The Patrol Scheduling Problem

Hoong Chuin Lau • 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 at varying time periods of the network subject to rostering as well as security-related constraints. We present a mathematical programming model for this problem. We then discuss the aspect of injecting randomness by varying the start times, the break times for each team as well as the number of visits required 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 recent 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, 2010). A more challenging problem involves rostering of personnel that require them to move from one geographical location to another as they discharge 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. In order to enforce security, security personnel or teams are deployed to patrol the stations throughout the day. Unlike

Hoong Chuin Lau
School of Information Systems, Singapore Management University
80 Stamford Road, Singapore 178902
E-mail: hclau@smu.edu.sg

Aldy Gunawan
School of Informatics & IT, Temasek Polytechnic
21 Tampines Avenue 1
E-mail: agunawan@tp.edu.sg

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 an exact model to deploy security teams to stations in varying time periods of the network 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 Section 3. We provide a deterministic mathematical programming model that solves the problem, followed by a randomized strategy which allows the planner to generate solutions based on randomized start times, break times for each team as well as the number of visits required for each station. The next section is dedicated to the computational 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), which 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 there is based on the assumption of randomness. 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 arc flows 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.

Stern and Teomi (1986) studied and proposed two algorithms for 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 and Ghosh (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 research. Tsai et al. (2009) for example modeled the strategic security allocation problem as a Strackelberg game and developed the *Intelligent Randomization In Scheduling* (IRIS) system – a tool for strategic security allocation in transportation networks. The algorithmic advances in multi-agent systems research are being used to solve the class of massive security games with complex constraints, the Federal Air Marshals (FAMs) that provide law enforcement aboard U.S. commercial flights.

Ordóñez et al. (2012) described 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. They proposed fast algorithms for solving large instances of Bayesian Stackelberg games to two real-world security applications: 1) the police at the Los Angeles International Airport and 2) the Federal Air Marshal Service. 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). 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.

3 Problem Definition

This paper focuses on a patrol scheduling the mass rapid transit rail network. Figure 1 shows the subway systems of London, Beijing, Paris and Singapore respectively. A common feature of these networks is that each network consists of many stations 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 work day divided into time periods. A shift is defined as a consecutive set of time periods, and in this paper, we assume each period to be one hour, and there are two 8-hourly shifts (7am – 3pm and 3pm – 11pm respectively). Each team is rostered to a single shift duty during which it is responsible for patrolling/visiting a subset of stations in the network. We assume each station patrol/visit takes one period and each shift is made up of exactly six visits plus two breaks.

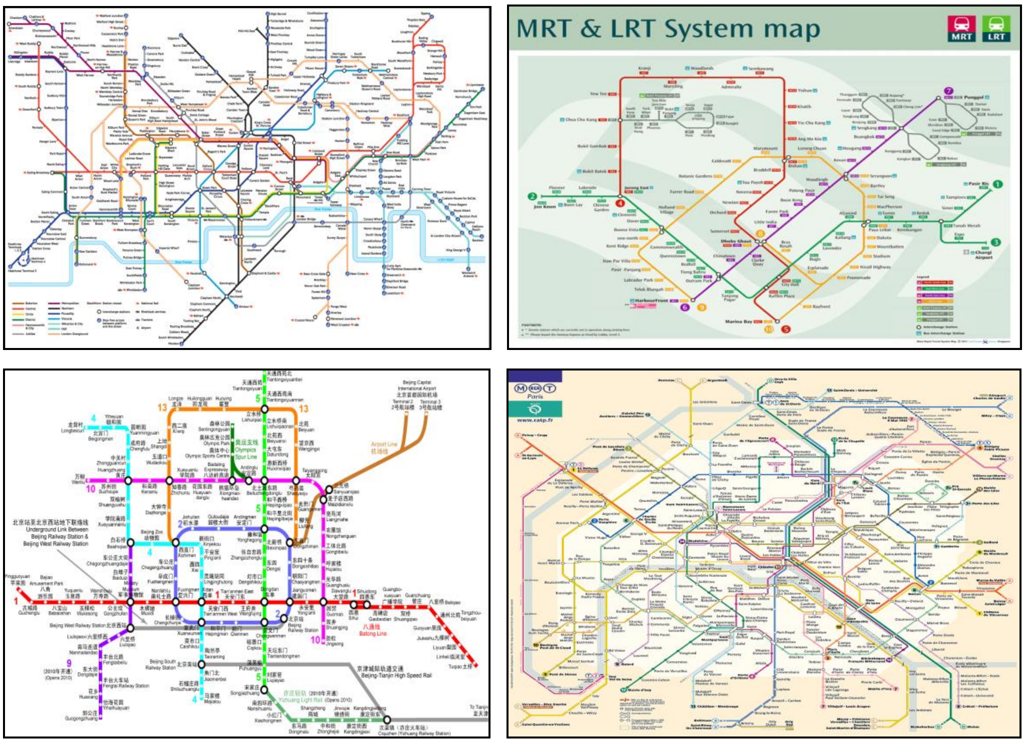

Figure 1. Examples of the mass rapid transit rail network in some cities

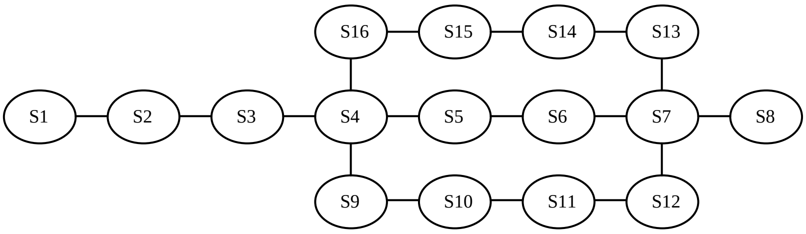

Figure 2. Example of the mass rapid transit rail network

Time Period	1	2	3	4	5	6	7	8
Team1	S1	S3	Break	S5	S6	Break	S7	S13
Team2	S10	S12	Break	S14	S16	Break	S4	S2

Figure 3. Example of Patrol Scheduling Problem

As shown in Figure 2 as illustration, the 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 in the first shift, Figure 3 represents one possible patrol scheduling for both teams. Team1 has to visit S1, S3, S5, S6, S7 and S13 consecutively while Team2 has to visit S10, S12, S14, S16, S4 and S2 consecutively.

Our goal is to minimize the total distance travelled. We use this objective in order to generate solutions that minimizes unnecessary movements (where teams move in a certain path, rather than haphazardly or in loops). The distance travelled 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). For example, the distance between S1 and S4 is 3 stations. Furthermore, we also impose an additional penalty for distance between two stations using *different* lines. From S5 to S16, the distance travelled is 2 stations + Δ, where Δ is the penalty value. In our study, we set Δ to an arbitrarily number, e.g. 10. This can be set to any large number with the purpose to minimize unnecessary movements.

The following summarizes the requirements/constraints treated in this paper:

- The number of visits for each team should meet the requirement.
- At most one team can visit a particular station at a particular time period.
- Each station has a minimum and maximum number of visits per day.
- Each team has its own start and finish times. In this paper, we treat the start and finish times as input (i.e. assume they have been determined by the planner).
- Each team may only visit a particular station at most once during its duty.
- Each team visits at most one station at a particular time period.
- Consecutiveness constraints: describes whether a pair of stations can be visited consecutively (i.e. one after another).
- Break constraints: each team is given two breaks, and breaks cannot occur consecutively.

As discussed in the Introduction, the element of randomness is important to patrol scheduling to hedge against adversarial observations. To this end, game theory has been applied recently (see Literature Review above) which utilizes reports from Intelligence sources to compute mixed strategies. Since our focus in this paper is on the patrol scheduling problem defined above, we treat the computation of these mixed strategies as a pre-processing step, which is computed whenever a new roster needs to be generated (e.g. daily).

It is conceivably that Intelligence sources will provide different data on the vulnerability of stations from time to time, which in our context can be translated to the input required by our problem (more precisely, the randomized frequency of visits of each station, as well as start times and break times of each patrol team). Our purpose in this paper is to demonstrate that by appropriately randomizing the frequency of visits, we can effectively deter crimes compared to fixed frequency.

4 Mathematical Programming Model

In this section, we first present a deterministic mathematical programming model to solve the patrol scheduling problem. We then consider a simple strategy for randomizing inputs for feeding into our model.

4.1 Deterministic Model

The problem can be presented as a mathematical programming model, using the following sets, input parameters and decision variables:

Parameters

I = Set of patrol teams, $i \in \{1,2,\ldots,|I|\}$

J = Set of stations, $j \in \{1,2,\ldots,|J|\}$

K = Set of time periods, $k \in \{1,2,\ldots,|K|\}$

Req_i = number of visit required for patrol team i per day $(i \in I)$

Max_Visit_j = maximum number of visit required for station j per day $(j \in J)$

Min_Visit_j = minimum number of visit required for station j per day $(j \in J)$

$Start_i$ = start time for team i $(i \in I)$

$Finish_i$ = finish time for team i $(i \in I)$ $(\text{i.e,} Finish_i = Start_i + 7)$

$Break_i^1$ = team i's first break (i.e., the first break is at period $(Start_i + Break_i^1)$

$Break_i^2$ = team i's second break $(Break_i^2 > Break_i^1 + 1)$

$Dist_{j_1 j_2}$ = the distance between stations j_1 and j_2 $(j_1, j_2 \in J)$

$Cons_{j_1 j_2}$ = 1 if a patrol team can visit j_2 consecutively (i.e. at the next time period) after visiting station j_1, and 0 otherwise

Decision variables

X_{ijk} = 1 if patrol team i visits station j at time period k $(i \in I, j \in J, k \in K)$, 0 otherwise

The formulation for the Patrol Scheduling problem is then given by

$$\text{Minimize } Z = \sum_{i \in I} \sum_{\substack{j_1 \in J, j_2 \in J \\ (j_1 \neq j_2)}} \sum_{\substack{k \in \{0,\ldots,(Finish_i - Start_i - 1)\} \\ k \notin \{Break_i^1, Break_i^1 - 1, Break_i^2, Break_i^2 - 1\}}} Dist_{j_1 j_2} \times X_{ij_1(Start_i + k)} \times X_{ij_2(Start_i + k + 1)}$$
$$+ \sum_{i \in I} \sum_{\substack{j_1 \in J, j_2 \in J \\ (j_1 \neq j_2)}} \sum_{k \in \{Break_i^1 - 1, Break_i^2 - 1\}} Dist_{j_1 j_2} \times X_{ij_1(Start_i + k)} \times X_{ij_2(Start_i + k + 2)} \qquad (1)$$

subject to:

$$\sum_{j \in J} \sum_{k \in K} X_{ijk} = Req_i \qquad i \in I \qquad (2)$$

$$\sum_{i \in I} X_{ijk} \leq 1 \qquad j \in J, k \in K \qquad (3)$$

$$\sum_{i \in I} \sum_{k \in K} X_{ijk} \geq Min\_Visit_j \qquad j \in J \qquad (4)$$

$$\sum_{i \in I} \sum_{k \in K} X_{ijk} \leq Max\_Visit_j \qquad j \in J \qquad (5)$$

$$\sum_{k \in K} X_{ijk} \leq 1 \qquad i \in I, j \in J \qquad (6)$$

$$\sum_{j \in J} X_{ijk} \leq 1 \qquad i \in I, k \in K \qquad (7)$$

$$\sum_{j \in J} \sum_{\substack{k \in K \\ k < Start_i}} X_{ijk} = 0 \qquad i \in I \qquad (8)$$

$$\sum_{j \in J} \sum_{\substack{k \in K \\ k > Finish_i}} X_{ijk} = 0 \qquad i \in I \qquad (9)$$

$$\sum_{j \in J} X_{ij(Start_i + Break_i^1)} = 0 \qquad i \in I \qquad (10)$$

$$\sum_{j \in J} X_{ij(Start_i + Break_i^2)} = 0 \qquad i \in I \qquad (11)$$

$$X_{ij_2(Start_i + k + 1)} \leq \sum_{\substack{j_1 \in J \\ j_1 \neq j_2}} Cons_{j_1 j_2} \times X_{ij_1(Start_i + k)} \qquad \begin{array}{l} i \in I, j_2 \in J, k \in \{0, \ldots, Finish_i - Start_i - 1\} \\ k \notin \{Break_i^1, Break_i^2, Break_i^1 - 1, Break_i^2 - 1\} \end{array} \qquad (12)$$

$$X_{ij_2(Start_i + k + 2)} \leq \sum_{\substack{j_1 \in J \\ j_1 \neq j_2}} Cons_{j_1 j_2} \times X_{ij_1(Start_i + k)} \qquad i \in I, j_2 \in J, k \in \{Break_i^1 - 1, Break_i^2 - 1\} \qquad (13)$$

$$X_{ijk} \in \{0,1\} \qquad i \in I, j \in J, k \in K \qquad (14)$$

Equation (1) shows that the objective function consists of two terms. The first term of the objective function refers to the distance travelled between two consecutive time periods. Due to the break constraints, we introduce the second term in the objective function. Suppose a team has the start time at Period 1 and the breaks at Periods 3 and 6. Then the first term calculates the distance travelled between Periods 1 and 2, Periods 4 and 5 and Periods 7 and 8, while the second term computes the distance travelled between one period before and after the break (i.e. Periods 2 and 4 and Periods 5 and 7).

Constraint (2) ensures that all teams have to visit a certain number of stations during their duty. Constraint (3) restricts that only one team can visit a particular station at a particular time period. Constraints (4) and (5) represent the number of visits allowed for each station per day. In our problem, the interchange stations are visited more often than those of non-interchange stations. Each team can only visit a particular station at most once per day and each team can only patrol at most one station at a particular time period. These requirements are represented by Constraints (6) and (7).

Note that the start (and therefore finish) times of each team is an input to the model. Constraints (8) and (9) ensure that all teams can only perform their patrolling task during their shift. The break constraints are defined in (10) and (11). The consecutiveness constraint is represented by constraints (12) and (13). Constraint (12) defines the consecutiveness requirement between two consecutive periods of duty. Since the breaks occur at periods $Start_i + Break_i^1$ and $Start_i + Break_i^2$, we introduce constraint (13) to ensure consecutiveness between the stations visited the one period before and after a particular break.

Notice that the objective function of the model is not linear and our experiment shows that the model given above cannot be solved within reasonable time. Furthermore, standard linearization technique (see Hammer and Rudeanu, 1968) also yields unsatisfactory performance. In the following, we propose a linearization of the problem by introducing an additional set of binary variables $Y_{ij_1kj_2m} = X_{ij_1k} \times X_{ij_2m}$. The objective function (1) is replaced by the following equation:

$$\text{Minimize } Z = \sum_{i\in I} \sum_{j_1\in J} \sum_{\substack{j_2\in J\\(j_1\neq j_2)}} \sum_{\substack{k\in\{0,\ldots,(Finish_i - Start_i - 1)\}\\ k\notin\{Break_i^1, Break_i^1-1, Break_i^2, Break_i^2-1\}}} Dist_{j_1j_2} \times Y_{ij_1(Start_i+k)j_2(Start_i+k+1)}$$

$$+\sum_{i\in I} \sum_{j_1\in J} \sum_{\substack{j_2\in J\\(j_1\neq j_2)}} \sum_{k\in\{Break_i^1-1, Break_i^2-1\}} Dist_{j_1j_2} \times Y_{ij_1(Start_i+k)j_2(Start_i+k+2)} \quad (15)$$

To achieve this, the following constraints need to be added:

$$\sum_{\substack{j_2\in J\\ j_2\neq j_1}} Y_{ij_1(Start_i+k)j_2(Start_i+k+1)} = X_{ij_1(Start_i+k)} \qquad \begin{array}{l} i\in I, j_1\in J, k\in\{0,\ldots,Finish_i - Start_i - 1\} \\ k\notin\{Break_i^1, Break_i^2, Break_i^1-1, Break_i^2-1\}\end{array} \quad (16)$$

$$\sum_{\substack{j_1\in J\\ j_1\neq j_2}} Y_{ij_1(Start_i+k)j_2(Start_i+k+1)} = X_{ij_2(Start_i+k+1)} \qquad \begin{array}{l} i\in I, j_2\in J, k\in\{0,\ldots,Finish_i - Start_i - 1\} \\ k\notin\{Break_i^1, Break_i^2, Break_i^1-1, Break_i^2-1\}\end{array} \quad (17)$$

$$\sum_{\substack{j_2\in J\\ j_2\neq j_1}} Y_{ij_1(Start_i+k)j_2(Start_i+k+2)} = X_{ij_1(Start_i+k)} \qquad \begin{array}{l} i\in I, j_1\in J, k\in\{0,\ldots,Finish_i - Start_i - 1\} \\ k\notin\{Break_i^1-1, Break_i^2-1\}\end{array} \quad (18)$$

$$\sum_{\substack{j_1\in J\\ j_1\neq j_2}} Y_{ij_1(Start_i+k)j_2(Start_i+k+2)} = X_{ij_2(Start_i+k+2)} \qquad \begin{array}{l} i\in I, j_2\in J, k\in\{0,\ldots,Finish_i - Start_i - 1\} \\ k\notin\{Break_i^1-1, Break_i^2-1\}\end{array} \quad (19)$$

$$Y_{ij_1kj_2m} = \{0,1\} \qquad i\in I, j_1 \,\&\, j_2\in J, k \,\&\, m\in K \quad (20)$$

4.2 Randomized Strategy

The above model will work well in commercial rosters where the emphasis is on regularity. In security patrol scheduling context however, the element of randomness is important. In this section, we consider the problem of randomizing the start times, the break times for each team as well as the number of visit required for each station.

For start times and break times, we generate them randomly for each team based on a uniform distribution, which may be replaced with any other probability distributions. We would like to observe the impact of computational performance in solving the underlying deterministic mathematical model.

The frequency of visits is not purely random, but is dependent on the level of threats (vulnerability) of each station. In this paper, we assume the existence of Intelligence sources that provide information about the likelihood that crimes are going to occur at each particular station, from which we can calculate the randomized frequency distribution of visits.

More precisely, suppose X is a discrete random variable that represents the adversary's probability distribution of committing a crime, given as follows:

$$X = \begin{cases} Pr(\text{a crime occurs at Station1}) = p_1 \\ Pr(\text{a crime occurs at Station2}) = p_2 \\ \vdots \\ Pr(\text{a crime occurs at Station} \mid J \mid) = p_{|J|} \end{cases} \quad (21)$$

where $p_1 + p_1 + ... + P_{|J|} = 1$. Assume that there are $|I|$ teams where each team has to visit Req_i stations, so the total number of visits per day is $\sum_{i \in I} Req_i$. In this paper, we use the inverse transform method (Ross, 2009) to generate the randomized strategy (i.e. the distribution of the number of visits for all stations) as follows. We generate $\sum_{i \in I} Req_i$ random numbers drawn from the uniform distribution $U(0,1)$, and for each number U, we increment the number of visits by one to Station j if $\sum_{i=1}^{j-1} p_i < U \leq \sum_{i=1}^{j} p_i$ The result is a randomized vector which represents the frequency of visits for all stations.

To simulate the occurrence of crime, we apply the same method, namely, generate a random number U from the uniform distribution $U(0,1)$; a crime occurs at Station j if $\sum_{i=1}^{j-1} p_i < U \leq \sum_{i=1}^{j} p_i$ To test the effectiveness of our proposed randomized strategy, we perform a simulation of a number of replications. For each replication, we simulate the occurrence of crime at a particular station as described above, and determine whether the roster generated from the randomized visit frequency is able to counteract/deter this crime. This is benchmarked against a fixed frequency of visits described above. In the following section, we report results on the effectiveness of our random strategy against the fixed strategy.

5 Computational Results

In this section, we present the computation results together with our evaluation based on of the proposed mathematical programming model. All experiments that we report on this section were run on a 3.07 GHz Intel (R) Xeon (R) CPU with 128GB of RAM under the Microsoft Windows XP Operating System. The mathematical programming model was solved by CPLEX 10.0 solver engine. We first describe the experimental setup, followed by experimental results.

5.1 Experimental Setup

In order to demonstrate the capabilities of our proposed model, the Singapore rail network was chosen as a case study (Figure 4). In addition, two different random instances (Figures 5 and 6) were also generated with varying values of the following parameters – the number of teams, the number of stations, the number of interchange stations (Table 1).

Table 1. Characteristics of Problem Instances

Problem Set	Number of Teams	Number of stations	Number of interchange stations	Number of time periods per day	Number of stations visited per team
Random1	4	20	1	16	6
Random2	5	24	2	16	6
Case Study	24	90	10	16	6

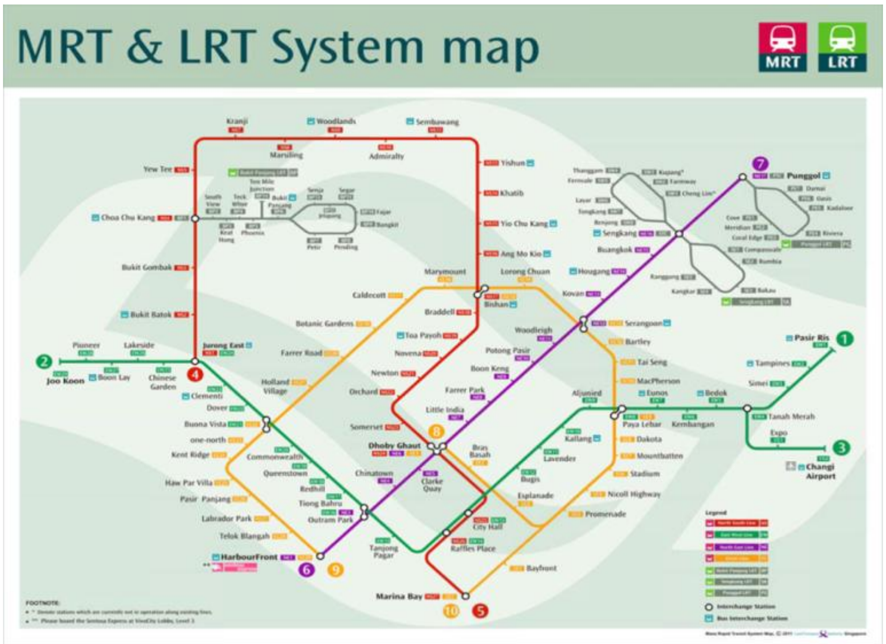

Figure 4. Singapore MRT map (source: http://www.smrt.com.sg/trains/network_map.asp)

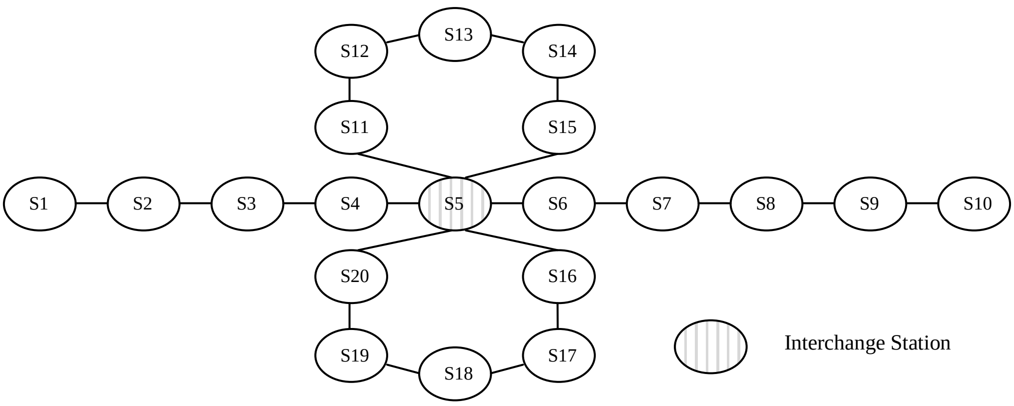

Figure 5. Random1 station map

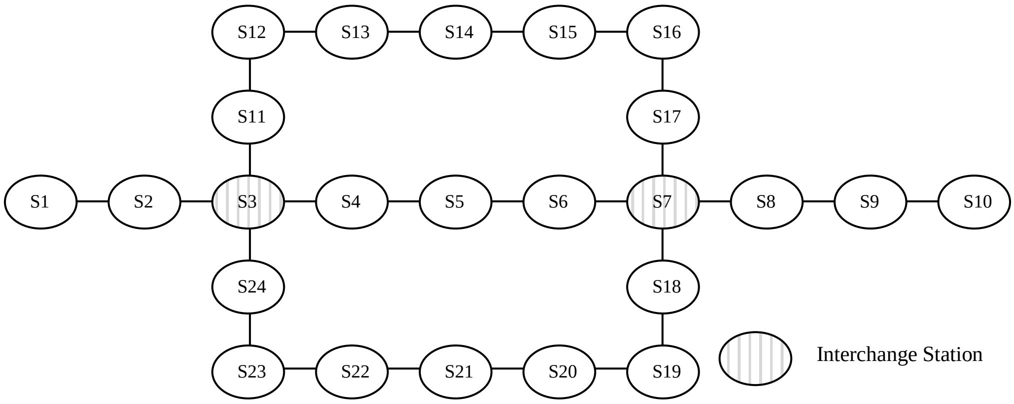

Figure 6. Random2 station map

In order to ensure the feasibility of random instances, the number of teams is set as:

$$|I| = \left| \left(\sum_{j \in J} Min\_Visit_j \right) / 6 \right| \tag{22}$$

For simplicity in our experiments, we assume that the break periods for all teams occur at periods $Start_i + 2$ and $Start_i + 5$, meaning that the break periods are at the third and sixth periods of an eight-period shift. In the following, we report a suite of computational results and analysis obtained from our mathematical model described above. We also conduct some additional experiments by varying the values of some parameters that would be described in Section 5.3.

5.2 Results of the Deterministic Model

*5.2.1 Results of Random*1 *and Random*2 *Instances*

Both Random1 and Random2 can be optimally solved by the CPLEX 10.0 solver engine. The following tables summarize the schedules of all teams. It is observed that all teams are not required to change to another line for both instances. This situation provides us the minimum total distance travelled for all teams. In Random1, three stations, S1, S5 and S10, are required to be visited twice a day. Here, we found that this requirement is satisfied. Similar observation can be obtained for Random 2 where S1, S3, S7 and S10 have to be visited twice as well. The total runtimes for both random instances are 111 and 155 seconds, respectively.

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

Figure 7. Result of Random1 instance

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

Figure 8. Result of Random2 instance

5.2.2 Results of Case Study

This is a large-scale problem with 90 stations, which could not be solved in CPLEX after 24 hours. We decompose the problem into four sub-problems where each sub-problem represents a single line. In our case study, there are four different lines, namely the East-West Line, North-South Line, North-East Line and Circle Line (Table 2). The number of teams allocated to each line is defined by equation (22). The details of lines and station names can be found in http://www.smrt.com.sg. In this network, some stations are interchange stations (such as Jurong East, Dhoby Ghaut, Buona Vista stations and so on) which serve more than one lines. By solving each line separately, there is a possibility that these interchange stations could be visited by more than one teams at the same time. This situation is acceptable since an interchange station is generally a large station with multiple platforms. For instance, the Buona Vista station has two different platforms for trains serve East West and Circle Lines. On the other hand, we ensure that other stations (non-interchange stations) may only be visited by at most one team at a particular time period.

Figure 9 summarizes the detailed route taken by each team for the entire network. In general, stations visited by each team are close to each other (in accordance with the minimum total distance objective we define for our model). We divide the number of teams for each line into two different groups, Groups I and II. The teams in Group I would start their duty at time period 1 while others in Group II would be at time period 9 (represent two different shifts). The runtimes for each line are as follows: 780 seconds (East West Line), 200 seconds (North South Line), 33 seconds (North East Line), and 8,322 seconds (Circle Line). Solving the mathematical model for the North East Line takes significantly less runtime than solving for the other lines since it has less number of teams and stations.

Table 2. Characteristics of lines in Case Study

Lines	Number of teams	Number of stations	Number of interchange stations	Number of time periods per day	Number of stations visited per team
East West Line	7	31	7	16	6
North South Line	6	25	7	16	6
North East Line	4	16	6	16	6
Circle Line	7	30	7	16	6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
East West Line																
1	Paya Lebar	Kallang	Break	Lavender	Bugis	Break	City Hall	Raffles Place								
2	Changi Airport	Expo	Break	Tanah Merah	Simei	Break	Tampines	Pasir Ris								
3	Tanah Merah	Bedok	Break	Kembangan	Eunos	Break	Paya Lebar	Aljuneid								
4	Chinese Garden	Jurong East	Break	Clementi	Dover	Break	Buona Vista	Commonwealth								
5									Jurong East	Chinese Garden	Break	Lakeside	Boon Lay	Break	Pioneer	Joo Kon
6									Redhill	Tiong Bahru	Break	Outram Park	Tanjong Pagar	Break	Raffles Place	City Hall
7									Buona Vista	Commonwealth	Break	Queenstown	Redill	Break	Tiong Bahru	Outram Park
North South Line																
1	Marina Bay	Raffles Place	Break	City Hall	Dhoby Ghaut	Break	Somerset	Orchard								
2	Kranji	Yew Tee	Break	Choa Chu Kang	Bukit Gombak	Break	Bukit Batok	Jurong East								
3	Bishan	Ang Mo Kio	Break	Yio Chu Kang	Khatib	Break	Yishun	Sembawang								
4									Newton	Novena	Break	Toa Payoh	Braddell	Break	Bishan	Ang Mo Kio
5									Admiralty	Woodlands	Break	Marsiling	Choa Chu Kang	Break	Bukit Batok	Jurong East
6									Orchard	Somerset	Break	Dhoby Ghaut	City Hall	Break	Raffles Place	Marina Bay
North East Line																
1	Harbour Front	Outram Park	Break	Chinatown	Clarke Quay	Break	Dhoby Ghout	Little India								
2	Potong Pasir	Woodleigh	Break	Serangoon	Buangkok	Break	Sengkang	Punggol								
3									Serangoon	Kovan	Break	Hougang	Buangkok	Break	Sengkang	Punggol
4									Boon Keng	Farrer Park	Break	Dhoby Ghaut	Clarke Quay	Break	Outram Park	Harbour Front
Circle Line																
1	Holland Village	Buona Vista	Break	One North	Kent Ridge	Break	Labrador Park	Harbour Front								
2	Harbour Front	Telok Blangah	Break	Labrador Park	Pasir Panjang	Break	Haw Par Villa	Buona Vista								
3	Dhoby Ghaut	Bras Basah	Break	Esplanade	Promenade	Break	Bay Front	Marina Bay								
4	Bishan	Lorong Chuan	Break	Serangoon	Bartley	Break	Tai Seng	Paya Lebar								
5									Farrer Road	Botanic Garden	Break	Caldecott	Marymount	Break	Bishan	Serangoon
6									Dhoby Ghaut	Bras Basah	Break	Esplanade	Promenade	Break	Bay Front	Marina Bay
7									Nicoll Highway	Stadium	Break	Mountbatten	Dakota	Break	Paya Lebar	Mac Pherson

Figure 9. Result of Case Study

5.3 Results of the Randomized Strategy

First, we generate different instances by randomizing the start and break times (Table 3). Note that the number of teams allocated has to be adjusted according to equation (22) in order to ensure feasibility. Scenario1 is the base problem that has been solved and shown in Figure 9. Scenario 2 is generated by varying the start time for each team. The same start time for all teams is set for Scenario 3. Finally, we increase the number of visits for some stations in Scenario 4. It turns out that the number of teams has to be increased by one team. We present and discuss the results of the East West line of the Singapore network.

Table 3. Randomized start and finish times

East West Line	Number of teams	Start and finish times for each team (team no [start – finish])
Scenario 1	7	1[1-8]*, 2[1-8], 3[1-8], 4[1-8], 5[9-16], 6[9-16], 7[9-16]
Scenario 2	7	1[1-8], 2[1-8], 3[3-10], 4[3-10], 5[7-14], 6[9-16], 7[9-16]
Scenario 3	7	1[1-8], 2[1-8], 3[1-8], 4[1-8], 5[1-8], 6[1-8], 7[1-8]
Scenario 4	8	1[1-8], 2[1-8], 3[3-10], 4[4-11], 5[5-12], 6[6-13], 7[8-15], 8[9-16]

*1[1-8]: Team 1 would start at time period 1 and finish at time period 8

As mentioned earlier, Scenario 1 with only two values of start time periods: time periods 1 and 9, could be solved within 780 seconds. When the start time for each team is randomly set to several time periods (Scenario 2), it turns out that the problem could be solved faster (within 629 seconds). On the other hand, if all teams have to patrol at the same time (Scenario 3), the runtime significantly increases to 6,400 seconds. When we increase the number of visits for some stations and change the start time for each team randomly (Scenario 4), the runtime is up to 17,078 seconds. Increasing the number of visits seems to make the problem harder. Similar observations can be obtained for other lines.

The next set of experiments is related to randomizing the break times. Initially, we assume that the break periods for all teams occur at periods $Start_i + 2$ and $Start_i + 5$, where the values of $Break_i^1$ and $Break_i^2$ are 2 and 5, respectively. Table 4 summarizes different values of the break times. In Scenario 5, each team might have different values of $Break_i^1$ and $Break_i^2$ with a constant gap between both values ($Break_i^2$ - $Break_i^1$ = 3), while in Scenario 6, the gap is not constant.

Table 4. Randomized $Break_i^1$ and $Break_i^2$

East West Line	Number of teams	$Break_i^1$ and $Break_i^2$ for each team (team no [$Break_i^1$ – $Break_i^2$])
Scenario 5	7	1[2&5]*, 2[2&5], 3[3&6], 4[3&6], 5[2&5], 6[2&5], 7[3&6]
Scenario 6	7	1[1&3], 2[2&5], 3[3&5], 4[2&5], 5[1&4], 6[2&5], 7[3&6]

1[2&5]: Team 1 would have two breaks at ($Start_1$*+2) and (*$Start_1$*+5)

The runtime for both scenarios, Scenario 5 and Scenario 6, are 33,240 and 28,017 seconds respectively. It seems that both scenarios are more difficult to solve compared with the previous scenarios in Table 3. When the break times for each team are randomly set to different values (Scenario 6), the runtime is less than that of Scenario 5.

Finally, we report results on randomizing the visit frequencies. As described in Section 4.2, assuming that the probability distribution of a crime is known, our goal is to simulate a number of realizations of crime occurrences based on this distribution, and evaluate the effectiveness of our solutions in crime deterrence against those generated by a fixed visit frequency. For this purpose, we choose the North East Line (with a total of 16 stations). The probability distribution *P*(crime occurs at Station) is presented in Table 5 column 3. We generate 100 realizations, and for each realization, the randomized vector for the minimum number of visits required for each station (*Min_Visit*) is as shown in Table 5.

Table 5. Simulating crime and visits at different stations

Station	Station Name	Pr(crime occurs at Station)	*Min_Visit* (1)	*Min_Visit* (2)	…	*Min_Visit* (100)
1	Harbour Front	0.08	2	3	…	0
2	Outram Park	0.07	1	2	…	0
3	China Town	0.02	1	1	…	1
4	Clarke Quay	0.09	1	1	…	3
5	Dhoby Ghaut	0.13	3	2	…	3
6	Little India	0.06	1	1	…	1
7	Farrer Park	0.06	1	1	…	1
8	Boon Keng	0.04	1	1	…	0
9	Potong Pasir	0.04	1	1	…	2
10	Woodleigh	0.07	2	1	…	4
11	Serangoon	0.10	1	4	…	1
12	Kovan	0.08	4	1	…	3
13	Hougang	0.04	0	1	…	2
14	Buangkok	0.04	1	1	…	1
15	Sengkang	0.06	1	1	…	2
16	Punggol	0.04	3	2	…	0
	Total	1	24	24	…	24

For each realization, we randomly generate the station where the crime occurs (according to the probability distribution). The randomized strategy is said to effectively deter the crime occurring that station if the *Min_Visit* value for that station exceeds that of the fixed strategy, and vice versa; otherwise, we have a tie. For convenience, we set the *Min_Visit* vector for the fixed strategy to be the *Min_Visit* vector of the first realization (i.e. Table 5 column 4). The entire simulation results are summarized in Table 6. In this table, we set the value of 1 for a particular replicate if the randomized model is more effective; otherwise 0. If a tie exists, we use the word "tie".

We observe that our randomized strategy can perform better than the fixed strategy which is based on the fixed strategy. Of the 100 replicates, the randomized strategy provides 53% successful deterrence versus 25% for the fixed strategy. Both are tied at 22% of the runs.

Table 6. Simulation results for North East Line

Run	Crime at station	*Min_Visit* Randomized Strategy	*Min_Visit* Fixed Strategy	Result	Run	Crime at station	*Min_Visit* Randomized Strategy	*Min_Visit* Fixed Strategy	Result
1	5	3	3	tie	51	5	5	3	1
2	11	4	1	1	52	15	2	1	1
3	12	5	4	1	53	4	1	1	tie
4	5	5	3	1	54	11	3	1	1
5	11	3	1	1	55	1	2	2	tie
6	15	1	1	tie	56	5	3	3	tie
7	1	3	2	1	57	15	0	1	0
8	10	3	2	1	58	10	2	2	tie
9	15	3	1	1	59	4	3	1	1
10	9	0	1	0	60	6	1	1	tie
11	5	3	3	tie	61	4	3	1	1
12	4	3	1	1	62	3	1	1	tie
13	4	3	1	1	63	12	4	4	tie
14	1	1	2	0	64	5	6	3	1
15	4	4	1	1	65	15	0	1	0
16	1	3	2	1	66	15	4	1	1
17	13	1	0	1	67	15	1	1	tie
18	5	5	3	1	68	5	2	3	0
19	13	0	0	tie	69	11	2	1	1
20	11	5	1	1	70	14	1	1	tie
21	5	8	3	1	71	13	1	0	1
22	16	1	3	0	72	1	3	2	1
23	5	4	3	1	73	14	2	1	1
24	16	4	3	1	74	2	3	1	1
25	2	0	1	0	75	15	4	1	1
26	5	4	3	1	76	9	3	1	1
27	10	2	2	tie	77	12	3	4	0
28	7	3	1	1	78	12	3	4	0
29	5	1	3	0	79	14	4	1	1
30	12	1	4	0	80	14	2	1	1
31	12	1	4	0	81	10	3	2	1
32	4	4	1	1	82	5	2	3	0
33	10	1	2	0	83	12	3	4	0
34	11	3	1	1	84	7	1	1	tie
35	5	2	3	0	85	4	0	1	0
36	12	0	4	0	86	12	4	4	tie
37	5	5	3	1	87	14	1	1	tie
38	13	2	0	1	88	15	1	1	tie
39	11	4	1	1	89	5	2	3	0
40	2	1	1	tie	90	12	0	4	0
41	1	2	2	tie	91	4	3	1	1
42	13	1	0	1	92	5	4	3	1
43	5	5	3	1	93	4	2	1	1
44	16	2	3	0	94	4	4	1	1
45	12	2	4	0	95	5	6	3	1
46	7	1	1	tie	96	11	3	1	1
47	10	1	2	0	97	5	4	3	1
48	4	3	1	1	98	1	3	2	1
49	5	5	3	1	99	11	0	1	0
50	11	0	1	0	100	14	1	1	tie

6 Conclusion

In this paper, we presented initial results from a research on generating patrol scheduling in mass rapid transit systems. We proposed a mathematical programming model to formulate the problem. Security is a major importance issue in the patrol scheduling problem. 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 a simple randomized strategy by randomizing the start (and therefore finish times), break times for each team and the frequency of visits for each station. We reported the efficiency and effectiveness of our proposed approach under different circumstances. 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. The random start 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 do 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., 2012, Tsai et al., 2009).

References

1. Basar, T., & Olsder, G.J. (1995). Dynamic Noncooperative Game Theory. Academic Press, San Diego, CA, 2nd edition.
2. Chu, S.C.K, & Chan, E.C.H. (1998). Crew scheduling of light rail transit in Hong Kong: from modeling to implementation. *Computers and Operations Research*, 25 (11), 887-894.
3. 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.
4. Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, *153*, 3-27.
5. Gunawan, A., & Lau, H.C. (2012). Master physician scheduling problem. *Journal of the Operational Research Society*. To appear.
6. Hammer, P.L., & Rudeanu, S. (1968). Boolean Methods in Operations Research and Related Areas. Springer, Berlin.
7. Jiang, A.X., Yin, Z., Johnson, M.P, Tambe, M., Kiekintveld, C. Leyton-Brown, K., & Sandholm, T. (2012). Towards optimal patrol strategies for fare inspection in transit systems. In AAAI Spring Symposium on Game Theory for Security, Sustainability and Health, Stanford, California, 26-28 March 2012.
8. Maenhout, B., & Vanhoucke, M. (2010). A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *European Journal of Operational Research*, 206, 155-167.
9. Ordóñez, F., Tambe, M., Jara, J.F., Jain, M., Kiekintveld, C., & Tsai, J. (2012). Deployed security games for patrol planning. In Handbook of Operations Research for Homeland Security (Book Chapter), ed. Herrmann, J.W., Springer. To appear.
10. Petrovic, S., & Berghe, G.V. (2008). Comparison of algorithms for nurse rostering problems. In Proceedings of the 7th International Conference of Practice and Theory of Automated Timetabling 2008, Montreal, Canada, 18 – 22 August 2008.

11. Rosenshine, M. (1970). Contributions to a theory of patrol scheduling. *Operational Research Quarterly* (*1970-1977*), *21*(*1*), 99-106.
12. Ross, S. (2009). A First Course in Probability. Prentice-Hall, Upper Saddle River, NJ, 8th edition.
13. Sharma, D.K., Ghosh, D., & Gaur, A. (2007). Lexicographic goal programming model for police patrol cars deployment in metropolitan cities. *Information and Management Sciences*, 18(2), 173-188.
14. Stern, Z.S., & Teomi, Y. (1986). Multi-objective scheduling plans for security guards. *The Journal of the Operational Research Society*, 37(1), 67-77.
15. 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.
16. Tsai, J., Rathi, S., Kiekintveld, C., Ordóñez, F., & Tambe, M. (2009). IRIS – A tool for strategic security allocation in transportation networks. In Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, 10-15 May 2009, 37-44.