Network based formulations for roster scheduling problems

**Hanjing Zhang · Lisa Jackson ·
Antuela Tako · Jiyin Liu ·
Sarah Dunnett**

Abstract Roster scheduling problems are about sequences of working shifts and rest periods with respect to demand coverage and work preferences. Identifying efficient modelling structures gives a better understanding of roster scheduling problems. It enables efficient problem specific algorithms to be designed. This paper presents a refined integer programming formulation for roster scheduling problems which is based on two connected minimum cost network flows. Because of the integrity property of network structures, both the network flows can be solved in polynomial time. The common variables denoting work status link these networks to each other. The refined formulation is compared with other formulations by solving a set of roster scheduling problems in literature. Computational experiments indicate that the refined formulation provides an efficient structure in terms of solution quality and computation time for roster scheduling problems.

Keywords Roster scheduling problems · Network flows · Integer programming

1 Introduction

With a reduction of government funding of public service systems, managers face an unprecedented challenge for assigning shifts to employees in order to maintain high quality services. During the last decades, lots of researchers have worked on algorithms for assigning a sequence of shifts and rest over the planning horizon for employees to take. In practice, rosters are usually generated

H. Zhang; J. Liu and A. Tako
School of Business and Economics, Loughborough University
E-mail: H.Zhang@lboro.ac.uk; J.Y.Liu@lboro.ac.uk; A.Takou@lboro.ac.uk

L. Jackson; S. Dunnett
Aeronautical and Automotive Engineering Department, Loughborough University
E-mail: L.M.Jackson@lboro.ac.uk; S.J.Dunnett@lboro.ac.uk

Proceedings of the 11th International Confenference on Practice and Theory of Automated Timetabling (PATAT-2016) – Udine, Italy, August 23–26, 2016

manually by experienced managers, which takes a considerable amount of time and effort to produce a feasible set of rosters and the roster quality cannot be guaranteed. Computer aided methods have been developed in recent years to improve the efficiency in finding good solutions, but it is still hard to find the optimal rosters as typical roster scheduling problems are NP-complete problems (Lau, 1996). The difficulty of typical roster scheduling problems comes from constraints on particular shift assignments (Osogami and Imai, 2000) and consecutive days of work (Brunner et al, 2013).

Early research in roster scheduling mainly focused on integer programming models of assignment formulations (Ryan and Foster, 1981). These models are effective in solving small scale roster problems, but they are too computationally expensive to deal with large scale problems. Meta-heuristic approaches have been widely studied over the last two decades. Approaches including tabu search, genetic algorithm and variable depth search have had significant successes in solving large scale roster problems (Brucker et al, 2011). However, the quality of rosters cannot be guaranteed because solutions obtained by meta-heuristic approaches always depend on the random selection of initial seed rosters. Another disadvantage of heuristic methods is that if they fail to find any feasible solutions, it cannot be determined whether the heuristic method is too simple to find a solution or there is no feasible solution for this problem. Meanwhile, attention has been paid on identifying efficient modelling structures for simplified roster scheduling problems (Smet et al, 2016). These simplified problems are usually formulated in network structures so that optimal solutions can be found within polynomial time. As some important practical constraints are ignored, results may not be able to be applied in practice. In this paper, two network structures are built to accommodate separate subsets of roster scheduling constraints. Based on the network structures, an integer programming model is proposed to efficiently solve complex roster scheduling problems of large scales.

This paper is organized as follows. Section 2 describes roster scheduling problems using six different characteristics which reflect both employees' working preferences and system requirements for maintaining service quality. Section 3 reviews some traditional integer programming formulations for roster scheduling problems. Section 4 describes two network structures and introduces a new network based integer programming model for roster scheduling. Numerical experiments are presented in Section 5 to prove the computational efficiency of the proposed integer programming model. Section 6 gives the conclusion and identifies future work.

2 Problem Description

Ernst et al (2004) reviewed a large number of papers and provided some basic criteria to classify roster scheduling problems. Inspired by their work, a roster scheduling problem can be defined by the following six characteristics:

- *Problem Dimension* defines the scale of a scheduling problem, including the scheduling period, the number of different shifts and the number of employees involved;
- *Series Constraints* describe rotation rules among shifts and rest over consecutive periods.
- *Contract Constraints* record shift assignment limitations for each employee over the scheduling period.
- *Coverage Constraints*, sometimes referred to as *Demand Constraints*, indicate the required staff levels on all shifts over the scheduling period.
- *Coverage Penalty* weights the importance of minimising under and over staffing.
- *Work Preference* personalises desirable rosters with penalty.

A shift-stretch is defined as a set of consecutive days working on shifts without any days-off. Similarly, a rest-stretch is defined as a rest which consist of several days-off. Series constraints are the constraints which restrict feasible domains of shift-stretches and rest-stretches. The most common series constraints are about forward rotation, which means that rotations containing a shift of a day followed by any earlier shifts of the next day are forbidden unless there are some days off in between. Enough rest between two consecutive shifts needs to be guaranteed. For example, in a typical round-the-clock service system with three eight-hour shifts (Early, Late and Night), an Early shift cannot be assigned to the same employee who just finished a Late or Night shift on the day before. Series Constraints also impose upper and lower bounds on the length of valid shift-stretches so that employees will neither have too long nor too short periods of consecutive working days. Sometimes, the number of consecutive days-off in feasible rest-stretches also have a predefined minimum and maximum.

Contract constraints are related to work allocation on specific days or specific shifts. A good example of day specific contract constraint is "*at least A out of every B Sundays off during the scheduling period*". Shift specific contract constraints will define a range for each type of shift so a fair distribution of popular and unpopular shifts among employees can be guaranteed.

Coverage constraints are the most common constraints in roster scheduling problems. These constraints identify required staff levels for each shift on each day so that temporal service quality can be met. Usually a small difference between the number of staff scheduled for each shift on each day and the preferred staff level is permitted since managers may compromise service quality with limited operation budgets.

Coverage penalty is introduced to push scheduled staff levels moving towards preferred staff levels. For each shift on each day, two coverage penalties are defined in the number of staff below and above the preferred staff level. In practice, the coverage penalty is not proportional to the number of staff below the preferred staff level since the larger the over staffing level the more serious consequences it will have. However, most roster scheduling models formulate coverage penalties as fixed parameters for convenience.

Work preferences always help in pairing employees with their preferred shifts. Each pair of an employee and a shift has a penalty cost. The larger the cost, the more reluctant the employee is to work on the shift. Besides the work preference for each shift of day, variations of consecutive work patterns may also have effect on their preferences. For example, long consecutive work periods are always assumed to be more attractive because they usually result in long consecutive rest periods.

Among all the defined characteristics, problem dimensions, series constraints generally describe scheduling rules for all the rosters. Contract constraints detail the feasible rosters for each employee. Coverage constraints point out possible combinations of personal rosters. Coverage expenses and work preferences are non-negative parameters that reflect desires of both managers and employees. While most roster scheduling problems can be defined with these six characteristics, extensions can still be made to include other characteristics that are not mentioned here. In the following contents of this paper, roster scheduling problems involving $\mathcal{E}$ employees and $\mathcal{S}$ shifts are discussed over a $\mathcal{D}$ days planning horizon. Related parameters are summarised in Table 1.

3 Assignment Formulation

A general integer programming model for roster scheduling problems can be traced back to Ryan and Foster (1981) in which the model takes the form

$$\min \sum_{e\in\mathcal{E}}\sum_{l\in\mathcal{L}} c_{el}\ x_{el} + \sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}} \left\{ pc^-_{ds}\ y^-_{ds} + pc^+_{ds}\ y^+_{ds} \right\} \tag{1}$$

$$s.t. \sum_{e\in\mathcal{E}} A_{ds\times e}\ x_{el} + y^-_{ds} - y^+_{ds} = D_{ds} \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{2}$$

$$\sum_{l\in\mathcal{L}} x_{el} \leqslant 1 \qquad \forall e \in \mathcal{E} \tag{3}$$

$$x_{el} = \{0,1\} \qquad \forall e \in \mathcal{E}, l \in \mathcal{L} \tag{4}$$

where A is a matrix of 0-1 recording feasible rosters by shift of day in columns, D_{ds} is the required minimum staff level on shift s and day d. The cost c_{el} of assigning roster l to employee e depends on personal work preference. Decision variable x_{el} takes the value of 1 when employee e follows roster l, and vice versa. The variables y^-_{ds} and y^+_{ds} are the numbers of staff below and above the preferred staff level for each shift s on each day d so they are nonnegative integer variables. Succession constraints and contract constraints are already embedded when constructing feasible rosters in columns of A. Coverage constraints (2) align scheduled staff levels to preferred demand. Each employee can work on one roster at most, as shown in constraint (3). Objective function (1)

Table 1 Parameters defined in roster scheduling problems 1

Problem Scale	Employee $e \in \mathcal{E}$ $\mathcal{E} = \{1, \ldots, \hat{e}\}$;	Day $d \in \mathcal{D}$ $\mathcal{D} = \{1, \ldots, \hat{d}\}$;	Shift $s \in \mathcal{S}$ $\mathcal{S} = \{1, \ldots, \hat{s}\}$;
Series Parameters	$\forall$ employee $e \in \mathcal{E}$, shift-stretches starting at shift $s \in \mathcal{S}$ $L^{\text{succ}}_{e,\text{on}} \leqslant$ number of consecutive days worked $\leqslant U^{\text{succ}}_{es}$ $L^{\text{succ}}_{e,\text{off}} \leqslant$ number of consecutive days-off $\leqslant U^{\text{succ}}_{e,\text{off}}$		
Contract Parameters	$\forall$ employee $e \in \mathcal{E}$, day subset $\mathcal{D}_k \subseteq \mathcal{D}$ and shift $s \in \mathcal{S}$ $L^{\text{shift}}_{es} \leqslant$ number of days work in $\mathcal{D}_k$ $\leqslant U^{\text{shift}}_{es}$ $L^{\text{day}}_{ek} \leqslant$ number of days work on shift $s \leqslant U^{\text{day}}_{ek}$		
Coverage Parameters	$\forall$ day $d \in \mathcal{D}$, shift $s \in \mathcal{S}$ D_{ds} : preferred staff level		
Coverage Penalty	$\forall$ day $d \in \mathcal{D}$, shift $s \in \mathcal{S}$ pc^-_{ds} : under staffing penalty pc^+_{ds} : over staffing penalty		
Work Preference	$\forall$ employee $e \in \mathcal{E}$, day $d \in \mathcal{D}$, shift $s \in \mathcal{S}$ $\forall$ shift-stretch of $t \in \mathcal{T} = \left[L^{\text{succ}}_{e,\text{on}}, \max_{s \in \mathcal{S}}\left(U^{\text{succ}}_{es}\right)\right]$ days penalty of assigning shift s to employee e on day d ps^N_{eds} : if employee e prefer not to work on shift s day d ps^Y_{eds} : if employee e prefer to work on shift s day d penalty of assigning shift-stretch from day d to $(d+t)$ pt^N_{edt} : if employee e prefer not to work this shift-stretch pt^Y_{edt} : if employee e prefer to work this shift-stretch		

minimizes total cost of rosters giving a solution with the minimum violation of preferred staff levels and the maximum desirability of working preferences. The total number of feasible rosters depends on the length of scheduling periods and number of different shifts, so it is impossible to enumerate all rosters for large-scale roster scheduling problems.

As suggested in Valouxis and Housos (2000) and Curtois and Qu (2014), an assignment model can be used to solve roster scheduling problems, where x_{eds} and z_{edt} are decision variables defined as follows.

$$x_{eds} = \begin{cases} 1, & \text{if employee } e \text{ work on day } d \text{ shift } s. \\ 0, & \text{otherwise.} \end{cases}$$

$$z_{edt} = \begin{cases} 1, & \text{if employee } e \text{ work } t \text{ consecutive days from day } d. \\ 0, & \text{otherwise.} \end{cases}$$

Same as the previous formulation, y^-_{ds} and y^+_{ds} are non-negative valued integer variables denoting the numbers of staff below and above the preferred staff level for shift s on day d. The objective function models the requirement to minimise under and over staffing when trying to maximise staff working preferences.

$$\begin{aligned}\min &\sum_{e\in\mathcal{E}}\sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\Big\{ps^N_{eds}\ x_{eds} + \ ps^Y_{eds}\ (1 - x_{eds})\Big\} \\ + &\quad \sum_{d\in\mathcal{D}}\sum_{s\in\mathcal{S}}\Big\{\ pc^-_{ds}\ y^-_{ds}\ +\ pc^+_{ds}\ y^+_{ds}\ \Big\} \\ + &\sum_{e\in\mathcal{E}}\sum_{d\in\mathcal{D}}\sum_{t\in\mathcal{T}}\Big\{pt^N_{edt}\ z_{edt}\ +\ pt^Y_{edt}\ (1 - z_{edt})\Big\}\end{aligned} \tag{5}$$

On each day d of the scheduling period, employee e can either work a single shift or have a rest.

$$\sum_{s\in\mathcal{S}} x_{eds} \leqslant 1 \qquad\qquad \forall e\in\mathcal{E}, d\in\mathcal{D} \tag{6}$$

Succession constraints that restrict the feasible domains of shift-stretches and rest-stretches are included in (7)-(11). Assume shifts are labelled in ascending order of their starting times, so shift 1 is the first shift of day. Succession constraints about forward rotation can be formulated as follows,

$$x_{e(d-1)s'} + \sum_{s=1}^{s'-1} x_{eds} \leqslant 1 \qquad\qquad \forall e\in\mathcal{E}, d\in\mathcal{D}-\{1\} \tag{7}$$

The second set of series constraints are about the number of consecutive working days. For shift-stretches starting at shift s, employee e can be assigned to work U^{succ}_{es} shifts at most in (U^{succ}_{es}+1) consecutive days.

$$\sum_{i=0}^{U^{\text{succ}}_{es}}\sum_{j=0}^{\hat{s}-s} x_{e(d+i)(s+j)} \leqslant U^{\text{succ}}_{es} \tag{8}$$

$$\forall e\in\mathcal{E}, d\in\Big\{1,\dots,\hat{d}-U^{\text{succ}}_{es}\Big\}, s\in\mathcal{S}$$

It should be emphasized that long periods of shift-stretches are usually more attractive since they result in more consecutive days-off. For example, when $L^{\text{succ}}_{e,\text{on}}$=3, the sequences 'off-on-off' and 'off-on-on-off' will not exist in rosters, where 'off' denotes a day-off and 'on' is a working day.

$$\Big\{1 - \sum_{s\in\mathcal{S}} x_{eds}\Big\} + \sum_{i=1}^{I-1}\sum_{s\in\mathcal{S}} x_{e(d+i)s} + \Big\{1 - \sum_{s\in\mathcal{S}} x_{e(d+I)s}\Big\} \leqslant I \tag{9}$$

$$\forall e\in\mathcal{E}, I\in\Big\{2,\dots,L^{\text{succ}}_{e,\text{on}}\Big\}, d\in\Big\{1,\dots,\hat{d}-I\Big\}$$

Together with forward rotation constraint (7), constraint (8) and (9) form the feasible set of shift-stretches. Another set of series constraints are about number of consecutive days-off. Constraint (10) models the maximum number of consecutive days-off in rosters.

$$\sum_{i=0}^{U^{\text{succ}}_{e,\text{off}}} \sum_{j=0}^{\hat{s}-s} x_{e(d+i)(s+j)} \leqslant U^{\text{succ}}_{e,\text{off}} \tag{10}$$

$$\forall e \in \mathcal{E}, d \in \left\{1, \ldots, \hat{d} - U^{\text{succ}}_{e,\text{off}}\right\}, s \in \mathcal{S}$$

Similar to constraint (9), when $L^{\text{succ}}_{e,\text{off}}$ =3, the sequences 'on-off-on' and 'on-off-off-on' are not permitted. Succession constraints about the minimum consecutive days-off can be modelled as follows.

$$\sum_{s \in \mathcal{S}} x_{eds} + \sum_{i=1}^{I-1} \sum_{s \in \mathcal{S}} \left\{1 - x_{e(d+i)s}\right\} + \sum_{s \in \mathcal{S}} x_{e(d+I)s} \leqslant I \tag{11}$$

$$\forall e \in \mathcal{E}, I \in \left\{2, \ldots, L^{\text{succ}}_{e,\text{off}}\right\}, d \in \left\{1, \ldots, \hat{d} - I\right\}$$

Contract constraints limit shift assignments over a certain period or in a certain shift type. Day specific and shift specific contract constraints can be modelled in (12) and (13), respectively.

$$L^{\text{day}}_{ek} \leqslant \sum_{d \in \mathcal{D}_k} \sum_{s \in \mathcal{S}} x_{eds} \leqslant U^{\text{day}}_{ek} \qquad \forall e \in \mathcal{E}, \mathcal{D}_k \subseteq \mathcal{D} \tag{12}$$

$$L^{\text{shift}}_{es} \leqslant \sum_{d \in \mathcal{D}} x_{eds} \leqslant U^{\text{shift}}_{es} \qquad \forall e \in \mathcal{E}, s \in \mathcal{S} \tag{13}$$

Similar to (2), coverage constraints that express the relationship between preferred staff levels and scheduled staff levels can be formulated as follows.

$$\sum_{e \in \mathcal{E}} x_{eds} + y^{-}_{ds} - y^{+}_{ds} = D_{ds} \qquad \forall d \in \mathcal{D}, s \in \mathcal{S} \tag{14}$$

One additional set of constraints should be formulated to model the relationship between x_{eds} and z_{edt}. As defined before, employee e work on a shift-stretch for t consecutive days from day d implies that

$$\left\{\sum_{s \in \mathcal{S}} x_{eds} = 0\right\} \text{ and } \left\{\sum_{s \in \mathcal{S}} x_{e(d+i)s} = 1\right\} \text{ and } \left\{\sum_{s \in \mathcal{S}} x_{e(d+t+1)s} = 0\right\} \Longleftrightarrow \left\{z_{edt} = 1\right\}$$

$$\forall e \in \mathcal{E}, d \in \mathcal{D}, i \in \{1, \ldots, t\}, t \in \mathcal{T}.$$

With non-negative parameters pt^N_{edt} and pt^Y_{edt}, the relationship between x_{eds} and z_{edt} can be expressed as follows.

$$\sum_{s \in \mathcal{S}} \left\{ -x_{eds} + \sum_{i=1}^{t} x_{e(d+i)s} - x_{e(d+t+1)s} \right\} \geqslant (t+2) \cdot z_{edt} - 2 \qquad \forall e \in \mathcal{E}, d \in \mathcal{D}, t \in \mathcal{T} \tag{15}$$

$$\sum_{s \in \mathcal{S}} \left\{ -x_{eds} + \sum_{i=1}^{t} x_{e(d+i)s} - x_{e(d+t+1)s} \right\} \leqslant t + z_{edt} - 1 \qquad \forall e \in \mathcal{E}, d \in \mathcal{D}, t \in \mathcal{T} \tag{16}$$

This assignment model can easily accommodate other constraints in roster scheduling problems, but it is proved to be a NP-hard problem. For small scale roster scheduling problems, optimisation solvers such as FICO-Xpress can be used to solve the problem. For large scale roster scheduling problems, it is necessary to find other reformulations which can explore good solutions more efficiently.

4 Network Formulation

Classical integer programming formulations for roster scheduling problems are NP-hard, but when integer constraints are removed, the corresponding linear programming formulations are polynomial solvable. A popular method for solving roster scheduling problems is to formulate problem constraints into an integer flow network structure because network structures have the property of integrity. As long as flows from source to sink are of integer units, the units of flow go through any arc in networks are integer (Bertsimas and Tsitsiklis, 1997). According to the network integer property, network based integer programming problems can be reduced to linear programming problems.

The work of Balakrishnan and Wong (1990) is an early attempt to solve cyclic roster scheduling problems using a network structure. Millar and Kiragu (1998) continued this study and proposed a network model for series constraints in roster scheduling problems. Later in this century, Brucker et al (2011) have sorted several special cases of roster scheduling problems which are either polynomial solvable or NP-complete. Their study suggested that all the polynomial solvable roster scheduling problems can be reformulated into network based integer programming models. A further study conducted by Smet et al (2016) detailed some minimum cost network flow formulations for several roster scheduling problems. It is interesting to note that the work of Millar and Kiragu (1998) mainly focused on series constraint reformulations. but left constraints of contract and coverage as side constraints. Network based reformulations proposed by Smet et al (2016) only incorporate contract constraints, coverage constraints, and some simple series constraints defined in pairwise disjoint day subsets. Their research indicates that it is sufficient to summarise general roster scheduling constraints by these two kinds of network models.

In the following sub-sections, a review of network-based series constraint reformulation will be first given. Our proposed series constraint network will be provided then by introducing layer index for nodes. The existing network-based models about contract constraints and coverage constraints will also be discussed. Finally, an integer programming model will be presented by formulating constraints in the two networks with linking arcs.

4.1 Series Constraint Reformulation

Early research such as Balakrishnan and Wong (1990), Millar and Kiragu (1998) and Xie and Suhl (2015) uses network structures to model series constraints in roster scheduling problems.

For example, in the work of Millar and Kiragu (1998), a network model was proposed for generating roster schedules with 12-hour shifts, Day shift (D) and Night shift (N), over the scheduling period of 14 days. Each node in the network represents a shift-stretch or a rest-stretch. Table 2 lists all the feasible shift-stretches and rest-stretches in this problem. Some nodes are no

Table 2 Feasible shift-stretches and rest-stretches in Millar and Kiragu (1998) 1

4-shift stretches	3-shift stretches	2-shift stretches	1-shift stretches	rest-stretches
D - D - D - D	D - D - D	D - D	D	Rest-Rest-Rest-Rest
D - D - D - N	D - D - N	D - N	N	Rest-Rest-Rest
D - D - N - N	D - N - N	N - N		Rest-Rest
D - N - N - N	N - N - N			Rest

longer feasible at the end of the scheduling period. Otherwise, rosters will be scheduled beyond the scheduling period. Figure 1 presents the network for series constraints of the problem in Millar and Kiragu (1998), where parameter $\hat{d}$ is the last day of the scheduling horizon, as define in Table 1.

Because a roster consists of an alternating sequence of shift-stretch and rest-stretch patterns, an obvious advantage of formulating series constraints in the network of Figure 1 is that the resulting model is essentially a shortest-path problem with side constraints.

Côté et al (2007) introduced layered graph structures for construct integer programming models. Inspired by their work, a t-shift stretch node in Figure 1 can be decomposed into t shift-state nodes. Each shift-state node has two index. One is the shift type. The other one is called the layer which is a indice recording the remaining working days in the shift-stretch. For example, a shift-stretch D-D-D-N can be decomposed in to four shift-state nodes D4, D3, D2 and N1. Rest-stretches can be decomposed into rest-state nodes in a similar way. Table 3 lists all the feasible shift-state and rest-state nodes for the problem in Millar and Kiragu (1998).

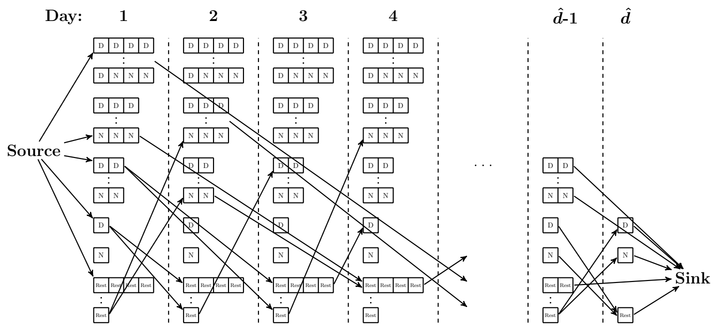

Fig. 1 The network of series constraints for problem Millar and Kiragu (1998)

Table 3 Feasible shift-state and rest-state nodes in Millar and Kiragu (1998) 1

Layer4	Layer3	Layer2	Layer1
D4	D3	D2	D1
N4	N3	N2	N1
Rest4	Rest3	Rest2	Rest1

Since the rest-state node Rest1 denotes the last day in rest-stretches, the following node of Rest1 has to be a shift-state node. The layer of this shift-state node indicates the length of the shift-stretch. Similarly, shift-state nodes with layer 1 are the last working shift in shift-stretches and their following nodes denote the beginning of a rest-stretch. In this problem, the beginning nodes of rest-stretches can be Rest4, Rest3, Rest2 and Rest1 indicating rest-stretches of 4 days, 3 days, 2 days and 1 days, respectively. The arcs, as shown in Figure 2, represents feasible transtions between two days. Similar to Figure 1, the refined

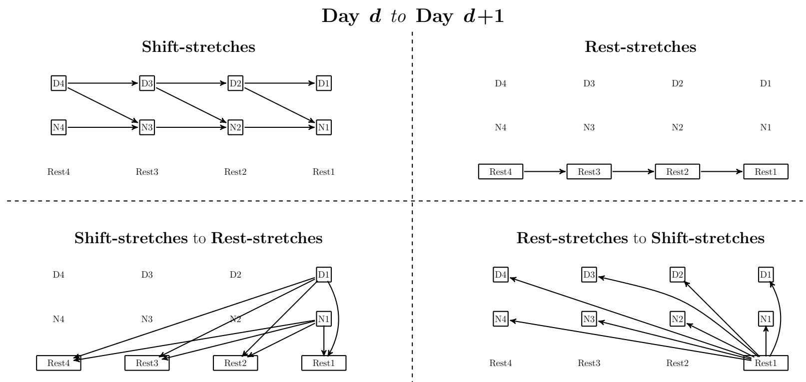

Fig. 2 Arcs for two consecutive days in the refined network model

network has a common source and a common sink. The units of flow that go through this network are equal to the number of staff to be scheduled. Figure 3 presents the refined network for series constraints in problem Millar and Kiragu (1998).

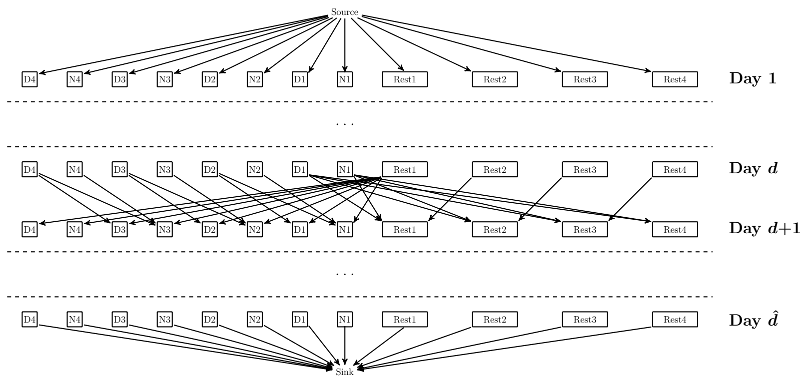

Fig. 3 The refined network of series constraints for problem Millar and Kiragu (1998)

This refined network use 12 state nodes to denote the working status for each employee on each day. Compared to the network structure in Figure 1, the refined network constructs five less nodes per employee per day. For network based integer programming models, all constraints are generated upon the flow conservation property of nodes (Bertsimas and Tsitsiklis, 1997). The fewer the number of nodes in a network model, the easier the problem is to be solved. Also it is easier to construct the remaining constraints as side constraints with the refined network model. In the orginal model, nodes in day d denotes shift-stretches and rest-stretches starting at d. There is no direct access to check whether a certain shift is assigned to a certain employee on a certain day since all the relevant shift-stretches need to be scanned. However, in the refined model, a node contains the information of employee, day and shift type. It facilicates the construction of contract contraints and coverage constraints.

4.2 Contract and Coverage Constraint Reformulation

As summariesed in Section 2, contract constraints are about the number of shifts in the same type or in certain days that can be assigned to employees. They may vary from person to person depending on their contract types. A problem is discussed in Smet et al (2016) with a constraint that generalises the number of days worked constraint (12). They suggest that if these constraints are only defined in pairwise disjoint sets of days, they can be reformulated as a minimum cost network flow problem with the network in Figure 4.

The network consists of three kinds of nodes, employee nodes, subset nodes and task nodes. It is a multi-source multi-sink network, where employee nodes

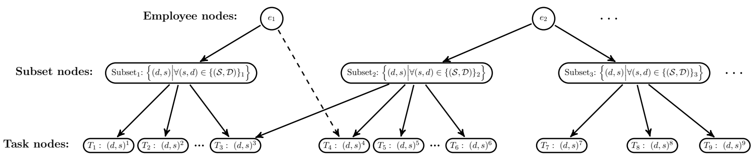

Fig. 4 The network of the number of days constraints in Smet et al (2016)

are the source nodes and the task nodes are the sink nodes. The amount of flow provided by one employee node equals the number of shifts this employee can be assigned to over the scheduling horizon. Define a task (d,s) as a pair of shift s and its day of working d. For all $d \in \mathcal{D}_k$, the flow in the arc connecting a task node (d,s) and a subset node k is a binary variable. Consider a problem that the number of shifts employee $e \in \mathcal{E}$ worked during the day subset $\mathcal{D}_k \subseteq \mathcal{D}$ is in the range of $[L_{ek}^{\text{day}}, U_{ek}^{\text{day}}]$. The flow in the arc which connects employee node e and subset node k has the capacity range. The path from employee node e to subset$_1$ and to subset$_2$ can not exist at the same time since there is an overlapping task T_3 in both the subsets. When shift s on day d can be assigned to employee e, there is a path which connects the employee node e and task node (d,s). The path will either visit one subset node or connect the employee node and the task node directly depending on whether there is a contract constraint involving this task. For example, in Figure 4, there is no contract of employee e_1 related to task T_4 so employee node e_1 is dircetly connected to task node T_4 as shown in the dashed line.

In some problems, a subset can be further divided into several sub-subsets. Valouxis and Housos (2000) described a 3-shift nurse roster scheduling problem over a 28-day planning horizon. The ranges for a nurse working on Early, Late and Night shifts are {5,..,8},{5,..,8} and {2,..,5}. The lower bound and the upper bound for each nurse of the total working days are 15 and 18, respectively. The working day constraints provide slightly more tight bounds than the sum of the contract bounds for each type of shift. Figure 5 illustrates the refined network structure for contract constraints in Valouxis and Housos (2000).

Coverage constraints, as formulated in (14), involves over staffing and under staffing variables. The network model shown in Figure 4 can only deal with the situation that the scheduled staff level equals the preferred staff level. By introducing two additional nodes for each task node, modifications can be made in task node as shown in Figure 6.

Take task (d,s) as an example. As shown in the dashed box in Figure 6, the task node T_3 has one proceding node T_3^- and one following node T_3^+. If the amount of flow arriving at T_3 is less than the preferred staff level, node T_3^- as a back-up source node will provide the difference between the scheduled and the preferred staff level. When task T_3 is over staffed, node T_3^+ will receive the extra. Since both the penalty costs for under-staffing and over-staffing

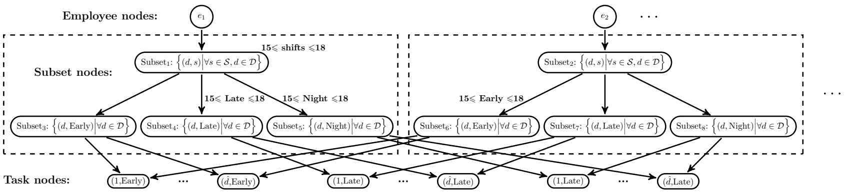

Fig. 5 The refined network of contract constraints for problem Valouxis and Housos (2000)

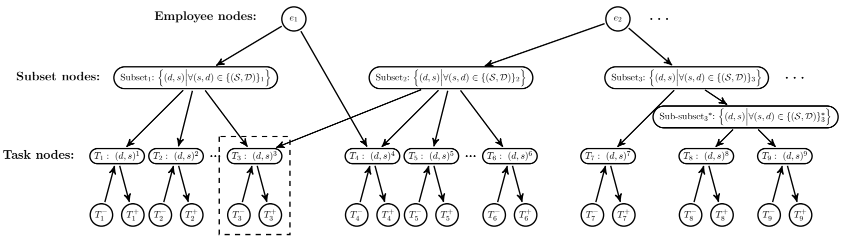

Fig. 6 The refined network of contract and coverage constraints for rostering problems

are non-negative valued, the units of flow in the arc (T_3^-,T_3) and in the arc (T_3,T_3^+) cannot be positive valued at the same time.

4.3 The Refined Network Based Model

Figure 3 and Figure 6 present two networks dealing with series constraints, contract and coverage constraints. Summaries of nodes and arcs for the two networks are given in Table 4. Consider a directed network with the node set $\mathcal{V}$ and the arc set $\mathcal{A}$.

In a minimum cost network flow problem (Bertsimas and Tsitsiklis, 1997), each edge $(i, j) \in \mathcal{A}$ is associated with a cost c_{ij} and a capacity bound u_{ij}. The sum of the supplies generated by source nodes equals the sum of the demands required in sink nodes. There is one decision variable x_{ij} per edge (i, j). Each x_{ij} represents a flow from i to j. The cost of delivering a flow of x_{ij} is $c_{ij}x_{ij}$. Each node $j \in \mathcal{V}$ satisfies a flow constraint

$$\sum_{\{k|(i,k)\in\mathcal{A}\}} x_{ik} - \sum_{\{k|(k,j)\in\mathcal{A}\}} x_{kj} = b_k, \tag{17}$$

where

$$b_k = \begin{cases} -\text{supply}\,, & \text{if node } k \text{ is a source node;} \\ \text{demand}\,, & \text{if node } k \text{ is a sink node;} \\ \quad 0 \quad\;\;, & \text{otherwise.} \end{cases}$$

Table 4 Summary for the refind network models in Figure 3 and Figure 6 1

Network in Figure 3			**Network in Figure 6**		
Source Nodes	**Sink Nodes**	**Supply/Demand**	**Source Nodes**	**Sink Nodes**	**Supply/Demand**
(Source)		$\hat{e}$	(e)		$\geqslant 0$
			(T^-)		$\geqslant 0$
	(Sink)	$\hat{e}$		(T)	D_{es}
				(T^+)	$\geqslant 0$

Arcs for Day *d*-1	to	**Day *d***	**Bound**	**Cost**
$(s,U_{es}^{\text{succ}})_{\text{day } d\text{-1}}^{\text{employee } e}$ … $(s,2)_{d\text{-1}}^e$	$\rightarrow$ … $\rightarrow$	$(s,U_{es}^{\text{succ}}-1)_d^e$ … $(s,1)_d^e$	$\{0,1\}$	$-$
$(\text{s},1)_{d\text{-1}}^e$	$\rightarrow$	$(\text{Rest},U_{\text{e,off}}^{\text{succ}})_d^e$ … $(\text{Rest},L_{\text{e,off}}^{\text{succ}})_d^e$	$\{0,1\}$	$-$
$(\text{Rest},U_{\text{e,off}}^{\text{succ}})_{d\text{-1}}^e$ … $(\text{Rest},2)_{d\text{-1}}^e$	$\rightarrow$ … $\rightarrow$	$(\text{Rest},U_{\text{e,off}}^{\text{succ}}-1)_d^e$ … $(\text{Rest},1)_d^e$	$\{0,1\}$	$-$
$(\text{Rest},1)_{d\text{-1}}^e$	$\rightarrow$	$(s,U_{es}^{\text{succ}})_d^e$ … $(s,t)_d^e$ … $(s,L_{es}^{\text{succ}})_d^e$	$\{0,1\}$	pt_{edt}^N or pt_{edt}^Y

Arcs			**Bound**	**Cost**
(e)	$\rightarrow$	$\{(d,s)\vert e\}$	$\{L_{es}^{\text{shift}},U_{es}^{\text{shift}}\}$ or $\{L_{ek}^{\text{day}},U_{ek}^{\text{day}}\}$	$-$
$\{(d,s)\vert e\}$	$\rightarrow$	$(T:d,s)$	$\{0,1\}$	ps_{eds}^N or ps_{eds}^Y
$(T^-:d,s)$	$\rightarrow$	$(T:d,s)$	$\geqslant 0$	pc_{ds}^-
$(T:d,s)$	$\rightarrow$	$(T^+:d,s)$	$\geqslant 0$	pc_{ds}^+

A pair of arc sets link both the networks. They are:

$$\begin{aligned} & \big\{(d,s)\big|e\big\} \rightarrow (T:d,s) && \text{in Figure 3} \\ \text{and} \\ & (s,t')_{d\text{-}1}^e \rightarrow (s,t'\text{-}1)_d^e \ \ , \forall t' \in \Big[\ \ 2\ , U_{es}^{\text{succ}}\Big] \\ &&& \text{in Figure 6} \\ & (r,1)_{d\text{-}1}^e \rightarrow (s,t'')_d^e \quad , \forall t'' \in \Big[L_{es}^{\text{succ}}, U_{es}^{\text{succ}}\Big] \\ & \qquad\qquad\qquad \forall \text{ employee } e \in \mathcal{E}, \text{day } d \in \mathcal{D}, \text{shift } s \in \mathcal{S}. \end{aligned} \tag{18}$$

Denote the linking arc set of networks in Figure 3 as $\mathcal{A}_{1,eds}^{\text{LINK}}$ for employee e, day d and shift s. The arc set $\mathcal{A}_{2,eds}^{\text{LINK}}$ stores the linking arcs in Figure 6. An additional set of constraints (19) that link these two networks can be formulated as follows,

$$\sum_{(i,j)\in\mathcal{A}_{1,eds}^{\text{LINK}}} x_{ij} = \sum_{(i,j)\in\mathcal{A}_{2,eds}^{\text{LINK}}} x_{ij}\ , \qquad \forall e \in \mathcal{E}, d \in \mathcal{D}, s \in \mathcal{S}. \tag{19}$$

With the linking constraints (19), the constraints (17) for both the networks and the sum of both objective functions, a network based integer programming model can be constructed to solve roster scheduling problems.

5 Computational Experiment

The strength of the refined model is tested using the new instance data published by The University of Nottingham (2014). These instances are designed to be realistic and challenging but straightforward to use. All core constraints found commonly in staff rostering problems are included in each instance problem of the dataset. As stated in Smet et al (2016), network based models can only accommodate the contract constraints whose domains are defined in disjoint task subsets. Contract constraints about the numbers of all working shifts and the shifts of different types are embedded into the network formulation, but the constraints about the number of weekends for each employee are ignored. Since the contract constraints defined in The University of Nottingham (2014) are in working hours but in the number of working shifts, instances that involve different lengths of shifts are excluded in this experiment. A general summary about the eligible instances and the computational experiment results of applying FICO-Xpress 7.9 are listed in Table 5.

Table 5 Computational results for instances in The University of Nottingham (2014) 1

Instance	Weeks	Staff	Shifts	M_1: Assignment			M_2: Network in Figure 3 with side constraints			M_3: Network in Figure 6 with side constraints			M_4: Refined Network		
				solution	**time**	**gap**	**solution**	**time**	**gap**	**solution**	**time**	**gap**	**solution**	**time**	**gap**
Instance 1	2	8	1	14	<1s	0.00%	14	<1s	0.00%	14	<1s	0.00%	14	<1s	0.00%
Instance 2	2	14	2	216	<1s	0.00%	216	<1s	0.00%	216	<1s	0.00%	216	<1s	0.00%
Instance 3	2	20	3	2	3600s	50.00%	2	5s	0.00%	2	470s	0.00%	2	9s	0.00%
Instance 4	4	10	2	1113	18s	0.00%	1113	1s	0.00%	1113	15s	0.00%	1113	1s	0.00%
Instance 5	4	16	2	166	3600s	82.44%	131	11s	0.00%	143	3600s	76.36%	131	1s	0.00%
Instance 6	4	18	3	668	3600s	50.34%	351	49s	0.00%	666	3600s	49.90%	351	49s	0.00%
Instance 7	4	20	3	147	3600s	74.48%	46	35s	0.00%	87	3600s	56.27%	46	28s	0.00%
Instance 8	4	30	4	1254	3600s	1.36%	1243	833s	0.00%	1254	3600s	1.36%	1243	582s	0.00%
Instance 11	4	50	6	1	12s	0.00%	1	1301s	0.00%	1.00	29s	0.00%	1	639s	0.00%
Instance 12	4	60	10	1637	3600s	56.20%	1531	3600s	46.44%	1531.00	3600s	46.44%	1020	3600s	19.61%
Instance 16	8	20	3	161	3600s	69.96%	155	27s	0.00%	156.00	3600s	0.64%	155	25s	0.00%
Instance 17	8	32	4	349	3600s	88.44%	44	1220s	0.00%	349.00	3600s	88.44%	44	369s	0.00%
Instance 18	12	22	3	-	3600s	-	1088	73s	0.00%	-	3600s	-	1088	34s	0.00%
Instance 20	26	50	6	-	3600s	-	-	3600s	-	-	3600s	-	-	3600s	-
Instance 22	52	50	10	-	3600s	-	-	3600s	-	-	3600s	-	-	3600s	-

Table 5 shows that the network based formulation usually have an advantage over the assignment formulation, but it is not the case for instance 11 where the assignment model performs best. A reasonable assumption could be made that the assignment model is more efficient when the optimal solution has less soft constraint violations. Results obtained from the model M_1 and model M_3 are quite close. It indicates that the improvement of applying contract and coverage network structure to construct the integer programming model is limited. The models that embedded series network structures, M_2 and M_4, perform significantly better than the models without the series network structure. It confirms the assumption that series constraints are the most difficult constraints to be solved. No feasible integer solutions can be found for instance 20 and instance 22 with the maximum running time of 1 hour. Since roster scheduling problems have been proved to be NP-hard (Smet et al, 2016), better results can be expected by applying heuristic algorithms to search for good feasible solutions instead of using exact algorithms to search for the optimal solution when solving large scale roster scheduling problems with limited time.

6 Conclusion and Future work

In this paper, a refined network based formulation is proposed to incorporate various constraints in roster scheduling problems. Two underlying networks in the refined network, series network (Figure 3) and contract and coverage network (Figure 6), are described to accommodate different sets of constraints in roster scheduling problems. With the linking arcs as stated in (18), two independent networks can be connected. The computational experiments demonstrated the effectiveness of the refined formulation on roster scheduling data from literature. Compared to the assignment integer programming formulation and the network based formulations with side constraints, the refined formulation not only gives more insight into the structure of roster scheduling problems, but also performs better in solving scheduling problems of small to medium scales. Other than using common optimisation software, problem specific heuristic algorithms for solving large scale problems are the focus of our future research.

References

Balakrishnan N, Wong RT (1990) A network model for the rotating workforce scheduling problem. Networks 20(1):25–42

Bertsimas D, Tsitsiklis JN (1997) Introduction to linear optimization, vol 6. Athena Scientific Belmont, MA

Brucker P, Qu R, Burke E (2011) Personnel scheduling: Models and complexity. European Journal of Operational Research 210(3):467–473

Brunner JO, Bard JF, Köhler JM (2013) Bounded flexibility in days-on and days-off scheduling. Naval Research Logistics (NRL) 60(8):678–701

Côté MC, Gendron B, Rousseau LM (2007) Modeling the regular constraint with integer programming. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, Springer, pp 29–43

Curtois T, Qu R (2014) Computational results on new staff scheduling benchmark instances

Ernst AT, Jiang H, Krishnamoorthy M, Sier D (2004) Staff scheduling and rostering: A review of applications, methods and models. European journal of operational research 153(1):3–27

Lau HC (1996) On the complexity of manpower shift scheduling. Computers & Operations Research 23(1):93–102

Millar HH, Kiragu M (1998) Cyclic and non-cyclic scheduling of 12h shift nurses by network programming. European journal of operational research 104(3):582–592

Osogami T, Imai H (2000) Classification of various neighborhood operations for the nurse scheduling problem. In: Algorithms and computation, Springer, pp 72–83

Ryan DM, Foster BA (1981) An integer programming approach to scheduling. Computer scheduling of public transport urban passenger vehicle and crew scheduling pp 269–280

Smet P, Brucker P, De Causmaecker P, Berghe GV (2016) Polynomially solvable personnel rostering problems. European Journal of Operational Research 249(1):67–75

The University of Nottingham (2014) Employee shift scheduling benchmark data: new instances. URL http://www.cs.nott.ac.uk/ psztc/NRP/, last access: 2016-06-24

Valouxis C, Housos E (2000) Hybrid optimization techniques for the workshift and rest assignment of nursing personnel. Artificial Intelligence in Medicine 20(2):155–175

Xie L, Suhl L (2015) Cyclic and non-cyclic crew rostering problems in public bus transit. OR Spectrum 37(1):99–136