments using randomly generated test problems was conducted. The proposed models will be validated on a set of test instances generated by *RanGen* (Demeulemeester et al., 2000), a recently developed network generator for activity-on-the-node networks, which has the advantage of being able to generate so-called 'strongly random' networks. For various values of n, we generate a dataset of 300 problems, 100 instances for each of 3 values of the network parameter *order strength* (OS): OS=0.25, 0.5 and 0.75. Activity durations d_i are discrete uniform random variables between 1 and 25. Disturbance length L_i for activity i is a discrete random variable for which g_i is a discretized version of a continuous linearly decreasing function, for which the intercept with the abscissa is a uniform random variable I_i with support [2,25]. Scenarios $k\in E_i$ correspond with $l_{ik}=I_i-1$, I_i-6, ... until non-positive values are obtained. For each activity, we draw a random variable q_i from [1,3]; the values q_i, $i=0,\ldots,n-1$, are normalized to probabilities p_i. Cost coefficients c_i are integer variables selected from domains [1,3]. We impose due date $\omega=(1+\omega_0)s_n$(ESS), with $\omega_0\geq 0$. Unless otherwise specified, we choose $\omega_0=0.15$ and $n=31$.

For (EWD1), (EWD2), (WPF) and (LPH), we obtain an optimal solution by solving their dual formulated as a MCNFP. We use CS2‡ (Goldberg, 1997), version 3.9, a practical implementation of a scaling push-relabel method for solving MCNFPs, which is used as a subroutine. The code was slightly adapted to run under Windows and to guarantee convenient memory management in order to function as a subroutine, the algorithm itself was kept unchanged. The version of the code we obtained takes all integer inputs; as we select integer d_i, c_i and l_{ik}, fractional values

originate from probabilities p_i, pmf g_i and nonzero γ values. All models are input with primal right hand side (rhs) coefficients multiplied by factor 100 and rounded to the lower integer. (LPH) is handed to the solver with unchanged primal objective row, (WPF), (EWD1) and (EWD2) with primal objective row multiplied by factor $100n$ (although (EWD2) had perhaps better be corrected by a factor $\sim n^2$). Given this multiplication factor, the difference between the objective function obtained by (EWD1) and the result of simulation with 1 activity disturbed is 3.05% on average. This reduces to 0.29% for factor $1000n$, such that we can conjecture an inversely proportional relationship. An upper bound on arc flow needs to be specified for each arc in the MCNFPs; for uncapacitated arcs we follow Ahuja et al. (1993) by setting the capacity equal to B, where B is the sum of all arc capacities (all node supplies are zero), since in none of the models there is a negative cycle (in case of minimization objective) with infinite capacity. All programs have been implemented in C++, using the Microsoft Visual C++ 6.0 programming environment, on a Dell XPS B800r personal computer with Pentium III processor with 800Mhz clock speed and 128 Mb RAM.

Evaluation of a particular solution will take place in the following way: we select a prespecified number r of activities without replacement out of N, with probability of selection of activity i each time proportional to p_i. For each selected activity i, one disruption length l_{ik} will be selected by picking exactly one scenario k out of E_i; scenario k logically has probability $g_i(l_{ik})$ of being picked. Earliest start calculations can then be performed, in which each activity starts at the maximum of the finishing times of all its predecessors *and* its pre-scheduled starting time. The weighted deviation for one realization of the project can now easily be obtained. Per scheduling instance and corresponding schedule, we estimate expected weighted deviation by averaging the objective of 25,000 runs. In Section 5.1, the performance of the benchmark heuristics will be studied and compared with the output of the basic model (EWD1). We also study if this performance endures when the disruption lengths deviate from the discrete input scenarios. Section 5.2 will deal with the relative performance of models (EWD1) and (EWD2), and applicable model refinements. Section 5.3 will report on the CPU times utilized by the models, for different problem sizes.

5.1 *Performance of the heuristics*

We start our analysis by studying the performance of (LPH) as a function of γ. The graph below illustrates the evolution of the expected weighted deviation of the instances in the dataset when schedules are constructed according to (LPH). The abscissa represents γ, the ordinate corresponds with the difference in expected weighted deviation compared with case γ=1. r indicates the number of activities that undergo an increase in activity duration.

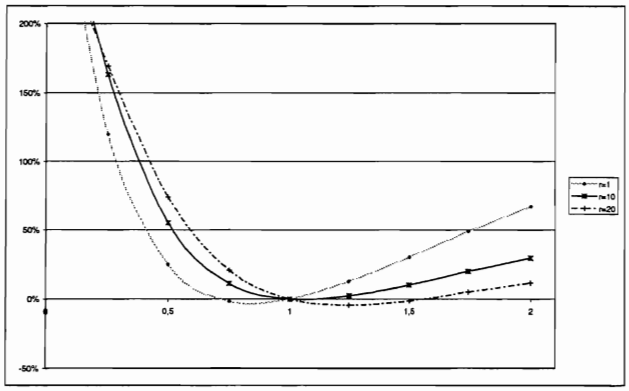

Figure 3. Performance of (LPH) as a function of γ (smoothed lines).
The curves correspond with different numbers (r) of disruptions in the simulation.

For γ=0, the values of expected weighted deviation are 350.72%, 334.38% and 285.10% for cases r=1, 10 and 20, respectively (the ordinate scale was limited such that these values are not represented). We can conclude that the choice γ=1.00 will never yield results that deviate much from the best attainable using model (LPH), independent of number of disruptions. In the remainder of this section, we will make use of this parameter value for all applications of (LPH).

Figure 4 illustrates the evolution of the expected weighted deviation of schedules that have been derived using the 'classical' float factor model (CFF) of Tavares et al. (1998). The independent variable is α, the input parameter of CFF, and the dependent variables are the expected weighted deviation values for various values of r. Boundary cases α=0 and α=1 correspond with schedules ESS and LSS, respectively. We see that the choice α=0.625 performs acceptably for all values of r and subsequently, we will opt for this parameter value in all applications of (CFF).

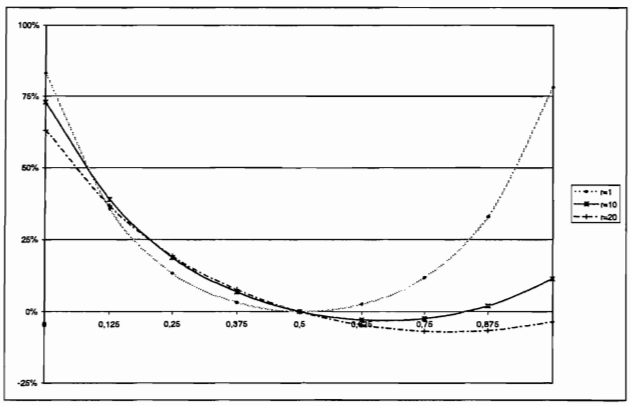

Figure 4. Performance of the (CFF) model as a function of α (smoothed lines).
The curves correspond with different numbers (r) of disruptions in the simulation.

We notice that for a large number of disrupted activities, the latest start schedule LSS performs about as good as intermediate α values and considerably

better than the earliest start schedule ESS, although the construction of the two schedules is symmetrical. This paradox can be attributed to the following fact: for LSS, set $\sigma^F(i)$ determines the value s_i whereas for ESS, $\pi^F(i)$ determines the baseline starting time of activity i. For a *given* baseline schedule, however, only the disruptions in $\pi^F(i)$ will determine the actually realized activity starting times S_i. Consider the following two subgraphs. A 'splitting node' has a large number of successors (and one or more predecessors), an 'assembly node' has a large number of predecessors (and one or more successors).

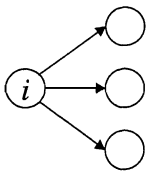

Figure 5a. i is a splitting node.

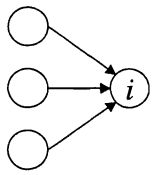

Figure 5b. i is an assembly node.

For a splitting node i, a disruption might affect the starting times of $\sigma(i)$, and afterwards may be propagated throughout the network; the effect of disruption is thus initially summed over the immediate successors of i. For ESS, the activities in $\sigma(i)$ are stuck with their starting times against the ending time of i, unless they have more critical predecessors, whereas in LSS the successors will most often start at different time instances. Thus, LSS will normally have more buffers in place in case of a splitting node.

For an assembly node i, disruption of one or more activities in $\pi(i)$ influences the starting time of activity i, which suffers a delay equal to the maximum of the disruptions minus the corresponding buffer sizes. In ESS most often, the activities in $\pi(i)$ will have different ending times and thus a number of buffers are automatically in place. For LSS on the other hand, unless the activities in $\pi(i)$ have other, more critical successors, they are stuck with their ending time to the start of i. In conclusion, ESS normally has more protection for assembly nodes then LSS.

In a project network, every arc gives rise to possible disruption propagation. An assembly node is better protected in ESS than in LSS, the successors of a splitting node are better protected in LSS than in ESS. The first case, however, only affect the starting time of one activity, whereas the second case concerns the disruption of the starting times of a large number of activities at once, thus having a much higher impact on the expected weighted deviation in the starting times. Or stated differently: ESS and LSS are symmetric in construction, but the way the disruptions are imposed on the schedules is more favourable to LSS than to ESS. In our opinion, this is the main reason for the awkward performance pattern observed in Figure 4. A bias in the shape of the generated networks may be a second cause, but we have no immediate reason to think such bias is present.

We shall now continue with our analysis of the performance of the benchmark heuristics. The main results have been collected in Figure 6. Performance is studied as the average deviation in objective function from model (EWD1) with γ=0. The expression 'ADFF' refers to the 'activity-dependent float factor' model.

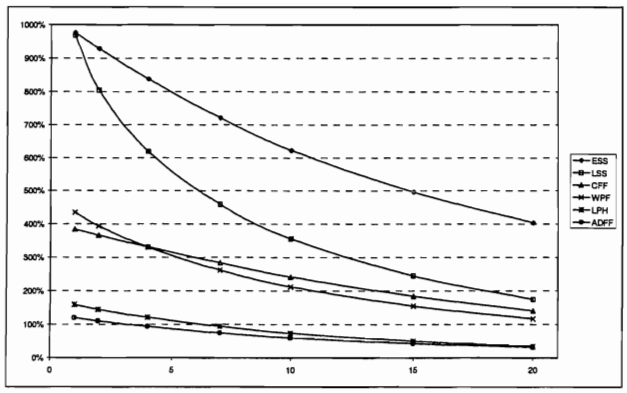

Figure 6. Estimated expected weighted deviation of the heuristic models as a function of the number of disruptions (r) (smoothed lines). Deviation is given as average percentage excess over (EWD1).

All curves are decreasing in r, which is evident given the logic underlying model (EWD1). The results obtained for ESS, LSS and the schedule constructed by (CFF) are in accordance with Figure 4. We observe that (CFF) and (WPF) have comparable average deviations from (EWD1), which evolve from some 400% (r=1) to about 120% for r=20. The best heuristics are (LPH) and (ADFF), for which the deviation from (EWD1) for r=1 still exceeds 100% (remember that (EWD1) is the optimal schedule for this case!). For a large number of disrupted activities (r=20), both models still have an expected weighted deviation in starting times that is 34.4% (for (LPH)) and 31.45% (for (ADFF)) beyond the values obtained by (EWD1), although this last model in principle only deals with a single activity that is expected to be disrupted.

The results pictured in Figure 6 are based on a simulation that draws the disruption lengths out of the discrete input scenarios, as explained at the beginning of this section. Often, the data that is available from past experience will indeed take the form of such a discrete rather than a continuous distribution. There is no guarantee however that new disruptions in the project under study will take on exactly the same values. We shall study the robustness of (EWD1) by sampling disruption lengths from the continuous function that was the basis for the selection of the input scenarios. This yields the following figure.

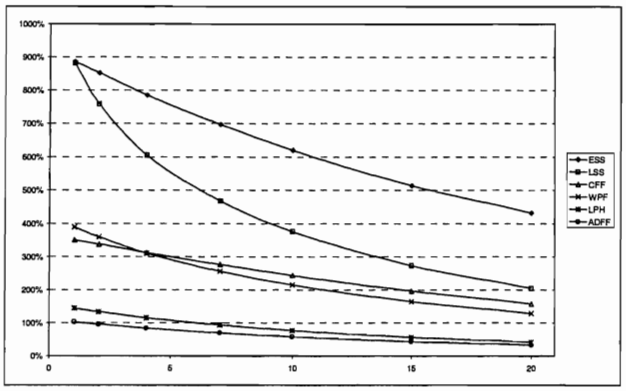

Figure 7. Estimated expected weighted deviation of the heuristic models as a function of the number of disruptions (r) (smoothed lines), when the disruption lengths are drawn from the original continuous distribution function. Deviation is given as average percentage excess over (EWD1).

For low r-values, the average excess is lower than in the case where the input scenarios where the basis for simulation; for higher values however, the average excess is at least as large. For r=20 for instance, the objective value equals 41.8% for (LPH) and 33.28% for (ADFF). The patterns are comparable with those in Figure 6, only are the curves somewhat flatter. This can be explained as follows: for a small number of disrupted activities, disturbances will often be 'stand-alone', without interaction with other disturbances. Therefore, if the exact disruption lengths are known, as is the case in model (EWD1), this can be exploited by inserting between two activities i and j exactly amount l_{ik}, for some $k \in E_i$, no more and no less. For high r values, even for 'discrete' simulation, the initial information contained in the l_{ik}-values is less useful.

Experimentation with discretisation of continuous distribution functions is possible. This seems to us of little practical value, however. As mentioned, the original data will most often be discrete, and the detour via a continuous function may only obscure the information contained therein. In other words, the actual probability distribution that applies during project execution is not known beforehand, and the discrete input scenarios form the best approximation available. This situation was envisaged by the simulation that resulted in Figure 7.

5.2 *Computational results for the models* (EWD1) *and* (EWD2)

It can be seen that in (EWD2), variable $f_{ijkz}^{[c]}$ often has a very low capacity, which will be a function of the primal objective row multiplication coefficient applied. We eliminate all $f_{ijkz}^{[c]}$ variables with 0 capacity beforehand, which allows us to limit the size of the model considerably, especially since we can immediately also drop the corresponding $w_{ijkz}^{[c]}$, $u_{ijkz}^{[c]}$ and q_{ijkz} variables. Without loss of feasible dual solutions, we can also impose on variables $w_{ijkz}^{[c]}$, $u_{ijkz}^{[c]}$ and q_{ijkz} the upper bound on flow that is associated with the corresponding variable $f_{ijkz}^{[c]}$, rather than B. The only uncapacitated arcs remaining are x_{ij} and v (as in (EWD1)). When n=31, the MCNFP corresponding with the dual of (EWD1) has exactly 32 nodes, and 903 arcs on average. (EWD2) has 9,091 nodes and 315,192 arcs on average, which can be reduced

to 5,979 nodes and 21,339 arcs by the procedure described above. This corresponds with an average reduction factor of 35% and 93%, respectively.

Let us refer to (EWD2) incorporating the arcs in $C_3(i,j)$ as (EWD2C3), as opposed to (EWD2C2). The following table gives the average percentage deviations of the basic model (EWD2C2) from (EWD2C3) (γ=0 for both models).

Table 3. Average percentage deviation of model (EWD2C2) from (EWD2C3).

	r=1	r=2	r=4	r=7	r=10	r=15	r=20
deviation	1.91%	1.80%	1.45%	0.99%	0.55%%	0.12%	–0.11%

We notice that (EWD2C3) achieves slightly better objective values (EWD2C2) for all numbers of disrupted activities (r). This is not unlogical: if we omit case $k_1 \in C_2(k_2,j)$ and use $k_2 \in C_3(k_1,j)$ to cover for this (identical) situation, we shall run through all scenarios of k_1 and approximate the scenarios of k_2 by replacing them by 1 (expected) value; this corresponds with (EWD2C3). (EWD2C2) on the other hand considers all scenarios of $k_1 \in C_2(k_2,j)$ explicitly. Now, remember that $C_3(k_1,j)=\pi^T(k_1)$, so on studying disruptions that affect j, (EWD2C3) uses exact information about disruptions nearer to j, whereas activities more distant from j are considered only for the expected value of their possible disruption length. Model (EWD2C2) is set up the other way round; this explains the difference in expected weighted deviation performance. As r increases, the value of exact information about possible disruption lengths decreases because accumulation will occur more frequently, a fact which can also be read from Table 3. Henceforward, (EWD2) refers to model (EWDC3).

We now examine the benefits of using nonzero γ values in models (EWD1) and (EWD2). The results are depicted in Figures 8a and 8b.

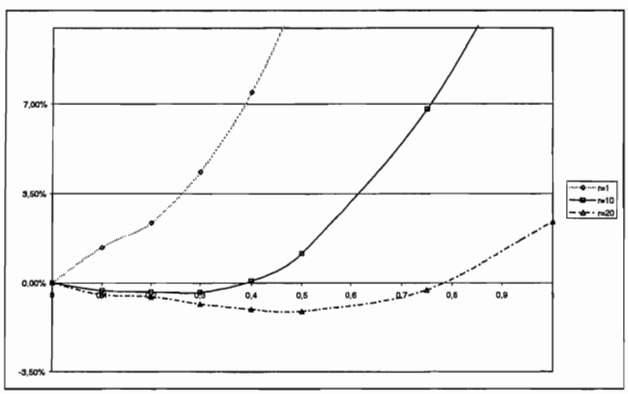

Figure 8a. Estimated expected weighted deviation of the model (EWD1) as a function of γ, for various numbers of disrupted activities (r) (smoothed lines). Deviation is given as average percentage excess over case γ=0.

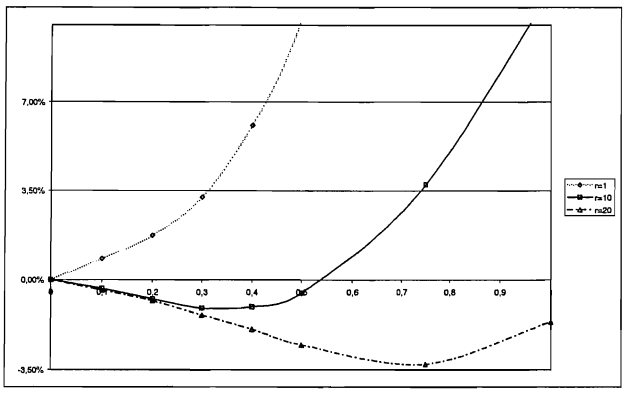

Figure 8b. Estimated expected weighted deviation of the model (EWD2) as a function of γ, for various numbers of disrupted activities (r) (smoothed lines). Deviation is given as average percentage excess over case γ=0.

From Figures 8a and 8b, we see that, especially for (EWD2), the expected weighted deviation can be considerably decreased by using nonzero γ-values: improvements of over 3% can be achieved for r=20, and over 1% for r=10. For (EWD1), up till –0.5% for r=10 and –1% for r=20 can be attained. The models run in virtually the same time as case γ=0 does, so this may be an easy way to obtain slight improvements in results. Nevertheless, this approach is very sensitive: if r turns out to be small, performance deteriorates quickly.

Figure 9 represents the evolution of expected weighted deviation averaged over all instances in the dataset, when expressed as percentage deviation from the objective obtained by model (EWD1) with γ=0. We conclude that as long as $r \leq 5$, (EWD1) with γ=0 (whose curve is the abscissa) cannot be improved upon. More generally, extra protection for a high number of disrupted activities goes at the expense of higher expected weighted deviations in start times for a low number of activities disrupted: (EWD2) with γ=0.6 for instance exceeds the base case by 17.70% on average when r=1, and outperforms the reference model by 6.41% when r=20. Improvements on (EWD1) for low r values are impossible since the model is optimal for r=1 (making abstraction of rounding errors). For larger values of r, (EWD2) appropriately counts in activity disruption interaction, and as explained above, the results are less sensitive to the exchange of the maximum and expectation operator: information about individual scenarios is less valuable. Although (EWD2) is intended to optimize the expected weighted deviation even for r=2, it is beat by (EWD1) until r=5 because of approximations in the model. It is to be noted that the CPU running times for (EWD1) (with arbitrary γ) (full lines) are significantly lower than those for (EWD2) (dotted lines). Computational effort will be further examined in Section 5.3.

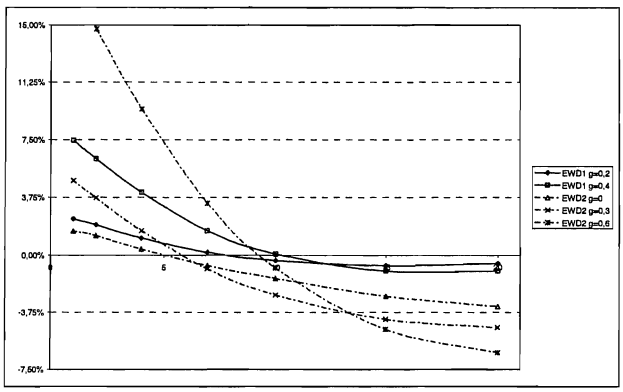

Figure 9. Estimated expected weighted deviation the models (EWD1) and (EWD2) as a function of r, for various values of γ (smoothed lines). Deviation is given as average percentage excess over model (EWD1) with γ=0.

5.3 *Computation times and results for larger datasets*

All important results regarding computational effort for varying problem sizes have been summarized in Table 4. For (EWD2), the number of nodes and arcs of the transformed dual in the table is the value obtained after reduction of nodes and arcs as discussed at the beginning of Section 5.2. CPU time for (ADFF) has not been included since there is no optimization involved in the construction of the associated schedule, and running times of the algorithm are negligeable for the values of n under consideration.

Table 4. Computation times and size of the dual for the different optimization models, as a function of n. Values are averaged over the dataset consisting of 300 problem instances. All CPU times are given in seconds.

n	CPU (EWD1)	nr arcs (EWD1)	CPU (EWD2)	nr nodes (EWD2)	nr arcs (EWD2)	CPU (WPF)	CPU (LPH)
31	0.00450	903	0.4052	5,979	21,339	0.00091	0.00072
41	0.00654	1,520	0.8711	12,074	42,764	0.00203	0.00131
51	0.00962	2,333	1.5668	20,345	71,976	0.00303	0.00161
61	0.01520	3,273	2.3927	30,172	106,804	0.00391	0.00237
71	0.02180	4,458	3.4792	42,166	149,586	0.00518	0.00317
81	0.03561	5,888	4.7027	55,393	197,303	0.00651	0.00395

We notice that, when n is increased by 10, all corresponding values in the table are increased approximately by factor 1.5. Model (EWD2) uses about 100 times the computation time required by (EWD1). The CPU time needed by (EWD1) equals about 5 times the time required by (WPF), which in turn exceeds the computer time for (LPH) but is of the same order of magnitude. These computational results are encouraging: as n increases, computational effort seems to grow exponentially, but the multiplication factor is not discouragingly high.

6. Conclusions and suggestions

This paper has examined various procedures for the development of a stable pre-schedule, which is unlikely to undergo major changes when it needs to be repaired as a reaction to activity duration disruptions. The main contribution of the paper are two mathematical programming models to minimize the expected weighted deviation in activity start times, when one or two activities are expected to incur an increase in their duration. Efficient solution of the models was possible by exploring the network flow nature of their duals. Additionally, some benchmark heuristics were discussed. The maximization of the weighted sum of pairwise floats between precedence-related activities revealed some links with the early papers in network flow theory. The linear programming based heuristic that was developed by Mehta and Uzsoy (1998) for job shop scheduling has been adapted to function in a project scheduling environment. Finally, we have discussed the float factor model proposed by Taveres et al. (1998), which has the project's discounted cost and the overall delay as its two main objectives. An adaptation of this model was presented that better fits the objective of this paper, the expected weighted deviation in activity start times.

Extensive computational results have been performed to analyze the performance of the heuristics and the basic optimization models. The obtained results are satisfactory: depending on the number of disruptions occurring during project execution, the expected weighted deviation (estimated by simulation) of the benchmark heuristics exceeds the values corresponding with our mathematical programming models by some 30% to 100%, and more. This indicates large opportunities for the application of the proposed models in both project and shop scheduling as well as in related areas as airline and railway scheduling, all of which domains in which stability and on-time performance of every activity separately are highly desirable, in the face of activity duration disruptions that will invariably occur.

The information we assumed available about activity disruptions was based on discrete scenarios. Relevant information based on passed experience will indeed most often take such form. We would like to make a case for the use of only the information that stems from observation of reality as a basis for schedule development, and to avoid the detour via the fitting continuous disruptions, the presence of which only trails multiple computational intricacies anyway. In Section 5.1, we showed that the models we propose are robust against variations in the disruption generation mechanism: the models maintain their performance relative to the benchmark heuristics, when we sample from a continuous distribution that is akin to the discrete input pmf.

The proposed models can be extended to include activities into the network that will only be executed with a certain probability: their baseline duration can be chosen as 0. In order to restrain the increase in computation time for growing problem sizes, it may be advantageous to limit the set TA of transitive precedence relations to only those pairs of activities that are not too far apart in the network, based on for instance a limit on the number of arcs on the shortest path from the first activity to the second. Another interesting topic for further research would be the extension of model (EWD2) to more accurately account for a larger number of anticipated activity disruptions.

References

Adlakha, V.G. and Kulkarni, V.G. (1989). A classified bibliography of research on stochastic PERT networks: 1966-1987. *INFOR*, **27**, 3, 272-296.

Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993). *Network flows. Theory, algorithms, and applications*. Prentice-Hall.

Akturk, M.S. and Gorgulu, E. (1999). Match-up scheduling under a machine breakdown. *European Journal of Operational Research*, **112**, 81-97.

Alagoz, O. and Azizoglu, M. (2001). Rescheduling under machine eligibility constraints. *Paper presented at the INFORMS 2001 Annual Meeting, Miami Beach, November 4-7.*

Bean, J.C., Birge, J.R., Mittenthal, J. and Noon, C.E. (1991). Matchup scheduling with multiple resources, release dates and disruptions. *Operations Research*, **39**(3), 470-483.

Bolat, A. (2000). Procedures for providing robust gate assignments for arriving aircrafts. *European Journal of Operational Research*, **120**, 63-80.

Bowers, J.A. (1995). Criticality in resource constrained networks. *Journal of the Operational Research Society*, **46**, 80-91.

Chapman, C. and Ward, S. (1997). *Project risk management: processes, techniques and insights*. Wiley.

Demeulemeester, E., Vanhoucke, M. and Herroelen, W. (2000). A new random network generator for activity-on-the-node networks. *Research report 0032, Department of Applied Economics, KU Leuven.*

Fulkerson, D.R. (1961). A network flow computation for project cost curves. *Management Science*, **7**, 167-178.

Goldberg, A.V. (1997). An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, **22**, 1-29.

Goldratt, E.M. (1997). *Critical Chain*. The North River Press.

Kelley, J.E. Jr. (1961). Critical-path planning and scheduling: mathematical basis. *Operations Research*, **9**, 296-320.

Kouvelis, P. and Yu, G. (1997). *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers.

Lawler, E.L. (1976). *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston.

Leon, V.J., Wu, S.D. and Storer, R.H. (1994). Robustness measures and robust scheduling for job shops. *IIE Transactions*, **26**, 5, 32-43.

Leus, R. and Herroelen, W. (2001). Models for robust resource allocation in project scheduling. *Research report 0128, Department of Applied Economics, KU Leuven.*

Levner, E.V. and Nemirovsky, A.S. (1994). A network flow algorithm for just-in-time scheduling. *European Journal of Operational Research*, **79**, 167-175.

Malcolm, D.G., Roseboom, J.H., Clark, C.E. and Fazar, W. (1959). Applications of a technique for research and development program evaluation. *Operations Research*, **7**(5), 646-669.

Mehta, S.V. and Uzsoy, R.M. (1998). Predictable Scheduling of a Job Shop Subject to Breakdowns. *IEEE Transactions on Robotics and Automation*, **14**, 3, 365-378.

Meredith, J.R. and Mantel, S.J. Jr. (2000). *Project management. A managerial approach. Fourth edition*. Wiley & Sons.

Neumann, K., Nübel, H. and Schwindt, C. (2000). Active and stable project scheduling. *Mathematical Methods of Operations Research*, **52**, 441-465.

Rosenhead, J., Elton, M. and Gupta, S.K. (1972). Robustness and Optimality as Criteria for Strategic Decisions. *Operations Research Quarterly*, 23, 4, 413-431.

Stork, F. (2001). Stochastic resource-constrained project scheduling. *Ph.D. Thesis, TU Berlin*.

Tavares, L.V., Ferreira, J.A.A. and Coelho, J.S. (1998). On the optimal management of project risk. *European Journal of Operational Research*, **107**, 451-469.

Valls, V., Laguna, M., Lino, P., Pérez, A. and Quintanilla, S. (1998). Project scheduling with stochastic activity interruptions. *In*: Weglarz, J. (ed.) *Project scheduling. Recent models, algorithms and applications*, Chapter 15, 333-353.

Wu, S.D., Storer, H.S. and Chang, P.-C. (1993). One-machine rescheduling heuristics with efficiency and stability as criteria. *Computers and Operations Research*, **20**(1), 1-14.

Wiest, J.D. and Levy, F.K. (1977). *A Management Guide to PERT/CPM*. Prentice-Hall.

Appendix

We can rearrange model (WPF) as follows:

$$\max \quad \sum_{(i,j)\in TA} c_j p_i F_{ij} \tag{A.1}$$

subject to

$$s_i - s_j + F_{ij} \le -d_i \qquad \forall\ (i,j) \in TA \tag{A.2}$$

$$s_n - s_0 \le \omega \tag{A.3}$$

all $F_{ij} \ge 0$, all s_i unrestricted in sign

The dual of this model is the following, associating nonnegative dual variables x_{ij} and v with constraints (A.2) and (A.3), respectively.

$$\min \quad -\sum_{(i,j)\in TA} d_i x_{ij} + \omega v \tag{A.4}$$

subject to

$$\sum_{(i,j)\in TA} x_{ij} - \sum_{(j,i)\in TA} x_{ji} = \begin{cases} 0 & \forall i \in N, i \neq 0, n \\ v & i = 0 \\ -v & i = n \end{cases} \tag{A.5}$$

$$x_{ij} \ge p_i c_j \qquad \forall\ (i,j) \in TA \tag{A.6}$$

This can be seen to be a MCNFP. If capacity constraints (A.6) were not present, zero flow would be optimal (ω is larger than or equal to the sum of d_i on any path from 0 to n). Therefore, we can impose an upper bound $\sum_{ij} p_i c_j$ on v and each x_{ij} on passing the model to a network solver.

Model (LPH) can be rewritten as follows:

$$\min \quad \sum_{i \in N \setminus \{0\}} c_i \Delta_i \tag{A.7}$$

subject to

$$s_j - s_i \geq d_i \qquad \forall\ (i,j) \in A \tag{A.8}$$

$$s_0 - s_n \geq -\omega \tag{A.9}$$

$$s_i - s_0 + \Delta_i \geq s_i(\Omega(\gamma)) \qquad \forall\ i \in N \setminus \{0\} \tag{A.10}$$

$$-s_i + s_0 \geq -s_i(\Omega(\gamma)) \qquad \forall\ i \in N \setminus \{0\} \tag{A.11}$$

all $\Delta_i \geq 0$; all s_i unrestricted in sign

(A.11) states that starting times cannot exceed those in $\Omega(\gamma)$; (A.10) is the direct link between the objective and the schedule. The functions of these two constraints were jointly taken care of by set of equations (17). Splitting the equality constraints (18) into $\leq$- and $\geq$-constraints would yield extra term $-\Delta_i$ in the left hand side of (A.11), but it can be intuitively seen that this extra term is not necessary, and in this way, the resulting formulation is the dual of the MCNFP which is given below. Nonnegative dual multipliers x_{ij}, v, y_i and z_i correspond with equations (A.8), (A.9), (A.10) and (A.11), respectively.

$$\max \quad \sum_{(i,j) \in A} d_i x_{ij} - \omega v + \sum_{i \in N \setminus \{0\}} s_i(\Omega(\gamma)) y_i - \sum_{i \in N \setminus \{0\}} s_i(\Omega(\gamma)) z_i \tag{A.12}$$

subject to

$$\sum_{(i,j) \in A} x_{ij} - \sum_{(j,i) \in A} x_{ji} - y_i + z_i = \begin{cases} 0 & \forall i \in N, i \neq 0, n \\ -v & i = n \end{cases} \tag{A.13}$$

$$\sum_{(0,j) \in A} x_{0j} + \sum_{j \in N \setminus \{0\}} y_j - \sum_{j \in N \setminus \{0\}} z_j = v \tag{A.14}$$

$$y_i \leq c_i \qquad \forall\ i \in N \setminus \{0\} \tag{A.15}$$

We see that the network of the MCNFP defined on the nodes in N contains the arcs in A, return arc $(n,0)$, and 2 extra arcs $(0,i)$ and $(i,0)$ per activity $i \in N \setminus \{0\}$. There are no positive profit cycles with infinite capacity: flow backwards in the network would use either v or z_i, flow forward on an infinite capacity path is only possible on x_{ij} arcs. Clearly, any path the sum of d_i values on any $P(0,n)$ is not larger than ω, and the sum of d_i values on any $P(0,k)$ is not larger than $s_k(\Omega(\gamma))$, for any activity $k \neq 0$.