DOI: 10.17065/huniibf.259136

THE SHIFT MINIMIZATION PERSONNEL TASK SCHEDULING PROBLEM: AN EFFECTIVE LOWER BOUNDING PROCEDURE

Hacettepe University Journal of Economics and Administrative Sciences
Vol. 34, Issue 2, 2016
pp. 115-132

Oğuz SOLYALI
Assoc.Prof.Dr., Middle East Technical University
Business Programme
Northen Cyprus Campus
solyali@metu.edu.tr

Abstract: This study considers the shift minimization personnel task scheduling problem, which is to assign a set of tasks with fixed start and finish times to a minimum number of workers from a heterogeneous workforce. An effective lower bounding procedure based on solving a new integer programming model of the problem is proposed for the problem. An extensive computational study on benchmark data sets reveals that the proposed lower bounding procedure outperforms those existing in the literature and consistently and rapidly yields high quality lower bounds that are necessary for the decision makers to assess the quality of the obtained schedules.

Keywords: *Scheduling, integer programming, personnel task scheduling.*

DOI: 10.17065/huniibf.259136

VARDİYA ENKÜÇÜKLEYEN PERSONEL GÖREV ÇİZELGELEMESİ PROBLEMİ: ETKİN BİR ALT SINIR YÖNTEMİ

Hacettepe Üniversitesi
İktisadi ve İdari Bilimler
Fakültesi Dergisi,
Cilt 34, Sayı 2, 2016
s. 115-132

Oğuz SOLYALI
Doç.Dr., Orta Doğu Teknik Üniversitesi
İşletme Programı
Kuzey Kıbrıs Kampusu
solyali@metu.edu.tr

Öz: Bu çalışmada, başlangıç ve bitiş zamanları belli olan bir grup görevin, türdeş olmayan bir işgücünden en az sayıdaki çalışana atandığı bir vardiya enküçükleyen personel görev çizelgelemesi problemi ele alınmıştır. Bu problem için, problemin yeni bir tamsayılı programlama modelini çözmeye dayalı etkin bir alt sınır yöntemi önerilmiştir. Sayısal sonuçlar, önerilen modelin, literatürde varolan yöntemlerden daha üstün olduğunu ve karar vericilerin elde edilen çizelgelerin kalitelerini değerlendirebilmeleri için gerekli olan yüksek kaliteli alt sınırları tutarlı ve hızlı bir şekilde verdiğini göstermektedir.

Anahtar Sözcükler: *Çizelgeleme, tamsayılı programlama, personel görev çizelgelemesi.*

INTRODUCTION

Personnel scheduling can be defined as determining timetables of personnel and assignment of tasks to personnel. Because optimizing the process of personnel scheduling yields significant cost savings for firms, different variants of personnel scheduling problems arising in many areas like airlines, railways, call centers, and health care systems are considered in the literature (see e.g. Ernst *et al*. 2004). In this study, we consider a particular personnel scheduling problem, the shift minimization personnel task scheduling problem (SMPTSP), which is encountered in many practical cases like days-of-operations rostering (see Krishnamoorthy *et al*. 2012 for details).

The SMPTSP, introduced by Krishnamoorthy *et al*. (2012), is to assign a set of tasks with fixed start and finish times to a minimum number of workers from a heterogeneous workforce. Because personnel work in shifts with fixed start and finish times, and their shifts have already been determined in the SMPTSP, workers and shifts can be used interchangebly. Each worker can perform a set of tasks with one task at a time only if those tasks' start and finish times fit those of the worker and the worker has the required qualifications to perform those tasks. Each task has to be performed by one of the eligible workers without interruption.

The SMPTSP, a strongly NP-hard problem (Kroon *et al*., 1997), has been studied by Krishnamoorthy *et al*. (2012), Smet and Berghe (2012), Smet *et al*. (2014), Fages and Lapegue (2013), and Lin and Ying (2014). Krishnamoorthy *et al*. (2012) proposed Lagrangian relaxation-based upper and lower bounding procedures using a mixed integer programming (MIP) model of the problem (see Section 1) and assessed the performance of these procedures on a data set of 137 instances, referred to as KEB instances, which they generated considering real-life applications. The lower (resp. upper) bounding procedure of Krishnamoorthy *et al*. (2012) found lower (resp. upper) bounds that deviate on average 1.36% (resp. 4.50%) from the optimal values. Smet and Berghe (2012) developed a heuristic which starts with an initial feasible solution found by a construction heuristic and then improves it by optimally solving the MIP model of a randomly selected part of the problem. Their heuristic yielded solutions that deviate on average 0.56% from the optimal values. Smet *et al*. (2014) proposed a two-phase heuristic composed of a construction heuristic in the first phase and a local branching-based improvement heuristic in the second phase. The two-phase heuristic managed to solve all KEB instances optimally within a short time. Their computational results also revealed that the trivial lower bound based on the maximum number of overlapping tasks is equal to the optimal value on all KEB instances. They made an analysis on what makes the problem difficult with regard to finding good quality upper bounds and generated ten difficult instances, referred to as SWMB instances. Their two-phase heuristic solved five SWMB instances to optimality and found feasible solutions with an average gap of 0.99% from the optimal values. Fages and Lapegue (2013) proposed constraint programming-based upper and lower bounding procedures for the problem. Because the KEB and SWMB instances are trivial with regard to finding good quality lower bounds, Fages and Lapegue (2013) generated a new data set of 100 instances, referred to as FL instances, by considering the hardness results of Smet *et al*. (2014). The computational experiments of Fages and Lapegue (2013) on the FL data set showed that their lower bounds outperform those found by solving the MIP model of the problem with CPLEX and those of the lower bound proposed by Smet et al. (2014). Their constraint programming-based approach also produced high quality solutions for the KEB, SWMB and FL instances. Lin and Ying (2014) developed a three-phase heuristic for the problem. In the first phase, they obtain an initial solution using a simple construction heuristic which is then improved using an iterated greedy heuristic in the second phase. In the last phase, the objective function value found at the end of second phase is used as an initial upper bound while solving the MIP model of the problem with GUROBI to optimality. The computational experiments on the KEB instances indicated that the three-phase heuristic optimally solved 105 instances with an average gap of 0.38% over all KEB instances.

The SMPTSP is closely related to the fixed job scheduling (FJS) problems, where machines (resp. jobs) represent workers (resp. tasks) in the SMPTSP, and machines are identical and can process all jobs (i.e. no qualification requirements). FJS problems have both tactical and operational versions. In tactical FJS problems, all jobs are required to be processed and the aim is to minimize the cost of machines used whereas in operational FJS problems the aim is to maximize total value/weight associated with assigning a subset of jobs to a given set of machines. As shown by Gupta *et al.* (1979), the tactical FJS problem (or equivalently the SMPTSP problem without qualification requirements) can be solved in polynomial time by finding the maximum number of overlapping jobs (or tasks). Kroon *et al*. (1997) proposed exact and approximation procedures for a generalization of the tactical FJS problem in which machines are nonidentical and can process a subset of all jobs (i.e. a tactical FJS problem with qualification requirements). The problem considered in Kroon *et al*. (1997) is almost the same as the SMPTSP with the only difference being no availability restriction on machines in the former (i.e. the start and finish times of jobs are within those of the machines). Operational FJS problems with qualification requirements were considered by Kroon *et al*. (1995) and Eliiyi and Azizoğlu (2009). See Kovalyov *et al*. (2007) and Kolen *et al*. (2007) for reviews on fixed interval/job scheduling problems.

Although several lower bounding procedures have been proposed for the SMPTSP in the literature, none of them consistently provides high quality lower bounds that are necessary for the decision makers to assess the quality of the obtained schedules. To fill this gap, in this study, we propose a new integer programming model that provides a valid and effective lower bound for the SMPTSP. This model aims to find which workers are to be selected without taking into account which worker will perform which task. Because the constraints of the proposed model are exponential in number, this model is solved through branch-and-cut which means that instead of adding all constraints of the model a priori, we start with a small number of constraints and then identify and add only the violated constraints iteratively. We show that the problem of identifying the violated constraints can be solved in polynomial time, enabling us to obtain the solution of the proposed model fast. The computational experiments on benchmark data sets reveal that the proposed model yields lower bounds that outperform the existing ones in the literature. Moreover, the proposed lower bounding procedure is computationally fast.

The rest of the paper is organized as follows. In Section 1, we define our notation and provide a MIP model of the SMPTSP. We present our lower bounding procedure for the SMPTSP in Section 2. In Section 3, we present the computational results on benchmark data sets from the literature, and compare our lower bounds with those existing in the literature. Finally, we summarize and conclude the paper in Section 4.

1. PROBLEM DESCRIPTION AND FORMULATION

In the following we define our notation.

J: The set of tasks that must be performed.

W: The set of personnel available to perform tasks.

s_j: The start time of task $j \in J$.

f_j: The finish time of task $j \in J$.

T_w: The set of tasks that can be performed by worker $w \in W$.

P_j: The set of workers that can perform task $j \in J$.

$P(S)$: The set of workers that can perform at least one of the tasks in a set S.

K_t: The set of tasks that overlap in a time interval t. This set, known as a maximal clique, is maximal in that there is no other task that overlaps with all tasks in K_t at any given time interval. All maximal cliques (i.e. K_t $\forall t$) are found using the polynomial-time procedure in Krishnamoorthy et al. (2012). Note that $|K_t| = 1$ if there is only one task which does not overlap with other tasks in a time interval t.

K_t^w: The set of tasks that overlap in a time interval t and can be performed by worker $w \in W$. $K_t^w = K_t \cup T_w$.

C: The set that involves the sets K_t for all t. $C = \{K_1, K_2, \ldots\}$.

C^w: The set that involves the sets K_t^w for all *t*. $C^w = \{K_1^w, K_2^w, \dots\}$.

b_w: The fixed cost of selecting worker *w*.

y_w: Binary variable which is equal to 1 if worker *w* is selected, and 0 otherwise.

x_{jw}: Binary variable which is equal to 1 if task j is performed by worker *w*, and 0 otherwise.

The MIP model for the SMPTSP, given in Krishnamoorthy et al. (2012), is as follows:

$$\text{(F) Min} \sum_{w \in W} b_w y_w \qquad (1)$$

$$\text{s.t.} \sum_{w \in P_j} x_{jw} = 1 \qquad \forall j \in J \qquad (2)$$

$$\sum_{j \in K_t^w} x_{jw} \le y_w \qquad \forall w \in W, K_t^w \in C^w \qquad (3)$$

$$0 \le y_w \le 1 \qquad \forall w \in W \qquad (4)$$

$$x_{jw} \in \{0,1\} \qquad \forall j \in J, w \in P_j. \qquad (5)$$

The objective function (1) is to minimize the total cost of selected workers. Note that, as in Krishnamoorthy *et al*. (2012), we will consider $b_w = 1\ \forall w \in W$ (i.e., shift minimization) without loss of generality in the rest of the paper. Constraints (2) ensure that each task is assigned to one of the eligible workers. Constraints (3) stipulate that at most one of the tasks that overlap in a time interval can be assigned to a worker that is able to perform those tasks. Constraints (3) also guarantee that a worker is selected if any tasks are assigned to that worker. Constraints (4) ensure that a worker is selected at most once. Constraints (5) are for integrality of variables. Note that *y* variables will automatically be 0 or 1 due to (3) – (5).

2. A LOWER BOUNDING PROCEDURE

In this section, we propose an integer programming model that provides a lower bound for the SMPTSP. The integer programming model we propose for the SMPTSP is as follows:

$$\text{(RF) Min (1)}$$

$$\text{s.t.} \sum_{w \in P(S)} y_w \ge |S| \qquad \forall K_t \in C, S \subseteq K_t \qquad (6)$$

$$y_w \in \{0,1\} \qquad \forall w \in W. \qquad (7)$$

Because RF considers sets K_t separately and it does not take into account which worker will perform which task, it is a relaxed model for the SMPTSP. Constraints (6) ensure that the number of workers selected for performing a subset of overlapping tasks in any given time interval is no less than the cardinality of that subset. As constraints (6) are exponential in number, we will use all tasks in K_t, i.e., $S = K_t$, for (6), instead of considering all subsets of tasks in K_t, and the rest of the constraints (6) will be dynamically added if they are violated. Thus, we start with the following model:

(RF') Min (1)

s.t. (7),

$$\sum_{w \in P(K_t)} y_w \geq |K_t| \qquad \forall K_t \in C. \tag{8}$$

Then, as a standard branch-and-cut implementation, at each node of the branch-and-bound tree, the following MIP model for each K_t is solved to detect if any of the rest of the constraints (6) are violated by the current solution:

$$\text{(SP) Max } \sum_{j \in K_t} u_j - \sum_{w \in P(K_t)} \hat{y}_w v_w \tag{9}$$

$$\text{s.t. } u_j \leq v_w \qquad \forall j \in K_t, w \in P_j \tag{10}$$

$$u_j \geq 0 \qquad \forall j \in K_t \tag{11}$$

$$v_w \in \{0,1\} \qquad \forall w \in P(K_t), \tag{12}$$

where $\hat{y}_w$ for $w \in W$ is obtained by solving the current RF (i.e., RF' with possibly some violated constraints (6)), u_j is equal to 1 if task j is selected to the subset of overlapping tasks in K_t, and 0 otherwise, and v_w is equal to 1 if worker *w* can perform at least one of the tasks that are selected to the subset of overlapping tasks in K_t, and 0 otherwise.

The objective function (9) is the right-hand side of (6) less left-hand side of (6), i.e., the cardinality of selected subset of overlapping tasks less the number of workers selected for performing the same subset. If the objective function value of SP for K_t is positive, it indicates that constraints (6) are violated. Constraints (10) ensure that each worker that can perform at least one of the tasks in the selected subset of overlapping tasks in K_t is included only once to the number of workers selected for performing the same subset. Constraints (10) together with (12) also ensure that $u_j \leq 1 \ \forall j \in K_t$. Constraints (11) are for nonnegativity of u_j variables and constraints (12) are for integrality of v_w variables. Note that u_j variables will automatically take value 0 or 1. Given the optimal solution of SP for K_t, denoted by u_j^* for $j \in K_t$ and v_w^* for $w \in W$, the constraint $\sum_{w \in P(K_t)} v_w^* y_w \geq \sum_{j \in K_t} u_j^*$ is added to the current RF if the optimal objective function value of SP for K_t is positive.

Although the SP model for each K_t seems to be a MIP model, we show that it suffices to solve its linear programming relaxation (i.e. the removal of integrality restrictions on the *v* variables) in the following theorem. Thus, the SP model for each K_t can be solved efficiently.

Theorem 1: The linear programming relaxation of SP always yields an integral solution.
Proof: See Appendix.

In order to clarify the notation in RF and RF', we next present an example instance of the problem for which we derived the RF and RF' models.

Example 1: Consider a four-worker instance where the set of tasks with their start and finish times is given in Table 1 and the four workers ($W = \{1,2,3,4\}$) can perform the following tasks: $T_1 = \{2,3\}$, $T_2 = \{1,3\}$, $T_3 = \{1,2\}$, and $T_4 = \{4\}$.

Table 1. Start and Finish Times of Tasks in Example 1.

Task (j):	1	2	3	4
s_j:	2	7	10	20
f_j:	5	18	25	28

Using the given data, one can easily derive the following information: $K_1 = 1$, $K_2 = \{2,3\}$, $K_3 = \{3,4\}$, $P_1 = \{2,3\}$, $P_2 = \{1,3\}$, $P_3 = \{1,2\}$, $P_4 = \{4\}$.

The RF model is explicitly written for the example instance as follows:

Min $y_1 + y_2 + y_3 + y_4$

s.t. $y_2 + y_3 \geq 1$ for $S = K_1 = \{1\}$

$y_1 + y_3 \geq 1$ for $S = \{2\}, S \subset K_2$
$y_1 + y_2 \geq 1$ for $S = \{3\}, S \subset K_2$ or $S \subset K_3$
$y_1 + y_2 + y_3 \geq 2$ for $S = K_2 = \{2,3\}$

$y_4 \geq 1$ for $S = \{4\}, S \subset K_3$

$y_1 + y_2 + y_4 \geq 2$ for $S = K_3 = \{3,4\} y_w \in \{0,1\}$for $w = \{1,2,3,4\}$

For the example instance, the RF' model is explicitly written as follows:

Min $y_1 + y_2 + y_3 + y_4$

s.t. $y_2 + y_3 \geq 1$ for $S = K_1 = \{1\}$

$y_1 + y_2 + y_3 \geq 2$ for $S = K_2 = \{2,3\}$

$y_1 + y_2 + y_4 \geq 2$ for $S = K_3 = \{3,4\}$

$y_w \in \{0,1\}$ for $w = \{1,2,3,4\}$

While the optimal objective function value of RF is 3, that of RF' is 2. It is easy to see by inspection that the optimal solution of RF' is $y_1 = y_2 = 1$, which violates $y_4 \geq 1$ constraint in RF. Thus, in our branch-and-cut implementation, $y_4 \geq 1$ constraint is found to be violated after the RF' is first solved. Then, RF' with the violated constraint added is solved again and the optimal objective function value of RF, which is also the optimal value for the example instance, is obtained.

Note that the maximal clique-based trivial lower bound of Smet et al. (2014), referred to as CLB, is equal to $max_t\{|K_t|\}$, i.e. the maximum number of overlapping tasks. Because RF' is a generalization of CLB, it is guarantee that RF' yields an equivalent or better lower bound than the CLB.

3. COMPUTATIONAL RESULTS

We have performed computational experiments on benchmark data sets introduced by Krishnamoorthy et al. (2012), Smet et al. (2014) and Fages and Lapegue (2013) in order to assess the effectiveness of our lower bounding procedure and compare it with those existing in the literature and the solution of model F using CPLEX 12.5. We have coded the F, RF and RF' models, and CLB in C++, and solved models using Concert Technology of CPLEX 12.5 with its default settings using two threads. In the branch-and-cut implementation of RF, we treat the constraints (6) as lazy constraints which are checked if violated only when an integer feasible solution is found. We have performed the experiments on a 2.4 GHz Workstation with 48 GB RAM and 12 cores operating on Windows 7 (64-bit). A time limit of 1800 seconds is set for all procedures and models.

An overview of the benchmark instances that are used in the computational experiments are given in Table 2. Each instance has a filename in the form of *Data_no_|W|_|J|_msl* indicating the generator by *Data*, the number of the instance by *no*, the number of workers by |*W*|, the number of tasks by |*J*|, and the multi-skilling level by *msl*, respectively. To illustrate, an instance with the filename FL_079_638_1052_25 is the 79th instance generated by Fages and Lapegue (2013), has 638 workers and 1052 tasks with each worker being able to perform about 25% of all tasks on average. All instances are available at a web page: *https://sites.google.com/site/ptsplib/smptsp/home*.

The interested reader can refer to Krishnamoorthy et al. (2012), Smet *et al*. (2014) and Fages and Lapegue (2013) for the details on their instance generation schemes.

For the sake of convenience, here we repeat the abbreviations of the procedures and models that are frequently used in the following tables. We use F to denote the lower bounds obtained by solving the F model to optimality, CP to denote the lower bounds obtained by the constraint programming approach of Fages and Lapegue (2013), CLB to denote the lower bound of Smet et al. (2014), RF' and RF to denote the lower bounds obtained by solving the proposed RF' and RF models, respectively.

Table 2. Overview of Benchmark Instances

Data set	KEB[a]	SWMB[b]	FL[c]
Number of instances	137	10	100
Range of number of workers	[22,420]	[44,153]	[69,948]
Range of number of tasks	[40,2105]	[258,1577]	[71,1605]
Multi-skilling level[d]	33%, 66%	20%, 30%	25%

[a] Krishnamoorthy *et al.* (2012)
[b] Smet *et al.* (2014)
[c] Fages and Lapegue (2013)
[d] Each worker can perform a certain percent of all tasks on average.

We start with assessing the quality of lower bounding procedures proposed by different researchers on the KEB data set. We present the summary of lower bound results on the KEB data set in Table 3, where the first row shows the lower bounding procedures, the second row the number of instances in which the lower bound is equal to the optimal value, the third row the average of lower bounds over 137 instances, and the fourth row the average time (in seconds) needed to obtain the lower bounds over 137 instances.

Table 3. Summary of Lower Bound Results on 137 KEB Instances

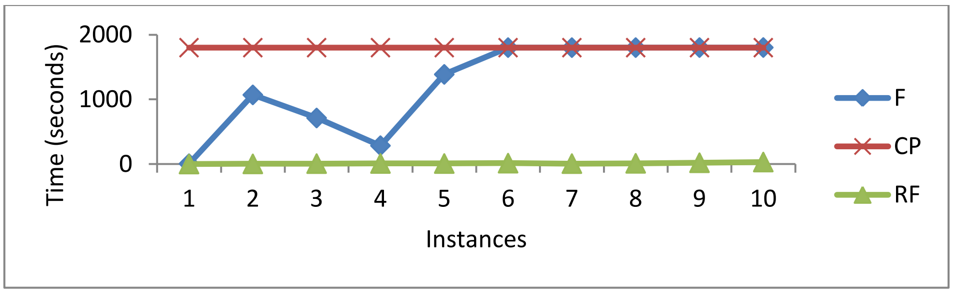
Lower bounding procedure	VA[a]	CLB	F	CP[b]	RF'	RF
Number of optimal solutions	39	137	98	137	137	137
Average lower bound value	120.2	122.2	75.3	122.2	122.2	122.2
Average time (seconds)	114.0	0.1	795.2	751.7	0.2	11.8

[a] Performed on an 2.93 GHz PC by Krishnamoorthy *et al*. (2012)
[b] Performed on an Intel Xeon CPU 2.67 GHz PC by Fages and Lapegue (2013)

Results given in Table 3 show that the lower bounds found by CLB, CP, RF' and RF are equal to the optimal values in all KEB instances. CPLEX 12.5 found lower bounds equal to the optimal values in 98 KEB instances out of 137 but could not find any lower bound greater than 0 within the time limit in the rest of the KEB instances. The volume algorithm (VA) of Krishnamoorthy et al. (2012) yielded inferior lower bounds than those found by CLB, CP, RF' and RF. Specifically, VA found lower bounds that deviate on average 1.36% from the optimal values and are optimal in only 39 instances out of 137.

Because CLB, F, CP, RF' and RF found the same lower bounds in all SWMB instances, we did not report any lower bound result. However, we present the computing times of F, CP and RF on the SWMB instances in Figure 1, which shows that RF was solved quite fast whereas CP fully used the allowed time in all instances and F reached the time limit as the size of the instances increased. Note that because the computing times for CLB and RF' were negligible, they were not plotted in Figure 1.

Figure 1. Computing Times of F, CP, and RF on 10 SWMB Instances



Thus, as also stated by Fages and Lapegue (2013) and in Introduction, the KEB and SWMB instances are trivial with regard to finding good quality lower bounds. Therefore, we assess in detail all lower bounding procedures on the FL instances which were generated by Fages and Lapegue (2013) as challenging instances with regard to finding good quality lower bounds.

Table 4. Summary of Lower Bound Results on 100 FL Instances

Lower bounding procedure	CLB	F	CP[a]	RF'	RF
Number of optimal solutions	0	71	44	60	98
Average lower bound value	63.3	86.1	162.8	162.1	164.4
Average time (seconds)	0.1	880.5	1136.5	0.3	20.7

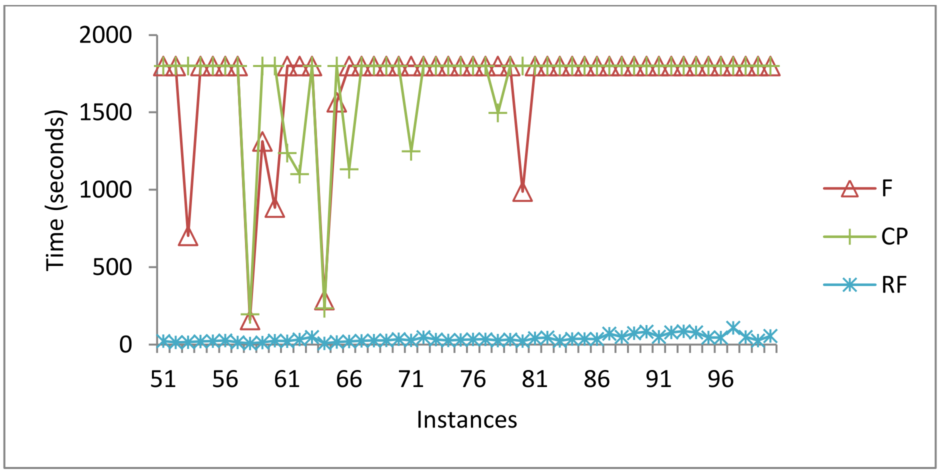
[a] Performed on an Intel Xeon CPU 2.67 GHz PC by Fages and Lapegue (2013)

We present the summary of lower bound results on the FL instances in Table 4, where the first row shows the lower bounding procedures, the second row the number of instances in which the lower bound is equal to the optimal value, the third row the average of lower bounds over 100 instances, and the fourth row the average time (in seconds) needed to obtain the lower bounds over 100 instances.

Results presented in Table 4 reveal that RF outperformed its competitors with regard to the quality of lower bounds. Specifically, RF managed to find the optimal objective value in all instances except for two. CLB yielded very poor lower bounds compared to others. While F found the best lower bounds in 71 instances out of 100, it could not find any lower bounds greater than zero in the rest of the instances within the time limit. That is why F performed better than CP with regard to the number of optimal solutions but not for the average lower bound value. CP exhibited the second best performance regarding the quality of lower bounds. See Appendix B for the detailed results of lower bounding procedures on the FL instances.

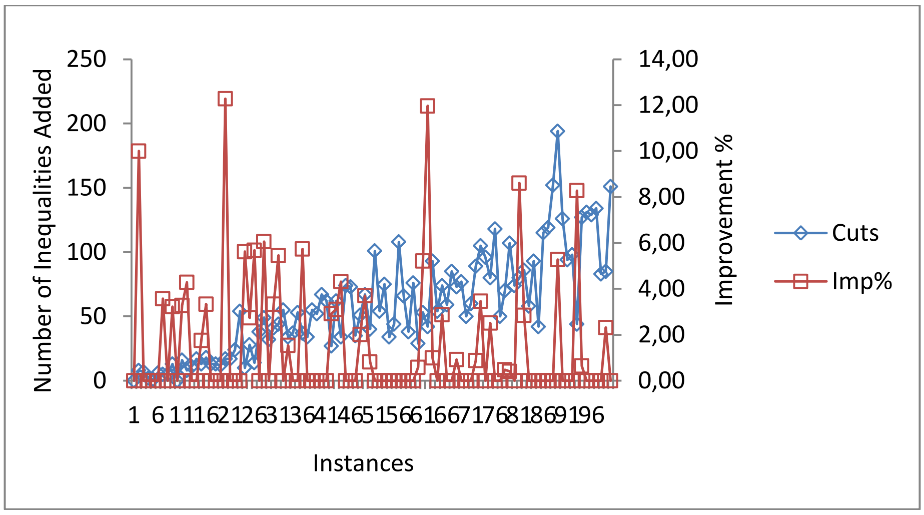
Besides yielding high quality lower bounds, RF is computationally fast with its average time being less than 21 seconds. Although CP was implemented in a different computing environment, it seems that RF is much faster than CP. Being implemented on the same computing environment, RF is significantly faster than F. Like KEB and SWMB data sets, the computing times needed by CLB and RF' were less than a second for the FL instances. We present the computing times of F, CP and RF on the larger FL instances (i.e. instances 51–100) in Figure 2, which indicates that F and CP reached to the time limit for the majority of these FL instances whereas the largest computing time for RF is less than two minutes.

Figure 2. Computing times of F, CP, and RF on FL instances 51–100



In order to show the magnitude of improvement obtained from adding inequalities (6) to RF' and the computational burden of adding these inequalities, we present the number of inequalities added (Cuts) and the percentage improvement (Imp%) obtained by implementing RF over RF' on the FL instances in Figure 3. As seen from Figure 3, significant improvements in lower bounds have been achieved within short times by adding a small number of inequalities (6) to RF'. There are several instances for which RF' and RF yielded the same lower bound though some number of inequalities (6) have been added. We should note that there are no inequalities (6) identified for any of the KEB and SWMB instances. Finally, we should also note that the RF' and RF models were solved at the root node without any need for branching of variables in all KEB, SWMB and FL instances.

Figure 3. Number of Inequalities Added and Improvement% Obtained by RF over RF' for the FL Instances



CONCLUSION

We have addressed the shift minimization personnel task scheduling problem by proposing a lower bounding procedure based on solving a new integer programming model of the problem through branch-and-cut. The computational results on benchmark data sets have shown that our lower bounding procedure outperformed those existing in the literature in terms of both the quality of bound and the computing time. Thus, the decision makers are given an effective tool for assessing the quality of the schedules they obtain.

As a future research avenue, one can address more complicated personnel task scheduling problems. For instance, studying the problem of assigning workers to shifts besides the decisions made in the SMPTSP would be useful and interesting. Another future work may involve adapting the proposed lower bounding procedure to the variants of personnel task scheduling problems and fixed interval/job scheduling problems.

REFERENCES

Eliiyi, D.T., M. Azizoğlu (2009), "A Fixed Job Scheduling Problem with Machine-Dependent Job Weights", *International Journal of Production Research*, 47, 2231–2256.

Ernst, A.T., H. Jiang, M. Krishnamoorthy, B. Owens, D. Sier (2004), "An Annotated Bibliography of Personnel Scheduling and Rostering", *Annals of Operations Research*, 127, 21–144.

Fages, J.G., T. Lapegue (2013), "Filtering Atmostnvalue with Difference Constraints: Application to the Shift Minimisation Personnel Task Scheduling Problem", *Lecture Notes in Computer Science*, 8124, 63–79.

Gupta, U.L., D.T. Lee, J.T. Leung (1979), "An Optimal Solution for the Channel-Assignment Problem", *IEEE Transactions on Computers*, 28, 807–810.

Kolen, A.W.J., J.K. Lenstra, C.H. Papadimitriou, F.C.R. Spieksma (2007), "Interval Scheduling: A survey", *Naval Research Logistics*, 54, 530–543.

Kovalyov, M.Y., C.T. Ng, T.C.E. Cheng (2007), "Fixed Interval Scheduling: Models, Applications, Computational Complexity and Algorithms", *European Journal of Operational Research*, 178, 331–342.

Krishnamoorthy, M., A.T. Ernst, D. Baatar (2012), "Algorithms for Large Scale Shift Minimisation Personnel Task Scheduling Problems", *European Journal of Operational Research*, 219, 34–48.

Kroon, L.G., M. Salomon, L.N.V. Wassenhowe (1995), "Exact and Approximation Algorithms for the Operational Fixed Interval Scheduling Problem", *European Journal of Operational Research*, 82, 190–205.

Kroon, L.G., M. Salomon, L.N.V. Wassenhowe (1997), "Exact and Approximation Algorithms for the Tactical Fixed Interval Scheduling Problem", *Operations Research*, 45, 624–638.

Lin, S.-W., K.C. Ying (2014), "Minimizing Shifts for Personnel task Scheduling Problems: A three-Phase Algorithm", *European Journal of Operational Research*, 237, 323–334.

Nemhauser, G.L., L.A. Wolsey (1988), *Integer and Combinatorial Optimization*, New York: Wiley.

Smet, P., G. Vanden Berghe, "A Matheuristic Approach to the Shift Minimization Personnel task Scheduling Problem", 9th International Conference on the Practice and Theory of Automated Timetabling, 2012, 145–160.

Smet, P., T. Wauters, M. Mihaylov, G. Vanden Berghe (2014), "The Shift Minimization Personnel Task Scheduling Problem: A new Hybrid Approach and Computational Insights", *OMEGA*, 46, 64–73.

APPENDIX A. PROOF OF THEOREM 1

The coefficient matrix **A'** of the linear programming relaxation of SP is of the form **(A,I)** where the matrix **A** is due to constraints (10) and the identity matrix **I** is due to $v_w \leq 1\ \forall w \in P(K_t)$. It suffices to show that the transpose of **A** (i.e., $\mathbf{A^T}$) is totally unimodular (see p.540, Proposition 2.1 of Nemhauser and Wolsey, 1988). As all constraint coefficients are 0, –1 or +1, and each column of $\mathbf{A^T}$ contains one –1 and one +1 coefficient, $\mathbf{A^T}$ is totally unimodular (see, p.542, Proposition 2.6 of Nemhauser and Wolsey, 1988). Hence, the linear programming relaxation of SP yields integral solution.

APPENDIX B. DETAILED RESULTS ON THE FL DATA SET

The detailed results of lower bounding procedures on the FL data set are provided in Tables 5–7, where column 1 indicates the name of instances, column 2 the optimal objective function value (z^{opt}), and columns 3–7 the lower bounds yielded by CLB, F, CP, RF' and RF, respectively. The values in z^{opt} column are obtained by the heuristic of Lin and Ying (2014) upon request by the authors.

Table 5. Results on the FL instances 1–33

Instance	**z^{opt}**	**CLB**	**F**	**CP**	**RF'**	**RF**
FL_001_0062_0071_025	29	13	29	29	29	29
FL_002_0080_0105_025	33	13	33	33	30	33
FL_003_0082_0093_025	31	13	31	31	31	31
FL_004_0078_0086_025	29	13	29	29	29	29
FL_005_0081_0110_025	31	11	31	31	30	30
FL_006_0084_0094_025	31	12	31	31	31	31
FL_007_0076_0094_025	29	11	29	29	28	29
FL_008_0083_0097_025	33	14	33	33	33	33
FL_009_0082_0117_025	32	12	32	32	31	32
FL_010_0072_0095_025	28	11	28	28	28	28
FL_011_0164_0228_025	63	27	63	63	61	63
FL_012_0186_0273_025	73	35	73	73	70	73
FL_013_0138_0212_025	52	19	52	52	52	52
FL_014_0182_0232_025	69	32	69	69	69	69
FL_015_0152_0204_025	58	24	58	58	57	58
FL_016_0161_0235_025	62	23	62	62	60	62
FL_017_0164_0217_025	62	27	62	62	62	62
FL_018_0147_0204_025	57	20	57	57	57	57
FL_019_0150_0196_025	55	27	55	55	55	55
FL_020_0152_0235_025	64	24	64	64	57	64
FL_021_0197_0292_025	79	31	79	79	79	79
FL_022_0288_0456_025	110	41	110	109	110	110
FL_023_0287_0414_025	110	41	110	108	110	110
FL_024_0236_0362_025	94	37	94	93	89	94
FL_025_0192_0311_025	75	29	75	75	73	75
FL_026_0212_0376_025	93	33	93	93	88	93
FL_027_0282_0437_025	107	38	107	107	107	107
FL_028_0262_0402_025	105	39	105	105	99	105
FL_029_0248_0355_025	95	40	95	93	95	95
FL_030_0236_0375_025	93	39	93	93	90	93
FL_031_0290_0488_025	116	38	116	115	110	116
FL_032_0332_0527_025	127	55	127	125	127	127
FL_033_0338_0534_025	132	45	132	132	130	132

Table 6. Results on the FL instances 34–66

Instance	**z^{opt}**	**CLB**	**F**	**CP**	**RF'**	**RF**
FL_034_0300_0468_025	114	45	114	113	114	114
FL_035_0308_0469_025	118	43	118	117	118	118
FL_036_0320_0535_025	129	52	129	129	122	129
FL_037_0296_0437_025	115	60	115	114	115	115
FL_038_0340_0525_025	129	51	129	128	129	129
FL_039_0284_0446_025	108	41	108	108	108	108
FL_040_0384_0576_025	147	55	147	144	147	147
FL_041_0358_0556_025	137	54	137	136	137	137
FL_042_0360_0601_025	141	51	141	140	137	141
FL_043_0422_0674_025	166	58	166	166	161	166
FL_044_0364_0572_025	145	53	145	145	139	145
FL_045_0376_0586_025	144	60	144	143	144	144
FL_046_0409_0635_025	157	66	157	155	157	157
FL_047_0373_0600_025	142	58	142	142	142	142
FL_048_0390_0614_025	152	56	152	152	149	152
FL_049_0354_0549_025	140	55	140	140	135	140
FL_050_0318_0536_025	123	49	123	123	122	123
FL_051_0514_0832_025	197	68	197	192	197	197
FL_052_0448_0767_025	171	60	171	169	171	171
FL_053_0486_0746_025	186	66	186	183	186	186
FL_054_0498_0850_025	190	67	190	189	190	190
FL_055_0524_0920_025	201	79	201	198	201	201
FL_056_0538_0911_025	206	73	206	203	206	206
FL_057_0440_0737_025	169	61	169	168	169	169
FL_058_0348_0562_025	132	54	132	132	132	132
FL_059_0460_0689_025	176	81	176	175	176	176
FL_060_0443_0783_025	173	68	173	172	172	173
FL_061_0551_0891_025	222	83	222	222	211	222
FL_062_0610_1096_025	262	90	0	262	234	262
FL_063_0524_0905_025	203	76	203	201	201	203
FL_064_0366_0570_025	140	49	140	140	140	140
FL_065_0456_0764_025	179	67	179	176	174	179
FL_066_0492_0775_025	189	70	189	189	189	189

Table 7. Results on the FL instances 67–100

Instance	z^{opt}	CLB	F	CP	RF'	RF
FL_067_0597_0949_025	230	90	0	228	230	230
FL_068_0561_0958_025	219	102	219	217	217	219
FL_069_0550_0891_025	211	75	211	207	211	211
FL_070_0550_0990_025	211	76	0	209	211	211
FL_071_0528_0895_025	202	79	0	202	202	202
FL_072_0604_0997_025	230	82	230	229	228	230
FL_073_0604_0999_025	239	87	0	236	231	239
FL_074_0563_0941_025	217	80	0	216	217	217
FL_075_0518_0870_025	204	76	204	200	199	204
FL_076_0642_1107_025	246	93	0	244	246	246
FL_077_0648_1123_025	248	95	0	248	248	248
FL_078_0548_0941_025	210	97	0	210	209	210
FL_079_0638_1052_025	246	89	0	245	245	246
FL_080_0578_0885_025	222	93	222	220	222	222
FL_081_0647_1181_025	265	108	0	264	244	265
FL_082_0734_1265_025	289	108	0	286	281	289
FL_083_0551_0941_025	206	91	0	203	206	206
FL_084_0644_1121_025	247	90	0	243	247	247
FL_085_0688_1111_025	263	101	0	256	263	263
FL_086_0628_1136_025	241	92	0	238	241	241
FL_087_0765_1323_025	288	102	0	281	288	288
FL_088_0808_1346_025	310	126	0	310	310	310
FL_089_0790_1371_025	320	102	0	317	303	319
FL_090_0810_1443_025	312	117	0	308	312	312
FL_091_0754_1287_025	289	119	0	283	289	289
FL_092_0948_1583_025	364	143	0	355	364	364
FL_093_0820_1478_025	340	107	0	329	314	340
FL_094_0812_1394_025	313	114	0	311	311	313
FL_095_0696_1162_025	266	127	0	259	266	266
FL_096_0710_1250_025	272	106	0	269	272	272
FL_097_0886_1605_025	340	120	0	335	340	340
FL_098_0750_1334_025	289	107	0	286	289	289
FL_099_0564_0955_025	221	95	221	218	216	221
FL_100_0806_1374_025	310	130	0	306	310	310