Efficient Heuristics for Solving Precedence Constrained Scheduling Problems

Dr Amy Khalfay[1], Dr Alan Crispin[2], and Dr Keeley Crockett[2]

[1] IBM Services, Global Business Services amy.khalfay@ibm.com
[2] Manchester Metropolitan University
a.crispin@mmu.ac.uk, k.crockett@mmu.ac.uk

Abstract. This paper discusses the occurrence of dependency relationships within NP hard personnel scheduling problems. These dependencies, commonly referred to as precedence constraints, arise in a number of industries including but not limited to: maintenance scheduling, home health care, and unmanned aerial vehicle scheduling. Precedence relationships, as demonstrated in this research, can significantly impact the quality of solution that can be obtained. In such a competitive market it is imperative that new and innovative ways of finding high quality solutions in short computational times are discovered. This paper presents novel datasets, containing 100-1000 jobs to allocate, that are used to benchmark two heuristic algorithms; an intelligent decision heuristic and a greedy heuristic. Each heuristic is coupled with a multi start metaheuristic to provide a set of benchmark results.

Keywords: Precedence constraints · NP hard · Heuristic algorithm · Multi start metaheuristic

1 Introduction

Regardless of the industry, the underlying complexity of interrelationships between jobs or indeed subsets of jobs has and will continue to occur in many vital activities. Precedence constraints can be used to model many different features of scheduling problems for example in the housing development trade, plastering before painting, in service maintenance, the collection of a tool before a job can commence and, finally, in home health care, the minimum time between doses of medication.

Many organisations face the complex problem of scheduling staff to complete a set of jobs in the least costly way Castillo-Salazar et al., (2012). The presence of precedence relationships between jobs can significantly impact the quality of solution that can be obtained, and so, finding intelligent ways of handling these dependencies is of high importance.

Industries such as maintenance and repair include precedence relationships, for example, the need to collect a tool or spare part before going to complete a job Pillac et al., (2013) Additionally, in the housing development trade, there must be coordination of many different types of skilled workers, for example, plumbers, electricians, plasterers, roofers and scaffolders, and certain tasks must be completed before others may commence. There is a similar problem in the home health care industry, which is growing due to the number of private health care services and an ageing population. Many people prefer to be cared for in their own home and rely on third parties to administer care, such as medications, that have strict guidelines Hiermann et al., (2015). Indeed, even a maintenance task that requires a part or tool to be delivered before the fault can be fixed can be modelled as a precedence constraint.

In this paper, a set of personnel scheduling problems has been created that includes varying levels of precedence constraints. The problems created have used the problem definition of the ROADEF 2007 challenge Society,, (2016) where skilled workers must service jobs that require certain skill levels. These problems include the complexity of teaming, priority levels and an outsourcing budget. There are 25 datasets that range from scheduling 100-1000 jobs, with levels of precedence relationships ranging from 0-100%.

This paper is structured as follows, a literature review is undertaken which describes current research in the field of personnel scheduling problems and the motivation for this work, next, the mathematical formulation of the problem is presented. The datasets created are then discussed before giving an overview of the heuristic approaches used. A description of the metaheuristic is then presented, followed by the computational results obtained. Lastly, the results obtained are discussed, and the conclusion identifies the contributions and areas for further work.

2 Literature Review

The field of personnel scheduling has received much research attention and many reviews have been undertaken, such as work by Brucker et al., (2011), Van den Bergh et al., (2013) and Paraskevopoulos et al., (2016. Reviews into the field of technician and task scheduling problems have also been performed by Castillo-Salazar et al., (2012) and Khalfay et al., (2017).

Many problems in the literature have included constraints such as routing and location, multi day problems, time windows, priority levels, tools and spare parts, and technician unavailability. Some of the problems that have included routing and travel time have actually been adapted from vehicle routing problem instances. The vehicle routing problem has been widely studied over many decades and is a generalization of the technician and task scheduling problem. Research by Kovacs et al.,(2012) and Pillac et al.,(2013) adapted vehicle routing problem instances, initially proposed by Solomon, (1987), to solve technician and task scheduling problem instances. The problems studied included scheduling up to 100 jobs, over a single scheduling day, where technicians departed from and returned to a centralised depot.

As many problems studied have been adapted from vehicle routing problems, it is common for the problem to only require the scheduling of technicians over a single scheduling day. However, some problems have required the scheduling of technicians over multiple days such as Mathlouthi et al.,(2016), Zamorano and Stolletz,(2016) and Society, (2016). When considering multiple scheduling days, the constraint of unavailability of resources can occur and was studied in Society,(2016. The use of time windows has also been studied extensively, Tricoire et al., (2013), Kovacs et al., (2012), Zamorano and Stolletz, (2016) and Mathlouthi et al., (2016). In many sectors, there are tasks to be performed which cannot be completed by a single technician, so a team of technicians is needed to satisfy the job's skill requirements. The complexity of teaming has been considered in the ROADEF 2007 challenge problem, Fırat and Hurkens, (2012) and by Kovacs et al., (2012).

This paper is focused on the occurrence of precedence constraints. One of the most notable problems that include precedence constraints is the ROADEF 2007 challenge. This problem was based on real world data instances from France Telecom. This challenge included solving 30 datasets, that require scheduling between 5 and 800 jobs, to teams over multiple scheduling days Dutot et al., (2006). Although this work included precedence constraints, there was no comparison made against the complexity that is added through the occurrence of precedence constraints to finding a high-quality solution. More recent work by Park et al., (2016) required scheduling unmanned aerial vehicles to perform tasks within the manufacturing sector. This problem included precedence constraints between tasks that must be completed. This work also included constraints such as travel time, recharge time of the unmanned aerial vehicles and time windows in which the task must be completed. In addition, some large scale problems have also been created that included precedence constraints and required scheduling up to 2500 jobs, over multiple days to a set of teams, with constraints such as technician unavailability, outsourcing budgets, and priority levels Khalfay et al., (2017).

As shown there are limited problems in the literature that have included the complexity of precedence relationships, even though it is a common complication in many sectors. The researchers are not aware of any work that specifically focuses on the impact that precedence relationships have on the quality of solution that can be obtained. In this paper, 25 personnel scheduling problems, split into 5 sets of data that require scheduling between 100 and 1000 jobs are presented. Within each set of data, the percentage of precedence relationships between the jobs varies from 0% to 100%, in order to measure the effects of precedence relationships. The problem definition of the ROADEF 2007 challenge has been used to generate the datasets since this problem comes from the business world and includes a multitude of complexities that are present in many personnel scheduling problems.

3 Problem Formulation

The problem requires constructing a set of teams, over a set of days, $K = [1...k]$, to complete a set of jobs. Each job $i \in N$ has set of skill requirements $s^i_{\delta\alpha}$ (where δ is the domain and α is the skill level), a priority level $p \in [1...4]$, a duration d_i, an outsourcing cost c_i and a set of successor jobs σ_i.

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 $p = [1, 2, 3, 4]$, as shown in Equation (1).

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

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 $x_{t,k,r} = 1$ if technician t belongs to team r 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.

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

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.

$$\sum_{r=1}^{m} x_{t,k,r} = 0 \quad \forall k \in K, t \notin T_k \tag{5}$$

Let $y_{i,k,r} = 1$ if job i is assigned to team r 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.

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

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

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

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.

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

$$120(k) * \sum_{r=1}^{m} y_{i,k,r} \geq b_i + d_i \quad \forall i \in N, k \in K \tag{10}$$

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'.

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

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.

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

In the Set B and X 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; $x_{t,k,r}$, $y_{i,k,r}$, $u_{i,i'}$ and z_i are binary.

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

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

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

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

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

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

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

4 Generating Precedence Constrained Technician and Task Scheduling Problem Instances

This section describes the data generator that has been developed to produce datasets under the framework of the ROADEF 2007 challenge problem that has the desired characteristics.

4.1 Creating Varying Precedence Levels

In order to create datasets which contain varying levels of precedence constraints within the same set of jobs (in terms of skill requirements, priority levels and outsourcing costs) a methodology for dataset creation had to be designed. The methodology had to ensure that every dataset, containing the same number of jobs, of precedence level p has all the precedence relationships present in the datasets of precedence levels lower than p. For example, dataset P2 has 100 jobs to allocate and contains 25% precedence relationships, therefore dataset P3, contains all the precedence relationships contained in dataset P2, plus another 25% of precedence relationships.

4.2 Precedence constrained technician and task scheduling problem generator

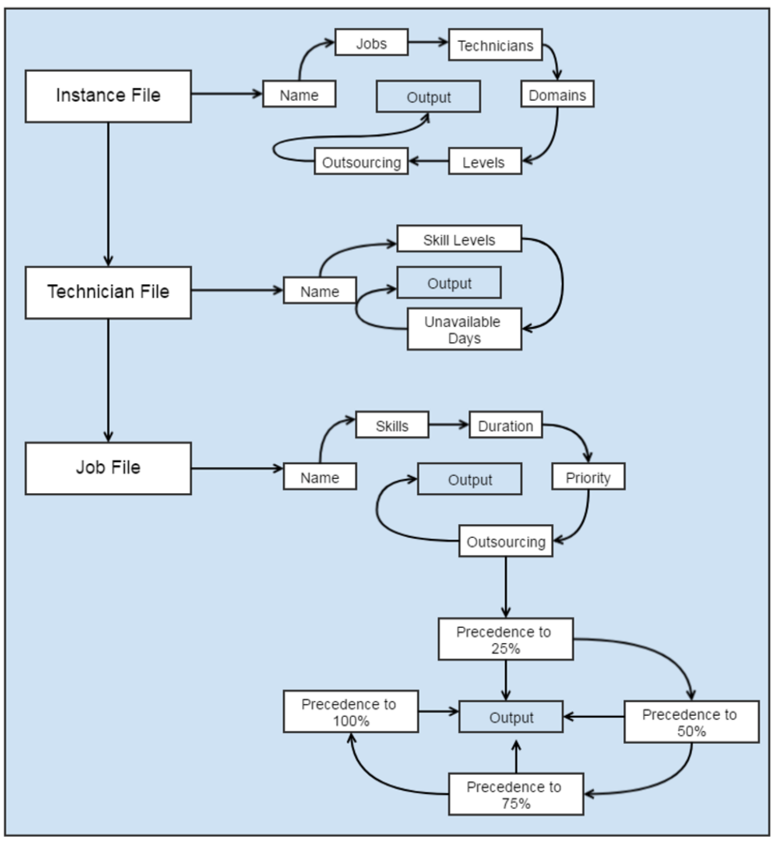

Fig. 1: Precedence constrained data generator

Figure 1 illustrates how the precedence constrained datasets were generated under the framework of the ROADEF 2007 challenge problem. Firstly, the instance file is generated containing information about the problem (dataset name, number of jobs, number of technicians, number of domains, number of levels within each domain, and, the outsourcing budget available). Next, the set of technicians are created and are each assigned skills and days off. The jobs are then created and assigned skill requirements, durations, priority levels and outsourcing costs.

Skill requirements are given to each job, making sure each job requires no more than 5 technicians, as in the ROADEF 2007 challenge. Jobs are then assigned durations between 15 and 120 time units (the length of a working day), a priority level from 1 to 4, and lastly an outsourcing cost.

The precedence algorithm then begins which creates multi layered relationships between jobs. An output file is written before any precedence relationships are added, thus created a dataset with no relationships present. Precedence levels to 25% are created and an output file is written, then the precedence relationships are built to 50% and again an output file is written. This continues until the precedence levels have reached 100%.

5 Precedence constrained technician and task scheduling problems

Using the problem formulation of the ROADEF 2007 challenge a set of technician and task scheduling problems has been created. Table 1 shows the datasets that have been created. The data can be separated into 5 groups, P1-P5, P6-P10, P11-P15, P16-P20 and P21-25 based on the number of jobs to schedule.

Within each group, the percentage of precedence constraints vary, 0%, 25%, 50%, 75% and 100%. Within each group of data, the remaining characteristics of the datasets are the same, i.e. number of technicians, domains and levels and outsourcing budgets, in order to evaluate the effect of precedence constraints on the quality of solution that can be achieved.

6 Heuristic approaches

In order to provide a performance comparison on the precedence constrained technician and task scheduling problem instances, two heuristic procedures outlined in the following subsections have have implemented .

6.1 Intelligent Decision heuristic

The intelligent decision heuristic initially proposed in Khalfay et al., (2016), has been applied to the ROADEF 2007 challenge problem and large scale technician and task scheduling problems in Khalfay et al., (2017). This heuristic is a novel approach due to its ability to consider multiple scenarios before making a job allocation decision, thereby making more intelligent decisions.

Whilst there are jobs to allocate, a schedule is created and all available technicians are initialised into single technician teams. Whilst this schedule is not full, the algorithm

Table 1: Technician and task scheduling problem datasets

Dataset	Jobs	Precedence	Technicians	Domains	Levels	Budget
P1	100	0%	15	3	2	100
P2	100	25%	15	3	2	100
P3	100	50%	15	3	2	100
P4	100	75%	15	3	2	100
P5	100	100%	15	3	2	100
P6	200	0%	25	2	3	200
P7	200	25%	25	2	3	200
P8	200	50%	25	2	3	200
P9	200	75%	25	2	3	200
P10	200	100%	25	2	3	200
P11	400	0%	50	3	3	400
P12	400	25%	50	3	3	400
P13	400	50%	50	3	3	400
P14	400	75%	50	3	3	400
P15	400	100%	50	3	3	400
P16	800	0%	80	4	2	800
P17	800	25%	80	4	2	800
P18	800	50%	80	4	2	800
P19	800	75%	80	4	2	800
P20	800	100%	80	4	2	800
P21	1000	0%	100	3	4	1000
P22	1000	25%	100	3	4	1000
P23	1000	50%	100	3	4	1000
P24	1000	75%	100	3	4	1000
P25	1000	100%	100	3	4	1000

iterates through each priority class. Given a set of jobs, of priority p, the algorithm creates a dummy team for each job (the optimal team configuration for the job). Next, each dummy team is then utilized, by checking which further job allocations could be made, if this team existed. Each possible team configuration is then scored based on the utilisation of the team's skill set and time. The highest scoring scenario is selected and the team is made and job allocations are assigned.

The algorithm continues until no more jobs within the current priority class can be allocated. Once each priority class has been iterated through, another schedule is created and the process continues. Once all jobs have been allocated, an initial solution has been created.

6.2 Greedy heuristic

In order to benchmark the intelligent decision heuristic, a greedy heuristic has been implemented on the technician and task scheduling problem instances. The greedy heuristic does not include the intelligent step that checks the implications of an allocation decision.

Whilst there are jobs to allocate, a schedule is created and all available technicians are initialized into single technician teams. The algorithm iterates through each priority class from 1-4 until no more jobs can be allocated. Within each cycle through the priority class, a single job is selected at random for allocation. A team is then created by adding the technician that covers the most skill requirements of the job until all requirements have been met. The job is then allocated to the team. Next, if the newly constructed team has available time, further job allocations are made considering the skill cover of the team to the set of possible allocations.

The algorithm continues until no more job allocations can be made, and so a new schedule is created. Once all jobs have been allocated an initial solution has been found.

7 Multi start metaheuristic

A multi start metaheuristic to guide the lower level heuristics has been implemented. Multi start techniques have proved popular strategies in many combinatorial optimisation problems Blum and Roli, (2003).

Variables: S: current solution, S': neighbouring solution, S_{Best}: the best solution, O: the set of local operators

Initialise S_{Best}
while total time not met **do**
    $S \leftarrow$ Generate initial solution using heuristic
    while improvement time not met **do**
        $o \leftarrow O$
        $S' \leftarrow S(o)$
        if $S' \leq S$ **then**
            $S \leftarrow S'$
            if $S \leq S_{Best}$ **then**
                $S_{Best} \leftarrow S$
            end if
        end if
    end while
end while
return S_{Best}

Fig. 2: Multi start metaheuristic

The multi start metaheuristic presented in this paper repeatedly takes an initial solution generated by the heuristic procedure and then uses a pre-defined amount of computational time or number of iterations to improve it. The best solution found is updated as better quality results are discovered. This process continues, until the total amount of computational time or number of iterations has been reached. The best solution found throughout the search is then output.

Figure 2 shows the implementation of the multi start metaheuristic. On line 1, the best solution is initialized, which will keep track on the best quality solution found over the whole run. Whilst the algorithm's total run time has not been met, an initial solution is generated on line 3 using the heuristic procedure (either intelligent decision or greedy). Next, whilst there is improvement time remaining, a local operator is selected and then applied to the current solution, which generates the neighbouring solution S' on line 6. If solution S' is of better quality than S then it replaces S on line 8. If the new solution S is of better quality than the best solution found then S_{Best} is updated. The algorithm continues to iterate generating initial solutions and then trying to improve them until the total run time has been used. Once the total run time has been used, the best solution found over the whole run is output.

7.1 Local operators

A variety of operators have been used in this research to perturb a solution. These operators have been featured in previous work Khalfay et al., (2016) and Khalfay et al., (2017).

- **Move a job-** this operator selects a single job, that belongs to a team on a day within the scheduling horizon. This job is then removed from its current position, and reallocated to a different team within the scheduling horizon.
- **Swap two jobs-** this operator randomly selects two teams and removes a job belonging to each team. The jobs are then reallocated to the opposite team if skill and time constraints allow.
- **Shuffle-** this operator randomly selects a team and reorders the order of the jobs that have been allocated to the team.
- **Decompose and rebuild-** this operator selects a day within the scheduling horizon, removes all jobs that have been allocated to this day and removes all team configurations. The set of jobs is then reallocated using the construction heuristic allowing for new team configurations to form.
- **Decompose and rebuild N schedules-** this operator selects multiple consecutive days within the scheduling horizon, removing all jobs and team configurations. The construction heuristic is then used to reallocate the set of jobs allowing for new team configurations to form.
- **Remove N jobs-** this operator selects a number of jobs to remove from the scheduling horizon. The set of removed jobs is then reallocated to the scheduling horizon.

8 Experimental Results

The datasets have been tested using the following framework. For each run a 10 minutes computational time has been allowed with each phase of improvement within the multi start procedure being limited to 30 seconds. Each heuristic was run 5 times and the best solution found recorded. The algorithms were written in Java and programmed on a Tower Workstation, i7, 16 GiB.

Table 2: Computational results for the technician and task scheduling problem datasets

Name	Intelligent Decision	Greedy
P1	31500	33960
P2	32340	36240
P3	35040	38220
P4	35580	39150
P5	36240	40440
P6	52200	54480
P7	53730	54630
P8	57510	59400
P9	62520	63750
P10	65220	66330
P11	46290	49680
P12	46290	49770
P13	46410	49770
P14	46670	49800
P15	48600	54060
P16	53310	63840
] P17	54270	63960
P18	55320	64950
P19	56220	68235
P20	58530	69180
P21	48840	51720
P22	49830	51720
P23	51180	56880
P24	51210	56850
P25	51450	57480

Table 2 shows the results obtained using the two heuristic procedures, the intelligent decision heuristic and the greedy heuristic.

8.1 Equality of Means across Precedence levels

This research has been conducted based on the belief that precedence constraints impact the quality of solution that can be obtained. If two datasets are taken, for example P1 and P2, which contain the same set of 100 jobs, but P1 has no precedence constraints whereas P2 has 25% precedence constraints then it would be anticipated that the mean expected value on each of these datasets will not be equal.

To test this a hypothesis test has been carried out. The null hypothesis, h_0 claims there is no difference in the expected objective value between the groups. Conversely, h_1 specifies that there is a difference in the mean value obtained.

$$h_0 : x_1 - x_2 = 0 \tag{21}$$

$$h_1 : x_1 - x_2 \neq 0 \tag{22}$$

A series of experiments was conducted where the heuristic was ran 20 times on each dataset, P1 and P2, with the objective values recorded. The mean value obtained for each was calculated along with the standard deviation. Next, the standard error between both groups was calculated along with the degrees of freedom. A T statistic was calculated, 14.47 which is compared to a t value, in a two tailed test on 37/38 degrees of freedom. A T value is calculated and compared to the T statistic. In this case, with a significance level of $\alpha = 0.01$, on a two tail test, the T Value is 2.712 which is less than 14.747. So there are significant grounds to reject h_0 and accept the alternative hypothesis, that there are differences in the means.

9 Discussion

The results presented show that overall the intelligent decision heuristic has outperformed the greedy heuristic. In each dataset, P1-P25, the intelligent decision heuristic found a solution of better quality than the greedy heuristic. The performance improvement found by the intelligent decision heuristic can be attributed down to its ability to consider multiple scenarios and the consequences of each decision it takes.

The T-tests have shown that increased levels of precedence constraints do impact the quality of solutions that can be obtained.

10 Conclusion

This paper has demonstrated that precedence relationships are an important consideration for many personnel scheduling problems that arise in multiple sectors such as; service maintenance, housing developments and home health care. The ability to effectively deal with precedence constraints and find a low cost schedule has significant benefits and allows organisations to minimise staffing costs. It is hoped that this research provides a set of benchmark problems that future researchers can test their algorithms on and provide a comparative performance analysis.

The contributions of this research are; (i) 25 personnel scheduling datasets that include varying levels of precedence constraints, (ii) a set of benchmark results for the datasets, and (iii) a comparison of the effects of precedence relationships across two heuristic procedures, the ID heuristic and the greedy heuristic.

Future research will investigate other important complexities and constraints that arise in personnel scheduling problems such as location and travel time.

Bibliography

[1] J Arturo Castillo-Salazar, Dario Landa-Silva, and Rong Qu. A survey on workforce scheduling and routing problems. In *Proceedings of the 9th international conference on the practice and theory of automated timetabling*, pages 283–302. Citeseer, 2012.

[2] Victor Pillac, Christelle Gueret, and Andrés L Medaglia. A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, 7(7): 1525–1535, 2013.

[3] Gerhard Hiermann, Matthias Prandtstetter, Andrea Rendl, Jakob Puchinger, and Günther R Raidl. Metaheuristics for solving a multimodal home-healthcare scheduling problem. *Central European Journal of Operations Research*, 23(1): 89–113, 2015.

[4] French Operational Research Society. What is the roadef 2007 challenge, 2016. URL http://challenge.roadef.org/2007/en/.

[5] Peter Brucker, Rong Qu, and Edmund Burke. Personnel scheduling: Models and complexity. *European Journal of Operational Research*, 210(3):467–473, 2011.

[6] Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

[7] Dimitris C Paraskevopoulos, Gilbert Laporte, Panagiotis P Repoussis, and Christos D Tarantilis. Resource constrained routing and scheduling: Review and research prospects. 2016.

[8] Amy Khalfay, Alan Crispin, and Keeley Crockett. A review of technician and task scheduling problems, datasets and solution approaches. In *Intelligent Systems Conference (IntelliSys) 2017*, pages x–xx. IEEE, 2017.

[9] Attila A Kovacs, Sophie N Parragh, Karl F Doerner, and Richard F Hartl. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of scheduling*, 15(5):579–600, 2012.

[10] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[11] Ines Mathlouthi, Michel Gendreau, and Jean-Yves Potvin. Mixed integer programming for a multi-attribute technician routing and scheduling problem. 2016.

[12] Emilio Zamorano and Raik Stolletz. Branch-and-price approaches for the multi-period technician routing and scheduling problem. *European Journal of Operational Research*, 2016.

[13] Fabien Tricoire, Nathalie Bostel, Pierre Dejax, and Pierre Guez. Exact and hybrid methods for the multiperiod field service routing problem. *Central European Journal of Operations Research*, 21(2):359–377, 2013.

[14] Murat Fırat and CAJ Hurkens. An improved mip-based approach for a multi-skill workforce scheduling problem. *Journal of Scheduling*, 15(3):363–380, 2012.

[15] Pierre-François Dutot, Alexandre Laugier, and Anne-Marie Bustos. Technicians and interventions scheduling for telecommunications. *France Telecom R&D*, 2006.

[16] Youngsoo Park, Yohanes Khosiawan, Ilkyeong Moon, Mukund Nilakantan Janardhanan, and Izabela Nielsen. Scheduling system for multiple unmanned aerial vehicles in indoor environments using the csp approach. In *Intelligent Decision Technologies 2016*, pages 77–87. Springer, 2016.

[17] Amy Khalfay, Alan Crispin, and Keeley Crockett. Solving large scale technician and task scheduling problems with an intelligent decision heuristic. In *Intelligent Decision Technologies 2017*, pages x–x. Springer, 2017.

[18] Amy Khalfay, Alan Crispin, and Keeley Crockett. Solving technician and task scheduling problems with an intelligent decision heuristic. In *Intelligent Decision Technologies 2016*, pages 63–75. Springer, 2016.

[19] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3): 268–308, 2003.