Applying the Intelligent Decision Heuristic to Solve Large Scale Technician and Task Scheduling Problems

Amy Khalfay, Alan Crispin, and Keeley Crockett

Department of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, Manchester, UK
a.khalfay@mmu.ac.uk, a.crispin@mmu.ac.uk, k.crockett@mmu.ac.uk

Abstract. Scheduling personnel to complete tasks is a complex combinatorial optimisation problem. In large organisations, finding quality solutions is of paramount importance due to the costs associated with staffing. In this paper we have generated and solved a set of novel large scale technician and task scheduling problems. The datasets include complexities such as priority levels, precedence constraints, skill requirements, teaming and outsourcing. The problems are considerably larger than those featured previously in the literature and are more representative of industrial scale problems, with up to 2500 jobs. We present our data generator and apply two heuristics, the intelligent decision heuristic and greedy heuristic, to provide a comparative analysis.

Keywords: combinatorial optimisation, large scale technician and task scheduling problems, data generator, intelligent decision heuristic

1 Introduction

The importance of finding quality solutions to scheduling problems was highlighted in [5]. There are many benefits to optimised scheduling; such as maintaining customer satisfaction and providing a balanced working schedule for employees. The importance of field service scheduling is growing due to the increasing number of machines that are used and therefore the number of specialized technicians needed to meet the demand [8]. There are many different types of scheduling problems that can be categorized by the constraints that they include. This paper focuses on generating and solving a set of large scale technician and task scheduling problems.

This problem is NP-hard, there are no known polynomial time algorithms for solving them optimally, which makes using exact methods prohibitive. In large businesses, there is also often a conflict between computational time and solution quality, whilst a solution of high quality is desired, it is desired within a reasonable computational time. The use of heuristic methods is popular for larger sized problems and has produced competitive results in many combinatorial optimization problems such as; graph colouring [17], the vehicle routing problem

[3] and nurse rostering [1]. Whilst approximate methods have no guarantee of finding a globally optimal solution, they generally produce high quality results in short computational times and are scalable and robust.

Literature in the field of technician and task scheduling problems has included solving both artificial and real world problems. Technician and task scheduling problems have been studied by [11], [12] and [14]. In addition the ROADEF 2007 challenge was based on France Telecom's technician and task scheduling problem and has attracted much research interest; [2], [6], [7], [9] and [10].

The ROADEF 2007 challenge problem used real world datasets [4]. France Telecom aimed to reduce the cost of its workforce whilst maintaining a satisfied client base and dominating the market share. This optimisation problem involves creating a set of teams over a scheduling horizon to service or outsource a set of jobs [15]. The problem includes many constraints such as technician unavailability, priority levels, outsourcing, skill requirements and precedence relationships. Instances in the ROADEF 2007 challenge ranged from 5 to 800 jobs, however, the problems that arise in industrial settings may include many more jobs to allocate.

A technician routing and scheduling problem was proposed by Pillac et al.[14]. In this work, vehicle routing problem instances proposed by Solomon [16] were extended to create a technician routing and scheduling problem. To do this, random skill requirements were created for each customer and each vehicle (technician) had intrinsic skill levels. This problem included location information and tools and spare parts constraints but did not include the complexity of teaming or precedence relationships. Instances in these datasets contained at most 100 jobs. A service technician routing and scheduling problem was created by Kovacs et al [11] which also extended instances from [16]. This work, concatenated skill domain information from the ROADEF 2007 challenge problem on to the customer information from the vehicle routing instances. Again these instances contained at most 100 jobs.

A personnel task scheduling problem was also studied by [12] who procedurally generated the datasets used. In this work, the objective was to minimize the make span of the scheduling horizon (similar to ROADEF 2007 challenge but without priority levels). A heuristic approach was compared against mixed integer programming using data instances with up to 2105 jobs. This work considered a heterogeneous workforce, however not in the same way as the ROADEF 2007 challenge, which used skill domain areas and multiple levels of skill within those domains. This research concluded that mixed integer programming was not an appropriate solution technique for large scale scheduling problems. Also the heuristic approaches tested produced quality solutions in short computational times and, most importantly, were scalable.

It appears that the ROADEF 2007 challenge problem includes the most relevant features of the problems studied in the literature. The problem includes the complexity of outsourcing which itself is an NP-hard problem, as well as skill compatibility, unavailability, precedence and priority etc. However, the problems featured range from 5 to 800 jobs. To our knowledge, there is no current literature that includes solving large scale (1000+ jobs) scheduling problems that have precedence relationships, skill requirements and teaming. In this paper, we have generated large scale technician and task scheduling problems (created under the ROADEF 2007 challenge problem definition) and evaluated our intelligent decision heuristic and greedy heuristic on the data.

This paper is structured as follows; Section 2 presents the mathematical formulation of the ROADEF 2007 challenge problem. Section 3 describes the large scale technician and task scheduling problem datasets that have been generated. Section 4 describes the two heuristics, an intelligent decision heuristic and a greedy randomized heuristic, that have been used to test the large scale data instances. Section 5 presents the experimental results and Section 6 discusses the performance of the intelligent decision and greedy heuristic. Lastly, Section 7 identifies areas for further research.

2 ROADEF 2007 Challenge Mathematical Formulation

The aim of the ROADEF 2007 challenge problem is to construct a set of teams to service a set of jobs over a scheduling horizon $K = [1...k]$. Each job i belonging to set N has certain properties, a priority level p where $p \in [1...4]$, an execution time d_i, a domain skill requirement matrix $s^i_{\delta\alpha}$ (where δ is the domain and α is the skill level), an outsourcing cost c_i and a set of successor jobs σ_i. The set of teams is denoted by $M = [1...m]$, which are made up of technicians $T = [1...t]$. The objective function set in the challenge is shown in Equation (1). The objective function is a weighted sum of the latest ending times, e_p, of each priority group where $w_p = [28, 14, 4, 1]$ for $p = [1, 2, 3, 4]$.

$$Minimize \sum_{p=1}^{4} w_p * e_p \tag{1}$$

The start times of jobs are denoted as b_i. Equation (2) ensures that the latest ending time for each priority group, $p \in [1...3]$, must be greater than, or equal to, the start time of every job plus the duration of the job.

$$e_p \geq b_i + d_i \qquad \forall p \in 1, 2, 3, i \in N_p \tag{2}$$

In addition, Equation (3) ensures the latest ending time overall e_4, is greater than, or equal to, the start time of every job plus the duration of every job belonging to the entire set of jobs.

$$e_4 \geq b_i + d_i \quad \forall, i \in N \tag{3}$$

Let $x_{t,k,m} = 1$ if technician t belongs to team m on day k. Equation (4) guarantees that if a technician is available to work i.e. belongs to the set T_k, then the technician may only be a member of one team that day.

$$\sum_{m \in M} x_{t,k,m} \leq 1 \quad \forall k \in K, t \in T_k \tag{4}$$

Conversely, Equation (5) confirms if a technician may not work i.e. does not belong to the set T_k, then the technician is not a member of any team on that day.

$$\sum_{m \in M} x_{t,k,m} = 0 \quad \forall k \in K, t \notin T_k \tag{5}$$

Let $y_{i,k,m} = 1$ if job i is assigned to team m on day k. Equation (6) states that every job belonging to the set of jobs N, must be either outsourced, $z_i = 1$, or scheduled during the scheduling horizon.

$$z_i + \sum_{k \in K} \sum_{m \in M} y_{i,k,m} = 1 \quad \forall i \in N \tag{6}$$

Equation (7) ensures that if a team is assigned a job i.e. $y_{i,k,m} = 1$, then the collective skill levels of the team are greater than or equal to the skill requirements needed to complete the job.

$$y_{i,k,m} * s^i_{\delta\alpha} \leq \sum_{t \in T_k} v^t_{\delta\alpha} * x_{t,k,m} \quad \forall i \in N, k \in K, m \in M, \alpha \in A, \delta \in D \tag{7}$$

Equation (8) deals with the precedence relationships between jobs, so that if job i' is a successor of job i, i.e. belongs to the set σ_i, i' may not begin until i has been completed.

$$b_i + d_i \leq b'_i \quad \forall i \in N, i' \in \sigma_i \tag{8}$$

Equations (9 and 10) deal with the working hours of the day. Equation (9) ensures that if a job is scheduled to begin on day k, then the start time of the job is greater than or equal to the beginning of that day. Equation (10) states that if a job is scheduled to be completed on day k then the job must be completed before the working day ends.

$$120(k-1) * \sum_{m \in M} y_{i,k,m} \leq b_i \quad \forall i \in N, k \in K \tag{9}$$

$$120(k) * \sum_{m \in M} y_{i,k,m} \geq b_i + d_i \quad \forall i \in N, k \in K \tag{10}$$

Let $u_{i,i'} = 1$ if jobs i and i' are assigned to the same team on the same day and i' begins after i is completed. Equation (11) ensures time continuity, if two jobs happen sequentially then the end time of job i is less than or equal to the start time of the job i'. Here, G is a large number to satisfy the constraint when jobs do not happen sequentially.

$$b_i + d_i - G(1 - u_{i,i'}) \leq b'_i \quad \forall i, i' \in N, i \neq i' \tag{11}$$

Equation (12) helps with the ordering of jobs. If two jobs happen sequentially then they must both be allocated to the same team and one must be scheduled before the other.

$$y_{i,k,m} + y_{i',k,m} - u_{i,i'} - u_{i',i} \leq 1 \quad \forall i, i' \in N i \neq i', k \in K, m \in M \tag{12}$$

In some problem instances of the ROADEF 2007 challenge problem there is an outsourcing budget available, C. Jobs that are outsourced do not contribute to the objective function, therefore utilization of this budget is important. Let $z_i = 1$ if job i is outsourced. Equation (13) ensures that the outsourcing budget is not exceeded.

$$\sum z_i * c_i \leq C \quad \forall i \in N \tag{13}$$

The set of jobs that are outsourced must adhere to precedence constraints, so if a job is outsourced then so are all successor tasks, Equation (14).

$$|\sigma_i| * z_i \leq \sum_{i \in \sigma_i} z'_i \quad \forall i \in N^{\sigma} \tag{14}$$

Equations (15-18) show that variables; $x_{t,k,m}$, $y_{i,k,m}$, $u_{i,i'}$ and z_i are binary.

$$x_{t,k,m} = [0,1] \quad \forall k \in K, m \in M, t \in T \tag{15}$$

$$y_{i,k,m} = [0,1] \quad \forall k \in K, m \in M, i \in N \tag{16}$$

$$u_{i,i'} = [0,1] \quad \forall i, i' \in N, i \neq i' \tag{17}$$

$$z_i = [0,1] \quad \forall i \in N \tag{18}$$

Lastly, Equations (19 and 20) show that the start and end times of jobs are non-negative.

$$e_p \geq 0 \quad \forall i \in N_p \tag{19}$$

$$b_i \geq 0 \quad \forall i \in N \tag{20}$$

3 Large Scale Technician and Task Scheduling Problem Instances

Table 1 shows the large scale technician and task scheduling problems that have been designed for this research. To our knowledge, there is only one piece of research that generated technician and task scheduling problem datasets independently, this was [12]. Two other works we are aware of extended existing vehicle routing datasets, [11] and [13], by concatenating skill requirements from other scheduling problems or generating them randomly.

The datasets created in this research are novel, they involve solving a multi-period scheduling problem, with an outsourcing budget, respecting unavailability of resources and teaming. Column one shows the name of each dataset created, column two (Jobs) shows the number of jobs to be scheduled, column three (Techs) displays the number of available technicians and column four (Budget) displays the outsourcing budget available. Lastly, columns five and six (Domains and Levels) show the number of domains and levels.

There are twelve data instances that range from 1000 to 2500 jobs. These new datasets can be split into four groups; L1-L3, L4-L6, L7-L9 and L10-L12. Each group of data contains the same set of jobs to be scheduled, but contains a varying number of available technicians.

Table 1. Large scale technician and task scheduling problem instances

Dataset	Jobs	Techs	Budget	Domains	Levels
L1	1000	25	500	3	3
L2	1000	50	500	3	3
L3	1000	100	500	3	3
L4	1500	25	1000	4	4
L5	1500	50	1000	4	4
L6	1500	100	1000	4	4
L7	2000	25	1500	3	3
L8	2000	50	1500	3	3
L9	2000	100	1500	3	3
L10	2500	25	2000	4	4
L11	2500	50	2000	4	4
L12	2500	100	2000	4	4

3.1 Generating Large Scale Instances

Each data instance was made up of three files, an instance file, a technician file, and a job file. Firstly, the instance file was generated which contains the number of jobs, technicians, domains, levels and outsourcing budget. Next the set of technicians can be created. Each technician is randomly given a level of expertise in each of the domains and assigned days off within the scheduling horizon. Lastly, the job file is created. Each job is randomly assigned a duration, an outsourcing cost and a priority level. However, the jobs also have two other important attributes, domain skill requirements and precedence and successor relationships.

Generating Job Durations In the ROADEF 2007 challenge problem the length of a working day is limited to 120 time units. In the original problem instances, the job durations ranged from 15 time units to 120 time units, in 15 time unit intervals. Therefore, the job durations in the new datasets have been randomly assigned to be of a length that is a multiple of 15 time units and not greater than 120 time units.

Generating Skill Domain Requirements The total number of technicians skilled in each domain skill level is recorded. When it comes to generating the skill domain requirements of a job, for the first level in each domain a random number is selected from 0 to the maximum number of technicians who possess this area of expertise. For each subsequent level of expertise in a domain a random number is selected between 0 and the previous required level of expertise. This is to ensure that skill levels are hierarchical i.e if four technicians are required to be skilled in domain 2 to level 3, then the next level, i.e domain 2 level 4 must require four or fewer technicians.

Generating Precedence and Successor Relationships Generating precedence and successor relationships between jobs was a complex task. In the ROADEF 2007 challenge problem, there were multi layered precedence and successor relationships that contained many layers and many jobs. In order to generate these types of constraints an algorithm had to be designed.

This algorithm randomly selected a set of jobs, and ordered them in terms of their priority levels in descending order of importance. This is to ensure that jobs of priority group p are dependent on jobs that are priority p or higher, as this is the way the objective function is calculated, a weighted sum of priority end times.

The algorithm then iterates through the list of priority ordered jobs selecting a random number of jobs for each layer of the relationship tree. Next, each job is assigned to its layer. The algorithm then iterates through each job in the tree ensuring each job has at least one connection to another layer; either upwards (successor) or downwards (precedence).

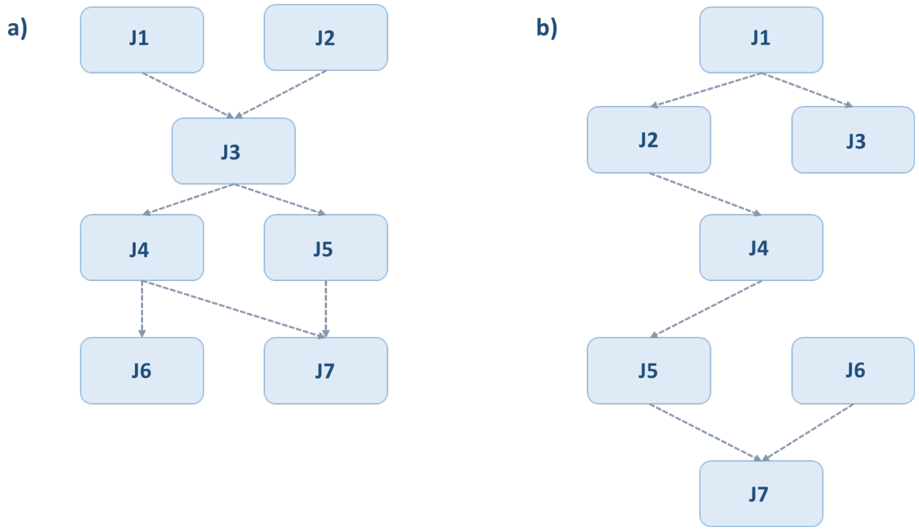

Fig. 1. Example of the dynamic precedence and successor relationship trees

As this algorithm has a random nature the following relationship trees, as shown in Figure 1, have been created using the same set of jobs. In Figure 1a the relationship tree has four layers. On the first layer are jobs 1 and 2, on the second, job 3 (which is a successor of jobs 1 and 2), the third layer contains jobs 4 and 5 (which are successors of job 3), and lastly, on the fourth layer, jobs 6 and 7 (both dependent on job 4, and one dependent on job 5).

In Figure 1b, the relationship tree has 5 layers, with one initial job node and one end job node. This figure depicts that jobs can be a member of the relationship tree without having to have both a successor and predecessor (jobs

3 and 6), reiterating the complexity of job relationships within the ROADEF 2007 challenge problem framework.

4 Heuristic Approaches

4.1 Intelligent Decision Heuristic

The intelligent decision heuristic considers multiple scenarios before making a job allocation decision [10]. Given a set of jobs, the heuristic checks to see if a dummy team could be made for each job. Each dummy team is then checked to see which further job allocations could be made. Each scenario is then scored for skill utilization of the team and the utilisation of available time. The highest scoring scenario is selected and the job allocations are made.

4.2 Greedy Heuristic

In order to benchmark the intelligent decision heuristic we will also implement a greedy heuristic on the large scale technician and task scheduling problem instances. The greedy heuristic does not include the intelligent step that checks the implications of an allocation decision. The greedy heuristic selects a single job randomly, creates a team and then makes further allocation decisions based on the skill waste of the team.

5 Experimental Results

Under the competition rules of the ROADEF 2007 challenge, each run of the heuristic is allowed a 20 minute computational time limit and so in this work, we have used a 20 minute run time. The heuristics were programmed in Java and tested on an HP Z210 Workstation, with an i7-2600 CPU with 3.4 GHZ with 12GB of RAM. Table 2 presents the best result obtained over five runs for each heuristic. In column two the results of the intelligent decision heuristic are displayed, in column three the best result obtained for the greedy heuristic is shown and in column four (% Gap) the percentage gap between the results of the intelligent decision and greedy heuristic are displayed.

6 Discussion

The previous section shows the results obtained for each heuristic approach on the large scale technician and task scheduling problem instances. It is shown that overall the ID heuristic outperforms the greedy heuristic in all problem instances.

Also as the number of technicians available increases, the gap in solution quality generally decreases. This can be expected, because as the number of available technicians on each day increases, there are fewer shortages of skills, and therefore less consideration can be made for team configurations.

Table 2. Experimental results for the large scale problem instances

Dataset	ID	Greedy	% Gap
L1	192810	203850	5.7
L2	97725	103440	5.8
L3	48330	50700	4.9
L4	296940	315210	6.2
L5	147480	156960	6.4
L6	76110	80880	6.3
L7	420660	445335	5.8
L8	198900	207405	4.3
L9	97080	102870	6
L10	574465	607890	5.8
L11	280260	290745	3.7
L12	140970	144840	2.7
Average	214311	225844	5.3

Overall the intelligent decision heuristic finds a solution that is on average 5% better than the quality of solution found by the greedy heuristic. A saving of 5% in personnel costs has the potential to save significant amounts of money in large businesses.

This paper has demonstrated that although the ID heuristic is far more computationally expensive than the greedy heuristic, it can produce better quality results in the same computational time limit. The research has shown that the ID heuristic is both a robust and scalable approach to solving technician and task scheduling problems within strict computational time limits.

7 Conclusion

This paper has shown the benefits of finding efficient ways to solve large scale technician and task scheduling problems in time constrained conditions. In these large scale problems, finding a better quality solution of even 1% can result in large financial savings.

Our contributions to the field are; (i) a methodology for creating technician and task scheduling problem instances, (ii) twelve new large scale technician and task scheduling problems available at *https://akhalfay.wordpress.com/large-scale-ttsps/*, and (iii) a comparative analysis of the greedy heuristic and the intelligent decision heuristic. It is hoped that other researchers will also evaluate their algorithms on these datasets.

Future work will explore related large scale technician and task scheduling problems, that contain the complexity of routing or time windows in which a customer must be visited.

Acknowledgements This research is sponsored by ServicePower Technologies PLC, a worldwide leader in providing innovative mobile workforce management solutions, in cooperation with Manchester Metropolitan University and KTP.

References

1. Burke, E., De Causmaecker, P., Berghe, G.V.: A hybrid tabu search algorithm for the nurse rostering problem. In: Asia-Pacific Conference on Simulated Evolution and Learning. pp. 187–194. Springer (1998)
2. Cordeau, J.F., Laporte, G., Pasin, F., Ropke, S.: Scheduling technicians and tasks in a telecommunications company. Journal of Scheduling 13(4), 393–409 (2010)
3. Crispin, A., Syrichas, A.: Quantum annealing algorithm for vehicle scheduling. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics. pp. 3523–3528. IEEE (2013)
4. Dutot, P.F., Laugier, A., Bustos, A.M.: Technicians and interventions scheduling for telecommunications. France Telecom R&D (2006)
5. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European journal of operational research 153(1), 3–27 (2004)
6. Estellon, B., Gardi, F., Nouioua, K.: High-performance local search for task scheduling with human resource allocation. In: Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics, pp. 1–15. Springer (2009)
7. Hashimoto, H., Boussier, S., Vasquez, M., Wilbaut, C.: A grasp-based approach for technicians and interventions scheduling for telecommunications. Annals of Operations Research 183(1), 143–161 (2011)
8. Haugen, D.L., Hill, A.V.: Scheduling to improve field service quality*. Decision Sciences 30(3), 783–804 (1999)
9. Hurkens, C.A.: Incorporating the strength of mip modeling in schedule construction. RAIRO-Operations Research 43(04), 409–420 (2009)
10. Khalfay, A., Crispin, A., Crockett, K.: Solving technician and task scheduling problems with an intelligent decision heuristic. In: Intelligent Decision Technologies 2016, pp. 63–75. Springer (2016)
11. Kovacs, A.A., Parragh, S.N., Doerner, K.F., Hartl, R.F.: Adaptive large neighborhood search for service technician routing and scheduling problems. Journal of scheduling 15(5), 579–600 (2012)
12. Krishnamoorthy, M., Ernst, A.T., Baatar, D.: Algorithms for large scale shift minimisation personnel task scheduling problems. European Journal of Operational Research 219(1), 34–48 (2012)
13. Pillac, V., Guéret, C., Medaglia, A.: On the dynamic technician routing and scheduling problem. In: Proceedings of the 5th International Workshop on Freight Transportation and Logistics (ODYSSEUS) (2012)
14. Pillac, V., Gueret, C., Medaglia, A.L.: A parallel matheuristic for the technician routing and scheduling problem. Optimization Letters 7(7), 1525–1535 (2013)
15. Society, F.O.R.: What is the roadef 2007 challenge (2016), http://challenge.roadef.org/2007/en/
16. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research 35(2), 254–265 (1987)
17. Titiloye, O., Crispin, A.: Quantum annealing of the graph coloring problem. Discrete Optimization 8(2), 376–384 (2011)