

Solving a mixed-integer linear programming model for a multi-skilled project scheduling problem by simulated annealing

H. Kazemipoor^{*a}, R. Tavakkoli-Moghaddam^b and P. Shahnazari-Shahrezaei^c

^aDepartment of Industrial Engineering, Islamic Azad University-Science and Research Branch, Tehran, Iran

^bDepartment of Industrial Engineering, College of engineering, University of Tehran, Tehran, Iran

^cDepartment of Industrial Engineering, Islamic Azad University- Firoozkooh Branch, Firoozkooh, Iran

ARTICLE INFO

Article history:

Received July 20, 2011

Received in Revised form

October, 12, 2011

Accepted 15 October 2011

Available online

20 October 2011

Keywords:

Project scheduling

Simulated annealing

Mixed integer linear programming

ABSTRACT

A multi-skilled project scheduling problem (MSPSP) has been generally presented to schedule a project with staff members as resources. Each activity in project network requires different skills and also staff members have different skills, too. This causes the MSPSP becomes a special type of a multi-mode resource-constrained project scheduling problem (MM-RCPSP) with a huge number of modes. Given the importance of this issue, in this paper, a mixed integer linear programming for the MSPSP is presented. Due to the complexity of the problem, a meta-heuristic algorithm is proposed in order to find near optimal solutions. To validate performance of the algorithm, results are compared against exact solutions solved by the LINGO solver. The results are promising and show that optimal or near-optimal solutions are derived for small instances and good solutions for larger instances in reasonable time.

© 2012 Growing Science Ltd. All rights reserved.

1. Introduction

Project scheduling, which involves the planning of project activities over time by taking some constraints into account in order to minimize or maximize some objectives, is a crucial task in project management. For this reason, the project scheduling problem has been a popular research area over the years. The resource-constrained project scheduling problem (RCPSP) is one of the most complicated project scheduling problems. For the first time Blazewicz et al. (1983) showed that the RCPSP is a generalization of the classical job shop scheduling problem, belonging to NP-hard optimization problems. For the first time, the multi-skilled project scheduling problem (MSPSP), which is an expansion of RCPSP, has been presented by Neron and Boptista (2002). As a general rule, this problem is an extended model of the multi-mode RCPSP (MM-RCPSP) (Bellenguez-Morineau & Neron, 2005).

The main differences between the MSPSP and classical project scheduling problems are in the resources and the requirement type of each resource. In the MSPSP, the resources are staff members

* Corresponding author. Tel:+98 21 88021067

E-mail address: h.kazemipoor@piau.ac.ir (H. Kazemipoor)

with different skills assigned to the activities of project. A staff member can be assigned at most to one requirement in a determined time to perform an activity in one required skill. Different combinations of usage of these staff members are determined during the project assignment and on the basis of required skills of each activity and skills of a staff member. As a matter of fact, to accomplish each activity in any skill, the proper resources are selected from a set of qualified resources. In the other words, if all staff members in the whole project have a single skill, the MSPSP will be the same as a classic RCPSP.

This fact that resources can be used to satisfy different requirements has been presented in the definition of job-shop scheduling (Dauzere-peres et al., 1996; Jurish, 1992). Taking the related definitions with the given problem into account, the MSPSP can be classified into the category of the multi-mode RCPSP (MM-RCPSP) with very large modes. This issue causes that the MSPSP cannot be solved even in medium sizes by the available solution methods of MM-RCPSP that is often based on the enumeration (Neron & Boptista, 2002). Regarding these considerations, the new techniques have been proposed to solve this type of problem (Artigues, 2008). Bellenguez-Morineau (2008) summarized several proposed methods. They mentioned three methods to solve the problem: lower bounds, different heuristics and meta-heuristics and finally a branch and bound procedure. Tabu search (TS) and two genetic algorithms (GA) are proposed as meta-heuristics, it is concluded that for the problem genetic algorithms are dominated by TS.

The non-preemptive MSPSP with hierarchical levels of skills, which minimizes makespan including ten activities, was considered by Bellenguez and Neron (2005). They proposed two lower bounds to evaluate a heuristic to be used in a Branch-and-Bound algorithm. Another sample in this direction is the work carried out by Kadro and Najid (2006). They extended a solution method by TS accompanied by strong local search for a manpower constrained MM-RCPSP. In order to exhibit the strength of the algorithm, they also found a lower bound for the problem and compared the generated solutions with this bound. Valls et al. (2009) proposed a solution to the skilled workforce project scheduling problem. They classified worker in three levels, as senior, standard, and junior. They have also more than one objective that deal with the criticality levels of activities, balanced worker loads, and efficient assignment of workers. Then, they recommended a hybrid GA to solve a skilled manpower scheduling problem.

Heimerl and Kolisch (2009) proposed a mixed-integer linear programming (MILP) model with hard constraints in order to minimize the manpower cost considering the in-source and out-source manpower for information technology projects portfolio. Gutjahr et al. (2008) took a project portfolio into account and introduced a non-linear mixed-integer programming model on the basis of maximizing the average economic income for selecting the projects, optimizing the time, and assigning the people to selected projects. They presented a heuristic process, a greedy heuristic algorithm to schedule and assign the people, and a meta-heuristic algorithm to choose the project.

They compared their solutions with the bounds acquired by the exact solution of the simplified mathematical model and proved the efficiency of the heuristic. Gürbüz (2010) considered two objectives, which are the makespan and the total skill wasted. While trying to minimize the makespan, minimizing the total skills wasted is aimed. By the second objective over-qualification for the jobs is minimized in order to prevent job dissatisfaction. He solved the problem using a multi-objective GA, namely NSGA-II, compared the algorithm with exact results available for small problems, and with randomly generated solutions for larger problems. Pessan et al., (2007) formulated a preventive maintenance problem with human skills as an MSPSP.

This paper is organized as follows. Section 2 deals with used parameters, decision variables, and mixed-integer linear programming (MILP). An efficient simulated annealing algorithm to solve the problem is presented in Sections 3. Section 4 is allotted to computational results. Finally, Section 5 summarizes the concluding remarks.

2. MILP for MSPSP

In this section, the mixed integer linear programming is presented. The model is an extension of that of Gürbüz (2010) for the MSPSP. In what follows model requirements are first presented, and then the MILP is considered. In this paper, it is assumed that all precedence networks are based on activity on node (AON).

2.1. Model Requirements

- i, i' Activities counters (1,...,I)
- a Skill counters (1,...,A)
- k Time counters (1,...,K)
- j Resources counters (1,...,J)
- d_i Time of executing activity i .
- b_{ik} Number of required resources for doing skill S_k during the execution of activity A_i
- $x_{jik} = \begin{cases} 1 & \text{If person } j \text{ performs activity } i \text{ at day } k \\ 0 & \text{Otherwise} \end{cases}$
- $y_{ji}^a = \begin{cases} 1 & \text{If person } j \text{ performs activity } i \text{ at skill } a \\ 0 & \text{Otherwise} \end{cases}$
- $z_{ik}^a = \begin{cases} 1 & \text{If activity } i \text{ at skill } a \text{ finishes at day } k \\ 0 & \text{Otherwise} \end{cases}$
- FT_k Finish time of activity k in all skills

2.2. The MILP for the MSPSP

The MILP model for the MSPSP is presented below.

$$\min FT_N \tag{1}$$

subject to

$$\sum_{k=1}^K kz_{nk}^a \leq FT_i \tag{2}$$

$$\sum_{i=1}^N x_{jik} \leq 1; \forall (i, k) \tag{3}$$

$$x_{jik} \leq y_{ji}^a; \forall (j, i, k, a) \tag{4}$$

$$\sum_{k=1}^K \sum_{j=1}^J x_{jik} = \sum_{j=1}^J y_{ji}^a; \forall (i, a) \tag{5}$$

$$\sum_{k=1}^K z_{ik}^a = 1; \forall (i, a) \tag{6}$$

$$\sum_{k=1}^K x_{jik} = \sum_{t=k}^{k+d_i-1} z_{ik}^a; \forall (i, a); \forall (j, i, t \in K) \tag{7}$$

$$\sum_{i=1}^N r_{ja} y_{ji}^a = b_{ia}; \forall (i, k) \tag{8}$$

$$\sum_{k=1}^K kz_{ik}^a \geq \sum_{k=1}^K (k - d_{i'}) z_{i'k}^a; \forall (a, (i, i') \in \{\text{Precedence Relations}\}) \tag{9}$$

$$x_{jik} \in \{0,1\}; \forall (j, i, k) \tag{10}$$

$$y_{ji}^a \in \{0,1\}; \forall (j, i, a) \tag{11}$$

$$z_{ik}^a \in \{0,1\}; \forall (i, k, a) \tag{12}$$

$$FT_i \geq 0 \tag{13}$$

In the model, the objective function minimizes the finish time of the last activity. Finish time of each activity is defined by maximum finish time of an activity in all skills. This is guaranteed by constraints (2). Constraints (3) ensure that each worker is assigned to at most one activity in one skill in a day. Constraints (4) define the relationship between two variables; the worker is assigned to a job on a day if and only if he/she is assigned to that activity during the whole duration of that job. Constraints (5) ensure the number of days that a worker assigned to an activity is equal to the activity's duration. Constraints (6) oblige each activity to be completed once in each skill. By constraints (7) daily assignments made to the activities in the skill are constrained by the completion time of activities. In other words, the day that a worker assigned to an activity should be within the starting and ending day of that activity. Constraints (8) ensure the skill requirements of the activities in each skill are satisfied. Constraints (9) enforce the precedence relationships between the activities. Finally, constraints (10)-(13) define the domains of the decision variables. Since the proposed model still has time complexity, Section 3 provides two meta-heuristics to solve the model.

3. A simulated annealing meta-heuristic for MSPSP

Simulated Annealing (SA) is a stochastic search algorithm based on the “annealing” concept. The annealing process consists of first raising the temperature of a solid to a point where its atoms can freely move and then lower the temperature, forcing the atoms to rearrange themselves into a lower energy state. During this process the free energy of the solid is minimized. The cooling schedule is vital in this process. If the solid is cooled too quickly, or if the initial temperature of the system is too low, it is not able to become a crystal and instead, the solid arrives at an amorphous state with higher energy. In this case, the system reaches a local minimum instead of the global minimum (Press et al., 1989; Dowsland, 1993).

Metropolis et al., (1953) proposed an algorithm to simulate the evolution of a solid in a heat bath until it reaches its thermal equilibrium. The Monte Carlo method was used to simulate the process, which started from a certain thermodynamic state of the system, defined by a certain energy and temperature. Then, the state was slightly perturbed. If the change in energy produced by this perturbation was negative, the new configuration was accepted. If it was positive, it was accepted with a probability given by $\exp(\delta/kT)$, where k is the so-called Boltzmann constant, which is a constant of nature that relates temperature to energy (Press et al., 1989). This process is repeated until a frozen state is achieved (Sait, 1999).

Thirty years after the publication of Metropolis' approach, Kirkpatrick et al., (1983) and Cerny (1985) independently pointed out the analogy between this “annealing” process and combinatorial optimization.

-
1. Choose an initial feasible solution for the MSPSP by feasible solution generation s_0
 2. Select an initial temperature $t=t_0 > 0$
 3. Repeat
 - Repeat
 - Randomly select $s \in N(s_0)$ (N = neighborhood to improve a solution)
 - $\delta = f(s) - f(s_0)$ (f = objective function)
 - If $\delta < 0$, then $s_0 \leftarrow s$. Else, generate random x with the uniform distribution in $[0, 1]$
 - If $x < \exp(-\delta/t)$ then $s_0 \leftarrow s$
 - Until maximum iteration
 - $t_{new} = \alpha \times t_{old}$
 4. Until the stopping condition is met
-

Fig. 1. The proposed SA algorithm for the MSPSP

These researchers indicated several important analogies: a system state is analogous to a solution of the optimization problem; the free energy of the system corresponds to the cost of the objective function to be optimized; the slight perturbation¹ imposed on the system to change it to another state corresponds to a movement into a neighboring position; the cooling schedule corresponds to the control mechanism adopted by the search algorithm; and the frozen state of the system corresponds to the final solution generated by the search algorithm. These important analogies led to the development of an algorithm called simulated annealing (SA). The proposed SA is for solving the MSPSP is depicted in Fig. 1.

3.1. Feasible solution generation

In this research, a bi-directional scheduling generation scheme (Seifi, 2008) is used to generate initial solutions. In this method, a forward and backward partial scheduling is performed simultaneously and at any iteration, an activity can be scheduled when it is just schedulable in one direction. In this process, three sets of activities are defined, namely 1) the set of activities that their precedent activities have been scheduled, 2) the set of activities that their succeeding activities have been scheduled, and 3) the set of activities that both their precedent and succeeding activities have been scheduled. Based on the descriptions of the sets, Fig. 2 shows the pseudo code of this procedure.

-
1. Repeat the following for $N=50$ times
 - 1.1. Make the sets 1, 2, and 3 for the project.
 - 1.2. Repeat the following steps for the selected project until emptying the sets 1 and 2.
 - 1.2.1. In the case of being non-empty the set 1: Choose an activity from the set 1 of the selected project randomly and schedule it at the earliest time
 - 1.2.2. In the case of being non-empty the set 2: Choose an activity from the set 2 of the selected project randomly and schedule it at the latest time
 - 1.2.3. Update all sets
 - 1.3. In order to determine the final project scheduling for each solution, add backward scheduled activities to the set of forward scheduled activities for the project against their selection order.
 2. Find the best feasible solution.
 3. End
-

Fig. 2. Initial feasible solution for the MSPSP

3.2. Neighborhoods structure

To improve initial solutions (see Fig. 1), two neighborhood structures are developed:

- **Neighborhood structure 1:** In this procedure, two activities' indices that can be started simultaneously are selected randomly and displaced, considering the precedence relations. Resources to the relevant skills of these activities are also changed randomly.
- **Neighborhood structure 2:** Index a is randomly generated in the interval $[1 \dots N]$. b is the index of the first preceding activity existing in the a^{th} cell and c is the index of the last precedence activity existing in the a^{th} cell. Index d is also generated randomly in the interval $[b \dots c]$. If d is less than a , the activity existing in the a^{th} cell is placed in the d^{th} cell. Else, the activity existing in the a^{th} cell is placed in the $(d-1)^{\text{th}}$ cell.

4. Computational results

Since there is no special instance set in the field of MSPSP (Artigues et al., 2008), several instances are generated randomly with different number of activities and features, listed in Table 1.

Table 1

The parameters of MSPSP generated instances

| Parameter | Value |
|-------------------------|---|
| r_{mk} | $D \sim \text{Uniform}(0,1)^*$ |
| b_{ik} | $D \sim \text{Uniform}(0,6)$ |
| p_i | $D \sim \text{Uniform}(3,15)$ |
| Number of skills | $D \sim \text{Uniform}(3,7)$ |
| Number of staff members | 6 to 30 (depends on the number of activities) |

In order to evaluate the developed SA algorithm further, the MILP is solved optimally for small instances using LINGO 9 solver by LINDO Inc. It should be noted that in order to determine the parameters of SA algorithm, a trial and error method has been used. These parameters have been chosen so that the solutions acquired by the SA are as close as possible to the optimal solutions of solvable small instances. These parameters are as follow.

$$\alpha=0.95$$

Initial temperature = Objective function of initial solution obtained (see Fig. 2).

In this paper, the MSPSP is coded by the Matlab R2009a and all computational experiments were carried out on a Pentium 4 PC with 4 GB of memory and 2.20 GHz of CPU. Results of small instances are shown in Table 2. As shown in Table 2, LINGO took considerably long time to derive optimal solutions for small instances. The proposed SA, however, produces the same results taking less time than LINGO (these were shown bold in Table 2). In fact, the results of Table 2 demonstrate the proposed SA can produce very high quality and near optimal solutions. Regarding these comments, the problem in larger and more real sizes is presented in Table 3. It is worthy to note that efficient procedures have been developed in this study to solve the real size MSPSP exactly by mathematical programming, and heuristically by SA meta-heuristic.

Table 2

Computational results of small instances

| Number of (activities, skill, worker) | Results (ten runs) | | | | LINGO | | GAP (%) |
|--|-------------------------|-------------|------------|------------------|-------|-------|------------|
| | SA | | | | OFV | Time | |
| | Min OFV ¹ | Ave. OFV | Max OFV | Ave. time (s) | | | |
| (10,3,6) | 278 | 278 | 278 | 9 | 278 | 765 | 0 |
| (10,4,7) | 281 | 281 | 281 | 11 | 281 | 581 | 0 |
| (10,5,6) | 285 | 285 | 285 | 12 | 285 | 654 | 0 |
| (20,6,8) | 601 | 601 | 601 | 32 | 601 | 21701 | 0 |
| (20,5,8) | 593 | 595 | 598 | 25 | 593 | 21085 | 0.03 |
| (20,3,9) | 582 | 582 | 582 | 23 | 582 | 23073 | 0 |
| (20,7,7) | 681 | 681 | 681 | 39 | 681 | 24011 | 0 |
| (20,5,6) | 584 | 584 | 584 | 31 | 584 | 22431 | 0 |

5. Conclusion

In information technology projects, research projects, engineering activities, preventive maintenance activities, and the alike projects, staff members should be assigned to project activities considering their skills and the precedence network of projects. As it was mentioned before, this problem has

¹ Objective function value (OFV)

become famed under the multi-skilled project scheduling problem (MSPSP). Since the MSPSP is a special type of the multi-mode resource-constrained project scheduling problem (MM-RCPS) with huge modes that has attracted less attention of a number of researchers, a novel linear mixed-integer programming (MILP) model has been presented for the problem in this study, and was solved by an efficient simulated annealing algorithm. To verify the efficiency of the proposed algorithm, the generated random problems have been solved and compared with the optimal solutions obtained by LINGO solver.

Table 3

Computational results for medium and large instances of running SA after ten times

| #of Activity | #of worker | Initial feasible solution | Min OFV | Ave. OFV | Max OFV | Ave. time(s) |
|--------------|------------|---------------------------|---------|----------|---------|--------------|
| 30 | 10 | 1351 | 1323 | 1323 | 1323 | 41 |
| | 10 | 1241 | 1208 | 1208 | 1208 | 48 |
| | 11 | 1102 | 1002 | 1002 | 1002 | 55 |
| | 12 | 890 | 871 | 871 | 871 | 52 |
| | 12 | 909 | 900 | 901 | 902 | 43 |
| 50 | 15 | 1287 | 1258 | 1258 | 1258 | 53 |
| | 15 | 1449 | 1389 | 1401 | 1421 | 49 |
| | 16 | 1431 | 1300 | 1307 | 1331 | 61 |
| | 16 | 1579 | 1371 | 1391 | 1421 | 55 |
| | 16 | 1441 | 1299 | 1299 | 1299 | 47 |
| 100 | 20 | 1812 | 1501 | 1501 | 1501 | 291 |
| | 20 | 1744 | 1465 | 1481 | 1512 | 286 |
| | 20 | 1902 | 1631 | 1636 | 1652 | 301 |
| | 22 | 1611 | 1571 | 1571 | 1571 | 322 |
| | 22 | 1642 | 1498 | 1498 | 1498 | 299 |
| 150 | 25 | 2421 | 2201 | 2201 | 2201 | 511 |
| | 25 | 2579 | 2181 | 2185 | 2193 | 487 |
| | 25 | 2581 | 2167 | 2173 | 2187 | 491 |
| | 30 | 2001 | 1960 | 1967 | 1986 | 691 |
| | 30 | 2274 | 1921 | 1921 | 1921 | 707 |

References

- Artigues, C., Demasse, S., & Neron, E. (2008). *Resource constraint project scheduling problem*. John Wiley and Sons.
- Blazewicz, J., Lenstra, J.K. & Rinnooy Kan, A.H.G. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, 5, 11–24.
- Bellenguez-Morineau, O. Neron, E. (2005). Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills, in: E.K. Burke and M. Trick, Eds., *Practice and theory of automated timetabling*. *Lecture Notes in Computer Science*, 3616, 229–243.
- Bellenguez-Morineau, O. (2008). Methods to solve multi-skill project scheduling problem, *4OR*, 6, 85–88.
- Cerny, V. A. (1985). Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications*, 45(1), 41–51
- Dauzere-peres, S., Roux, W., & Lasserre, J.B. (1996). Multi-resource shop scheduling with resource flexibility. *European Journal of Operation Research*, 107, 289–305.
- Demeulemeester, E.L., & Herroelen, W.S. (2002). *Project scheduling: A research handbook*, Kluwer Academic Publisher.
- Dowland, K.A. (1993). *Simulated Annealing*. In C. R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons
- Gutjahr, W.J., Katzensteiner, S., Raiter, P., Stummer, & C., Denk, M. (2008). Competence-driven project portfolio selection scheduling and staff assignment. *Central European Journal of Operation Research*, 16, 281–306.

- Gürbüza E. (2010). Genetic algorithm for bi-objective multi-skill project scheduling problem with hierarchical levels of skills, Thesis on industrial engineering department. Middle East Technical University, Turkey
- Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165, 289-306.
- Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207, 1-14.
- Heimerl, C., Kolisch, R. (2009). Scheduling and staffing multiple project with a multi-skilled workforce. *OR Spectrum*, 32 (2), 343-368.
- Jurish, B. (1992). Scheduling jobs in shops with multi-purpose machines, PhD Thesis, University of Osnabruck, Germany.
- Kadrou, Y., & Najid, N.M. (2006). Tabu search algorithm for the MRCPSP with multi-skilled personnel, *Computational Engineering in System Application, IMACS Multi Conference*, 2(4-6), 1302-1309.
- Kirkpatrick, S., Gellatt, C., & Vecchi., M. (1983). Optimization by Simulated Annealing. *Science*, 220 (4598), 671-680.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M. , Teller, A., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6), 1087-1092.
- Neron, E., & Boptista, D. (2002). Heuristics for the multi-skill project scheduling problem, *International Symposium on Combinatorial Optimization*, Paris.
- Pessan, C., Bellenguez-Morineau, O., & Neron, E. (2007). Multi-skill project scheduling problem and total preventive maintenance, *Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA)*, 608-610, Paris, 28-31 August.
- Press, W. H. Flannery B. P., Teukolsky S. A., & Vetterling W. T. (1989). Numerical Recipes in Pascal. Cambridge University Press, Cambridge, UK
- Sait S. M., Youseff H., & Ali H. (1999). Fuzzy Simulated Evolution Algorithm for Multi-objective Optimization of VLSI Placement, Congress on Evolutionary Computation, Washington, D.C., July IEEE Service Center. 91-97
- Seifi, M. & Tavakkoli-Moghaddam, R. (2008). A new bi-objective model for a multi-mode resource-constrained project scheduling problem with discounted cash flows and four payment models. *International Journal of Engineering, Transaction A: Basic*, 21(4), 347-360.
- Valls, V., Perez, A., & Quintanilla, S. (2009). Skilled workforce scheduling in service centres. *European Journal of Operational Research*, 193, 791-804.