Ann Oper Res
DOI 10.1007/s10479-014-1648-9

Patrol scheduling in urban rail network

Hoong Chuin Lau · Zhi Yuan · Aldy Gunawan

Abstract This paper presents the problem of scheduling security teams to patrol a mass rapid transit rail network of a large urban city. The main objective of patrol scheduling is to deploy security teams to stations of the network at varying time periods subject to rostering as well as security-related constraints. We present several mathematical programming models for different variants of this problem. To generate randomized schedules on a regular basis, we propose injecting randomness by varying the start time and break time for each team as well as varying the visit frequency and visit time for each station according to their reported vulnerability. Finally, we present results for the case of Singapore mass rapid transit rail network and synthetic instances.

Keywords Patrol scheduling problem · Preferences · Mass rapid transit rail network · Mathematical programming

1 Introduction

Personnel scheduling and rostering is concerned with the process of constructing optimized work timetables for staff in order to satisfy the demand for the organization. Ernst et al. (2004) provide a review of staff scheduling and rostering in specific applications areas. Some are concerned with rostering within a physical premise such as hospitals, and examples of such problems include nurse rostering (e.g. Petrovic and Berghe 2008) and physician scheduling (e.g. Gunawan and Lau 2013). A more challenging problem involves rostering of personnel

H. C. Lau (✉) · Z. Yuan · A. Gunawan
School of Information Systems, Singapore Management University,
Singapore, Singapore
e-mail: hclau@smu.edu.sg

Z. Yuan
e-mail: zhiyuan@smu.edu.sg

A. Gunawan
e-mail: aldygunawan@smu.edu.sg

Published online: 12 June 2014

that are required to move from one geographical location to another as they are discharged from their duties, such as airline crew scheduling (e.g. Maenhout and Vanhoucke 2010) and train crew scheduling (e.g. Chu and Chan 1998).

In this paper, we are concerned with the planning problem of assigning security teams to patrol a public transportation network (such as subways) of a large urban city. This is termed the patrol scheduling problem. This problem is motivated by increasing need for protecting major public facilities (such as urban transport systems) in response to global threats. Clearly, the problem of protecting critical infrastructures against terrorist threats is a very complex problem involving many dimensions (e.g. cybersecurity, video surveillance and many others)—many of such systems are in place in the real-world today. The paper is concerned with only one dimension, the aspect of deploying limited security manpower resources for preventing or reacting to attacks. In order to enforce such security, security personnel or teams are typically deployed to patrol the various stations in the network throughout the day. Unlike standard employee rostering which follows prescribed patterns, patrol activities should ideally exhibit randomness so as to hedge against adversarial observations. It goes without saying that in view of limited manpower resources, it is necessary to maximize the impact of patrolling duties through solving the problem optimally.

The main objective of this paper is to develop exact models that deploy security teams to stations of the network at varying time periods while ensuring rostering and other security-related constraints. We also consider aspects of randomness to hedge against adversarial observations. To our knowledge, this study is one of very few attempts to solve the patrol scheduling problem on a mass rapid transit rail network.

The remaining part of the paper is organized as follows. We first provide a brief literature review. We then give a detailed description of our patrol scheduling problem in Sect. 3. We provide several mathematical programming models that solve different variants of the problem, followed by a randomized strategy which allows the planner to generate solutions based on randomized start time and break time for each team as well as the number of visits required for each station. The next section is dedicated to the experimental analysis of the model on the Singapore MRT Rail System, as well as on randomly generated problem instances. Finally, we provide some concluding perspectives and directions for future research.

2 Literature review

Crew rostering in public transport systems is an active area of research. An example of a rail transport scheduling problem is Chu and Chan (1998), who studied the problem of crew scheduling for the Hong Kong Light Rail Transit. The complex schedule construction is decomposed into separate solution stages by network and heuristic algorithms. They reported that the entire crew schedule can be constructed iteratively in less than an hour, which is better than the manual allocation. Although optimality cannot be claimed, the feasibility of the solution was ensured, which can still be further improved manually.

A more recent work of Elizondo et al. (2010) considers the problem of conductors duty generation in the Santiago Metro System. With regard to operational and labor conditions, the goal is to use the lowest possible number of conductors and minimize total idle time between trips. They solved the problem using a constructive hybrid approach which takes advantage of the benefits offered by evolutionary methods. Their hybrid method produced solutions with the minimum number of duties in six of the ten problems solved.

On patrol scheduling, the major purpose is to ensure the safety of the commuters and to discourage those who might commit crimes (Rosenshine 1970). The patrol scheduling method developed in this seminal paper is based on the assumption of randomness, so that the arrival patterns of the security patrol to a particular station could not be predicted. On the other hand, the irregularity of patrol schedules would increase the awareness of the commuters that patrol is taking place. The routes were determined by solving a linear programming problem while the random arrival patterns on each arc were generated by choosing exponential inter-dispatch times along the generated routes.

Sinuany-Stern and Teomi (1986) studied the problem of scheduling security guards in a large organization in Israel. The problem was formulated as a multi-objective problem and solved by a simpler heuristic intuitive algorithm. Taylor and Huxley (1989) considered the problem of assigning police officer shifts so that under cover is minimized. The optimization-based decision support system was developed and implemented in the Police Patrol Scheduling System at the San Francisco Police Department. Sharma et al. (2007) proposed an optimal deployment of police patrol cars for the department of traffic police on the metropolitan city, Delhi (Central). A goal programming model was designed to determine the number of patrol cars to have on duty per shift and road segment.

The application of game theory to patrol scheduling took center stage in recent AI research. Jain et al. (2010), Ordóñez et al. (2013) highlighted the importance of police and security agencies to protect transportation networks, such as buses, trains and traces the recent development of game-theoretic models to assist security forces in randomizing their patrols and their deployment by assuming intelligent adversary responses to security measures. More precisely, this group of researchers sought to mitigate the ability of adversaries to exploit patterns by using randomized strategies derived from solving Bayesian Stackelberg games. Stackelberg games are a bilevel model that account for the ability of an adversary to gather information about the defense strategy before planning an attack (Basar and Olsder 1995).

As an illustration, Tsai et al. (2009) modeled the strategic security allocation problem as a Stackelberg game and developed the Intelligent Randomization In Scheduling (IRIS) system for deploying the Federal Air Marshals (FAMs) onboard U.S. commercial flights. The generic mathematical formulation is described as the set covering model where the set of schedules of security forces are pre-determined. Jiang et al. (2012) presented an approach to generate fare-inspection strategies in urban transit systems using a Stackelberg game. The problem is to deploy security personnel randomly to inspect passenger tickets. The real problem from the Los Angeles Metro Rail System was formulated and solved as an LP relaxation with a maximum-revenue patrol strategy. The solutions obtained seem to effectively deter fare evasion and ensure high levels of revenue.

Beyond scheduling/allocation of security teams, other researchers have focused on different aspects of security management. For example, the effectiveness and efficiency of the airport access control has become a major concern, involving screening people, baggage, in order to detect and prevent entry of unauthorized personnel and items. Babu et al. (2006) developed a model in order to classify passengers into different groups, the fractions of passengers and the assignment of check stations for each group under the assumption of constant passenger threat probability. The model minimizes the false alarm probability for the given parameters of the access control systems. The data used is from general knowledge of different types of check stations that exist at many of the larger US airports. The authors highlighted a possible direction for future work by considering models that captures the effect of joint response of more than one station.

Zhang and Brown (2013) studied the importance of police patrols in public service by responding to incidents, deterring and preventing crimes. The study concerns on how to design police patrol districts. This problem is reduced to the graph-partition problem with the constraints of contiguity and compactness. The algorithm proposed generates some police

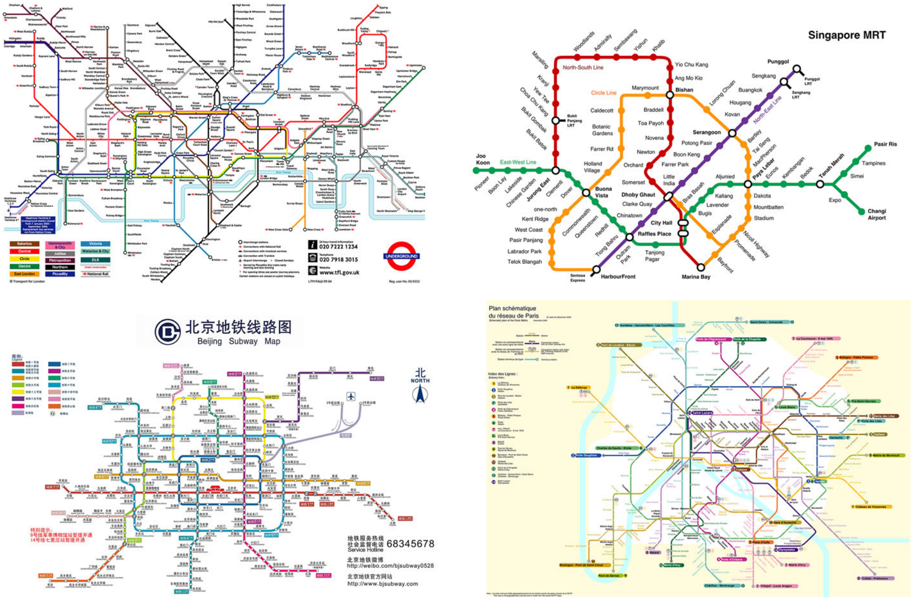

Fig. 1 Fig. 1 Examples of urban rail networks in metropolises such as London (Source: https://www.tfl.gov.uk/modes/tube), Singapore (Source: http://www.smrt.com.sg), Beijing (Source: http://www.bjsubway.com), and Paris (Source: http://www.parismetro.com)

patrol districting plans. A certain number of promising plans are then evaluated. The last step is to perform evaluation using an agent-based simulation that provides high fidelity measures of performance. This study includes a case study from the Charlottesville, VA, USA police department. Simulation results show improvement on both average response time and variation of the workload. Curtin et al. (2010) presents a new method for determining optimal police patrol areas using maximal covering and backup covering location models. This study addresses the lack of objective quantitative methods for police area design in the literature. The Police Patrol Area Covering (PPAC) model integrates geographic information systems (GIS) with linear programming optimization to generate and display alternative optimal solutions. The proposed approach was tested on the Police geography of Dallas, TX. The improvement in terms of the increased number of incidents within an acceptable service distance was reported.

3 Problem definition

This paper is concerned with scheduling security resources to patrol a mass rapid transit rail network in a large urban city. Figure 1 shows the subway systems of London, Singapore, Beijing, and Paris. A common feature of the networks is that each network consists of many lines of different topologies linked by hub (interchange) stations.

We define the *patrol scheduling* problem as follows. We are given a number of security teams responsible for the patrolling task. We assume the time horizon to be a single working day divided into time periods. In this paper, we assume each time period to be 1 h, and a shift is defined as a consecutive eight time periods (hours). Each team within its shift is rostered to patrol / visit a subset of stations in the network. We assume each station patrol or visit takes one period, and each shift is made up of six visits plus two breaks, totaling eight periods long.

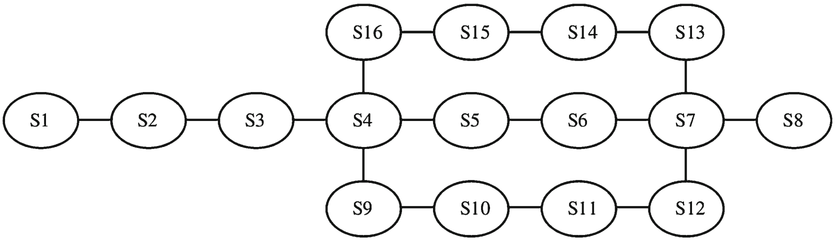

Fig. 2 Example of the mass rapid transit rail network

Table 1 Example of patrol scheduling problem

Time period	1	2	3	4	5	6	7	8
Team 1	S1	S3	Break	S5	S6	Break	S7	S13
Team 2	S10	S12	Break	S14	S16	Break	S4	S2

As shown in Fig. 2 as illustration, this mass rapid transit rail network consists of two different lines. There are 16 stations in total, where Station 4 (S4) and Station 7 (S7) are interchange stations. Assuming there are two teams starting at the time, Table 1 represents one possible patrol schedule for both teams. Team 1 has to visit S1, S3, S5, S6, S7 and S13 consecutively while Team 2 has to visit S10, S12, S14, S16, S4 and S2 consecutively.

One of the optimization objectives in the patrol scheduling problem is to minimize the total cost (which usually translates to total distance traveled). The travel distance is minimized such that as much time as possible in each patrol team's shift is spent on patrolling rather than movement from one station to another. An alternative travel distance objective includes minimizing the maximum total distance traveled by any team, as considered in Varakantham et al. (2013), which can be easily modeled with minor modification to our proposed model. The distance traveled between any two stations is assumed known. It usually refers to the time needed to travel by train. Note that a travel between two stations on different lines, for example, from station S3 to S16 in Fig. 2, requires transit at station S4, and the transit time should also be considered.

In this paper, we propose three different mathematical models for three variants of the patrol scheduling problem. In the first model termed Base introduced in Sect. 4.1, the optimization objective is to minimize the total travel distance, which satisfies the following constraints:

- Each team visits one and only one station in a working (i.e. non-break) period.
- At most one team can visit a particular station at a particular time period.
- Each team may only visit a particular station at most once during its shift duty.
- Consecutiveness constraints: describes whether a pair of stations can be visited consecutively (i.e. one after another). Although in most of the cases, any two stations can be connected, the reason for this constraint is to avoid assigning a team to two consecutive visiting stations that are far away, so that most of the time in each working period is spent on patrolling rather than traveling.
- Each station has a minimum and maximum number of visits per day.

In this model, the starting time of each team's shift, and the two break time periods during its shift, are treated as input (i.e. assume they are determined by the planner). Besides, the number of required minimum and maximum visits of each station are also assumed given as input by the planner.

In the second model termed BI-OBJ in Sect. 4.2, we introduce a second objective (in addition to the travel distance minimization) of maximizing the coverage of expected loss of human lives. This roughly amounts to covering the most populous stations at the busiest time, so that as many people as possible may be saved if an attack is launched. The expected number of passengers present at a certain station during a certain time period is assumed known, from which the second objective may be derived. This can be simply extracted from the electronic card reader at the entrance or exit of each station. In this paper, we make use of the data obtained from the Land Transport Authority of Singapore, which maintains such a database from the Singapore mass rapid transit network. We further assume that the probability of a certain station being attacked at a certain time period is known. This adversary probability model can be constructed based on, for example, geographical density, criminology analysis, or intelligence source, etc. Then the expected loss (of human lives) at a station at a certain period can be estimated as a product of the population at this time and its probability of being attacked. In other words, the more passengers present and the higher the chance of being attacked, the higher the utility value for patrolling. In the study of the BI-OBJ model, we also free the planner from determining the minimum or maximum visits per station, i.e. by not enforcing the last constraint listed above, but instead allow the optimization algorithm to decide the optimum visit frequency for each station. In fact, even more than the frequency, the BI-OBJ model can determine the optimal time to visit a certain station, making the schedule time-dependent, which is not the case in the BASE model. A time-dependent schedule is useful for covering the most crowded hours of each station, since usually the number of passengers at a station depends heavily on time. For example, some business districts, such as Raffles Place or Tanjong Pagar in Singapore, may expect a really rush hour during the morning and late afternoon periods due to the regular office starting and finishing time there; while some shopping and recreation centers, such as Orchard and City Hall, may not have a lot of crowd during the early morning, but may become crowded from noon till late evening.

In the third model termed VAR introduced in Sect. 4.3, we further free the human planner from the decision of specifying the actual starting time and break time of each team's shift, and leave it to be decided by the optimization algorithm. This model makes use of the bi-objective model described above, and the constraints for handling the starting time and break time include:

- The total number of working periods for each team must be satisfied.
- Break constraints: each team is given a number of breaks per shift; no breaks can occur at the starting time or the ending time of a shift; and no two breaks can occur at two consecutive periods.

All the mathematical models mentioned above are deterministic, with the objective of generating *one* optimal patrol schedule based on some security or operational criteria. These models are built based on the assumption that the probability distribution of which stations are under attack at a certain time period is known or can be estimated. This is usually the case for unplanned spontaneous attacks. In this article, we also consider a different problem setting, where the patrol schedule is required to be generated recurrently, for example, on a day-to-day basis. Furthermore, we assume the adversaries are intelligent enough to be able to observe the daily patrol schedules. In such case, the objective of generating a patrol schedule is not only to maximize coverage and minimize distance, the generated schedules should also be as random as possible to make it difficult for adversaries to predict. In this article, we investigate the idea of using one of the deterministic models above, and randomize the input data. Different randomization strategies, including randomizing the starting time, break times of each team, the visit frequency and visit time of each station are discussed in more details in Sect. 6.

4 Mathematical programming models

In this section, we present mathematical programming models to solve three variants of the patrol scheduling problem.

4.1 Base model for minimizing travel distance

The goal is to assign a set of patrol teams I to a set of stations J at a set of time periods K per day. In our first model, the minimum number of visits required for a station $j \in J$ is given as V_j^{min}, as well as the maximum number of visits allowed for each station is given as V_j^{max}. The shift of a team $i \in I$ starts at a fixed time period $T_i^{start} \in K$ and ends at $T_i^{end} \in K$, with a set of breaks at time periods $B_i \subset [T_i^{start}, T_i^{end}] \subseteq K$. We also define the set of working periods $W_i \in K$ to be $W_i = [T_i^{start}, T_i^{end}] \backslash B_i$ for each patrol team $i \in I$, and W_i^h $(1 \leq h \leq |W_i|)$ denotes the h-th working period of the team i. Given the distance matrix $D_{j,l}$ for each pair of stations $j, l \in J$, one of the objectives is to minimize the total travel distance of all teams, so that most working time is devoted to patrolling rather than traveling from one station to another. To this end, a consecutive matrix $C_{j,l}$, indicating whether station $l \in J$ can be visited immediately after visiting $j \in J$, is also used to restrict long-distance travels. We use binary decision variables $x_{i,j,k}$ to indicate whether a team $i \in I$ patrols a station $j \in J$ at a time period $k \in K$. Another binary decision variable $y_{i,j,l}$ is used with $y_{i,j,l} = 1$ if a team i travels from station j to station l. The value of variable $y_{i,j,l}$ cannot be greater than 1, since any team cannot visit the same station twice per day. An integer linear programming model Base can be set up as follows:

Base $(I,\ J,\ K,\ W,\ V^{min},\ V^{max},\ T^{start},\ T^{end},\ D,\ C)$:

$$\min \quad d \tag{1}$$

$$\sum_{j \in J} x_{i,j,k} = 1, \qquad \forall i \in I; k \in W_i \tag{2}$$

$$\sum_{k \in K \backslash W}^{j \in J} x_{i,j,k} = 0, \qquad \forall i \tag{3}$$

$$\sum_{i \in I, k \in K} x_{i,j,k} \geq V_j^{min}, \qquad \forall j \tag{4}$$

$$\sum_{i \in I, k \in K} x_{i,j,k} \leq V_j^{max}, \qquad \forall j \tag{5}$$

$$\sum_{i \in I} x_{i,j,k} \leq 1, \qquad \forall j, k \tag{6}$$

$$\sum_{k \in K} x_{i,j,k} \leq 1, \qquad \forall i, j \tag{7}$$

$$y_{i,j,l} \geq x_{i,j,W_i^h} + x_{i,l,W_i^{h+1}} - 1, \qquad \forall i \in I; j, l \in J; h \in W_i \tag{8}$$

$$y_{i,j,l} \leq C_{j,l}, \qquad \forall i \in I; j, l \in J \tag{9}$$

$$d = \sum_{i \in I; j,l \in J: l \neq j} D_{j,l} \cdot y_{i,j,l}$$

$$x_{i,j,k}, y_{i,j,l} \in \{0, 1\}, \qquad \forall i \in I; j, l \in J; k \in K \tag{10}$$

Equation (2) constrains that a team visits exactly one station at a working period, and visits no stations outside the working periods according to (3); (4) ensures the minimum number of visits per station is satisfied, while (5) constrains the maximum number of visits allowed for each station; (6) enforces that each station at a time period should not be patrolled by more than one team; in addition, each team can only visit a station at most once during its shift according (7); the flow consistency constraint (8) defines the connectivity variable y by setting $y_{i,j,l}$ to 1 if a team i patrols station j and l consecutively; then the connectivity variable y is used to ensure the consecutive constraint in (9) and to compute the total distance traveled by all teams in (10). An alternative distance objective is to minimize the maximum total distance traveled by any team. This can be easily modeled by changing the Eq. (10) to the following:

$$d \geq \sum_{j,l \in J: l \neq j} D_{j,l} \cdot y_{i,j,l}, \qquad \forall i. \tag{10'}$$

The optimal solution of the Base model defined above yields an optimal patrol schedule. However, this model poses great computational challenge to the CPLEX solver: it fails to converge within reasonable computation time even with the smallest instance. One of the reasons is that the lower bound computed by the linear programming relaxation of flow consistency constraint (8) is not tight enough. The formulation can be substantially tightened by adding the following problem-specific valid inequalities:

$$\sum_{l \in J \setminus \{j\}} y_{i,j,l} = \sum_{k \in W \setminus \{T_i^{end}\}} x_{i,j,k}, \qquad \forall i, j \tag{11}$$

$$\sum_{l \in J \setminus \{j\}} y_{i,l,j} = \sum_{k \in W \setminus \{T_i^{start}\}} x_{i,j,k}, \qquad \forall i, j \tag{12}$$

where constraint (11) states that if a team i patrols a station j when their shift does not end, then team i will have a trip starting from j; similarly, constraint (12) states that team i will have a trip ending at j if they patrol station j at a non-starting period. These two valid inequalities have been found experimentally to speed up the solution process of CPLEX by two orders of magnitude.

4.2 Bi-objective model for maximizing coverage

Next, we consider a bi-objective model, where a two-fold objective is considered: the minimization of the total distance traveled by all patrol teams as defined by Eqs. (1) and (10) in the Base model; and also the maximization of the coverage of expected loss. The expected loss can be obtained as follows. We consider an adversary model which launches an attack according to a probability distribution ($P_{j,k}$ denotes the probability that a station $j \in J$ is attacked at certain time period $k \in K$). We also assume we know the expected number of passengers present at a certain station j during a certain time period k, denoted $N_{j,k}$. Then the expected loss (of human lives) $L_{j,k}$ of station j at time period k if not covered (i.e. if not patrolled by any team) can be computed as $L_{j,k} = P_{j,k} \cdot N_{j,k}$.

Hence, a bi-objective model can be set up as follows: on one hand, we should maximize the *coverage* of expected loss denoted by u; on the other end, as in model Base, we minimize the total travel distance d of all patrol teams. We extend the model BASE by modifying its objective in Eq. (1) to the following:

BI-OBJ ($I,\ J,\ K,\ W,\ V^{min},\ V^{max},\ T^{start},\ T^{end},\ D,\ C,\ L,\ \lambda$):

$$\min \quad -\lambda \cdot u + d \tag{13}$$

subject to constraints (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12) and

$$u = \sum_{i \in I, j \in J, k \in K} L_{j,k} \cdot x_{i,j,k} \tag{14}$$

where the parameter λ scales between both objectives u and d. In our experiments, we will also try to remove the minimum visit constraint (4) and maximum visit constraint (5), and let the optimization process automatically determine the visit frequency and visit time for each station that maximizes the coverage of expected loss, instead of given manually.

4.3 Variable starting time and break time

In the previous two models, the starting time and the break time for each patrol team is pre-determined. Here, we explore the option of leaving the starting time and break time as a variable to be determined by the optimization algorithm. Our objective is to investigate the aspects of schedule quality and computational scalability.

To this end, the working periods W_i for each team i is not given a priori, instead, given as input is the number of working period N_i^{work} which satisfies

$$\sum_{j \in J, k \in K} x_{i,j,k} = N_i^{work}, \qquad \forall i. \tag{15}$$

and the constraint (2) is modified as:

$$\sum_{j \in J} x_{i,j,k} \leq 1, \qquad \forall i \in I; k \in K. \tag{2'}$$

also given is the number of breaks N_i^{break} for each team $i \in I$. $N_i^{shift} = N_i^{work} + N_i^{break}$ denotes its total shift length. We introduce binary decision variables $t_{i,k}^{start}$ and $t_{i,k}^{end}$, with $t_{i,k}^{start} = 1$ if team i starts patrolling at time period k, and $t_{i,k}^{end} = 1$ if team i ends patrolling at time period k. They take value 0 otherwise. Then

$$\sum_{k \in K} t_{i,k}^{start} = \sum_{k \in K} t_{i,k}^{end} = 1, \qquad \forall i \tag{16}$$

$$t_{i,k}^{start} = t_{i,k+N_i^{shift}-1}^{end}, \qquad \forall i, k \tag{17}$$

Equation (16) restricts only one starting time and one ending time for each team, and each shift ends $N^{shift} - 1$ periods of time after it starts according to (17). We further introduce binary decision variables $b_{i,k}$ with $b_{i,k} = 1$ if team i takes a break at period k, and $b_{i,k} = 0$ otherwise. Then the following constraints can be imposed:

$$\sum_{k \in K} b_{i,k} = N_i^{break}, \qquad \forall i \tag{18}$$

$$b_{i,k} + b_{i,k+1} \leq 1, \qquad \forall i, k \tag{19}$$

$$b_{i,k} \leq 1 - t_{i,k}^{start}, \qquad \forall i, k \tag{20}$$

$$b_{i,k} \leq 1 - t_{i,k}^{end}, \qquad \forall i, k \tag{21}$$

where (18) ensures the number of breaks, (19) guarantees no two breaks occur consecutively, (20) and (21) forbids a break to occur at the starting period or the end period, respectively. Then the time period consistency constraint can be stated as follows:

$$\sum_{j\in J} x_{i,j,k} + t^{start}_{i,k+1} + b_{i,k} - b_{i,k+1} - t^{end}_{i,k} = \sum_{j\in J} x_{i,j,k+1}, \qquad \forall i, k. \tag{22}$$

which guarantees that the working period and the break period of each team only occurs between their starting time and ending time. We can further tighten the linear programming formulation above by introducing the following problem-specific valid inequalities, which ensures each team not to work or take a break before its starting period:

$$\sum_{j\in J} x_{i,j,k'} \leq 1 - t^{start}_{i,k}, \qquad \forall i \in I; k, k' \in K : k' < k$$

$$b_{i,k'} \leq 1 - t^{start}_{i,k}, \qquad \forall i \in I; k, k' \in K : k' < k,$$

and these two equations above can be even further tightened by merging the working period and break period as follows:

$$\sum_{j\in J} x_{i,j,k'} + b_{i,k'} \leq 1 - t^{start}_{i,k}, \qquad \forall i \in I; k, k' \in K : k' < k. \tag{23}$$

Similarly, the valid inequality for no-work and no-break after the ending period can also be added:

$$\sum_{j\in J} x_{i,j,k'} + b_{i,k'} \leq 1 - t^{end}_{i,k}, \qquad \forall i \in I; k, k' \in K : k' > k \tag{24}$$

The flow consistency constraint (8) cannot be directly applied here, since the break time is unknown. To this end, an auxiliary binary variable z is introduced as $z_{i,j,k} = 1$ if a team i is at station j at time k, be it patrolling or taking a break. Variable z can be defined by x as follows:

$$z_{i,j,k} \geq x_{i,j,k}, \qquad \forall i, j, k \tag{25}$$

$$z_{i,j,k} \geq x_{i,j,k} + b_{i,k+1} - 1, \qquad \forall i, j, k, \tag{26}$$

where (25) and (26) defines the location of working period and break period, respectively. Although constraining variable z by a lower bound in (25) and (26) already suffice for its definition, since the minimization objective function will try to minimize its value. However, we can still provide an upper bound for it as a problem-specific valid inequality, by restricting that team i is at station j at period k only if either team i works at this station at this period, or they are taking a break at k:

$$z_{i,j,k} \leq x_{i,j,k} + b_{i,k}, \qquad \forall i, j, k$$

which can be made even tighter as follows:

$$\sum_{j\in J} (z_{i,j,k} - x_{i,j,k}) \leq b_{i,k}, \qquad \forall i, k. \tag{27}$$

With the auxiliary variable z, the flow consistency constraint can be reformulated by replacing x in (8) by z:

$$y_{i,j,l} \geq z_{i,j,k} + z_{i,l,k+1} - 1, \qquad \forall i \in I; j, l \in J : j \neq l; k \in K. \tag{8'}$$

and the valid inequality (11) and (12) for constraining the out-flow and in-flow of y can be formulated respectively as follows:

$$\sum_{l \in J \setminus \{j\}} y_{i,j,l} \geq \sum_{k \in K} (z_{i,j,k} - t_{i,k}^{end} - b_{i,k+1}), \qquad \forall i, j \tag{11$'$}$$

$$\sum_{l \in J \setminus \{j\}} y_{i,l,j} \geq \sum_{k \in K} (z_{i,j,k+1} - t_{i,k+1}^{start} - b_{i,k+1}), \qquad \forall i, j. \tag{12$'$}$$

Note that the out-flow equality in (11) becomes an inequality in (11′) due to the variable starting time and break time. We can further tighten the formulation by providing an upper bound for variable y by adding

$$\sum_{l \in J \setminus \{j\}} y_{i,j,l} \leq \sum_{k \in K} z_{i,j,k}, \qquad \forall i, j \tag{28}$$

$$\sum_{l \in J \setminus \{j\}} y_{i,l,j} \leq \sum_{k \in K} z_{i,j,k+1}, \qquad \forall i, j, \tag{29}$$

for the out-flow and in-flow of y, respectively, stating that a trip is traveled from, or to, station j by team i only if team i was located at station j. Then the integer linear programming model VAR for handling variable starting time and break time can be stated as follows:

VAR ($I, J, K, D, C, L, N^{work}, N^{break}$):

minimize (13)

subject to (6), (7), (10), (14), (15), (2′), (16), (17), (18), (19), (20), (21), (22), (23), (24), (25), (26), (27), (8′), (11′), (12′), (28), (29).

$$x_{i,j,k}, y_{i,j,l}, z_{i,j,k}, t_{i,k}, b_{i,k} \in \{0, 1\}, \qquad \forall i \in I; j, l \in J; k \in K$$

5 Experimental results

In this section, we present the experimental results together with our evaluation based on the proposed mathematical programming models presented in the previous section. All experiments that we report in this section were run on a 3.07 GHz Intel (R) Xeon (R) CPU with 128 GB of RAM under the Microsoft Windows XP Operating System. The mathematical programming models were solved by the CPLEX 12.4 solver engine. We first describe the experimental setup, followed by experimental results.

5.1 Experimental setup

In this paper, the Singapore rail network was chosen as a real-world case study (Fig. 3). In addition, two different random instances, `Random1` illustrated in Fig. 4 and `Random2` illustrated in Fig. 5, were also generated with multiple lines and different topology.

The entire Singapore network is a large-scale problem with 79 stations, nine of which are interchange stations where multiple lines meet. For a more fine-grained study of computational scalability of our proposed mathematical models, a number of smaller problem instances are extracted from the whole network.

We first decompose the whole network into four instances, each of which represents a single line. There are four different lines in our case study, namely the East–West Line (`EW`), North–South Line (`NS`), North–East Line (`NE`) and Circle Line (`CC`). Each single-line instance

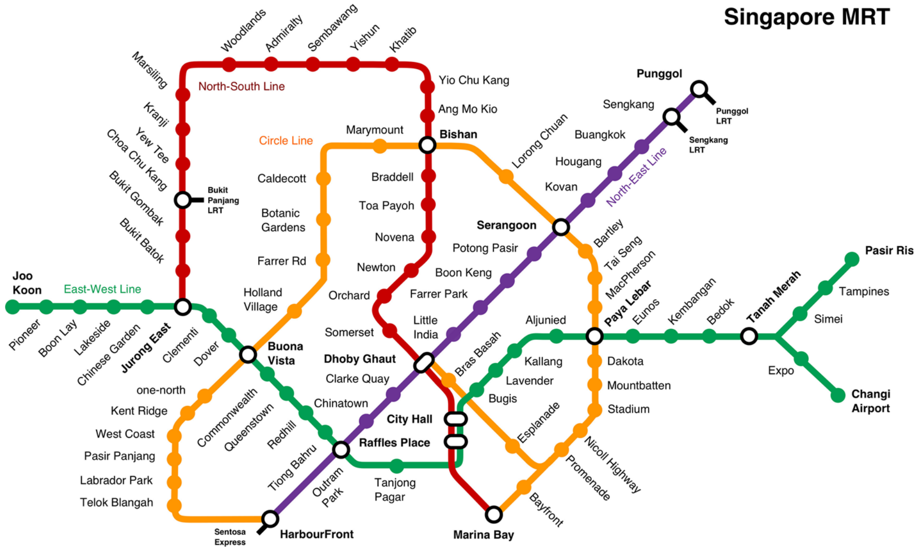

Fig. 3 Singapore MRT map (source: http://www.smrt.com.sg/Trains/NetworkMap.aspx)

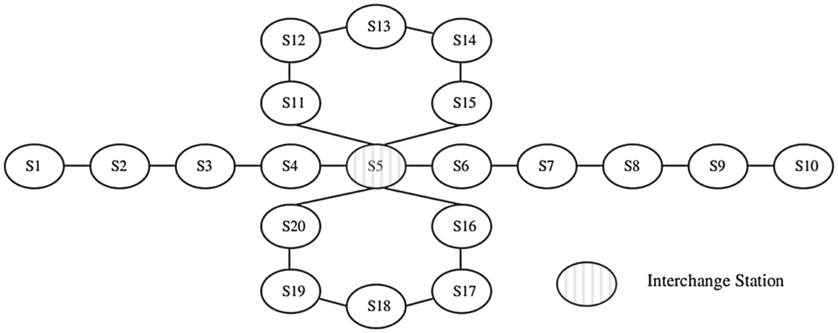

Fig. 4 Random1 station map

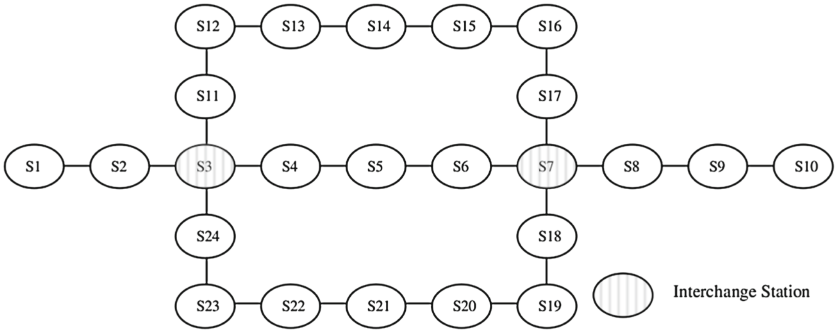

Fig. 5 Random2 station map

is composed of a number of stations ranging from 16 to 31. The details of each line and the station names can be found in http://www.smrt.com.sg. Furthermore, we created two more instances with a combination of multiple lines: `EN` with 53 stations (including 3 interchange stations) by combining the two most populous lines `EW` and `NS`, and `ENN` with 67 stations (including 5 interchange stations) by combining three lines `EW`, `NS`, and `NE`.

For simplicity reason, the distance traveled between two stations is computed as the smallest number of stations passed (since there may be more than one path from one station to another). Taking the `Random2` illustrated in Fig. 5 for example, the distance between S1 and S4 is 3 stations, and the distance between S14 to S22 is 7 stations. Furthermore, we also impose an additional penalty for traveling between two stations on different lines. For example, the distance measure from S2 to S12 becomes 3 stations plus Δ, where Δ is the penalty value for changing lines. In our study, we set Δ to a large number, e.g. 10 stations, for the purpose of minimizing unnecessary changes between different lines during each travel. This is necessary especially when the transit at certain large interchange stations may take a long time.

In the BASE model and the BI-OBJ model, the number of minimum required visits V_j^{min} and maximum allowed visits V_j^{max} for each station $j \in J$ needs to be pre-determined. In our experiments, V_j^{min} is set to one for all stations, and V_j^{max} is set to two, i.e. each station is visited at least once and at most twice per day. This setting is chosen for its simplicity, although different values of V_j^{min} and V_j^{max} can be set for different stations in our model, for example, the most populous or most vulnerable stations may need patrolling more often than the others.

In order to ensure the feasibility of the BASE model, the number of teams is set as:

$$|I| = \lceil \max \left\{ |J|, \sum_{j \in J} V_j^{min} \right\} /6 \rceil \tag{30}$$

since it is assumed that each team patrols six stations per day in an eight hours' shift with two non-consecutive hours' break. In the BASE model and the BI-OBJ model that require a fixed shift starting time and break time as input, the shift starting time T_i^{start} for each team $i \in I$ is assigned as evenly as possible to cover all the time periods K of the day, and we assume that the break periods for all teams occur at the third and the sixth period of an eight-period shift. However, other input values for the number of working periods and break periods can also be set in our models.

Table 2 summarizes the details of all the instances, including the number of stations on each line ($|J|$), the number of interchange stations, the number of allotted patrol teams $|I|$ defined by Eq. 30, the number of time periods per day ($|K|$), and the number of working periods $|W|$ and break periods $|B|$ per team per day.

5.2 Results of BASE model

5.2.1 Results of `Random1` *and* `Random2` *instances*

For the random instances `Random1` and `Random2`, the minimum and maximum number of visits V_j^{min} and V_j^{max} are set similarly as in the real-world case studies described in Sect. 5.1, except that three stations in `Random1`, S1, S5 and S10, are required to be visited twice a day, while S1, S3, S7 and S10 in `Random2` are required to be visited twice as well. In addition, the entire time horizon of 16 periods are covered by two different types of shift starting from 1 or 9 respectively. Both `Random1` and `Random2` can be optimally solved by the CPLEX

Table 2 Characteristics of problem instances

Instance name	$\|I\|$	$\|J\|$	#interchanges	$\|K\|$	$\|W\|$	$\|B\|$
Random instances						
Random1	4	20	1	16	6	2
Random2	5	24	2	16	6	2
Real-world case studies in Singapore MRT network						
NE	3	16	0	20	6	2
CC	3	16	0	20	6	2
NS	5	25	0	20	6	2
EW	6	31	0	20	6	2
EN	9	53	3	20	6	2
ENN	12	67	5	20	6	2
All	14	79	9	20	6	2

$|I|$ denotes the number of security teams; $|J|$ shows the number of stations, from which the number of interchange stations is also listed; $|K|$ presents the total number of time periods per day considered, during which each team works $|W|$ time periods and takes $|B|$ breaks per shift

Table 3 Optimal patrol schedule of Random1, total travel distance 28

I	Time periods															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	S1	S2	B	S3	S7	B	S8	S10								
2	S17	S18	B	S19	S20	B	S5	S16								
3									S5	S15	B	S14	S13	B	S12	S11
4									S10	S9	B	S6	S4	B	S3	S1

Table 4 Optimal patrol schedule of Random2, total travel distance 27

I	Time periods															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	S6	S5	B	S4	S3	B	S2	S1								
2	S10	S8	B	S7	S17	B	S16	S15								
3									S14	S13	B	S12	S11	B	S3	S1
4									S19	S20	B	S21	S22	B	S23	S24
5									S10	S9	B	S8	S7	B	S18	S19

12.4 solver engine in 172 and 39.9 s, respectively. The following Tables 3 and 4 summarize the optimal schedules of Random1 and Random2, respectively. The break period is marked as "B". It is observed that all constraints, including the minimum and maximum number of visits are complied, and no teams require to change to another line for both instances.[1] This optimal solution provides us the minimum total distance traveled for all teams, which is 28 for Random1 and 27 for Random2.

[1] Note that the team 2 in Table 4 serves two lines: the horizontal line (S10 S8 S7) and the circle line (S7 S17 S16 S15). However, this situation is not counted as changing lines, since both the trip from S8 to S7 and the trip from S7 to S17 are traveled on only one line.

Table 5 Computational result of BASE model on Singapore MRT network

Instance	NE	CC	NS	EW	EN	ENN	All
Runtime (s)	0.84	0.37	2.14	6.66	2,079	22,025	37,752
d^*	15	15	25	30	45	60	71

5.2.2 *Results of Singapore network*

Next, the BASE model is deployed to solved the seven problem instances extracted from the Singapore MRT network. The details of the seven instances are presented in Table 2. The computational results are listed in Table 5, including the computation time taken by CPLEX and the optimal total distance d^*. In general the independent lines are solved very quickly to optimality, ranging from 0.37 s for CC line with 16 stations and 3 teams, to 6.66 s for EW line with 31 stations and 6 teams. The instances of multiple lines brings in great computational challenge to the CPLEX solver, and the computation time quickly increases from a few seconds in the single-line instances to more than half an hour for the EN instance to more than 10 h for the whole network. The first feasible solution of the All instance only appears after around 2 h with a duality gap of 76.3 %. Since the minimum number of visits for all stations is set to one, the optimal solution of the BASE model (i.e. the optimal total travel distance d^* in Table 5) usually appears to be quite regular: its objective value is always five times the number of teams. Since each team makes only five travels during each shift with six working periods, it seems that the optimal schedule in these instances is for each patrol team to always travel to a neighboring station at each working period. Obviously, this regularity will no longer hold if the visit frequencies are different for different stations. In fact, this regularity is already violated when the entire network is solved, where the minimum total distance is 71 in a total of 70 travels.

5.3 Results of BI-OBJ model

In this section, we present the computational results of the BI-OBJ model. One additional input to this model is the population data for each station at each time period $N_{j,k}$. This dataset was acquired from the Singapore Land Transport Authority, which releases public transportation data to participating research units upon request. In essence, this dataset contains the time-stamps of the taps of the electronic fare cards at the entrance and exit gantries of all the stations. Using this dataset, we could approximate the total number of people who are present at each station at each time period. Another important input to the model for this case study is the probability distribution of being attacked ($P_{j,k}$). Without prior knowledge, in this paper, we set the probability of being attacked $P_{j,k}$ to be uniformly equal for each station at each time. Other probability models may also be considered, for example, set the probability to be proportional to the population of the region surrounding the station, or based on some game theoretical approaches (Varakantham et al. 2013). In this paper, we focus on our primary goal—which is to test the computational scalability of our proposed models.

We present three different scenarios for using BI-OBJ model:

BI-OBJ-1: the original model as presented in Sect. 4.2, and the input parameters V_j^{min} and V_j^{max} are set to 1 and 2 respectively for all stations, i.e. each station is required to be visited at least once, and at most twice, as is also used in the BASE model;

BI-OBJ-2: the BI-OBJ model without the minimum-visit constraint (4), which can be derived by simply setting V_j^{min} to 0 for all stations;

Bi-Obj-3: the Bi-Obj model without both the minimum-visit constraint (4) and maximum-visit constraint (5), which can be derived by setting V_j^{max} to a large value for all stations.

The three scenarios are experimented on seven real-world instances extracted from the Singapore MRT network.[2]

Table 6 shows the computational results of the seven instances computed by each of the three scenarios of the Bi-Obj model. For each instance, we consider two extreme cases:

coverage-first case ($cover^*$), which regards the coverage objective to be dominantly higher than the distance objective by assigning a very large value to λ in the objective function (13);

distance-first case ($dist^*$), on the contrary, sets λ to a tiny value.

These two cases above correspond to the primary consideration during heightened security and normal patrol operations, respectively.

We examine both the computational scalability and the solution quality of the Bi-Obj model in both cases with different scenarios on the seven real-world instances. The computation time for each instance is listed in the column marked by `sec`. It is shown that the whole network can be solved to optimality or near-optimality by CPLEX within a reasonable runtime. In general, the coverage-first case appears to be computationally less difficult than the distance-first case, even the whole network can be solved to optimality within 40 min in the coverage-first case. In the distance-first case, removing visit frequency constraints seems to reduce the model difficulty noticeably: Bi-Obj-1 solves the whole network to a 0.0000240 % gap after running CPLEX for 24 h; Bi-Obj-2 which removes the minimum-visit constraint solves it to 0.0001 % within 2 h; while Bi-Obj-3 removing further the maximum-visit constraint solves it optimally within 10 min. Each independent line can be solved to optimality in less than 15 s, and it appears that the instances with combination of multiple lines, such as `EN` and `ENN`, increases computational difficulty dramatically by a factor of ten to hundred.

The optimal solutions, including coverage u^c and total distance d^c of the coverage-first case, and u^d and d^d for the distance-first case, are listed in columns 3, 4, 7, and 8 of Table 6, respectively. In general, the coverage-first case usually covers around 10–25 % more expected loss than the distance-first case, as shown in the first of the Δu columns ($\frac{u_i^c - u_i^d}{u_i^d}, i = 1, 2, 3$). The comparison of the optimal coverage of expected loss in two cases by three scenarios is also summarized in Fig. 6. On the other hand, the average travel distance per trip in the coverage-first case $\overline{d^c}$ is usually two to four times as in the distance-first case $\overline{d^d}$. Note that in the distance-first case, the average trip distance $\overline{d^d}$ is usually one station, which means each team only moves to the neighboring station at each working period. Similar travel patterns can be also found in Sect. 5.2.2. One exception that $\overline{d^d}$ is not exactly one is when solving the whole network by Bi-Obj-1 model, where the optimal total distance of 70 trips is found to be 71 due to the visit frequency constraints. In the coverage-first case, $\overline{d^c}$ seems to increase noticeably on instances with multiple lines, such as `EN`, `ENN`, and `All`. Note also although each trip is not allowed to traveled over four stations according to the consecutiveness constraint, some of the average trip distance $\overline{d^c}$ is above four in some instances of multiple lines, since there exists many trips that change between different lines, each of which is penalized by 10.

Comparing the optimal schedules computed by the three Bi-Obj scenarios, we observe that removing the visit frequency constraints significantly increases the coverage capability

[2] The two random instances `Random1` and `Random2` are not considered for the studies hereafter due to the unavailability of population data.

Table 6 Computational result of Bi-Obj model on the Singapore MRT network

Ins.	*cover**				*dist**				Δu		
	Sec.	u_1^c	d^c	$\overline{d^c}$	Sec.	u_1^d	d^d	$\overline{d^d}$	$\frac{u_1^c-u_1^d}{u_1^d}$ (%)		
Bi-Obj-1											
NE	0.23	422	42	2.80	0.16	354	15	1	19.2		
CC	0.78	301	36	2.40	0.75	241	15	1	24.8		
NS	3.48	1,199	57	2.28	7.91	1,025	25	1	17.0		
EW	8.21	1,311	94	3.13	14.6	1,101	30	1	19.0		
EN	164	1,784	203	4.51	2,611	1,471	45	1	21.2		
ENN	330	2,389	301	5.02	12,899	2,008	60	1	19.2		
All	1,019	2,532	383	5.47	86,400[a]	2,083	71	1.01	21.6		
	Sec.	u_2^c	d^c	$\overline{d^c}$	Sec.	u_2^d	d^d	$\overline{d^d}$	$\frac{u_2^c-u_2^d}{u_2^d}$ (%)	$\frac{u_2^c-u_1^c}{u_1^c}$ (%)	$\frac{u_2^d-u_1^d}{u_1^d}$ (%)
Bi-Obj-2											
NE	0.44	465	46	3.07	0.58	404	15	1	15.1	10.2	14.1
CC	0.42	328	38	2.53	0.17	261	15	1	25.6	8.97	8.30
NS	3.12	1,283	65	2.60	4.84	1,095	25	1	17.2	7.01	6.83
EW	5.62	1,411	69	2.30	12.7	1,217	30	1	15.9	7.63	10.5
EN	185	2,185	154	3.42	1196	1,880	45	1	16.3	22.5	27.8
ENN	1,253	2,760	297	4.95	2,063	2,411	60	1	14.4	15.5	20.1
All	2,267	3,016	305	4.36	7,200[b]	2,616	70	1	15.3	19.1	25.6
	Sec.	u_3^c	d^c	$\overline{d^c}$	Sec.	u_3^d	d^d	$\overline{d^d}$	$\frac{u_3^c-u_3^d}{u_3^d}$ (%)	$\frac{u_3^c-u_1^c}{u_1^c}$ (%)	$\frac{u_3^d-u_1^d}{u_1^d}$ (%)
Bi-Obj-3											
NE	0.36	487	33	2.20	0.25	417	15	1	16.8	15.4	17.8
CC	0.67	376	44	2.93	0.75	267	15	1	40.8	24.9	10.8
NS	2.51	1,435	59	2.36	3.46	1,201	25	1	19.5	19.7	17.2
EW	5.26	1,723	58	1.93	5.77	1,589	30	1	8.44	31.4	44.3
EN	44.7	2,660	161	3.58	32.3	2,369	45	1	12.2	49.1	61.0
ENN	325	3,471	217	3.62	450	3,032	60	1	14.4	45.3	51.0
All	1,797	3,935	225	3.21	578	3,462	70	1	13.6	55.4	66.2

[a] Cut after 86,400 s with a gap of 0.00002 %
[b] Cut after 7,200 switch a gap of 0.0001 %

of the schedule, not only in the coverage-first case but also in the distance-first case. The increased coverage in both cases are shown in the second and the third of the Δu columns of Table 6, being $\frac{u_i^c-u_1^c}{u_1^c}$ and $\frac{u_i^d-u_1^d}{u_1^d}$ with $i = 2, 3$, respectively. As is shown, the Bi-Obj-2 model by removing the minimum-visit constraint in Bi-Obj-1 covers around 10 % more expected loss on the four independent lines NE, CC, NS, and EW; When multiple lines EN, ENN and All are considered, the average increased coverage rises up to around 20 % in the coverage-first case, and around 25 % in the distance-first case. The Bi-Obj-3 model by removing both the minimum-visit and the maximum-visit constraint in Bi-Obj-1 covers averagely 20 % more

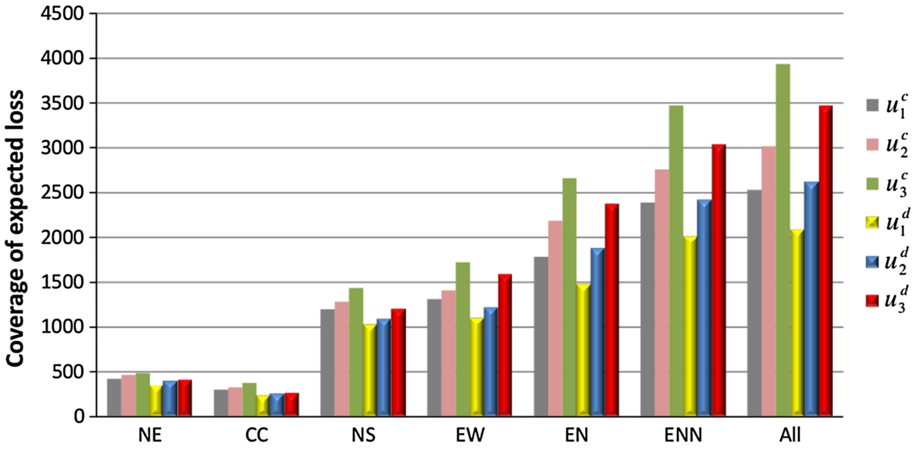

Fig. 6 The optimal coverage of expected loss on the seven real-world instances extracted from the Singapore MRT network. *Six rectangles* for each instance represents from *left to right*: u_1^c, u_2^c, u_3^c, optimum coverage of the coverage-first case of model Bi-Obj-1, Bi-Obj-2, Bi-Obj-3, respectively; and u_1^d, u_2^d, u_3^d, the optimum coverage of the distance-first case of the three Bi-Obj models, respectively

on the small instances of independent lines; and when it comes to the larger instances with multiple lines, the coverage increases by averagely 50 % in the coverage-first case, and around 60 % in the distance-first case. Interestingly, in the coverage-first case, not only the primary objective, the coverage, is increased, the total travel distance in the largest instance is also significantly reduced by 20–40 %.

Although removing the visit-frequency constraints results in significant better patrol schedule in terms of coverage and travel distance, a side-effect is that it may concentrate only on a few most populous stations. For example, the optimal schedule of Bi-Obj-2 patrols 47 stations instead of 79 stations in the Bi-Obj-1 model; and the optimal schedule of Bi-Obj-3 visits only 18 stations out of 79 stations in their totally 84 visits. Besides, within these 18 visited stations, half of them are visited only once; while some of the most-visited stations, for example, Orchard is visited 13 times, City Hall is visited 12 times, and 11 visits are at Raffles Place. This means, these three stations are almost visited by each of the 14 teams, in most of the 20 time periods. Therefore, in practice, the maximum-visit constraint (5) and minimum-visit constraint (4) can be used, if a more dispersed schedule is desired. More details about how to generate visit frequency data on a regular (e.g. daily) basis are discussed in Sect. 6.1.3.

5.4 Results of Var model

The reason for using variable starting time and break time for each patrol team is well explained by the computational results presented in Table 7 where the Var model is deployed. The optimal or best-known coverage of expected loss u_v^c improves over the coverage-first case of the Bi-Obj-3 model in Sect. 5.3 using fixed starting and break time by over 10 % to over 30 %, as indicated in the Δu column of Table 7. Note that this increased coverage is obtained without further concentrating the station visits, e.g. the number of station visited in the whole network is 18, same as obtained in the Bi-Obj-3 scenario. However, the computational overhead is also quite considerable. As shown in the second column of Table 7 even the independent lines NS and EW cannot be solved to optimality within 2 h, leaving a 1.15 and

Table 7 Computational result of VAR model on Singapore MRT Network

Instance	VAR				
	Sec.	Gap (%)	u_v^c	$\Delta u = \frac{u_v^c - u_3^c}{u_3^c}$ (%)	d^c
NE	30.5	0	641	31.6	38
CC	60.7	0	478	27.1	48
NS	7,200	1.15	1,677	16.9	61
EW	7,200	4.58	1,979	14.9	69
EN	86,400	4.92	3,040	14.3	137
ENN	86,400	5.28	3,918	12.9	278
All	86,400	7.01	4,395	11.7	276

4.58 % gap, respectively. Solving the complete network with 79 stations obtains a feasible solution with 14.0 % gap in 2 h, we let the computation continue for 24 h, and the gap is reduced to 7.01 %.

6 Generating random schedules

In the previous sections, we present and discuss several deterministic mathematical model for generating a schedule for patrolling MRT rail stations. The objective is to generate *one* optimal patrol schedule to cover as much expected loss as possible while the total travel distance is minimized. In this section, we discuss a different problem, for which the patrol schedule is required to be generated recurrently, for example, on a day-to-day basis. In such case, the generated schedules should be as random as possible so that it is difficult for adversaries to predict. In the following, we will first discuss the randomization strategies used in our work, and then present the resulting schedules generated during a simulation of 30 days.

6.1 Random strategies

The element of randomness is important to patrol scheduling to hedge against adversarial observations. For example, the series of random schedules should not allow the adversaries to spot a never-guarded station, or spot a never-guarded time period. Besides, the randomization approach should be computable within a reasonable time for daily use. Therefore, one idea is to use one of the deterministic models studied in this article, and randomize the input data. In the following subsections, we discuss the choice of deterministic model and different randomization strategies, including randomizing starting time and break time for each patrol team, the visit frequency and visit time of each station.

6.1.1 Choice of mathematical model

Although enabling variable starting times and break times by the VAR model allows substantially more coverage of expected loss, totally automating the decision of working period still faces two drawbacks: the computability and convergent patrol timetable. From the computability aspect, as shown in Sect. 5.4, it takes 2 h to find a first feasible solution, and leaves a 7 % gap after 24 h computation. On the other hand, as observed from some of the sub-optimal solutions computed by the VAR model, most teams gather between the busiest hours from 13 to 21 o'clock, while leaving some security vacuum hours totally unguarded.

To this end, we propose to use the Bi-Obj model with fixed starting time and break time for generating random daily patrol schedules. On one hand, this model solves the whole Singaporean MRT network with 79 stations and 14 teams within reasonable computation time. On the other hand, it allows a more dispersed patrol time pattern, and it also has a nice property for randomizing the starting time input, as will be discussed in more details in the following section.

6.1.2 Randomizing patrolling time

The Bi-Obj model uses a fixed starting time and break times for each patrol team, which is given as input parameters. The starting time and the break times can be generated randomly either based on a uniform distribution, which allows a maximum dispersion of patrol periods, or based on a distribution that is biased by the total amount of expected loss for each time period, which allows more coverage at the more crowded hours. Additional constraints can be easily added to this randomized generation of starting time process, e.g. to ensure each time period to be patrolled by at least one team.

6.1.3 Randomizing patrolling stations

As the deterministic model Bi-Obj may concentrate the patrol resources on a few busiest stations shown in Sect. 5.3, one should enforce visits to the less populous stations from time to time by using the minimum-visit constraint (4). However, requiring each station to be visited every day by setting V_j^{min} to one for all stations may not be a good idea, since it may greatly reduce the coverage of expected loss from our studies in Sect. 5.3. Here, we propose to randomly generate the minimum-visit vector V_j^{min} for each station j based on a daily coverage probability vector P_j. Suppose a station j is expected to be visited every 4 days, this can be simply realized by setting $P_j = 0.25$, i.e. the probability of setting $V_j^{min} = 1$ is 25 %. We propose to set V_j^{min} to no greater than 1 for all stations, since on one side this guarantees feasibility of the model given the current available number of patrol teams according to Eq. (30), on the other side it leaves more space for the optimization algorithm to decide the number of visits for each station so that better quality of the patrol schedule can be expected.

In order to further disperse the patrolling time of these stations, from the set of stations that require a visit, dubbed $J^v := \{j \in J : V_j^{min} \geq 1\}$, we can further randomly select n stations, dubbed $J^{v,t} \subseteq J^v : n = |J^{v,t}|$, and randomly specify their time to be visited. This can be done as follows:

1. let $j_1, j_2, \ldots, j_n \in J^{v,t}$ denote the stations whose visit time needs to be specified;
2. assign each of the stations $j_r \in J^{v,t}$ to a distinct team $i_r \in I$, whose working periods W_{i_r} is determined by the procedure described in Sect. 6.1.2;
3. randomly select a time k_r from working periods W_{i_r} of team i, either based on a uniform distribution, or based on a distribution that is biased by the data of expected loss;
4. specify that station j_r is visited by team i_r at time k_r by setting $x_{i_r,j_r,k_r} = 1$.

We propose to set the number n of stations whose time need to be prefixed to be less than the number of teams, i.e. $n \leq |I|$, so that each team is assigned with at most one prefixed station. This setting on one hand guarantees the model feasibility (otherwise, assigning two stations to one team may lead to infeasibility due to the consecutiveness constraint); on the other hand, it also leaves more space for the optimization algorithm in the hope of finding a better schedule. Besides, fixing some variables to certain values usually does not increase the computational difficulty of the model, making this approach applicable for everyday use.

6.2 Results of randomized schedules

In this section, we set up experiments to simulate the generation of random daily patrol schedules in a time span of 30 days. To this end, we followed the randomization strategies discussed in Sect. 6.1, adopt the deterministic model Bi-Obj with the *coverage-first* case, assign the starting time as dispersed as possible, and randomize the visit frequency and visit time as described in Sect. 6.1.3. The probability P_j of setting the minimum number of visiting a station j—V_j^{min} to one is set to 0.25, i.e. each station is expected to be visited once every 4 days. We randomly generate 30 sets of minimum visit frequency for all the 79 stations in the Singaporean MRT rail network, where each set represents an input for generating one day's patrol schedule. From each daily set of stations that require a visit, we randomly select 14 stations, and assign each of them randomly to one of the patrol teams at one of its working hours. Besides, as discussed at the end of Sect. 5.3, in order to obtain reasonably dispersed schedule, the maximum number V_j^{max} of visiting a station j is empirically set to 4. Then the original Bi-Obj-1 model with both the minimum-visit and maximum-visit constraint is used.

The experimental results show that our randomization approach is promising in terms of both computational and schedule quality. From the computational aspect, as shown in Table 8, the median computation time for generating daily schedule is around 108 s, and the maximum computation time required among 30 instances is 550 s, i.e. less than 10 min. Considering the schedule quality, as summarized in Table 8, the coverage of expected loss is averaged 3,019 with a standard deviation of around 94, and the median coverage is 3,032, which indicates a slightly better schedule than the deterministically optimal schedule computed by Bi-Obj-2 (see Table 6) in terms of coverage; however, the average distance traveled is 357 with median value 350 is higher than the optimal distance 305 computed by Bi-Obj-2, which is due to the random enforcement of visits at some rural stations. The number of different stations visited is around 40 (out of 79 stations) per day, which indicates a reasonably dispersed daily patrol pattern. The number of visits at each station on each day in our 30-day simulation is further listed in Table 9. As can be observed, on one hand, every station, no matter how rural or unpopulated, is in general visited at least once every 4 days, and the visit pattern is random and reasonably dispersed; on the other hand, the more populated the station, the more often it is visited, e.g. the most populated stations such as Raffles Place, City Hall, Ang Mo Kio, and Orchard (indexed 16, 17, 44, 50) are visited exactly 4 times each day, which is maximum number of daily visits. We further examine the patrolling time of these four most populous stations. As illustrated in Fig. 7, none of the stations are visited at the same time every day. The visit frequency depends in general on the number of passengers in the station at that hour. For example, business district such as Raffles Place, may concentrate

Table 8 The statistical results of generated random schedules

	Mean ± sd.	Min.	25%-quant.	Median	75%-quant.	Max.
Runtime	127 ± 115	0.5	67.3	108	152	550
Coverage	3,019 ± 94.1	2,803	2,947	3,032	3,080	3,184
Distance	357 ± 42.2	282	324	350	392	443
#stations	40.5 ± 2.42	37	39	40	42	47

Table 9 Number of visits at each station on in 30 days

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
2	0	2	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1
3	2	3	2	4	1	2	3	2	0	1	3	2	3	4	3	3	2	1	1	2	3	0	2	4	2	3	4	4	1	2
4	0	1	0	1	0	0	1	0	0	0	1	0	0	1	0	1	0	0	0	0	2	0	1	0	0	1	0	1	1	1
5	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
6	2	4	2	4	0	2	2	4	0	1	3	3	2	3	2	3	2	1	1	2	4	0	1	4	1	3	2	1	1	2
7	1	2	2	4	1	1	3	3	0	1	2	3	3	4	3	3	1	1	2	2	4	0	2	3	1	3	4	3	1	2
8	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0
9	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0
10	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
11	0	0	0	3	0	0	1	1	0	0	0	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	0	0	0	1
12	1	0	1	1	0	1	1	0	0	2	0	1	0	1	0	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0
13	1	1	1	0	1	0	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
14	0	0	2	0	3	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	2	2	0	0	0	0	0	1	0	0
15	4	4	4	4	4	4	4	4	3	4	4	4	4	3	4	4	4	4	4	4	2	4	4	3	4	4	4	2	4	3
16	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
17	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
18	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
19	2	0	0	1	1	0	1	0	0	1	0	0	0	1	1	0	1	1	0	2	0	1	0	1	1	0	2	0	2	0
20	1	0	1	1	0	0	1	1	1	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0
21	1	1	0	1	1	1	1	2	2	0	1	0	2	0	0	1	2	3	0	0	0	1	2	0	0	1	2	2	2	0
22	2	1	1	1	0	1	2	2	3	2	2	2	1	0	1	1	2	1	3	2	1	0	3	3	1	0	2	3	3	4
23	1	0	0	1	0	0	1	0	0	0	0	0	1	1	0	1	0	0	1	1	0	2	1	1	0	2	1	0	0	0
24	0	0	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0	1	0	1	1	0	0	0	1	0	0	0
25	2	2	0	0	1	1	3	3	3	1	1	1	2	1	1	2	2	2	2	2	1	2	3	1	0	2	2	4	3	2
26	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	0	1
27	1	1	1	1	1	1	1	1	0	0	0	0	1	1	0	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	4	2	2	2	1	0	2	1	1	1	1	1	2	2	0	1	1	3	0	1	3	4	2	1
29	0	1	0	1	1	1	3	1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	2	0	1	1	2	1	1
30	0	0	0	0	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
31	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
32	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	0	0	1
33	0	0	1	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	0	0
34	1	3	0	4	1	2	1	2	1	1	2	3	1	4	3	3	0	1	1	1	4	0	2	4	2	3	4	2	3	2
35	1	0	1	0	0	1	1	0	0	0	1	0	0	2	0	1	0	0	0	0	2	0	1	0	0	1	0	0	0	0
36	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0
37	0	0	0	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	0
38	3	2	1	2	1	2	2	2	1	2	2	2	4	4	3	4	1	2	2	4	3	4	3	3	3	4	4	2	2	4

Table 9 continued

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
39	0	0	0	0	0	1	1	0	0	0	0	1	1	2	1	2	0	3	1	2	1	1	0	1	1	0	0	1	0	0
40	1	0	1	0	0	1	0	0	1	1	0	0	0	2	0	1	1	1	0	2	0	1	1	0	1	0	1	0	0	1
41	4	4	4	4	4	4	2	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
42	0	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1
43	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
45	4	3	3	2	4	2	2	3	4	3	3	3	4	2	2	3	4	4	4	3	4	4	2	2	4	3	0	2	4	2
46	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1
47	4	3	4	4	4	3	1	3	4	4	4	2	3	3	4	4	4	4	4	4	3	4	4	2	4	2	3	4	4	3
48	4	4	3	1	4	3	2	2	4	4	4	3	4	2	3	4	4	4	4	3	4	3	4	2	4	3	2	1	3	2
49	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0
50	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
51	0	0	0	0	3	0	0	1	0	1	0	2	0	1	1	0	1	0	1	1	2	0	0	1	1	0	0	2	1	0
52	4	3	4	2	4	4	1	3	4	4	4	4	4	1	4	4	4	4	4	4	2	4	2	1	4	2	2	2	3	4
53	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0
54	3	4	4	1	4	4	0	2	4	4	4	2	1	2	3	2	4	4	4	3	2	3	1	1	2	3	2	0	1	4
55	1	1	1	0	2	0	0	0	1	1	0	2	0	0	2	0	1	2	3	1	0	1	1	1	2	0	1	0	1	1
56	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	2	0	0	1	0	0	0	1	0	0	0	0
57	1	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	2	0	0
58	0	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1
59	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0
60	0	0	1	0	0	1	1	0	0	1	0	0	2	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
61	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0
62	4	3	4	2	4	3	3	3	4	4	4	2	2	2	2	1	4	4	3	1	1	2	3	1	4	3	1	2	4	4
63	0	1	0	0	0	0	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1	0	0
64	1	2	1	0	0	1	1	2	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	0	2	0	0	0	1	0
65	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0
66	1	1	0	0	0	1	0	1	1	0	1	0	1	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0	1	1
67	1	0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1
68	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
69	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
70	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0
71	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	2	0	1	0	0	0	0	0	1
72	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0
73	0	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	1
74	0	0	0	0	1	1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
75	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0

Table 9 continued

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
76	1	0	1	1	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
77	0	0	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1	0	1	1	1	0	0	0	0
78	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0
79	0	0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0

The first row indicates one of 30 days, and the first column indicates the station indices

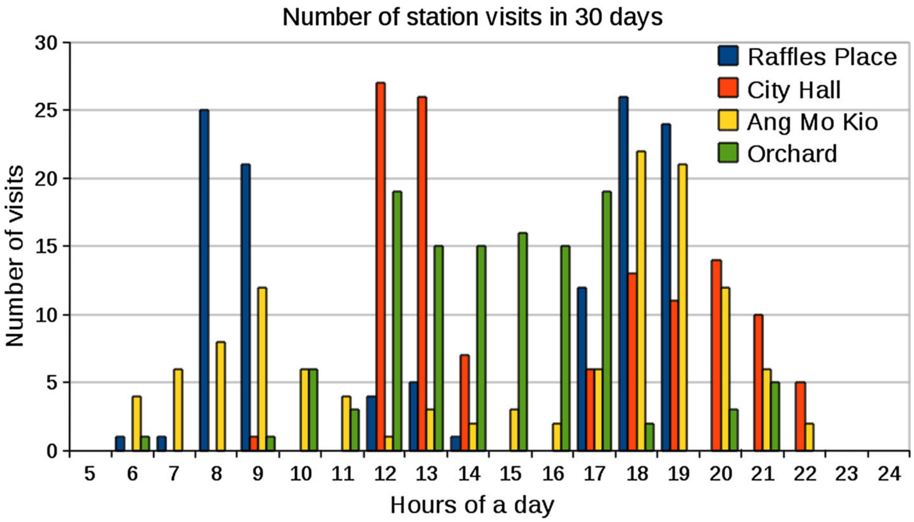

Fig. 7 The number of visits of the four most visited stations, namely, Raffles Place, City Hall, Ang Mo Kio, and Orchard, by the time periods in a 30 days' simulation

its visiting periods around office hours such as 8–9 in the morning or 6–7 in the afternoon, since the number of passengers there dominates other stations during these hours. And we have observed that our randomization strategy still manages to deviate its visitation to around 25 times in 30 days. Shopping and leisure centers such as Orchard and City Hall, as well as residential area Ang Mo Kio, have in general a more dispersed visitation time pattern. For example, Orchard as shown in Fig. 7, is patrolled less than 20 times in each hour, as its most visited hours are from 12 at noon to 5 in the afternoon; residential area Ang Mo Kio has the most dispersed distribution of patrolling time, as it is patrolled in all the 17 h from 6 a.m. to 10 p.m. This simulation experiment shows that our randomization strategy is capable of generating random and dispersed patrol schedules in terms of visit frequency and visit time, while keeping computational scalability and high schedule quality in terms of coverage of expected loss.

7 Conclusion

In this paper, we defined the problem of scheduling security forces to patrol a mass rapid transit rail network and provided an elaborate case study for patrol scheduling of the Singapore

Mass Rapid Transit network. Methodologically, we proposed three integer linear programming models to formulate different variants of the patrol scheduling problem. We further experimentally examined the computational scalability of these models and investigated the impact different models under different scenarios may have on the optimal schedules. Since security is a major concern of patrol scheduling, deterministic schedules are undesirable due to predictable vulnerabilities. Strategic randomization is one aspect that has to be considered in this problem. In this paper, we proposed simple randomization strategies by randomizing the starting time and break time for each team, and the visit frequency and visit time for each station. We reported the efficiency and effectiveness of our proposed approach under different circumstances. Interestingly, by randomizing the visit frequency and some visit times, we see that the resulting schedules over a 1-month horizon are sufficiently randomized in that there is no discernible patterns. This achieves the purpose of generating patrol schedules that hedge against adversarial observations.

We believe that our model does not require major customizations for use in other mass rapid transit systems with similar constraints and requirements.

There are many possible extensions to our work. For the purpose of reducing the computing time needed to solve the proposed model, we can consider approaches such as strengthening its LP relaxation by adding valid inequalities or reducing the number of variables by using pricing procedures. Developing an effective heuristic algorithm for quickly providing a feasible solution as an upper bound in the branch-and-bound procedure should also speed up the optimization process. The random starting time for each team can also be obtained by sampling from marginal probability of a certain distribution (Rosenshine 1970). This paper merely considers a simple randomization strategy for the operator, but does not take the strategic behaviour of adversaries into account. Extending our proposed model to cover adversarial aspects is a very interesting area. One approach is to consider Stackelberg game models which have been applied in a variety of security domains (Ordóñez et al. 2013; Tsai et al. 2009).

References

Babu, V. L. L., Batta, R., & Li, L. (2006). Passenger grouping under constant threat possibility in an airport security system. *European Journal of Operational Research*, *168*(2), 633–644.

Basar, T., & Olsder, G. J. (1995). *Dynamic noncooperative game theory*. London: Academic Press.

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

Curtin, K. M., McCall, K. H., & Qiu, F. (2010). Determining optimal police patrol areas with maximal covering and backup covering location models. *Networks and Spatial Economics*, *10*(1), 125–145.

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.

Gunawan, A., & Lau, H. C. (2013). Master physician scheduling problem. *Journal of the Operational Research Society*, *64*(3), 410–425.

Jain, M., Tsai, J., Pita, J., Kiekintveld, C., Rathi, S., Tambe, M., et al. (2010). Software assistants for randomized patrol planning for the lax airport police and the federal air marshals service. *Interfaces*, *40*(4), 267–290.

Jiang, A.X., Yin, Z., Kietkintveld, C., Leyton-Brown, K., Sandholm, T., & Tambe, M. (2012). Towards optimal patrol strategies for fare inspection in transit systems. In *AAAI spring symposium on game theory for security, sustainability and health*.

Maenhout, B., & Vanhoucke, M. (2010). A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *European Journal of Operational Research*, *206*(1), 155–167.

Ordóñez, F., Tambe, M., Jara, J. F., Jain, M., Kiekintveld, C., & Tsai, J. (2013). *Handbook of operations research for homeland security*. International series in operations research & management science (Vol. 183, pp. 45–72).

Petrovic, S., & Berghe, G. V. (2008). Comparison of algorithms for nurse rostering problems. In *Procedings of the 7th international conference on the practice and theory of automated timetabling* (pp. 1–18).

Rosenshine, M. (1970). Contributions to a theory of patrol scheduling. *Operational Research Quarterly, 21*(1), 99–106.

Sharma, D. K., Ghosh, D., & Gaur, A. (2007). Lexicographic goal programming model for police patrol cars deployment in metropolitan cities. *International Journal of Information and Management Sciences*, *18*(2), 173–188.

Sinuany-Stern, Z., & Teomi, Y. (1986). Multi-objective scheduling plans for security guards. *Journal of the Operational Research Society, 37*, 67–77.

Taylor, P. E., & Huxley, S. J. (1989). A break from tradition for the san francisco police: Patrol officer scheduling using an optimization-based decision support system. *Interfaces*, *19*(1), 4–24.

Tsai, J., Kiekintveld, C., Ordóñez, F., Tambe, M., & Rathi, S. (2009). Iris-a tool for strategic security allocation in transportation networks. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems* (pp. 37–44).

Varakantham, P., Lau, H. C., & Yuan, Z. (2013). Scalable randomized patrolling for securing rapid transit networks. In *Twenty fifth annual conference on innovative applications of artificial intelligence (IAAI)*.

Zhang, Y., & Brown, D. E. (2013). Police patrol districting method and simulation evaluation using agent-based model & gis. *Security Informatics*, *2*(7), 1–13.