IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 5, 2019, accepted January 31, 2019, date of publication February 19, 2019, date of current version March 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900028

Integrated Crew Scheduling and Roster Problem for Trainmasters of Passenger Railway Transportation

DUNG-YING LIN[1] AND MENG-RUNG TSAI[2]

[1]Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan
[2]Department of Transportation and Communication Management Science, National Cheng Kung University, Tainan 70101, Taiwan

Corresponding author: Dung-Ying Lin (dylin@ie.nthu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 105-2628-H-006-003-MY3.

ABSTRACT Train crew management is an imperative task in a passenger railway system and is typically decomposed into two sub-problems: crew scheduling problem and crew rostering problem. The decomposition can make the problem easier to solve but may produce degraded solutions. In this paper, we propose a formulation to integrate these two critical sub-problems and develop a branch-and-price-and-cut algorithm and a depth-first search-based algorithm to solve the composite problem. The numerical results show that an integrated framework can yield better solutions than the decomposition strategy. Furthermore, results also show that the rostering constraints have a more notable effect on the results compared with the scheduling constraints in the integrated framework. This type of observation can only be accurately characterized when these two sub-problems are considered in an integrated manner.

INDEX TERMS Crew scheduling, crew rostering, staff management, branch-and-price-and-cut, rail transport.

I. INTRODUCTION

In a passenger railway transportation system, crew management is an imperative task. In addition to the train driver, the trainmaster, who is responsible for handling passengers' requests, complaints and ticketing on a train, is also important. However, the scheduling and rostering of trainmasters is an extremely challenging job. Many railways systems rely heavily on manual work and rule of thumb, which leaves considerable room for improvement. Even with the assistance of a decision support system, train crew management is typically decomposed into crew scheduling and crew rostering sub-problems, which are solved sequentially due to the significant computational resources that are needed to determine a feasible schedule and roster. As might be expected, the decomposition can result in inferior solutions compared to the solutions obtained from the integrated framework. To address this discrepancy, the current research proposes an integrated crew scheduling and rostering formulation. To effectively solve the resulting complicated program, we develop a depth-first search based (DFS-based) and a branch-and-price-and-cut (BPC) algorithms. The proposed algorithms are empirically applied to real-world cases of the trainmaster planning cases. Empirical results show that the integrated framework produces a superior solution compared to solving scheduling and rostering sub-problems sequentially. In addition, the proposed algorithms outperform a commonly used commercial optimization package and determine the solutions that are comparable to the schedule and roster that are currently used in practice. Finally, the rostering rules have more impact than the scheduling rules in the planning process, which is an observation that can only be made within an integrated framework.

The remainder of the paper is structured as follows. Section II provides a critical overview of recent developments in the field of crew management and related fields of research. Section III presents a mathematical formulation of the integrated scheduling and rostering problem, which will be used in a later section to develop the solution algorithms. Section IV describes the proposed solution procedure, BPC, to find the solution to the mathematical model described in Section III. In Section V, we develop a DFS-based solution

The associate editor coordinating the review of this manuscript and approving it for publication was Bin Yu.

approach that uses a two-stage solution strategy such that its solution can be compared to the integrated solution. In Section VI, the proposed method is applied empirically to problems of various sizes to demonstrate its efficacy and efficiency. The final section concludes the paper and suggests potential directions for future research.

II. LITERATURE REVIEW

In the literature, train crew management is typically decomposed into two sub-problems: crew scheduling problem and crew rostering problem [12]. Train crew scheduling is the creation of duties such that the tasks within a timetable can be covered [9]. Train crew rostering is constructing a roster based on the duties generated in crew scheduling [9]. The duties generated in the first stage (crew scheduling) is sequenced in the second stage (crew rostering) to form a feasible roster so that staff members can participate in a rotating manner. Ernst *et al.* [10] reviewed the staff scheduling and rostering problem and identified various applications (e.g., transportation, emergency services, hospitality and tourism) and solution approaches (i.e., artificial intelligence, constraint programming, metaheuristics and mathematical programming).

In passenger railway transportation, crew scheduling has attracted the attention of numerous researchers. Chu and Chan [8] investigated crew scheduling for the Hong Kong Light Rail Transit and proposed a network-based heuristic approach to iteratively construct the schedule. Liu *et al.* [19] presented a genetic algorithm-based column generation approach for passenger rail crew scheduling. The problem was formulated as a set partitioning problem, and the genetic algorithm was used to generate the columns that can improve the objective restricted master program. Jutte *et al.* [15] investigated the crew scheduling problem associated with a freight railway crew and proposed a column generation solution. Kwan [16] discussed the key successful factors of the crew scheduling system in the UK and discussed several cases to highlight the factors. Hoffmann *et al.* [13] incorporated the consideration of attendance rates in a railway crew scheduling problem and formulated it as a set covering problem. The resulting problem is solved by a genetic algorithm-based column generation approach. These studies provide interesting and valuable insight when a practical crew scheduling system is to be introduced to a railway company.

Similarly, crew rostering has also been the focus of several studies. Bianco *et al.* [4] proposed an integer programming formulation for the crew rostering problem in a mass transit system and devised an iterative heuristic approach to solve it. Caprara *et al.* [7] studied the crew rostering problem for railway applications and developed an approximation solution method based on the Lagrangian lower bound. Lezaun *et al.* [17] studied the rostering problem for a passenger railway system, emphasizing the equality issue of the rostering. Practical issues (such as preferences of employees, similar numbers of morning, evening and night shifts, and working days/hours per year) are considered, and the resulting binary programming problem is solved using a commercial software package. Nishi *et al.* [20] presented a two-level decomposition approach to solve the railway crew rostering problem with the objective of fair working conditions. A branch-and-bound algorithm with additional valid cuts was developed to reduce the feasible search space and to tighten the duality gap.

With the gradual maturation of scheduling and rostering techniques, studies in the literature have begun to incorporate more practical issues when considering scheduling and rostering. For instance, crew re-scheduling after disruptions and integrated timetabling and scheduling are popular research topics. Huisman [14] investigated the crew re-scheduling problem that occurs when the original timetable is modified due to disruptions. The problem is formulated as a set covering problem and is solved by a column generation based algorithm. Rezanov and Ryan [22] studied the train driver recovery problem (TDRP) and defined the disruption neighborhood by identifying the set of drivers affected by a disruption. The resulting TDRP was formulated as a set partitioning problem that can be solved by a depth-first search constraint branching strategy in a branch-and-bound scheme. Potthoff *et al.* [21] presented a column generation-based algorithm for crew rescheduling for when disruptions occur. Similarly, Veelenturf *et al.* [25] discussed how a railway operator can adjust the timetable and crew scheduling when disruptions occur. A column generation-based algorithm combined with Lagrangian heuristics was proposed to solve the problem. Bach *et al.* [1] investigated the integration of train timetabling and the crew scheduling problem. The output of train timetabling is used to generate the crew schedule, while the results of the crew schedule are employed to adjust the timetable. The resulting model is solved using a column generation approach.

Although several studies investigate the scheduling and rostering of railway transportation within one study, the integration is not necessarily complete. For instance, [6] outlined various approaches to model railway crew scheduling and rostering problems and illustrated solution approaches for the problems. However, the two sub-problems are solved independently and are not integrated to date. In many studies, the railway crew scheduling and rostering problems are solved sequentially, which can result in solutions that are not as good as solutions yielded from an integrated framework. For instance, [5] provided the experience of developing the crew planning system in which the crew planning problem is similarly decomposed into crew scheduling (specifically, pairing generation and pairing selection) and crew rostering sub-problems. The sub-problems are solved sequentially. Ernst *et al.* [9] proposed an integrated optimization model to solve both the crew scheduling and crew rostering problems. To solve the proposed model for a realistic scale, they relaxed the integer constraint of variables and rounded up the fractional solutions to nearest integer. Freling *et al.* [11] discussed practical issues when implementing a decision support system for crew planning. The solution was a branch-and-price algorithm in which crew scheduling and the crew roster are determined in a sequential manner.

Several observations can be summarized after reviewing relevant studies in the literature. First, the column generation approach is one of the widely used mainstream techniques due to the problem features associated with crew scheduling and rostering. Second, as the crew scheduling and crew rostering each pose significant computational challenges in railway transportation, only a limited number of studies have integrated the two stages into one unified model. Even with the intention of integration, the proposed framework still solve the two sub-problems sequentially and is not necessarily ideal. Meanwhile, recent studies in airline crew management also investigated in eliminating the discrepancy, which demonstrates the importance of this integration. For instance, [24] presented a genetic algorithm (GA)-based approach and showed that the solution obtained by integrated approach is better. However, similar to relevant GA studies, the solution quality of the proposed solution method cannot be guaranteed. Saddoune *et al.* [23] proposed a dynamic constraint aggregation method to solve the integrated in airline crew management problem and showed that integrating crew scheduling and rostering can yield significant savings. Therefore, the current research proposes an integrated railway crew scheduling and rostering formulation and develops efficient solution algorithms to address the resulting mathematical program.

III. MATHEMATICAL FORMULATION

In this section, we present the formulation for the integrated crew scheduling and rostering program. We first formally state the problem in Section III-A followed by introducing the scheduling/rostering rules that must be considered. In Section III-C, we present the mathematical formulation.

A. PROBLEM STATEMENT

Let us define the following terms before defining the problem.

1. *Home-base station*: Each trainmaster has her/his home-base station where she/he resides in nearby areas. When preparing the schedule/roster, the planner must allow trainmasters to return to the home-base station every one or two days.
2. *Overnight station*: The stations that are equipped with amenities for the trainmaster to stay overnight when she/he cannot return to the home-base station. When a crew member stays overnight at a station other than her/his home-base depot, this is called barracking [9].
3. *Dummy station*: The virtual station that is designed to make a schedule/roster feasible.
4. *Task*: For a train timetable, each train journey can be divided into a sequence of tasks. Each task is characterized by its start time, end time, start station and end station and will be performed by one trainmaster.
5. *Duty*: A duty can contain one or more tasks and is the job that should be completed by a trainmaster within one day.
6. *Roster*: A sequence of duties performed by a set of trainmasters in a rotating manner over a specified planning period.
7. *Feasible solution*: Let us use Figure 1 to explain the solution design that is used in our mathematical formulation. For each feasible solution, we consider i weeks and j days. For each day, there can be P positions and each position can be assigned a task, the number of P is based on the maximum number of task that a trainmaster can be assigned in one day. Each j corresponds to a duty that is to be performed by a trainmaster. The working days in one week is defined as W. Further, there can be M weeks in the planning period and the sequence of duties forms a roster.

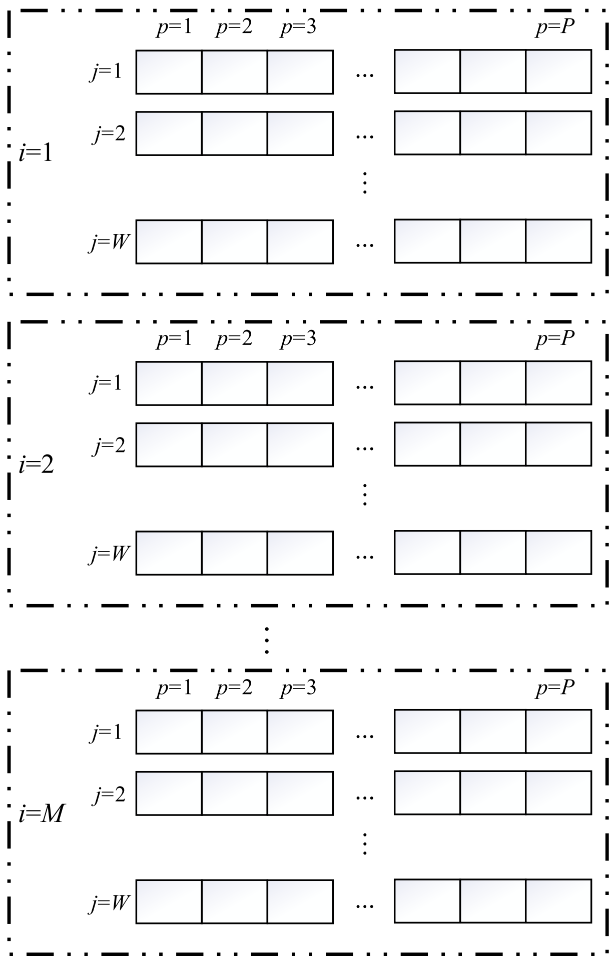

FIGURE 1. Solution design.

8. *Basic task:* There are three basic types of tasks considered in this research: (1) Working task (set of T^w): the task that must be carried out by a trainmaster. (2) Lift task (set of T^r): the ride that a trainmaster takes from station A to station B to perform the task based on station B. During the ride, the trainmaster does not perform any tasks to service the customer. (3) Standby task (set of T^{SD}): the task assigned to the trainmaster to

stand by at a station in case of emergencies. The start and end station of this type of task is the home-base station.

9. *Dummy task:* Three types of dummy tasks are included to ensure the feasibility of the solution structure depicted in Figure 1. In the solution structure, every position should be assigned a task to examine the feasibility of connecting the earlier task to the following task. If less than P positions are assigned, the remainder of the positions should be assigned dummy tasks. The three dummy tasks are the following: (1) Tasks in set T^{mn}: the start station of this type of task is a home-base station and the end station is a dummy station. (2) Tasks in set T^{on}: the start station of this type of task is an overnight station and the end station is a dummy station. If a task t_1 is connected to a task that belongs to these sets ($t_2 \in T^{mn} T^{on}$), then task t_1 is the last task of the corresponding duty, and the end station of the duty can be determined by the start station of task t_2. (3) Tasks in set T^{nn}: The start and end stations of this type of task are dummy stations.
10. *Two-day duty*: A typical duty is the job performed by a trainmaster for one day. However, for the railway company we investigate, it has a special duty that spans two days. Essentially, a trainmaster that executes this type of duty must stay overnight at a station that is not her/his home-base station. The scheduling and rostering rules of this type of duty are different from those of a typical one-day duty. These rules will be elaborated more in Section III-B

Based on the definitions, the integrated crew scheduling and rostering problem for the trainmaster for a passenger railway system can be stated as follows: to generate the duties required to cover a timetable and to sequence the duties to form a feasible roster that requires the minimum number of trainmasters to execute. The duty and roster that is formed should comply with the following labor or contractual rules.

B. LABOR/CONTRACTUAL RULES FOR SCHEDULING AND ROSTERING

The scheduling rules for trainmasters are the following:

1. All tasks should be assigned. In other words, no train service can run without a trainmaster.
2. Only when the end station of a task is identical to the start station of the following task can these two tasks be performed consecutively by the same trainmaster. Otherwise, a lift task is needed to connect the tasks.
3. Only when the end time of the previous task and the start time of the following task is greater than 5 minutes can these two consecutive tasks be performed by the same trainmaster. However, if the two consecutive tasks are for the same train service, the two tasks can be performed by the same trainmaster without the intervening 5 minutes.
4. The working hour per day cannot exceed the upper bound regulated by labor and contractual laws.
5. A duty cannot contain two consecutive lift tasks.
6. A duty cannot contain only lift tasks.

The following rules should be followed when building the roster for the trainmaster:

1. The duty in the first day of the week should begin from the home-base station whereas the duty in the last day of the week should end at the same home-base station.
2. The end station of the duty from the previous day should be identical to the first station of the duty in the current day.
3. A trainmaster should have at least 12 hours of rest time between the duty from the previous day and the duty of the current day. However, a two-day duty is considered as one duty and therefore is not constrained by this rule.
4. A trainmaster must return to her/his home-base station every two days. If a two-day duty is generated, the end station of the first day duty should be an overnight station, which has space for rest.
5. The weekly working hour cannot exceed the upper and lower bounds regulated by labor and contractual laws.

C. MATHEMATICAL FORMULATION OF INTEGRATED SCHEDULING AND ROSTERING PROBLEM

We next introduce the notation that will be used throughout the paper followed by a detailed description of the model.

Parameters

Max	An arbitrary large number.
M	The maximum number of weeks needed to complete the roster.
D	Number of working days in a week. D is typically 6.
P	The maximum number of tasks in one duty.
N	The number of tasks.
SD	The number of standby duties.
SS	The number of overnight stations.
RT^{bw}	The preparation time required before a working task.
RT^{br}	The preparation time required before a lift task.
WH^{max}_{day}	The maximum working hours for a one-day duty.
WH^{max}_{2days}	The maximum working hours for a two-day duty.
WH^{min}_{week}	The minimum weekly working hours. This value is set in the railway company's contractual rule and is 40 hours. This minimum is usually set for the sake of salary or fairness.
WH^{max}_{week}	The maximum working hours every week. This value is regulated by labor laws and is 44 hours.
R	The rest hours needed between two consecutive duties. It is typically 12 hours according to local labor laws. Note that a two-day duty does not require R.

Day The number of hours in one day (i.e., 24 hours or 1,440 minutes).

WT_t^s The clock-in time of task t. It is the time for a trainmaster start the preparation before the assigned task.

WT_t^e The clock-out time of task t. It is the time when a trainmaster finish the work after task.

S_t^s The start station of task t. Each station is assigned an arbitrary unique number. S_t^s will be assigned a value that is equal to its start station's given number. For instance, if a station A is assigned number 1, $S_t^s \leftarrow 1$ for the tasks start from station A.

S_t^e The end station of task t. Similar to the example above, $S_t^e \leftarrow 1$ for the tasks end at station A.

$WH_{t_1t_2}$ The time required for a trainmaster to finalize task t_1 (i.e., clean the train, etc.) and prepare for task t_2 (i.e., check in or examine relevant equipment). In other words, $WH_{t_1t_2}$ is the minimum time needed to connect tasks t_1 and t_2.

$\tau^{i,j}$ The penalty to prevent assigning tasks to later weeks in the objective. $\tau^{i,j}$ is discussed with the objective function.

Set

T The set of tasks, including working, lifting and standby tasks.

T^w The set of working tasks, $T^w \subseteq T$.

T^r The set of lift tasks, $T^r \subseteq T$.

T^{SD} The set of standby tasks, $T^{SD} \subseteq T$.

T^m The set of tasks that start from the home-base station, $T^m \subseteq T$.

T^o The set of tasks that start from overnight stations, $T^o \subseteq T$.

T^{mn} The set of dummy tasks that start from the home-base station and end at a dummy station, $T^{mn} \subseteq T$.

T^{on} The set of dummy tasks that start from an overnight station and end at a dummy station, $T^{on} \subseteq T$.

T^{nn} The set of dummy tasks that start and end at dummy stations, $T^{nn} \subseteq T$.

T^{t_1} The set of tasks that task t_1 can connect to, $T^{t_1} \subseteq T$.

Variables

$x_t^{i,j,p}$ If task t is assigned to the p^{th} position of day j in week i, then $x_t^{i,j,p} = 1$. Otherwise, $x_t^{i,j,p} = 0$.

$y_{t_1,t_2}^{i,j,p}$ If the p^{th} position is task t_1 and the $(p+1)^{th}$ position is task t_2, then $y_{t_1,t_2}^{i,j,p} = 1$. Otherwise, $y_{t_1,t_2}^{i,j,p} = 0$.

$z_{mm}^{i,j}$ If the duty in day j in week i starts and ends at the home-base station, $z_{mm}^{i,j} = 1$. Otherwise, $z_{mm}^{i,j} =$.

$z_{mo}^{i,j}$ Intermediate decision variable. If $z_{mo}^{i,j} = 1$, the duty in day j in week i starts at the home-base station and ends at an overnight station. $z_{mo}^{i,j} =$ otherwise.

$z_{om}^{i,j}$ Intermediate decision variable. If the duty in day j in week i starts at an overnight station and ends at the home-base station, $z_{om}^{i,j} = 1$. Otherwise, $z_{om}^{i,j} =$.

$\gamma^{i,j}$ Intermediate decision variable. If $\gamma^{i,j} = 1$, the duties of i and j together are considered as a two-day duty. Otherwise, $\gamma^{i,j} = 0$.

$$\text{Minimize} \sum_{i=1}^{M}\sum_{j=1}^{D}\sum_{t\in T^w,T^r} \tau^{i,j} \times x_t^{i,j,1} \quad (1)$$

$$\text{Subject to} \sum_t x_t^{i,j,p} = 1 \quad \forall i,\ \forall j,\ \forall p \quad (2)$$

$$\sum_i\sum_j\sum_p x_t^{i,j,p} = 1 \quad \forall t \in T^w \quad \forall i, \forall j, \quad (3)$$

$$x_{t_1}^{i,j,p} + x_{t_2}^{i,j,p+1} - 2y_{t_1,t_2}^{i,j,p} \geq 0 \quad \forall p = 1\ldots P-1 \quad \forall t_1 \in T \quad \forall t_2 \in T \quad \forall i, \forall j, \quad (4)$$

$$x_{t_1}^{i,j,p} + x_{t_2}^{i,j,p+1} - y_{t_1,t_2}^{i,j,p} \leq 1 \quad \forall p = 1\ldots P-1 \quad \forall t_1 \in T, \quad \forall t_2 \in T \quad \forall i, \forall j, \quad (5)$$

$$y_{t_1,t_2}^{i,j,p} = 0 \quad \forall p = 1\ldots P-1, \quad \forall t_1 \in T, \quad \forall t_2 \notin T^{t_1} \quad (6)$$

$$\sum_{t\in T^m} x_t^{i,j,1} + \sum_{p=2..P}\sum_{t\in T^{mn}} x_t^{i,j,p} \geq 2z_{mm}^{i,j} \quad \forall i, \forall j, \quad (7)$$

$$\sum_{t\in T^m} x_t^{i,j,1} + \sum_{p=2..P}\sum_{t\in T^{on}} x_t^{i,j,p} \geq 2z_{mo}^{i,j} \quad \forall i, \forall j, \quad (8)$$

$$\sum_{t\in T^o} x_t^{i,j,1} + \sum_{p=2..P}\sum_{t\in T^{mn}} x_t^{i,j,p} \geq 2z_{om}^{i,j} \quad \forall i, \forall j, \quad (9)$$

$$z_{mm}^{i,j} + z_{mo}^{i,j} + z_{om}^{i,j} = 1 \quad \forall i, \forall j \quad (10)$$

$$z_{mm}^{i,1} + z_{mo}^{i,1} = 1 \quad \forall i \quad (11)$$

$$z_{mm}^{i,D} + z_{om}^{i,D} = 1 \quad \forall i \quad (12)$$

$$\sum_{p=1..P-1}\sum_{t_1}\sum_{t_2\in T^{mn},T^{on}} S_{t_1}^e \times y_{t_1,t_2}^{i,j,p} = \sum_t S_t^s \times x_t^{i,j+1,1} \quad \forall i, j = 1\ldots D-1 \quad (13)$$

$$\sum_{t\in T^m} WT_t^s \times x_t^{i,j+1,1} + Day - \sum_{p=1..P-1}\sum_{t_1}\sum_{t_2\in T^{mn}} WT_{t_1}^e \times y_{t_1,t_2}^{i,j,p} \geq R \quad \forall i, \quad j = 1\ldots D-1 \quad (14)$$

$$\sum_{t\in T^w} RT^{bw}\times x_t^{i,j,1}+\sum_{t\in T^r} RT^{br}\times x_t^{i,j,1} +\sum_p\sum_{t_1}\sum_{t_2\in T^{t_1}} WH_{t_1t_2}\times y_{t_1,t_2}^{i,j,p}\le WH_{day}^{max} \quad \forall i,\ \forall j \tag{15}$$

$$\sum_j\left(\sum_{t\in T^w} RT^{bw}\times x_t^{i,j,1}+\sum_{t\in T^r} RT^{br}\times x_t^{i,j,1} +\sum_p\sum_{t_1}\sum_{t_2\in T^{t_1}} WH_{t_1t_2}\times y_{t_1,t_2}^{i,j,p}\right)\ge WH_{week}^{min} \quad \forall i \tag{16}$$

$$\sum_j\left(\sum_{t\in T^w} RT^{bw}\times x_t^{i,j,1}+\sum_{t\in T^r} RT^{br}\times x_t^{i,j,1} +\sum_p\sum_{t_1}\sum_{t_2\in T^{t_1}} WH_{t_1t_2}\times y_{t_1,t_2}^{i,j,p}\right)\le WH_{week}^{max} \quad \forall i \tag{17}$$

$$\left(\sum_{t\in T^w} RT^{bw}\times x_t^{i,j,1}+\sum_{t\in T^r} RT^{br}\times x_t^{i,j,1} +\sum_p\sum_{t_1}\sum_{t_2\in T^{t_1}} WH_{t_1t_2}\times y_{t_1,t_2}^{i,j,p}\right) +\left(\sum_{t\in T^w} RT^{bw}\times x_t^{i,j+1,1}+\sum_{t\in T^r} RT^{br}\times x_t^{i,j+1,1} +\sum_p\sum_{t_1}\sum_{t_2\in T^{t_1}} WH_{t_1t_2}\times y_{t_1,t_2}^{i,j+1,p}\right)\le WH_{2days}^{max} +\left(1-\gamma^{i,j}\right)\times Max \quad \forall i,\ j=1\ldots D-1 \tag{18}$$

$$z_{mo}^{i,j}+z_{om}^{i,j+1}=2\gamma^{i,j} \quad \forall i,\ j=1\ldots D-1 \tag{19}$$

$$x_t^{i,j,p}\in\{0,1\} \quad \forall i, \forall j, \forall p, \forall t \tag{20}$$

$$y_{t_1,t_2}^{i,j,p}\in\{0,1\} \quad \forall i, \forall j,\ \forall p=1\ldots P-1,\ \forall t_1\in T,\ \forall t_2\in T \tag{21}$$

$$z_{mm}^{i,j}\in\{0,1\} \quad \forall i, \forall j \tag{22}$$

$$z_{mo}^{i,j}\in\{0,1\} \quad \forall i, \forall j \tag{23}$$

$$z_{om}^{i,j}\in\{0,1\} \quad \forall i, \forall j \tag{24}$$

$$\gamma^{i,j}\in\{0,1\} \quad \forall i,\ j=1\ldots D-1 \tag{25}$$

The objective function aims to minimize the days required to complete the roster iteration. As each duty in each day requires a trainmaster to service, the objection equivalently minimizes the total number of trainmasters needed for the roster. If the first assigned task to a duty (indexed by $p=1$) is a working/lifting task ($x_t^{i,j,1}=1 \quad \forall t\in T^wT^r$), then the duty is considered to be a working duty. Otherwise, the duty is defined as a standby duty and will not be considered in the objective function. As the objective function (1) aims to minimize the days in the final rostering, $\tau^{i,j}$ is designed to increase based on the week a duty is in such that the duties assigned in later weeks will be punished. For instance, $\tau^{1,j}\le\tau^{2,j}\ldots\le\tau^{M,j}$.

Constraint (2) ensures that each position p for day j of week i is assigned exactly one task $t\in T$. Constraint (3) makes sure that each task t $\in T^w$ is assigned and one trainmaster will perform the task. Constraints (4), (5) and (6) together calculate intermediate variable $y_{t_1,t_2}^{i,j,p}$ based on the values of variables $x_{t1}^{i,j,p}$ and $x_{t2}^{i,j,p+1}$. If $x_{t1}^{i,j,p}=1$ and $x_{t2}^{i,j,p+1}=1$, task $t2$ is performed following $t1$ while task $t1$ and $t2$ are respectively assigned to positions p and $p+1$ of day j of week i. In this case, the intermediate variable $y_{t_1,t_2}^{i,j,p}=1$ for later calculation. If either $x_{t1}^{i,j,p}\neq 0$ or $x_{t2}^{i,j,p+1}\neq 0$, then $y_{t_1,t_2}^{i,j,p}=0$. Suppose that connecting tasks $t1$ and $t2$ violates any scheduling rules (e.g., time and location), then constraint (6) together with constraints (4) and (5) ensure that $y_{t_1,t_2}^{i,j,p}=0$ to prevent this type of task connection.

Constraint (7) determines if the start and end stations of the first/last tasks in a duty is the home-base station. If $\sum_{t\in T^m} x_t^{i,j,1}=1$, then the first task of the duty in day j of week i starts from the home-base station. If $\sum_{p=2..P}\sum_{t\in T^{mn}} x_t^{i,j,p}=1$, the last task of the duty in day j of week i ends at the home-base station. Suppose that $\sum_{t\in T^m} x_t^{i,j,1}=1$ and $\sum_{p=2..P}\sum_{t\in T^{mn}} x_t^{i,j,p}=1$, then $z_{mm}^{i,j}$ must be 1 to represent that the first and last tasks in a duty respectively start and end at that home-base station. Similarly, in constraint (8), if $\sum_{t\in T^m} x_t^{i,j,1}=1$, the first task of the duty in day j of week i starts from the home-base station. If $\sum_{p=2..P}\sum_{t\in T^{on}} x_t^{i,j,p}=1$, the last task of the duty in day j of week i ends from an overnight station. Suppose that $\sum_{t\in T^m} x_t^{i,j,1}=1$ and $\sum_{p=2..P}\sum_{t\in T^{on}} x_t^{i,j,p}=1$, $z_{mo}^{i,j}=1$, indicating that the duty starts from the home-base station and ends at an overnight station. Constraint (9) is a scenario in which a duty starts from an overnight station ($\sum_{t\in T^o} x_t^{i,j,1}=1$) and ends at the home-base station ($\sum_{p=2..P}\sum_{t\in T^{mn}} x_t^{i,j,p}=1$). The interpretation is similar to those of constraints (7) and (8). Only one scenario specified in constraints (7)(9) for each duty can occur, which is ensured in constraint (10).

The first duty of a week must start at the home-base station, therefore constraint (11) ensures that either $z_{mm}^{i,1}=1$ or $z_{mo}^{i,1}=1$. Based on this assumption, $z_{om}^{i,1}$ must be 0. The last duty of a week must end at the home-base station. Constraint (12) makes sure that either $z_{mm}^{i,D}=1$ or $z_{om}^{i,D}=1$. Similarly, $z_{mo}^{i,D}=0$. The start time and start station of each duty can be identified based on the first task of that duty. For instance, if $x_t^{i,j,1}=1$, then the first task is t, the start station is S_t^s and the start time is WT_t^s. The end station and end time of a duty can be determined based on T^{mn} and T^{on}. If $y_{t_1,t_2}^{i,j,p}=1$ while $t_2\in T^{mn}T^{on}$, the last task of the duty is t_1, the end station is $S_{t_1}^e$ and the end time of the duty is $WT_{t_1}^e$. Constraints (13)-(19) are constructed based on this principle. Constraint (13) makes sure that the end station of the duty in day j of week i is the start station of the duty in day j+1. At the same time, to connect the two duties and form a feasible roster, the trainmaster must have enough rest time. This labor law constraint is enforced in constraint (14). The clock-in time of the next day ($\sum_{t\in T^m} WT_t^s \times x_t^{i,j+1,1}$), plus 24 hours, minus the clock-out time of the current day ($\sum_{p=1..P-1}\sum_{t_1}\sum_{t_2\in T^{mn}} WT_{t_1}^e \times y_{t_1,t_2}^{i,j,p}$) must be greater than the required rest time (R). The daily working hour constraint is described in constraint (15). For each duty, the preparation time ($\sum_{t\in T^w} RT^{bw}\times x_t^{i,j,1} + \sum_{t\in T^r} RT^{br}\times x_t^{i,j,1}$) and working time ($\sum_p\sum_{t_1}\sum_{t_2\in T^{t_1}} WH_{t_1t_2}\times y_{t_1,t_2}^{i,j,p}$) should not exceed the working hour limit of a day (WH_{day}^{max}). The preparation time (either RT^{bw} or RT^{br}) is calculated based on what the first task is (i.e., working task or lifting task).

Based on the daily working hours calculated in constraint (15), constraints (16) and (17) specify the lower and upper bounds of weekly working hours, respectively.

Constraints (18) and (19) limit the working hours of a two-day duty. If a duty is a two-day duty ($\gamma^{i,j}$ will be 1 in constraint (19)), then $\left(1-\gamma^{i,j}\right)\times Max = 0$ and the working hours are constrained by WH_{2days}^{max}. For duties that are not two-day duties, the working hours are not bound by this constraint because the right-hand-side of the equation is a large number.

If $z_{mo}^{i,j} = 1$ and $z_{om}^{i,j+1} = 1$, the duties of day j and j+1 in week i are considered as a two-day duty and $\gamma^{i,j} = 1$. Otherwise, $\gamma^{i,j} = 0$, meaning that these two duties are not a two-day duty. The decision variables of the model are binary integers (constraints (20)-(25)).

The mathematical formulation described above is a complicated program, especially when large-scale problem is to be solved. In fact, only crew scheduling and set covering problems have been shown to be NP-complete problems in the literature (i.e., [3]). The integration of crew rostering with crew scheduling can be even more difficult to solve. Therefore, we propose two solution schemes to solve the problem in the following sections and demonstrate their effectiveness in numerical experiments.

IV. SOLUTION APPROACH: A BRANCH-AND-PRICE-AND-CUT ALGORITHM (BPC)

In this section, we propose a BPC algorithm that can solve the integrated formulation that is described in Section III. The branch-and-price-and-cut algorithm integrates the branch and bound and column generation methods for solving large-scale integer programming problems [2]. The cut generation is included to further accelerate the solution process.

A. OVERALL ALGORITHMIC PROCESS

The proposed BPC is illustrated in Figure 2. We first decompose the problem into master and pricing programs. The master program identifies the roster of the duties/columns generated from the pricing program while the pricing program finds good duties/columns for the master program. The master and pricing programs will be detailed in Section IV-C and Section IV-D respectively.

FIGURE 2. Branch-and-Price-and-Cut Algorithm.

As the number of columns in the original master program can be large, we form the restricted master program (RMP), which only contains a limited number of columns/duties generated on the fly from the pricing program. In addition, as the RMP can remain difficult to solve, we solve the relaxation of RMP, whose integer variables are relaxed as continuous variables. After solving the linear relaxation of RMP, we generate the pricing program based on the dual variables from the relaxed RMP, which indicates what kind of duties should be generated to improve the roster in RMP. We then solve the pricing program to find the best reduced cost. If the resulting reduced cost is negative, the column/duty identified is added to the RMP, and we return to step 2. If the pricing program fails to identify new column with a negative reduced cost, we continue to examine if all determined variables are integers. If not, we go to step 8 and branch on the non-integral decision variable. Based on the identified solution, we add cuts to fathom the infeasible solutions from the enumeration tree to accelerate the solution procedure and move to step 2. If all decision variables are integer values, the final step terminates the process and reports the incumbent solution.

B. INITIAL FEASIBLE SOLUTION

To begin the algorithm with the feasible solution in Step 0 of Figure 2, we develop a heuristic solution. The heuristic begins from constructing the subnetwork that is illustrated

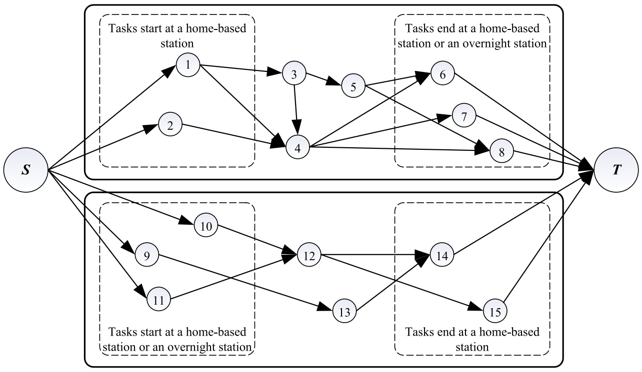

FIGURE 3. Illustration of subnetwork.

in Figure 3. The network contains a super source S and super sink node T. The remainder of the nodes in this network represent the tasks that must be serviced. The arcs in the network represent connecting the prior tasks to the following tasks that are specified in the nodes.

In the design of the subnetwork, each path corresponds to a feasible duty that can be performed by a trainmaster. To ensure the feasibility of each path, time, location and labor law constraints should be satisfied. For instance, if a trainmaster can perform task 4 after completing task 1 without violating the time constraints (i.e., the rest time is sufficient and does not violate labor laws) and location constraints (i.e., the end location of task 1 is the starting location of task 4), then node 1 and node 4 are connected. That is, when connecting two task nodes, constraints (4)(6) are satisfied automatically.

The arc of this network has two properties: cost and time. The arc cost can be adjusted based on the dual variables associated with the constraints in RMP. The adjustment is detailed in Section IV-D. The arc time is $WH_{t_1t_2}$. The time of each duty generated must satisfy the time constraint (15). Therefore, a constrained shortest path (CSP) algorithm adapted from [18] is used to obtain a solution. With this design, the duty hours of all identified paths in this network are feasible and do not violate any constraints. In other words, only the feasible duties will be generated in the design.

Based on the design, the following heuristic determines the initial feasible solution in an efficient manner. We use Figure 4 as an example to illustrate the proposed heuristic.

Step 1: Initialize node cost as -1 for all nodes. The costs of corresponding incoming arcs are initialized based on that node as -1. For instance, if the cost of node 4 is -1, then arcs $(1, 4)$ and $(3, 4)$ are of cost -1.

Step 2: Find the constrained shortest path from source node **S** to destination node ***T***. For all arcs connected to the nodes in that path, update the node and arc costs to 0. For instance, in Figure 4 (a), the first shortest path is $S-1-3-4-T$. Then, the costs of nodes 1, 3, 4 are updated as 0. The costs of all incoming arcs for these nodes are updated to 0 (as shown in Figure 4 (b)).

Step 3: Repeat Step 2 until the costs of all nodes are 0.

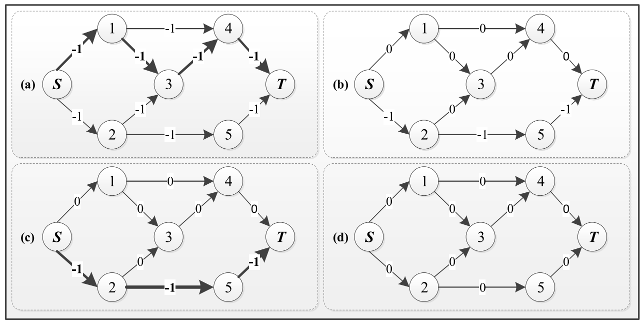

FIGURE 4. Illustrative network for the initial feasible solution heuristic.

Note that all the arc can be re-used when identifying the constrained shortest path, regardless of the arc cost (either 0 or -1). If a task is covered more than once, the additional coverages can be considered as lift tasks. Therefore, all solutions identified in this method are feasible.

In the network illustrated in Figure 3, it is also possible to generate a two-day duty. Suppose that a path that starts from a home-base station and ends at an overnight station is generated (for instance, path $S-2-4-7-T$ in Figure 3) and the end/overnight station is station k. Next, we modified the arc cost of the task whose start station is k (i.e., that the task associated with node 9 start from station k) in the lower half network to a large negative number in the next iteration (i.e., the arc cost of $(S, 9)$ is modified as a large negative number). With this modification, the next generated path will start from station k (i.e., path $S-9-13-14-T$ in Figure 3). Then, path $S-2-4-7-T$ and path $S-9-13-14-T$ together form a two-day duty.

With generated duties that cover all of the tasks, we can solve the RMP described in Section IV-C to form a feasible roster. As the number of duties generated in this heuristic is very limited, it is not computationally difficult to solve the resulting RMP when identifying the initial feasible solutions.

C. RESTRICTED MASTER PROGRAM (RMP)

In the algorithm design, because the duties are generated by the pricing sub-problem, the duties are known when solving the RMP (rostering problem). Therefore, the RMP can be simplified in the following manner.

Denote S^{home} as the home-base station, W_c^s as the clock-in time for duty c, W_c^e as the clock-out time for duty c, S_c^s as the start station of duty c, S_c^e as the end station of duty c, WH_c as the working hour of duty c and $d_{c,t}$ as the indicator parameters ($d_{c,t} = 1$ if task t is contained in duty c; otherwise, $d_{c,t}= 0$).

Let Du be the set of all duties, Du^s be the set of standby duties ($Du^s \subseteq Du$), Du^t be the set of working duties ($Du^t \subseteq Du$, $Du^s \cap D^{ut} = \emptyset$), Du^{mo} be the set of duties whose start station is a home-base station and end station is not a home-base station ($Du^{mo} \subseteq Du$), Du^{om} be the set of duties whose start station is not a home-base station and end station is a home-base station ($Du^{om} \subseteq Du$) and Du^{mm} be the duties whose start and end stations are a home-base station.

Based on the notation above, the decision variable of the RMP is $X_{i,j,c}$, whose value determines if duty c is assigned to day j of week i ($X_{i,j,c} = 1$ yes; $X_{i,j,c}= 0$, otherwise). We present the formulation of RMP, followed by an explanation.

$$\text{Minimize} \sum_{i=1}^{M}\sum_{j=1}^{D}\sum_{c\in Du^t} \tau^{i,j} \times X_{i,j,c} \tag{26}$$

$$\textit{Subject to} \sum_{c\in Du} X_{i,j,c} = 1 \quad \forall i,\ \forall j \tag{27}$$

$$\sum_{i=1}^{M}\sum_{j=1}^{D}\sum_{c\in Du^t} d_{c,t} \times X_{i,j,c} \geq 1 \quad \forall t \in T^w \tag{28}$$

$$\sum_{c\in Du} S_c^s \times X_{i,j,c} = S^{home} \quad \forall i, j = 1 \tag{29}$$

$$\sum_{c\in Du} S_c^e \times X_{i,j,c} = S^{home} \quad \forall i, j = D \tag{30}$$

$$\sum_{c\in Du} S_c^e \times X_{i,j,c} = \sum_{c\in Du} S_c^s \times X_{i,j+1,c} \quad \forall i, j = 1..D-1 \tag{31}$$

$$\sum_{c\in Du^{mo},Du^{mm}} W_c^s \times X_{i,j+1,c} + Day - \sum_{c\in Du^{om},Du^{mm}} W_c^e \times X_{i,j,c} \geq \text{R} \quad \forall i, j=1..D-1 \tag{32}$$

$$\sum_{j=1}^{D}\sum_{c\in Du} WH_c \times X_{i,j,c} \geq WH_{week}^{min} \quad \forall i \tag{33}$$

$$\sum_{j=1}^{D}\sum_{c\in Du} WH_c \times X_{i,j,c} \leq WH_{week}^{max} \quad \forall i \tag{34}$$

$$\sum_{c\in Du^{mo}} WH_c \times X_{i,j,c} + \sum_{c\in Du^{om}} WH_c \times X_{i,j+1,c} \leq WH_{2days}^{max} \quad \forall i, j = 1..D-1 \tag{35}$$

$$X_{i,j,c} \in \{0, 1\} \quad \forall \text{i}, \forall \text{j}, \forall \text{c} \tag{36}$$

In the simplified formulation, the duties can be classified as working duties (Du^t) and standby duties (Du^s) and the objective function (1) in the original problem is reduced to (26). Constraint (27) is similar to constraint (2). Constraint (2) ensures that each position p for day j of week i is assigned exactly one task $t\in T$. Constraint (27) makes sure that only one duty will be assigned to day j of week i. Constraint (28) corresponds to constraint (3) and ensures that all tasks will be covered by one trainmaster.

Because the tasks in each duty are known when solving RMP, based on the tasks assigned to a duty, the start station (S_c^s), end station (S_c^e), clock-in time (W_c^s), clock-out time (W_c^e) and working hours (W_c^h) of that duty are also known. Therefore, the location constraints that are specified in constraints (11)(12) can be modified to constraints (29)(30). Constraint (13) is reduced to constraint (31). The resting time constraint (14) can be stated as constraint (32). The weekly working hours constraints (16)(17) can be simplified as constraints (33)(34). The working hours limit of a two-day duty constraint (18) can be reduced to constraint (35)

Because the RMP determines the roster with given duties, the following pricing problem generates duties on-the-fly to improve the quality of the roster in RMP.

D. PRICING PROGRAM

Let $\boldsymbol{\omega}$ be dual variables associated with constraints (27), (29)(35) (i.e., $\omega_{m,n}^{27}$ is the dual variable associated with constraint (27)) and let π_t be the dual variable that corresponds to constraint (28). Based on the RMP described in Section IV-C, the reduced cost of variable $X_{i,j,c} \quad \forall c\in Du^t$ is:

$$\begin{aligned}
&\tau^{i,j} - \sum_{(m,n)\in(i,j)} \omega_{m,n}^{27} - \sum_{t\in T^w} d_{c,t} \times \pi_t - \sum_{m\in i, j=1} S_c^s \times \omega_m^{29} \\
&- \sum_{m\in i, j=D} S_c^e \times \omega_m^{30} - \sum_{(m,n)\in(i,j)} S_c^e \times \omega_{m,n}^{31} \\
&- \sum_{(m,n)\in(i,j+1)} S_c^s \times \omega_{m,n}^{31} - \sum_{(m,n)\in(i,j+1), c\in Du^{mo},Du^{mm}} W_c^s \times \omega_{m,n}^{32} \\
&- \sum_{(m,n)\in(i,j), c\in Du^{om},Du^{mm}} \left(-W_c^e\right) \times \omega_{m,n}^{32} - \sum_{m\in i} WH_c \times \omega_m^{33} \\
&- \sum_{m\in i} WH_c \times \omega_m^{34} - \sum_{(m,n)\in(i,j), c\in Du^{mo}} WH_c \times \omega_{m,n}^{35} \\
&- \sum_{(m,n)\in(i,j+1), c\in Du^{om}} WH_c \times \omega_{m,n}^{35}
\end{aligned} \tag{37}$$

In each iteration, a duty/column should be generated based on the reduced cost. However, because the day and week of a duty to be assigned are determined in the RMP, the pricing program only generates the duty/column that covers the tasks that must be covered the most. Therefore, dual variables related to week i and day j (namely, $\boldsymbol{\omega}$) can be ignored and the index (ijp) is replaced by p. For instance, $x_t^{i,j,p}$ and $y_{t_1,t_2}^{i,j,p}$ are respectively replaced by x_t^p and y_{t_1,t_2}^p. As a result, the objective function can be reduced to:

$$\min - \sum_{t\in T^w} d_{c,t} \times \pi_t \tag{38}$$

Because the parameter $d_{c,t}$ indicates whether or not a task t is contained in duty c, we can replace $d_{c,t}$ with $\sum_{p=1}^{P} x_t^p$ and form the following pricing program:

$$\textit{Maximize} \sum_{t\in T^w}\sum_{p=1}^{P} x_t^p \times \pi_t \tag{39}$$

$$\textit{Subject to} \sum_{t} x_t^p = 1 \quad \forall \text{p} \tag{40}$$

$$x_{t_1}^p + x_{t_2}^{p+1} - 2y_{t_1,t_2}^p \geq 0 \quad \forall \text{p} = 1\ldots\text{P}-1 \quad \forall t_1 \in T, \forall t_2 \in T \tag{41}$$

$$x_{t_1}^p + x_{t_2}^{p+1} - y_{t_1,t_2}^p \leq 1 \quad \forall \text{p} = 1\ldots\text{P}-1 \quad \forall t_1 \in T, \forall t_2 \in T \tag{42}$$

$$y^{p}_{t_1,t_2} = 0 \quad \forall p = 1 \ldots P-1, \quad \forall t_1 \in T,\ \forall t_2 \notin T^{t_1} \tag{43}$$

$$\sum_{t\in T^m} x^1_t + \sum_{p=2..P}\sum_{t\in T^{mn}} x^p_t \geq 2z_{mm} \tag{44}$$

$$\sum_{t\in T^m} x^1_t + \sum_{p=2..P}\sum_{t\in T^{on}} x^p_t \geq 2z_{mo} \tag{45}$$

$$\sum_{t\in T^o} x^1_t + \sum_{p=2..P}\sum_{t\in T^{mn}} x^p_t \geq 2z_{om} \tag{46}$$

$$z_{mm} + z_{mo} + z_{om} = 1 \tag{47}$$

$$\sum_{t\in T^w} RT^{bw} \times x^1_t + \sum_{t\in T^r} RT^{br} \times x^1_t + \sum_p \sum_{t_1} \sum_{t_2\in T^{t_1}} WH_{t_1t_2} \times y^p_{t_1,t_2} \leq WH^{max}_{day} \tag{48}$$

$$x^p_t \in \{0,1\} \quad \forall p,\ \forall t \tag{49}$$

$$y^p_{t_1,t_2} \in \{0,1\} \quad \forall p = 1 \ldots P-1 \quad \forall t_1 \in T,\ \forall t_2 \in T \tag{50}$$

$$z_{mm} \in \{0,1\} \tag{51}$$

$$z_{mo} \in \{0,1\} \tag{52}$$

$$z_{om} \in \{0,1\} \tag{53}$$

The objective function (39) aims to identify the column/duty that benefits the rostering in the RMP the most based on the dual variable that corresponds to constraint (28). Constraint (40) is equivalent to constraint (2), stating that each position can be assigned a task. Constraints (41)(47) are comparable to constraints (4)(10) with minor modifications of indices. The interpretations of these constraints are similar to the original program and are referred to in Section III-C. Constraint (48) is equivalent to constraint (15), indicating the upper bound of daily working hours. The decision variable is a binary integer (constraints (49)(53)).

With this formulation, we can design a sub-problem network similar to the network shown in Section IV-B, set the costs of the corresponding incoming arcs of node t as $-\pi_t t\in T^w$ and solve it with the constrained shortest path algorithm that was proposed by Lin (2014)

E. CUT GENERATION

To accelerate the solution process, we develop the following cuts to prune the branch-and-price-and-cut tree whenever possible.

Cut I *(Cut to Avoid Multiple Duties in One Day):* As shown in constraint (27), each day can only be assigned one duty. If $X^r_{i,j,c}$ at the r^{th} iteration is a binary value (i.e., $X^r_{i,j,c} = 1$), then the following cut (54) can be added for the r'^{th} iteration after r^{th} $(\forall r' > r)$:

$$\sum_{c'\in Du} X^{r'}_{i,j,c'} = 1 - \sum_{c} X^r_{i,j,c} \quad \forall (i,j),\ \forall r' > r \tag{54}$$

Cut II *(Cut for Infeasible Connecting Location):* For any $X_{i,j,c} = 1$, then $X_{i,j+1,c'} = 0 \quad \forall S^s_{c'} \neq S^e_c$. Considering constraint (31), if $X^r_{i,j,c} = 1$, meaning that $\sum_{c'\in Du} S^e_{c'} \times X^r_{i,j,c'} = S^e_c$, then constraint (31) can be simplified as:

$$\sum_{c'\in Du} S^s_{c'} \times X^{r'}_{i,j+1,c'} = S^e_c, \quad \forall i,j = 1..D-1,\ \forall r' > r \tag{55}$$

As $X^{r'}_{i,j+1,c'} \in \{0,1\}$, the following cut (56) can be added to restrict the value of variable $X^{r'}_{i,j+1,c'}$ whose duty c''s start station $S^s_{c'}$ is not equal to end station S^e_c of duty c:

$$\text{if } S^s_{c'} \neq S^e_c, \quad \text{then } X^{r'}_{i,j+1,c'} = 0, \quad \forall i,j = 1..D-1,\ \forall r' > r \tag{56}$$

In other words, if $X_{i,j,c} = 1$, the tree node of $X_{i,j+1,c'}$ $\forall S^s_{c'} \neq S^e_c$ and its following leaf nodes can be pruned. This cut is a location-based cut to ensure the connecting locations of duties.

Cut III *(Cut for Infeasible Connecting Time):* Derived from constraint (32), if $X^r_{i,j,c} = 1$ and $c\in Du^{om}Du^{mm}$, indicating that $\sum_{c'\in Du} W^e_{c'} \times X^r_{i,j,c'} = W^e_c$, then constraint (32) can be simplified as:

$$\sum_{c'\in Du^{mo},Du^{mm}} W^s_{c'} \times X^{r'}_{i,j+1,c'} \geq W^e_c + \text{R}-Day \quad \forall i,j = 1..D-1,\ \forall r' > r \tag{57}$$

Similar to ***Cut II***, if $c'\in Du^{mo}Du^{mm}$ and $W^s_{c'} < W^e_c + \text{R}-Day$, then cut (58) can be added to restrict the value of variable $X^{r'}_{i,j+1,c'}$:

$$X^{r'}_{i,j+1,c'} = 0, \quad \forall i,j = 1..D-1,\ \forall r' > r \tag{58}$$

This cut is a time-based cut that is used to ensure that the rest time between duties does not violate the scheduling/rostering rule.

Proposition 1: ***Cuts I-III*** or their combination never exclude the optimal solutions of the integrated program.

Proof: Define a solution set S that represents all solutions of the problem. Assume that the optimal solution of problem is S^*. Let $S^* \subseteq S'$ and assume that Cuts I-III exclude S'. Since solution S' is excluded by cuts due to its infeasibility, S^* is neither a feasible nor an optimal solution, which contradicts with the assumption.

Proposition 2: The BPC algorithm finds an optimal solution in a finite number of iterations.

Proof: From ***Proposition 1***, ***Cuts I-III*** never exclude the optimal solutions. Since these cuts are always added into the master problem, the feasible search space can be reduced. Because the solution space is finite, the algorithm can find the optimal solution in a finite number of iterations.

V. DEPTH-FIRST SEARCH-BASED (DFS-BASED) ALGORITHM

In this section, we propose a DFS-based algorithm that can solve the crew scheduling and rostering problems independently and sequentially. With this DFS-based algorithm, we can compare the two-stage solutions with the integrated solutions from the BPC.

Algorithm 1 DFS

```
Begin
Initialize a list Duty= {Ø} for duty storage
Sort the tasks by departure time and number them as 1...N
  For i := 1 to N do
  Begin
    Order ← i;
    Let k be task i;
    While Order≠ {Ø}
    Begin
      For j := k + 1 To N
      Begin
      Let l be the last element in Order;
      If connecting l and j satisfies the scheduling rules
      Then
        Order ← Order ∪ j;
      End If
      If j = N then
        If Order satisfies the scheduling rules
         Then
            Duty ← Duty ∪ Order;
        End If
      End If
      End For
      Let k be the last element in Order;
      Delete the last element in Order;
    End While
  End For
End
```

The DFS-based algorithm has three steps: column/duty enumeration, scheduling optimization and roster optimization. The first step enumerates the potential columns/duties with the DFS search procedure *Algorithm DFS.* During column enumeration in the DFS, the scheduling rules are examined so that infeasible duties can be eliminated. The second step is a set partitioning problem that identifies the minimum number of duties required to cover a timetable. The final step solves the roster optimization that is described in Section IV-C with the columns/duties to ensure that the roster is formed without violating the rostering rules. The DFS search procedure can be summarized in the following pseudocode.

The DFS search procedure enumerates all of the potential duties. We then solve the following set partitioning program to identify the minimum number of duties to cover all of the tasks. In the program, the parameter $d_{c,t}$ indicates whether or not task t is contained in duty c and the decision variable X_c determines if duty c is selected.

$$\text{Minimize} \sum_{c \in Du} X_c \tag{59}$$

$$\textit{Subject to} \sum_{c \in Du} d_{c,t} \times X_c = 1 \quad \forall t \in T^w \tag{60}$$

$$X_c \in \{0, 1\} \quad \forall c \in Du \tag{61}$$

With the selected duties, we solve the program described in Section IV-C so that a feasible roster can be formed.

VI. EMPIRICAL STUDY

To validate the effectiveness and evaluate the efficiency of the proposed solution frameworks, we empirically apply them to problems of different sizes. In the validation, we compare the solutions derived using our methods to the solutions from a commonly used commercial optimization package Gurobi. The solution from Gurobi can be considered to be the optimal solution and serves as a benchmark for the other two algorithms. The proposed BPC and DFS-based algorithms are implemented in the ANSI C++ programming language. The numerical experiments for both Gurobi 7.5.2 and our solution methods are conducted on a Windows-based machine with an Intel 3.4 GHz CPU processor and 16 GB of memory.

A. VALIDATION

The data and results from the experiment are respectively summarized in Table 1 and Table 2 There are fix groups of trainmasters tested. The study area is located in southern Taiwan and is a portion of the network that the Taiwan Railways Administration (TRA) currently operates. The groups of trainmasters service all of the train services in this area that are listed in the timetable.

TABLE 1. Experiment data.

	Group A	Group B	Group C	Group D	Group E	Group F
Number of tasks	59	82	156	183	255	305

TABLE 2. The comparison of solutions from different approaches.

Group	*Current Practice*		*Gurobi*			*DFS-based*			*BPC*			
	WD[1]	SD[2]	WD	SD	CPU[3]	WD	SD	CPU	WD	SD	Gap (%)	CPU
A	17	1	15	3	73.4	15	15	14.8	15	3	0.00	12.8
B	12	0	-	-	-	11	7	0.2	11	1	0.46	13.9
C	41	1	-	-	-	-	-	-	41	1	2.89	1,897
D	47	1	-	-	-	-	-	-	46	2	0.00	2,762
E	61	5	-	-	-	-	-	-	60	6	1.69	3,158
F	82	2	-	-	-	-	-	-	82	2	2.86	3,606

In Table 2, we present the results from TRA's current practice, the solution from the integrated formulation that was described in Section III-C using Gurobi, and the solutions from the DFS-based and BPC algorithms. In addition, the convergence of the proposed BPC is depicted in Figure 5. The TRA's current practice is determined by a manual approach and relies heavily on rule of thumb. Because it is challenging to prepare a schedule or roster, TRA rarely prepares the schedule/roster from scratch. Instead, they typically modify the current schedule/roster and attempt to find a feasible schedule/roster with minimal changes.

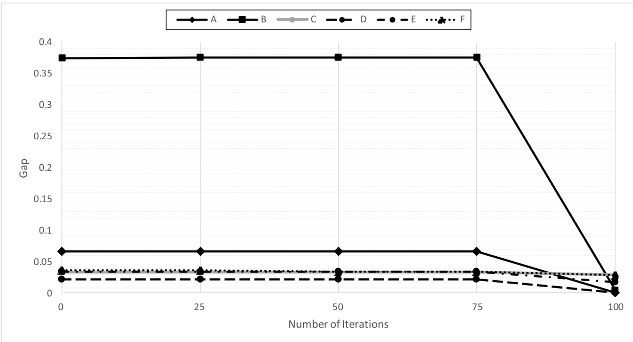

FIGURE 5. The convergence of BPC.

Group A is the problem instance that can be solved by all four solution approaches. In TRA's practice, there are 17 working trainmasters and 1 standby trainmaster. In the optimal solution determined by Gurobi, the same number of trainmasters is needed. However, servicing the same timetable, only 15 working trainmasters are needed and 3 trainmasters can be standby which significantly improves the flexibility of TRA's operation. If there is any emergency, TRA can have more standby trainmasters to handle the sudden occurrence. In this instance, the DFS-based algorithm obtains the number of working trainmasters and standby trainmasters as 15 and 15, respectively, which is worse than the Gurobi solution. The DFS-based algorithm corresponds to a typical two-stage solution approach that is frequently adopted in the literature (i.e., to solve scheduling and rostering independently and sequentially). The aim of the first stage is minimizing the number of duties required to cover a timetable. The output of the first stage of the problem is then used in the second stage to determine the minimal roster. However, in determining duties in the first stage, the roster of the second stage is not considered. Therefore, the resulting second stage requires many standby trainmasters to form a feasible roster. The result demonstrates the benefit of integrating the scheduling and rostering within one formulation. For the proposed BPC algorithm, the solution is identical to that from Gurobi, which shows the effectiveness of the BPC algorithm. In terms of computational time, both the DFS-based and BPC algorithms outperform Gurobi and can obtain solutions faster. Compared to TRA's current practice, although the total number of trainmasters is identical, the solutions from Gurobi/BPC are more flexible because the number of standby trainmasters is higher than in TRA's current practice.

In Group B, because there are many short and frequent train tasks in the timetable, the number of potential combinations and resulting schedules/rosters is significant. Thus, although the number of trainmasters is smaller than the number of Group A, the resulting program is beyond the reach of Gurobi because the numbers of decision variables and constraints are extremely large. Therefore, in this and following experiments, only the solutions from our methods are reported. In this experiment, both the DFS-based and BPC algorithms can determine solutions within reasonable computational times. The DFS-based algorithm outperforms the BPC algorithm in computational time in this case. However, the solution quality of the BPC algorithm remains better than that of the DFS-based algorithm. In the BPC solution, the required trainmasters require a lower number of standby trainmasters when compared to the DFS-based solution. With the same number of total trainmasters, the solution from BPC is still more flexible because one more standby trainmaster is arranged.

When the problem size grows larger (i.e. Groups C, D, E and F), even the DFS-based algorithm struggles and cannot determine the solution, as does the BPC algorithm. Therefore, the BPC is the most efficient solution approach and can potentially solve problems of realistic sizes. In terms of solution quality, BPC identifies the solutions that are at least as good as those developed by TRA in practice over several years of modifications (i.e., Groups C and F) with reasonable computational effort. In Group D and E, BPC even finds a solution with one less working duty and one more standby duty. This kind of solution can improve the flexibility of the company's operations.

B. SENSITIVITY ANALYSIS

We next conduct a sensitivity analysis of the problem parameters for Group A to examine the impact of these parameters. The results are summarized in Table 3. In this table, we present more details such that the impact of parameter adjustment can be accurately evaluated.

When perturbing the maximum daily working hours, we found that no feasible solution can be identified for total daily working hours less than 9 hours, indicating that some tasks required more than 9 hours to complete. When the maximum daily working hours increase, the number of working and standby duties remain identical, which means that the maximum number of working hours may not be the decisive factor in the scheduling/rostering. However, the average preparation time per week and the maximum waiting time between tasks decreases when compared to the base case. These decreases show a better usage of trainmasters' working hours.

In the sensitivity analysis (b) of Table 3, when the maximum weekly working hours increase, the numbers of working and standby duties do not change. However, the required working and standby duties increase when the maximum weekly working hours decrease. Compared to the results shown in analysis (a) of Table 3, the maximum weekly working hours is a more critical factor than the maximum daily working hours. This phenomenon is more apparent in analysis (c) of Table 3. Even with a very high maximum daily working hours (i.e., 12 hours), only the maximum weekly working hours has an impact on the number of duties required. It is believed that the maximum weekly working hours actually constrains the feasible region and thus the impact of the maximum daily working hours is

TABLE 3. Sensitivity analysis.

(a) Maximum daily working hours					
Maximum daily working hours	**8**	**9**	**10***	**11**	**12**
The number of working duties	-	-	15	15	15
The number of standby duties	-	-	3	3	3
Average working hours per week (in minutes)	-	-	2,623	2,625	2,616
Average preparation time per week (in minutes)	-	-	662	644	655
Maximum waiting time between tasks (in minutes)	-	-	404	189	189
(b) Maximum weekly working hours					
Maximum weekly working hours	**40**	**42**	**44***	**46**	**48**
The number of working duties	16	15	15	15	15
The number of standby duties	8	9	3	3	3
Average working hours per week (in minutes)	2,391	2,507	2,623	2,673	2,749
Average preparation time per week (in minutes)	470	497	662	712	748
(c) Maximum daily and weekly working hours					
Maximum daily working hours	**12**	**12**	**12****	**12**	**12**
Maximum weekly working hours	**40**	**42**	**44***	**46**	**48**
The number of working duties	16	15	15	15	15
The number of standby duties	8	9	3	3	3
Average working hours per week (in minutes)	2,391	2,507	2,616	2,676	2,699
Average preparation time per week (in minutes)	470	497	655	715	738
(d) Rest Time					
Rest time between duties	**10**	**11**	**12***	**13**	**14**
The number of working duties	15	15	15	15	15
The number of standby duties	3	3	3	3	3
Average working hours per week (in minutes)	2,618	2,634	2,623	2,618	2,616
Average preparation time per week (in minutes)	657	673	662	657	655

*current value; **current value is 10

less significant. This kind of observation is only possible when the scheduling and rostering problems are considered in an integrated manner, which demonstrates the benefit of integration in our research. In analysis (d) of Table 3, the rest time also has no obvious impact on the number of duties required. The explanation for this is similar to the explanation presented above.

VII. CONCLUDING REMARKS

In this research, we formulate the integrated scheduling and rostering problem of trainmasters for passenger railway transportation. To efficiently solve the resulting program, we propose two solution approaches: DFS-based and BPC algorithms. Empirical results show that the BPC algorithm performs best and can solve problems of realistic size using reasonable computational effort. Further, an integrated framework yields better solutions when compared to a conventional two-stage solution strategy. Finally, we found that rostering rules had more impact on the final results than scheduling rules. Although encouraging results are obtained with the proposed solution framework, the current study can be extended in many directions. For instance, as shown in many studies, fairness is as important as efficiency in practice. In addition, disruptions and integration with timetables can be an interesting direction of extension. Because the current research integrated scheduling and rostering problems, it sheds light on taking these practical issues into consideration.

REFERENCES

[1] L. Bach, T. Dollevoet, and D. Huisman, "Integrating timetabling and crew scheduling at a freight railway operator," *Transp. Sci.*, vol. 50, no. 3, pp. 878–891, 2016.
[2] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Oper. Res.*, vol. 46, no. 3, pp. 316–329, 1998.
[3] J. J. Bartholdi, III, "A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering," *Oper. Res.*, vol. 29, no. 3, pp. 501–510, 1981.
[4] L. Bianco, M. Bielli, A. Mingozzi, S. Ricciardelli, and M. Spadoni, "A heuristic procedure for the crew rostering problem," *Eur. J. Oper. Res.*, vol. 58, no. 2, pp. 272–283, 1992.
[5] A. Caprara, M. Fischetti, P. L. Guida, P. Toth, and D. Vigo, "Solution of large-scale railway crew planning problems: The Italian experience," in *Comput.-Aided Transit Scheduling*, vol. 471, pp. 1–18, 1999.
[6] A. Caprara, M. Fischetti, P. Toth, D. Vigo, and P. L. Guida, "Algorithms for railway crew management," *Math. Program.*, vol. 79, no. 1–3, pp. 125–141, 1997.
[7] A. Caprara, P. Toth, D. Vigo, and M. Fischetti, "Modeling and solving the crew rostering problem," *Oper. Res.*, vol. 46, no. 6, pp. 820–830, 1998.
[8] S. C. K. Chu and E. C. H. Chan, "Crew scheduling of light rail transit in Hong Kong: From modeling to implementation," *Comput. Oper. Res.*, vol. 25, no. 11, pp. 887–894, 1998.
[9] A. T. Ernst, H. Jiang, M. Krishnamoorthy, H. Nott, and D. Sier, "An integrated optimization model for train crew management," *Ann. Oper. Res.*, vol. 108, nos. 1–4, pp. 211–224, 2001.
[10] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier, "Staff scheduling and rostering: A review of applications, methods and models," *Eur. J. Oper. Res.*, vol. 153, no. 1, pp. 3–27, 2004.
[11] R. Freling, R. M. Lentink, and A. P. M. Wagelmans, "A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm," *Ann. Oper. Res.*, vol. 127, nos. 1–4, pp. 203–222, 2004.
[12] K. Ghoseiri, F. Szidarovszky, and M. J. Asgharpour, "A multi-objective train scheduling model and solution," *Transp. Res. B, Methodol.*, vol. 38, no. 10, pp. 927–952, 2004.
[13] K. Hoffmann, U. Buscher, J. S. Neufeld, and F. Tamke, "Solving practical railway crew scheduling problems with attendance rates," *Bus. Inf. Syst. Eng.*, vol. 59, no. 3, pp. 147–159, 2017.
[14] D. Huisman, "A column generation approach for the rail crew re-scheduling problem," *Eur. J. Oper. Res.*, vol. 180, no. 1, pp. 163–173, 2007.
[15] S. Jütte, M. Albers, U. W. Thonemann, and K. Haase, "Optimizing railway crew scheduling at DB schenker," *Interfaces*, vol. 41, no. 2, pp. 109–122, 2011.
[16] R. S. K. Kwan, "Case studies of successful train crew scheduling optimisation," *J. Scheduling*, vol. 14, no. 5, pp. 423–434, 2011.

[17] M. Lezaun, G. Pérez, and E. Sáinz de la Maza, ''Rostering in a rail passenger carrier,'' *J. Scheduling*, vol. 10, nos. 4–5, 245–254, 2007.
[18] D.-Y. Lin, ''A dantzig–wolfe decomposition algorithm for the constrained minimum cost flow problem,'' *J. Chin. Inst. Eng.*, vol. 37, no. 5, pp. 659–669, 2014.
[19] M. Liu, A. Haghani, and S. Toobaie, ''Genetic algorithm-based column generation approach to passenger rail crew scheduling,'' *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2159, pp. 36–43, 2010. doi: 10.3141/2159-05.
[20] T. Nishi, T. Sugiyama, and M. Inuiguchi, ''Two-level decomposition algorithm for crew rostering problems with fair working condition,'' *Eur. J. Oper. Res.*, vol. 237, no. 2, pp. 465–473, 2014.
[21] D. Potthoff, D. Huisman, and G. Desaulniers, ''Column generation with dynamic duty selection for railway crew rescheduling,'' *Transp. Sci.*, vol. 44, no. 4, pp. 493–505, 2010.
[22] N. J. Rezanov and D. M. Ryan, ''The train driver recovery problem—A set partitioning based model and solution method,'' *Comput. Oper. Res.*, vol. 37, no. 5, pp. 845–856, 2010.
[23] M. Saddoune, G. Desaulniers, I. Elhallaoui, and F. Soumis, ''Integrated airline crew pairing and crew assignment by dynamic constraint aggregation,'' *Transp. Sci.*, vol. 46, no. 1, pp. 39–55, 2012.
[24] N. Souai and J. Teghem, ''Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem,'' *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 674–683, 2009.
[25] L. P. Veelenturf, D. Potthoff, D. Huisman, and L. G. Kroon, ''Railway crew rescheduling with retiming,'' *Transp. Res. C, Emerg. Technol.*, vol. 20, no. 1, pp. 95–110, 2012.

DUNG-YING LIN received the bachelor's and M.B.A. degrees in transportation and communication management (TCM) science from National Cheng Kung University, Tainan, Taiwan, and the Ph.D. degree in civil engineering from The University of Texas at Austin, TX, USA. From 2009 to 2012 and from 2012 to 2016, he was an Assistant Professor and an Associate Professor, respectively, of TCM with National Cheng Kung University, where he was a Professor, from 2016 to 2018. He is currently a Professor with the Department of Industrial Engineering and Engineering Management, National Tsing Hua University. He has published 49 papers in the fields of optimization, scheduling, and logistics.

MENG-RUNG TSAI received the B.B.A. degree and the M.B.A. degree in transportation and communication management science from National Cheng Kung University, Tainan, Taiwan, in 2015 and 2017, respectively, where she is currently a Graduate Student of transportation and communication management science.