Int J Adv Manuf Technol (2001) 17:305–314

The International Journal of
Advanced Manufacturing Technology

Application of Petri Nets and a Genetic Algorithm to Multi-Mode Multi-Resource Constrained Project Scheduling

J. Prashant Reddy, S. Kumanan and O. V. Krishnaiah Chetty
Manufacturing Engineering Section, Indian Institute of Technology Madras, Chennai, India

Multi-mode and multi-resource constrained scheduling of a project is a complex task. This paper addresses the use of a Petri net as a modelling and scheduling tool in this context. The benefits of Petri nets in project scheduling are discussed. We propose extensions to Petri nets to suit scheduling of activities in a decision CPM. We also propose the use of a P-matrix for token movements in Petri nets. A genetic algorithm is used to find a better solution. Petri-net-aided software including genetic-algorithm-based search and heuristics is described to deal with a multi-mode, multi-constrained scheduling problem with pre-emption of activities.

Keywords: Activity pre-emption; Genetic algorithm; Multi-mode multi-resource scheduling; Petri nets; Project management

1. Introduction

Project management is a complex process involving many resource types that require optimum use. Often, the requirement of a type of resource may influence the requirement of other types. Each activity in a project may be performed in one out of a set of prescribed ways, called modes, with a mode specific duration and resource requirements. Several techniques have been developed in the past, for planning, scheduling and controlling of projects. The problem of scheduling activities bounded by resource constraints, resource availability and precedence relations is an exceedingly difficult task for projects of even modest size. This is especially true if an attempt is made simultaneously to minimise project duration and meet some other reasonable scheduling criteria [1]. The bulk of constrained resource procedures can be grouped under 1. mathematical, and 2. heuristic methods. Several mathematical techniques are reported for scheduling project activities, especially where resource constraints are a major factor [2]. Unfortunately, the mathematical formulations are extremely cumbersome because of the complexity inherent in large projects. Even computer implementations of the mathematical techniques sometimes become too cumbersome for real-time managerial applications [3]. Hence, a practical approach to scheduling is through the application of heuristics. The resource-constrained project-scheduling problem (RCPSP) is a challenging combinatorial optimisation problem which has lead to numerous publications dealing with the development of suboptimal and optimal solution procedures [4–9]. Most published articles on resource constrained project-scheduling algorithms have focused on algorithm quality using project duration as the only criterion. They have assumed fixed activity duration, fixed resource requirements over the activity duration, and fixed resource limits over time [10]. Emphasis is often placed on the production of accurate raw data and correct management of data for modern project planning and control [10]. In spite of the use of computers in data management as revealed by a recent survey of Indian managers [11], the practical managerial problems seem to outweigh data management. A fundamental assumption, often found in published works, is that activities are not pre-emptable [12–14]. Hence, very little is known about the potential benefits of activity pre-emption in the presence of varying resource availability [6]. This gap needs to be bridged. The project-scheduling problem with disconnected cash flows and generalised precedence relationships is highly complex and needs special attention [15]. A realistic model can be obtained if an activity duration is considered as a discrete function of the resources and the quantities of the resources allocated, along with multi-mode operation, and the possible pre-emption of activities. Such a problem is an extension of classical RCPSP and can be called a multi-mode multi-resource constrained project-scheduling problem (MMRCPSP). The use of resource-constrained project-scheduling is gaining importance in the area of manufacturing systems [16]. All the above-mentioned factors and alternative methods of performing the activities of a project (concepts of decision CPM) make the multi-mode multi-resource constrained scheduling a complex task. Hence, the need arises in project management for powerful graphical and analytical tools such as Petri nets (PNs) and search techniques such as genetic algorithms (GA).

Correspondence and offprint requests to: Professor O. V. K. Chetty, Manufacturing Engineering Section, Indian Institute of Technology Madras, Chennai 600 036, India. E-mail: ovk@acer.iitm.ernet.in

1.1 Petri Nets

A Petri net (PN) is a graphical and mathematical modelling tool which can be used as a visual communication aid in a similar way to flowcharts, block diagrams and network diagrams. The primary difference between Petri nets and modelling tools is the presence of tokens which are used to simulate dynamic, concurrent and asynchronous activities in a system. The behaviour of a system can be represented in a Petri net by setting up state equations, algebraic equations and other mathematical models.

Petri nets have been applied successfully in the areas of performance evaluation, communication protocols, legal systems, decision making, etc. [17]. A variety of Petri nets are reported in the literature: untimed [18]; timed [19]; coloured [20]; stochastic [21]; predicate [22]; priority [23] etc. Recently, many complex systems encountered in industry have been analysed using high-level Petri nets involving extensions with "colour", "time" and "hierarchy". These PNs are also used for prototyping of software, (re)design of logistic systems, and (re)design of administrative organisations. Use of stochastic coloured Petri nets for modelling flexible manufacturing systems, material handling systems and machines are available [24]. Many extensions to PNs are suggested to consider issues specific to problems on hand. Hierarchical timed extended Petri nets, are a form of extended Petri net that allows the development of structured MIMO subnets to model complex system functionalities [25]. A class of modelling tools called augmented timed Petri nets (ATPNs) have been introduced for modelling and analysis of robotic assembly systems with breakdown [26]. Project management has also been identified by some researchers [18] as a prospective area, where the modelling power of Petri nets could be explored. Our attempts to represent projects through PNs have revealed the need for additional considerations for representing decision nodes (a decision node is one from which alternative courses of action are indicated and one from amongst them will have to be chosen) and for the movement of tokens in a decision CPM. The details of decision CPM are given in [27,28]. A convenient way should be designed to meet the precedence relations in project management. General details of Petri nets are given in the Appendix.

1.1.1 What Petri Nets can do for Project Management

Petri nets offer many advantages to project managers. They are summarised as follows:

A PN is capable of modelling a system where many activities take place concurrently and asynchronously. It is capable of modelling concurrence, and conflicts. System deadlocks can be determined.

It provides information for a project manager to help, check, and reason about the tardy progress of activity.

It is capable of regenerating and rescheduling of activities (at the time of breakdowns and resource constraints).

A PN is a dynamic representation of a system, and hence, is suitable for on-line monitoring.

Logic expressed in words is transferred not only into a graphical form but also into a mathematical form suitable for analysis. Analytical models have maximum modelling flexibility, whereas simulation has reduced flexibility, Petri nets lie between these two approaches, providing analytical results with much of the modelling flexibility of simulation.

Dynamic simulation of a project can be visualised graphically. Using subnets, the parts of the project can be conveniently modelled. It is possible to simulate the entire project keeping any one of the subnets in the foreground and the rest in background.

A PN is capable of representing resource interdependency, partial allocation, substitution of resources, and mutual exclusivity.

Using behavioural properties such as reachability and boundedness, project planning can be improved (with constraints of resources).

A PN can be simplified by combining similar places and transitions. Using higher-level Petri nets, the graphical size of the network can be reduced.

Inclusion of dummy activities in the arrow diagrams of the project evaluation and review technique (PERT) could be cumbersome [1]. Representation of precedence relationships is easy with PNs.

1.2 Genetic Algorithm

A genetic algorithm is a search tool for global optimisation in a complex search space. Though genetic algorithms have already been applied to a wide range of different problem domains, only a few approaches have tried to apply them to scheduling problems. Masao and Ching [29] proposed a genetic algorithm for a multi-mode multi-resource constrained scheduling problem but with an assumption of non-pre-emptiveness.

2. Problem Definition

We consider the problem described as follows:

A project consists of a number of activities including decision nodes. Only one activity of the alternatives available at any decision-node can be selected to complete the project.

Multi resources required for activities are limited.

Each activity can be performed in any one of the modes specified. An activity once initiated in a specific mode (either interrupted or not), must be finished without changing the mode.

An activity cannot start until all its predecessors have been completed (precedence networks are outside the scope of the present paper).

Activities can be pre-empted or interrupted at any discrete time instant and resumed later without any set-up cost.

The objective is to minimise the project duration/cost.

2.1 Scope of the Paper

The present paper focuses on the applicability of PNs and GA for effective scheduling of activities of projects involving multi-modes, multi-resources, permitting pre-emption and includes decision nodes. Section 3 deals with extensions we propose to PNs and the software *MMRCS_Net* developed for the above-mentioned scheduling problem, and use of heuristics in this context is explained. The proposal to use a P-matrix for token movement in PNs is also detailed. Section 4 deals with a case study involving multi-mode, multi-resource constraints allowing pre-emption of activities, and containing decision nodes. Section 5 deals with conclusions and scope for further research.

3. Software Development and Extensions to PNs

A PN-aided software, including a GA-based search, package *MMRCS_Net* has been developed to deal with the multi-resource, multi-mode scheduling problem. The development of the software is explained with the help of flowcharts in Figs 1 and 2. The details of the software are as follows:

The *MMRCS-Net* developed in Borland C, consists of three modules Project_net module, GA_search module, and Sch_reso module.

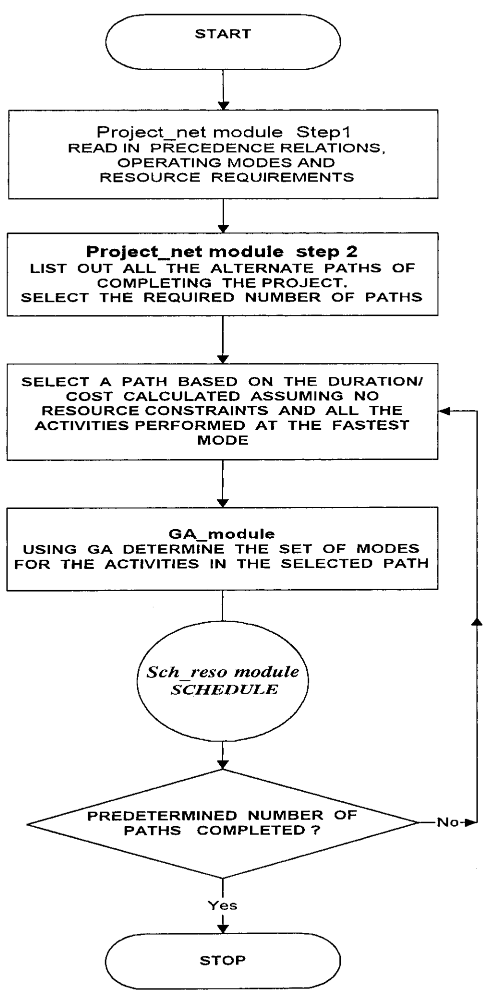

Fig. 1. Flowchart of MMRCS-Net.

3.1 Project_net Module

The first module, the Project_net, has been developed to deal with project modelling and analysis. In addition to the details of PNs presented in the Appendix, extensions to PNs are required considering the intricacies of project management.

In the Project_net:

A *place* represents start or completion of an activity.

A *transition* represents an activity. An activity is a time/resource consuming part of a project.

We propose decision places to represent decision nodes of a project. These are represented by two concentric circles (Fig. 3). Only one of the activities emanating from a decision place will be enabled.

Colours of the tokens will include: status (completed, incomplete); type of token (decision, non-decision); resources required (type and quantity); successor activities (list).

3.1.1 Token Movements

A token is removed from a place p_i when all/the last activity (transition) emanating from p_i are/is completed, unless it is a decision place when any one of the emanating activities will result in token removal.

A token is deposited into an output place p_j after the completion of an activity; and its colours are upgraded. If the output place p_j has a token already, its colour will be updated.

We propose the removal or deposition of a token with the help of a precedence matrix (P-matrix). The details of a P-matrix and its use are explained in Section 3.3.1.

Considering the execution of the first step (Fig. 1), the information regarding the project, i.e. precedence relations, decision nodes, if any, and the number of alternatives at each decision node are read in from the input file, Net_file. Information about resources indicating total resource requirements and requirement per time unit in each operating mode and for each activity are read in from the input file, Reso_file.

In the second step, all the alternative paths for completing the project are generated. The duration/cost associated with each alternative method is calculated with the assumption that there are no resource constraints and all the jobs are operating in their fastest mode (this provides information about the minimum duration). These alternatives are arranged in ascending order of their duration. A user-specified number of them are selected and are used by the genetic algorithm. The default option is one alternative.

3.2 GA_search Module

In the third step, the proposed genetic algorithm will evolve a near optimum, if not the optimum set of modes for the activities. The objective function for the genetic algorithm determines the duration associated with each set of modes of activities. Various steps in the proposed GA_search module are detailed below.

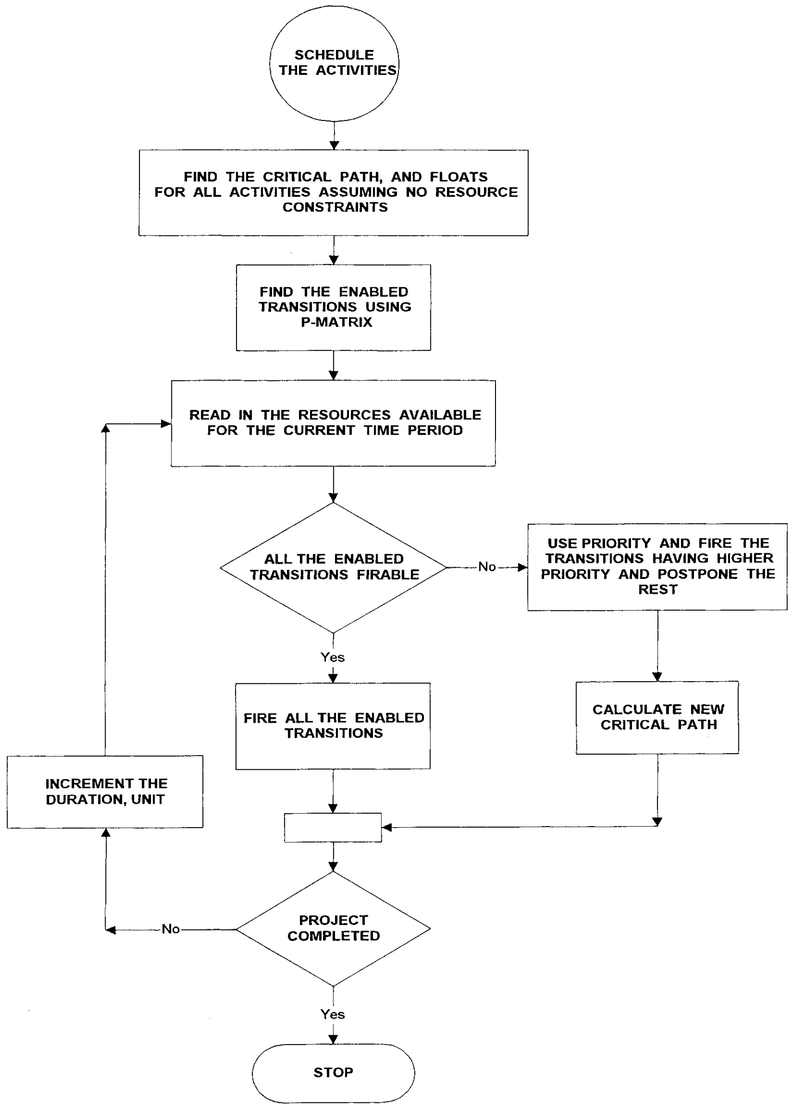

Fig. 2. Flowchart for scheduling using the Sch-reso module.

3.2.1 Coding of the String

In this application, a string or chromosome corresponds to the set of modes (identify the first mode as "0" and second mode as "1") for the activities in the selected path. For example, in Fig. 4, the string indicates that activity 1 must be performed in mode 1, 21 in mode 0, and activity 3 in mode 1 and so on.

3.2.2 Initial Population

The initial population, which is set to twice the number of elements in the string, is generated randomly.

3.2.3 Crossover

This is the next operation in the genetic approach. It partially exchanges information between the two selected strings. In the present approach, each string in the reproduction population is subjected to a crossover operation with a specified probability of crossover. The probability of crossover is usually high and can be in the range of 0.6–0.9 and is specific to the problem [30]. Once the string is chosen for crossover, its mate and crossover site are selected randomly.

Consider the two strings in Fig. 5, parent 1 and parent 2 with a crossover site 3. Offspring 1 is generated using the first three elements from parent 1 and the rest from parent 2. Similarly, offspring 2 is generated using the first three elements from parent 2 and rest from parent 1.

3.2.4 Mutation

This is the process of randomly modifying the string with a low probability. This helps to prevent the loss of some potentially useful strings and avoids sticking at the local optimum. This may produce an improved string that will be beneficial in the next population. The probability of mutation is usually low owing to the destructive nature of this operation. The

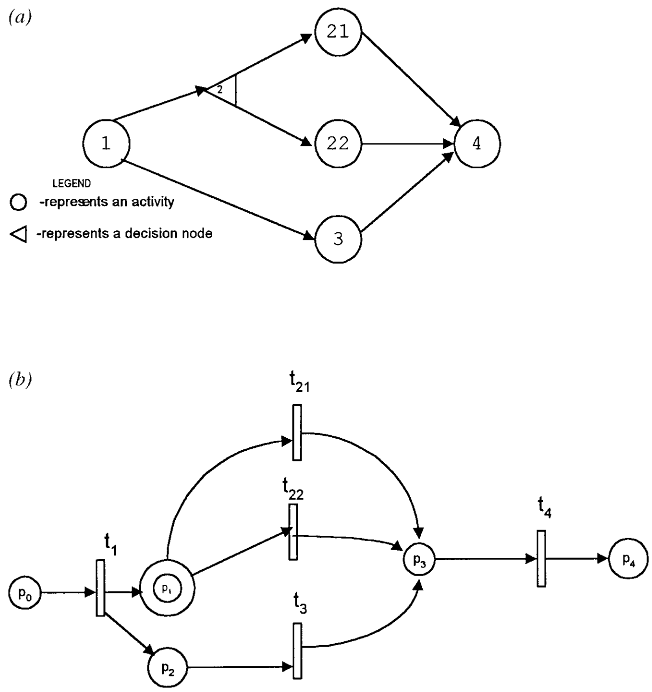

Fig. 3. (*a*) A project involving a decision mode. (*b*) The PN equivalent of (*a*).

1	0	1	1	0	0

Chromosome or string

1	21	3	4	51	6

Activity set

Fig. 4. Coding of a string.

1	0	0	1	1

Parent 1

1	1	0	0	1

Parent 2

1	0	0	0	1

Off spring 1

1	1	0	1	1

Off spring 2

Fig. 5. Crossover.

strings, after the crossover operation, are subjected to mutation with 0.3 probability of mutation. In order to determine whether a string is to be subjected to mutation or not, a random number is generated. If the random number is within the probability of mutation, the string is subjected to mutation. In Fig. 6, "2" and "4" are selected randomly as the sites for the parent string to be mutated. The new string formed after the mutation will be same as the parent except that its 2nd and 4th elements

1	0	0	1	1

Parent.

1	1	0	0	1

Off spring

Fig. 6. Mutation.

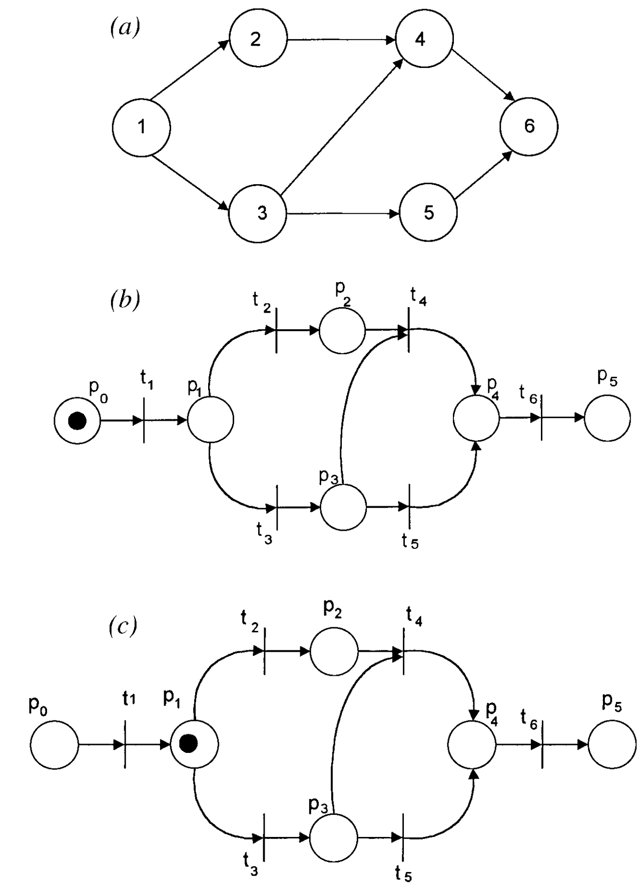

Fig. 7. (*a*) A PERT network (activity on node diagram). (*b*) PN before firing t_1. (*c*) PN after firing t_1.

have interchanged their positions. After the mutation is completed the strings are arranged in ascending order of fitness value, i.e. duration/cost. After sorting the strings, the new population is cut down to the size of the old population. Thus, one generation of the genetic process has been completed. Based on the complexity of the problem, the entire process is repeated to obtain the desired number of generations. The best string after the final generation is passed on to the Sch_reso module.

3.3 Sch_reso Module

The final stage is aimed at scheduling. A GA gives the set of activities and the corresponding set of modes for the selected path. The methodology used in the present context for resource

Table 1. (*a*) Initial P-matrix of Fig. 7(*b*) (before firing transition t_1). (*b*) P-matrix after transition t_1 completes firing (Fig. 7(*c*)).

(*a*)

Transition number	1	2	3	4	5	6	Row sum
1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	1
3	1	0	0	0	0	0	1
4	0	1	1	0	0	0	2
5	0	0	1	0	0	0	1
6	0	0	0	1	1	0	2

(*b*)

Transition number	1	2	3	4	5	6	Row sum
1	1	0	0	0	0	0	1
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	1	1	0	0	0	2
5	0	0	1	0	0	0	1
6	0	0	0	1	1	0	2

allocation involves the use of a set of heuristics and a P-matrix. This matrix indicates precedence relationships. In scheduling, enabled activities have to be identified. The usual procedure is to construct a precedence graph or precedence table, from which the enabled activities are obtained. We propose a convenient way, using P-matrix manipulation in connection with this work, using the rudimentary concepts of PNs. The information available in the P-matrix is also found to be useful for critical path calculations (such as earliest start (ES), latest start (LS), floats, etc.) involved in traditional project management using PERT [31]. Considering the requirement of the Petri net analysis, the same matrix could be used to determine the status of the project and identify deadlocks.

3.3.1 P-matrix and its Use

In a P-matrix an element "1" at position (i, j) indicates that activity in the jth column is a predecessor to activity in the ith row. A transition is considered enabled if its row sum is zero. After an enabled transition in the ith row completes its firing, "1" is placed in the position (i, i), and the precedence constraints, namely "1"s in the ith column, are removed. A new set of enabled transitions can then be identified, looking for rows with zero sums. This is illustrated in Fig. 7 and Table 1 which show that t_2 and t_3 are enabled after the completion of t_1. This process is repeated until all the elements of the leading diagonal contain "1", indicating that the project is completed.

We propose the following extensions to PNs for token movement:

A token is deposited in the output place of a transition t_i, when "1" is placed in its corresponding position in the leading diagonal of the P-matrix. Colours of the tokens are upgraded.

A token is removed from p_i when all the emerging activities from it are completed. The completion of emerging activities is established if the corresponding elements of the leading diagonal in the current P-matrix contain "1". For example, in Fig. 7, the token is removed from p_1 only if t_2 and t_3 complete firing (leading diagonal elements of 2 and 3 contain "1"). Colours of the tokens are upgraded.

A token from a decision place will be removed, when any one of the emanating activities complete firing. (It may be noted that the P-matrix has been designed to contain the activities on the chosen path.) Colours of the tokens are upgraded.

Table 2. Data of the problem with single resource constraint [9].

Activity	Immediate predecessors	Mode 1		Mode 2	
		Duration	Resource required per unit time	Duration	Resource required per unit time
1	–	2	4	3	3
2	–	2	4	4	2
3	1	2	3	3	2
4	1	1	3	3	1
5	2	2	3	3	2
6	3	1	2	2	1
7	4	3	2	5	1
8	5	2	4	3	3
9	6, 7	3	4	4	3
10	7, 8	3	4	4	3

Table 3. Final solution for the single resource constrained problem [9].

Duration	Activities in progress (mode)	Cumulative resource requirement
1	1(0), 2(1)	6
2	1(0), 2(1)	6
3	2(1), 3(0), 4(1)	6
4	2(1), 3(0), 4(1)	6
5	4(1), 5(0), 6(0)	6
6	5(0), 7(0)	5
7	7(0), 8(0)	6
8	7(0), 8(0)	6
9	9(1), 10(1)	6
10	9(1), 10(1)	6
11	9(1), 10(1)	6
12	9(1), 10(1)	6

3.3.2 Resource Constraints

Scheduling of multi-resources is shown in Fig. 2. In the first step, the enabled transitions are identified from the P-matrix and they are arranged in the order of importance, i.e. in the ascending order of total float.

In the next step, all types of resource available for the current time period are checked with the cumulative resource requirements of all the enabled transitions.

If any resource is not sufficient to fire all the enabled transitions (requiring this resource) at a particular time unit,

Table 4. Network data for the second case study.

Sample number	Activity/transition	Decision node (d)/non-decision node (n)	Pre-emptable (pe)/non-pre-emptable (npe) activity	Predecessors
1	1	n	npe	–
2	2	n	pe	1
3	3	n	npe	1
4	4_1	d	pe	1
5	4_2	d	npe	1
6	5	n	pe	2
7	6	n	pe	3
8	7	n	npe	4_1, 4_2
9	8	n	pe	6, 7
10	9	n	pe	8, 5
11	10	n	npe	8
12	11_1	d	pe	9
13	11_2	d	npe	9, 10
14	12	n	npe	9
15	13	n	pe	10, 11_1, 11_2, 12

Note: Suffix of an activity/transition indicates the alternative activity available at that decision node.

Table 5. Total resources required in the first mode of operation.

Sample number	Activity	Total resource requirement for resource type				
		1	2	3	4	5
1	1	30	24	24	18	30
2	2	14	14	14	14	14
3	3	32	24	48	48	32
4	4_1	64	64	64	64	64
5	4_2	70	70	70	70	70
6	5	35	35	30	30	30
7	6	36	36	36	36	36
8	7	30	30	30	20	20
9	8	30	24	24	18	30
10	9	24	24	24	24	24
11	10	30	30	30	30	30
12	11_1	45	45	45	45	45
13	11_2	36	36	36	36	36
14	12	30	30	30	30	30
15	13	30	36	24	36	30

Table 7. Total resources required in the second mode of operation.

Sample number	Activity	Total resource requirement for resource type				
		1	2	3	4	5
1	1	30	24	24	18	30
2	2	14	14	14	14	14
3	3	32	24	48	48	32
4	4_1	64	64	64	64	64
5	4_2	70	70	70	70	70
6	5	36	36	32	32	32
7	6	36	36	36	36	36
8	7	30	30	30	20	20
9	8	30	24	24	18	30
10	9	32	24	32	24	24
11	10	30	30	30	30	30
12	11_1	45	45	45	45	45
13	11_2	36	36	36	36	36
14	12	30	30	30	30	30
15	13	30	36	24	36	30

Table 6. Resource requirement per unit time in the first mode.

Sample number	Activity	Resource requirement per time unit for resource type				
		1	2	3	4	5
1	1	5	4	4	3	5
2	2	7	7	7	7	7
3	3	4	3	6	6	4
4	4_1	5	5	5	5	5
5	4_2	7	7	7	7	7
6	5	7	7	6	6	6
7	6	6	6	6	6	6
8	7	8	8	8	5	5
9	8	5	4	4	3	5
10	9	8	8	8	8	8
11	10	6	6	6	6	6
12	11_1	5	5	5	5	5
13	11_2	6	6	6	6	6
14	12	10	10	10	10	10
15	13	10	12	8	12	10

Table 8. Resource requirement per unit time in the second mode.

Sample number	Activity	Resource requirement per time unit for resource type				
		1	2	3	4	5
1	1	15	12	12	9	15
2	2	7	7	7	7	7
3	3	8	6	12	12	8
4	4_1	16	16	16	16	16
5	4_2	10	10	10	10	10
6	5	9	9	8	8	8
7	6	12	12	12	12	12
8	7	15	15	15	10	10
9	8	15	12	12	9	15
10	9	16	12	16	12	12
11	10	10	10	10	10	10
12	11_1	9	9	9	9	9
13	11_2	12	12	12	12	12
14	12	15	15	15	15	15
15	13	15	18	12	18	15

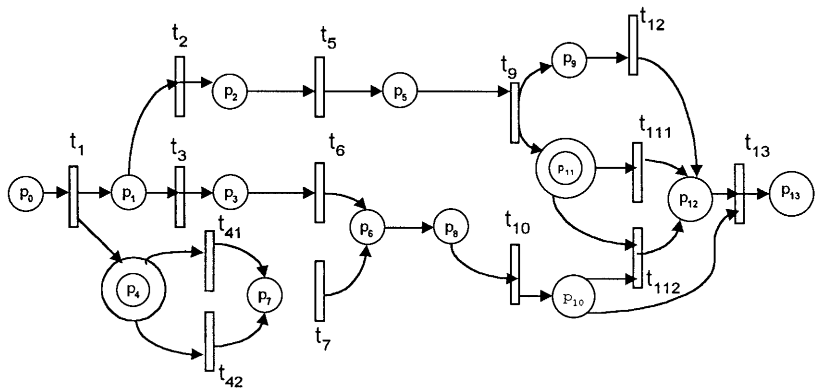

Fig. 8. The Petri net representation of the second case study.

Table 9. Project duration obtained in each path.

Path	Selected decision activities		Duration without resource constraints	Duration with resource constraints
1	4_1	11_1	20	24
2	4_1	11_2	19	23
3	4_2	11_1	22	23
4	4_2	11_2	21	22

Table 10. Final schedule.

Time unit	Progress of activities activity (mode)	Cumulative requirement of resource types				
		1	2	3	4	5
1	1(1)	15	12	12	9	15
2	1(1)	15	12	12	9	15
3	3(1), 4_2(1)	18	16	22	22	18
4	3(1), 4_2(1)	18	16	22	22	18
5	3(1), 4_2(1)	18	16	22	22	18
6	3(1), 4_2(1)	18	16	22	22	18
7	6(0), 4_2(1), 2(1)	23	23	23	23	23
8	6(0), 4_2(1), 2(1)	23	23	23	23	23
9	6(0), 4_2(1), 5(1)	25	25	24	24	24
10	6(0), 7(1)	21	21	21	16	16
11	6(0), 7(1)	21	21	21	16	16
12	6(0), 5(1)	15	15	14	14	14
13	8(1), 5(1)	24	21	20	17	23
14	8(1), 5(1)	24	21	20	17	23
15	9(0), 10(1)	18	18	18	18	18
16	9(0), 10(1)	18	18	18	18	18
17	9(0), 10(1)	18	18	18	18	18
18	11_2(1), 12(0)	22	22	22	22	22
19	11_2(1), 12(0)	22	22	22	22	22
20	11_2(1), 12(0)	22	22	22	22	22
21	13(1)	15	18	12	18	15
22	13(1)	15	18	12	18	15

priority is given to the enabled transitions according to the following order of heuristics:

1. Activities which have started previously but cannot be pre-empted.
2. Activities which are on the critical path.
3. Activities with minimum float, if in tie, choose arbitrarily.

Enabled activities which can be interrupted (pre-empted) and activities which are currently enabled but cannot be started owing to the resource non-availability, are postponed to the next time unit.

The above steps are repeated until all the leading diagonal elements of the P-matrix are turned to “1”, indicating the completion of the project. The final schedule detailing the activities in progress, cumulative resource requirement for each time unit and total duration/cost of the project is obtained.

4. Case Studies and Discussion

Most of the published work on project scheduling addresses project execution with a single mode. Published literature indicates a large number of heuristics but it is found that none of them performed consistently with data sets. In addition, the study of projects with pre-emption of activities is restricted and, hence, the benefits of pre-emption are not fully known. Researchers are therefore encouraged to develop better heuristics and efficient tools to help deal with real-life projects. Two case studies were considered. The first case study is used for validation of the proposed methodology. The second case study is formulated with an attempt to solve a problem which is close to a real-life scenario and considers decision nodes, multi-mode operations, multi-resource constraints, and pre-emption of activities. The multi-mode single resource constrained problem in [9] is considered as first case study and solved using the *MMRCS_Net*. This project involves 10 activities. Each activity has two possible execution modes and the resource limit is 6 units per period. The objective is set to minimise the project duration. Table 2 shows the precedence relations, resource requirements and periodwise resource requirements in both the modes. The final schedule is shown in Table 3. The results obtained tally with the published results of Boctor [9], thereby validating the MMRCS_net.

A second case study dealing with decision activities, multi-mode, multi-resource constraints is considered. This consists of 15 activities in total, including two decision nodes having two alternatives each as detailed in Table 4. Some of the activities are pre-emptable. The PN for this case study is shown in Fig. 8. The two decision nodes with two alternatives each will result in four alternative paths (2×2). In each path, considering two modes of operation for each activity, there will be 2^{13} ways of selecting the modes for the selected 13 activities. Five types of resource are considered. The maximum availability of each resource is 25 units, for every time unit over the entire span of the project. The totals as well as the periodwise resource requirements for each activity, in both the modes, are shown in Tables 5 to 8.

This case study is solved using MMRCS_net and the results obtained are shown in Table 9. The best among the four paths is taken and the activities are scheduled according to their modes of operation. The final schedule is shown in Table 10. Though many activities are pre-emptable, it may be noted that in the present case, only activity 5 is pre-empted on day 10 and rescheduled on day 12. The GA search in the present context enables the much desired minimisation of the number of pre-emptions.

The manager of a project can benefit in a number of ways. It should be emphsised that, apart from graphical representation of a project to help him, the use of PNs along with the proposed P matrix manipulation is a simple and convenient way to determine the status of the project at any instant, to evaluate dead locks, if any, and to identify the enabled activities at any instant, to minimise the number of pre-emptions and thereby to deal with real-life projects involving complexities. Conventional approaches lead to local optimisation, hence, the global optimum may be lost, but a GA search, being global, gives a possibility of reaching the optimum solution. There is a need for further research to explore the potential of PNs in combination with GA for applications to project management.

5. Conclusion and Scope for Further Research

There is an apparent gap between the theoretical work and the requirements of project schedulers and managers. This paper elucidates the numerous benefits that Petri nets can bring to project managers. It attempts to exploit Petri nets for modelling and resource-constrained scheduling in project management. Extensions required to PNs to help project management are proposed herein. The use of a P-matrix for token movements is also proposed. This work indicates the efficacy of GA as a directed quick search technique to find an improved solution in a complex search space in the case of decision networks. In this context, a PN-aided software including a GA-based search and heuristic for planning, scheduling and resource allocation in projects, has been developed. PN-aided software can conveniently deal with projects having decision activities with multi-modes, multi-resource constraints and can efficiently manage pre-emption of activities.

5.1 Scope for Further Research

PNs can aid in the study of practical management aspects such as resource substitution and resource sharing in projects. The relationships found in Precedence Networks (the relationships of activities such as Start–Start, Finish–Finish, Start–Finish and Finish–Finish) may be implemented to represent a more realistic situation. Partial or total rework of an activity may also be implemented. The use of adaptive probabilities in GA search may be advantageous.

Acknowledgement

The authors express sincere thanks to the Department of Science and Technology, Government of India, for the financial support under the Science and Engineering Research Council, to undertake this research work.

References

1. J. D. Wiest and F. K. Levy, A Management Guide to PERT/CPM: with GERT/PDM/DCPM and Other Networks, New Delhi, Prentice-Hall of India, 1977.
2. A. Shtub and R. Etgar, "A branch and bound algorithm for scheduling project to maximize npv", International Journal of Production Research, 35(12), pp. 367–378, 1997.
3. B. A. Badiru, Project Management in Manufacturing and High Technology Operations, Wiley-Interscience, 1988.
4. A. Sprecher, S. Hartmann and A. Drexl, "An exact algorithm for project scheduling with multiple modes", OR Spectrum, 19, pp. 195–203, 1997.
5. E. A. Elsayed and N. Z. Nasr, "Heuristics for resource constrained scheduling", International Journal of Production Research, 24(2), pp. 299–310, 1986.
6. S. H. Demeulemeester and W. S. Herroelen, "An efficient optimal solution procedure for the pre-emptive resource constrained scheduling problem", European Journal of Operations Research, 90, pp. 334–348, 1996.
7. F. F. Boctor, "Some efficient multi-heuristic procedures for resource constrained project scheduling", European Journal of Operations Research, 49, pp. 3–13, 1990.
8. F. F. Boctor, "Heuristics for scheduling projects with resource restrictions and several duration modes", International Journal of Production Research, 31(11), pp. 2547–2558, 1993.
9. F. F. Boctor, "A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes", European Journal of Operations Research, 90, pp. 349–361, 1996.
10. R. J. Willis, "Critical path analysis and resource constrained project scheduling – theory and practice", European Journal of Operations Research, 21, pp. 149–155, 1985.
11. K. Ananthanarayanan and N. Kalyanasundaram, "Management techniques in Indian Construction industry – a questionnaire survey", Journal of Institution of Engineers (India), pp. 73–75, 1991.
12. F. Salewski, A. Schirmer and A. Drexl, "Project scheduling under resource and mode identity constraints: model, complexity, methods and applications", European Journal of Operations Research, 102(1), pp. 86–110, 1997.
13. J. Nabrzyski and J. Weglarz, "On an ES with multi criteria tabu search for multi objective project scheduling", Proceedings of IEEE, 6th International Conference on Emerging Technology and Factory Automation, pp. 287–291, 1997.
14. R. Kolisch and A. Drexl, "Local search for non pre-emptive multi mode resource constrained project scheduling", IIE Transactions, 29, pp. 987–1088, 1997.
15. B. DeReyck and W. Herroelen, "An optimisation procedure or resource constrained project scheduling problem with discounted cash flows and generalized precedence relations", Journal of Computer and Operations Research, 25(1), pp. 1–17, 1998.
16. B. DeReyck and W. Herroelen, "Assembly line balancing by resource constrained project scheduling technique – a critical appraisal", Journal of Foundations of Computations and Decision Sciences, 22(3), pp. 143–167, 1987.

17. T. Murata, "Petri nets: properties, analysis and applications", Proceedings of the IEEE, 77(4), pp. 541–580, 1989.
18. J. L. Peterson, Petrinet Theory and Modelling of Systems, Englewood Cliffs, NJ, Prentice Hall, 1981.
19. C. Ramachandani, "Analysis of synchronous concurrent systems by timed Petri nets", PhD dissertation, MIT Cambridge, 1974.
20. K. Jensen, "Coloured Petrinets and invariant method", Theoretical Computer Science, 14, pp. 317–336, 1981.
21. M. K. Molloy, "Performance analysis using stochastic Petrinets", IEEE Transactions on Computers, VC-31(9), pp. 913–917, 1982.
22. H. J. Genrich, Predicate/Transition Nets, Advances in Petrinets, Lecture Notes in Computer Science, Springer-Verlag, 1987.
23. K. Ravi Raju and O. V. Krishnaiah Chetty, "Priority nets for scheduling flexible manufacturing systems", Journal of Manufacturing Systems, 12(4), pp. 326–340, 1993.
24. K. E. Moore and S. M. Gupta, "Stochastic coloured Petrinet models of flexible manufacturing systems: material handling systems and machining", Computers and Industrial Engineering, 29, pp. 333–337, 1995.
25. S. Ramaswamy, K. P. Valavinis and S. Barer, "Petri net extensions for the development of MIMO net models of automated manufacturing systems", Journal of Manufacturing Systems, 16(3), pp. 175–191, 1997.
26. V. Kurapati and I. Mohammad, "Real time Petri nets for modeling, controlling, and simulation of local area networks in FMS", Computers Industrial Engineering, 28(1), pp. 147–162, 1995.
27. W. Crowston and G. L. Thomson, "Decision CPM: a methodology for simultaneous planning scheduling and control of projects", International Journal of Production Research, pp. 406–426, 1965.
28. W. Crowston and B. S. Wallace, "Decision network planning models", Management Sciences Research Report 138, Carnegie–Mellon University, Pennsylvania, 1968.
29. M. Masao and C. T. Ching, "A genetic algorithm for multi-mode resource constrained scheduling problem", European Journal of Operational Research, 100, pp. 134–141, 1997.
30. D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
31. V. Jeetendra, S. Kumanan and O. V. Krishnaiah Chetty, Application of Petrinets to Project Management, Proceedings of International Conference on CAR and FOF '98, Coimbatore, India, pp. 853–860, December 1998.

Appendix

Untimed Petri Nets

The graph of a Petri net is a directed, weighted, bipartite graph consisting of two kinds of node, called places and transitions, connected with arcs.

A Petri net is a 5-tuple

PN=$(P,T, IN(p,t)\ OUT(p,t), M_o)$ where
P=$\{p_1,p_2,p_3,\ldots,p_n\}$ is a finite set of places,
T = $\{t_1,t_2,t_3,\ldots,t_n\}$ is a finite set of transitions.
$IN(p,t)$: $(P \times T) \rightarrow S$ and $OUT(p,t)$: $(T \times P) \rightarrow S$ are input and output function defining directed arcs, where S is a set of all positive integers.
M_o: $P \rightarrow \{0,1,2,3,\ldots\}$ is the set of initial marking
$P \cap T = \phi$ and $P \cup T \neq \phi$.

Timed Petri Nets

A timed Petri net (TPN) includes a set of firing transition duration, D. The TPN is formally defined as:
TPN = $(P, T, IN(p,t)\ OUT(p,t), M_0, D)$
Where D:$\rightarrow \{0, R+\}$ is the set of all non-negative real numbers.

A transition has an associated deterministic firing. There are two events namely, "Start firing" and "End firing". In between those two events, the firing is in progress. The removal of tokens from a transition's input place occurs at start firing. The placement of a token in a transition's output place occurs at the end of firing. While the firing of a transition is in progress, the time to end firing, called the remaining firing time (R) decreases from the firing duration to zero (when R=0, the firing of transition is completed).

Stochastic Coloured Petri Nets

A stochastic coloured Petri net (SCPN) consists of the following elements:

SCPN = $\{P,T,C(p),C(t),IN(p,t)\ OUT(p,t),\Psi\}$
$C(p)$ = colour set associated with each place $p \in P$, $C(t)$ = colour set associated with each transition $t \in T$, also,
$C(p_i) = \{a_{i1},a_{i2},\ldots,a_{iui}\}$, $i = 1,2,\ldots,n$, and $u_i = |C(p_i)|$; $C(t_j) = \{b_1,b_2,\ldots,b_{jvj}\}$,
$j = 1,2,\ldots,m$, and $v_j=|C(t_j)|$; a and b are the associated colours.
$IN(p,t) : C(p) \times C(t) \rightarrow N$ is an input function and, $OUT(p,t) : C(p) \times C(t) \rightarrow N$ is an output function, where N is the set of all non-negative integers.
Ψ = firing rate associated with a transition.

The input arc from a place p_i, with respect to a colour a_{ih}, to a transition t_j, with respect to a colour b_{jk}, is denoted by scalar $IN(a_{ih},b_{jk})$, for some h and k. Similarly, the output arc is denoted by $OUT(a_{ih},b_{jk})$.