

Activity Nets: A Guided Tour Through Some Recent Developments

S. E. Elmaghraby

Center for Communications and Signal Processing
Graduate Program in Operations Research
North Carolina State University

OR Reprt. No. 282

TR-94/13
May 1994

ACTIVITY NETS: A GUIDED TOUR THROUGH SOME RECENT DEVELOPMENTS

Salah E. Elmaghraby

North Carolina State University, Raleigh NC 27695-7913, USA

May, 1994

1. PERSPECTIVE

The first papers on activity nets (under the names of the CPM and PERT) appeared in 1959. Therefore, it is interesting to note that the first review by Bigelow [10] appeared in 1962, a mere three years later! It covered the years 1959-1961, and classified contributions according to: basic principles, industrial applications, and military applications of the PERT methodology. In 1966 Lerda-Olberg [47] provided a review covering the years 1962-1965 in which contributions were classified according to: general, theory, programming, and applications. The next landmark review is due to Adlakha & Kulkarni [1] which appeared in late 1989 and covers, in the main, the years 1966-1987 (with few exceptions from prior years). It concentrates on *stochastic PERT networks* and is 'methodology oriented', discussing, in sequence: pre-estimation analysis, error/bias due to assumptions, exact analysis, and Monte Carlo Sampling approaches.¹

Mention should also be made of the books on activity nets, since they constitute important landmark summaries of the state-of-the-art at the time of their writing (which is typically some 12 to 18 months before their appearance), arranged chronologically: Battersby in 1970 [7], Whitehouse in 1973 [67], Elmaghraby in 1977 [21], Kerzner in 1979 [43], Moder, Phillips & Davis in 1983 [50], and Słowiński & Weglarz in 1989 [59].

This guided tour is more 'problem oriented' in the sense of addressing the issues that are of concern to managers of large scale projects, and hence are, or should be, also of concern to scholars and researchers in the field. Consequently, this tour is 'applied' in perspective because of that orientation, though we hasten to emphasize that we discuss contributions to theory and methodology everywhere.

¹A 'state of the art' review from a practitioner's point of view is provided in the 1987 paper by Archibald [4].

Still, our classification stems from the point of view of issues and concerns rather than theoretical results or methodology. We shall confine ourselves, on the whole, to developments that occurred in the years 1987-94 with some forays into earlier contributions to maintain continuity of presentation.

What are the issues of interest to managers? They may be grouped under four general headings: (i) *Representation* and *Modeling* for visualization and analysis; (ii) *Scheduling* activities subject to resource constraints; (iii) *Financial* issues, either related to project ‘compression’ or to cash flows; and (iv) *Uncertainty* in activity durations as well as in resource availabilities and/or cash flows, and how to cope with it.

This tour shall concentrate on issues (i)-(iii) the proximity of the excellent review by Adlakha & Kulkarni should fill the needs on issue (iv) with very few exceptions that can be easily secured by the serious reader.

Section 2 discusses the available software for the PC. Sections 3 and 4 discuss issues of representation: activity-on-node *versus* activity-on-arc, and presents some subtle concerns that may escape the unwary. Sections 4-7 address the issue of ‘complexity’, first from the general concern of management that seeks ‘minimal execution time’ in response to its queries, then from the technical point of view of algorithmic procedures. Section 8 returns to the issue of representation to discuss ‘generalized precedence relations’. The important questions related to limited resources are discussed in section 9. Finally, section 10 presents three issues grouped under the rubric of ‘financial considerations’: the ‘bidding’ problem, the ‘net present value’ problem, and the ‘optimal project compression’ problem. The references given at the end of the paper are not comprehensive; we are content, in the main, to cite the latest contributions in a particular area. The interested reader may consult the references cited therein.

2. PC-BASED SOFTWARE PACKAGES

A plethora of “off-the-shelf” project planning and control software packages for PC’s are currently available, and the popular magazines as well as scientific journals regularly publish reviews of new entries and improved versions of older releases; see for example, *Edwards et al* [20] and *Fresko-Weiss* [33] in *PC Magazine*; *Hogan et al* [37] in *Business Software*; *Assad & Wasil* [5]; and *De Wit & Herroelen* [15] in more scientifically oriented journals.

The newcomers to the field of ANs, as well as weathered project managers, are oftentimes bewildered by the extensive variety and the exaggerated claims of the

available software. How can one sort out the “good” from the “mediocre” from the “bad”, and how does one go about deciding which software to acquire, if any?

We are aware of a couple of checklists and guidelines to assist in the decision making process; they are: Project Management Advisory Group, Government of Canada, in Hull, Quebec, and the Belgian Building Research Institute (WTCB-CSTC). Lozenberg I, 7, 1932 St. Stevens-Woluwe (Zaventem), Belgium. Unfortunately, to the best of our knowledge, in the USA there is no official governmental office or professional society that conducts periodic evaluation of available software. This is left to private consulting agencies.²

A review of the vast amount of publications on this subject reveal several important and interesting facets, which we group under the titles “nomenclature”, “desiderata”. and “performance”.

2.1. Nomenclature

It is a pity indeed that the popular jargon on project planning and control ignores the established scientific terminology that has found its way into textbooks on project planning and that is internationally accepted among teachers and researchers in the field. For instance, it is “street language” to speak of “arrow diagrams” and “precedence diagrams” instead of *activity-on-arc* (AoA) and *activity-on-node* (AoN) diagrams. The street language is meaningless since both modes of representation have arrows and both represent precedence! (This is in contrast to Gantt charts, for instance.) And for another example, the process of smoothing resource requirements by re-scheduling activities within their available floats subject to *fixed* project duration is known in the scientific literature as *resource leveling*, naturally enough; its street name is “resource loading” a name that is void of meaning! As a third and final example, the process of re-arranging the schedule of the activities to remain within the confinements of limited resources and minimize the project duration is known in the scientific literature as *resource-constrained scheduling*; yet it is commonly referred to among shop people as “resource leveling”!

This disparity is not only lamentable but also harmful, since it creates a schism between the research/academic community and the practitioners. The schism is easily avoidable. It is the purpose of language to facilitate communication, which is destroyed when the same object or activity is called by different names by different groups. Concerted effort should be directed towards establishing a

²For instance: One Soft Decision, Box 6123, San Rafael, CA 94903-0123.

unified terminology that describes precisely and concisely the various objects and activities in this field, and that is used by all.

2.2. Desiderata

Any software for project planning and control must satisfy a set of minimal requirements, which typically relate to the time planning aspects of the project such as the earliest and latest realization times of events, the critical path(s), the activity floats, *etc.* But the majority of users demand more; in fact, much more, especially with regard to resources and financial considerations. The paper by De Wit & Herroelen [15] gives a comprehensive and definitive exposition of these demands, and outlines the criteria against which a software should be judged. These criteria include what the software requires as inputs, how it reflects such inputs in its representation of the project as a network or another format (such as Gantt charts, tables, charts, *etc.*), how (and how well) it responds to the dynamic updating of information, how (and how correctly) it resolves representation and optimization questions, how it reports the results of the analysis. These criteria represent a different view of the process of software selection and performance evaluation from that usually presented in trade journals, which usually report the ‘outside parameters’ of the software³ rather than an in-depth evaluation of the correctness of its outputs.

2.3. Performance

The conclusions of the study by De Wit & Herroelen are illuminating. To a varying degree of sophistication, all software satisfy the basic requirements of analysis and reporting, especially when the size of the project is small. Differences start to appear as the size (as reflected in the number of activities and resources) grows: inputting the data of the project gets to be a chore by itself, let alone conducting a meaningful analysis *from a decision-making point of view* (rather than ‘accounting’ point of view). The following quotations are sample excerpts from the paper.

“... we alert the reader to the fact that some of the packages suffer from the serious drawback that the Gantt charts will show *incorrect*

³Such as the maximal number of activities and precedence relations, the maximal number of resources, the type of calendar(s) it uses, the reports that can be generated, *etc.*

latest start times when activities are related by start-start and end-end relationships on the precedence diagrams.”

“Most of the programs tested will not allow networks to be represented in AoA representation commonly used in *PERT/CPM* type of networks. ... The user is urged to check carefully this matter because the vendor publications often misleadingly refer to the *PERT/CPM* method where the software actually supports the AoN diagramming method, or *vice versa*.”

“... the packages ... differ widely in their ability to handle *non-work days* and *holidays* ... (which is) an important feature (since) long projects with a duration of 2 or 3 years require the ability to specify at least some 80-100 holidays.”

“It is almost inevitable that *errors* are made during the data entry phase. The creation of *activity loops* and *dangles* ... are typical examples. ... A diagnostic such as ‘a loop has been detected, please check your data’ is not much help in even moderate-sized networks”

“The various packages ... seem to have their own philosophy for the computation of the *start and finish times of the network activities* when certain types of precedence relations are specified. The surprising result of this feature is that the (minuscule) test project (when presented to different packages) was ‘given’ a duration which varied from 64 to 84 days!”

“The software may also update the duration of some activities without having any impact on the total project completion time. ... The computed start and finish times are correct. The length of the activity bars in the Gantt chart, however, may not be correctSome packages do not update the activity duration nor the duration of the precedence relations, but give *erroneous latest start times*.”

“None of the packages tested enable the project planner to perform true *time-cost trade-off analysis* in order to estimate the effect of different activity resource allocations and the resulting activity direct costs on total project cost.”

“Although several packages do offer *resource planning* features allowing the planner to obtain reports on the resource usage in specific time intervals, resource profile charts and cumulative resource usage charts, the *resource monitoring* capabilities are not only very primitive but dangerously misleading. ... Regrettably, we cannot recom-

mend the use of the packages' suboptimizing resource monitoring procedures to the project planner, except in those rare situations where the technical know-how and expertise are available to evaluate the precise algorithmic steps as well as their impact on the project schedule."

To sum up, the user beware of 'off the shelf' software and their exaggerated claims: apart from cosmetic differences, many fail to respond to some basic needs, and all are incapable of correctly carrying out optimization procedures.

3. PROJECT REPRESENTATION IN THE AoA MODE

Normally, the statement of the project is in the form of a set of activities and the *immediate* precedence relations among them. If activity u precedes activity v , it is written as $u \prec v$. Henceforth we shall refer to the set of precedence relations by " \prec ". The following table gives such specification for a small, albeit most interesting, project composed of only four activities, a, b, c, d ; with four precedence relations \prec , which result in the so-called *interdictive graph* (IG).⁴, and which we shall use as a vehicle for introducing the concepts underlying the construction of the AoA representation.

The specification of the IG.

$$\begin{array}{ll} \emptyset \prec & a, b \\ a \prec & c, d \\ b \prec & d \\ c, d \prec & \emptyset \end{array}$$

The AoN representation (typically denoted by the letter G) of this (minuscule) project is the most straightforward and "natural" representation, which is shown in Fig.1a. Its AoA representation is shown in 1b. Note the need for the *dummy activity* joining nodes 2 and 3 to maintain the integrity of the precedence relations.

Figure 1. (a) The AoN representation of the IG.
(b) Its AoA representation.

⁴Also variably known, as far as we are aware, as the *forbidden graph*, the *cross-over graph*, the *Wheatstone's bridge*, the *Z-graph*, and the *N-graph*.

On the other hand, the AoA representation of the same project, commonly referred to as the graph D is *not* unique, and is cluttered with dummy activities and dummy events for three different reasons. They are: (i) to comply with the requirement that each activity is uniquely identified by its terminal nodes (avoid multigraph - *ie*, a graph with two or more activities between the same two nodes); (ii) to respect the specified precedence relations; and (iii) to comply with the requirement that the resulting network is *two-terminal*. For instance, Figs.2b,c give two different AoA representations of the *same* project of Fig.2a. Note that Fig.2b had 9 nodes and 6 dummy activities, while Fig.2c has 10 nodes and 10 dummy activities!

Figure 2. A project and its two different AoA representations.

Undoubtedly, the AoN representation is the more direct, the more frugal, and is unique. Why, then, do analysts find it sometimes preferable to adopt the AoA mode of representation? From a purely *representational* point of view, the answer is: when it is important to graphically identify the *events* of the project. This is the case, for instance, when payment is related to the realization of certain events, in which case the AoA representation is convenient to highlight these *key events*. It is also the case when it is desired to visually identify all completed activities at a particular event, or the activities leading to the event's realization. Finally, the AoA mode is preferred when it is desired to give a visual representation of the duration of the activities, then the arc length is made proportional to the duration of the activity. From an *analytical* point of view, the AoA mode is preferred when there are more complex relations among the activities of the project, such as in the presence of *generalized precedence relations* (see §8). Or when it is desired to represent the activity "floats" (see §4). Or when we wish to construct mathematical models that depend on the definition of nodes, such as the linear program models for the optimal time-cost trade-off (see §10.3.1), or any of the various models for the determination of the probability distribution functions of the time of realization of events. Or when we wish to determine the "complexity index" of the project network (see §6).

To satisfy one or more of the desiderata in the construction of the AoA representation the process may be carried out with different objectives in mind. We enumerate five of them:

1. Minimize the number of nodes and the number of dummy activities. (Even a 10% savings in a network of 50,000 activities represents 5,000 arcs which we need not draw or be concerned with!)
2. Minimize the number of nodes.
3. Minimize the number of dummy activities. (Note that the number of arcs representing the “real” activities is constant.)
4. Minimize the number of dummy activities given the minimum number of nodes.
5. A fifth objective relates to minimizing the *complexity index* of the resulting AoA network. The full import of this criterion will become clearer after we introduce the concepts related to measures of complexity, see §6 below.

It is known that it is impossible to minimize *both* the number of nodes and the number of dummy activities. Consequently, Objective 1 is infeasible. Furthermore, it has been recently established that the minimization of the number of dummy activities, whether or not subject to the minimum number of nodes, is NP-Complete, implying that the task becomes onerous, then impossible, as the size of the network grows. The proof that the “dummy-arc” and the “constrained dummy-arc” problems are NP-Hard can be found in Krishnamoorthy & Deo [45].

That, however, did not detain researchers in the field from attacking various aspects of the problem. Objective 2 has been addressed and it is possible to construct an AoA representation with the minimum number of nodes. This can be achieved in polynomial time; see Cantor & Dimsdale [12] and Sterboul & Wertheimer [61]. See also Syslo [64] for a good review of the minimal dummy arc problem as of that date, and for the demonstration, through a simple counter-example, that one cannot simultaneously minimize both the number of dummy arcs and the number of nodes.

Then, realizing that one is usually interested in constructing a network with a minimal number of nodes anyway, we skip over objective 3 and assert that it is possible to satisfy objective 4, *viz.*, the construction of the AoA representation of a project that has the minimum number of nodes and also minimizes the number of dummy arcs over this set of nodes, but at a price! This is because the problem is translated into a minimum *set-cover* problem, which is known to be NP-Complete. But it is a set-cover problem over a much smaller set than would originally be

envisioned, and solution may still be achieved optimally for medium size problems (of a few hundred activities), and approximately for large size problems. The constrained minimization of the number of arcs has been recently solved by Michael, Kamburowski & Stallmann [49]. A discussion of the need for in-dummy and out-dummy nodes may be found in the paper by Elmaghraby & Kamburowski [26]

The following remarks are pertinent to the construction of the AoA representation.

- The construction may lead to a multigraph (*ie*, two or more arcs start and end at the same two nodes). To see this, construct the AoA representation of the simple project composed of only five activities a, b, c, d, e with the following precedence relations: $(a \prec b)$, $(b \prec c, d)$, $(c, d \prec e)$. You shall discover that there is no need for any dummies, but that activities c and d run in parallel between the same two nodes. If multigraphs are prohibited then other nodes will have to be added as required. The resultant graph is still of minimal nodes among all AoA representations that respect the restriction to non-multigraphs.
- At the outset, the numbering of the nodes is arbitrary. After the AoA graph is complete, one may need to re-number the nodes to adhere to topological ordering (*ie*, an arc always leads from a small number to a larger one).
- While the movement from the AoN to the AoA is difficult and raises a number of issues relative to events (nodes) and the precedence among activities, the reverse movement from AoA to AoN is straightforward. As always, the resulting graph is unique except for isomorphism.

3.1. In-Dummy and Out-Dummy nodes

Two interesting phenomena appear in the solutions of the dummy-arc problem of a project, namely, the appearance of nodes whose input arcs *or* their output arcs are all dummies, and the absence of nodes whose input arcs *and* output arcs are all dummies!

It is easy to explain the latter phenomenon since the search for the minimal dummy arcs necessarily precludes the presence of nodes whose input arcs *and* output arcs are all dummies. It can be easily demonstrated that such nodes can be deleted with no loss in the \prec relations but with a possible gain in the number of dummies.

As to the former phenomenon, let us call the nodes whose input arcs are all dummies by *in-dummy nodes*, and call the nodes whose output arcs are all dummies by *out-dummy nodes*. It is well known (see §4 below) that the presence of these nodes can play havoc with three out of the four activity floats in the AoA representation of the project and, as a consequence, lead to erroneous scheduling and resource allocation decisions that are based on these floats. Therefore their presence should be avoided. Can they be eliminated altogether? The answer is no, in general: under certain conditions the AoA net must contain in-dummy or out-dummy nodes in order to maintain the integrity of the specified precedence relations.

4. ON THE DEPENDENCE OF ACTIVITY FLOATS ON PROJECT REPRESENTATION

A project is defined by a set of “real” activities A and a set of immediate precedence relations R . In the AoA mode, $D = (N, A)$, in which N is the set of nodes of the network, usually referred to as the “events”, and the “real” activities A_r are now represented by a subset of the arcs A , with the difference $A_d = A - A_r$ constituting the *dummy activities* that are required to satisfy the three conditions imposed on such representation (see page 7).

The temporal analysis of the AN’s resulted in the definition of several parameters, not the least important among which are the *earliest* and *latest start and finish times of an activity*, which gave rise to the concept of *activity floats*. There are four such floats, namely (with reference to activity v , with $A(v)$ denoting the activities immediately after v and $B(v)$ denoting the activities immediately before v),

$$\text{“total float”}: \quad TF(v) = LS(v) - ES(v) = LF(v) - EF(v);$$

$$\text{“free float”}: \quad FF(v) = \min_{w \in A(v)} \{ES(w)\} - EF(v);$$

$$\text{“safety float”}: \quad SF(v) = LS(v) - \max_{u \in B(v)} \{LF(u)\};$$

$$\text{“interference float”}: \quad IF(v) = \min_{w \in A(v)} \{ES(w)\} - \max_{u \in B(v)} \{LF(u)\} - y(v).$$

These floats play an important role in two issues of central concern to managers: resource allocation and activity scheduling, since floats give a measure of the flexibility in scheduling the activities during the project execution without delaying the project completion time. Since the problems of optimal resource allocation and activity scheduling subject to the known precedence constraints are NP-Hard, practical solutions are achieved through the use of heuristics. Most importantly, almost all the known heuristics used in practice rely on ranking the activities according to their float – any one of the four. Therefore, the correct evaluation of the activity floats plays an important role in achieving “good” results in these two areas of endeavor.

The activity floats defined on the AoN mode of representation have intuitive interpretations. Relative to an activity, the TF is its maximum permissible delay without delaying the project duration. The FF is the maximum permissible delay when all its succeeding activities start as early as possible and all its preceding activities finish as early as possible. The SF is the maximum permissible delay when all its preceding activities finish as late as possible and all its succeeding activities finish as late as possible. If the interference float is positive, then it describes the maximum permissible delay when all its succeeding activities start as early as possible and all its preceding activities finish as late as possible. In case it is negative, the IF is the minimum required shortening of the duration of the activity to allow all its succeeding activities to finish as early as possible and all its preceding activities to finish as late as possible.

The AoN representation of a project is unique, and it is easy to demonstrate that the earliest and latest start and finish times of the activities are uniquely defined, and so are the activity floats, since all of them are based solely on the activity durations and the precedence relations among them.

But the AoA representation is not unique, due mainly to the presence of the dummy arcs. Obviously, different AoA representations may result in different number of nodes n and dummies $|A_d| = |A| - |A_r|$. There are several objectives in the construction of the AoA representation, among which are the objectives of minimizing either the number of nodes, or the number of dummies over the minimal number of nodes. Observe that since the minimization of the number of nodes is achievable in polynomial time, the networks constructed shall be, in all probability, of minimal number of nodes, or close to it.

In the AoA mode of representation, one defines the earliest and latest start and finish times of the events. Let $t_i(E)$ denote the earliest realization time,

and $t_i(L)$ the latest realization time of node i . It can be proved that for any activity (ij) , $t_i(E)$ and $t_j(L)$ are representations invariant. For a dummy, this is interpreted as $t_j(L)$ for each activity terminating at the start node of the dummy activity, and $t_i(E)$ for each activity initiating at the end node of the dummy (such activities must exist due to the absence of dummy nodes; *ie*, nodes with all dummies incoming and outgoing), see Fig.3.

Figure 3. a_r is a real activity; a_d is a dummy activity.

It immediately follows that the TF is the only float in the AoA mode that is *representation-invariant* *ie*, it is the same in both modes of representation and in all AoA models of the same project, since it is the only one that depends solely on these two parameters. The other three floats are *representation-dependent*; *ie*, they do indeed *depend on the structure of the AoA*.

Why is this important? Simply because if there are several AoA representations of the *same project*, one may obtain different activity floats, and consequently different resource allocations and different schedules while applying the *same heuristic procedure*. Worse still, the floats not only vary among themselves but they may also vary from their AoN values, a most disconcerting result!

It turns out that the culprit for such representation-dependent behavior of the floats in the AoA mode of representation are the *in-dummy nodes* (*ie*, nodes with all incoming arcs dummy) and the *out-dummy nodes* (*ie*, nodes with all outgoing arcs dummy). This is because when one considers dummy activity $v \equiv (ij)$ joining an out-dummy node i to an in-dummy node j , the values of $t_j(L)$ for the in-dummy node j and $t_i(E)$ for the out-dummy node i are not representation-invariant, since they coincide with $t_{j'}(E)$ for some preceding activity u , and to $t_{i'}(L)$ for some succeeding activity w , respectively. Neither $t_{j'}(E)$ nor $t_{i'}(L)$ is representation-invariant. If we are to avoid the pitfall of representation-dependent parameters, in-dummy nodes and out-dummy nodes should be eliminated. Can this be achieved? The answer is no, in general. It seems that in order to give a faithful representation of the precedence relations R of the project, the terminal nodes of some activities must be out-dummy, and the start nodes of some other activities must be in-dummy. Only when the AoN graph is series-parallel that one can construct an AoA representation that is totally free of dummies. (For a definition of series-parallel digraphs and procedures for their recognition, see Valdes, Tarjan & Lawler [65].)

If one cannot eliminate in-dummy and out-dummy nodes, can one correct for their presence? Fortunately, the answer to this question is yes: $t_i(E)$ for out-dummy nodes and $t_j(L)$ for in-dummy nodes must be re-defined to correspond to invariant activity set properties rather than as defined by the standard CPM model, where they are subject to the vagaries of the chosen structure of the AoA representation; see the paper by Elmaghraby & Kamburowski [26], which elaborates in full on the theoretical considerations underlying the correction.

5. ON NETWORK COMPLEXITY AND ITS MEASUREMENT

Managers of projects, as well as analysts of ANs, have a definite sensation of the *varying degrees of difficulty in the analysis and synthesis* of different projects. That this difficulty is ‘real’ - in the sense of being felt by a large majority of workers in the field - is an empirically verifiable fact. We take this sensation as the *definition* of complexity, in the same way as the sensation of ‘heaviness’ defines ‘weight’. Which brings forth the issue of *measurement*: can one measure the complexity of a project network in the same way one measures, say, viscosity of a substance or the time constant of a dynamical system? This question evokes another consideration: *why* would one need to measure the ‘complexity’ of a project? There are two possible applications: (a) The measure would serve as a *predictor* of the processing time requirements for a particular software package using a particular set of computing facilities, once the software/hardware system has been calibrated; (b) The measure would be used to *compare* two proposed algorithms for the analysis of ANs. Indeed, it is unfair to pronounce one software package to be superior to another unless both have been tested against the same set of projects of varying degrees of difficulty. Hence the need for a *standard testset*, and a prerequisite for the establishment of such a set is the presence of a measure of an AN.

This issue was discussed at length by Elmaghraby & Herroelen [24]. Three conclusions followed from that research:

1. The possible confounding of the measure of network complexity by the procedure of analysis. Indeed, one cannot usually separate the measurement of complexity (or, for that matter, anything) from the technology available to accomplish it.

2. The measure of network complexity may not inherit any of the properties of its constituent elements, such as additivity, linearity, conservation of inequality over the real line, *etc.* Again, this is quite common in measures of physical entities.
3. One cannot separate the measure of complexity from the purported use of that measure. In other words, an AN may be “easy” for one application (such as determining the CP) but “hard” for another (such as estimating the optimal resource allocation for early project completion) ! This is in harmony with the general theory of combinatorial complexity: the classification is relative to the *objective* and not relative to the *content* of the problem.

The need for a standard test set was felt by several researchers. Patterson [53] provided a set of 110 problems which served as the standard for a number of years. These problems have been supplanted recently by a set of problems due to Kolisch, Sprecher & Drexel [44]. Both sets were developed without regard to the theory of complexity index in AN's, developed by Bein, Kamburowski, & Stallmann [8], which constitutes the subject matter of §6. The significance of the complexity index in accounting for the difficulty in analysis has been investigated recently by De Reyck & Herroelen [14]. They discovered that it plays a dominant role when attention is focused on resource constrained project scheduling. Its significance in the estimation of the project parameters under the PERT model assumption of random activity durations, and in the discrete time/cost trade-off issues have already been established based on theoretical considerations.

As of the time of this writing (mid-1994) work has just been completed by Agrawal, Elmaghraby & Herroelen [2] on the first phase of an ambitious research program directed towards the establishment of software for the generation of test sets based on the theory of complexity index. When completed, an analyst wishing to examine a new procedure, or a manager contemplating the acquisition of a new software package, shall be able to generate any desired number of test projects to the analyst's, or the manager's, own specification with regard to size (arcs and nodes), complexity index, as well as resources and other pertinent data such as cost, probability distribution functions, *etc.*, and to assess the performance of the new procedure or the new software under a wide spectrum of conditions before making a firm commitment to the new procedure or to the purchase of the software.

6. THE DETERMINATION OF THE COMPLEXITY INDEX (CI) OF AN AoA REPRESENTATION OF A PROJECT

In four separate attempts at optimization over a *dag* ⁵, two of which are in the mainstream of ANs (resource allocation *via* dynamic programming, and the estimation of the *pdf* of the time of realization of an event in the PERT model), it has been observed that achieving an analytical solution demands that one must first achieve the desired result *conditional upon certain activities* then remove the conditioning through either enumeration (in the case of resource allocation) or through multiple integration (in the case of the estimation of the *pdf*). To minimize the computing effort one must minimize the number of activities on which such conditioning takes place. In each instance of any one of these four problems it was noted that the resolution of the problem is theoretically ‘straightforward’ (though it may be computationally demanding) if the *dag* were series-parallel, and that departure from such structure changes the problem from polynomially bounded to NP-Hard.

The problem thus reduces to the following: Given an *irreducible* AoA representation of a project ⁶ as *1,n-dag* $D = (N, A)$, how to determine the smallest number of activities to *fix*? (Which is shorthand for saying: fix the amount of resource allocated to the activities when we are interested in the optimal compression of the project in time-cost trade-off problems; or fix the activities’ durations when we are concerned with determining the *pdf* of the time of realization of an event.)?

The answer lies in determining the “*complexity index*” (*CI*) of the *dag*. The *CI* defines the minimum *number* of activities to fix; their *identity* is deduced from the *complexity graph* $C(D)$, which is constructed in the process of such determination.

The key to understanding the procedure is the realization that it is the presence of the interdictive graph (IG) that is at the root of the problem. A project’s *dag* is irreducible through a sequence of series-parallel reductions to the “trivial graph” of one arc $(1, n)$ because of the presence of at least one IG. ⁷ Thus, we are after the detection of the presence of the embedded IG’s in the project’s *dag* and the determination of the minimal set of arcs whose fixing would “neutralize” the presence of these IG’s. The complexity graph $C(D)$ identifies these IG’s. The

⁵For a detailed description of all four problems see the paper by Elmaghraby et al [30].

⁶Interestingly enough, the determination of the CI of the project when it is represented by its AoN is still an open problem.

⁷This is a necessary and sufficient condition.

issue then reduces to determining the minimal node cover of $C(D)$, which would identify the minimal arc fixings that would nullify the presence of the IG's, in the sense of rendering D series-parallel.

The following describes the procedure for achieving this objective. But first we make the following two observations:

Observation 1. Fixing an arc which is the unique arc into a node (out of a node) is equivalent to *reducing* (ie, eliminating, deleting) the node in the *dag* and replacing the linkage at the node with an arc that joins the preceding (succeeding) node with the succeeding (preceding) nodes directly; see Fig.4.

Figure 4. A node reduction.

- (a) The original irreducible *dag*, the IG.
- (b) Reducing node 2.
- (c) Reducing node 3.

Consider a node of in-degree (out-degree) 1. Once the arc going into (out of) the node is fixed, it may be “merged” with the arcs going out of (into) the node with no loss of information⁸, and the node disappears. Therefore one may speak of *arc fixing* and *node reduction* interchangeably.

Observation 2. Minimal node reduction is concerned only with nodes of either in-degree 1 or out-degree 1 in the *dag*.

Reducing a node of in- and out-degrees > 1 must involve the fixing of at least two arcs, which cannot be minimal. Note that, because of the topological numbering of nodes, in any $1, n$ -*dag* that is the AoA representation of a project we are always assured of at least two nodes satisfying the condition; in particular, node 2 which is of in-degree 1, and node $n-1$ which is of out-degree 1. Reducing a node of in- or out-degree 1 may lead to other nodes becoming of in- or out-degree 1, which are then eligible for reduction; etc.

⁸The process of “merging” the fixed activity with other activities depends on the objective of analysis. In the estimation of the *pdf* of the project duration, a constant is added to each other activity incident on the node. In the case of optimal resource allocation, the amount of resource allocated to the fixed activity is deducted from the total available resource, and the duration of the fixed activity is set at the value corresponding to that allocation.

There are three steps to the determination of the complexity index CI ; they are:

- (i) construct the *dominator tree* ($\vec{T}(D)$) and the *reverse dominator tree* ($\overleftarrow{T}(D)$),
- (ii) construct the complexity graph $C(D)$ of the $1, n$ -dag; and finally,
- (iii) determine the “minimum node cover” of $C(D)$ which identifies the (minimal) activities to “fix”. This is the CI of the *dag*.

This construction has two interesting and important properties: first, the set of activities to be fixed is *not unique*; a consequence of the process by which these activities are determined; and, second, these activities may be *compound activities*, in the sense of being a subgraph of the original activities of the project, some of which may have been already “fixed”; a consequence of the fact that reducing a node may cause other nodes to become of in- or out-degree 1.

The theory of the determination of the CI is due to Bein, Kamburowski & Stallmann [8]. Aho *et al* [3] describe a polynomial procedure to compute the dominator tree and the reverse dominator tree for a rooted *dag* with m edges.

7. AN ADDENDUM TO PROJECT REPRESENTATION IN THE AoA MODE: THE MINIMIZATION OF THE COMPLEXITY INDEX.

Section 6 has argued for the need to identify the minimal number of arcs to *fix* (or equivalently, the minimal number of nodes to *reduce*) in order to achieve the ultimate economy in the computational effort required to resolve many problems in ANs (as well as other areas of operations research and computer science). In response to this need, a new measure was established, *viz.*, the complexity index CI , as the measure of that effort. In some sense, the CI measures the deviation of the network from the series-parallel structure. It has been remarked then (see footnote 6) that the determination of the CI of the AoN representation of a project is still an open problem. The implications of this remark are significant, since it means that we must always represent the project in the AoA mode, which is not unique - the same project may have several AoA representations that differ in the number of nodes and number of arcs!

Section 3 discussed four realizable criteria which may be adopted in the translation from the AoN representation to the AoA representation, the last one of which was the “minimization of the resulting network complexity”. However, discussion of that criterion had to be delayed until we have laid the groundwork and the necessary “machinery” required for its understanding. This we have just accomplished, and it is time to address it.

We remark that the AoA representation with the *minimal complexity index* may be different from the representation with the *minimal number of arcs conditional upon the minimal number of nodes*. If one is interested in using the AoA representation in any of the operations research or computer science applications mentioned above, one is then interested in minimizing the CI of the resulting network. Can this be achieved? The answer is yes; and it turns out that a simplified version of the construction described in §6 serves this end quite well.

The procedure is explained in the report of Elmaghraby *et al* [30]. The proof that the minimal complexity can be thus secured is due to Michael [48].

8. GENERALIZED PRECEDENCE RELATIONS (GPRs): REPRESENTATION AND TEMPORAL ANALYSIS

The (now classical) CPM approach introduced two basic concepts, among several, into the methodology for the planning and control of large scale projects. The first is the concept of *precedence* which reflects the partial ordering that exists among the activities of the project, due to technical or other reasons. The second is the concept of *time-cost trade-off* between the duration of an activity and its cost. Implicit in such a trade-off is the assumption that an activity may be shortened from its “normal” (and typically most economical) duration, at a price.

The precedence relation suggested by the CPM model⁹ between two activities, say $h \prec k$, may be characterized as *strict precedence* because it implies that activity h must be *completed* before activity k can be *initiated*. Slight reflection reveals that precedence relations among activities need not be confined to the straitjacket of such strict precedence since other relations can, and do in fact, exist between activities. We shall refer to these latter as *generalized precedence relations (GPRs)*.

We assume that every activity is *indivisible*, and hence must be treated in its entirety as one entity with its “*start*”- and “*finish*”-times. It is then natural

⁹We use “CPM model” as shorthand for the network representation in the CPM and PERT models.

to conceive of four possible relations that constrain the start- and finish-times of two activities by specified “*lead/lags*” as follows: the start-to-start (SS), the finish-to-finish (FF), the start-to-finish (SF), and the finish-to-start (FS) relations.

GPRs afford us the flexibility of modeling relations that are present among activities in many practical situations. These relations cannot be accommodated in the standard CPM model. But the increased flexibility is gained at a price: greater care must be taken in the analysis of the resulting model lest one be led to erroneous conclusions. Additionally, the concept of *criticality* of an activity takes on a new meaning, since now an activity need not be *on* the CP to be declared “critical”; it suffices that either of its defining nodes be on the CP! Furthermore, the concept of activity *float* takes on a new meaning, necessitating the definition of a new concept of *flexibility*, since activities may be “compressed” or “expanded” from their “normal” durations. All this lead, in turn, to additional issues in cost minimization, which are discussed in §10.3.2 below.

The introduction of GPRs adds several elements of complexity to the *modeling* as well as the *analysis* of deterministic ANs that are absent under strict precedence, mainly due to the presence of *cycles*, which were prohibited in classical CPM analysis. First, there is need for a new and consistent notation. Second, GPRs introduce new elements from a conceptual modeling points of view. Third, it presents new issues of *temporal analysis and feasibility*, and of the conversion of GPRs into more familiar relations if such conversion is desired.

We recognize that some managers may not be receptive to GPRs that include the SS, FF, SF, BF and SE relations because of their unfamiliar nature.¹⁰ The question that arises may be phrased as follows: can these relations be “converted” into the more familiar FS-relations? The answer is yes *in the case of fixed activity durations*, see Fig.5 for illustration.

Figure 5. The translation of SS, SF, FF relations
into the more familiar FS relations.

It should be stressed here that the above conversion to the FS-relation does not eliminate the existence of cycles or negative lengths of arcs representing the GPRs,

¹⁰Actually, the complete representation of *GPRs* requires the definition of a new set of precedence relations, denoted by *SE* (for early start) and *BF* (for before finish) which are now necessitated by the need to represent *availability* and *completion* restrictions.

whenever such cycles and lengths exist in the original network. Moreover, the resultant network may possess arcs in parallel; *ie*, the network may be a multi-digraph.

8.1. Anomalies

Classical CPM is free of anomalies: results conform to expectations. Such is no longer the case under *GPRs*. There are two manifestations of such anomalies. The first occurs when *prolonging* (*shortening*) an activity results in the *reduction* (*increase*) in project completion! This anomaly is a characteristic of the *GPRs*, not of their *representation*. To bring this point home, consider the following example: there are three-tasks, *a, b, c* which can be accomplished by different processors in parallel, but they must respect the *GPRs* as specified:

$$s(b) \geq s(a) + 1; f(b) \geq f(a); s(c) \geq s(b) + 1; f(c) \geq f(b)$$

Task	Duration
a	5
b	3
c	4

The network representation is shown in Fig.6a, from which it can be easily seen that the earliest completion time of all three tasks is 7. Now suppose that the duration of task *b* is prolonged to 4. The result is the network of Fig.4c, from which it is seen that the earliest completion time of all tasks is only 6! The reason for this anomaly can be gleaned from the Gantt charts for both durations; these are given in Fig.6b,d; respectively: The prolongation of the duration of task *b* permitted the *earlier* start of task *c* and, consequently, its earlier completion. You should satisfy yourself that *shortening* the duration of task *b* to 2, say, would *delay* the completion time of all three tasks to 8.

Figure 6. An example of the first anomaly.

The second anomaly occurs when *diminishing* the duration of an activity results in *infeasibility* of the AN! We see a manifestation of this anomaly in the network example of Fig.7, where the feasibility of the network shall be destroyed

by shortening the duration of non-critical activity 1 (cycle 1,5,6,2,1 shall have positive length in both instances, indicating that event i , $i = 1, 2, 5, 6$, must be realized before it is realized!). Note that both activities are backward-inflexible. In fact, any manipulation of the activity durations is meaningful only in the context of cost minimization, which is treated in §10.3.2.

Figure 7. Illustration of the second anomaly.

The discussion of GPRs may be found in the paper by Elmaghraby & Kam-burowski [28] which, in turn, may be viewed as an extension of that of Kerbosh & Schell [?], who based their discussion on the original work of Roy [56].

An approach towards expanding the strict precedence of the CPM/PERT models is present in the “CPM Procedure” of SAS/OR [57] under the title “Nonstandard Precedence Relations”. The SAS development permits only lower bounds on the specified lags. (GPRs permit lower *and* upper bounds.) In the SAS approach the four relations are handled by introducing dummy activities of specified durations, some of which may be negative.

Though activity nets with GPRs have been traditionally associated with *projects*, it should be remarked that the resulting model has generality well beyond project work. For instance, such networks constitute an excellent representation of scheduling problems associated with automatic guided vehicles (AGVs); in the representation of materials handling activities; and in the representation of concurrence and parallelism in the design and analysis of computer systems; see the technical report of Elmaghraby & Kam-burowski [27].

9. PROJECT PLANNING UNDER CONSTRAINED RESOURCES

Considerations of resource availabilities have always occupied center stage in project planning and its *dynamic* control. For the sake of clarity of thought we distinguish between two eventualities: (i) It may be that the resources available are limited with no prospect of (or no desire for) increasing their availabilities; which is the case for specialized equipment or labor skills, financial credit limits, storage

space; *etc.*, and it is desired to complete the project as early as possible within the confines of these limitations; or (ii) It may be that management has a free hand, more or less, to acquire any amounts of the necessary resources *at a price* (buy, rent, or lease more equipment and space; subcontract portions of the project; hire new personnel, *etc.*), and the issue is the determination of the optimal quantities to be acquired of each resource to achieve a given target date for completing the project. It is easily seen that real life projects usually present a mixture of these two “pure” extremes, in the sense that the availabilities of some resources are considered fixed while the availabilities of others are considered flexible.

In all circumstances one is faced with selecting the *criterion* against which the planning is conducted. We have already mentioned one such criterion; *viz.*, the *minimization of the project duration*. Others are also possible, such as: the *smoothing of resource usage*, the *minimization of the maximum resource utilization*, the *minimization of the cost of resource usage*, the *maximization of the present value of the project*; *etc.*

Consideration of limited resources plays havoc with some of the very basic concepts that have become household terms in project planning and control. For instance, the concept of *critical path* as a chain of activities whose durations control the total duration of the project is no longer valid! The same is true for the concepts of *event slacks* and *activity floats* (all four of them). For examples of the inapplicability of these concepts under considerations of limited resources, see Elmaghraby [21] pages 145-149.

The analysis and optimization of projects under constrained resources is “difficult” in a purely technical sense as well as in a managerial sense.

It is universally recognized that the scheduling of activities and the timing of resource acquisition and allocation are among the most important functions of management. Technically speaking, the scheduling of activities related by arbitrary precedence relations subject to resource availabilities is an NP-hard problem [11]. This means that the prospect of finding an algorithm that resolves the problem optimally (under any criterion) and runs in “reasonable” time is almost nil as the size of the project gets large. This explains (and, in the minds of operations researchers, justifies) the use of compu-search approaches (*ie*, heuristics which depend crucially on the availability of the computer, such as Neural Nets, Genetic Algorithms, Tabu Search, Simulated Annealing) and implicit enumeration methods (branch-and-bound approaches), among others.

From management point of view, the problem in its real life manifestations is difficult to *state* precisely for quantitative analysis, and even if it were stated

completely and correctly it is difficult to *model* mathematically, let alone to solve. For a more detailed explanation of the nature of the difficulties alluded to here, see Elmaghraby [21], pages 149-155.

The criterion most widely discussed in the literature is the minimization of the project duration. However, we would be remiss not to mention the recent paper by Deckro *et al* (1991) [16] in which an attempt is made to use the ‘decomposition principle’ of linear programming in the resolution of optimization problems with a more general criterion, such as the minimization of a cost function that is defined on the completion times of the activities. The authors report encouraging computational experience on small problems of a rather specialized structure.

Additionally, the book edited by Slowinski & Weglarz [59] devotes the majority of its chapters (eleven out of twenty) to discussions of project scheduling under constrained resources, to which we draw attention: Part I, chapters 1,3,4,5,6,7; Part II, chapters 4,5,6,7,8. In particular, chapter 5, Part I, is an updated evaluation of the various heuristics that have been proposed in the literature, and chapter 5, Part II, is a summary of a long sequence of articles by Weglarz dealing with the interpretation of project scheduling under limited resources as a *control* problem in continuous time.

Typically, resources are available in one or several units (*e.g.*, one asphalt laying machine, one test bed, eight trucks of varying capacities, seven electricians of varying skills, a warehouse of 125,000 square feet of floor space; *etc.*), or may be acquired at a cost. Mathematical models abound for the minimization of the project duration; see Elmaghraby [21], pages 168-201 for examples. Given the analytical difficulty of a frontal attack on the problem utilizing these models, which are usually integer linear programs in several thousands of variables and constraints, a great deal of effort has been devoted to devising clever branch-and-bound (BaB) approaches. The lineage of such attempts appears to be: Johnson (1967) [40], Schrage (1970) [58], Gorenstein (1972) [35], Patterson & Roth (1976) [54], Stinson, Davis & Khumawala (1978) [62], Talbot & Patterson (1978) [63], Christofides, Alvarez-Valdes & Tamarit (1987) [13], Patterson *et al* (1989) [55], Deckro & Hebert (1992) [17], and Demeulemeester & Herroelen (1992) [18]. In the latter paper the authors present an alternative scheme to the one offered by Talbot & Patterson [63]. (In a privately circulated research report, Demeulemeester & Herroelen (1992) [19] extend their approach to projects specified with generalized precedence relations (GPRs), see §8, and report some computing experience.)

The logic of the BaB-based procedure of Demeulemeester & Herroelen may be summarized as follows. The nodes of the BaB search tree correspond to *tempo-*

rary partial schedules at time m , denoted by $\{PS_m\}$, in which finish times have been assigned temporarily to a subset of the activities. The partial schedules are feasible, in the sense of satisfying both precedence and resource constraints. Time is incremented to the earliest of those time instants m which correspond to the completion time of one or more activities in PS_m . Scheduling decisions are “temporary” in the sense that temporarily scheduled activities may be delayed as a result of decisions made at later stages in the search process. *Unscheduled activities* at time m are those for which the temporary scheduling decisions have not yet been made. At time m the corresponding partial schedule PS_m is composed of two subsets: F_m , the set of finished activities, and S_m , the set of activities in progress; ie, $PS_m = F_m \cup S_m$ and $F_m \cap S_m = \emptyset$. Define U_m as the set of *unfinished activities*, $U_m = A - F_m$. Note that U_m contains the set of activities in progress. Define the *eligible set* E_m at time instant m as the set of activities which are not in the partial schedule PS_m and whose predecessor activities have finished; ie, $E_m = \{i : B(i) \subset PS_m\}$. Activities in E_m can start at time m if the resource constraints are not violated. It is intuitively clear that if at time m the partial schedule PS_m has no activity in progress, ie, $PS_m = F_m$, and an eligible activity i cannot be scheduled together with any other unscheduled activity at any time $m' \geq m$ without violating the precedence or resource constraints, then an optimal continuation of PS_m will contain activity i starting at time m . A similar statement can be made relative to *two activities* which may be scheduled concurrently starting at m , occupy the same length of time, and cannot be run concurrently with any other activity in U_m . Extension to more than two activities is immediate but computationally unwieldy.

If it is not possible to schedule all eligible activities at time m because of resource limitations, a new set of branches in the BaB search tree is initiated. Here the procedure defines a *delaying set* $D(p)$ which consists of all subsets of activities D_q either “in progress”¹¹ or eligible, the delay of which would resolve the current resource conflict at node p of the BaB search tree. Naturally we are interested in *minimal* delaying alternatives, ie, alternatives which do not contain other delaying alternatives as subsets. It must be remarked here that it is oftentimes the case that one cannot enumerate *all* the delaying sets because of the enormity of their number - one must decide beforehand on the maximum number allowed. This may impact the *optimality* of the final schedule achieved at the termination of the search procedure. In the procedure, extra precedence relations among activities

¹¹The words “in progress” are placed in quotation marks to highlight the fact that all schedules are in fact *tentative* and all start and completion times are *temporary*.

are introduced to force the activities in the set D_q to wait until the completion of some other activities. These extra precedence relations, denoted by G_q , are constructed by taking as the predecessor activity for every activity $i \in D_q \subseteq D(p)$ the earliest finishing activity j that is either in progress or eligible to start at time m and that is not delayed, *ie*, $G_q = \{(j, i) : f_j = \min_{k \in PS_m} f_k; i \in D_q\}$. The totality of all schedule-introduced precedence relations (as opposed to the original precedence relations among the activities) at any point of time shall be denoted by H . Ties are broken randomly.

There remains only the determination of the lower bound L_q for each delaying alternative D_q . Based on the added precedence relations for alternative D_q , determine the corresponding partial schedule $PS'_m = PS_m + E_m - D_q$ and perform the precedence-based critical path computations for the network; denote its length by z . All unscheduled activities $i \notin PS'_m$ which are not on this path are inserted into a set NC . The computation of their early start times $\{es_i\}$ and late finish times $\{lf_i\}$ is based solely upon precedence constraints (*ie*, ignoring resource constraints). If there exists an unscheduled activity $i \in NC$ of duration d_i and if sufficient resources of one or more types are only available to allocate to activity i continuously for *at most* $e_i \leq d_i$ time periods somewhere in the interval from es_i to lf_i then the project completion will be delayed by a minimum of $d_i - e_i$ time periods. Thus the critical sequence lower bound is evaluated from

$$L_q = \max_{i \in NC} (z + d_i - e_i)$$

Branching continues from the node corresponding to the delaying alternative D_q^* with the smallest lower bound; ties are broken randomly.

Two dominance rules are used to prune the BaB search tree. The first is based on the so-called *left-shift dominance rule*, and the second is based on the concept of a *cutset*, denoted by C_m , and defined as the set of all unscheduled activities for which all predecessor activities belong to the partial schedule PS_m ; *ie*, $C_m = \{i \in U_m - S_m : B(i) \in PS_m\}$.

The left-shift dominance rule applies whenever an activity can be advanced to an earlier time than its start time in the current (temporary) schedule. This happens when the activity in question was previously delayed but now is eligible (at time m) because one (or more) of the activities that caused it to be delayed (due to resource conflict) has itself been delayed.

The cutset dominance rule is implemented as follows. Consider a cutset C_m at time m which contains the same activities as a cutset C_k that was previously saved during the search of another path in the BaB search tree. If time $k \leq m$ and if

all activities in progress at time k did not finish later than $\max\{m, \text{finish time of the corresponding activities in } PS_m\}$, then the current partial schedule PS_m is dominated. The analysis of PS_m is discontinued.

Demeulemeester & Herroelen report reasonably good computing time on the PC - of the order of a few seconds for projects containing up to 35 activities and 3 resources - which is superior to other BaB-based procedures. Performance of such BaB procedures under more realistic network sizes (of the order of tens of thousands of activities) is currently unknown. We conjecture that optimization of such large scale projects can be achieved with current computer architecture (single processor arithmetic unit) through decomposition of the project into smaller units that are loosely connected. Otherwise, we have to be content with the use of heuristics of undefined error bounds. Heuristics procedures, let alone optimization algorithms, for activity scheduling under the other criteria mentioned above have not been forthcoming, to the best of our knowledge.

10. SOME FINANCIAL CONSIDERATIONS

10.1. THE BIDDING PROBLEM

Most of the literature on project planning and control is concerned with projects that already exist (in the form of a commitment to accomplish a set of 'deliverables' at certain times). Yet most, if not all, projects come into being as a result of a process of *adjudication* in which interested parties are invited to bid on a set of specifications of deliverables. The winner is selected on the basis of technical as well as financial considerations.

The process of bidding is an important phase in the life of a project: it is the process that brings it into being as far as the contractor is concerned. And although bidding and its precursor, cost and resources estimation, have been topics of writing by practitioners for decades, they have been largely overlooked by researchers in ANs. This fact seems odd in view of the fact that the Department of Defense in the USA has demanded the use of the PERT model of the project in any proposal submitted to it since the day of inception of PERT. Other major 'owners' soon followed suit, and today one hardly finds a major project that is not subject to this process. If the contractor must submit in the proposal a network detailing the plan by which s/he intends to integrate the various elements (*ie*, activities) of the project, then it should stand to reason that s/he would use

that information in the preparation of the bid itself. It seems odd indeed that contractors do not choose to do so!

Cost considerations that bear some relation to the issue of bidding have appeared in the writings of Stark [60], who addressed a problem of bidding that is related not to ANs but to the so-called ‘unbalanced’ contract tendering; Badger [6], who discussed a method of cross referencing estimate accounts to network activities to provide compatibility between the two accounting control documents; and Farid & Boyer [31], who indicate that the bid that results should be the total cost of the project multiplied by a ‘fair and reasonable markup’ (FaRM) of the project cost. See also the next discussion of the ‘net present value’ problem for references to prior contributions that bear on the bidding issue.

When faced with preparing a bid to submit to the owner of a project, the contractor must analyze the project specifications. Included in this analysis is the formulation of a project network, and the determination of the resources (and consequently the costs of acquiring and utilizing these resources) that are required to complete it. Typically, the contractor requires an initial payment from the owner before the work on the project begins. The contractor will also define various *key events (KEs)* in the project at which s/he will demand partial payments for work that has been completed. The last event, indicating project completion, is always a KE. The issue then resolves itself into the following consideration. There are two streams of cash flow inherent in any project. they are,

1. *Outflows*: payments made by the contractor in executing the project.
2. *Inflows*: receipts by the contractor at the time of realization of various KEs.

In the CPM model of ANs, where all the activities are assumed of deterministic duration and cost, there is little problem in defining the cost of an activity, and subsequently the cost of the project. The two streams of cash flow can then be easily obtained *given a particular schedule of the activities*. The issue then revolves around the *scheduling* of activities to maximize the net present value of the project, which is the subject of discussion of section 10.2 below. Unfortunately, when one deals with PERT type ANs the realization time of a KE as well as the costs of the activities leading to its realization can be determined only in a probabilistic sense. At the time of bidding the manager of the project must make a decision on the amount of risk s/he wishes to take when *setting a delivery date* and *specifying a desired value* of the KE. These, and related, issues are discussed in a paper by Elmaghraby [22].

10.2. THE NET PRESENT VALUE (NPV) PROBLEM

Issues concerning the “net present value” (NPV) of a project are equally important to those interested in *bidding* on a proposed project to secure it, and those who are *committed* to carry out an already agreed-upon project. It may be succinctly stated as follows.

Given specified ‘net cash flows’ (ncf ’s) at selected key events in the AoA mode of representation of a project, what is the optimal schedule of the realization times of these KE’s in order to maximize the NPV of the project as a whole? Note that the ncf at node i , denoted by a_i , may be positive or negative, reflecting net receipts or net disbursements, respectively. In a typical project, positive and negative ncf ’s are interspersed, with the majority of the earlier cash flows being negative, reflecting outlays by the ‘contractor’ which are not fully recovered by ‘owner’ payments, and the majority of the later ncf ’s being positive, reflecting the recoupment by the contractor of expenditures plus a reasonable profit.

As of the time of this writing, the theoretical treatment of the NPV problem is still in its formative stages. This is because a full treatment requires taking into account the uncertainty in the durations of the activities, which is no minor feat. In addition, one must recognize the interdependence between the ncf at event i , a_i , and its time of realization — in most realistic contracts on projects, one earns a ‘bonus’ for early realization ¹², and pays a ‘penalty’ for late realization of $KE(i)$. Current state of the art of analysis assumes independence of these two variables, which is a gross simplification of reality. Finally, current approaches require knowledge of the ‘discount rate’ β , $0 < \beta < 1$, (which reflects the time value of money). This is feasible in projects of short durations — say, six months or less, but is highly questionable in large scale projects that span a longer period of time of two or more years.

The following simple example illustrates the issues at play and suggests the approach for their resolution. Consider the AoA representation of the (minuscule) project of Fig.8. The number on each arc is the duration of the activity, and the number next to each node is its ncf .

Figure 8. A simple example. The number on the arc represents the activity duration; and the number next to a node represents the ncf at the time of its realization.

¹²In some instances a *penalty* may accrue due to early completion due to inventory or other costs.

A naive approach would evaluate the *CP*¹³, which is path 1-3-4 of duration 11, and try to schedule the activities within this total duration. The only freedom one has is in the time of realization of *KE*(2), denoted by T_2 . Clearly, since $a_2 < 0$ then T_2 should be as large as possible within the specified *CP* length, yielding $T_2 = 3$. Assuming that the project starts at time $T_1 = 0$, and a discount factor $\beta \approx 0.99$ we would have a *NPV* of

$$-5000(.99)^3 + 3000(.99)^8 + 3000(.99)^{11} = 602.75$$

The correct approach proceeds as follows. Since a_3 and a_4 are both > 0 , the realization times T_3 and T_4 should be as early as possible, which results in the tree shown in heavy lines in Fig.9. Respect for the precedence relations results in $T_3 = T_2 + 4$, and $T_4 = T_2 + 8$. The criterion function may then be written as

$$-5000\beta^{T_2} + 3000\beta^{T_2+4} + 3000\beta^{T_2+8} = \beta^{T_2}[-5000 + 3000\beta^4 + 3000\beta^8]$$

which is to be maximized. Since the multiplier of β^{T_2} is a positive constant equal to 650.02, the value of the objective function is maximized if we put T_2 as small as possible within the defined tree; ie, put $T_2 = 4$, yielding a total value of 637.08, which is larger than the value obtained above!

Two pertinent remarks should be made relative to this result. First, it was arrived at by an argument that reached the *opposite* conclusion from the naive approach: T_2 is put as small as possible within the defined tree, not as large as possible within the specified *CP* length! Second, the optimal project duration is 12, which is longer than the *CP* duration! Here is one more instance in which the conventional wisdom on the meaning of the *CP* is shattered.

Evidently, what is needed is a formal procedure that encapsulates the essential features of the argument that led to the above result. Such a procedure was specified by Elmaghraby & Herroelen [26] (who also give a comprehensive analysis of past contributions), and implemented by Herroelen & Gallens [37]. The latter paper also reports on computational experience with the algorithm.

More recently, interest has focused on resolving the ‘payment scheduling problem’ in which the amount *and* timing of progress payments are determined *simultaneously*. In a paper by Dayanand & Padman [14] the assumption is made that

¹³We use the notation ‘CP’ here to denote its customary use in CPM literature as the ‘critical path’ in the network. See below for the definition of the ‘cp’, which is a different use of ‘critical path’.

the timing and magnitude of costs are known, the issue is to determine the optimal payment schedule to maximize the npv of the project (from the contractor's point of view). An ILP model is presented which is rather difficult to solve due to the large number of binary decision variables involved. It is supplemented by a host of heuristics to accommodate large scale projects. A contribution by Sepil & Kazaz [60] treats the *NPV* problem under the assumption that costs are incurred at the termination of the activity but income payments are realized at regular intervals of time. This would be the case, for example, of a 'general contractor' who is paid by the owner at the end of each month for work accomplished during the month, but the activities are farmed out to subcontractors who are paid by the general contractor when the activity (for which each subcontractor is responsible) is completed. Sepil & Kazaz introduce the concept of *activity profit curve* which relates the net present value of the activity to its time of completion. They approximate the nonlinear curve with piecewise linear segments, and present an ILP for the maximization of the project npv. Despite the great economy in the number of integer variables achieved by the introduction of the activity profit curves the model is still too demanding in computing time. Further development of this concept may lead to practical results.

10.3. THE OPTIMAL PROJECT COMPRESSION PROBLEM

10.3.1. PROJECT COMPRESSION UNDER STANDARD PRECEDENCE RELATIONS

A perennial problem faced by managers of large scale projects is how to shorten the duration of a project most economically. We call this the 'optimal project compression problem'.

It is commonly recognized that most activities encountered in real life projects can be accomplished in shorter or longer time by increasing or decreasing the resources available to them. Naturally, acceleration of operations entails additional costs. Such action would be rational only if these additional costs are more than offset by the anticipated rewards from the completion of the project at an earlier date. Conversely, a more leisurely pace implies reduced costs of the activities, but probably a higher penalty for late termination. Consequently, it is meaningful to inquire into the utility of such trade-off between duration and cost, and most studies in this area give form to such inquiry.

It is assumed that the complex interaction between the required resources and duration of the activity can be summarized in a single functional relationship

between cost and duration. In other words, it is assumed that cost reflects the aggregate utilization of the requisite resources *in the most efficient manner* to complete the activity in the specified time. The determination of this optimal combination of the resources at various activity durations is not the subject of discussion - it is assumed known, as is its cost. Herein lies what may be the most crucial assumption of all OR technology that is applied to this problem. From a managerial point of view, this assumption may be 'reasonable' or 'unreasonable' depending on the activity and on management's past experience. The saving grace in what may seem at first blush as demanding too much from the practitioner is that the precise determination of parameters is not needed - only approximation to relative magnitudes and functional relations.

Management is interested in several issues, which are not unrelated, among which we enumerate three:

1. How much *time should be allowed each activity* to complete the project at a certain target date with minimum cost?
2. Given a fixed amount of investment capital to spend, what is the *shortest duration possible* of the project?
3. For any specified duration of the project, what is the *marginal cost of reduction* in that duration?

In a more technical vein, the assumption of a time-cost trade-off is translated into a non-increasing function $c_a = \phi(y_a)$, where y_a is the duration of activity a and c_a is its cost. The approach to the resolution of the issues raised above is dependent on the form of the function ϕ ; the treatment of the linear, convex, concave, and discrete functions may be found in Elmaghraby, pp. 61-119 [22]. Figure 8 illustrates the case of linear, concave, and discrete ϕ 's.

Figure 9. Three possible forms of ϕ
 (a) Linear; (b) Concave; (c) Discrete.

When the cost relations of the activities are linear and there are rewards for the early realization of certain KE (the rewards may be negative, indicating penalties for late realization) - also assumed linear in the difference between a target

date and the actual time of realization – the issues cited above were treated by Elmaghraby & Pulat [24]. To the best of our knowledge, none of the approaches that have been proposed in the literature for the resolution of the problem of optimal project compression has been implemented in any of the commercially available software. This reflects a gap between theory and practice, which should be eliminated.

10.3.2. PROJECT COMPRESSION UNDER GPR's

GPR's were introduced in §8. Recall that we assumed the duration of activity a , denoted by y_a , to be bound from above and below as follows: $0 < \ell_a \leq y_a \leq u_a < \infty$. We now assume that we are also given a time-cost function $g_a(y_a)$ that is piecewise linear and convex over the interval $[\ell_a, u_a]$. Note that we do not assume g_a to be nonincreasing, as is commonly done in standard CPM analysis. In fact, we shall assume that it is piecewise linear and decreasing to a point (the so-called *normal* activity duration), after which it is increasing; see Fig.10.

Figure 10. A piecewise linear $g_a(y_a)$.

In addition to the three problems mentioned in §10.3.1, three other problems immediately come to mind. (i) Suppose there is imposed a *due date* by which time the project should be completed; what are the optimal durations of the activities that satisfy this requirement at minimum total project cost and simultaneously respect the GPRs? (ii) Determine, in an efficient manner, the complete optimal project duration-cost function in the interval of its possible durations: (iii) Suppose that the due date is to be interpreted as a *target date* with rewards for early, and penalties for late completion of the project; what are the optimal activity durations?

The first two problems have their counterparts in standard CPM analysis cited above. However, their analysis under GPRs is more complex. For one, while the limits of the duration of the project are easily derived in the case of standard CPM, it is not immediately obvious what these limits are under GPRs, mainly due to considerations of infeasibility. And for another, in standard CPM a prolongation (reduction) in the duration of an activity could only lead to a delayed (advanced) completion of the project, or, at best, leaving it unchanged. Under GPRs this need not be true; see “Anomalies” in our discussion of representation of GPRs

(§8.1). Consequently, the individual activity time-cost function must now allow for the *prolongation* of the activity, in addition to its reduction, an element that was absent in standard CPM analysis.

As illustration of the difficulty in the analysis introduced by the introduction of GPRs, we address the problem of determining the optimal project duration-cost function $\phi^*(\lambda)$ where λ is in the interval $[\underline{\lambda}, \bar{\lambda}]$, where $\underline{\lambda}$ denotes the minimum project duration, and $\bar{\lambda}$ denotes the least-cost project duration. As to be expected the project duration-cost function is decreasing, piecewise-linear and convex in this interval. The identification of the whole project duration-cost function also permits us to solve the problem of finding the project schedule with the minimal duration when its total cost cannot exceed a given value.

In standard CPM the identification of the project cost curve starts from the point corresponding to the upper bound $\bar{\lambda}$, which is easily determined by simply assuming that all activities are realized at their “normal”; *ie, least cost*, durations. Then we construct the optimal project cost curve for successively shorter durations until the project duration cannot be diminished any further. Such an approach, however, may fail in ANs with GPRs because the network with the “normal” (*ie, least cost*) activity durations may be inconsistent with the prescribed precedence relations, (hence there is no feasible schedule!). It is then apparent that the problem of finding a starting point on the optimal project duration-cost function becomes crucial in identifying this curve. For this reason Elmaghraby & Kamбуrowski [28] proposed another approach, whose general scheme is as follows:

- Step 1. Use the modified Bellman algorithm [9] to find the earliest project schedule and the minimum project duration $\underline{\lambda}$.
- Step 2. Use Orlin’s algorithm [53] to find the cheapest project schedule for $\lambda = \underline{\lambda}$.
- Step 3. Starting from the point on the project cost curve found in Step 2 proceed, iteratively, to determine the optimal cost function for increasing project durations until no further decrease in the project cost is observed.

The algorithm presented in outline form above, and the complete resolution of the three problems stated at the outset, may be found in the paper of Elmaghraby & Kamбуrowski [28].

References

- [1] Adlakha, V.G. and Kulkarni, V.G. (1989). "A classified bibliography of research on stochastic PERT networks: 1966-1987," *INFOR* 27, 272-296.
- [2] Agrawal, M., Elmaghraby, S.E. and Herroelen, W., (1994). "On the generation of test sets for activity nets," *O.R. Report No. 281, North Carolina State University, Raleigh, NC 27695-7913*.
- [3] Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (1975). *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA.
- [4] Archibald, R.D. (1987). "The history of modern project management," *Project Management J.* 17(4), 29-31.
- [5] Assad, A. A. and Wasil, E.A. (1986). "Project management using a micro-computer," *Computers and Operations Research* 13, 231-260.
- [6] Badger, A.A. (1974). "Interrelaing project estimates to CPM schedules," *AACE Trans.* 18, 38-42.
- [7] Battersby, A. (1970). *Network Analysis for Planning and Scheduling*, 3rd ed., John Wiley & Sons, N.Y., NY.
- [8] Bein, W.W., Kamburowski, J., and Stallmann, M.F.M. (1992). "Optimal reduction of two-terminal directed acyclic graphs," *SIAM J. Comp.* 21, 1112-1129.
- [9] Bellman, R.E. (1958). "On a routing problem", *Quart. Appl. Math.* 16, 87-90.
- [10] Bigelow, C.G. (1962). "Bibliography on project planning and control by network analysis: 1959-1961," *Oper. Res.* 10, 728-731.
- [11] Blazewicz, J., Lenstra, J.K., and Rinnooy Kan, A.H.G. (1983). "Scheduling projects to resource constraints: Classification and complexity," *Discrete Appl. Math.* 5, 11-24.
- [12] Cantor, D.G. and Dimsdale, B. (1969). "On direction-preserving maps of graphs," *J. Combinatorial Theory* 6, 165-176.

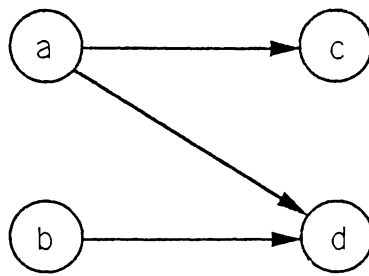
- [13] Christofides, N.R., Alvarez-Valdes, R., and Tamarit, J.M. (1987). "Project scheduling with resource constraints: A branch-and-bound approach," *Euro. J. Operl. Res.* 262-273.
- [14] Dayanand, N. and Padman, R. (1993). "Payments in projects: A contractor's model," *Res. Report*, The H. John Heinz III School of Public Policy and Managemen, Carnegie Mellon University, Pittsburgh PA 15213.
- [15] De Reyck, B. and Herroelen, W.S. (1993). "On the use of the complexity index as a measure of complexity in activity networks," *Research Report No. 9332*, Department of Applied Economics, Katholieke Universiteit Leuven, Huis Renaer. Debériotstraat 36, B-3000 Leuven, Belgium.
- [16] De Wit, J. and Herroelen, W.S. (1990). "An evaluation of microcomputer-based software packages for project management," *Euro. J. Operl. Res.* 49, 102-139.
- [17] Deckro, R.F., Winkofsky, E.P., Hebert, J.E., and Gagnon, R. (1991). "A decomposition approach to multi-project scheduling," *Euro. J. Operl. Res.* 51, 110-118.
- [18] ——— and Hebert, J.E. (1992). "Resource constrained project crashing," *Omega* 17, 106-111.
- [19] Demeulemeester, E. and Herroelen, W.S. (1992). "A branch-and-bound procedure for the multiple resource-constrained project scheduling problem," *Management Sci.* 38, 1803-1818.
- [20] ——— and ——— (1992). "A branch-and-bound procedure for the generalized resource-constrained project scheduling problem," *Res. Report 9206*, *Department of Applied Economic Sciences*, Katholieke Universiteit Leuven, Leuven, Belgium.
- [21] Edwards, K. *et al.* (1984). "Project management with the PC: Part I and Part II," *PC Magazine* 3 (21), 109-156 and (24) 193-277.
- [22] Elmaghraby, S.E. (1977). *Activity Networks: Project Planning and Control by Network Models*, John Wiley & Sons, N.Y., NY.
- [23] ——— (1990). "Project bidding under deterministic and probabilistic activity durations," *Euro. J. Operl Res.* 49, 14-34.

- [24] ——— and Pulat, P.S. (1979). "Optimal project compression with due-dated events,"
- [25] ——— and Herroelen, W.S. (1980). "On the measurement of complexity in activity networks," *Euro. J. Operl Res.* 5, 223-234.
- [26] ——— and ——— (1990). "The scheduling of activities to maximize the net present value of projects," *Euro. J. Operl Res.* 49, 35-49.
- [27] ——— and Kamburowski, J. (1990). "On project representation and activity floats," *Arabian J. Sci. and Eng.* 15, 627-637.
- [28] ——— and ———. (1989). "The analysis of activity networks under generalized precedence relations (GPR), Parts I and II," OR Reports No.231 and 232, NC State University, Raleigh NC, 27695-7913.
- [29] ——— and ——— (1992). "The analysis of activity networks under generalized precedence relations (GPRs), *Management Sci.* 38, 1245-1263.
- [30] ——— and ——— (1990). "On project representation and activity floats," *Arabian J. Sci. & Eng.* 15, 627-637
- [31] ———, Kamburowski, J., Michael, D.J. and Stallmann, M.F.M. (1993). "Minimizing the complexity of a stochastic PERT network," *OR Report No.* , North Carolina State University, Raleigh NC 27695-7913.
- [32] Farid, F. and Boyer, L.T. (1985). "Fair and reasonable markup (FaRM) pricing model," *J. Construction Eng. & Management* 111/4. 374-390.
- [33] Ford, L. and Fulkerson, D. (1962). *Flows in Networks*, Princeton University Press, Princeton, NJ.
- [34] Fresko-Weiss, H. (1989). "High-End project managers make the plans," *PC Magazine* 8 (9), 155-195.
- [35] Fulkerson, D.R. (1961). "A network flow computation for project cost curve", *Management Sci.* 7, 167-178.
- [36] Gorenstein, S. (1972). "An algorithm for project sequencing with resource constraints," *Oper. Res. Qaurt.* 23, 261-275.

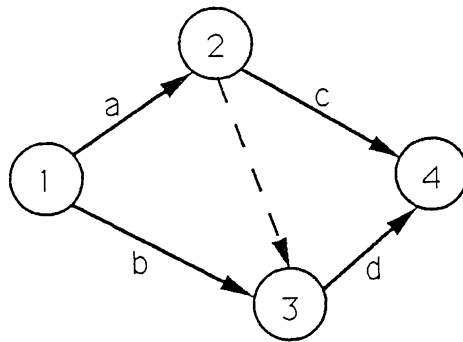
- [37] Herroelen, W.S. and Gallens, E. (1993). "Computational experience with an optimal procedure for the scheduling of activities to maximize the net present value of projects," *Euro. J. Operl Res.* 65, 274-277.
- [38] Hogan, T. et al. (1985). "Project planning programs put to the test," *Business Software* 3(3), 21-56.
- [39] Hopcroft, J. and Karp, R. (1973). "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM J. Comput.*, 225-231.
- [40] Jensen, P.A. and Barnes, J.W. (1980). *Network Flow Programming*, Wiley, NY.
- [41] Johnson, T.J.R. (1967). "An algorithm for the resource-constrained project scheduling problem," unpublished Ph.D. dissertaion, MIT.
- [42] Kelley, J.E. Jr. (1961). "Critical path planning and scheduling - Mathematical basis," *Oper. Res.* 9, 296-320.
- [43] Kerbosh, J.A.G.M. and Schell, H.J. (1975). "Network planning by the extended METRA Potential Method", Report KS-1.1, University of Technology Eindhoven, Department of Industrial Engineering, Netherlands.
- [44] Kerzner, H. (1979). *Project Management: A Systems Approach to Planning, Scheduling and Control*, Van Nostrand Reinhold, New York.
- [45] Kolisch, R., Sprecher, A. and Drexl, A. (1992). "Characterization and generation of a general class of resource-constrained project scheduling problems: Easy and hard instances," *Research Report N0301*, Institut für Betriebswirtschaftslehre, Christian-Albrechts Universität zu Kiel.
- [46] Krishnamoorthy, M.S. and Deo, N. (1979). "Complexity of the minimum-dummy-activities problem in a PERT network," *Networks* 9, 189-194.
- [47] Lawler, E.L. (1976). *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehard and Winston, NY.
- [48] Lerda-Olberg, S. (1966). "Bibliography on network-based project planning and control techniques: 1962-1965," *Oper. Res.* 14, 925-931.

- [49] Michael, D.J. (1991). "Optimal representation of activity networks as directed acyclic graphs," *Ph.D. Thesis*, Program in Operations Research, North Carolina State University, Raleigh NC 27695-7913.
- [50] ———, Kamburowski, J. and Stallmann, M. (1993). "On the minimum dummy-arc problem," *RAIRO Rech. Oper.* 27, 153-168.
- [51] Moder, J.J., Phillips, C.R., and Davis, E.L., (1983). *Project Management with CPM and PERT*, 3rd ed., Van Nostrand-Reinhold, N.Y., NY.
- [52] Nemhauser, G.L. and Wolsey, L.A. (1988). *Integer and Combinatorial Optimization*, Wiley-Interscience.
- [53] Orlin, J.B. (1988). "A faster strongly polynomial minimum cost flow algorithm," *Proc. 20th ACM Symp. on the Theory of Comp.*, 377-387.
- [54] Patterson, J.H. (1983). "Exact and heuristic solution procedures for the constrained-resource, project scheduling problem: Volumes I, II, and III," *Research Monograph*, privately circulated.
- [55] ——— and Roth, G. (1976). "Scheduling a project under multiple resource constraints: A zero-one programming approach," *IIE Trans.* 8, 449-455.
- [56] ———, Slowinski, P., Talbot, F.B., and Weglarz, J. (1989). "An algorithm for a general class of precedence and resource constrained scheduling problems," *Chapter 1 in Advances in Project Scheduling*, R. Slowinski and J. Weglarz, eds., Elsevier.
- [57] Roy, B. (1962). "Graphes et ordonnancements", *Revue Francaise de Recherche Operationelle* 25, 323-326.
- [58] SAS/OR User's Guide, Version 6, First Edition (1989). SAS Institute Inc., SAS Circle, Box 8000, Cary, NC 27512-8000.
- [59] Schrage, L. (1970). "Solving resource-constrained network problems by implicit enumeration nonpreemptive case," *Oper. Res.* 18, 225-235.
- [60] Sepil, C. and Kazaz, B. (1994). "Project scheduling with discounted cash flows and progress payments," *Technical Report*, privately circulated.
- [61] Slowinski, R. and Weglarz, J., eds. (1989). *Advances in Project Scheduling*, Elsevier Science Publishers B.V., Amsterdam.

- [62] Stark, R.M. (1974). "Unbalanced highway contract tendering," *Operl Res. Quart.* 25, 373-388.
- [63] Sterboul, F. and Wertheimer, D. (1980). "Comment construire un graphe PERT minimal," *RAIRO Rech. Oper.* 14, 85-98.
- [64] Stinson, J.P., Davis, E.W., and Khumawala, B.M. (1978). "Multiple resource-constrained scheduling using branch and bound," *AIIE Trans.* 10, 252-259.
- [65] Talbot, B.F. and Patterson, J.H. (1978). "An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems," *Management Sci.* 24, 1163-1174.
- [66] ——— (1984). "Optimal constructions of reversible digraphs," *Discrete Appl. Math.* 7, 209-220.
- [67] Valdes, J., Tarjan, R., and Lawler, E. (1982). "The recognition of series-parallel digraphs," *SIAM J. Comput.* 11, 298-313.
- [68] Warren, Thomas (1969). "Four float measures for critical path scheduling", *J. Ind. Eng.* 10, 19-23.
- [69] Whitehouse, G.E. (1973). *Systems Analysis and Design Using Network Techniques*, Prentice-Hall, Englewood Cliffs, NJ.

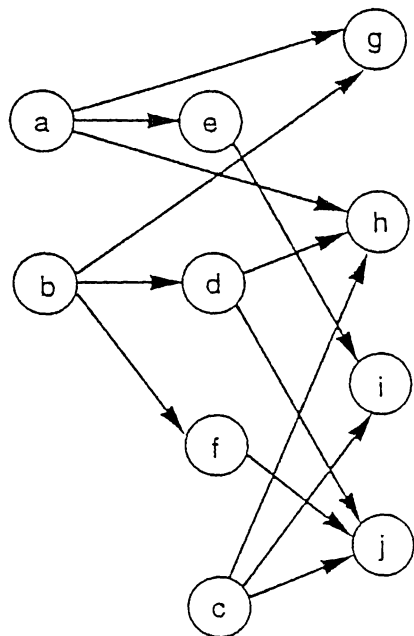


(a)

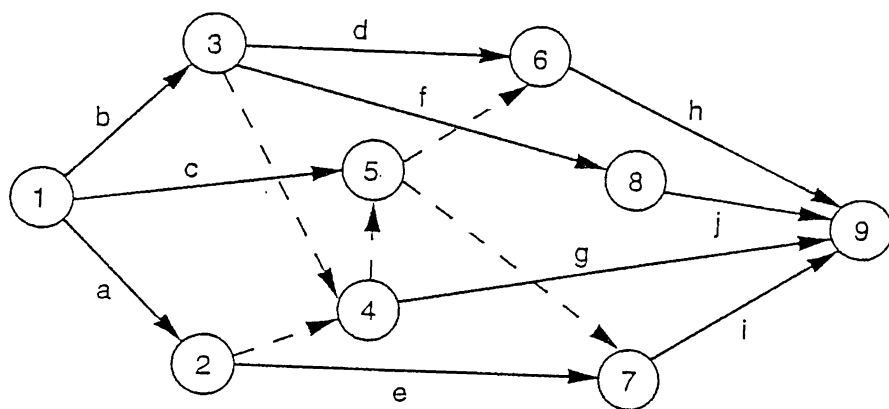


(b)

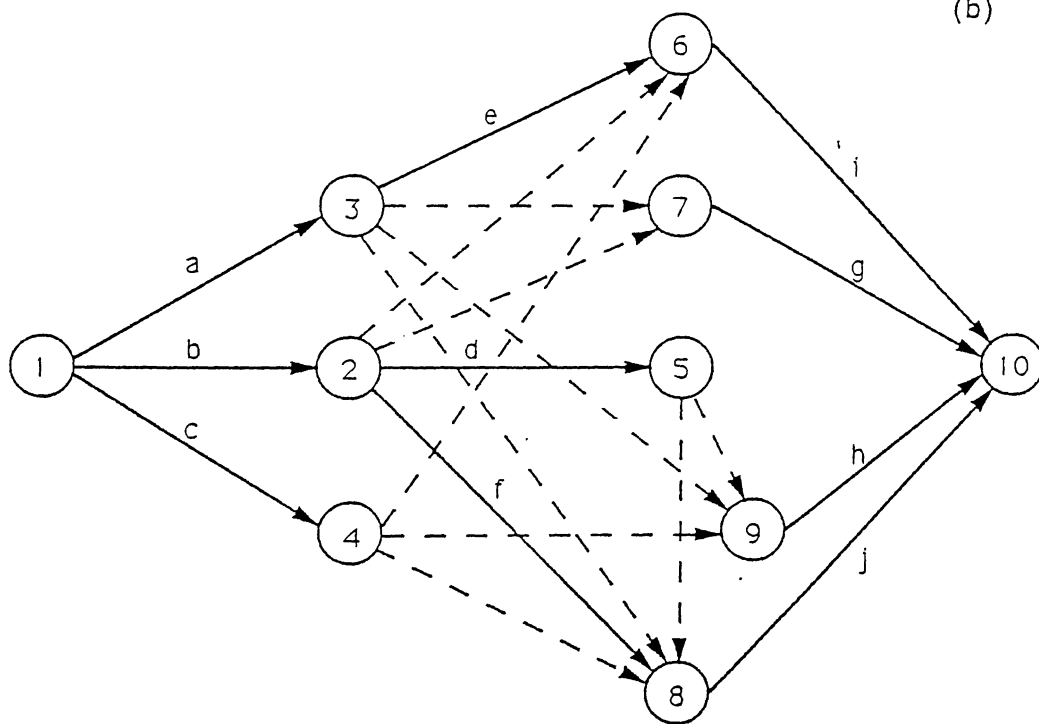
Figure 1



(a)



(b)



(c)

Figure 2

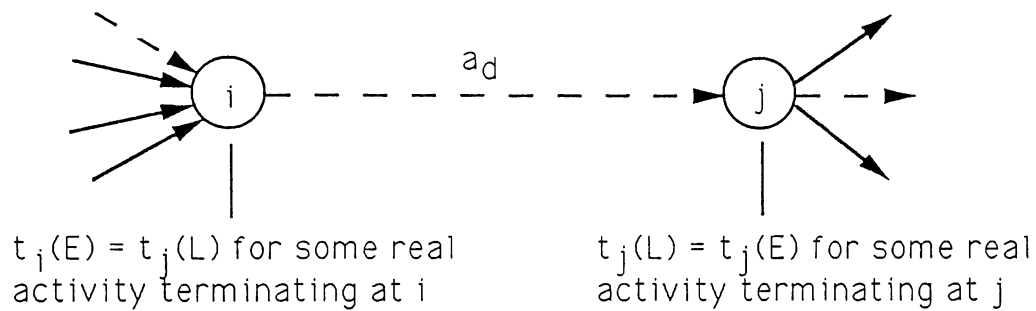
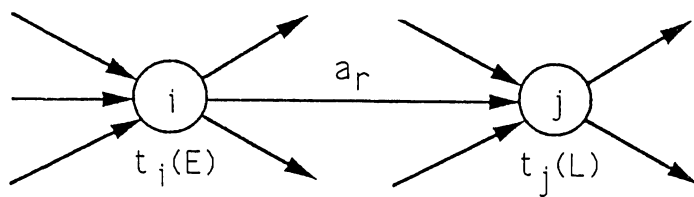
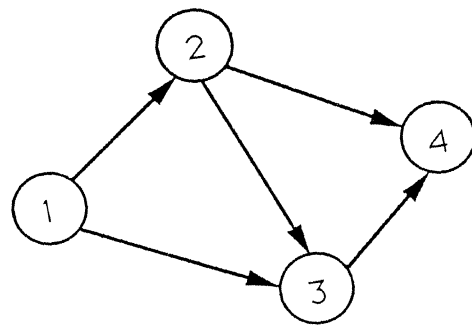
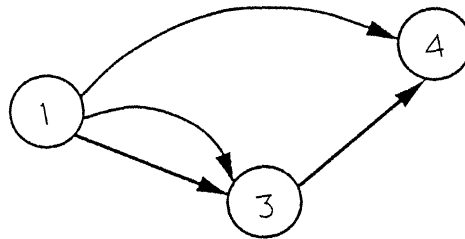


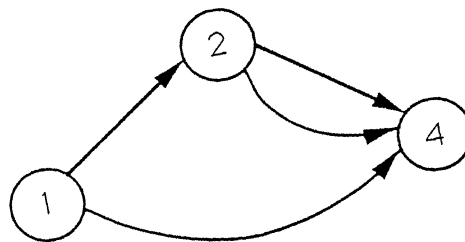
Figure 3



(a)

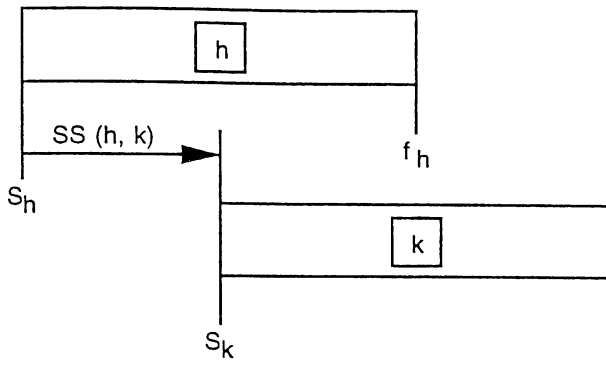


(b)



(c)

Figure 4



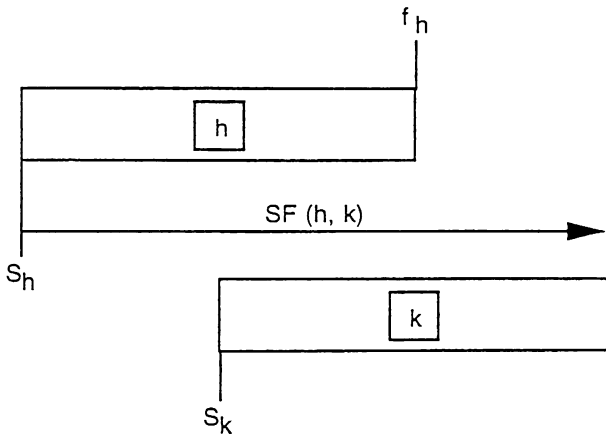
(a)

$$f_h = S_h + d_h$$

$$S_k \geq S_h + SS(h, k)$$

$$\geq f_h - d_h + SS(h, k)$$

$$FS(h, k) = SS(h, k) - d_h$$



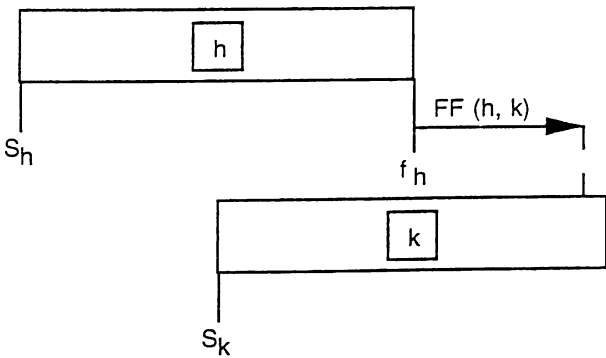
(b)

$$f_h = S_h + d_h$$

$$S_k \geq S_h + SF(h, k) - d_k$$

$$\geq f_h - d_h + SF(h, k) - d_k$$

$$FS(h, k) = SF(h, k) - d_h - d_k$$

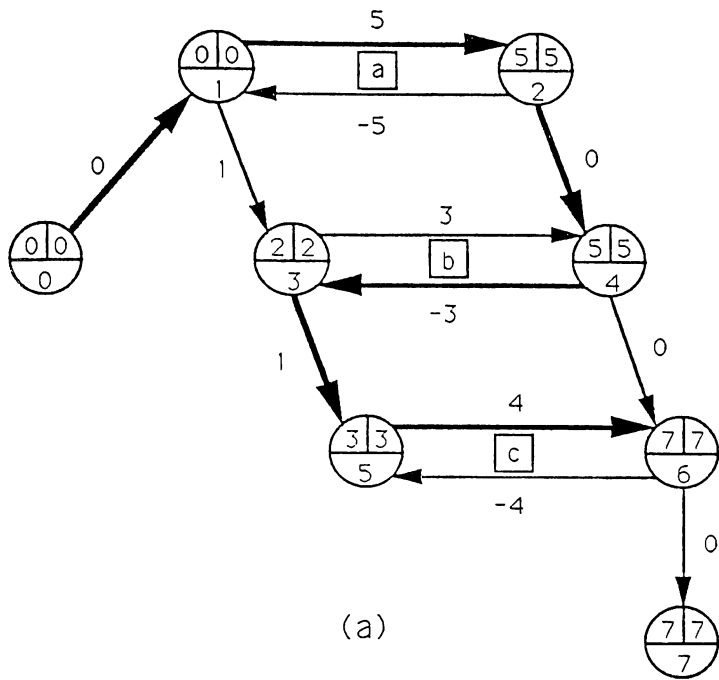


(c)

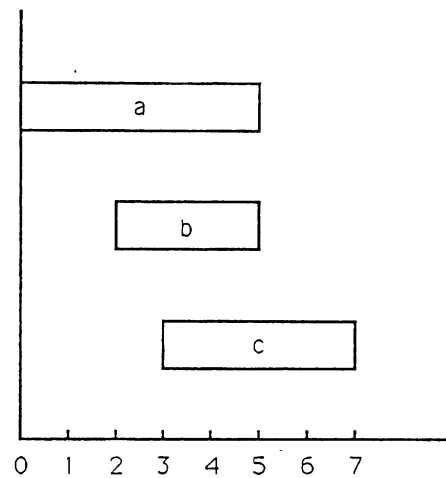
$$S_k = f_h + FF(h, k) - d_k$$

$$FS(h, k) = FF(h, k) - d_k$$

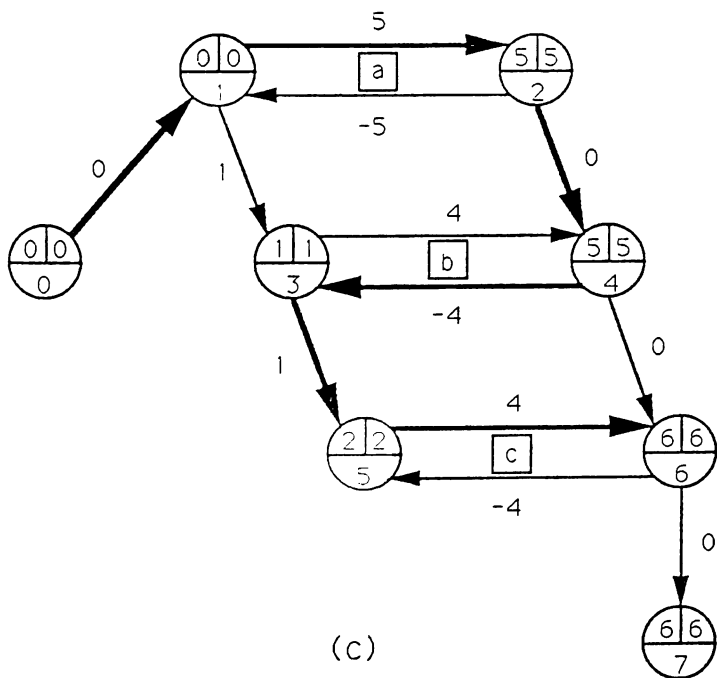
Figure 5



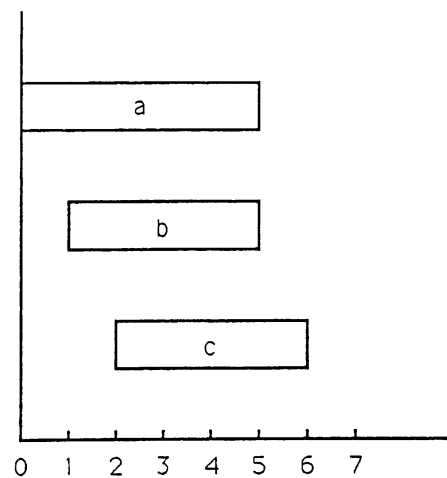
(a)



(b)



(c)



(d)

Figure 6

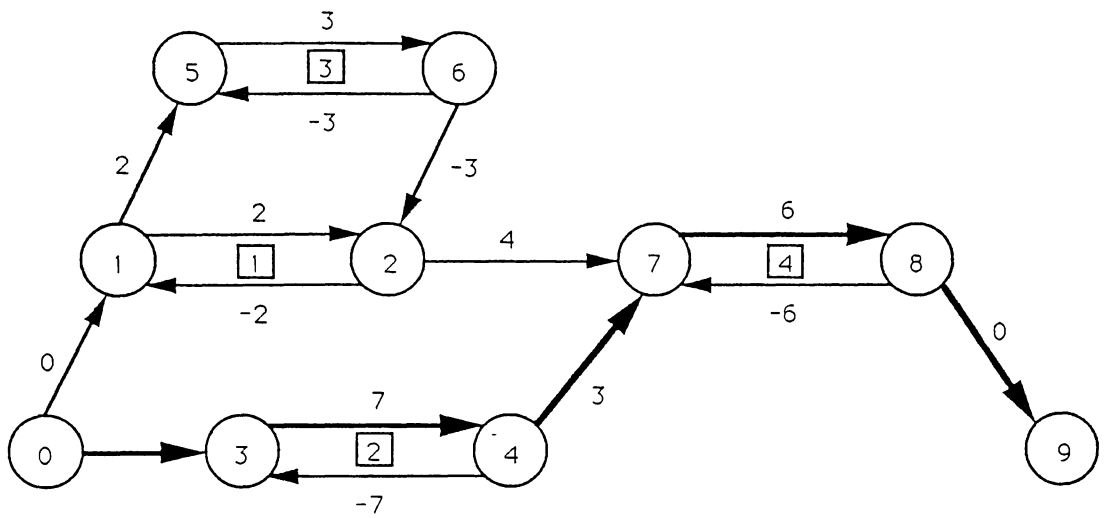


Figure 7

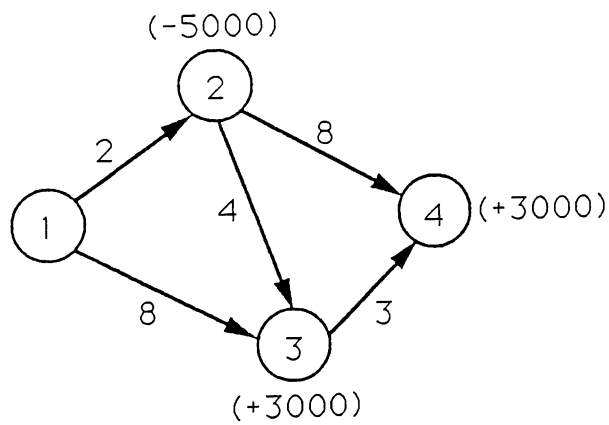
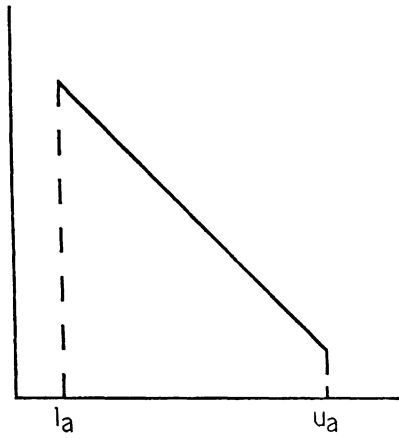
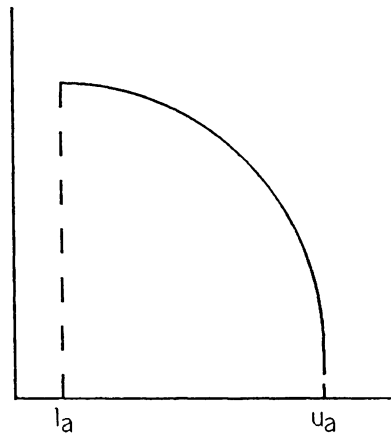


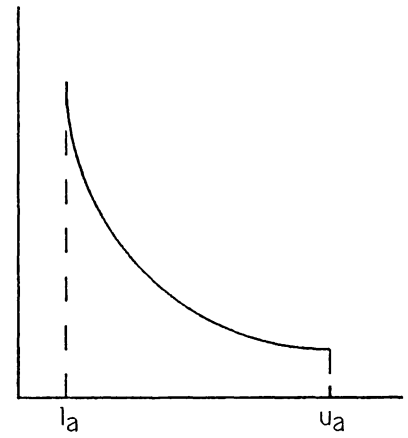
Figure 8



(a) Linear ϕ : proportional increase in cost



(b) Concave ϕ : high initial cost of reduction from the normal duration u_a



(c) Convex ϕ : low initial cost of reduction from the normal duration u_a

Figure 9

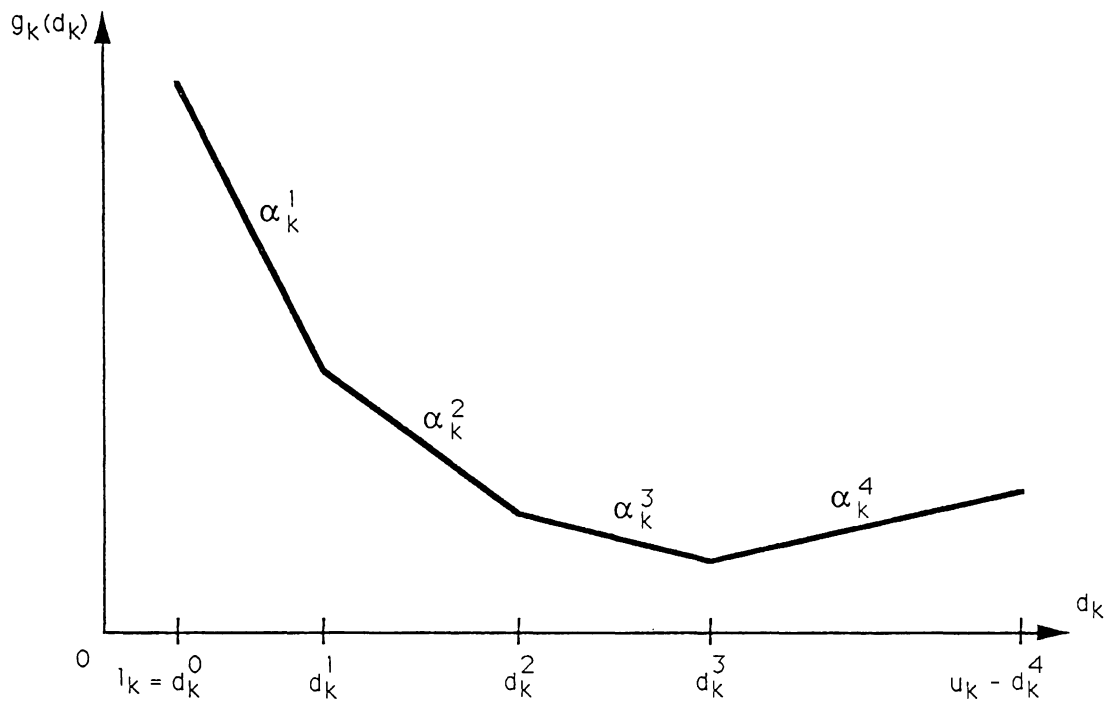


Figure 10