Ann Oper Res (2014) 223:173–193
DOI 10.1007/s10479-014-1619-1

A constraint programming-based approach to the crew scheduling problem of the Taipei mass rapid transit system

Anthony F. Han · Elvis C. Li

Published online: 30 May 2014

Abstract This paper addresses the crew scheduling problem for a mass rapid transit (MRT) system. The problem is to find a minimum number of duties to cover all tasks while satisfying all the hard and soft scheduling rules. Such rules are complicated in real-world operations and difficult to follow through optimization methods alone. In this paper, we propose a constraint programming (CP)-based approach to solve the problem. The approach involves a CP model for duty generation, a set covering problem model for duty optimization, and alternative ways to identify the final solution in different situations. We applied the proposed CP-based approach to solve a case problem for the Taipei MRT. Case application results using real-world data showed that our approach is capable of reducing the number of daily duties from 58 to 55 and achieving a 5.2 % savings in labor costs. We also incorporated the soft rule considerations into the CP model in order to generate alternative optimum solutions that would improve the workload balance. The coefficient of variation of the work time distribution improves significantly, falling from 21 % to approximately 5 %. Given the CP model's comprehensive coverage of various scheduling rules, our proposed approach and models would also be applicable to other MRT systems.

Keywords Crew scheduling problem · Mass rapid transit · Constraint programming · Set covering problem

1 Introduction

Crew scheduling is an important issue for crew management in public transportation systems. Crew management is concerned with building the rosters of crews needed to cover a planned

A. F. Han (✉) · E. C. Li
Department of Transportation Technology and Management,
National Chiao Tung University, Hsinchu 30010, Taiwan
e-mail: afhan@mail.nctu.edu.tw

E. C. Li
e-mail: zyun.tem94g@nctu.edu.tw

timetable while satisfying operational and regulation rules. In practice, this complex and challenging problem usually divides into crew scheduling and crew rostering (Caprara et al. 1997). Crew scheduling aims to fulfill a minimum number of duties to cover all tasks without violating any scheduling rules, such as break time requirements and maximum continuous driving time limits. Based on the results of crew scheduling, crew rostering then generates a roster that spans a given time period according to rostering rules (Yunes et al. 2005). Crew scheduling thus plays a critical role in assisting a transportation agency or company in minimizing its labor costs. In this paper, we analyze the crew scheduling problem for the Taipei mass rapid transit (MRT) system and propose a constraint programming (CP)-based approach to solve the problem.

Crew scheduling problems have been applied to different transportation sectors including airlines, bus transit, and the railway industry. Crew scheduling problems in the respective sectors share many elements, such as the constraints imposed by rules and regulations, and the need to cover all tasks while seeking to minimize labor costs. However, each application has its own unique characteristics and its own research challenges. Accordingly, most crew scheduling research focuses on a particular application area, rather than dealing with a general case (Ernst et al. 2004). In this paper, we focus on the crew scheduling problem for MRT systems. Good reviews of crew scheduling problems in other application areas include (Barnhart et al. 2003) for airlines, (Wilson 1999) for bus transit, and (Caprara et al. 1997) for railways.

Existing studies on the crew scheduling problem for an MRT system basically consider the problem as an optimization problem and propose approaches based on heuristic or metaheuristic methods to solve the issue. For example, Chu and Chan (1998) proposed a three-stage heuristic method for solving the crew scheduling problem for the light rail transit system in Hong Kong. Cavique et al. (1999) and Chew et al. (2001) applied tabu search metaheuristics and presented cases on the Lisbon Underground and the Singapore MRT, respectively. These three studies showed improvement over previous schedules. However, they fixed only two or three pieces of work in each of the duties generated, and thus might limit the potential for obtaining better results. Park and Ryu (2006) employed a genetic algorithm with unexpressed genes to solve a subway crew schedule, but the objective was to minimize uncovered tasks rather than to minimize the labor costs. Elizondo et al. (2010) applied an evolutionary algorithm to solve crew scheduling for the Santiago Metro System, but they did not compare their results with actual schedules.

While existing literature on MRT applications relies mainly on metaheuristic methods, this paper presents a CP-based approach to tackle the crew scheduling problem of the Taipei MRT. This approach consists of two phases: duty generation, which is actually a constraint satisfaction problem,and duty optimization. Specifically, we propose the use of a CP model for duty generation and a set covering problem (SCP) model for duty optimization. Conventional approaches to tackle the MRT crew scheduling problem mainly take it as a constrained optimization problem in the structure of integer programming (IP). However, the numerous side constraints due to those scheduling rules tend to make the problem difficult to solve even with prevailing metaheuristics such as tabu search (Cavique et al. 1999) and evolutionary algorithms (Elizondo et al. 2010). On the other hand, the roots of CP can be traced back to the work in the field of artificial intelligence (AI) on solving constraint satisfaction problems in the 1970s (Mackworth 1977). Because a constraint satisfaction problem may include constraints in the form of logic implications, a CP system usually interleaves search and logical inference to find a feasible solution. Such inferences generally involve constraint propagation, which propagates the information contained in one constraint to the neighboring constraints, and domain reduction, which reduces the range of variable domains that need to be traced, to expedite the search process (Lustig and Puget 2001). Brailsford et al. (1999) compared CA with OR techniques and deemed that, for ease of implementation and flexibility, CP scores highly relative to most OR techniques. For computational time and solution quality, according to Puget (1995) the success of a CP approach critically depends on the amount of propagation that the constraints permit. That is if the nature of a problem is such that a variable instantiation triggers the pruning of many values from the domains of other variables, then CP is likely to be successful. He argues that scheduling and timetabling problems are good candidates for CP. Therefore, it is reasonable for us to use CP to deal with the constraint satisfaction problem confronted in the duty generation phase. Moreover, a good CP model also serves as an efficient column generator which facilitates solving a large-scale SCP which might be encountered in the duty optimization phase as discussed below.

Although a CP-based approach is new to MRT applications, it is not uncommon in other application areas such as airlines (Fahle et al. 2002; Gabteni and Grönkvist 2009; Grönkvist 2006) and bus transit systems (Silva 2001; Ftulis et al. 1998; Yunes et al. 2005). One advantage of CP is that it can express complex real-life constraints easily. Another advantage is that it has the flexibility necessary to adopt changes in rules and to add new rules. Accordingly, a CP-based model for crew scheduling applications is more transferable to other cases than conventional models purely based on heuristic or metaheuristic methods. In our proposed approach, the CP model serves as a good tool for duty generation. Moreover, when the problem scale of the SCP in the duty optimization phase is too large for the problem to be solved exactly, the CP model can be used to help solve the pricing sub-problem by generating columns for the restricted SCP. Examples of how the CP-based column generation (CG) applies to airlines and bus systems were presented in Fahle et al. (2002) and Silva (2001), respectively. It has been found that the CP-based CG is more efficient than the MP-based CG (Yunes et al. 2005). In recent years, CP has advanced significantly in many ways (Apt 2003) and has been increasingly recognized as a powerful approach complimentary to OR methods in solving combinatorial problems (Hentenryck 2002; Hooker 2006).

The remainder of this paper is organized as follows. In Sect. 2, we first define the crew scheduling problem for a general MRT system and then list all the operational and regulatory rules and constraints that are most commonly seen in practice. We then propose the framework of a CP-based two-phase solution approach. In Sect. 3, we start with a brief description of the Taipei MRT and define the scope of the case problem as covering the Red–Orange line. This section also specifies the related data and parameters. Section 4 presents the modeling process for our case application. The decision variables, their domains, and all the constraints of the CP model for duty generation are described in detail. In addition, this section presents the SCP model for duty optimization. Section 5 reports the case results in which we find that our proposed methods can create 5 % savings in crew costs on the Red–Orange line. This section also shows how the CP model can be applied to improve the workload balance among the train drivers. Finally, Sect. 6 concludes the paper.

2 Problem description and CP-based solution framework

2.1 Problem description

When we refer to the crew of the MRT crew scheduling problem, we mean the operators or drivers of MRT trains. A full understanding of the MRT crew scheduling problem relies on the knowledge of the daily operation of the trains moving between MRT stations. The stations of a typical MRT line can be classified into three categories. First are those stations that provide services primarily to MRT riders, meeting their requirements for boarding, alighting, transferring, and the like, but that do not provide services to the MRT trains for their parking and maintenance needs. For simplicity, these only-for-transit stations, if not otherwise specified, will be referred to as transit stations in the remainder of this paper. Second, terminal stations, or terminals, are those stations that not only serve riders but also provide parking space and facilities for all the trains operated on the associated MRT line. Terminals are usually at the endpoints of an MRT line. Third, yard stations, or yards, are those stations that are not open to serve the riders, but provide facilities and services exclusively for the MRT trains for their maintenance and storage. A yard is usually located near a terminal station. Because yards are not open for transit services, they are not shown on the MRT map for the general public. However, yard stations are essential to the daily operations of MRT trains, and thus cannot be overlooked in the MRT crew scheduling problem. Figure 1 depicts a typical MRT line including the yard stations.

The daily operation of MRT trains can be divided into three periods or cycles: (1) public transit service, (2) pre-transit operation, and (3) post-transit operation. The public transit service cycle generally starts early in the morning and ends around midnight on the same day. Each trip of an MRT train operated in this cycle runs in a terminal-to-terminal pattern, running from the origin terminal to the destination terminal and stopping at each of the intermediate stations; for example, the routes would be B–C–D–E–F and F–E–D–C–B for the line depicted in Fig. 1. The pre-transit operation cycle usually starts about 2 h before the public transit service cycle. There are two operational functions involved in this cycle. The first is for dispatch-to-work, which requires a driver to move the MRT train from a yard to a terminal station in preparation for the following operations originating from that terminal, e.g., A–B and G–F for the line in Fig. 1. The second is for track inspection, which also runs in a terminal-to-terminal pattern but without carrying any passengers on board the train. This inspection operation repeats several times before the start of daily transit service to ensure safety for the public. The post-transit operation cycle, which immediately follows the public transit service cycle, involves dispatch-to-yard operations in which some of the empty trains are moved from a terminal station to a yard for required maintenance work scheduled that night, e.g., B–A and F–G for the line in Fig. 1. The time periods, operational functions, and train trip patterns of the three cycles of the daily operation of an MRT system are summarized in Table 1.

The MRT crew scheduling problem is based on the set of tasks that must be performed within a given timetable for each line, as well as scheduling rules, with the goal of arriving at a minimum number of duties. Before presenting approaches and models to solve the problem, we need to describe the tasks, duties, and scheduling rules in more detail. A task represents the minimum portion of work that can be assigned to a train operator and is derived from timetable and relief points (RP). The timetable, which lists all the work to be assigned to the crew in a given day, contains departure and arrival times of all the trains for each station, including departure and arrival times to terminals and yards. RPs are those stations at which

Fig. 1 An MRT line with yard stations

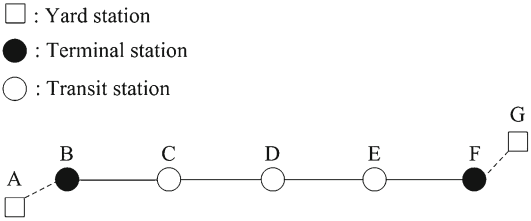


Table 1 Daily operation of MRT trains

	Operational function	Train trip pattern	Approximate time period
Pre-transit cycle	Dispatch-to-work track inspection	Yard-to-terminal terminal-to-terminal	4 am to 6 am
Public transit cycle	Transit service	Terminal-to-terminal	6 am to 12 pm
Post-transit cycle	Dispatch-to-yard	Terminal-to-yard	12 pm to 2 am

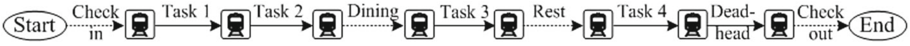


Fig. 2 Composition of a duty for MRT crew scheduling

the crew check-in and check-out for their daily duties and where they can relieve each other, rest, and dine. In general, the longer an MRT line is, the more RPs there are likely to be on that line. While not all MRT stations can serve as RPs, both yards and terminal stations are usually RPs. In terms of transit stations, those in the middle of a long line are more likely to be operated as RPs than others. The related data required to define a task thus include a train number, a departure or origin RP station, a departure time, an arrival or destination RP station, and an arrival time.

A duty consists of a sequence of tasks to be assigned to a single operator or driver for a day's work. On any given day, the tasks assigned to a driver must be separated by breaks, because the driver needs to rest and dine during his or her shift. Moreover, deadhead trips (in which train operators are transported as passengers to another relief point) are an inevitable part of a daily operation that best utilizes the workforce. Accordingly, the full contents of a duty include check-in, tasks, dining, rest, deadhead trips, and check-out. Figure 2 depicts an example of a duty that consists of four tasks, two breaks including one for rest and one for dining, and one deadhead trip. In the figure, train icons denote the RP stations; the non-RP stations, though not shown, are included implicitly between each of the succeeding RP pairs defined by a task or a deadhead trip.

Feasible duties for an MRT crew scheduling problem must satisfy all the operational constraints as well as regulatory and labor union rules. Scheduling rules are generally related to the time and/or location of the tasks involved in a single duty. The most common scheduling rules for an MRT system in general are described below.

(1) Location-related constraints

- *Relief location fit.* Each task must begin at the same RP where the previous task ended, or else a deadhead trip must be assigned to the duty. In a deadhead trip, a train operator is transported as a passenger to another relief point.
- *Duty start-and-end location fit.* The check-in and check-out of each duty must occur at the same RP, or else a deadhead trip must be assigned at the beginning or the end of the duty.

(2) Shift-related constraints

- *Work shift rule.* Due to the long hours of operation, an MRT system usually has multiple work shifts in a day for the crew to be able to cover all the tasks. Every duty to be scheduled must start and finish during the same shift.

- *Number of meals in a shift.* In most cases, the number of meals required in a full-time shift is one. For graveyard and midnight shifts, however, the number is zero because of the long rest breaks already included in those shifts.
- *Meal-time window constraint.* For those shifts requiring a meal, the time for dining is restricted to a certain permissible period of time, and the dining can be assigned only in that specific time window.

(3) Time-related constraints

- *Rest-time limit.* To allow the crew to have a reasonable amount of time to rest, the time of each rest period assigned is restricted by a lower and upper bound.
- *Meal-time limit.* Similar to the rest-time limit, the amount of time allotted for each meal is restricted by a lower and upper bound.
- *Continuous driving time limit.* To prevent overexertion, the continuous driving time (CDT) in each duty cannot exceed an upper bound. Sometimes a lower bound for CDT is also used to avoid inefficient assignment of work.

All the scheduling constraints described above are also called *hard rules*, because they all must be strictly satisfied in an MRT crew scheduling problem. In addition to these hard rules, the management may consider some *soft rules*. Soft rules generally represent issues that are of great concern but are difficult to capture explicitly in the problem formulation. A typical example of a soft rule for many MRT systems is the quest for workload balance. Another example is the attempt to assign all duties in as few shifts as possible. Soft rules, although more complicated than hard rules, can be handled by adding extra constraints or secondary objectives to the problem. While most of the MRT applications in the existing literature deal only with the hard rules, our proposed CP-based approach attempts to tackle both the hard and the soft rules most commonly seen in an MRT system.

Table 2 compares the coverage of scheduling rules consider in our proposed CP model with that considered in other models developed for the MRT crew scheduling problems in the cities of Hong Kong (Chu and Chan 1998), Lisbon of Portugal (Cavique et al. 1999), Singapore (Chew et al. 2001), Busan of Korea (Park and Ryu 2006), and Santiago of Chile (Elizondo et al. 2010). Only ours can cover all the scheduling rules listed in the table. For example, all the models except ours have overlooked the minimum and maximum rest time constraints and the number of meals requested in different shits. Moreover, while our model puts no restrictions on the structure of a duty, all others restrict either the number of tasks, the number of shifts, or that of pieces of work in their models. This comprehensive coverage of scheduling rules suggests that our proposed approach has the potential to be applicable to MRT systems in different cities, not just the single case of Taipei presented in this paper.

2.2 CP-based solution framework

In this paper, we propose a CP-based approach to solve the MRT crew scheduling problem. Our proposed approach consists of two phases: duty generation and duty optimization. In the first phase, the problem of identifying feasible duties that satisfy all the hard rules for scheduling is actually a constraint satisfaction problem, rather than an optimization problem. Thus, based on the input data related to the tasks, shifts, and all the rules, we propose the formulation of a CP model to solve the associated constraint satisfaction problem. Depending on the problem scale, the CP model may or may not be able to generate all the feasible duties for crew scheduling. Nevertheless, in both cases the results of the CP model provide valuable input to the following phase of duty optimization. Note that, with appropriately

Table 2 Comparison of crew scheduling rules considered in various MRT application studies

	Our approach	Chu and Chan (1998)	Cavique et al. (1999)	Chew et al. (2001)	Park and Ryu (2006)	Elizondo et al. (2010)
Application city	Taipei	Hong Kong	Lisbon	Singapore	Busan	Santiago
Timetable coverage	Yes	Yes	Yes	Yes	No	Yes
Location fit						
Relief point	Yes	Yes	Yes	Yes	Yes	Yes
Duty start-and-end	Yes	Yes	Yes	Yes	Yes	Yes
Rest time						
Lower bound	Yes	No	No	No	No	No
Upper bound	Yes	No	No	No	No	No
Number of dining	Yes	No	No	No	No	No
Meal time						
Time window	Yes	No	Yes	Yes	No	No[a]
Lower bound	Yes	Yes	Yes[b]	Yes	No	No
Upper bound	Yes	No	Yes[b]	Yes	No	No
Continuous driving						
Lower bound	Yes	Yes	Yes	No	No	No
Upper bound	Yes	Yes	Yes	Yes	Yes	Yes
Work time balance						
Lower bound	Yes	No	Yes	No	No	No
Upper bound	Yes	No	Yes	No	No	Yes
Range of tasks						
Full consideration	Yes	No[c]	No[c]	No[c]	No[d]	Yes
Work shifts						
Full consideration	Yes	Yes	Yes	Yes	Yes	No[e]

[a] The meal was always assigned to the break period following the maximum continuous driving time
[b] Both bounds were fixed to 1 h
[c] Only two or three pieces of work were included in each duty
[d] This study fixed ten tasks in each duty
[e] Only two full-time shifts were included in a day

added constraints, the CP model can consider soft rules in addition to all the hard rules. Later in our case application, we will show how the soft rule of workload balance can be adopted in our proposed CP model to improve the solution.

In the second phase of duty optimization, we formulate the problem as a SCP to find the minimal cost in terms of a minimal number of duties that cover all the required tasks. However, the solution procedure involved in this phase is not straightforward. Depending on the problem scale, we propose alternative ways to reach the minimum cost. When the CP model in phase one can generate all the feasible duties, we achieve a well-defined SCP in phase two. If the problem scale is not large, one can solve the SCP directly and obtain an exact optimal solution. However, if the SCP is too large to solve, then it can be relaxed as a large-scale LP, and conventional CG techniques can be applied to solve the problem approximately (Desaulniers et al. 2005). On the other hand, if all the feasible duties cannot be generated in reasonable time by the CP model in phase one, we only obtain a partially defined SCP in phase

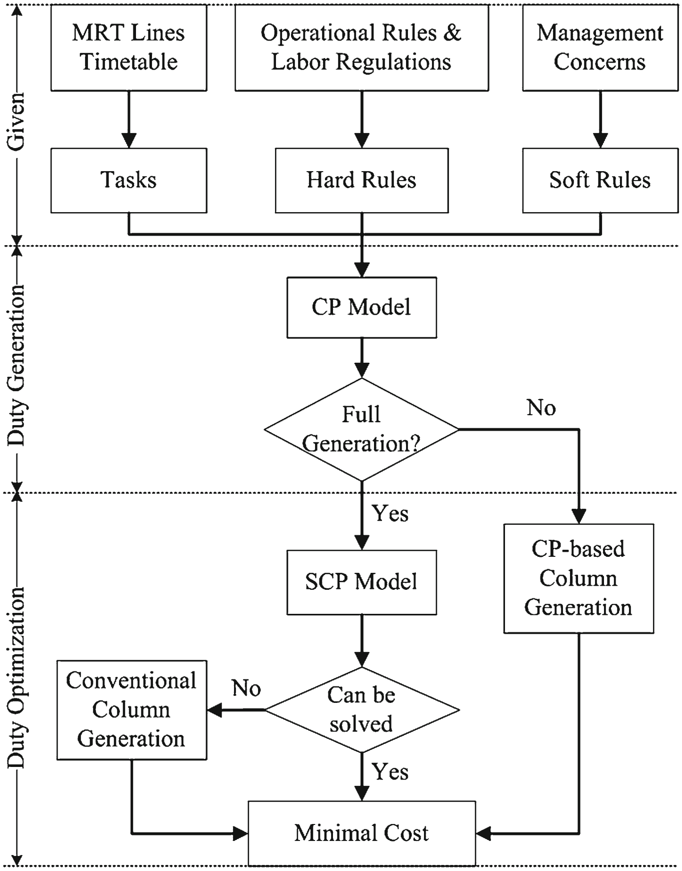


Fig. 3 A solution framework for an MRT crew scheduling problem

two. In this case, both MP-based and CP-based CG procedures are applicable. We propose applying CP-based CG techniques to solve the SCP for duty optimization because it has been found that the CP-based CG is more efficient than the MP-based CG (Yunes et al. 2005).

The overall framework of our proposed CP-based solution approach for an MRT crew scheduling problem is shown in Fig. 3. The CP model plays a critical role in this solution approach. It not only serves as the core for duty generation in the first phase, but also provides an efficient tool to solve the pricing sub-problem for generating columns to the restricted SCP in the CP-based CG procedures for duty optimization. Since we have proposed different ways to solve the problems for different scenarios, this solution approach is applicable to MRT systems of different scales. A successful application to the Taipei MRT is described next.

3 Case background and data description

3.1 Taipei MRT case background

Taipei is the capital city of the Republic of China, i.e., Taiwan. The Taipei MRT is the key public transportation system servicing the city of Taipei and neighboring towns, with a

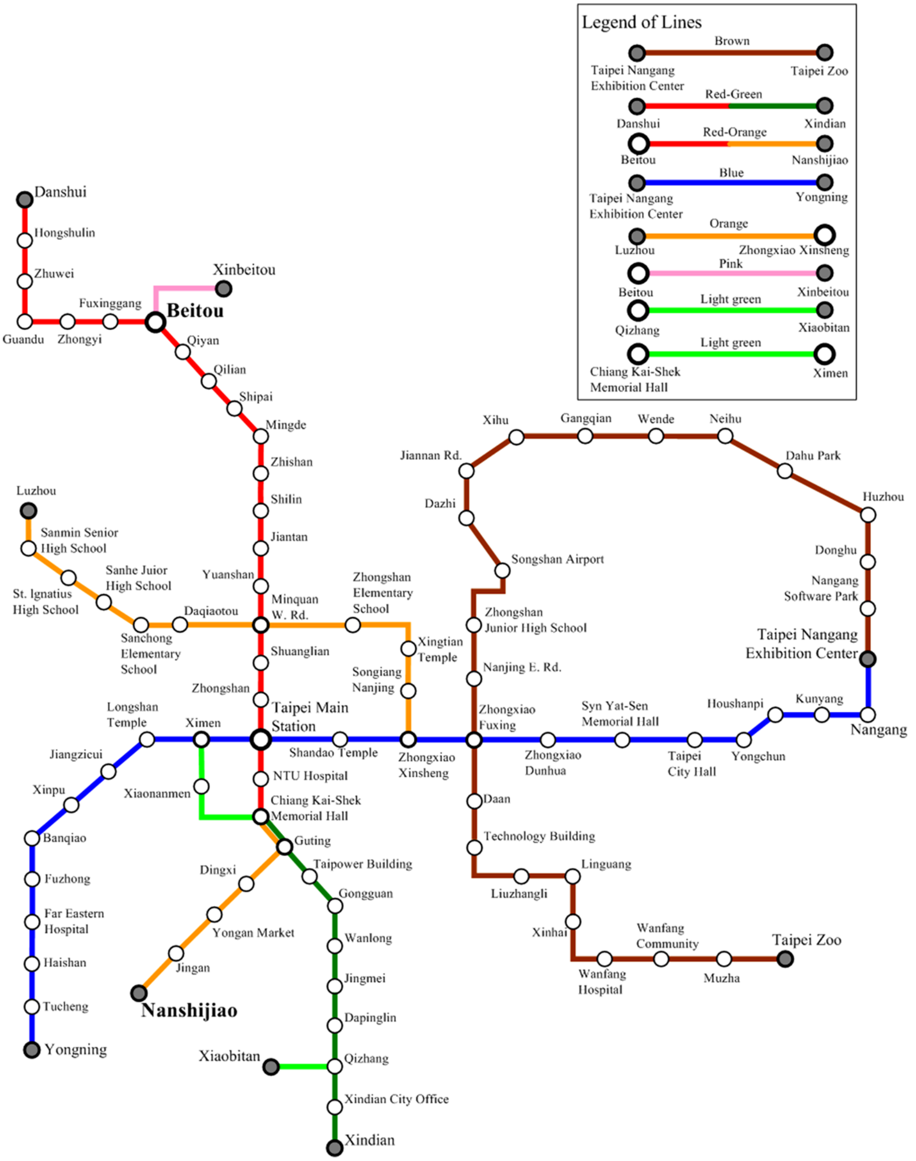


Fig. 4 Taipei MRT network as of 2011

total population of approximately 5.5 million (Taipei City Government 2013). The system is Taiwan's first MRT; the first line, the Muzha line, began operation in 1996. Currently, five main lines and three branch lines are in operation, with 89 stations and a total operational length of approximately 101.9 km. The network of the Taipei MRT is shown in Fig. 4. The five main lines are the Brown line (Taipei Nangang Exhibition Center-Taipei Zoo), the Red–Green line (Danshui–Xindian), the Red–Orange line (Beitou–Nanshijiao), the Blue line (Taipei Nangang Exhibition Center–Yongning), and the Orange line (Luzhou–Zhongxiao Xinsheng); the three branch lines are the Pink line (Beitou–Xinbeitou) and the two Light

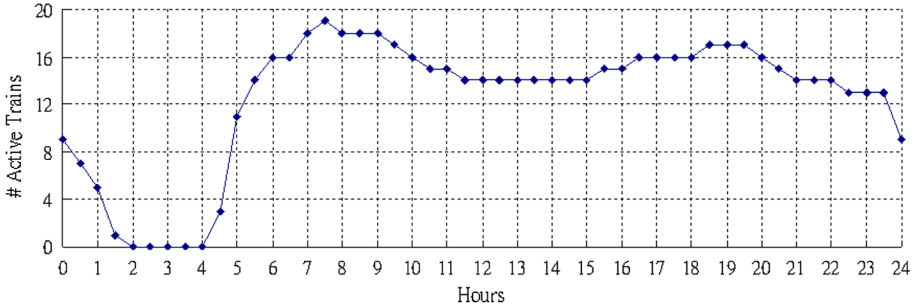


Fig. 5 Number of trains in operation on a typical weekday on the *Red–Orange line*

Green lines (Qizhang–Xiaobitan and the C.K.S. Memorial Hall–Ximen Line). During June 2013, the system carried an average of over 1.6 million passengers per day (Ticketing Center of Station Operations Division 2013). Based on the current operations of the Taipei MRT, the crew schedule is managed on a line-by-line basis such that the crew members assigned to a specific line will not be assigned to other lines. In our case study, we address the crew scheduling problem for the Red–Orange (Beitou–Nanshijiao) line.

For the Red–Orange line, there are 20 stations in operation, and only Beitou and Nanshijiao are RP stations. The yards on the line are Fuxinggang and Zhonghe, which is near Nanshijiao. The service period runs from 0530 to about 2400 every day, and the operational period for trains runs from approximately 0430 to 0200 the next day. The average operational headway is approximately 6 min in peak hours and 8 min in off-peak hours. However, the service headway for transit riders is only about half of the operational headway, because the Red–Orange line overlaps heavily with the Red–Green line, as shown in Fig. 4. For MRT riders boarding at the overlapping stations, the average service headway is approximately 3 min in peak hours, and 4 min in off-peak hours. Figure 5 shows the number of trains in operation on the line for every half-hour period in a typical weekday. Generally, as seen in the figure, there are 13–19 trains in operation for most of the day, from 0530 to 2330. There is a total number of 228 tasks derived from the requirements of daily operation.

3.2 Input data and parameters

In this section, we specify the input data and parameters required by our case study. For convenience, Table 3 gives a summary of the associated notations.

Most of the input data for the CP model are related to tasks or shifts. We use *Task* to denote the set of all the tasks, and $Task = \{0, 1, 2, \ldots, \mathrm{Nt}\}$, where 0 is a dummy task and Nt is the total number of train tasks (Nt = 228). For each task $t \in Task$, the task is defined by five parameters: the train number (Number_t), the origin RP station (Org_t), the departure time (Dept_t), the destination RP station (Des_t), and the arrival time (Arr_t). For each of the train tasks, these parameters are derived from a real operational timetable of the Red–Orange line. On the other hand, for the dummy task, i.e., $t = 0$, we set a null value to its train number and origin and destination stations, i.e., $\mathrm{Number}_0 = \mathrm{Org}_0 = \mathrm{Des}_0 = 0$, and a big number to its departure and arrival times, i.e., $\mathrm{Arr}_0 = \mathrm{Dept}_0 = 9999$, to facilitate our CP formulation.

The management of the Taipei MRT runs the daily operation with 16 work shifts. As defined in Table 4, there are three graveyard shifts (MNA, MNB, and MNC), and 13 eight-hour full-time shifts which include five morning shifts (MA, MB, MC, MD and MG), four afternoon shits (A1, A2, A3, and A4) and four night shifts (NA, NB, NC and ND). For each

Table 3 The summary of notations of parameters and input data sets

Task	The set of all tasks, $Task = \{0, 1, 2, \dots, \mathrm{Nt}\}$
Nt	Total number of train tasks
Number_t	Train number of task t, $\forall t \in Task$
Dept_t	Departure time of task t, $\forall t \in Task$
Arr_t	Arrival time of task t, $\forall t \in Task$
Org_t	Departure station of task t, $\forall t \in Task$
Des_t	Arrival station of task t, $\forall t \in Task$
S	The set of shifts (as defined in Table 4)
Start_s	Start time of shift s, $\forall s \in S$
End_s	End time of shift s, $\forall s \in S$
$\mathrm{dine\_LB}_s$	The start of the meal-time window in shift s, $\forall s \in S$
$\mathrm{dine\_UB}_s$	The end of the meal-time window in shift s, $\forall s \in S$
maxRT_s	Maximal rest time allowed in shift s, $\forall s \in S$
$\mathrm{numDine}_s$	The number of dining required in shift s, $\forall s \in S$
minRT	Minimal rest time
minDT	Minimal meal time
maxDT	Maximal meal time
CIT	Check-in time
COT	Check-out time
maxCDT	Maximal continuous driving time
DHT (i, j)	Deadhead travel time between relief point i and j

shift that requires a meal, the restricted time window for the meal is also given. For instance, when a train driver assigned to work in shift MG, which starts at 0800 and terminates at 1600, he or she must take a meal break between 1000 and 1300.

Other related data for our case study include the following:

- The check-in time (CIT) and check-out time (COT) are each 5 min.
- The rest-time limit: the lower bound (minRT) is 15 min for all shifts and the upper bound (maxRT) is 40 min for regular shifts and 270 min for the graveyard shifts.
- The number of meals allowed (numDine) is 0 for the three graveyard shifts and 1 for all the other shifts listed in Table 4.
- The meal-time limit: the lower bound (minDT) and the upper bound (maxDT) are 30 and 50 min, respectively.
- Continuous driving time limit: the lower bound (minCDT) and the upper bound (maxCDT) are 45 and 180 min, respectively.
- The deadhead travel times between all origin-destination pairs of RP stations are shown in Table 5.

4 Modeling and case application

4.1 Decision variables of the CP model

In our solution approach, the CP model is designed for duty generation. Each solution obtained from the CP model represents a feasible duty, and the duty can be defined as a sequence of tasks in the following way:

Table 4 Work shifts for the Red–Orange line

Shift Code	Start	End	dineLB	dineUB
MA	0700	1500	1000	1300
MB	0730	1530	1000	1300
MC	0630	1430	1000	1300
MD	0530	1330	1000	1300
MG	0800	1600	1000	1300
NA	1430	2230	1600	1900
NB	1500	2300	1600	1900
NC	1400	2200	1600	1900
ND	1530	2330	1600	1900
MNA	2200	0800	NA	NA
MNB	2230	0830	NA	NA
MNC	2130	0730	NA	NA
A1	1100	1900	1000	1300
A2	1100	1900	1600	1900
A3	1130	1930	1000	1300
A4	1130	1930	1600	1900

Table 5 Deadhead travel time (in minutes)

To From	Fuxinggang	Beitou	Nanshijiao	Zhonghe
Fuxinggang	0	30	70	80
Beitou	30	0	40	50
Nanshijiao	70	40	0	10
Zhonghe	80	50	10	0

$$Duty = \{x_1, x_2, \ldots, x_\mathrm{m}\} \tag{1}$$

where x_i is the i-th task of the duty, $i \in M = \{1, 2, \ldots, \mathrm{m}\}$. The domain of these major decision variables of x_is is $Task = \{0, 1, 2, \ldots, \mathrm{Nt}\}$. The total number of tasks in a duty, i.e., the value of m, represents the maximum number of train tasks that can be included in a single duty, and is determined as follows.

$$\mathrm{m} = \max_s \left\lfloor \frac{(\mathrm{End}_s - \mathrm{Start}_s - \mathrm{minDT} - \mathrm{minRT} - \mathrm{CIT} - \mathrm{COT})}{\min\limits_i (\mathrm{Arr}_i - \mathrm{Dept}_i)} \right\rfloor \tag{2}$$

where the numerator is the total time available for work in shift s and the denominator is the time of the shortest task in that shift. According to Eq. (2), we set m = 8 for our case problem. For generality, we would keep the notion of m in the formulation of the CP model. However, most of the feasible duties cannot include as many as m train tasks. When this occurs, we must assign dummy tasks to the duty. For example, if m = 6 for the duty depicted in Fig. 2, then both the fifth and sixth tasks are dummy tasks, i.e., $x_5 = x_6 = 0$, in the solution associated with that duty.

Between two tasks there may be a break for a rest or a meal. The break variable between two adjacent tasks of i and $i+1$, $break(x_i, x_{i+1})$, is a binary variable. When the two tasks are not consecutive and the latter is not a dummy task, the break variable is equal to 1; otherwise, the variable is 0. The break variable is defined as follows:

$$break(x_i, x_{i+1}) = \begin{cases} 1 & \text{if } \mathrm{Arr}_{x_i} \neq \mathrm{Dept}_{x_{i+1}} \wedge x_{i+1} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i = 1, 2, \ldots, \mathrm{m}-1 \tag{3}$$

When a break occurs, it must be designated as taking either a rest or a meal. Accordingly, we need to define two more binary or indicator variables for $dining(x_i, x_{i+1})$ and $rest(x_i, x_{i+1})$ as the following.

$$break(x_i, x_{i+1}) = dining(x_i, x_{i+1}) + rest(x_i, x_{i+1}) \text{ for } i = 1, 2, \ldots, \mathrm{m}-1 \tag{4}$$

In addition to the break variable, we define an intermediate variable, $breaktime(x_i, x_{i+1})$, to trace the duration of the break between x_i and x_{i+1}, as shown in Eq. (5).

$$breaktime(x_i, x_{i+1}) = \begin{cases} \mathrm{Dept}_{x_{i+1}} - \mathrm{Arr}_{x_i} - \mathrm{DHT}(\mathrm{Des}_{x_i},\ \mathrm{Org}_{x_{i+1}}) & \text{if } break(x_i, x_{i+1}) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$
$$\text{for } i = 1, 2, \ldots, \mathrm{m}-1$$

Here, $\mathrm{Dept}_{x_{i+1}}$–Arr_{x_i} is the time gap between the two adjacent tasks, and $\mathrm{DHT}(\mathrm{Des}_{x_i},\ \mathrm{Org}_{x_{i+1}})$ is the deadhead travel time between the two stations Des_{x_i} and $\mathrm{Org}_{x_{i+1}}$. If the two stations are the same, then $\mathrm{DHT}(\mathrm{Des}_{x_i},\ \mathrm{Org}_{x_{i+1}})$ is null, and the break time will simply be the time gap of $\mathrm{Dept}_{x_{i+1}} - \mathrm{Arr}_{x_i}$; otherwise, the deadhead travel time between the two stations must be subtracted from the break time.

Note that Eqs. (3) through (5) apply for $i = 1, 2, \ldots, \mathrm{m}-1$, but not for $i = \mathrm{m}$. This is because no further breaks can follow the very last task x_m in a duty as defined in Eq. (1).

The continuous driving time is another important variable in our CP model. We denote CDT_i as the cumulative CDT variable at the end of task i, and it counts only for the last task m and those tasks that end with a break. In transit scheduling, a piece of work, or a work piece, is a sequence of consecutive tasks without breaks. Therefore, for the last task of each work piece, the CDT variable has to keep track of the continuous driving time for all the tasks involved in that specific work piece. However, the accumulation of driving time starts with the same first task for all work pieces. Double counting in the accumulation process must be avoided. The CDT variables are thus defined as the following:

$$CDT_i = \begin{cases} \sum_{n=1}^{i} (\mathrm{Arr}_{x_n} - \mathrm{Dept}_{x_n}) - \sum_{k=1}^{i-1} CDT_k & \text{if } \mathrm{Arr}_{x_i} \neq \mathrm{Dept}_{x_{i+1}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$
$$\text{for } i = 1, 2, \ldots, \mathrm{m}-1$$

and

$$CDT_\mathrm{m} = \sum_{n=1}^{\mathrm{m}} (\mathrm{Arr}_{x_n} - \mathrm{Dept}_{x_n}) - \sum_{k=1}^{\mathrm{m}-1} CDT_k \tag{7}$$

As an example, there are three work pieces: $(x_1 - x_2)$, (x_3), and (x_4) for the duty shown in Fig. 2; only CDT_2, CDT_3, and CDT_4 have meaningful nonzero values, and the CDT_is for all other tasks are null. Table 6 summarizes the variables and their associated domains defined in our CP model.

As mentioned in Sect. 2.1, the CDT of a feasible duty cannot exceed the upper bound of maxCDT. This can be easily handled by taking the upper bound into the definition of the variable's domain. Specifically, we set maxCDT as the upper bound of the finite domain of

Table 6 The variables and their domains

Variables	Description	Domain
x_i	i-th task in the solution of a duty	$Task = \{0, 1, 2, \ldots, \mathrm{Nt}\}, \forall i \in M$
$break(x_i, x_{i+1})$	indicator of the break between x_i and x_{i+1}	$\{0, 1\}$, for $i = 1, 2, \ldots, \mathrm{m} - 1$
$breaktime(x_i, x_{i+1})$	the time length of the break between x_i and x_{i+1}	$\{0, 1, \ldots, 999\}$, for $i = 1, 2, \ldots, \mathrm{m} - 1$
$dining(x_i, x_{i+1})$	indicator of the dining between x_i and x_{i+1}	$\{0, 1\}$, for $i = 1, 2, \ldots, \mathrm{m} - 1$
$rest(x_i, x_{i+1})$	indicator of the rest between x_i and x_{i+1}	$\{0, 1\}$, for $i = 1, 2, \ldots, \mathrm{m} - 1$
CDT_i	the continuous driving time at the end of task i	$\{0, 1, \ldots, \mathrm{maxCDT}\}, \forall i \in M$

the set variable CDT_i for every $i \in M$. However, the lower bound of the CDT time, i.e., minCDT, cannot be treated similarly because the CDT_i as defined in Eq. (6) is always zero for those tasks not followed by breaks. Accordingly, the minCDT must be considered in the constraints as described in the following section.

4.2 Constraints and CP-based duty generation

Feasible duties must satisfy the scheduling rules described in Sect. 2.1. All those rules, except the timetable coverage constraint which will be considered in the next phase of duty optimization, are covered in the CP model. Before presenting the constraints for meeting the scheduling rules, we first propose Eq. (8) to ensure the appropriate order of the tasks contained in each duty.

$$\mathrm{Dept}_{x_{i+1}} \geq \mathrm{Arr}_{x_i} \quad \text{for } i = 1, 2, \ldots, \mathrm{m} - 1 \tag{8}$$

For simplicity, the constraints formulated in the CP model will also be referred to as equations in this paper.

Considering the location fit of the RP stations, we can trace the train that connects two consecutive tasks at the same station. Thus, this location fit rule can be formulated as:

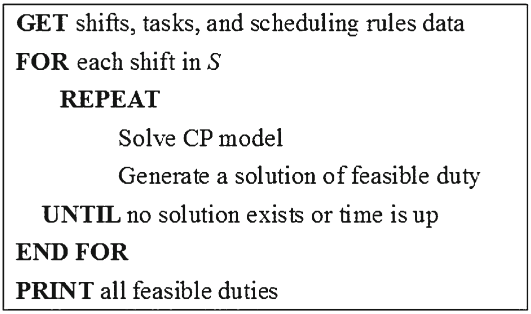
$$\mathrm{Arr}_{x_i} = \mathrm{Dept}_{x_{i+1}} \Rightarrow \mathrm{number}_{\mathrm{x}_i} = \mathrm{number}_{\mathrm{x}_{i+1}} \text{ for } i = 1, 2, \ldots, m - 1 \tag{9}$$

The work shift rule requires that all the tasks in a feasible duty must start and end within a time period of [Start_s + CIT, End_s −COT], which takes into account both the check-in and check-out time for the duty. However, this does not apply for dummy tasks. Therefore, we write two constraints to define the work shift rule as follows.

$$x_i \neq 0 \Rightarrow \mathrm{Dept}_{x_i} \geq \mathrm{Start}_s + \mathrm{CIT} \qquad \forall i \in M \text{ and } s \in S \tag{10}$$

$$x_i \neq 0 \Rightarrow \mathrm{Arr}_{x_i} \leq \mathrm{End}_s - \mathrm{COT} - \mathrm{DHT}(\mathrm{Des}_{x_i},\ \mathrm{Org}_{x_1}) \qquad \forall i \in M \text{ and } s \in S \tag{11}$$

Eqs. (10) and (11) restrict, respectively, the start and end of each non-dummy train task to the right time limit. While the former is clear, the latter is bit complicated because it also serves as a location fit constraint. In addition to the time limit, Eq. (11) also considers whether the end station Des_{x_i} of a non-dummy task, x_i, is the same as the station where the duty starts, i.e., Org_{x_1}. If these two stations are not the same, then we must reserve a deadhead trip of time $\mathrm{DHT}(\mathrm{Des}_{x_i},\ \mathrm{Org}_{x_1})$ for dispatching the driver back to Org_{x_1} to ensure the duty start-and-end location fit as described in Sect. 2.1.



```
GET shifts, tasks, and scheduling rules data
FOR each shift in S
      REPEAT
            Solve CP model
            Generate a solution of feasible duty
    UNTIL no solution exists or time is up
END FOR
PRINT all feasible duties
```

Fig. 6 Duty generation using a CP model

Eqs. (12) and (13) define the time limit constraint for rest and dining respectively. For both types of breaks, the break time must follow its associated time limits. Moreover, Eq. (13) not only constrains the meal time to its bounds of minDT and maxDT, but also stipulates that the meal time must be assigned in the period of [dine_LB, dine_UB].

$$rest(x_i, x_{i+1}) = 1 \Rightarrow \text{minRT} \le breaktime(x_i, x_{i+1}) \le \text{maxRT}_s \quad \text{for } i = 1, 2, \ldots, \text{m} - 1 \text{ and } s \in S \tag{12}$$

$$dining(x_i, x_{i+1}) = 1 \Rightarrow \text{minDT} \le breaktime(x_i, x_{i+1}) \le \text{maxDT} \wedge \text{dine\_LB}_s \le \text{Arr}_{x_i} \le \text{dine\_UB}_s \quad \text{for } i = 1, 2, \ldots, m - 1\ and\ s \in S \tag{13}$$

For the dining breaks, Eq. (14) demands that the number of meals in a shift must follow the pre-set numDine.

$$\sum_{i=1}^{\text{m}-1} dining(x_i,\ x_{i+1}) = \text{numDine}_s \ \ \forall s \in S \tag{14}$$

Finally, for the continuous driving time constraint, we need consider only the lower bound (minCDT) in our constraints, because the upper bound (maxCDT) has been adopted in the domain of the CDT_i variables as defined in Eqs. (6) and (7). And we can easily use the CDT_i variables to define this constraint as the following.

$$\text{Arr}_{x_i} \ne \text{Dept}_{x_{i+1}} \Rightarrow CDT_i \ge \text{minCDT} \ \ \text{for } i = 1, 2, \ldots, \text{m}-1 \tag{15}$$

$$x_\text{m} \ne 0 \Rightarrow CDT_\text{m} \ge \text{minCDT} \tag{16}$$

Our proposed CP model includes both the variables and constraints defined in this and previous sections. For duty generation in our case problem, we use the ILOG OPL Studio 3.7.1 (ILOG 2003) to solve the CP model. For each work shift, the feasible duties are generated one by one by solving the CP model; the procedure repeats itself until no more solutions or the feasible duties can be found, or a pre-determined computer time has expired. The whole duty generation process stops when the CP model has been solved for all the shifts. Figure 6 summarizes the duty generation process based on the CP model we proposed.

4.3 SCP model and duty optimization

Given all the feasible duties generated from the CP model, the final stage of the solution process is to find the minimum number of duties that will satisfy all the MRT scheduling rules. This duty optimization problem is formulated as the following SCP model.

$$\text{Min} \sum_{j \in FD} Duty_j \tag{17}$$

subject to

$$\sum_{j \in FD} a_{ij} Duty_j \geq 1 \quad \forall i \in \mathit{Task} \tag{18}$$

$$Duty_j \in \{0,\ 1\} \quad \forall j \in \mathit{FD} \tag{19}$$

where

Task is the set of tasks as given in the CP model, and

FD is the set of all feasible duties obtained from the CP model,

$$a_{ij} = \begin{cases} 1 & \text{if task } i \text{ is included in duty } j \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in Task,\ j \in \mathit{FD}$$

As shown in the objective (17), the SCP model assumes the same unity cost for all the duties and seeks to minimize the total number of duties to be selected. The constraint (18) requires that each task i be covered by at least one of the duties in the solution. The constraint (19) simply defines the binary variable of the problem.

When the CP model in the previous stage is capable of generating the full set of *FD*, we can either solve the SCP directly or tackle it with conventional column generation techniques. On the other hand, when the CP model cannot generate all the feasible duties, we propose the adoption of a CP-based column generation approach, as mentioned in Sect. 2.2. A generic version of CP-based CG solution procedures proceeds as follows. First, we solve the restricted master SCP as an LP and obtain the reduced cost π for each task to be covered. Second, we add two constraints to the CP model and use the modified CP model as the engine for generating new columns of feasible duties. Each new duty must (1) be different from the existing duties in *FD*, and (2) have a negative reduced cost, i.e., $rc = 1 - \sum_{i \in \mathrm{m}} \pi_{x_i} < 0$. Third, we include the new columns in the restricted master SCP and repeat the process until no more duties can be generated by the modified CP model. Finally, we have to convert the solutions into binary variables to obtain an approximate solution. More sophisticated CP-based CG techniques can be found in Silva (2001) and Yunes et al. (2005).

5 Results

5.1 Primary results of the case problem

We coded the CP and the SCP models in OPL (Hentenryck 1999), and also coded the whole solution process in the OPL scripting language to integrate the implementation of both models on the ILOG OPL Studio 3.7.1 platform (ILOG 2003). The whole solution process was carried out on an AMD 64 3000+ personal computer with 1 GB RAM in a Windows XP professional environment. Our proposed CP model did generate all the feasible duties in the first phase of duty generation, so there was no need for us to invoke a CP-based column generation procedure to solve our case problem. Moreover, due to the moderate scale of the SCP in our case, i.e., 56,562 feasible duties in total, the problem in the second phase for duty optimization was able to be solved to its integer solution on the ILOG OPL Studio 3.7.1 platform. This implies that our approach can solve the problem to its exact solution for those MRT lines

Table 7 Results of the case problem of the Red–Orange line in the Taipei MRT

	Total solution time	# Duties scheduled	# Shifts adopted	Total work time, TWT		
				Avg. (min)	SD (min)	CV (%)
Taipei MRT	1 week	58	14	320.7	67.4	21.0
CP-based approach	∼5 min (307”)	55	11	351.1	38.2	10.9

which have a scale similar to the Red–Orange line in Taipei. And only when applying this CP-based approach to an MRT line whose scale is larger than the Red–Orange line, the CP-based column generation procedure may be required to facilitate the solution process. Table 7 summarizes the results of our case application.

In the Taipei MRT, the crew schedule is generated manually by an experienced person. While the manual results provided by the Taipei MRT required 58 duties for weekday operation, our approach generated a solution of 55 duties. This generates approximately 5.2 % savings in labor costs for the Red–Orange line of the Taipei MRT. Our results have other merits beyond cost savings, as well. Our results can satisfy all the scheduling rules used by the Taipei MRT, while the manual results cannot fully meet the rest and dining-time requirements. Moreover, our results can make daily operations easier to manage, with 11 work shifts rather than the current 14 shifts.

The computer implementation of our solution is efficient. While the manual process usually takes a week, our approach takes only 307 s, i.e., approximately 5 min, to generate all the results. Specifically, the CP model implementation took 113 s to generate all the 56,562 feasible duties in the 16 shifts considered. And the optimal solution of the SCP took 194 s. As MRT crew scheduling is a planning-oriented problem, the computer efficiency of our proposed models should be adequate for real-world applications.

As mentioned earlier in Sect. 2.1, workload balance in general is a big concern in the management of an MRT system. In practice, workload balance does not consider the duties assigned in graveyard shifts because of the different nature of the work involved. For each of the duties in a regular or non-graveyard shift in the Taipei MRT, the workload is defined by the *total work time* (TWT), which includes the time of both train tasks and deadhead trips assigned in the duty. There are 14 graveyard-shift duties in both the manual results provided by the Taipei MRT and our results, so the numbers of regular-shift duties are 44 and 41, respectively. For such duties, the average value and standard deviation of the TWT are illustrated in Table 7. To evaluate the workload balance, the coefficient of variation (CV), which is the mean divided by the standard deviation, of the TWT is a reasonable performance measure. As shown in the table, the CV of the TWT reduces almost to half, from 21.0 to 10.9 %, with our improvement. It implies that even without soft rules, the CP-based approach can be effective in improving the workload balance. In fact, our CP model, described next, can further improve the workload balance.

5.2 Extended results with soft rule considerations

Our proposed CP model can be easily extended to consider the soft rule of seeking workload balance. The main idea is to expand the CP model with several additional constraints to govern the TWT in a more reasonable range so as to achieve a better workload balance. First, we define the TWT of a duty by the following three constraints (20), (21), and (22), and add them into our CP model.

Table 8 Alternative optimum solutions with soft-rule considerations

[minTWT, maxTWT]	# Duties scheduled	# Shifts adopted	CV of TWT (%)	# Feasible duties	Total CPU time (s) (CP, SCP)
None	55	11	10.9	56562	307 (113, 194)
[300, 400]	55	10	7.5	37101	285.85 (99.55, 186.3)
[320, 400]	55	11	6.7	23742	159.8 (86.8, 73)
[340, 400]	55	11	4.8	16159	129.53 (80.2, 49.33)
[300, 380]	55	10	5.3	36208	259.95 (98.9, 161.05)

$$TWT = \sum_{j=1}^{lst} (\mathrm{Arr}_{x_j} - \mathrm{Dept}_{x_j}) + \sum_{j=1}^{lst-1} \mathrm{DHT}(\mathrm{Des}_{x_j},\ \mathrm{Org}_{x_{j+1}}) + \mathrm{DHT}(\mathrm{Des}_{x_{lst}},\ \mathrm{Org}_{x_1}) \tag{20}$$

where *lst* denotes the last train task of the duty and is determined by

$$x_i \neq 0 \wedge x_{i+1} = 0 \Rightarrow lst = i \quad \text{for } i = 1, 2, \ldots, m-1 \tag{21}$$

$$x_\mathrm{m} \neq 0 \Rightarrow lst = \mathrm{m} \tag{22}$$

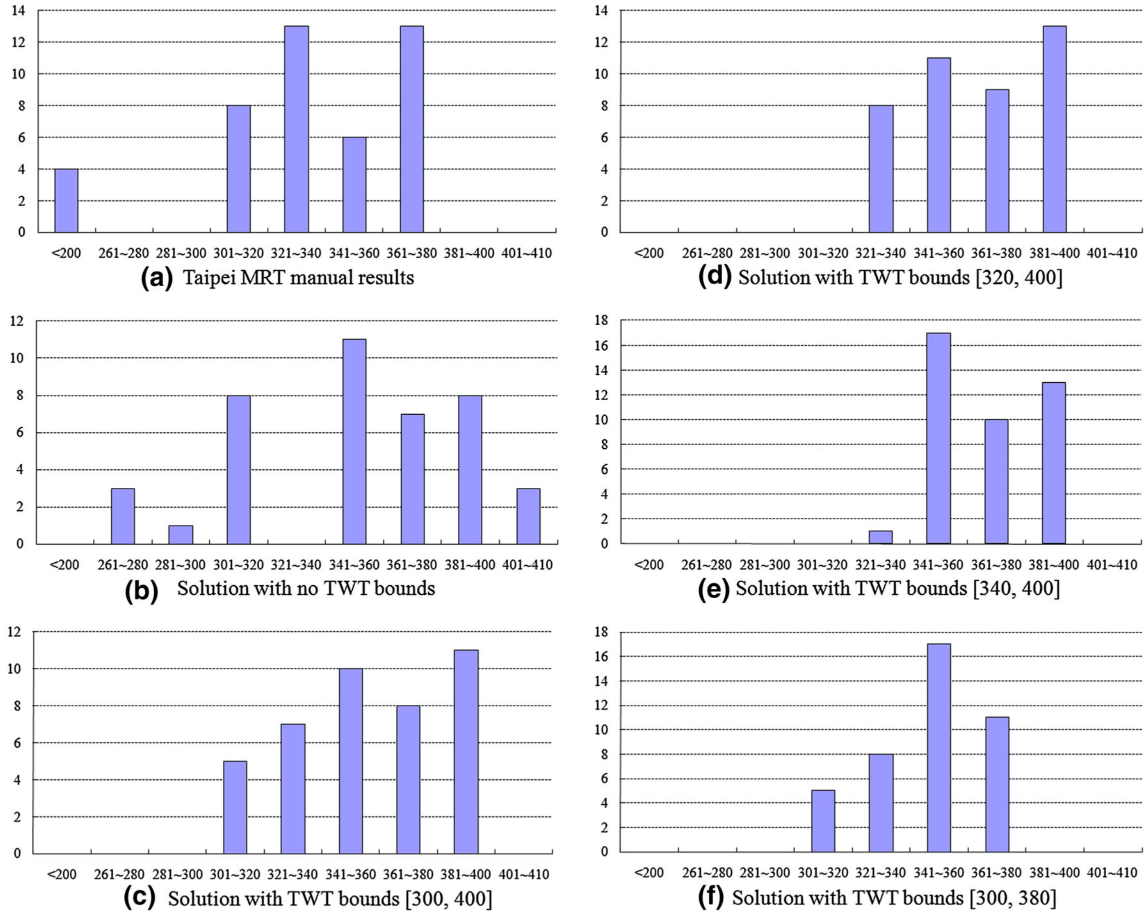
Moreover, we add in the fourth constraint (23) to restrict the TWT in the range of [minTWT, maxTWT], where minTWT and maxTWT represent the bounds of the time range.

$$\mathrm{minTWT}_s \leq TWT \leq \mathrm{maxTWT} \quad \text{for } s \notin \{\mathrm{MNA, MNB, MNC}\} \tag{23}$$

Based on the results of the average and standard deviation of the TWT of the first solution shown in Table 7, we have considered three values, 300, 320, and 340, for minTWT, and two values, 380 and 400, for maxTWT. Using these parameters, we run the expanded CP model with different settings for the TWT bounds and then obtain the optimal solution from the subsequent SCP model. Finally, we found four alternative minimum-cost solutions with the same minimum cost of 55 duties but with different degrees of workload balance. Table 8 summarizes these results. As compared with the first solution without TWT bounds, the CV of the TWT again can be significantly reduced, from 10.9 % to as low as 4.8 %.

However, the slight difference in the CV values among the alternative solutions is insufficient to identify the most desirable solution. To overcome this obstacle, we plot the bar charts of the work time distribution for the base case, with the first solution and the four alternative solutions depicted in Fig. 7 for comparison. This figure, together with Table 8, can provide valuable information for a decision maker who has in mind the soft-rule considerations of workload balance and minimum work shifts. One can easily observe from Fig. 7e that, although this solution has the lowest CV value of 4.8 %, its TWT distribution is not as evenly distributed as that of the solutions depicted in Fig. 7c, d and f. On the other hand, the TWT distribution of the solution with TWT bounds of [300, 380], as shown in Fig. 7f, looks quite attractive. It exhibits a narrow and rather symmetric bell-shaped distribution of TWT that has not been seen in other comparable solutions. Moreover, for work shifts, this solution requires a minimum number of 10 shifts to work with.

Note that, as shown in Table 8, the solution time for solving for alternative solutions with TWT bounds is even less than that for the first solution. The solution time is reduced significantly from 307 s to as low as 130 s. This shows another merit of a CP model, i.e., with properly designed additional constraints the model can be solved more efficiently. Similar



(a) Taipei MRT manual results

(b) Solution with no TWT bounds

(c) Solution with TWT bounds [300, 400]

(d) Solution with TWT bounds [320, 400]

(e) Solution with TWT bounds [340, 400]

(f) Solution with TWT bounds [300, 380]

Fig. 7 Work time distribution of alternative solutions

results can be found in Goumopoulos and Housos (2004) in the context of their applications to airline crew scheduling problems.

6 Conclusions

In this paper, we proposed a CP-based approach to solve the crew scheduling problem for MRT operations. Existing studies on MRT crew scheduling applications consider the problem as primarily an optimization problem and solve the problem using heuristic or metaheuristic methods. Our paper seems to be one of the first in the literature that applies CP models to MRT crew scheduling applications.

We applied the CP-based approach to tackle the crew scheduling problem for the Red–Orange line of the Taipei MRT and found encouraging results. As compared with the manual results generated by the Taipei MRT, which barely meet rest and dining-time constraints, our results can reduce the number of duties required in the daily operation from 58 to 55, which represents an approximate savings of 5.2 % in labor costs. Moreover, the number of work shifts required reduces from 14 to 11, and the workload balance, in terms of the coefficient of variation of the total work time distribution, improves significantly, from 21 to 11 %.

The proposed CP model also allows the flexibility to accommodate soft-rule considerations. In this paper, we demonstrated how to incorporate the workload balance consideration into the CP-based solution process and easily generate four alternative optimum solutions with better workload balance performance. With the same minimum cost of 55 duties, the coefficient of variation of the total work time distribution further improves from 11 % to approximately 5 %.

In our study, we have considered a comprehensive coverage of the scheduling rules most commonly seen in practice. Thus, the proposed CP-based approach and models may have the potential for application to MRT systems in cities other than Taipei.

References

Apt, K. R. (2003). *Pricinples of constraint programming*. Cambridge: Cambridge University Press.

Barnhart, C., Cohn, A., Johnson, E., Klabjan, D., Nemhauser, G., & Vance, P. (2003). Airline crew scheduling. In R. W. Hall (Ed.), *Handbook of transportation science* (Vol. 56, pp. 517–560). New York: Springer.

Brailsford, S. C., Potts, C. N., & Smith, B. M. (1999). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, *119*(3), 557–581.

Caprara, A., Fischetti, M., Toth, P., Vigo, D., & Guida, P. L. (1997). Algorithms for railway crew management. *Mathematical Programming*, *79*(1–3), 125–141.

Cavique, I., Rego, C., & Themido, I. (1999). Subgraph ejection chains and tabu search for the crew scheduling problem. *Journal of the Operational Research Society*, *50*(6), 608–616.

Chew, K. L., Pang, J., Liu, Q. Z., Ou, J. H., & Teo, C. P. (2001). An optimization based approach to the train operator scheduling problem at Singapore MRT. *Annals of Operations Research*, *108*(1), 111–122.

Chu, S. C. K., & Chan, E. C. H. (1998). Crew scheduling of light rail transit in Hong Kong: From modeling to implementation. *Computers Operations Research*, *25*(11), 887–894.

de Silva, A. (2001). Combining constraint programming and linear programming on an example of bus driver scheduling. *Annals of Operations Research*, *108*(1), 277–291.

Desaulniers, G., Desrosiers, J., & Solomon, M. M. (2005). *Column generation*. New York: Springer.

Elizondo, R., Parada, V., Pradenas, L., & Artigues, C. (2010). An evolutionary and constructive approach to a crew scheduling problem in underground passenger transport. *Journal of Heuristics*, *16*(4), 575–591.

Ernst, A. T., 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.

Fahle, T., Junker, U., Karisch, S. E., Kohl, N., Sellmann, M., & Vaaben, B. (2002). Constraint programming based column generation for crew assignment. *Journal of Heuristics*, *30*(1), 59–81.

Ftulis, S. G., Giordano, M., Pluss, J. J., & Vota, R. J. (1998). Rule-based constraints programming: Application to crew assignment. *Expert Systems With Applications*, *15*(1), 77–85.

Gabteni, S., & Grönkvist, M. (2009). Combining column generation and constraint programming to solve the tail assignment problem. *Annals of Operations Research*, *171*(1), 61–76.

Goumopoulos, C., & Housos, E. (2004). Efficient trip generation with a rule modeling system for crew scheduling problems. *Journal of Systems and Software*, *69*(1–2), 43–56.

Grönkvist, M. (2006). Accelerating column generation for aircraft scheduling using constraint propagation. *Computers and Operations Research*, *33*(10), 2918–2934.

Hooker, J. N. (2006). Operations research methods in constraint programming. In F. Rossi, et al. (Eds.), *Handbook of constraint programming* (pp. 527–570). Amsterdam: Elsevier.

ILOG. (2003). ILOG OPL Studio 3.7 Studio user's manual. SA: ILOG.

Lustig, I. J., & Puget, J. F. (2001). Program does not equal program: Constraint programming and its relationship to mathematical programming. *Interfaces*, *31*(6), 29–53.

Mackworth, A. K. (1977). Consistency in networks with relations. *Artificial Intelligence*, *8*(1), 99–118.

Park, T., & Ryu, K. R. (2006). Crew pairing optimization by a genetic algorithm with unexpressed genes. *Journal of Intelligent Manufacturing*, *17*(4), 375–383.

Puget, J. F. (1995). A comparison between constraint programming and integer programming. In Proceedings of the Conference on Applied Mathematical Programming and Modelling (APMOD95). Uxbridge: Brunel University

Taipei City Government, Department of Civil Affairs. (2013). *Statistics in population and each district households in Taipei city July-2013*. Retrieved August 2, 2013, from http://www.ca.taipei.gov.tw/public/Attachment/3829432175.xls

Ticketing Center of Station Operations Division. (2013). *Transport volume statistics: June-2013. Resource document*. Taipei Rapid Transit Corporation. Retrieved July 5, 2013 from http://web.trtc.com.tw/RidershipCounts/E/10206e.htm

Van Hentenryck, P. (1999). *The OPL optimization programming language*. Combridge, MA: MIT Press.

Van Hentenryck, P. (2002). Constraint and integer programming in OPL. *Journal on Computing*, *14*(4), 345–372.

Wilson, N. H. M. (1999). *Computer-aided transit scheduling* (Vol. 471). Berlin: Springer.

Yunes, T. H., Moura, A. V., & de Souza, C. C. (2005). Hybrid column generation approaches for urban transit crew management problems. *Transportation Science*, *39*(2), 273–288.