IIMB-WP N0. 682/2023

WORKING PAPER NO: 682

A Study on the Instance Space of the Shift Minimization Personnel Task Scheduling Problem

Reshma Chandrasekharan
Decision Sciences Area
Indian Institute of Management Bangalore
Bannerghatta Road, Bangalore – 560 076
reshmac@iimb.ac.in

Cimmo Nurmi
Satakunta University of Applied Sciences
Finland
cimmo.nurmi@samk.fi

Nico Kyngas
Satakunta University of Applied Sciences
Finland
nico.kyngas@samk.fi

Year of Publication – April 2023

A study on the instance space of the shift minimization personnel task scheduling problem

Reshma Chandrasekharan Cimmo Nurmi Nico Kyngas

Abstract

Much like any other problem in optimization, the shift minimization task scheduling problem has also greatly benefited from the presence of a benchmark instance set. This paper studies the properties of this benchmark dataset and identifies the factors that most influence the hardness of these instances based on the performance of the state of the art solution techniques available for the problem.

1 Introduction

Traditionally, workforce schedules were prepared manually. Literature suggests that there is significant improvement in efficiency while using computer generated schedules (Bergh, Beliën, Bruecker, Demeulemeester, and Boeck, 2013). In organizations with more than hundreds of employees, manual scheduling becomes practically impossible with the implementation of numerous practical constraints, fairness regulations and employee preferences. With the advancement in operations research, we are now able to quickly generate optimized schedules which allows organizations to better utilize their resources while maintaining a fair working environment for the employed workforce.

From estimating the workload to the generation of daily schedules, workforce scheduling in practice is performed in multiple stages, the assignment of shifts to employees being one of the final stages in the entire process. Recent research in personnel scheduling recommends that optimizing task assignment to employees within shifts can lead to significant improvement in resource utilization and service quality especially while dealing with multi-skilled workforce and tasks that are time-bound and skill-specific. It is often in the interest of employer organizations to generate such rosters, however, while minimizing the number of employees/shifts to execute it. This constitutes the primary objective of the shift minimization personnel task scheduling problem (SMPTSP).

The SMPTSP was introduced by Krishnamoorthy, Ernst, and Baatar (2012) as a variant of the personnel task scheduling problem. The paper characterised the problem and presented a benchmark dataset consisting of 137 instances, mathematical formulations for the problem and proposed two heuristics based on Lagrangean relaxation. Since then, several research articles such as Lin and Ying (2014) and Fages and Lapègue (2014) have addressed the problem until Smet, Wauters, Mihaylov, and Berghe (2014) proposed a heuristic that solved all the 137 benchmark instances to optimality and introduced 10 harder instances. Fages and Lapègue (2014) also introduced an instance set with comprised of instances with novel structural properties. The constructive heuristic proposed by Chandrasekharan, Smet, and Wauters (2021) solved all the proposed instances to optimality. Later on, Kyngäs and Nurmi (2021) presented an efficient ruin and recreate heuristic for the problem and proposed a new set of difficult instances.

It is proven that SMPTSP is an $\mathcal{NP}$-hard problem. However, such a classification only talks about the worst case complexity of the problem. The knapsack problem is $\mathcal{NP}$-hard whereas there exist efficient techniques that can solve most of its instances to optimality. In contrast, the traveling salesman problem, while belonging to the same complexity class, appears empirically harder to solve. Researchers for a long time have been trying to answer whether the experience of some $\mathcal{NP}$-hard problems appearing to be easier to solve is the result of intrinsic properties of problem or is instead due to the lack of representation of harder areas of the instance space in its benchmark dataset. The recent advancement in the area of instance space analysis (Smith-Miles, 2019) however has facilitated such studies. While the existing papers on the SMPTSP have discussed benchmark instances and its properties and built a large benchmark dataset, a systematic study of the instance space is missing in the literature.

The primary contribution of this paper is a detailed study of the instance space of the SMPTSP. Extensive experiments were conducted to identify relevant factors and test the extent of their influence in determining the hardness of problem instances. In addition to statistical experiments, an in-depth study of the structural properties of the problem based on graph theory is also presented. This is of significant practical relevance since algorithm performances are typically compared based on their performance on the benchmark instances. Moreover, many existing solution techniques tune their algorithms based on instance structure. As a result of these experiments a harder and diverse set of instances for the SMPTSP have been generated, which is one of the other major contributions of the paper. The instance analysis presented were performed on the basis of the results of the two best performing algorithms available for the SMPTSP. Performance of these algorithms on the existing benchmark instances and the newly generated instances are also published in this paper.

2 Problem definition

The SMPTSP arises when there is a need to optimally assign a multi-skilled workforce to skill-specific tasks. An SMPTSP instance is characterised by a set of shifts and tasks. In order to execute a certain task, some skills are required. The skill-set of the shift determines the tasks it is capable of executing. The schedule of these tasks are given and the aim of the SMPTSP is to identify the smallest possible subset of shifts which can execute these tasks.

Figure 1 represents a sample SMPTSP instance. Here, tasks and shifts are represented as intervals in time. Tasks are numbered and the numbers written on the shifts denotes the tasks they are qualified to execute. Note here that all shifts are of the same time duration (24 hrs). Such an assumption is made in order to simplify the instance structure while making it more general in terms of its applications to different kind of scheduling problems. One can always consider a shift as as two or more employees/shifts of similar skill structure scheduled one after the other without any loss of generality. Similarly, a shift that is longer than 24 hrs can be split in order to bring it to this form.

Let the set of shifts and tasks associated with an SMPTSP instance be denoted by $W = \{1, 2, ..., m\}$ and $J = \{1, 2, .., n\}$ respectively. $W_j \subset W$ represents the subset of shifts that are qualified to execute task $j \in J$. Shifts are not allowed to multi-task. Therefore, two tasks $t_i, t_j \in J$ which overlap in time cannot be assigned to the same shift. Figure 2 shows how such conflicts can be modelled as an interval graph, $G = (V, E)$. In G, vertices represent tasks and two tasks are connected by an edge if their corresponding time intervals intersect and hence cannot be assigned to the same shift. Clearly, no two tasks which are part of a clique in G can be assigned to the same shift. More precisely, let G_w be the subgraph of G induced by vertices J_w, the set of tasks that shift w is capable of executing. If $C^w = \{K_1, K_2, .., K_t\}$ are the set of maximal cliques in G_w, no two tasks in K_i can be assigned to the same shift.

Given the conflict graph G of the tasks of an SMPTSP instance, the problem of identifying the smallest subset of shifts that can execute all the tasks corresponds to the list coloring problem in interval graphs. This problem is proven $\mathcal{NP}$-hard. While finding all maximal cliques in a general graph is an equally hard problem, the problem is polynomial-time in interval graphs. This enables us to compute C^w efficiently and employ it in developing tight integer programming formulation of the problem. Note here that the size of the largest maximal clique in G is a valid lower bound for the number of shifts required and is called the clique lower bound (CLB). The integer programming formulation for the SMPTSP proposed by Krishnamoorthy

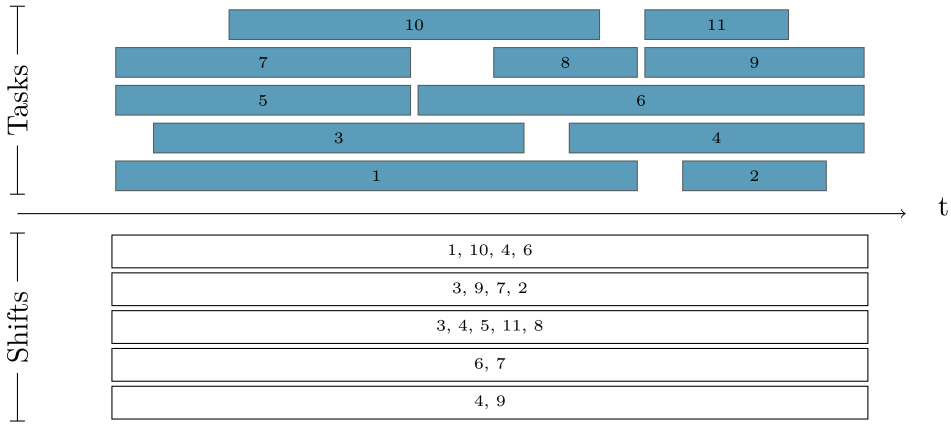

Figure 1: A sample SMPTSP instance. The blue rectangles are tasks while the white rectangles represent shifts.

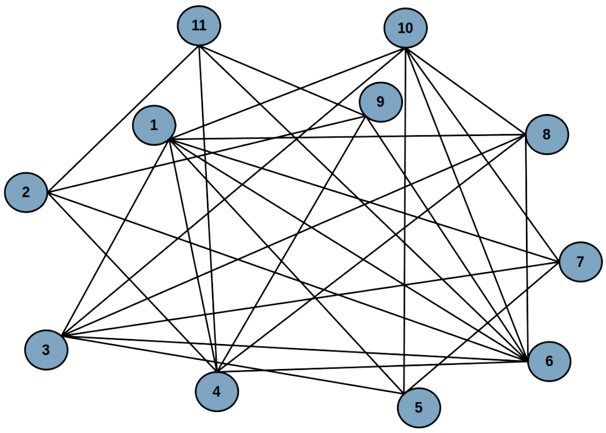

Figure 2: Conflict graph of the tasks involved in the sample SMPTSP instance presented in Figure 1

et al. (2012), SMPTSP-MIP, is presented below.

$$x_{jw} = \begin{cases} 1 & \text{if task } j \in J \text{ is assigned to employee } w \in W \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$y_w = \begin{cases} 1 & \text{if shift } w \in W \text{ is active} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\text{minimize:} \quad \sum_{w \in W} y_w \tag{3}$$

$$\text{subject to:} \quad \sum_{w \in W_j} x_{jw} = 1 \quad \forall j \in J \tag{4}$$

$$\sum_{j \in K_l^w} x_{jw} \leq y_w \quad \forall w \in W, \forall K_l \in C^w \tag{5}$$

$$0 \leq y_w \leq 1 \quad \forall w \in W \tag{6}$$

$$x_{jw} \in \{0, 1\} \quad \forall j \in J, \forall w \in W \tag{7}$$

Constraints (4) ensure that exactly one shift is assigned to a task. Conflicting tasks in maximal cliques of G_w are prevented from being assigned to the same shift by way of Constraints (5). In addition, these constraints ensure that the shift is counted as active if and only if it has been assigned to a task. Constraints (6) and (7) provide the variable bounds.

3 Benchmark instances and characteristics

Currently, there exists four different instance sets for the SMPTSP. The instance set proposed by Krishnamoorthy et al. (2012), denoted as KEB, is composed of 137 instances with up to 420 shifts ad 2105 tasks. These instances are characterised by their tightness and multi-skilling level aside from their size. Tightness (T) of an instance is defined as the percentage of total task lengths with respect to the total shift duration. Multi-skilling level (MSL) of an instance is defined as the average of skill-levels of the shifts involved, where skill-level of a shift is defined as the percentage of tasks they are qualified to execute. Krishnamoorthy et al. (2012) observes that instances with low tightness values are easier to solve.

A series of experiments on the empirical hardness of instances conducted by Smet et al. (2014) shows that instances with MSL around 33% are often harder to solve. In addition, they show that low values of average task duration (TD) is another factor that makes instances harder to solve. As a result of these experiments, they introduced 10 challenging instances, denoted as SWMB. In both these instance sets, the optimal solution equals the CLB. In other words, in these instances, shift skill-structure does not seem to significantly influence the shift utilization. Fages and Lapègue (2014) introduced 100 new instances, denoted as FL, such that the optimal solution of these instances do not coincide with the CLB. Later, Solyali (2016) came up with an efficient lower bounding technique.

A fourth set of benchmark instances, denoted as KN, was introduced by Kyngäs and Nurmi (2021). This paper discusses various instance parameters that may influence the instance hardness and the KN instances are designed such that they are significantly more challenging and structurally different compared to the benchmark data set that existed then. The following subsection explores instance structure in detail.

3.1 Benchmark datasets and their characteristics

Size of the instance, which is a combination of the number of shifts and tasks associated with the instance, alongside various other factors are known to impact the difficulty of a given instance. Table 1 summarizes basic instance properties. While very large sizes are often associated with longer runtimes, what exactly contributes to the hardness of the instance is still unknown. The literature has shown that high tightness, low mutli-skilling level and shorter task lengths have a positive correlation with instance hardness. In this study, we explore properties such as average task length and maximum task length of an instance in addition.

Data set	KEB	FL	SWMB	KN	KNC
size	137	100	10	30	70
—W—	22-420	60-950	44-153	20-500	300
—J—	40-2105	70-1600	258-1577	105-2473	1157-6170

Table 1: Summary of instance properties of benchmark instances

From a graph theoretic perspective, the SMPTSP corresponds to list coloring of interval graphs. When viewed purely as a coloring problem, the length of tasks does not correspond directly to any standard graph characteristics. However, having larger or shorter graphs could in result in large or small graph degrees, which in turn could have a significant impact. In order to test this, the average and maximum task length along with the average degree of the underlying conflict graph G are studied. One must note that two conflicting tasks in G are not in actual conflict unless there are shifts that are skilled to execute both. In order to study this, a new conflict graph G^s on tasks is considered, where two tasks with overlapping time intervals are connected by an edge if and only if there exist at least one shift qualified to execute both the tasks.

Figure 3 presents an overview of instance characteristics based on the parameters discussed. Clearly, the newly generated KN30 and KNC70 instances differ from the existing datasets by their relatively higher values of tightness, average and maximum task lengths and the degree of G and G^s. It is interesting to observe that average degree of the underlying conflict graphs G and G^s do not always follow the trends in the distribution of average task length values. Specifically, the KNC70 instances exhibit higher average degree values of G ans G^s despite having lower average task length values. This could be a result, however, of the presence of extremely long tasks, as one can observe from the large maximum task length values.

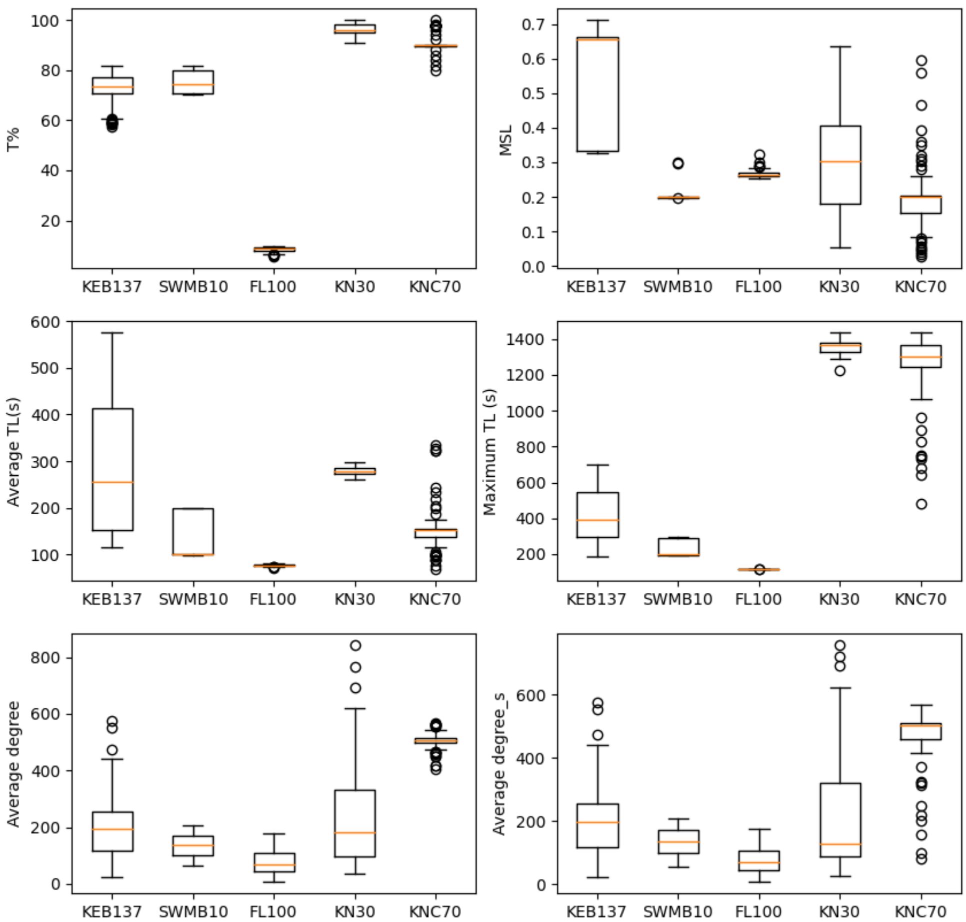

Figure 3: Benchmark datasets and their characteristics

Since many real-world instances are very large in size, we also particularly explore what makes some instances hard for decomposition-based algorithms and heuristics which involves solving parts of the problem locally. Figure 4 represents two instances of contrasting skilling structure. While such high decomposability can usually be associated with very low multi-skilling, it is important to note that even with moderately low values of multi-skilling, such hard-to-decompose structures are possible. What makes this more intriguing is that no single instance characteristic can sufficiently capture the complexity of the underlying graph structure. The presence of very large maximal cliques usually render the instance difficult to decompose, and this

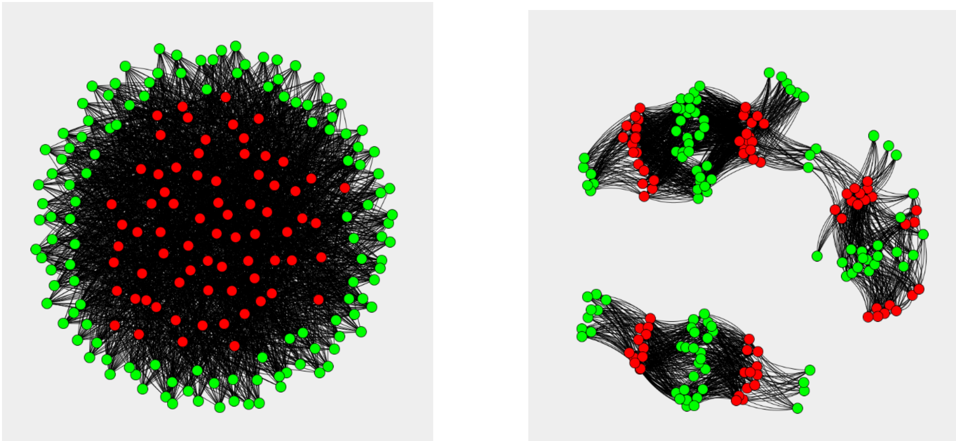

Figure 4: Graphs that represent the skilling structure of two SMPTSP instances. Here, red nodes represent employees and green nodes represent tasks. An employee node is adjacent to a task node if the employee is skilled to perform the task.

information is partially captured by the average degree of G and G^s. However, one must note that while the size of the maximal clique is a valid and good lower bound for the chromatic number of a graph, it is possible to generate triangle-free graphs of very large chromatic number, Mycielsky graphs being an excellent example. Along these lines, we computed the transitivity values of underlying graphs G and G^s for all the instances. However, the transitivity values did not show much variability across the instances.

4 Computational Experiments

Instance space analysis being the major aim of the paper, extensive experimentation and analysis have been carried out in order to test the influence of the instance characteristics on algorithm performance. For this purpose, performance of two state of the art algorithms for the TSP, GFA (Kyngäs and Nurmi, 2021) and CMH (Chandrasekharan et al., 2021) has been utilized. Note here that CMH is a decomposition-based technique. Experiments were performed on 4 threads of 11th Gen Intel(R) Core(TM) i9-11950H @ 2.60GHz machine and 32 GB RAM with a time limit set to 2 hrs.

The performance of CMH and GFA on the new instances KN30 and KNC70 are presented in tables 2 and 3. CMH finds 49 feasible solutions and 36 optimal solutions whereas GFA finds 61 feasible solutions out of which 48 are optimal.

Table 2: Results of CMH and GFA on the KN30 instances. LB denotes lower bound.

Instance	LB	CMH	Time(s)	GFA	Time(s)
kn01	20	20	0.3	20	1
kn02	25	NaN	0.13	25	1
kn03	30	NaN	1.88	30	1
kn04	35	35	2.42	35	2
kn05	40	40	0.85	40	1
kn06	45	45	0.36	45	1
kn07	50	NaN	3.79	50	5
kn08	55	55	4.26	55	1
kn09	60	60	2.78	60	1
kn10	65	65	30.71	65	1
kn11	70	NaN	4.45	70	1
kn12	75	75	3.46	75	1
kn13	80	NaN	5.34	80	2
kn14	85	NaN	56.31	85	37
kn15	100	100	9	100	10
kn16	110	110	124.15	110	14
kn17	120	120	13.1	120	1
kn18	130	130	11.21	130	6
kn19	140	140	8.86	140	1
kn20	149	149	205.32	149	35
kn21	160	160	6,767.65	160	2
kn22	180	180	354.36	180	51
kn23	199	200	2,477.4	199	1,432
kn24	239	239	321.68	239	9
kn25	279	280	7,200	279	13
kn26	309	NaN	7,200	315	7,200
kn27	356	356	197.89	356	24
kn28	399	399	376.75	399	24
kn29	443	447	7,200	445	7,200
kn30	488	496	7,200	492	7,200

4.1 Instance space analysis

In this section, the results of the CMH and GFA runs are plotted against the instance characteristics described in Section 3.1 in an attempt to explore their influence in algorithm performance trends. Figures 5, 6, 7, 8 and 9 plots the trends in the algorithm performances with respect to the instance characteristics. As observed by other papers in the literature, low multi-skilling values seem to be associated with high gaps and algorithm runtimes. In contrast, lower average task length values, which is often considered as a characteristic of difficult instances, do not seem to significantly influence algorithm performances. The most significant observation from these experiments is the correlation of average degree of G and G^s with CMH and GFA algorithm runtimes.

Table 3: Results of CMH and GFA on the KNC70 instances. LB denotes lower bound.

Inst	LB	CMH	T(s)	GFA	T(s)
knc00	283	285	267.83	283	2,772
knc10	285	290	49.29	285	295
knc11	287	288	82.09	287	15
knc12	285	286	100.2	285	94
knc13	287	289	134.18	287	24
knc14	285	285	174.06	285	8
knc15	287	289	524.9	287	89
knc16	284	286	724.46	284	198
knc17	285	286	1,806.62	285	573
knc18	288	288	2,765.56	288	149
knc19	284	NaN	7,200	285	7,200
knc20	285	287	729.75	285	20
knc21	287	289	763.49	287	3
knc22	284	285	483.38	284	42
knc23	285	287	315.91	285	29
knc24	287	289	245.35	287	50
knc25	287	289	136.13	287	38
knc26	285	290	292	285	1,278
knc27	283	291	185.01	289	7,200
knc28	288	298	186.67	294	7,200
knc29	285	NaN	7,200	300	7,200
knc30	262	263	185.33	262	34
knc31	264	266	196.19	264	102
knc32	272	274	234.69	272	13
knc33	277	279	180.31	277	9
knc34	282	282	177.06	282	22
knc35	290	291	253.55	290	715
knc36	296	297	354.73	296	143
knc37	298	299	1,456.34	299	7,200
knc38	300	NaN	7,200	NaN	7,200
knc39	300	NaN	7,200	NaN	7,200
knc40	284	286	276.78	284	887
knc41	286	287	282.51	286	56
knc42	285	286	155.14	285	87
knc43	283	284	295.73	283	60

Inst	LB	CMH	T(s)	GFA	T(s)
knc44	284	285	303.45	284	34
knc45	285	287	332.01	285	235
knc46	282	285	182.01	283	7,200
knc47	285	287	233.87	285	119
knc48	285	286	357.75	285	110
knc49	283	286	278.01	283	928
knc50	287	297	697.09	289	7,200
knc51	285	298	21.19	285	588
knc52	286	NaN	7,200	NaN	7,200
knc53	286	286	1,273.46	286	23
knc54	289	291	194.67	289	107
knc55	300	300	27	300	2
knc56	287	NaN	7,200	300	7,200
knc57	284	286	396.28	284	3
knc58	286	286	148.27	286	42
knc59	300	300	54.38	300	2
knc60	300	300	9.93	300	3
knc61	285	290	351.79	286	7,200
knc62	300	300	8.14	300	4
knc63	286	288	131.14	286	2,330
knc64	288	288	2,240.72	288	37
knc65	284	NaN	7,200	287	7,200
knc66	284	284	6,579.75	284	57
knc67	300	300	942.82	300	38
knc68	300	300	1,720.46	300	52
knc69	300	300	2,042.26	300	3
knc70	300	300	68.8	300	7,200
knc71	300	NaN	7,200	NaN	183
knc72	300	300	2,124.95	300	2,762
knc73	300	NaN	7,200	NaN	7,200
knc74	300	300	7,200	300	34
knc75	300	NaN	7,200	NaN	7,200
knc76	300	NaN	7,200	NaN	7,200
knc77	300	NaN	7,200	300	273
knc78	300	NaN	7,200	NaN	7,200
knc79	300	NaN	7,200	NaN	7,200

5 Conclusion

This paper presented an attempt to identify key SMPTSP instance characteristics from an algorithmic perspective and graph coloring problem perspective. These instance characteristics were analysed for their influence in determining the hardness of the problem instances based on the performance of two state of the art algorithms available for the problem. Based on the inference, a challenging dataset has been generated and the performance of GFA and CMH algorithms on this dataset has been reported. Even though the influence of individual factors on the instance hardness has been studied, it could be the combination of multiple factors that may explain why certain instances are challenging than the others. This could be an interesting area for future study.

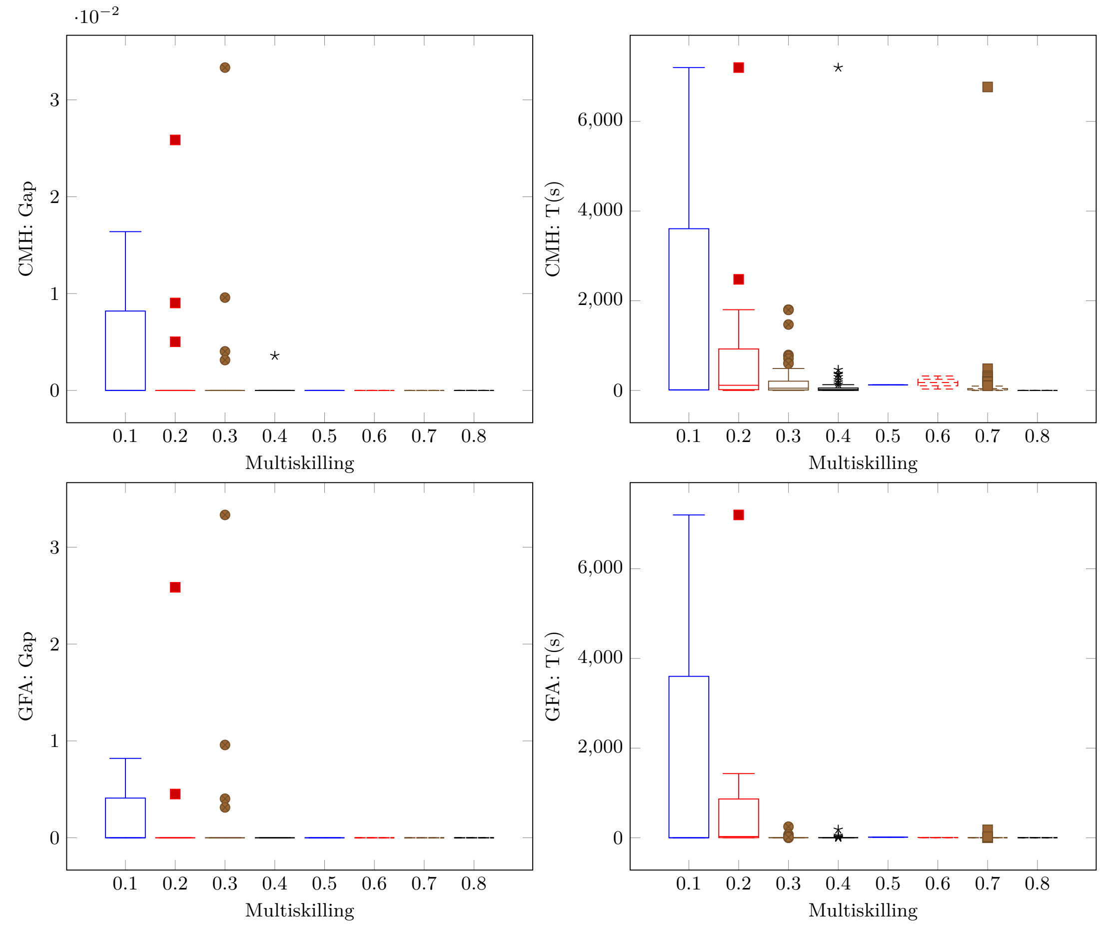

Figure 5: Influence of multiskilling on CMH and GFA performance

References

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*, 226:367–385, 5 2013. ISSN 03772217. doi: 10.1016/j.ejor.2012.11.029.

Reshma Chirayil Chandrasekharan, Pieter Smet, and Tony Wauters. An automatic constructive matheuristic for the shift minimization personnel task scheduling problem. *Journal of Heuristics*, 27:205–227, 4 2021. ISSN 15729397. doi: 10.1007/s10732-020-09439-9.

Jean Guillaume Fages and Tanguy Lapègue. Filtering atmostnvalue with difference constraints: Application to the shift minimisation personnel task scheduling problem. *Artificial Intelligence*, 212:116–133, 2014. ISSN 00043702. doi: 10.1016/j.artint.2014.04.001.

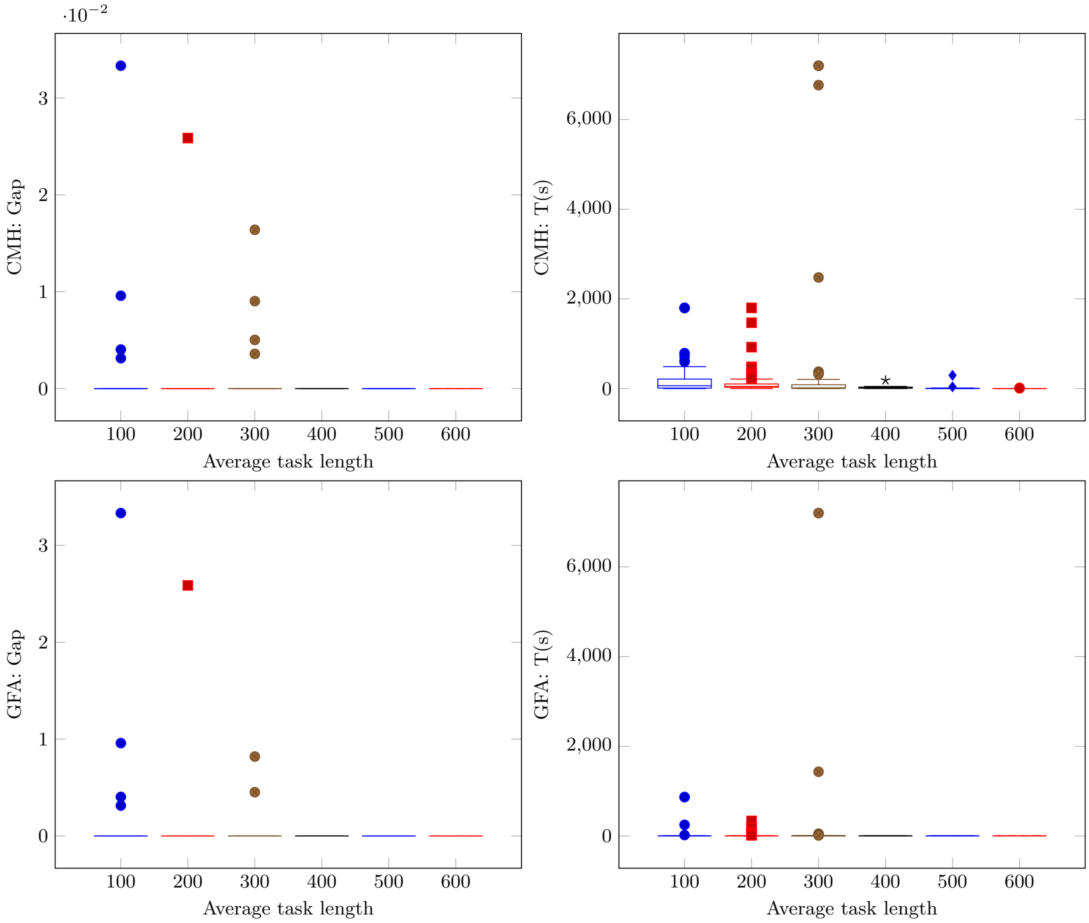

Figure 6: Influence of average task length on CMH and GFA performance

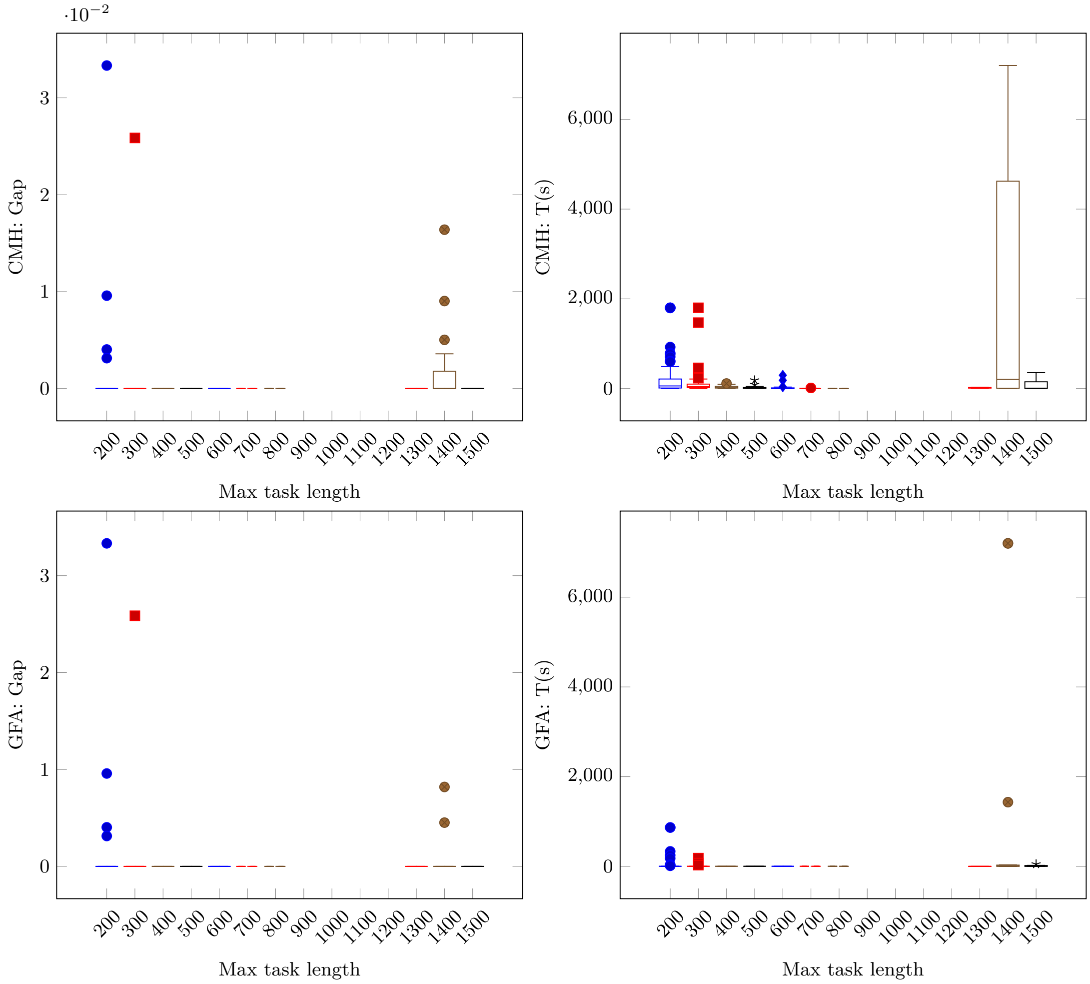

Figure 7: Influence of maximum task length on CMH and GFA performance

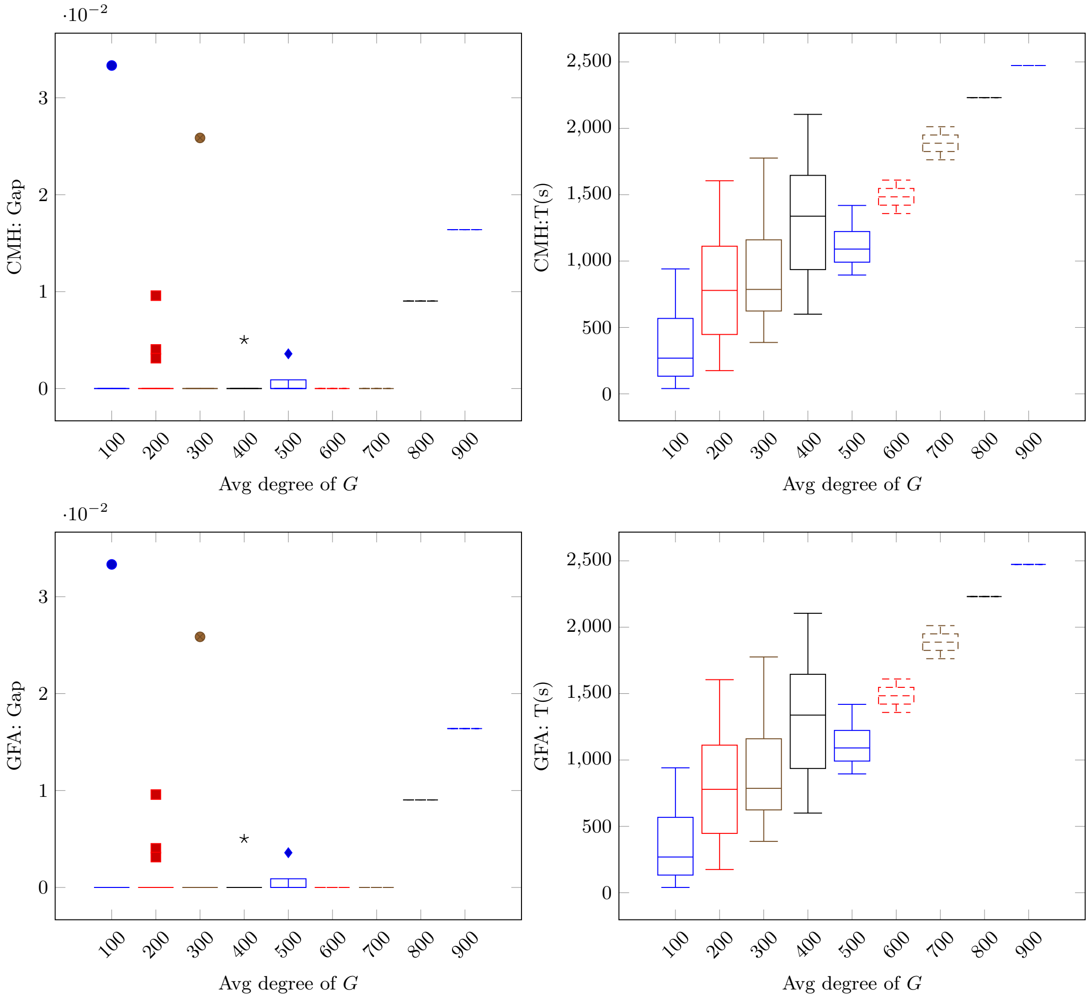

Figure 8: Influence of average degree of G, on CMH and GFA performance

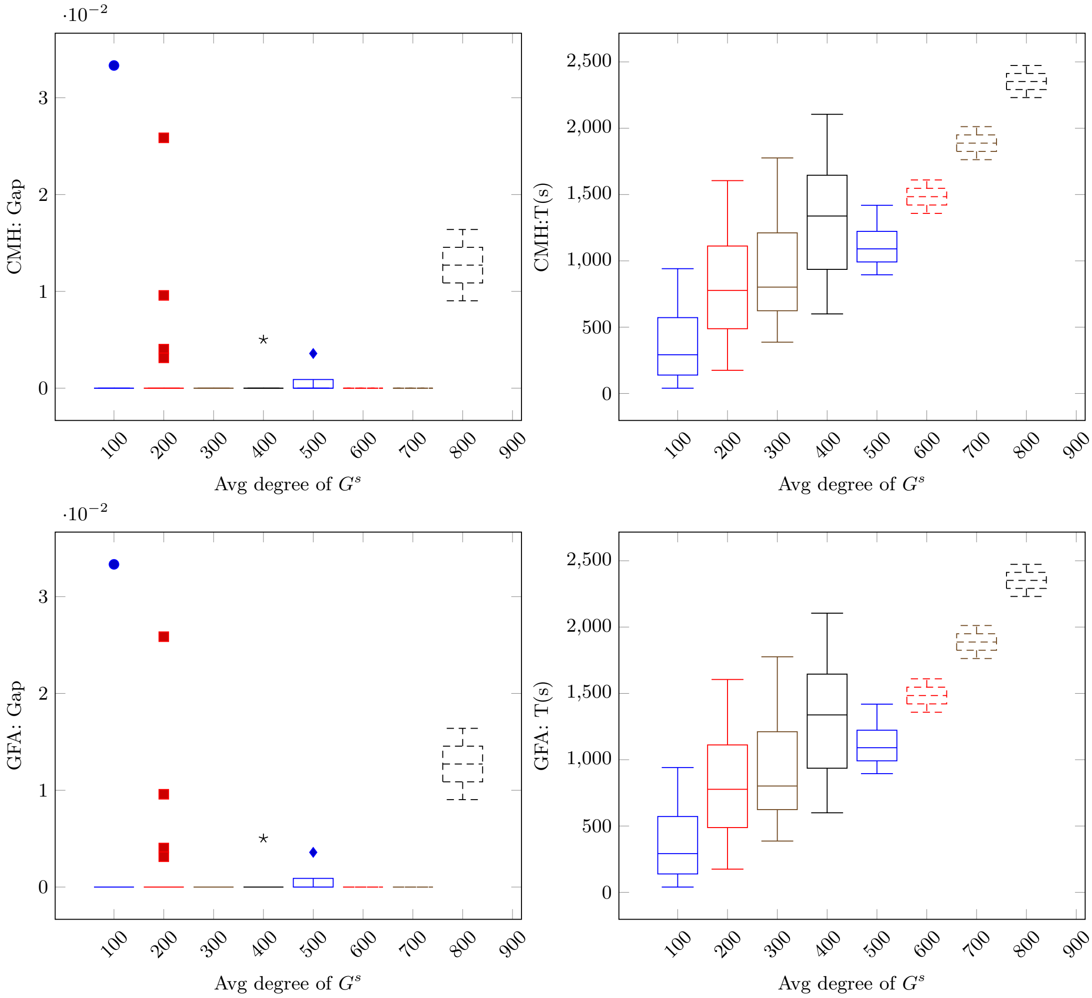

Figure 9: Influence of average degree of G^s, on CMH and GFA performance

M. Krishnamoorthy, A. T. Ernst, and D. Baatar. Algorithms for large scale shift minimisation personnel task scheduling problems. *European Journal of Operational Research*, 219:34–48, 5 2012. ISSN 03772217. doi: 10.1016/j.ejor.2011.11.034.

Nico Kyngäs and Kimmo Nurmi. The extended shift minimization personnel task scheduling problem. volume 26, pages 65–74. PTI, 9 2021. doi: 10.15439/2021f35.

Shih Wei Lin and Kuo Ching Ying. Minimizing shifts for personnel task scheduling problems: A three-phase algorithm. *European Journal of Operational Research*, 237:323–334, 8 2014. ISSN 03772217. doi: 10.1016/j.ejor.2014.01.035.

Pieter Smet, Tony Wauters, Mihail Mihaylov, and Greet Vanden Berghe. The shift minimisation personnel task scheduling problem: A new hybrid approach and computational insights. *Omega (United Kingdom)*, 46: 64–73, 7 2014. ISSN 03050483. doi: 10.1016/j.omega.2014.02.003.

K. Smith-Miles. Matilda, 2019. URL https://matilda.unimelb.edu.au/matilda/showMobileHomePage.

Ouz Solyali. The shift minimization personnel task scheduling problem: An effective lower bounding procedure. *Hacettepe University Journal of Economics and Administrative Sciences*, 34:115–132, 2016.