

## **Towards building an intelligent decision support system for project management**

C DEEPAK KUMAR<sup>‡</sup> and V V S SARMA<sup>†</sup>

Department of Computer Science & Automation, Indian Institute of Science, Bangalore 560 012, India

<sup>‡</sup>Present address: Electronics & Radar Development Establishment, Bangalore 560 093, India

<sup>†</sup>Present address: TRDDC (TCS), Tata Research Centre, Hadapsar Industrial Estate, Pune 411 013, India

email: vvs@research.trddc.ernet.in

**Abstract.** Management of large projects, especially the ones in which a major component of R&D is involved and those requiring knowledge from diverse specialised and sophisticated fields, may be classified as semi-structured problems. In these problems, there is some knowledge about the nature of the work involved, but there are also uncertainties associated with emerging technologies. In order to draw up a plan and schedule of activities of such a large and complex project, the project manager is faced with a host of complex decisions that he has to take, such as, when to start an activity, for how long the activity is likely to continue, etc. An Intelligent Decision Support System (IDSS) which aids the manager in decision making and drawing up a feasible schedule of activities while taking into consideration the constraints of resources and time, will have a considerable impact on the efficient management of the project. This report discusses the design of an IDSS that helps in project planning phase through the scheduling phase. The IDSS uses a new project scheduling tool, the Project Influence Graph (PIG).

**Keywords.** Intelligent decision support system; project management; semi-structured problems.

### **1. Introduction**

Management of typical large projects requires the manager to be adept in working with resource constraints and complex decision-making apart from meeting tight time schedules. And, if the project involves the development of a large, new, sophisticated system, such as a Surveillance Radar or a Satellite Launch Vehicle, the manager has to deal with the additional dimensions of ever-changing technologies, knowledge requirements in diverse

specialised areas of work, and geographical spread in the execution of various sub-activities of the project. To these, the problems of escalating project costs, unforeseen delays in certain activities, sudden non-availability of resources, and sometimes even changes in the specification of the end system, requires the project manager to constantly monitor the project progress, re-plan or re-schedule certain activities, or choose an alternate course of action.

Project scheduling has, till recently, followed the operations research (OR) approach of viewing a project as a set of activities alone, with each activity having a set of attributes such as duration, precedence relations etc., and, was represented as a network. The objective of project scheduling was to minimise the overall project duration using CPM/PERT methods. However, these techniques make an assumption of unlimited resources, which is generally not true in the real world domain. These techniques also have other inadequacies such as lack of recognition of rework cycle etc. (Cooper 1994). To tackle the resource constrained project scheduling problem, mathematical linear programming methods or heuristic rule methods are used, but with each having certain limitations.

Decision support systems (DSS) and artificial intelligence techniques provide a better toolkit to the project manager in dealing with semi-structured scheduling problems, such as those faced in R&D projects. In this paper, we use a new planning and scheduling tool, the Project Influence Graph (PIG), and AI search techniques, to develop an IDSS for project management.

## **2. The project scheduling problem**

A project may be defined as a collection of interrelated activities, each activity requiring various types of resources. In addition, a project possesses the following characteristics –

- a project should have a specific goal or a set of objectives to be accomplished in a finite time frame
- a project is homogeneous, in the sense that all activities that comprise the project are essential for the completion of the project, and no other outside activity has any influence on the completion of the project
- a large project typically, is of a complex nature involving a mixture a series and parallel activities requiring an inter-play of efforts, resources and time
- a project is usually of non-repetitive nature.

Every activity of the project is described by a set of attributes, though it is not necessary that all the attributes of all the activities need be known at the start of the project itself. The duration of the activity, resource requirements, the earliest date at which the activity can start, the latest date before which the activity must be definitely finished, precedence relationships with other activities etc., form some of the important attributes.

The project scheduling problem is to find a sequence of activities, i.e., assign a start date for each activity, such that some objective such as minimising the overall project duration, is achieved. PERT and CPM techniques result in such a sequence when the resource and timing constraints are ignored. The resulting sequence is a precedence-feasible schedule,

as only the precedence constraints are satisfied. Linear programming techniques and enumerative methods solve the problem with more realistic resource constraints being taken into consideration. However, the computational work load involved in these methods make them infeasible for solving even moderately large scheduling problems.

Noronha & Sarma (1989) describe the application of AI search techniques for the resource-constrained Project scheduling problem. They propose the use of informed search techniques such as A\* algorithm to come up with a feasible schedule, which has been employed in the development of this IDSS.

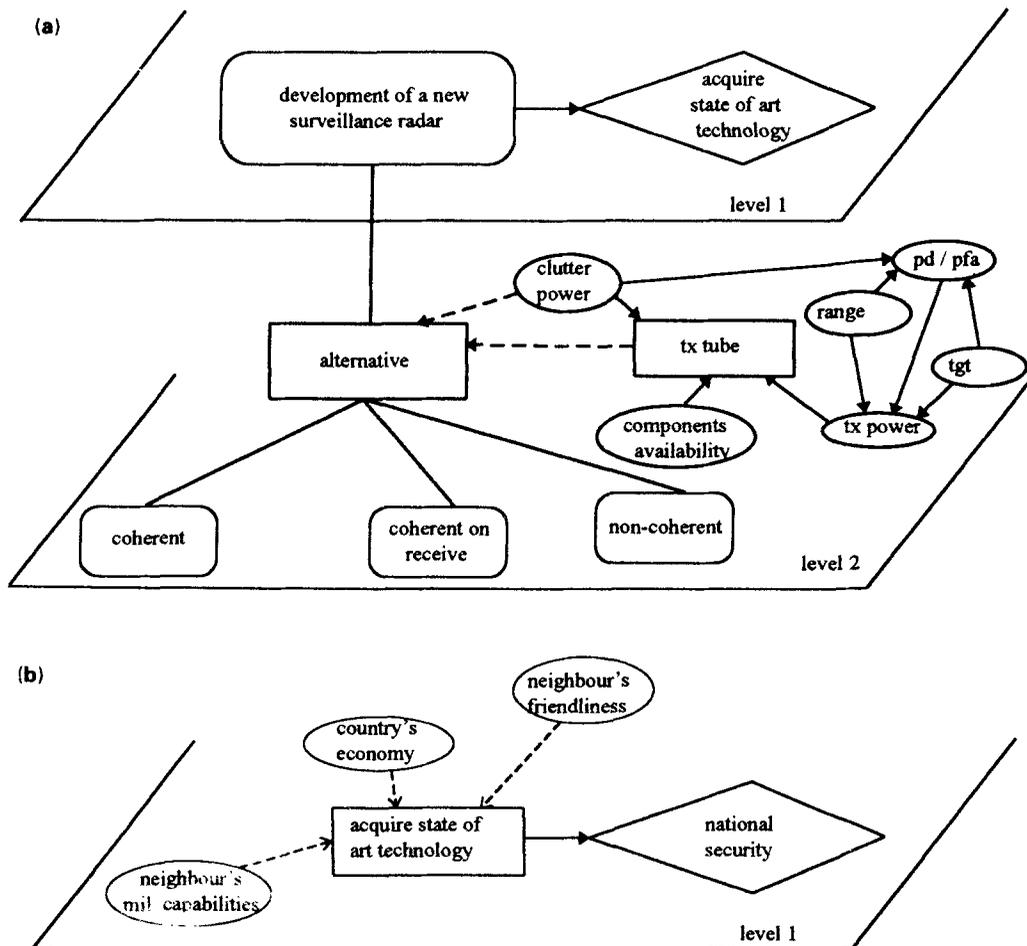
### 3. Project influence graphs (PIG)

The activity precedence network or PERT network has for long been used as a planning tool for project management. But, this graphical tool does not explicitly depict the decisions that were involved when the work of activities was being drawn up. During the monitoring and analysis stage of project management, it is often beneficial to know why a particular decision was taken, what were the factors that influenced the decision during the planning stage etc. Another drawback of PERT network is that it does not allow the manager to choose from alternate courses of action available to him, explicitly. A separate PERT network needs to be drawn up for each option of each activity. For example, a radar signal processor may be developed using DSP processors, using array processors, using microprogrammable slice processors, or by designing ASICs. Each course of action needs to be carefully evaluated with respect to duration, resource requirements, and impact on other activities, and the manager has to choose the most feasible option. Finally, a PERT network sometimes poses the problem of "Informational Overload" to the project manager. In large projects, involving thousands of small activities, a top level project manager need not be aware of all the details of each and every small activity. It is quite essential that the project be viewed at different levels of detail by different levels of project managers. In short, the requirement here is a hierarchical abstraction of activities and information.

Project Influence Graph (Noronha 1993) addresses these problems. A PIG effectively combines the features of AND/OR trees, activity precedence networks, influence diagrams and decision trees.

A PIG (figures 1 and 2) consists of a hierarchy of levels  $L_1, L_2, \dots, L_k$ , with each level providing a different view of the project. Higher the level (level  $L_i$  is higher than level  $L_j$  if  $i < j$ ) more is the abstraction of the project details. Each project description level consists of a set of statement nodes, decision nodes, object nodes and values nodes, interconnected by different types of arcs and links, such as dependency arcs, informational arcs, precedence arcs and state change arcs. The sub-graph formed by the set of activity (object) nodes along with precedence arcs is a directed acyclic graph (DAG). The sub-graph formed by the set of statement nodes, decision nodes, dependency arcs and informational arcs is also a DAG, in accordance to the definition of a "regular" influence diagram.

Statement nodes correspond to the chance nodes of influence diagrams. As in influence diagrams, statement nodes are depicted pictorially by circles or ellipses, and decision nodes by rectangles. As stated earlier the value nodes are represented by diamonds in a



**Figure 1.** (a) 'Growing' a PIG for a radar development project showing the different available alternative approaches. (b) An alternative viewpoint of layer 1.

PIG, while the object nodes are represented as rectangles with rounded corners. While in influence diagrams, both the dependency arcs and informational arcs are represented as solid boldface arrows, in a PIG, a distinction is made between the two, with dependency arcs being represented by solid boldface arrows and informational arcs being represented by dashed arrows. A dotted arrow depicts a time precedence arc.

The statement nodes help in representing the various facts or pieces of knowledge that need to be considered while reaching a decision, or, that which influences an activity etc. Each statement node  $S_i$  has a random variable  $X_i$  attached to it, which has either a discrete or a continuous probability distribution. In the case of computer representation and evaluation, the random variables are usually discrete in nature, with each having a finite set of outcomes. Alternatively, each statement node may have a fuzzy variable (Zadeh 1965) attached to it, thus forming a cognitive map instead of an influence diagram, and can be subjected to qualitative evaluation (Zhang *et al* 1989). The knowledge about the uncertainty associated with a statement is propagated across the dependency arcs linking

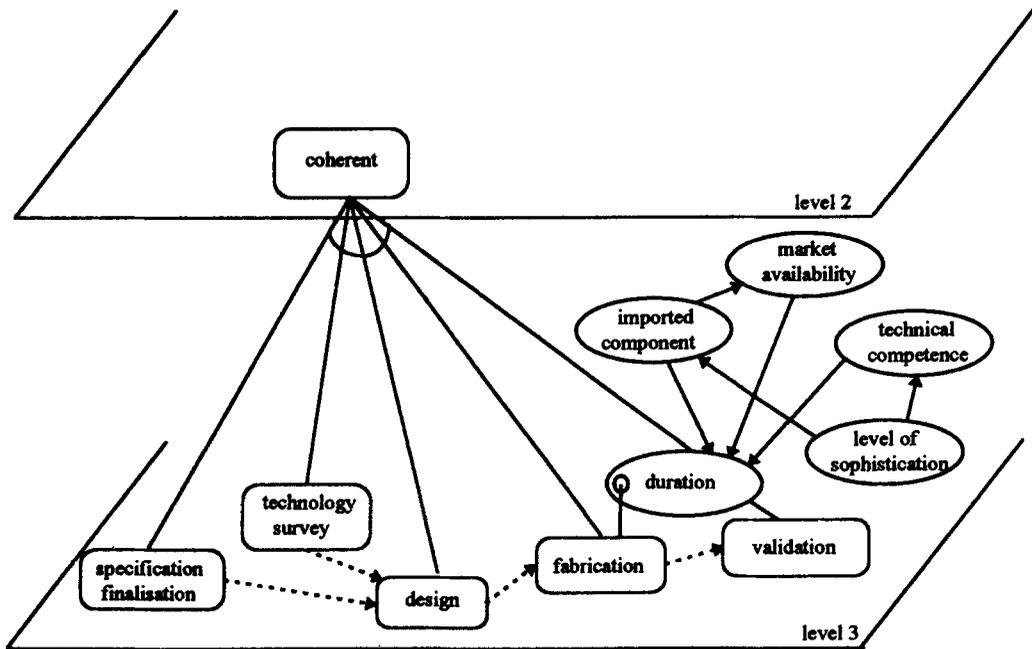


Figure 2. Decomposition of an activity into sub-activities and influence diagram for activity duration estimation.

the statement nodes using either the Bayesian probability analysis or fuzzy set calculus, depending on the type of variable that is attached to the nodes.

PIG distinguishes two important types of statement nodes – the observable chance nodes and the deterministic nodes. The observable chance nodes are those whose outcome is not deterministic at the time of planning, but its final outcome is observable at a later point in time and thus the subsequent decisions will have to be made based on this being observed. Observable chance nodes are drawn as meters and are always direct informational predecessors of decision nodes. The deterministic nodes are those whose outcome can be determined given the outcomes of its conditional predecessors. These are represented as triangles. However, in the present implementation, the statement nodes are not sub-classified.

The object nodes represent activities and resources. Activity nodes are linked to statement nodes via *structural links*. For example, the statement node ‘duration of activity’ is linked to the activity node ‘activity’ through a structural link. The lower level representation of these activity nodes (and of all nodes, in general) being *frames* (Fox 1985), statement nodes attached to object nodes are represented as *frame slots*.

The activities of different hierarchy level form the activity–subactivity structure. The set of activities in level  $L_j$  that are linked to an activity  $A_{ik}$  in level  $L_i (i = j - 1)$ , forms either the set of sub-activities of the activity  $A_{ik}$  or the set of alternate courses of action available to accomplish the activity  $A_{ik}$ . Thus, the links between activities across hierarchy levels forms an AND/OR tree structure. If an activity decomposition is an AND decomposition, the set of activities in the lower level to which it is linked forms the set of sub-activities, while an OR decomposition leads to a set alternatives. It may be noted here that activity decomposition

links may connect activities only between two adjacent hierarchy levels. This hierarchical structure is similar to the goal–subgoal decomposition used in AI techniques. This structure also permits information abstraction, providing simpler views of the project at higher levels, and in-detail views at the lower levels.

Hierarchy levels are represented as shaded planes, to give the impression of layers stacked along the vertical axis. Thus, activity decomposition takes place along this axis.

The advantages offered by PIG when compared to the other graphical representation tools for the project planning are elucidated by Noronha (1993).

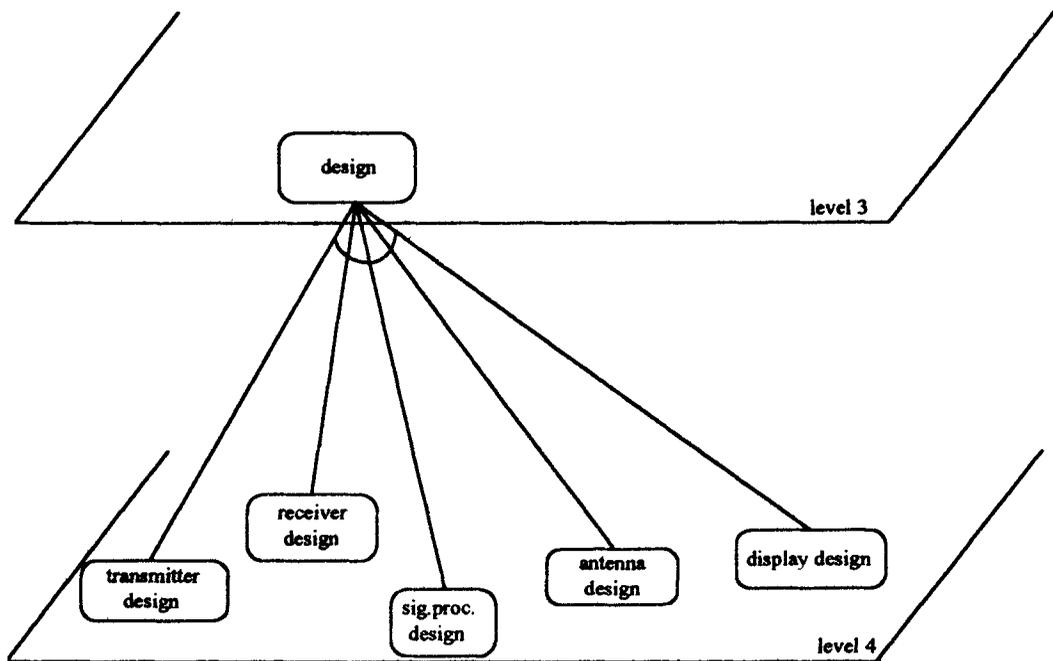
### 3.1 *An example*

Let us consider the example of a project for the development of a new surveillance radar system. The main objective of such a project may be to acquire the state of the art technology in the field of radars and improve upon the present radar systems. The decision to acquire such a technology on determining the necessity to embark on such a project may itself be a result of a number of factors, such as obsolescence of the present equipment, security scenario of the country, the relationship of the country with its neighbours, economic feasibility, availability and access to such a technology from elsewhere etc. The concepts of decision theory, cognitive mapping and influence diagrams may be used for arriving at this decision itself. However, we shall assume that such a decision has been made, and such a project has to be undertaken.

The main aim of the project being acquiring of the new technology, this forms the value node of level 1 of the PIG (figure 1a). This aim or objective is achieved by the successful development and fabrication of a new surveillance radar system, which forms the main activity of the project. Another viewpoint could be that level 1 has the value node of providing national security, which is determined by the decision of acquiring the state of the art technology. The various factors that influence this decision may also be depicted in layer 1. The outcome of this decision may be to develop a new radar system. Thus this activity replaces the decision node in layer 1. This is shown in figure 1b. Once the influence diagram shown in figure 1b has been evaluated and the decision for the development of a new radar system has been taken, then figure 1a replaces figure 1b and figure 1a is now the starting point for the ‘growth’ of a PIG.

Three different approaches may be candidates for consideration for building the new radar system—coherent radar approach, coherent-on-receive approach or the non-coherent radar approach. The decision to follow on the above approaches may depend upon a few technical factors, factors such as expected clutter power, and, basic requirements of the radar such as the required detection probability and false alarm probability, etc. Thus the fundamental activity of development of a new radar system of level 1, requires a decision to be made regarding the approach to be considered, depicted by the decision node of level 2. The various factors that influence this decision, are also shown linked to this decision node. The three alternatives of the decision node are shown as activity nodes in level 2, each linked to the decision node. The decision node is an OR node and the activity nodes are its children.

Assuming that the decision of following the coherent radar approach is finalised, further development of the PIG is shown in figure 2. However, while growing a PIG, the decision



**Figure 3.** Decomposition of an activity 'design' into sub-activities.

making is deferred till the PIG has been fully constructed. Thus, during the 'growing' phase, all the three options of the decision node are further expanded or decomposed. Only one option is considered here as representational example. The 'development of a coherent radar system' may be considered to consist of a number of sub-activities such as technology survey, feasibility study, design, development and fabrication, laboratory validation and final field trials. These sub-activities form a part of the hierarchical level 3, and linked to the parent activity in level 2, which is an AND node. This is because the activity of development of the coherent radar system can be considered complete only when each of the sub-activities has been successfully completed.

The activity 'fabrication' of level 3, in figure 2, is linked to the statement node 'duration', which is in turn shown to be linked to other statement nodes via dependency arcs. This represents the fact that the duration estimation of the activity 'fabrication' is influenced by these factors. The activity of design can only be embarked upon after the technology survey and feasibility study have been completed. This precedence constraint is represented by the dotted time precedence arrows between the nodes.

Figure 3 shows the decomposition of the activity 'design' into its component sub-activities, in hierarchy level 4.

#### 4. An IDSS using PIG

The ultimate goal of project scheduling is to decide upon a sequence of activities together with start and end dates for each activity, that is feasible within the constraints of resources,

precedence and time. While attempting to draw up such a plan, the project manager is confronted with two major decisions that he has to make for each activity, taking into consideration the constraints mentioned earlier –

- when to schedule an activity
  
- once an activity has been scheduled, what is the probable duration for which the activity lasts.

Another important type of decision that a project manager is faced in this phase of project execution, is the choice of course of action to be followed for an activity, amongst different alternatives. Since these decisions have to be made for each and every activity, for scheduling a large complex project consisting of numerous and varied activities with a large number of factors influencing these decisions, an intelligent decision support system will be of help for the project manager.

The IDSS has been developed in an Object Oriented Programming (1989) language, C++. The different nodes of the PIG are treated as objects, with interactions between them. At a higher level, the different nodes of the PIG are represented as frames (Fox 1985), with frame slots containing the various parameters that describe a node. Thus the entire PIG is represented as a frame network. However, the deterministic nodes and the observable chance nodes have not been separately represented, since they are basically chance nodes, with special properties.

The IDSS uses the PIG in facilitating the project manager to organise the information available regarding the project, with the hierarchical structure of the PIG helping him in “goal decomposition”. Once the PIG is “grown” by interacting with the project manager, the IDSS formulates the scheduling problem as a state space search problem and uses the A\* algorithm (Nilsson 1981) to arrive at the goal state, which is nothing but the state in which all the activities have been scheduled. To estimate the duration of an activity if it has not been specified by the manager, the IDSS forms an influence diagram (Miller *et al* 1976; Howard & Matheson 1981) of all the factors and decisions that influence the duration of the activity, and evaluates the same. The same methodology is employed for helping the project manager in choosing between the various alternative courses of action available to him, to pursue an activity.

The IDSS uses a combination of menu-driven and dialogue approaches to interact with the user. But instead, a graphical package that includes all the graph edit features will make the IDSS more user-friendly. The user starts a session by inputting the main activity node into the system. Going back to the example of the development of a radar system, the user inputs an activity node by that name into the hierarchy level 1. The system then switches over to the conversational mode, requesting the user for the various attributes of this node. Depending on the attribute values input by the user, the system proceeds to build the PIG with the help of the user. A session can also be commenced by loading a previously stored PIG from the database file.

IDSS consists of different modules, which may be grouped together into user interaction modules, file I/O modules and the evaluation modules.

#### 4.1 *User interaction modules*

These modules help in inputting the PIG, interacting with the user to develop a PIG, and display the same. Correspondingly, there are three modules, the graph edit module, the graph network module and the graph display module.

The graph edit module performs all the editing operations on the PIG. This module contains routines for adding a new node, modifying the parameters of an existing node and deleting a node. This module treats each node as an independent entity and while performing any operations on a node, the rest of the graph is considered. This is helpful because, in reality, the knowledge about the different activity nodes and its corresponding statement nodes, etc., may be better known to different people. For example, the attribute values of the 'transmitter design' activity may be better known to an expert in that field, while the attribute values of the 'signal processor design' activity may be better specified by an expert in signal processing techniques. Thus when the details of a particular activity are being input from one person, he or she may not be aware of the knowledge already acquired by the IDSS, and what information is still to be fed in.

Once all the nodes and their parameter values have been input using the edit module, the graph network module goes about linking these nodes based on the information present in each node. If in the process of linking, it finds that a particular node to which a link should exist has not yet been input, it interacts with the user to get the details of this node.

As the name suggests, the graph display module displays the presently active PIG on the terminal screen.

#### 4.2 *File I/O module*

The file I/O module stores the input PIG in a file. The file format is similar to a relational database file. The various nodes of the PIG are stored as objects while the links between the nodes are stored in the form of relational tuples. Apart from the PIG, the resource availability list also needs to be stored. This is stored in a separate file.

#### 4.3 *Evaluation modules*

Apart from the modules for evaluation of the influence diagram and activity scheduling, there is a module for checking the consistency of the PIG. This module ensures that there are no cycles in the activity precedence network. This module is also invoked by the influence diagram evaluation module to check that the influence diagram is regular and contains no cycles.

The scheduling problem is formulated as a state space search problem, with the state being defined as the set of activities that have been fully scheduled. The A\* algorithm is applied to reach the goal state of having all the activities scheduled. The initial start state has a null set of activities and is put on the open list. Each state has a heuristic value attached to it which is the estimated time for completion of the project when only the precedence constraints are considered. The open list of states is always maintained, sorted in ascending order with respect to this heuristic value. The first state in the open list is picked and checked if it is the goal state. If not, the state is put on the closed list, and is

```

BEGIN
  form the goal state
  initialise project calendar to the overall project start date
  make a list of activities that have no preceding activity constraint
  FOR each possible combination of this list of activities,
    check if resource requirements of the combination are met
    if yes, form a state node and calculate its heuristic value
  place all the state nodes in open list, sorted in ascending order
  DO BEGIN
    pick the first state node from the Open list
    IF the open list is empty, EXIT with failure
    advance calendar to the nearest end date or to the nearest
    resource availability date
    mark all the activities which have end dates equal to the
    present calendar date as finished
    IF present state is goal state, EXIT with success
    add the present state node to the closed list
    expand present state to create successor state nodes
    add the successors nodes to the open list, maintaining the
    ascending order
  END
END

```

**Figure 4.** Scheduling algorithm.

expanded to derive the successor states which are inserted into the open list based on their heuristic values. If the open list empties before the goal state is reached then the algorithm has failed to find a feasible schedule of activities.

The scheduling algorithm (figure 4) makes use of two other data structures, the Project Calendar and the Resource Availability List. The project calendar is akin to the simulation clock that is employed in Discrete Event System Simulations. The project calendar is initialised to the start date of the first set of activities that are scheduled, and is advanced to the nearest end date among the activities that are presently in progress. The activity is marked finished, thus enabling the scheduling of its successor activities, subject to the availability of resources and the earliest start date constraints. The resource availability list maintains the list of available resources and the dates at which they are made available.

The search module is invoked recursively while scheduling an activity which has sub-activities. Suppose an activity  $a_1$  has met all the constraints and can be scheduled, the algorithm first checks if all the sub-activities of the activity have been scheduled. If not, the search algorithm descends to the immediate lower hierarchy level invokes itself to schedule the sub-activities of  $a_1$ . In case an activity has different alternate courses of action and a choice has to be made, the influence diagram evaluation module is invoked. The influence diagram evaluation module is also invoked whenever an activity duration has to be estimated.

The influence diagram evaluation module requests the user for the conditional probability values to be entered in form of ratings, i.e., the user is asked to assign integral values to the outcomes of the chance node, dependant on the outcomes of its conditional predecessors. The ratings are then normalised to derive the probability values. This approach has been followed because humans find it relatively easier to deal with whole numbers as compared to fractions.

The influence diagram is then evaluated by effecting a string of graphical transformations which are informational preserving, and the expected value of the duration is derived. The algorithm proposed by Shachter (1986) for the evaluation of influence diagrams has been employed.

#### 4.4 *A sample session*

A sample session with the DSS, for the planning and scheduling of the R&D project of development of a new surveillance radar, is presented in this sub-section.

The final objective of the R&D project may be stated as “acquisition of the surveillance radar technology” which is depicted as the value node in level 1 of figure 1. The activity which realises this objective is the “development of a new surveillance radar system”. The user uses the edit mode of the IDSS to enter these two nodes into the system database. Once the user specifies the just stated activity node to the system, the system enters into a dialogue mode requisitioning the various attribute values of this activity with the help of a string of relevant questions. For example, the user is asked if there are a number of different options available for building this radar system. If no, the system checks if this activity can be decomposed into sub-activities. The user may decompose it into a set of sub-activities/alternatives that are depicted in level 2. Once the user specifies this decomposition, the system goes about eliciting the attribute values of these activities, finding out the various factors like the duration of these activities, the resources these activities require, etc. Thus, the PIG is grown recursively.

Once the input PIG has passed the consistency check successfully, the system invokes the A\* algorithm to come with a schedule of activities. The A\* algorithm initially starts at level 1, with the goal state being the scheduled activity “development of a new Surveillance Radar System”. In order to schedule this activity, the algorithm checks if a set of conditions are satisfied, such as

- if this activity has any predecessor activities that have to be completed before this activity may be scheduled, whether these activities have been completed.
- are all the resources required for this activity available
- if this activity has sub-activities/alternatives in the lower level, whether all have been scheduled.

Since this activity has a set of sub-activities, the A\* algorithm is invoked in a recursive fashion, with the goal state now being that this set of sub-activities be scheduled and finished. Now, there are a number of distinct possible options for scheduling this set of activities. The activities “specification finalisation” and “technology survey” are both eligible to be scheduled since both meet the set of conditions. Now, either activity may be scheduled followed by the other, or both the activities may be performed in parallel. The algorithm chooses that option which minimises the overall project duration, provided the resources required for that particular option are available. Once the activity “specification finalisation” is scheduled, i.e., the start date has been finalised the duration of the activity is essential to decide on its end date. If the duration is not very clearly known to the manager, but instead he comes up with a graph of factors that influences the duration, as shown in

figure 1, the influence diagram module is now invoked to evaluate this influence diagram and estimate the expected duration.

### 5. An example evaluation of an influence diagram

The evaluation of the hypothetical influence diagram that has been shown as a part of figure 3, for the decision on the choice of the alternative to be followed for the development of a new radar system, has been shown below.

The sequence of the probability tables shown below is the order in which the IDSS request the user for the information. The IDSS requests the user to provide ratings, instead of the probabilities, since the human mind finds it easier to deal with integral values.

Table 1 contains the probability of choosing a transmitter tube, considering the fact that the tube will have to be imported, or is available indigenously, or has to be developed. This table represents the preference of the project engineer to choose a particular type of transmitter tube given a particular constraint. In the given table, given the fact that the transmitter tube can be imported, the project engineer prefers to use the 'cfa' tube when compared to the other type of tube. The values of this table are thus input by the project engineer who is dealing with the design of the transmitter sub-system. The values of table 2 are input by the project manager. Table 2 shows that the project manager prefers to use a component that is available indigenously compared to either importing a component or developing it, possibly having considered the financial limitations. Thus, table 2 represents the knowledge or the preference of the overall project manager. The above is an example of how the knowledge of various experts is pooled together while taking a decision. In table 3 the project engineer inputs his choice of a particular type of transmitter tube given the different expected clutter scenarios. In case the radar is likely to encounter strong clutter, the engineer prefers to use either the 'cfa' tube or the 'twf'. In this clutter scenario, the 'mag'

**Table 1.** Probabilities of 'tx-tube' conditioned on the factor 'component-availability'.

	import	indigenous	develop
twf	0.3636	0.4000	0.2500
mag	0.1818	0.1000	0.6250
cfa	0.4545	0.5000	0.1250

**Table 2.** Marginal probabilities of 'component-availability'.

import	0.3333
indigenous	0.5555
develop	0.1111

**Table 3.** Probabilities of ‘transmitter tube’ conditioned on ‘clutter’.

	strong	weak	none
twt	0.5000	0.4167	0.2000
mag	0.0000	0.1667	0.6000
cfa	0.5000	0.4167	0.2000

**Table 4.** Probabilities of ‘transmitter tube’ conditioned on ‘power’.

	hi	low
twt	0.2500	0.4167
mag	0.3333	0.2500
cfa	0.4167	0.3333

**Table 5.** Probabilities of ‘power’ conditioned on ‘pd-pfa’.

	hi-low	hi-hi	low-low
hi	0.8000	0.8000	0.4000
low	0.2000	0.2000	0.6000

**Table 6.** Probabilities of ‘power’ conditioned on ‘range’.

	long	medium	short
hi	0.6667	0.5000	0.3333
low	0.3333	0.5000	0.6667

tube is not a good choice. In a similar fashion, the project engineer, as when prompted by the IDSS, inputs his preference based on his/her previous experiences or knowledge, the choice of a transmitter tube based on other factors such as the required operating power of the tube, which is shown in table 4. However, the transmitter power requirement is itself governed by other factors, the knowledge about which is better known to a radar system engineer rather than to the project engineer who is an expert in the transmitter sub-system alone. The radar system engineer inputs his knowledge in tables 5 and 6 which contain the probabilities of the transmitter power requirements for the various outcomes of the required ‘pd-pfa’ and ‘range’, based on certain system calculations. The two tables again represents the preference of the radar system engineer to either go in for high power system

**Table 7.** Probabilities of 'power' conditioned on 'target characteristics'.

	strong	weak
hi	0.2500	0.7500
low	0.7500	0.2500

**Table 8.** Probabilities of 'pd-pfa' conditioned on the influencing factor 'clutter'.

	strong	weak	none
hi-low	0.1429	0.2222	0.4000
low-hi	0.5714	0.3333	0.3000
low-low	0.2857	0.4444	0.3000

**Table 9.** Probabilities of 'pd-pfa' conditioned on 'range'.

	long	medium	short
hi-low	0.1250	0.2222	0.5714
low-hi	0.3750	0.4444	0.2857
low-low	0.5000	0.3333	0.1429

**Table 10.** Probabilities of 'pd-pfa' conditioned on 'target-characteristics'.

	strong	weak
hi-low	0.6667	0.1250
hi-hi	0.1667	0.5000
low-low	0.1667	0.3750

or a low power system considering the fact that the radar has a long operating range or a short operating range, and the probability of detection of the targets and probability of false alarms. Table 7 has the probabilities of the transmitter power requirements considering the fact that the target returns may be strong or weak, which are again input by the radar system engineer. The values of tables 8, 9 and 10 show the probabilities of the 'pd-pfa' values (probability of detection and of false alarm) that one is likely to get considering the various factors such as 'range', 'target' returns, and 'clutter power', which are provided by the radar system engineer based on his/her previous experiences. Tables 11, 12 and 13 show the marginal probabilities of the 'range', 'target characteristics' and 'clutter'

**Table 11.** Marginal probabilities of 'range'.

long	0.5714
medium	0.2857
short	0.1429

**Table 12.** Marginal probabilities of 'target characteristics'.

strong	0.2500
weak	0.7500

**Table 13.** Marginal probabilities of 'clutter'.

strong	0.4615
weak	0.3077
none	0.2308

**Table 14.** The ratings of the three choices for the 'tx\_tube' as evaluated by algorithm.

twt	0
mag	0
cfa	4

that the project manager expects the radar to encounter. Table 14 shows the result of the evaluation of the influence diagram of all the factors that are likely to influence the decision of selecting the transmitter tube. It is clear from the table that the choice of 'cfa' type of transmitter tube is the best given the present information. For the different clutter scenarios, the probabilities of choosing between the different types of radar, so that best results may be obtained, are presented in table 15. Table 16 shows the type of radar that one is likely to build considering the fact that the transmitter tube chosen is 'cfa'. Finally, table 17 shows the result of evaluation of the influence diagram corresponding to the decision about the choice of the alternative.

It may be reiterated here that the user may input the ratings instead of probabilities, since the human mind finds it easier to deal with integral values. For example, the radar engineer may input his preference for a transmitter tube given the fact that the component has to be chosen from the indigenously available tubes, in the form of 4:1:5 for twt:mag:cfa.

**Table 15.** Probabilities of the three options conditioned on 'clutter'.

	strong	weak	none
coherent	0.5714	0.3750	0.1429
coh-on-receive	0.2857	0.5000	0.1429
non-coherent	0.1429	0.1250	0.7143

**Table 16.** Probabilities of the three options conditioned on 'transmitter tube'.

	cfa
coherent	1.0000
coh-on-receive	0.0000
non-coherent	0.0000

**Table 17.** The ratings of the three options as evaluated by the algorithm.

coherent	3
coh-on-receive	0
non-coherent	0

## 6. Conclusion

This paper briefly discusses the necessity and importance of IDSS in project management, and goes on to describe an IDSS which uses a new project management tool, the project influence graph, AI search algorithm and influence diagram evaluation algorithm to output a feasible schedule of activities, taking into consideration the constraints of resources and time.

## References

- Cooper K G 1994 The rework cycle: Vital insights into managing projects. *IEEE Eng. Manage. Rev.* 21(3): 4-12
- Fox M S 1985 Knowledge representation for decision support. *Knowledge representation for decision support* (ed.) R H Sprague (Amsterdam: North-Holland)
- Howard R A, Matheson J E (eds) 1981 Influence diagrams. In *The principles and applications of decision analysis* (Menlo Park, CA: Strategic Decisions Group)
- Miller A C, Merkhofer M M, Howard R A, Matheson J E, Rice T R 1976 Development of automated aids for decision analysis (Menlo Park, CA: Stanford Res. Inst.)

- Nilsson N J 1981 Problem solving methods. In *Artificial Intelligence* (New York: McGraw-Hill)
- Noronha S J 1993 *Intelligent decision support systems for project planning and scheduling*. Ph D thesis, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore
- Noronha S J, Sarma V V S 1989 Artificial intelligence and knowledge-based approaches for scheduling problems. In *Project Management – Proc. Int. Conf. Expert Systems for Development*, Kathmandu, pp 105–114
- Object oriented programming: Special issue, Aug. 1989. *Comput. J.* 32: 4
- Shachter R D 1986 Evaluating influence diagrams. *Oper. Res.* 34: 871–882
- Zadeh L A 1965 Fuzzy sets. *Inf. Contr.* 8: 338–353
- Zhang W R, Chen S, Bedzek J C 1989 Pool2: A generic system for cognitive map development and decision analysis. *IEEE Trans. Syst. Man Cybern.* 19: 31–39