Int J Adv Manuf Technol (2006) 31: 360–366
DOI 10.1007/s00170-005-0199-2

ORIGINAL ARTICLE

S. Kumanan · G. Jegan Jose · K. Raja

Multi-project scheduling using an heuristic and a genetic algorithm

Received: 15 July 2004 / Accepted: 6 June 2005 / Published online: 13 May 2006

Abstract Managing multiple projects is a complex task. It involves the integration of varieties of resources and schedules. The researchers have proposed many tools and techniques for single project scheduling. Mathematical programming and heuristics are limited in application. In recent years non-traditional techniques are attempted for scheduling. This paper proposes the use of a heuristic and a genetic algorithm for scheduling a multi-project environment with an objective to minimize the makespan of the projects. The proposed method is validated with numerical examples and is found competent.

Keywords Genetic algorithm · Heuristic · Multiproject scheduling · Project management · Resource allocation · Resource constraints

1 Introduction

New product development, variety product manufacturing, maintenance of systems, infrastructure constructions are few vital areas of multiproject environment. In a multi project environment problem a number of projects must concurrently share limited resources with in the equivalent precedence constraints. Each activity in a project must be performed in a particular mode with specified duration and resource requirements. Most of the techniques developed in the past favored scheduling a single project or multi-project represented as a single project [1]. Mathematical techniques for scheduling project activities with constraints are extremely cumbersome [2] and become seldom possible for multiproject. Efficient and effective scheduling for multiproject environment needs attention.

S. Kumanan (✉) · G. Jegan Jose · K. Raja
Department of Production Engineering,
National Institute of Technology,
Tiruchirappalli 620015, India
e-mail: kumanan@nitt.edu

1.1 Heuristics in project scheduling

Heuristic refers to a particular approach to decision-making that is rapidly growing in application and importance [3]. Heuristic procedures and scheduling rules are aimed at the development of good solutions [4]. Researchers have contributed many such heuristics for project scheduling [5–10]. The heuristic approaches are discussed in the recent survey by Kolisch and Drexl [11], in which extensive experiments are carried out on a set of 536 test problems [12] and four different heuristic methods [13, 14], and are compared with the optimal solutions and are found to be a viable solution, but very inconsistent.

1.2 Genetic algorithm in scheduling

The ideas involved in genetic algorithms (GAs) were originally developed by Holland [15] and described in greater detail by Goldberg [16]. Genetic algorithms are search techniques for global optimization in a complex search space. As the name suggests, they employ the concepts of natural selection and genetics. Using past information they direct the search such that the expected performance will be improved. Although 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 until now and, moreover, most of them have been restricted to job shop, flow shop scheduling problems [17, 18] or production scheduling problems [19]. Portmann [20] has presented a detailed survey on GA's in the area of scheduling with analyses of the effects of various operators like encoding, crossover, and mutation. Maso and Ching [21] proposed a genetic algorithm for a multi-mode multi-resource constrained scheduling problem but with an assumption of non-preemptiveness. Leu and Yang [22] proposed a GA based multicriteria optimal model for construction scheduling. Hegazy [23] proposed a genetic algorithm in optimization of resource allocation and leveling. Latter Reddy et al. [24] proposed a genetic algorithm approach to multi-mode multi-

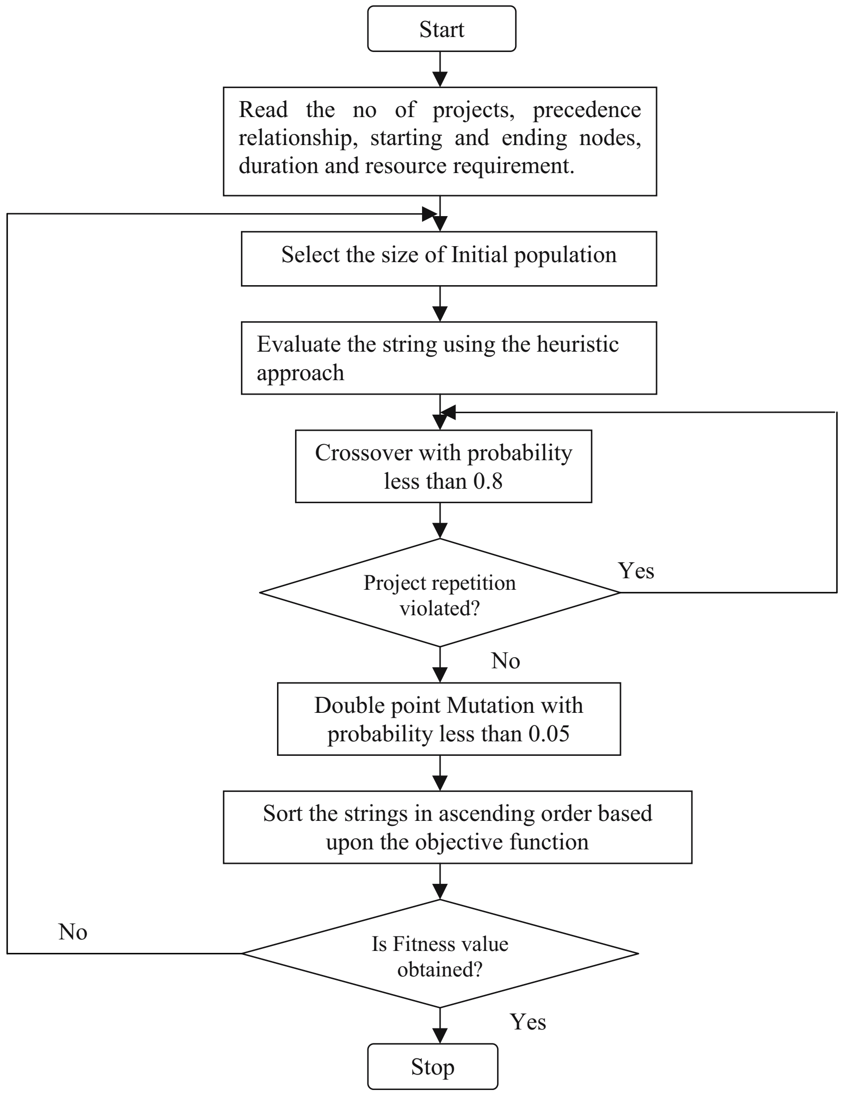


Fig. 1 Proposed genetic algorithm for multiproject scheduling

resource constrained project scheduling with pre-emption and decision nodes. Most of the researchers have mentioned single project scheduling with resource constraints using heuristic and non-traditional techniques. Attention is needed for scheduling multiproject with resource constraints using non-traditional techniques.

2 Problem definition

The problem consists of the number of projects P.

- Each project consists of N activities; activities are labeled form 1 to N, with activity N being the unique terminal job without successors.
- Activity i may be performed in any one mode j =1. Each job will have a specific mode and must be finished without changing mode.
- Scheduling activity i in mode j takes d_{ij} time units (duration).
- Activity i cannot start unless all of its predecessors have been completed.
- Activity preemption is not allowable.
- There are K kinds of the resources, where resource k is available in quantity Q_k per period. Scheduling activity i in mode j use q_{ijk} resources units per period from resource K.

3 Proposed methodology

The proposed methodology for multiproject scheduling is given in Fig. 1. The details of the genetic coding, evaluation, crossover and mutation for the proposed algorithm are as follows.

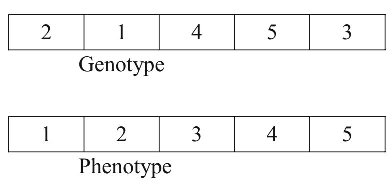
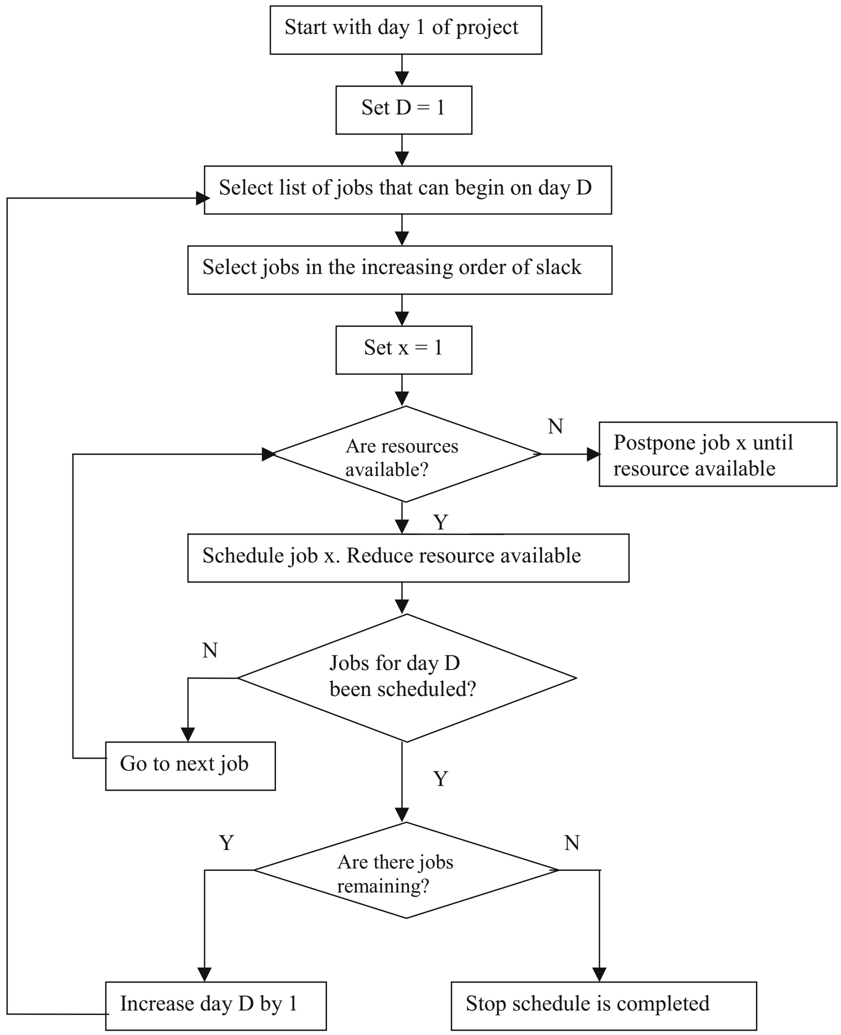


Fig. 2 Genetic coding for the proposed methodology

Fig. 3 Proposed heuristic for scheduling multi project



Genetic coding The scheme of the coding of string is given in Fig. 2. Each chromosome or string denotes a project sequence or scheduling order for resource allocation. In the example shown the second project is prioritized first, first project second, fourth project third, and so on.

Initial population The strings are generated randomly up to the desired number, which is enough to get an optimum solution. The population size is set as twice the number of projects under study or the minimum of 20 whichever is

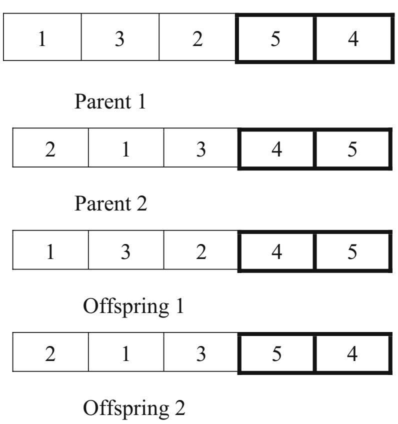


Fig. 4 Crossover operations for proposed GA

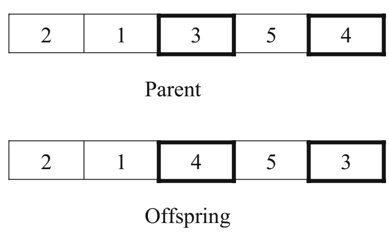


Fig. 5 Mutation operation for proposed GA

Table 1 Details of the nodes and the activities for all projects P1, P2, P3, P4 & P5

Project	Total nodes	Total activities
P1	6	10
P2	9	11
P3	10	12
P4	5	6
P5	8	10

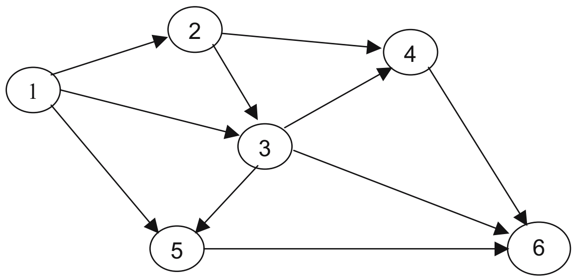


Fig. 6 Precedence network diagram of project–P1

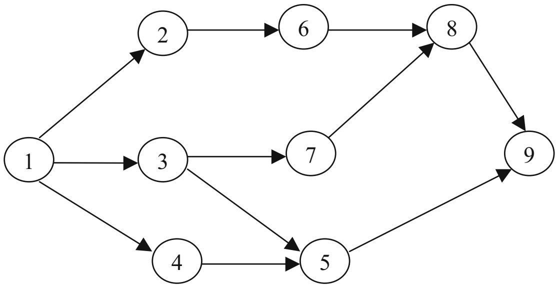


Fig.7 Precedence network diagram of project–P2

maximum. In this way a random and diverse generation of schedules can be produced up to the size of initial population.

Fitness function Fitness function is the driving force of the Darwinian natural selection and, likewise, of GAs. The accuracy of the fitness function with respect to the application is crucial for the quality of the results produced by the GA. The fitness function for the proposed GA is the make span of all projects which must be minimum.

Evaluation Evaluation is the process of evaluating the string in order to obtain the objective value. The evaluation procedure is developed as heuristic as given in Fig. 3. The steps involved are, initially set the day as one and select all the activities in the projects to be done in that particular day. Allocate the available resources as per the rank in the chromosome. If the resources are not available to perform some activities then postpone those activities. Suppose the same resource is required to perform two activities in a project then give preference to less slack activity. After the end of the first day, increase the day to

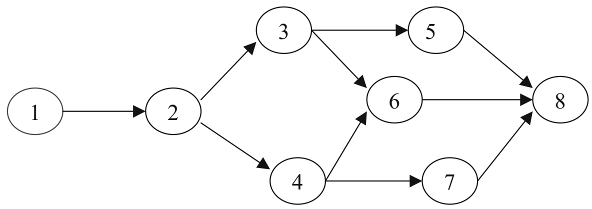


Fig. 9 Precedence network diagram of project–P4

next day and perform the same until completion of all activities of all projects.

Crossover Crossover is the process of generating new strings with new characters from the available parent strings. The details of the crossover operation are shown in Fig. 4. In this present approach each string in the reproduction population is subjected to crossover operation with a specified probability. The scheme of the crossover operator is depicted as follows: Select two good parent strings randomly. Choose a crossover point for parents randomly. Conduct the cross over such that the offspring will have the representation of all the projects.

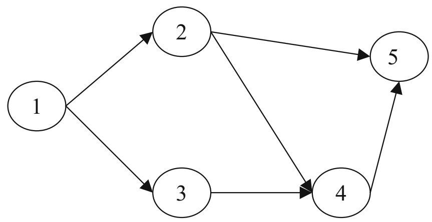


Fig. 10 Precedence network diagram of project–P5

Mutation Mutation is a process of randomly modifying the string with a probability. This helps to protect the loss of some good solution than the obtained one from cross over. The mutation process is explained in Fig. 5. Select the string to perform mutation and then set the mutation site for interchanging the genes to get new child.

After performing the mutation the strings are evaluated and are arranged in ascending order to select the best string. Finally from all the strings select the string, which is having the optimum solution. Then subject this optimum sequence to obtain the desired schedule that will be the optimum multi project schedule.

Fig. 8 Precedence network diagram of project–P3

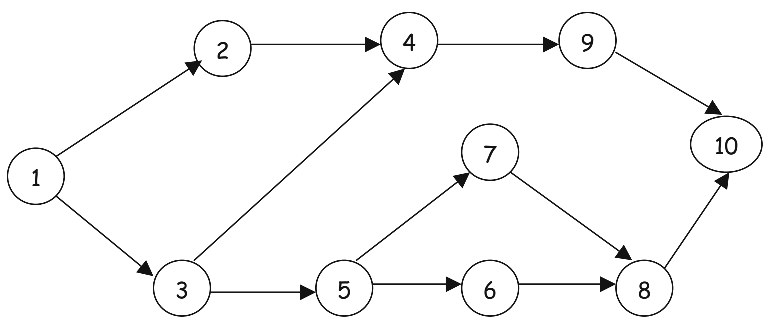


Table 2 Details of activity, duration and resources used for project–P1

Activity	1–2	1–3	1–5	2–3	2–4	3–4	3–5	3–6	4–6	5–6
Duration	8	7	9	4	10	3	5	10	7	4
Resource	6	8	3	5	1	10	2	9	4	7

Table 3 Details of activity, duration and resources used for Project – P2

Activity	1–2	2–3	2–4	3–5	3–6	4–6	4–7	5–8	6–8	7–8
Duration	5	3	2	4	1	3	6	4	2	1
Resource	3	2	4	5	2	4	6	8	2	5

Table 4 Details of activity, duration and resources used for Project–P4

Activity	1–2	1–3	1–4	2–6	3–5	3–7	4–5	5–9	6–8	7–8	8–9
Duration	2	2	1	4	8	5	3	5	2	4	3
Resource	8	9	2	5	7	6	4	2	3	5	6

Table 5 Details of activity, duration and resources used for project–P4

Activity	1–2	1–3	2–4	3–4	3–5	4–9	5–6	5–7	6–8	7–8	8–10	9–10
Duration	4	1	1	1	6	5	4	8	1	2	5	7
Resource	7	8	4	1	9	2	7	5	3	1	6	10

Table 6 Details of activity, duration and resources used for project–P5

Activity	1–2	1–3	2–4	2–5	3–4	4–5
Duration	3	7	2	5	4	2
Resource	5	2	7	3	2	8

```
 PROJECT - 5          PROJECT - 2           PROJECT - 4           PROJECT - 3                    PROJECT - 1
 -----------          -----------           -----------           -----------                    -----------

DAY  1->  1 2 * * * *   1 2 3 * * * * * * * * *   1 * * * * * * * * *   1 2 * * * * * * * * * * *     1 2 3 * * * * * * *
DAY  2->  1 2 * * * *   1 2 * * * * 7 * * * *     1 * * * * * * * * *   1 * * 4 5 * * * * * * * *     1 *  ** * * * * * *
DAY  3->  1 2 * * * *   * * * 4 5 6 7 * * * *     1 * * * * * * * * *   1 2 * * * * * * * * * * *     1 * * * * * * * * *
DAY  4->  * 2 3 4 * *   * * * * 5 6 7 * * * *     1 * * * * * * * * *   * * * * 5 * * * * * * * *     * 2 * * * * * * * *
DAY  5->  * 2 3 4 * *   * * * * 5 6 * * * * *     * * * * * * * * * *   * * 3 * 5 * * * * * * * *     * 2 * * * * * * * *
DAY  6->  * 2 * 4 * *   * * * 4 * 6 * * * * *     1 2 3 * * * * * * *   * * * * 5 6 * * * * * * *     1 2 3 * * * * * * *
DAY  7->  * 2 * 4 * *   * * * 4 5 6 * * 9 * *     * * 3 * * * * * * *   * * * * 5 * * * * * * * *     * 2 * * * * * * * *
DAY  8->  * * * 4 5 *   * * 3 4 5 * * * * 10 *  * * * * * * 6 7 * * *   * * * * 5 * 7 8 * * * * *     * 2 * * * * * * * *
DAY  9->  * * * * 5 *   * * * 4 5 * * * * * *     * * * 4 5 6 7 * * *   1 * * * 5 * * * * * * * *     * 2 * 4 5 * * * * *
DAY 10->  * * * * 5 *   * * * * 5 * * * * * *     * * * * * 6 7 * * *   * * * * * * * * * * * * *     * 2 * * 5 * * * * *
DAY 11->  * * * * 5 *   * * * * 5 * * 8 * * *     1 * * * * * 7 * 9 *   * * * * * * * * * * * * * 12  1 * * * 5 * * * * *
DAY 12->  * * * * * 6   * * 3 * 5 * * * 9 10 11 1 * * * * * * 7 * * *   * * * * * * * * * 9 * * 12    * * * * 5 * * * * *
DAY 13->  * * * * * 6   * * * * 5 * * * 9 10 11 1 2 * 4 5 * 7 8 * *     * * * * * * * * * * * * * 12  * * * * 5 6 7 8 * *
DAY 14->  * * * * * *   * * * * 5 * * * * 10 11 * 2 * * * * 7 * * 10    * * * * * * * * * * * * * 12  * * * * 5 * * 8 * *
DAY 15->  * * * * * *   * * * * * * * 8 * 10 *  1 2 * * * * * * 8 * *   * * * * * * * * * * * * * 12  * * * * 5 * * 8 * *
DAY 16->  * * * * * *   * * * * * * * 8 * * *     1 * * * 5 * * 8 * *   * * * * * 6 7 8 * 10 * 12     * * 3 * 5 * * 8 * 10
DAY 17->  * * * * * *   * * * * * * * 8 * * *     1 * * 4 * * 7 8 * *   1 * * * * * 7 8 * 10 * 12     1 * * 4 5 * 7 8 * *
DAY 18->  * * * * * *   * * * * * * * 8 * * *     1 2 * 4 * * 7 8 * 10 1 * * * * * * * * * 11 *      * * * * 5 * * 8 * *
DAY 19->  * * * * * *   * * * * * * * 8 * * *     1 * * 4 * * 7 * * *   1 * * * * * * * * 9 * * *     * * * * 5 * * 8 9 *
DAY 20->  * * * * * *   * * * * * * * * * * *     * * * 4 * * 7 * 9 *   1 * * * * * * 7 8 * * * *     * * * * 5 6 * 8 9 10
DAY 21->  * * * * * *   * * * * * * * * * * *     * * * * 5 * 7 * 9 *   * * * * * * * * * * * * *     * * 3 * 5 6 * 8 9 *
DAY 22->  * * * * * *   * * * * * * * * * * *     * * * * * * 7 * 9 10 * * * * * 6 * * * * * *       * * 3 * 5 6 7 8 9 *
DAY 23->  * * * * * *   * * * * * * * * * * *     * 2 * * * * * * * *   * * * * * * * * * * * * *     1 * 3 * * * * * 9 *
DAY 24->  * * * * * *   * * * * * * * * * * *     * 2 * * * * * * * *   * * * * * 6 7 8 9 * 11 *      1 * 3 * * * * * 9 *
DAY 25->  * * * * * *   * * * * * * * * * * *     * 2 * * * * * * * *   * * * * * * 7 8 * * 11 *      * * 3 * * * * * 9 *
DAY 26->  * * * * * *   * * * * * * * * * * *     * * * * * * * * * *   * * * * * * 7 8 * * 11 *      * * 3 * * * * * * *
DAY 27->  * * * * * *   * * * * * * * * * * *     * * * * * * * * * *   * * * * * 6 7 8 * * 11 *      * * 3 * * * * * * *
DAY 28->  * * * * * *   * * * * * * * * * * *     * * * * * * * * * *   * * * * * 6 * 8 * * 11 *      1 * 3 * * * * * * 10
DAY 29->  * * * * * *   * * * * * * * * * * *     * * * * * * * * * *   * * * * * 6 * 8 * * * *       * * 3 4 * * 7 * * 10
DAY 30->  * * * * * *   * * * * * * * * * * *     * * * * * * * * * *   * * * * * 6 * 8 * * * *       * * 3 * * * * * * 10
DAY 31->  * * * * * *   * * * * * * * * * * *     * * * * * * * * * *   * * * * * 6 * 8 * * * *       * * * * * * * * * 10
DAY 32->  * * * * * *   * * * * * * * * * * *     * * * * * * * * * *   * * * * * * * * * * * * *     * * * * * * * * * *

THE MAXIMUM TIME REQUIRED TO SCHEDULE ALL THE 5 PROJECTS WITH RESOURCE CONSTRAINT IS= 31 days
```

Fig. 11 A multiproject schedule generated by the proposed method

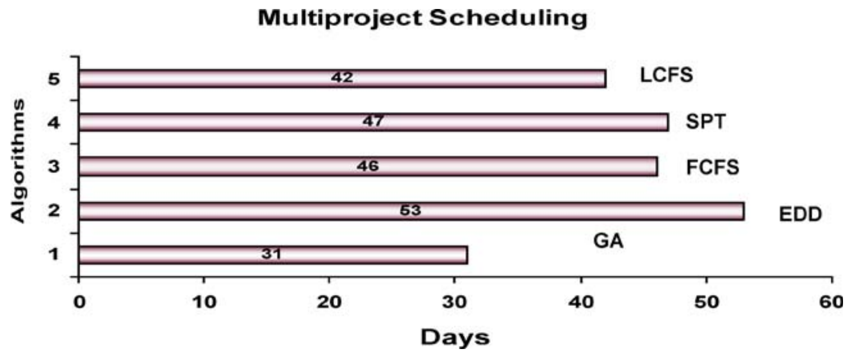


Fig. 12 Comparison of schedules generated by proposed GA and other algorithms

4 Numerical example

A case study from an engineering industry is given below. The manger has to schedule five projects concurrently namely P1, P2, P3, P4, & P5 with 10, 12, 11, 6, & 10 activities as shown in Table 1. Figs. 6, 7, 8, 9, and 10 describes the network of projects and Tables 2, 3, 4, 5, and 6 gives the details about the activities, duration and resource required for performing each activity of all projects. The total number of resources available in the organization is 10 and one in each variety. With these data the manager has to schedule the problem in an effective way and with less effort. Details of nodes and activities for all projects P1, P2, P3, P4, & P5 are given in Table 1. The schedule by the proposed method is given in Fig. 11. The comparison of the proposed algorithm with other heuristics approach is given in Fig. 12. The proposed GA approach is better than the Heuristic approach with four different priority rules.

5 Conclusion

This paper proposes the use of a heuristic and a genetic algorithm for scheduling a multi-project environment with an objective to minimize the makespan of the projects. Most of the techniques developed in the past favored scheduling a single project or multiproject represented as a single project. Mathematical techniques for scheduling project activities with constraints are extremely cumbersome and become seldom possible for multiproject. Heuristic procedures and scheduling rules are aimed at the development of good solutions but inconsistent so non-traditional techniques need attention. Hence a genetic algorithm with a heuristic approach has been proposed to solve the multi project-scheduling problem with resource constraints and the use of genetic algorithm with a heuristic proves to be simple and efficient.

References

1. Bowers MR, Groom K, Morris R (1996) A practical applications for a multi project scheduling heuristic. Prod Inventory J 37(4):19–24
2. Badiru BA (1996) Project management in manufacturing and high technology operations. Wiley, New York
3. Weist JD (1996) Heuristic programs for decision making. Harvard Bus Rev Sept−Oct
4. Davies EW (1973) Project scheduling under resource constraints: Historical review and categorization of procedures. AIIE Trans 5(4):297–313
5. Spreacher A, Hartrmann S, Drexl A (1997) An exact algorithm for project scheduling with multiple modes. OR Spectrum 19: 195–203
6. Elsayed EA, Nasr NZ (1986) Heuristic for resource constrained scheduling. Int J Prod Res 24(2):299–310
7. Demeulemeester SH, Herroelen WS (1996) An efficient optimal solution procedure for the pre-emptive resource constrained scheduling problem. Eur J Oper Res 90:334–348
8. Boctor FF (1990) Some efficient multi-heuristic procedures for resource constrained project scheduling. Eur J Oper Res 49: 3–13
9. Boctor FF (1993) Heuristics for scheduling projects with resource restrictions and several duration modes. Int J Prod Res 31(11):2547–2558
10. Boctor FF (1996) A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes. Eur J Oper Res 90:349–361
11. Kolisch R, Drexl A (1993) Local search for non-preemptive multimode resource constrained project scheduling. Christian-Albrechts-Unversitat-zu Kiel, Kiel, Germany, working paper
12. Kolisch R, Sprecher A, Drexl A (1995) Characterization and generation of a general class of resource constrained project scheduling problems. Manage Sci 41:1693–1730
13. Talbot FB (1982) Resource constrained project scheduling with time trade-offs: The non-preemptive case. Manage Sci 28: 1197–1210
14. Drexl A, Grunewald J (1993) Non-preemptive multi-mode resource constrained project scheduling. IIE Trans 25:74–81
15. Holland JH (1976) Adaptation in nature and artificial systems. Univ Michigan Press, Ann Arbor
16. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA
17. Croce ED, Tadei R, Volta G (1995) A genetic algorithm for the job shop problem. Comput Oper Res 22(1):15–24
18. Reeves CR (1995) A genetic algorithm for flow shop sequencing. Comput Oper Res 22(1):5–13
19. Bruns R (1993) Direct chromosome representation and advanced genetic operators for production scheduling. In: (Forrest S ed.) Proc Fourth International Conference of Genetic Algorithms, Morgan Kaufmann Publishers, CA, pp 352–359
20. Portmann MC (1996) Genetic algorithms and scheduling: A state of the art and some proportions. Proc Workshop Production Planning Contr., Mons
21. Masoa M, Ching CT (1997) A genetic algorithm for multi-mode resource constrained scheduling problem. Eur J Oper Res 100:134–141

22. Leu S-S, Yang C-H (1999) GA based multicriteria optimal model for construction scheduling. J Constr Eng Manage 125: 420–427
23. Hegazy T (1999) Optimization of resource allocation and leveling using Genetic algorithm. J Constr Eng Manage 125:167–175
24. Reddy JP, Kumanan S, Chetty OVK (2001) Application of Perti-nets and a genetic algorithm to multi-mode multi-resource constrained project scheduling. Int J Adv Manuf Technol 17: 305–314