A network flow algorithm for just-in-time project scheduling

TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculteit Wiskunde en Informatica

Memorandum COSOR 91-21

A NETWORK FLOW ALGORITHM FOR JUST-IN-TIME PROJECT SCHEDULING

E.V. Levner
A.S. Nemirovsky

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB Eindhoven
The Netherlands

ISSN: 0926-4493

Eindhoven, September 1991
The Netherlands

A NETWORK FLOW ALGORITHM FOR JUST-IN-TIME PROJECT SCHEDULING[1]

E.V. Levner

Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven
The Netherlands

A.S. Nemirovsky

Central Economical-Mathematical Institute
of the Academy of Sciences of the USSR
Krasikova st. 32
117418 Moscow
The USSR

[1] This report is a revised version of our two talks presented at the International School-Seminar on Optimization Methods and Applications (Baikal Lake, USSR, 1989), organized by the International Institute for Applied Systems Analysis, Laxenburg, Austria, and the Second International Workshop on Project Management and Scheduling, Universite de Technologie de Compiegne, France, 1990.

A NETWORK FLOW ALGORITHM
FOR JUST-IN-TIME PROJECT SCHEDULING

E.V. Levner
Eindhoven University of Technology, Eindhoven, Netherlands

A.S. Nemirovsky
Central Economical-Mathematical Institute, Moscow, USSR

Abstract

We show the polynomial solvability of the PERT-COST project scheduling problem in the case of: (i) the objective being a piecewise-linear, convex (possibly, non- monotone) function of the job durations as well as of job start/finish times, and (ii) the precedence relations between jobs being presented in the form of a general (not necessary, acyclic) directed graph with arc lengths of any sign. For the latter problem, we present a network flow algorithm (of pseudo-linear complexity) which is easy to implement and which behaves well when the objective values grow slowly with the growth of the problem size while the number of breakpoints in the objective grows fast.

1. Introduction

This work contributes to the mathematical analysis of a PERT-COST project scheduling problem with the "just-in-time"-oriented objective/constraints (JIT-PSP). The latter problem can be used as an auxiliary bounding problem within the branch-and-bound and Lagrangian relaxation computational schemes for solving more general resource-constrained project scheduling problems (see, e.g., [3], [12]) and, besides, it is of interest on its own, being a mathematical model of "just-in-time" scheduling, very important for contemporary flexible manufacturing.

Several studies have been reported in the literature related to some restricted cases of our project scheduling problem.

In their pioneering papers, Kelley [8] and Fulkerson [6] have employed linear programming duality for solving the linear project scheduling problem (PSP) in the (acyclic) PERT network and found out the following fundamental fact: the linear PSP is dual to the problem of finding the minimum cost flow in a network. Kelley [8] has also showed how to reduce the piecewise-linear PSP to the linear one by adding into the original network as many new arcs as the total amount of linear pieces in all piecewise-linear components of the objective function.

Adelson-Velsky, Voropaev and Kalinovska [1], and Kerbosch and Schell [9] have treated the so-called generalized PERT networks (with negative arc lengths and negative cycles), however they considered only temporal network characteristics, not touching upon the costs. Levner [10] has extended the latter approach by considering the case of the cost objectve linearly depending both on the job durations as well as on the job start/finish times, and has shown that the problem of finding the optimal project cost of this type in the generalized PERT nerworks is reducible (in polynomial time) to a min-cost network flow problem.

Elmaghraby and Pulat [4] (see, also, [12]) have extended the Kelley-Fulkerson project scheduling model introducing deadlines for certain 'milestone' events and developing an adequate modification of the Kelley-Fulkerson network flow algorithm. In [11], we have given a general description of the problem and algorithm to be considered in more detail in the present paper. In [5], Foldes and Soumis have described another, Lagrangian, approach to solving a similar network scheduling problem, in their model the PERT networks being acyclic and the convex objective functions depending only on the job durations.

The main result of this paper is to show the existence of an efficient network flow algorithm for solving the PERT-COST scheduling problem in the case if: *(i)* the objective is a piecewise-linear, convex (possibly, non-monotone) function of job durations as well as of job start/finish times, and *(ii)* the precedence relations between jobs are presented in the form of a general (not necessarily, acyclic) directed graph with arc lengths of any sign.

We start, in Section 2, by presenting a piecewise-linear programming formulation of our scheduling problem. Next, in Section 3, we show its polynomial solvability by reducing it to a minimum cost flow problem (in an extended network with cycles and with the cost coefficients of any sign). Finally, in Section 4, we present a network flow algorithm which does not require the excessive growth of the network size and escape calculations of flows in negative cycles. Each its iteration consists in finding maximal flow in a network of the same size as the original one; this algorithm has pseudo-polynomial (in fact, pseudo-linear) complexity. In Appendix we give the proof of the validity of the algorithm.

2. Problem formulation

The project scheduling problem can be formulated as follows:

There is a set **T** of jobs ("tasks", "activities") constituting a project, and there is a set E of events (starting and finishing points of various jobs of the project).

The jobs of the project are interrelated through precedence relations of two types, NOT BEFORE and NOT LATER. Let $E=\{1,\dots,n\}$ be a given set of events, and t_i the (unknown) time at which the event i occurs. The NOT BEFORE relations denote that some job can be started (and/or, equally possible, can be finished) *not before* than in $a(i,j)$, the given amount of time units, after some other job has started (and/or finished): $a(i,j)\le t_j-t_i$, $(i,j)\in V\subset E\times E$, V being the given set of ordered pairs of related events.

The NOT LATER relations denote that some job is to be started (and/or, equally possible, to be finished) *not later* than the given time $b(i,j)$ before some other job will start (finish):

$$t_j-t_i\le b(i,j),\quad (i,j)\in V\subset E\times E,$$

where $-\infty\le a(i,j)\le b(i,j)\le+\infty$ for all $(i,j)\in V$. Evidently, $a(i,j)=-\infty$ (and, respectively, $b(i,j)=+\infty$) means that there is no corresponding NOT BEFORE (NOT LATER) relation between the events i and j.

For each job $T\in\mathbf{T}$, there are:

- release times $r(\cdot)$ and deadlines $d(\cdot)$ for its start/finish,
- limits $a(\cdot)$ and $b(\cdot)$ on its time duration.

The objective function of the problem is the project cost which is assumed to involve components of two types: a piecewise-linear convex cost functions $\phi_T(t(T))$ depending on the (unknown) job duration $t(T)$ (and representing penalties incurred for the project jobs implemented too slowly as well as too hasty), and, also, a piecewise-linear convex cost functions $\phi_i(t_i)$ each depending on the (unknown) occurence time of the event i, and representing penalties incurred for the project jobs started (finished) too late as well as too early.

Let us construct the PERT network N corresponding to our problem: its nodes are identified with events $i\in E=\{1,\dots,n\}$, and its arcs are identified with the ordered pairs of events $(i,j)\in V$ appearing above, the numbers $a(i,j)$ and $b(i,j)$ being assigned to the arcs (it is clear that some arcs of N depict project jobs while some other are dummy).

Observe that if V contains both a pair (i,j) and its "inverse", (j,i) then one of these pairs (in fact, any one of them) is superfluous.

Indeed, a pair of inequalities

$$a(i,j)\le t_j - t_i \le b(i,j),$$

$$a(j,i)\le t_i - t_j \le b(j,i)$$

is equivalent to an inequality

$$a'(i,j)\equiv\max\{a(i,j),-b(j,i)\}\le t_j - t_i \le b'(i,j)\equiv\min\{b(i,j),-a(j,i)\}.$$

Thus, we can delete the arc (j,i) and modify the bounds (namely, $a(i,j):=a'(i,j)$, $b(i,j):=b'(i,j)$) and the cost function ($\phi_{ij}(\tau):=(\phi_{ij}(\tau)+\phi_{ji}(-\tau))$, assigned to the remaining arc (i,j), the latter guaranteeing the equivalence between the initial and new problems.

Thus, from now on we assume that

$$(i,j)\in V \Rightarrow (j,i)\in V.$$

The problem, which we call the "just-in-time" version of a project scheduling problem, JIT-PSP, is: by the choice of job durations $t(i,j)$ and event occurence times t_i, to find the minimal total cost of the project subject to the precedence and arrival/deadline constraints:

$$\text{minimize} \quad \sum_{(i,j)\in \mathbf{T}} \phi_{ij}(t(i,j)) + \sum_{i\in E} \phi_i(t_i)$$

$$\text{subject to}$$

$$r_i \le t_i \le d_i, \quad i\in E,$$

$$t(i,j)+t_i-t_j\le 0, \quad (i,j)\in V\subset E\times E,$$

$$a(i,j)\le t(i,j)\le b(i,j), \quad (i,j)\in V\subset E\times E.$$

It is quite clear that any variable $t(i,j)$ in the latter formulation may be substituted by a difference of variables of the form $t_u - t_i$, where u is a node added into arc (i,j), and after that our piecewise-linear programming problem may be rewritten in terms of variables t_i only (the proof is given in Appendix 1):

$$\text{minimize} \quad \phi(t_1,\dots,t_n) = \sum_{(i,j)\in V} \phi_{ij}(t_j - t_i) \tag{1}$$

subject to

$$a(i,j)\le t_j - t_i \le b(i,j), \quad (i,j)\in V\subset E\times E, \tag{2}$$

where associated with each pair $(i,j)\in V$ is its *cost function* $\phi_{ij}(t_j-t_i)$, $\phi_{ij}(\tau)$ being the given convex, piecewise-linear (not necessarily monotone) function on the real axis.

Let us denote by Γ the directed graph (E,V) with the set of nodes E and the set of arcs V, and let $V'=\{(j,i)\,|\,(i,j)\in V\}$ be the set of arcs "inverse" to the arcs of Γ.

Let $V^*=V\cup V'$ and Γ^* be the directed graph with the set of nodes E and the set of arcs V^*. Let us define $a(i,j)$ and ϕ_{ij} for $(i,j)\in V'$ as follows: $a(i,j)=-b(j,i)$, $\phi_{ij}(\tau)\equiv 0$.

Now $a(i,j)$ and ϕ_{ij} are defined for all $(i,j)\in V^*$.

JIT-PSP (1)-(2) is obviously equivalent to the following problem:

$$\text{minimize} \quad \phi(t_1,\dots,t_n)= \sum_{(i,j)\in V^*} \phi_{ij}(t_j-t_i) \qquad (3)$$

subject to

$$a(i,j)+t_i-t_j\le 0, \quad (i,j)\in V^*. \qquad (4)$$

Let us characterize now the difference between our JIT-PSP model and the conventional PSP.

First, while traditional PSP models usually work with acyclic PERT networks (see, e.g., [4]-[6], [8]), our JIT-PSP model focuses on the generalized PERT networks permitting of negative arc lengths and cycles (of non-positive length). The latter gives us the possibility to include into the model: (*i*) the "just-in-time"-oriented constraints of the form $t_i\le d_i$, $i\in E$, and $|t_j-t_i|\le b(i,j)$, $(i,j)\in V$, and, also, (*ii*) the job-overlapping constraints.

Second, while in the traditional PSP the job cost is a function of the job duration only, in JIT-PSP the cost function (1) is "just-in-time"-oriented, i.e., it embraces, also, convex penalties incurred both in the case if the activity begins too late/too early just as in the case if it ends too late/too early. Formally, in JIT-PSP each term in (1) depends on the time shift, t_j-t_i, between events i and j, whereas in the Kelley-Fulkerson PSP model each cost term depends on the duration of the activity, $t(i,j)$, which *may be less* than the corresponding time shift, the latter fact not permitting us to express the t_i variables in terms of job durations $t(i,j)$.

3. JIT-PSP and network flows

Now we show that even in the presence: *(i)* of the (non-negative) cycles in the PERT networks, and *(ii)* of the terms in the project cost (3) depending not only on the job durations but also on the event ocurrence times, our scheduling problem (3)-(4) can be reduced to a minimum cost network flow problem. The reduction is done just in the same way as it has been performed by Kelley [8] (see, also, [13]) for the restricted case of the acyclic network and the objective depending only on the job durations.

Let us fix $(i,j)\in V$. Let the piecewise-linear function $\phi(\tau)\equiv \phi_{ij}(\tau)$ be equal to $\beta_r+\gamma_r\tau$ for $T_{r-1}\le\tau\le T_r$, $1\le r\le R$, where $a(i,j)\equiv T_0<T_1<\ldots<T_R\equiv b(i,j)$ (all the parameters depending on (i,j)). Let us replace the arc (i,j) of the PERT network N by a set of sequential arcs $(u_0\equiv i,u_1)$, $(u_1,u_2),\ldots,(u_{R-1},u_R\equiv j)$, with the event occurance times subjected to the constraints

$$a(i,j)\le t_{u_1}-t_{u_0}\le T_1,$$

$$0\le t_{u_r}-t_{u_{r-1}} \le T_r-T_{r-1},\quad 2\le r\le R.$$

The corresponding cost functions are defined as follows:

$$\phi_{u_0,u_1}(\tau)=\beta_1+\gamma_1\tau,$$

$$\phi_{u_{r-1},u_r}(\tau)=\gamma_r\tau,\quad 2\le r\le R.$$

Observe that the linear JIT-PSP on the extended network obtained in the described above manner is equivalent to the initial JIT-PSP, the equivalence following from the convexity of the initial cost functions.

Now let us consider a linear JIT-PSP, i.e. the problem of minimizing the linear cost function

$$\sum_{(i,j)\in V^*} c_{ij}(t_j-t_i) \quad \text{(where } c_{ij}\equiv \frac{d}{d\tau}\phi_{ij}(\tau))$$

subject to the constraints

$$a(i,j)+t_i-t_j\leq 0,\quad (i,j)\in V^*.$$

V^* denoting (only here) the arcs of the extended network.

Clearly, the problem P, dual to this one, is formulated as follows:

$$\textit{minimize}\quad \sum_{(i,j)\in V^*} -a_{ij}\,\lambda_{ij}$$

by choosing nonnegative λ_{ij} *subject to the constraints*

$$\sum_{i:(i,j)\in V^*}\lambda_{ij} - \sum_{i:(j,i)\in V^*}\lambda_{ij} = \sum_{i:(i,j)\in V^*} c_{ij} - \sum_{i:(j,i)\in V^*} c_{ij}\,\rho_j,\quad j\in E$$

(notice that $\sum_j \rho_j=0$).

We see that P is a minimum cost network flow problem (with infinite capacities of the arcs). The coefficients $-a(i,j)$ can be negative as well as positive (observe that in the Kelley-Fulkerson formulation of PSP these coefficients are nonnegative).

Thus, one evident way to solve efficiently JIT-PSP is to reformulate it as a linear JIT-PSP on an extended network and then apply to its dual a flow network algorithm which is able to cope with cycles and negative cost coefficients. Such algorithms (for example, the so called "Out-of-Kilter Method") have been presented, e.g., in [4], [6], [8] (see, also, surveys [2], [7]). These network flow algorithms are not discussed in this paper.

On the other hand, in some practical applications, when the extended network turns out to be inadmissibly large, it is desirable to solve JIT-PSP without extending the original network. In the next section, we present the corresponding algorithm which reduces the solution of JIT-PSP to a series of calculations of maximal flow in a network of the same size as the original (not-extended) network. The computational complexity of the algorithm is $O(R*c)$, where c is a (polynomial) complexity of finding the maximal network flow (see, e.g., [2], [7] for further datails), and $R=\phi(t^*)-\phi(t^{**})$, t^* being some initial problem's solution, t^{**} the optimal one.

4. Algorithm

4.1. Obtaining a feasible solution

For the sake of simplicity from now on we assume that our JIT-PSP satisfies the following conditions, not too restrictive from the practical viewpoint:

(A) There exists an *initial node* $i_{\star}\in E$ such that for each other node $k\in E$ there exist two directed paths γ and γ' in $\Gamma^{\star}$, the first one going from $i_{\star}$ to k and the second going from k to $i_{\star}$ such that $a(i,j)>-\infty$ for all arcs $(i,j)\in\gamma\cup\gamma'$.

It is clear that Assumption (A) together with (2) implies the existence of a constant L such that

$$|t_i-t_{i_\star}|\le L \qquad (5)$$

for each feasible solution $\{t_i\}$ of (1)-(2).

(B) All the quantities $a(i,j)$ and $b(i,j)$ in (1)-(2) are integers (or $\pm\infty$); all the cost functions ϕ_{ij} are convex piecewise-linear functions taking integer values at integer points, their breakpoints being also integer.

Notice that the integrality condition (B) is important for providing the pseudo-polynomiality property of the algorithm to be presented below.

(C) For all $(i,j)\in V$, $a(i,j)<b(i,j)$ (see (2)).

This assumption, in fact, does not lead to the loss of generality, since in the case of $a(i,j)=b(i,j)$ for some $(i,j)\in V$, or, equivalently, $t_j-t_i=a(i,j)$, we can eliminate the variable t_j, by substituting $t_j=t_i+a(i,j)$, without breaking the problem structure.

The following Proposition is valid:

Proposition. JIT-PSP (3)-(4) is consistent iff the corresponding graph $\Gamma^{\star}=(E,\ V^{\star}\equiv V\cup V')$, with arc lengths being defined as

$$a^{\star}(i,j)=\begin{cases} a(i,j), \text{ if } (i,j)\in V, \\ -\,b(j,i), \text{ if } (i,j)\in V', \end{cases}$$

has no cycles of positive total length. In the case of consistency the set of numbers $\{t^{\star}_{i_\star}=0;\ t^{\star}_i\equiv$ *maximal length of paths in* $\Gamma^{\star}$ *starting at* $i_{\star}$ *and finishing at* i, $i\ne i_{\star}\}$ is a feasible solution to (3)-(4). A consistent JIT-PSP is solvable.

Proof. The set of inequalities (2) can be rewritten as $t_j-t_i\geq a^*(i,j)$, $(i,j)\in V^*$; so in the case of consistency each cycle in Γ^* has a nonpositive length. Conversely, if all the cycles in Γ^* are of nonpositive lengths, then the above numbers t_i^* are well defined (since in this situation the lengths of paths linking i_* and $i\neq i_*$ are bounded from above and not all of these lengths are equal to $-\infty$due to (A)). It is clear that t_i^*, by virtue of their origin, satisfy the relations $t_j^*-t_i^*\geq a^*(i,j)$, $(i,j)\in V^*$, so $\{t_i^*\}$ is a feasible solution.

It remains to verify that the consistent JIT-PSP is solvable. Indeed, $\{t_i\}$ is feasible to the problem iff $\{t_i'\equiv t_i-t_{i_*}\}$ is feasible, and both of the feasible solutions are of the same cost. So the additional restriction $t_{i_*}=0$ does not change objective's optimal value in (3)-(4); but under this restriction the set of feasible solutions, due to (5), is a compact (which is nonempty when the problem is consistent). So the solvability of the consistent JIT-PSP, satisfying (A), is obvious.

Proposition shows that, by using the standard techniques of graph theory, one can either find an integer feasible solution $\{t_i^*\}$ to our JIT-PSP problem or find out that the problem is inconsistent, performing this in no more than $O(nl)$ operations, n being the number of nodes and l being the number of arcs in Γ^*.

4.2. Constructing the flow network

Each iteration of the algorithm to be described below, consists of finding the maximum flow in a flow network $N^*(t)$ depending on the feasible solution $\{t\}$.

Let us describe now how we construct the auxiliary network $N(t)$ and then extend it to the network $N^*(t)$.

The node set of the network $N(t)$ is E.

The arc set, $V(t)$, of $N(t)$ will be constructed from V and for each arc $(i,j)\in V$, $V(t)$ will contain one and only one of two arcs $\{(i,j),(j,i)\}$. In order to define $V(t)$, we need to observe that for the given $(i,j)\in V$ and our $\{t_i\}$, one and only one case, A, B or C, is possible:

either $a(i,j)<t_j-t_i<b(i,j)$ (case $A(i,j)$),
or $a(i,j)=t_j-t_i<b(i,j)$ (case $B(i,j)$),
or $a(i,j)<t_j-t_i=b(i,j)$ (case $C(i,j)$).

These cases are disjoint since $a(i,j)<b(i,j)$, due to Assumption (C).

Now we can define the arcs in $V(t)$ and their capacities in the following way. In the case $A(i,j)$ we include (i,j) into $V(t)$ and assign to it the capacity $d(i,j)=\phi'(t_j-t_i+0)-\phi'(t_j-t_i-0)$, where $f'(t+0)$ denotes the right, and $f'(\tau-0)$ - the left derivative of a piecewise linear function f at a point τ. In the case $B(i,j)$ (respectively, $C(i,j)$) we include into $V(t)$ the arc (j,i) (respectively, (i,j)) and assign to it the capacity $d(\cdot,\cdot)=+\infty$. Notice that the capacities are nonnegative due to the convexity of $\phi_{\cdot\cdot}$.

It remains to describe the sources and the sinks in the flow network $N(t)$ we are constructing. Let for $(i,j)\in V(t)$ the quantities $\rho(i,j)$ be defined as follows:

$$\rho(i,j) = \begin{cases} \phi'_{ij}(t_j-t_i-0), \text{ in the cases } A(i,j) \text{ and } C(i,j), \\ -\phi'_{ji}(t_i-t_j+0), \text{ in the case } B(j,i). \end{cases} \qquad (6)$$

Let us for $k\in E$ define

$$\sigma(k) = \sum_{i:(i,k)\in V(t)} \rho(i,k) - \sum_{i:(k,i)\in V(t)} \rho(k,i). \qquad (7)$$

The nodes k with $\sigma(k)>0$ (respectively, $\sigma(k)<0$) by the construction, be the *sources* (respectively, the *sinks*) of our flow network; the value of a source/sink k is, by definition, $|\sigma(k)|$. The network we have produced is the desired auxiliary network $N(t)$.

Let us extend the network $N(t)$ by two nodes, s (a "supersource") and f (a "supersink") and by arcs from s to each of the sources k (of capacities $d(s,k) = \sigma(k)$) and by arcs from each of the sinks k to f of capacities $d(k,f) = |\sigma(k)|$. Let E^+ and $V^+(t)$ denote the set of nodes (arcs) of the extended network, $N^*(t)$.

4.3 Description of the algorithm

Our algorithm works as follows. At the beginning of each iteration of the algorithm there is an integer feasible solution to (3)-(4), $t=\{t_i\}$. At the iteration either it is established that this solution is optimal, or the solution is transformed into another integer feasible solution, $t'=\{t'_i\}$, such that

$$\phi(t') \le \phi(t)-1. \qquad (8)$$

The current iteration of the algorithm consists of solving the following Max-Flow Problem $P(t)$ in the network $N^*(t)$:

maximize the flow value

$$\pi(\lambda) \equiv \sum_{i:(s,i)\in V^+(t)} \lambda(s,i) \tag{9}$$

subject to

$$0\le\lambda(i,j)\le d(i,j), \quad (i,j)\in V^+(t), \tag{10}$$

and

$$\mathrm{div}_k\lambda\equiv \sum_{i:(i,k)\in V^+(t)} \lambda(i,k) - \sum_{i:(k,i)\in V^+(t)} \lambda(k,i) = 0, \quad k\in E \tag{11}$$

Notice that, by (B) and the integrality of *t*, the values of sources and sinks in the network $N^*(t)$ are integers, and the capacities of all arcs are integers or ±∞.

Let us solve $P(t)$ (there is a lot of appropriate alorithms solving the problem with the complexity polynomial in the size of the flow network, see, e.g. [2], [7]), and let λ^* be the corresponding maximal flow and π^* be its value. Since the capacities in $P(t)$ are integer, the quantities $\lambda^*(\cdot,\cdot)$ can be chosen to be integral too (this is the well-known property of the Max Flow Problem).

It is clear that

$$\pi^*\equiv \sum_{i:(s,i)\in V^+(t)} \lambda(s,i) = \sum_{i:(i,f)\in V^+(t)} \lambda^*(i,f) \le$$

$$\le\pi_t \equiv \sum_{k\in E,\ k \text{ is a source}} \sigma(k) = \sum_{k\in E,\ k \text{ is a sink}} \sigma(k) \tag{12}$$

(where the first equality holds by virtue of (11), the second by virtue of $\sum_{k\in E} \sigma(k)=0$, the latter being a corollary of (7)).

If the inequality in (12) turns out to be an equality, we claim that *t* is an optimal solution to our JIT-PSP and terminate.

Otherwise we find the minimal cut in $N^*(t)$, (E^s,E^f), corresponding to the maximal flow λ^* (recall that E^s consists of *s* and of all the nodes linked with *s* by paths with arcs being not saturated by the flow λ^*, while $E^f=E^+\backslash E^s$). It turns out that in the case under consideration (i.e. in the case of $\pi^*<\pi_t$), both of the sets $E^s\cap E\equiv E_s$ and $E^f\cap E\equiv E_f$ are nonempty. Then we construct the new solution, *t′*, in the following way:

$$t'_i = \begin{cases} t_i, & \text{if } i\in E_s, \\ t_i + 1, & \text{if } i\in E_f. \end{cases} \tag{13}$$

Under Assumptions (A)-(C), the above procedure, being applied to an arbitrary integer feasible solution to (3)-(4), *t,* has the following properties:

Theorem 1. In the case of $\pi^{*}=\pi_{t}$, the vector *t* is an optimal solution to (3)-(4).

Theorem 2. In the case of $\pi^{*}<\pi_{t}$, *t′* produced by (13) is well defined and it is the integer feasible solution to (3)-(4) such that (8) holds.

These theorems justifying the above algorithm are proved in Appendix 2.

Corollary. Under Assumptions (A)-(C), JIT-PSP (3)-(4) is either inconsistent, or possesses an integer optimal solution t^{**}. One can find out, in no more than *O(nl)* running time, which of the cases takes place by seeking for the feasible solution t^{*} described in Proposition, and then one can obtain the optimal t^{**} (if it exists) in no more than $R\equiv\phi(t^{*})-\phi(t^{**})$ iterations of the above algorithm, each consisting in solving the max-flow problem $P(t)$.

5. Concluding remarks

The algorithm presented in the paper is "pseudo-polynomial", (in fact, pseudo-linear) though not polynomial (recall that this means that the complexity of the algorithm is bounded by a polynomial in the length of input and in the magnitude of the residual R though not in the length of problem′s input only).

Apparently, this complexity bound for JIT-PSP is theoretically improvable. Indeed, JIT-PSP can be reduced to the minimum cost network flow problem, and, the latter problem admits polynomial and even *strongly polynomial* algorithms (the latter term means that the complexity of the algorithm is polynomial in the problem size, *(n,l);* see [2],[7] for further details).

However, if compared with the Ednmonds-Karp-Dinic scaling technique as well as with the strongly polynomial Tardos method, our algorithm has evident, practically important, advantages: it is very easy to implement and it behaves good if the objective values grow slowly with the growth of the problem size, while the number of breakpoints in (3) grows fast. More thorough experimental study of this algorithm seems to be rather useful; moreover, apparently, it can be transformed into the polynomial one (using techniques described, e. g., in [2], [7]), however, these questions fall out of limits of this paper and may be considered to be possible directions for further research.

Appendix 1

In this appendix we prove the reducibility of the conventional project scheduling problem, PSP, to JIT-PSP. Recall the formulation of PSP as presented in [6] and [8]:

By the choice of durations of activities $\tau(i,j)$ *and event occurence times* $\tau(i)$, *to minimize*

$$\sum_{(i,j)\in A} c(i,j)\ (B(i,j)\ -\ \tau(i,j))$$

subject to

$$\tau(f)-\tau(s)\le T,$$

$$\tau(i,j)+\tau(i)-\tau(j)\le 0,\quad (i,j)\in A,$$

$$A(i,j)\le\tau(i,j)\le B(i,j),\quad (i,j)\in A,$$

$A(i,j)$, $B(i,j)$, $c(i,j)$, T being the given integers, A being the set of arcs in the corresponding PERT network representing the project, s/f being the initial/final event of the project.

In order to reformulate the latter problem as the JIT-PSP, we, first, add into each arc $(i,j)\in A$ a new node, u, so replacing $(i,j)\in A$ by two sequential arcs (i,u) and (u,j). Then, by substituting (for each $(i,j)\in A$) $t_i=\tau(i)$, $t_j=\tau(j)$, $t_u=\tau(i)+\tau(i,j)$, we can reduce the unequalities in the PSP corresponding to a pair $(i,j)\in A$ to the JIT-PSP form: $a(i,u)\le t_u-t_i\le b(i,u)$, $0\le t_j-t_u$, where $a(i,u)=A(i,j)$, $b(i,u)=B(i,j)$.

As for the inequality $\tau(f)-\tau(s)\le T$, by including the arc (s,f) into the network representing the project, we can make it to become a JIT-PSP type inequality.

It remains to take the functions $\phi_{iu}(\tau)=c_{ij}(B(i,j)-\tau)$, $\phi_{uj}\equiv 0$ as the cost functions in the JIT-PSP reformulation of (3)-(6), and the reduction of PSP to JIT-PSP is completed.

Notice that we can easily include into the JIT-PSP model the *'deadline constraints'* $t_i\le d_i$, $i\in M$. Indeed, let us introduce a special, *initial*, event s and extend the original network by inserting new arcs (s,i), $i\in M$; let also $a(s,i)=-\infty$, $b(s,i)=d_i$, $\phi_{si}\equiv 0$, $i\in M$. It is clear that if $\{t_i\}$ is a feasible solution to the resulting JIT-PSP, then $\{t_i - t_s\}$ is also feasible (and of the same cost as the initial one); so the optimal solution to JIT-PSP can be found in the set of feasible solutions with $t_s=0$, and these solutions are exactly those which, besides the precedence conditions, satisfy the deadline constraints.

Appendix 2

In this appendix we prove Theorems 1 and 2.

For all $(i,j)\in V^+(t)$ let us denote

$$h_{ij}(u)=\begin{cases}\phi_{ij}(u+t_j-t_i)-\{\rho(i,j)+\lambda^*(i,j)\}\,u, & \text{case } A(i,j),\\ \phi_{ji}(-u+t_i-t_j)-\{\rho(i,j)+\lambda^*(i,j)\}\,u, & \text{case } B(j,i),\\ \phi_{ij}(u+t_j-t_i)-\{\rho(i,j)+\lambda^*(i,j)\}\,u, & \text{case } C(i,j).\end{cases}$$

Let $\tau=\{\tau_i\}$ be a feasible, not necessarily integral, solution to (3)-(4) and let $\Delta=\{\Delta_i\equiv\tau_i-t_i\}$. Then we have

$$\delta(\tau)\equiv\phi(\tau)-\phi(t) = \sum_{(i,j)\in V(t)} \{h_{ij}(\Delta_j-\Delta_i)-h_{ij}(0)\} +$$

$$+ \sum_{(i,j)\in V(t)} \{\rho(i,j)+\lambda^*(i,j)\}(\Delta_j-\Delta_i)=$$

$$=\left\{\sum_{(i,j)\in V(t)} \{h_{ij}(\Delta_j-\Delta_i)-h_{ij}(0)\}\right\}_1 + \left\{\sum_{j\in E} \gamma(j)\ \Delta_j\right\}_2 \tag{A1}$$

where for each $k\in E$

$$\gamma(k)=\sum_{i:(i,k)\in V(t)}\{\lambda^*(i,k)+\rho(i,k)\}-\sum_{i:(k,i)\in V(t)}\{\lambda^*(k,i)+\rho(k,i)\}=$$

$$=\sigma(k)+\mathrm{div}_k\lambda^*-\varepsilon(k)=\sigma(k)-\varepsilon(k), \tag{A2}$$

$$\varepsilon(k)=\begin{cases}\lambda^*(s,k), & \text{if } k \text{ is a source,}\\ 0, & \text{if } k \text{ is neither a source nor a sink,}\\ -\lambda^*(k,f), & \text{if } k \text{ is a sink}\end{cases} \tag{A3}$$

(we have taken into account (12) and (15); notice that λ^* is a flow, so $\mathrm{div}_k\lambda^*=0$).

Let us first prove the following

Lemma. Each term in the sum denoted by $\{\ \}_1$ (see (A1)) is nonnegative:

$$h_{ij}(\Delta_j-\Delta_i)\geq h_{ij}(0). \tag{A4}$$

Proof. Let us fix an arc $(i,j)\in V(t)$.

Assume that the case $A(i,j)$ takes place. Then we have

$$\{p(i,j)+\lambda^{*}(i,j)\}\in[p(i,j),\ p(i,j)+d(i,j)] =$$
$$= [\phi'_{ij}(t_j-t_i-0),\ \phi'_{ij}(t_j-t_i+0)] \qquad \text{(A5)}$$

(we have used in turn (10), (6) and the definition of $d(i,j)$ in the case under consideration). Inclusion (A5) means that $0\in[h'_{ij}(\tau-o)|_{\tau=0},\ h'_{ij}(\tau+o)|_{\tau=0}]$ and (A4) holds since h is convex.

The calculations above show that the following implications hold:

$\{(i,j)\in V(t)$ *and the flow* λ^{*} *saturates the arc* $(i,j)\}\Rightarrow$
$$h'_{ij}(\tau+0)|_{\tau=0}=0. \qquad \text{(A6)}$$

$\{(i,j)\in V(t)$, *the case* $A(i,j)$ *takes place, and* $\lambda^{*}(i,j)=0\}\Rightarrow$
$$h'_{ij}(\tau-0)|_{\tau=0}=0. \qquad \text{(A7)}$$

Indeed, in the cases $B(j,i)$ and $C(i,j)$ the capacity of the arc $(i,j)\in V(t)$ is infinite, so the arc cannot be saturated by the flow λ^{*}. Thus, the premise in (A6) implies that the case $A(i,j)$ takes place. Now $\lambda^{*}(i,j)=d(i,j)$ (the latter is the premise in (A6)) means that $\{p(i,j)+\lambda^{*}(i,j)\}=\phi'_{ij}(t_j-t_i+0)$ (due to (6) and to the definition of $d(i,j)$ in the case $A(i,j)$). The latter equality and the definition of $h_{ij}(\cdot)$ immediately lead to the conclusion in (A6).

Further, the premise in (A7) states that the case $A(i,j)$ takes place. Then $\lambda^{*}(i,j)=0$ (the latter is included into the premise in (A7)) implies that $\{p(i,j)+\lambda^{*}(i,j)\}=\phi'_{ij}(t_j-t_i-0)$ (by (6) and since $A(i,j)$ takes place). The latter equality and the definition of $h_{ij}(\cdot)$ lead to the conclusion of (A7).

Assume that the case $B(j,i)$ takes place. Then we have

$$\{p(i,j)+\lambda^{*}(i,j)\}\in[p(i,j),+\infty)=[-\Phi'_{ji}(t_i-t_j+0),+\infty)$$

(due to (10) and (6)).

The latter relation together with the definition of h_{ij} means that $h'_{ij}(t-0)|_{t=0}\le 0$, so, by the convexity of h_{ij}, $h_{ij}(u)\ge h_{ij}(0)$, if $u\le 0$.

Since $t_i-t_j=a(j,i)$ in the case $B(j,i)$ and, besides, t is feasible to (3)-(4), i.e. $t_i-t_j\ge a(j,i)$, we have $D_j-D_i\le 0$, and (A4) holds in the case $B(j,i)$. The above considerations also prove the implication

$\{(i,j)\in V(t)$, *the case* $B(j,i)$ *takes place,* $\lambda^{*}(i,j)=0\}\Rightarrow$
$$h'_{ij}(t-0)|_{t=0}=0. \qquad \text{(A8)}$$

Assume that the case $C(i,j)$ takes place. Then considerations similar to those of the previous case show that $h_{ij}(u)\geq h_{ij}(0)$, if $u\leq 0$; the latter in the case under consideration leads to (A4). By the same reasons the following implication holds:

By the similar reasons, the following holds:

$$\{(i,j)\in V(t),\ \text{the case } C(i,j) \text{ takes place},\ \lambda^{*}(i,j)=0\} \Rightarrow$$

$$h'_{ij}(\tau-0)\big|_{\tau=0}=0.$$

Lemma is proved.

Notice that (A7), and (A8) lead to the implication

$$\{(i,j)\in V(t),\ \lambda^{*}(i,j)=0\} \Rightarrow h'_{ij}(\tau-0)\big|_{\tau=0}=0. \qquad \text{(A9)}$$

Proof of Theorem 1. Let us prove that in the case of $\pi^{*}=\pi_{t}$, the right hand side in (A1) is nonnegative for each feasible τ. Since $\{\ \}_{1}$ is nonnegative due to Lemma, it suffices to verify that

$$\{\ \}_{2} = 0. \qquad \text{(A10)}$$

Obviously, in the network $N^{*}(t)$ there is the cut $(\{s\},\ E^{+}\setminus\{s\})$ with the capacity π_{t}, as well as the cut $(E^{+}\setminus\{f\},\ \{f\})$ of the same capacity (see (12)). In the case under consideration the maximal flow value, π^{*}, equals to the capacity of each of these cuts, so the arcs in $V^{+}(t)$ leading from s to the sources and from the sinks to f are saturated with the flow λ^{*}, or $\lambda^{*}(s,k)=\sigma(k)$ for any source k and $\lambda^{*}(k,f)=-\sigma(k)$ for any sink k. Hence, $\varepsilon(k)\equiv\sigma(k)$, and $\gamma(k)=0$, $k\in E$ (see (A2), (A3)), which immediately leads to (A10). Theorem 1 is proved.

Proof of Theorem 2. Assume that

$$\pi^{*}<\pi_{t}. \qquad \text{(A11)}$$

We are going to verify that $t'$is well defined, feasible for the problem (3)-(4), and that (8) holds.

First, the corresponding minimum cut $(E^{s},\ E^{f})$ of the network $(E^{+},\ V^{+}(t))$ (which, by the Max Flow - Min Cut Theorem, has the capacity π^{*}) can be neither $(\{s\},\ E^{+}\setminus\{s\})$ nor $(E^{+}\setminus\{f\},\ \{f\})$, since the capacities of the latter cuts are equal to π_{t}. Hence the sets $E_{s}=E\cap E^{s}$ and $E_{f}=E\cap E^{f}$ are both nonempty. By the standard properties of the maximal network flow each arc in $V(t)$ leading from E_{s} into E_{f} is saturated by λ^{*}, while the flow in arcs leading from E_{f} into E_{s} equals zero.

Let us verify that t' defined by (13) is feasible for (3)-(4), and satisfies (8).

Indeed, let $(i,j)\in V(t)$. By (13) the quantity $\tau'_{ij}\equiv t'_j-t'_i$ differs from $\tau_{ij}\equiv t_j-t_i$ iff the arc (i,j) either leads from E_s into E_f (*situation 1*), or leads from E_f into E_s (*situation 2*). In the situation 1 the arc is saturated by the flow λ^*, and hence it is of finite capacity; the latter can occur in the case $A(i,j)$ only. Thus, in the situation 1 the case $A(i,j)$ takes place, while the situation 2 can occur in each of the cases.

In the case $A(i,j)$, by definition of the cases, one has $a(i,j)<\tau_{ij}<b(i,j)$, while $|\tau_{ij}-\tau'_{ij}|=1$. These relations together with the integrality of the input data and of t, t' imply that $a(i,j)\le\tau'_{ij}\le b(i,j)$.

In the case $B(j,i)$ the arc (i,j) leads from E_f into E_s, so by (13) $\tau'_{ij}=\tau_{ij}-1=t_j-t_i-1=-a(j,i)-1$ (the latter is due to the definition of the case $B(j,i)$), so $t'_i-t'_j=a(j,i)+1$, and the integrality of the data leads to inequality (4) corresponding to the arc $(j,i)\in V$. By similar arguments in the case $C(i,j)$ the pair (t'_j,t'_i) satisfies inequality (4) corresponding to the arc $(i,j)\in V$.

Thus, t' is a feasible solution to (3)-(4).

It remains to prove that (8) holds. Let $\Delta_i=t'_i-t_i$, $i\in E$; we desire to verify that (see (A1))

$$\delta(t')\le-1.$$

By (A1), it sufficies to show that

$$h_{ij}(\Delta_j-\Delta_i)=h_{ij}(0),\quad (i,j)\in V(t) \tag{A12}$$

and

$$\{\ \}_2\le -1. \tag{A13}$$

Let us prove (A12). For given $(i,j)\in V(t)$, by the same reasons as above, one of three cases is possible:

(*i*) either $\Delta_j-\Delta_i=0$, or (*ii*) $\Delta_j-\Delta_i=1$, or (*iii*) $\Delta_j-\Delta_i=-1$.

In the case (*i*), (A12) is obvious.

Further, in the case (*ii*) we have:

$$\{(j\in E_f)\text{ and }(i\in E_s)\}\text{ (due to (13))},$$

$$\{(j\in E_f)\text{ and }(i\in E_s)\}\Rightarrow\{A(i,j)\text{ and }(\lambda^*(i,j)=d(i,j))\}$$

(by the reasons described above, in the proof of the feasibility of t'),

$\{A(i,j)$ and $(\lambda^*(i,j)=d(i,j))\}\Rightarrow\{h'_{ij}(\tau+0)|_{\tau=0}=0\}$ (due to (A6)),

$$\{h'_{ij}(\tau+0)|_{\tau=0}=0\} \Rightarrow \{h_{ij}(1)=h_{ij}(0)\}$$

(since $h_{\bullet\bullet}$, as well as $\phi_{\bullet\bullet}$, has only integer breakpoints). So in the case *(ii)* relation (A12) also holds.

In the case *(iii)* we similarly have

$$\{(j\in E_s) \text{ and } (i\in E_f)\} \text{ (due to (13))},$$

$$\{(j\in E_s) \text{ and } (i\in E_f)\} \Rightarrow \{\lambda^*(i,j)=0\}$$

(since the flow λ^* in the arcs leading from E_f into E_s equals zero),

$$\{\lambda^*(i,j)=0\} \Rightarrow \{h'_{ij}(\tau-0)|_{\tau=0}=0\} \text{ (due to (A9))},$$

$$\{h'_{ij}(\tau-0)|_{\tau=0}=0\} \Rightarrow \{h_{ij}(-1)=h_{ij}(0)\}$$

(since $h_{\bullet\bullet}$, as well as $\phi_{\bullet\bullet}$, has only integer breakpoints). So in all the cases relation (A12) holds.

Now let us prove (A13). From (13) it is clear that

$$\{\ \}_2 = \sum_{k\in E_f} \gamma(k).$$

If $k\in E_f$ is a source, then the arc (s,k) leads from E^s into E^f and hence is saturated by the flow λ^*; hence, by (A2) and (A3), $\gamma(k)=0$. Further, if $k\in E_f$ is neither a source nor a sink, then, by (A2),(A3) and by the definition of sources and sinks, $\gamma(k) = 0$. Now let $k\in E_f$ be a sink. Then, by (A2), and (A3), $\gamma(k)=\lambda^*(k,f)-|\sigma(k)|$. Thus,

$$\{\ \}_2 = \sum_{k\in E_f:\ k \text{ is a sink}} \{\lambda^*(k,f)-|\sigma(k)|\}. \qquad (A14)$$

Notice now that if k is a sink and k does not belong to E_f, then the arc (k,f) leads from E^s into E^f and hence it is saturated by the flow λ^*, therefore $\lambda^*(k,f)=|\sigma(k)|$. Thus, (A14) leads to

$$\{\ \}_2 = \sum_{k\in E:k \text{ is a sink}} \{\lambda^*(k,f)-\sigma(k)\}=\pi^*-\pi_t.$$

The latter quantity is a negative integer due to (A11)and therefore (A13) holds. The proof is over.

References

[1] Adelson-Velsky, G.M., Voropaev, V.I., and Kalinovska S.S. "Planning on generalized network models", **Proceedings of 4th Winter School on Mathematical Programming**, Central Economical-Mathematical Institute, Moscow, v. 3, 1972, 3-18 (in Russian).

[2] Ahuja, R.K., Magnanti, T.L., and Orlin, J.B., "Network flows", in:G.L.Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd (eds.), **Handbooks in Operations Research and Management Science, Volume 1: Optimization**, North-Holland,Amsterdam, 1989, 211-369.

[3] Anthonisse, J.M., Hee,K.M. van, and Lenstra, J.K., "Resource-constrained project scheduling: an international exercise in DSS", **Decision Support Systems**, 3 (1988) 249-257.

[4] Elmaghraby, S.E., and Pulat, P.S., "Optimal project compression with due-dated events", **Naval Research Logistics Quarterly**, 2 (1979) 331-348.

[5] Foldes, S, and Soumis F., "PERT and crashing revisited: mathematical generalizations", **Proceedings of 2nd International Workshop on Project Management and Scheduling**, Universite de Technologie de Compiegne, France, 1990, 398-406.

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

[7] Goldberg A.V., Tardos E., and Tarjan, R.E. "Network flow algorithms", in: B.Korte, L.Lovasz, H.J.Promel, and A.Schrijver (eds), **Algorithm and Combinatorics, Volume 9: Paths, Flows, and VLSI-Layout**, Springer-Verlag, Berlin, 1990, 101-164.

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

[9] Kerbosch, J.A.G.M., and Schell H.J., "Network planning by the Extended METRA Potential Method", Report KS-1.1, Eindhoven University of Technology, Eindhoven, Netherlands, 1975.

[10] Levner, E.V., "A parametric network analysis model", in: A.A.Fridman (ed.), **Studies in Discrete Optimization**, Nauka, Moscow, 1976, 382-394 (in Russian).

[11] Levner, E.V., Nemirovsky A.S., "A network flow algorithm for just-in-time project scheduling", **Proceedings of** the **2nd International Workshop on Project Management and Scheduling**, Universite de Technologie de Compiegne, France, 1990, 224-227.

[12] Moder, J.J., Phillips, C.R., and Davis, E.W., **Project Management with CPM, PERT and Precedence Diagramming**, Van Nostrand Reinhold Company, New York, 1983

[13] Zukhovitsky, S.I., and Radchik, I.A. **Mathematical Methods of Project Management**, Nauka, Moscow, 1965 (in Russian).]

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computing Science
PROBABILITY THEORY, STATISTICS, OPERATIONS RESEARCH
AND SYSTEMS THEORY
P.O. Box 513
5600 MB Eindhoven, The Netherlands

Secretariate: Dommelbuilding 0.03
Telephone : 040-473130

-List of COSOR-memoranda - 1991

Number	Month	Author	Title
91-01	January	M.W.I. van Kraaij W.Z. Venema J. Wessels	The construction of a strategy for manpower planning problems.
91-02	January	M.W.I. van Kraaij W.Z. Venema J. Wessels	Support for problem formulation and evaluation in manpower planning problems.
91-03	January	M.W.P. Savelsbergh	The vehicle routing problem with time windows: minimizing route duration.
91-04	January	M.W.I. van Kraaij	Some considerations concerning the problem interpreter of the new manpower planning system formasy.
91-05	February	G.L. Nemhauser M.W.P. Savelsbergh	A cutting plane algorithm for the single machine scheduling problem with release times.
91-06	March	R.J.G. Wilms	Properties of Fourier-Stieltjes sequences of distribution with support in [0,1).
91-07	March	F. Coolen R. Dekker A. Smit	Analysis of a two-phase inspection model with competing risks.
91-08	April	P.J. Zwietering E.H.L. Aarts J. Wessels	The Design and Complexity of Exact Multi-Layered Perceptrons.
91-09	May	P.J. Zwietering E.H.L. Aarts J. Wessels	The Classification Capabilities of Exact Two-Layered Peceptrons.
91-10	May	P.J. Zwietering E.H.L. Aarts J. Wessels	Sorting With A Neural Net.
91-11	May	F. Coolen	On some misconceptions about subjective probability and Bayesian inference.

COSOR-MEMORANDA (2)

91-12	May	P. van der Laan	Two-stage selection procedures with attention to screening.
91-13	May	I.J.B.F. Adan G.J. van Houtum J. Wessels W.H.M. Zijm	A compensation procedure for multiprogramming queues.
91-14	June	J. Korst E. Aarts J.K. Lenstra J. Wessels	Periodic assignment and graph colouring.
91-15	July	P.J. Zwietering M.J.A.L. van Kraaij E.H.L. Aarts J. Wessels	Neural Networks and Production Planning.
91-16	July	P. Deheuvels J.H.J. Einmahl	Approximations and Two-Sample Tests Based on P - P and Q - Q Plots of the Kaplan-Meier Estimators of Lifetime Distributions.
91-17	August	M.W.P. Savelsbergh G.C. Sigismondi G.L. Nemhauser	Functional description of MINTO, a Mixed INTeger Optimizer.
91-18	August	M.W.P. Savelsbergh G.C. Sigismondi G.L. Nemhauser	MINTO, a Mixed INTeger Optimizer.
91-19	August	P. van der Laan	The efficiency of subset selection of an almost best treatment.
91-20	September	P. van der Laan	Subset selection for an ε-best population: efficiency results.
91-21	September	E. Levner A.S. Nemirovsky	A network flow algorithm for just-in-time project scheduling.