

A Balanced Route Design for Min-Max Multiple-Depot Rural Postman Problem (MMMDRPP): a police patrolling case

Huanfa Chen, Tao Cheng & John Shawe-Taylor

To cite this article: Huanfa Chen, Tao Cheng & John Shawe-Taylor (2017): A Balanced Route Design for Min-Max Multiple-Depot Rural Postman Problem (MMMDRPP): a police patrolling case, International Journal of Geographical Information Science, DOI: [10.1080/13658816.2017.1380201](https://doi.org/10.1080/13658816.2017.1380201)

To link to this article: <http://dx.doi.org/10.1080/13658816.2017.1380201>



© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 11 Oct 2017.



Submit your article to this journal [↗](#)



Article views: 86



View related articles [↗](#)



View Crossmark data [↗](#)



A Balanced Route Design for Min-Max Multiple-Depot Rural Postman Problem (MMMDRPP): a police patrolling case

Huanfa Chen^a, Tao Cheng ^a and John Shawe-Taylor ^b

^aSpaceTimeLab, Department of Civil, Environmental & Geomatic Engineering, University College London, London, UK; ^bDepartment of Computer Science, University College London, London, UK

ABSTRACT

Providing distributed services on road networks is an essential concern for many applications, such as mail delivery, logistics and police patrolling. Designing effective and balanced routes for these applications is challenging, especially when involving multiple postmen from distinct depots. In this research, we formulate this routing problem as a Min-Max Multiple-Depot Rural Postman Problem (MMMDRPP). To solve this routing problem, we develop an efficient tabu-search-based algorithm and propose three novel lower bounds to evaluate the routes. To demonstrate its practical usefulness, we show how to formulate the route design for police patrolling in London as an MMMDRPP and generate balanced routes using the proposed algorithm. Furthermore, the algorithm is tested on multiple adapted benchmark problems. The results demonstrate the efficiency of the algorithm in generating balanced routes.

ARTICLE HISTORY

Received 11 May 2017

Accepted 13 September 2017

KEYWORDS

Arc routing; rural postman problem; tabu search; police patrol

1. Introduction

Providing distributed services on road networks is an essential concern for many applications, such as mail delivery, logistics and police patrolling. For example, a production company may pick up materials from different places and deliver the products to the customers. Similarly, a security service may manage the patrolling activities of a region conducted by several police stations. Obviously, a well-designed route can reduce the cost of the distributed service substantially. For these reasons, routing problems have been an essential area of operations research, and are receiving increasing attention in both research and applications (Dror 2000, Corberan and Laporte 2014).

Routing problems can be divided into two categories, namely node routing problems (NRPs) and arc routing problems (ARPs). The major difference concerns where the demands for service are located. In NRPs, the demands are on the nodes of the network, hence the objective to traverse the nodes. In contrast, in ARP, the demands are on the edges and the goal is to traverse the edges. Therefore, NRP and ARP require different models of the network, and algorithms for one category cannot be directly applied for the other.

The study of ARP dates back to Leonhard Euler and his notable Königsberg seven-bridge problem (Euler 1736). To date, ARP has evolved into a rich research area in

combinatorial optimisation. Important applications of ARP include mail delivery, garbage collection, snow removal and security guard patrolling. A thorough review of ARPs can be found in Corberan and Laporte (2014), Dror (2000) and Eiselt *et al.* (1995). Since we are dealing with applications where the demands are on the edges, such as police patrolling, we will focus on ARP.

Two basic and important ARPs are the well-known Chinese Postman Problem (CPP) and the Rural Postman Problem (RPP). Both problems are aimed at determining a minimum length cycle for a single postman. The CPP requires that the complete edge set E has to be covered at least once, and the RPP requires only a subset of E to be covered.

The CPP/RPP has been generalised to use multiple postmen and have varying objectives, as practically more than one postman or vehicle would be serving a medium-size or large-size area. A common objective is to minimise the sum of all route costs (Ulusoy 1985, Akyurt *et al.* 2015). However, minimising the sum of all route costs for multiple postmen may be inefficient and unfair, as it neglects the workload distribution. In the worst case, one postman would service all required edges while all other postmen just leave the depot and return right away. Therefore, a balance objective that guarantees similar route lengths for all postmen is preferable.

This balance objective can be represented by different functions. Degenhardt (2004) and Osterhues and Mariak (2005) suggested functions that use a predefined allowed time for each postman: minimum absolute deviation function minimising the sum of all deviations from the allowed service time, minimum square deviation function minimising the square sum of the deviations from the allowed service time and minimum overtime function minimising the sum of overtime for all postmen. These functions, however, are limited to the applications where the allowed time can be predefined. Furthermore, the balance objective can also be represented by minimising the deviation of all route lengths from the average length. While this function does not require predefined allowed service time, it may lead to excessively long and inefficient routes. In the worst case, all postmen follow the same cycle to traverse all required edges, which has the minimal route length deviation but is a waste of resources. Finally, Frederickson *et al.* (1976) and Ahr and Reinelt (2002) proposed a min-max objective function to minimise the length of the longest route. This function not only ensures the fair scheduling and balanced routes but also increases the overall route efficiency and serves the demands as early as possible (Ahr and Reinelt 2002). Thus, this paper focuses on the min-max objective function.

The min-max RPP (MMRPP) or min-max CPP (MMCPP) has been studied by several authors (Ahr and Reinelt 2006, Benavent *et al.* 2009, Willemse and Joubert 2012), but all of them are limited to a single depot for all postmen. So far, there is no study on the MMRPP/MMCPP in the multiple-depot case. This problem is, however, critical for cases in which vehicles or postmen have to start from different depots and balanced task distribution is required. As an example, in a city or a district, multiple police stations are evenly located in the area in order to respond quickly to potential calls for service, and police patrollers belong to different stations as well. To cover and monitor a set of important locations in the area, the design of efficient patrol routes should consider two factors: patrollers start from different stations before patrolling, and the patrol routes should be balanced in order to avoid potential work overload and dissatisfaction.

Therefore, it is necessary to study the combination of MMRPP/ CPP and multiple-depot RPP/ CPP.

In the wider field of routing problems, a similar problem has been investigated, which is called Min-Max Multi-Depot Vehicle Routing Problem (Carlsson *et al.* 2009), and a linear programming-based algorithm for it was developed. However, as vehicle routing aims at covering the nodes instead of the arcs in RPP/ CPP, the algorithm is not applicable for the multiple-depot RPP/ CPP.

The aim of this paper is to formulate and solve the Min-Max Multiple-Depot Rural Postman Problem (MMMDRPP), and to apply this problem to route design of police patrolling. Here we use MMMDRPP to represent both MMMDRPP and MMMDCPP. Specifically, our contributions are as follows:

- We introduce a new ARP, namely Min-Max Multiple-Depot Rural Postman Problem (MMMDRPP), which can generate balanced routes for postmen from different depots.
- We propose a routing algorithm TABU-PATROL that approximately solves MMMDRPP.
- To approximate the optimal solution value of MMMDRPP, we propose three lower bounds.
- We model the route design of police patrolling as the MMMDRPP, and demonstrate how to design and choose routes for police patrolling in practice using the proposed algorithm.
- We further test the TABU-PATROL algorithm with multiple benchmark routing problems, and show that TABU-PATROL is robust to generate quality solutions on different instances.

This paper is organised as follows. The next section reviews the approaches and solution evaluations for MMRPP/ MMCPP. In [Section 3](#), we focus on the problem definition of MMMDRPP and present a tabu search algorithm to solve this problem. Following that, we devise three innovative lower bounds and the solution evaluation for this problem in [Section 4](#). In [Section 5](#), we demonstrate how to model the route design of police patrolling using the MMMDRPP, and how to choose routes for implementation using a case study in Camden. In [Section 6](#), the proposed algorithm is further tested on several benchmark problems. The final section summarises the major findings and future research directions.

2. Background and related work

2.1. Approaches for min-max postmen problem

As MMMDRPP can be treated as a generalisation of MMRPP, approaches for MMRPP may be adapted to solve MMMDRPP. In this part, we start with the computational complexity of CPP/ RPP/ MMCPP/ MMRPP, then review the existing approaches for the MMRPP, and discuss their applicability to MMMDRPP.

CPP and RPP have different computational complexities. For CPP, the optimal solution can be achieved in polynomial time from a matching algorithm (Edmonds and Johnson

1973). In contrast, RPP is non-deterministic polynomial-time hard (NP-hard), implying that it is computationally challenging to find the optimal solution in polynomial time (Lenstra and Kan 1976). RPP can be approximately solved by a heuristic algorithm developed by Christofides *et al.* (1981).

Both MMCPP and MMRPP are NP-hard. MMCPP is proved as NP-hard as the classic NP-hard k -partition problem is reducible to MMCPP (Frederickson *et al.* 1976). Meanwhile, MMRPP is NP-hard, as MMRPP with a single postman reduces to the classic RPP, which is NP-hard. Since MMCPP is a special case of MMRPP, these two problems are very similar in NP-hardness and solution approaches. For convenience, we use MMRPP to represent both MMRPP and MMCPP. Approaches for MMRPP can be classified into three categories: exact approaches, approximation approaches and heuristic approaches.

Exact approaches are guaranteed to find at least one of the optimal solutions, but their time complexity may be exponential (Bertsimas and Tsitsiklis 1997). As MMRPP is NP-hard, the corresponding exact approaches are quite limited. Ahr (2004) proposed a branch-and-cut algorithm for MMCPP, and Benavent *et al.* (2009) developed another branch-and-cut algorithm for the MM k -Windy RPP. The MM k -Windy RPP is an extension of MMRPP in that each edge is given different traversal costs for the two directions. The algorithm for MM k -Windy RPP is capable of solving small- and medium-size instances with up to 78 required edges in a few minutes (Benavent *et al.* 2009). However, these two approaches fail to solve some difficult instances within specified time limits. Essentially, the excessive computation time incurred by such exact methods does restrict their applicability to small problems effectively excluding large-size and complicated problems, such as the route design of police patrol in the road network in our application.

Approximation approaches provide a sub-optimal solution in polynomial time, along with a bound on the degree of sub-optimality. The only such approach for MMRPP is the 7-approximation algorithm by Arkin *et al.* (2006). This algorithm aims at minimising the length of the longest route. Assuming that the optimal solution value is λ^* , this algorithm receives a guess λ , and would either construct a solution with the cost at most seven times λ or claim that λ is not a feasible solution value and λ is smaller than λ^* . Therefore, a binary search in combination with this procedure would lead to a solution with the cost at most $7\lambda^*$. However, this algorithm assumes that the depot of each postman is arbitrary, thus making it unsuitable for applications with a predefined depot for each postman.

Heuristic algorithms usually generate a suboptimal solution without a guarantee on its quality. Empirically, some of these algorithms return a good solution fast, although no guarantee on the polynomial computation time. Given that MMRPP is NP-hard, most solution strategies for these problems are based on heuristic or metaheuristic methods. A typical process to solve MMRPP follows two stages: (1) a constructive method to generate initial solutions, (2) a procedure to improve the solutions.

The first constructive method for MMCPP was proposed by Frederickson *et al.* (1976). Subsequently, two methods for the same problem were proposed by Ahr and Reinelt (2002): the Augment-Merge heuristic and the Cluster heuristic. These methods with proper improved heuristics obtained a better solution in most instances than the Frederickson heuristic. Later, a Parallel-Insertion method (henceforth referred to as PI) was proposed to solve both MMCPP and MMRPP by Willemse and Joubert (2012). Essentially, this algorithm generates multiple routes that traverse the required edges

that are farthest from the depot, and then adds each untraversed required edge to an existing route such that the resulting increase of route length is minimised. This is valid if all tours share the same depot. However, in the case of multiple depots, this algorithm may generate excessively long routes, thus making it unsuitable for MMMDRPP.

Most of the procedural improvements for MMRPP are based on the tabu search approach. Tabu search attempts to iteratively find new solutions possibly with shorter length of the longest route by reallocating the edges amongst routes.

Tabu search is a local search-based method used for mathematical optimisation. It starts from a feasible solution and iteratively attempts to improve it by checking its immediate neighbours, which are only slightly different from the current solution. To avoid being stuck in suboptimal regions or areas with many equally fit solutions, tabu search uses memory to guide the search, and discourages coming back to previously visited solutions. At each iteration, worsening moves can be accepted if no acceptable and improving move exists. Tabu search was proposed by Glover (1986), and a detailed description of tabu search is given by Glover and Laguna (1997).

To improve solutions of MMRPP, Ahr and Reinelt (2006) devised a tabu search procedure. Later, Willemsse and Joubert (2012) introduced a tabu search algorithm TABU-GUARD, which can be applied to both MMCPP and MMRPP. TABU-GUARD outperformed the other strategies and achieved the best solutions for MMCPP test instances. Moreover, it is the only existing solution algorithm for the MMRPP.

Meanwhile, tabu search methods have been applied to a similar problem to the MMRPP, the Capacitated Arc Routing Problem (CARP). CARP is a generalisation of RPP, in which each required edge is given a non-negative demand and must be traversed by a postman. CARP aims at determining a set of closed walks with the minimum sum of edge weights, where the sum of demands on edges traversed by each walk does not exceed the given capacity.

There are many tabu-search-based algorithms to solve CARP. Hertz *et al.* (2000) developed a tabu search algorithm called CARPET to solve CARP, and later Greistorfer (2003) proposed a tabu scatter search metaheuristic, which is a hybrid approach and is competitive with CARPET. Moreover, a deterministic tabu search algorithm for CARP was introduced by Brandão and Eglese (2008), which found best solutions for some benchmark instances. It is also worth noting that a tabu search algorithm using capacitated trees has been proposed to solve the Multiple Centre CARP (Amberg *et al.* 2000). However, as CARP is different from MMRPP, these existing algorithms cannot be applied directly to MMRPP, or the MMMDRPP.

2.2. Solution evaluation for min-max postmen problem

Given that MMRPP is NP-hard, it is computationally unfeasible to find the optimal solution. Typically, a solution of MMRPP is evaluated by comparing with the lower bounds of MMRPP. Informally, an MMRPP is said to have a lower bound c if c is no larger than the length of the longest route in any solution. To this end, we review the lower bounds that have been developed for MMRPP and discuss how to adapt these lower bounds for MMMDRPP.

Three lower bounds have been developed for MMRPP (Ahr and Reinelt 2002). First, the Shortest Path Tour Lower Bound (SPT-LB) comes from the observation that in an optimal solution, the farthest required edge from the depot must be traversed. Thus, the

longest route must be no shorter than the shortest route to traverse u from the depot. In the case of different depots, it is unclear which edge is the farthest from the depots. Therefore, these lower bounds cannot be directly used for MMMDRPP.

Second, the CPP Tour Lower Bound is computed by finding the optimal 1-postmen route and then divide the length by k . This lower bound is only valid for MMCP, and the MMRPP where the required edges constitute a connected sub-graph. To our knowledge, there is no research on extending this lower bound for the MMRPP with disconnected R , since it would involve solving the NP-hard RPP. Therefore, it is challenging to apply this lower bound for MMMDRPP.

Third, the Integer Programming Relaxation Lower Bound (IPR-LB) is computed by solving a relaxed linear programming formulation for MMRPP. This lower bound can directly be applied for MMMDRPP, as it does not consider the depot nodes.

3. A TABU-PATROL algorithm for MMMDRPP

In this section, we first formulate the MMMDRPP, then briefly review the TABU-GUARD algorithm for MMRPP and propose a TABU-PATROL algorithm for MMMDRPP, which is based upon TABU-GUARD. Hereafter, we use the terms route and tour interchangeably.

3.1. Problem definition

In the formulation of MMMDRPP, a transportation network is represented as a connected graph consisting of nodes and edges. Each node represents a street intersection in the geographic space, and each edge between two nodes represents a street segment with a travelling distance.

Each required edge must be traversed at least once by some route. A traversal is defined as a pair (e, C_i) , where e is an edge and C_i is a route. In a solution, an edge may be traversed by multiple routes and for multiple times. To simplify the MMMDRPP formulation, given a required edge e , we distinguish two types of traversal, namely a service and a traversal without service. A service is randomly selected from all traversal of e , and e must be serviced once and only once.

The MMMDRPP is defined as follows:

Input: A connected and undirected graph G with

- A set of nodes V , and a set of edges E ,
- A set of required edges $R \subseteq E$,
- A non-negative distance function for edges $f: E \rightarrow \mathbb{R}_{\geq 0}$,
- The number of postmen k and a list of k depots D . The depots may be different.

Output: A set of k closed routes, T .

Objective: Minimise the distance of the longest route of T .

Constraints:

- Closed route constraint: the k th route of T should be a closed route and traverse the k th depot of D ;
- Traversal constraint: each edge in R should be traversed at least once by T .

A mathematical model of MMMDRPP is formulated as follows, which is adapted from the MMRPP (Ahr and Reinelt 2002):

$$\min L \quad (1)$$

subject to:

$$L \in \mathbb{R}^+ \quad (2)$$

$$\sum_{e \in E} w(e)x^i(e) + w(e)y^i(e) \leq L, \quad \forall i = 1, \dots, k \quad (3)$$

$$x^i(e) \in \{0, 1\}, \quad y^i(e) \in \mathbb{N}_0, \quad \forall e \in E, i = 1, \dots, k \quad (4)$$

$$\sum_{i=1}^k x^i(e) = 1, \quad \forall e \in R \quad (5)$$

$$\sum_{e \in \delta(v)} x^i(e) + y^i(e) \equiv 0 \pmod{2}, \quad \forall v \in V, i = 1, \dots, k \quad (6)$$

$$x^i(\delta(S)) + y^i(\delta(S)) \geq 2^*x^i(e), \quad \forall S \subseteq V \setminus d_i, e \in E(S), i = 1, \dots, k \quad (7)$$

The binary decision variable $x^i(e)$ equals 1 if edge e is serviced by tour C_i and $e \in R$. The integer variable $y^i(e)$ represents number of times that edge e is traversed by C_i without being serviced. $\delta(v)$ denotes the set of edges that are incident to the node v , and $\delta(S)$ denotes the set of edges that have one and only one endpoint belonging to the node set S . $E(S)$ denotes the set of required edges whose both endpoints belong to the node set S . L_{\max} is defined as the positive length of the longest patrol route, and is expressed by constraints (2) and (3). Constraint (5) guarantees that each required edge is serviced. Constraint (6) enforces that each single route must be a closed walk. It requires that the number of times any node is traversed by any route must be even. If the number is odd, the route cannot form a closed walk. Constraint (7) ensures that the depot is traversed. For any route and any sub-graph $(S, E(S))$ excluding the depot node, if any edge in the sub-graph is serviced ($x^i(e) = 1$), this route must 'enter' or 'leave' the sub-graph at least twice. Otherwise, the route will probably fail to contain its depot. For example, a route would violate constraint (7) if it traverses only one required edge and excludes the depot.

The integer program of MMMDRPP is extremely difficult to solve directly. Assuming that there are m required edges ($|R| = m$) and k postmen, there would be approximately 2^m sub-graphs and more than $2^m \times k$ inequalities from constraint (7).

The MMMDRPP is an NP-hard problem, which can be proved in two steps. First, the MMRPP is NP-hard, as RPP reduces to MMRPP with a single postman, with RPP known to be NP-hard (Lenstra and Kan 1976). Second, the MMRPP reduces to MMMDRPP with only one depot.

A feasible solution for the MMMDRPP is a set T of k tours $T = \{C_1, \dots, C_k\}$. We denote by $w(C_i)$ the length of a route C_i , which is the total length of its traversed edges. For a feasible solution T , $w_{\max}(T) = \max(w(C_i))$ denotes the length of the longest route. The objective of the MMMDRPP is to find a solution T^* with the minimum $w_{\max}(T)$.

$SP(u, v)$ denotes the shortest path between two nodes u and v . This can be calculated by Dijkstra's algorithm (Dijkstra 1959). The shortest path between two edges $e_1 = (u_1, u_2)$ and $e_2 = (v_1, v_2)$, which is called $SP(e_1, e_2)$, is defined as the shortest path out of $\{SP(u_1, v_1), SP(u_1, v_2), SP(u_2, v_1), SP(u_2, v_2)\}$. Note that $SP(e_1, e_2)$ does not include e_1 and e_2 . Furthermore, $SPR(e_1, e_2)$ denotes the set of required edges that are contained in $SP(e_1, e_2)$.

3.2. TABU-GUARD for MMRPP

TABU-GUARD was proposed to tackle MMRPP (including MMCPP) by Willemse and Joubert (2012). This algorithm follows three stages: (1) generating multiple random initial solutions by a PI method (see Introduction), (2) IMPROVE-SOLUTION, which iteratively attempts to shorten the longest route by exchanging or reallocating edges between different routes, (3) TABU-SEARCH, which further improves the solutions.

The TABU-SEARCH procedure has several components and different choices. First, in terms of neighbourhood exchange procedures to create neighbouring solutions, there are three choices: REMOVE-INSERT-NEIGHBOURHOOD (RIN); EXCHANGE-NEIGHBOURHOOD (EN); a combination of RIN and EN (RINEN). Second, there are three tabu list strategies to determine whether a move is forbidden: COMPLEX-TABU; SIMPLE-TABU; and SIMPLE-AGGRESSIVE-TABU. Third, TABU-SEARCH has two parameters: the maximum number of iterations with improvement t_{max} , and the tabu tenure α .

In the application of TABU-GUARD, the first step is to manually determine the component combination and select parameter values. Given a set of problem instances, nine different combinations of neighbourhood exchange procedures and tabu list strategies are tested over all instances, with α ranging from 1 to 16 and t_{max} fixed at 500. Then, the best three combinations with appropriate α range are chosen and used.

TABU-GUARD is not readily applicable to MMMDRPP, as it does not consider multiple depots. Specifically, if TABU-GUARD is used for MMMDRPP, the first step would be to generate k routes by assigning each depot with the farthest required edge. However, the resulting routes would be excessively long and overlap considerably with each other. For example, assuming there are two depots d_1 and d_2 as well as a required edge e_1 , where e_1 is the farthest required edge to d_1 and $distance(d_1, e_1) > distance(d_2, e_1)$, intuitively it would be more efficient to assign e_1 to d_2 rather than to d_1 . Therefore, it is necessary to develop new algorithms that are well-suited to MMMDRPP.

3.3. TABU-PATROL for MMMDRPP

Here, we propose the TABU-PATROL algorithm for MMMDRPP. TABU-PATROL is influenced by the TABU-GUARD, but it is different from TABU-GUARD in three aspects. First, it includes two novel constructive methods to generate the initial routes, which are better suited for the MMMDRPP than the PI of TABU-GUARD. Second, it contains only the most efficient neighbourhood exchange procedures and tabu list strategy, which makes the algorithm simpler than TABU-GUARD. Third, in contrast to the manual parameter selection in TABU-GUARD, TABU-PATROL uses an automatic and systematic parameter selection.

TABU-PATROL works in three stages (Figure 1). First, it produces a number of random initial solutions by calling GENERATE-INITIAL-SOLUTION. Second, the initial solutions are improved by calling IMPROVE-SOLUTION. Third, the solutions are further improved by a SIMPLE-TABU-SEARCH procedure, which includes automatic parameter selection.

3.3.1. Stage 1: generating initial solutions

The first stage of TABU-PATROL is to generate a number of different initial solutions using two new constructive methods. The first method, called TOUR-SPLIT-EXCHANGE (referred to as TSE), follows the route-first-cluster-second paradigm (Prins *et al.* 2014). It creates a giant tour to traverse all required edges, decomposes the tour into a given number of sub-tours and then improves the routes by exchanging stations. The other method, called CLUSTER-ROUTE-SPLIT (referred to as CRS), follows the cluster-first-route-second paradigm (Prins *et al.* 2014). It firstly forms clusters of required edges based on the distance to stations, then computes the tour to traverse the required edges within each cluster, and further splits the tours if necessary.

TSE is illustrated in Figure 2. Firstly, it creates a single tour (called a 1-tour) that traverses all required edges at least once. Constructing such a tour is an undirected Rural Postman Problem, and can be near-optimally solved by the heuristic of Christofides *et al.* (1981). The length of 1-tour is $w(1\text{-tour})$. The tour is encoded by the scheme of Willemse and Joubert (2012), in which only required edges are explicitly represented per route and it is assumed that the shortest path is used between consecutive required edges. Thus, 1-tour is denoted as $RPPT = \{r_1, SPR(r_1, r_2), r_2, \dots, r_{n-1}, SPR(r_{n-1}, r_n), r_n, SPR(r_n, r_1)\}$,

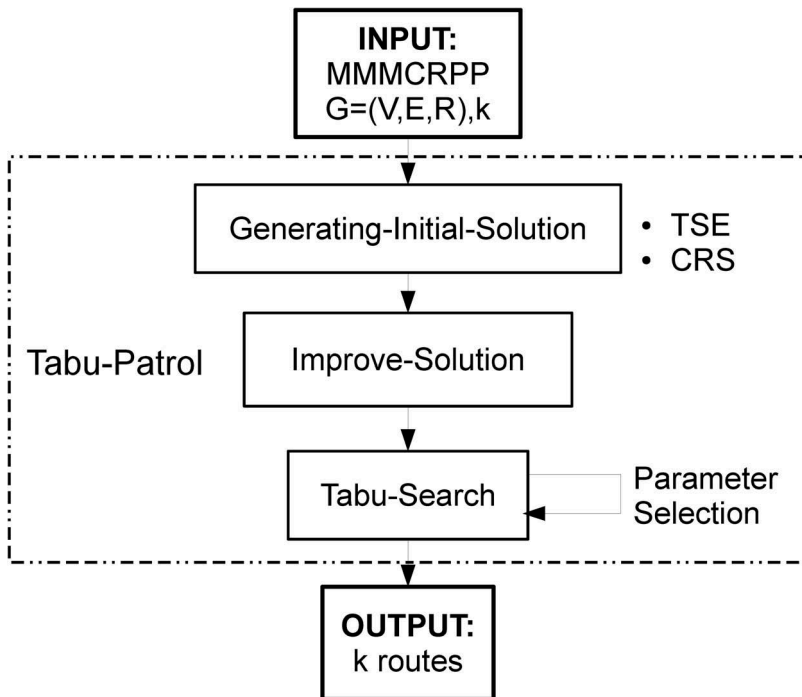


Figure 1. Flow chart of TABU-PATROL.

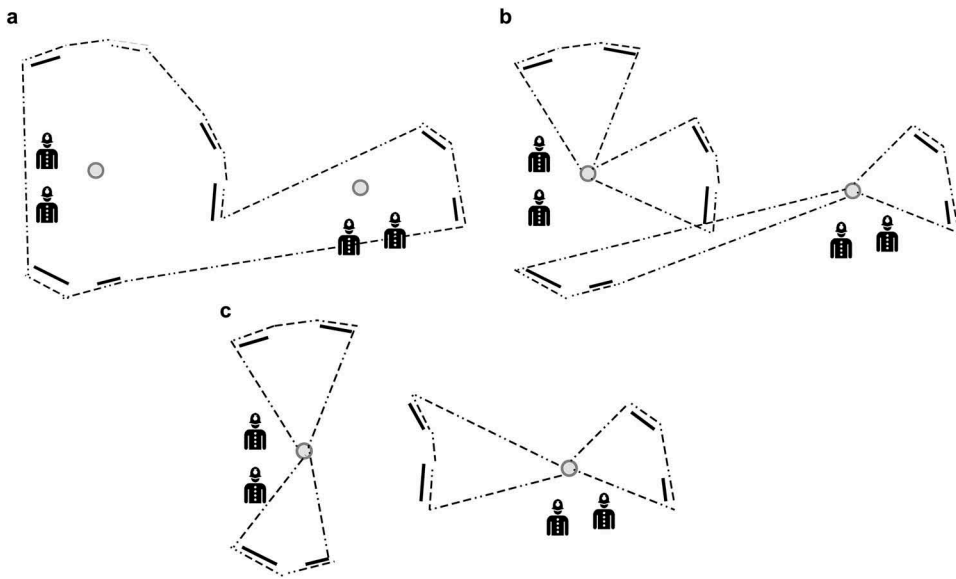


Figure 2. Illustration of Tour-Split-Exchange (TSE).

where $r_i R$. Then, we simplify 1-tour by only retaining the first position of any duplicated required edge.

Second, the obtained 1-tour is split into k partitions with similar lengths. Here, D' denotes the list of unused depots, and initially $D' = D$. The first splitting node is selected randomly on 1-tour, and then $(k-1)$ splitting nodes are determined to divide the tour into k approximately long segments $\{P_1, P_2, \dots, P_k\}$. The i th route $\{d, P_i, d\}$ is formed by finding the depot d from D' such that length of the route $\{d, P_i, d\}$ is (one of) the shortest, and then d is removed from D' . This step is repeated k times to form k routes.

The resulting solution $T = \{C_1, \dots, C_k\}$ is feasible for the MMMDRPP, as all required edges are traversed. It is then further improved by exchanging the depots between any pair of tours. If the exchange reduces the longer tour length, the depots are exchanged to form two new tours. This process is repeated until no improvements can be gained.

Since the first splitting node is chosen randomly and is different each time that TSE is invoked, the TSE is capable of generating different initial solutions.

The second new method CRS is illustrated in Figure 3. Given the list of station as D , We denote by $D_u = \{d_1, \dots, d_m\}$ the set of unique stations. For each $i \in [1, m]$, k_i is the number of occurrences of d_i in D . Then, the set of required edges is divided into m clusters $\{CL_1, \dots, CL_m\}$ based on the shortest distance to the depots, and the i th cluster corresponds to the i th depot d_i . Afterwards, for each cluster CL_i , k_i tours should be formed such that each tour traverses d_i and each edge in CL_i should be traversed at least once. This sub-problem can be solved by invoking the TSE procedure. As the random factor of TSE is brought into CRS, CRS is also capable of generating different initial solutions.

Both TSE and CRS take into account the different depots for postmen. TSE generates routes from different depots by assigning each partition of the 1-tour to the closest depot, and by exchanging the depots of routes to further shorten the tour length. In this

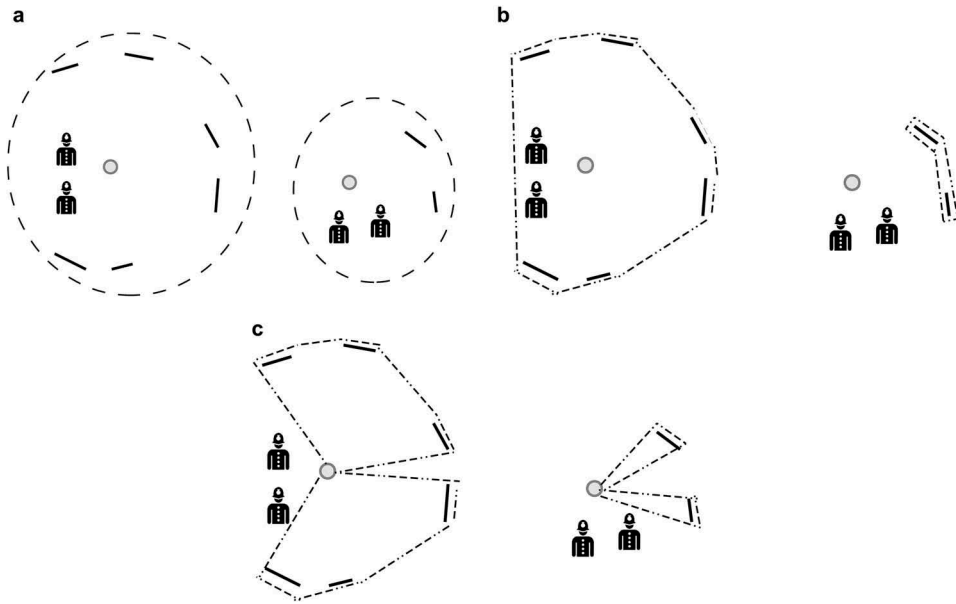


Figure 3. Illustration of Cluster-Route-Split (CRS).

way, the generated routes may have similar lengths and contain edges that are close to each other, as they are from a partition of the 1-tour. However, the route lengths are affected by the depot-edge distance. Meanwhile, CRS deals with different depots by assigning each required edge to the nearest depot before generating routes. It is guaranteed that the generated routes contain edges in close proximity, but not necessarily have similar lengths.

As TSE and CRS follow different paradigms, we cannot easily judge which method is better. Therefore, we will use both of them in the TABU-PATROL, and select the one that generates a better solution.

3.3.2. Stage 2: improving the initial solutions

The initial solutions from TSE and CRS may be excessively long and unbalanced, and will be improved in Stages 2 and 3. At this stage, we apply the IMPROVE-SOLUTION procedure developed by Willemse and Joubert (2012) without adjustment. IMPROVE-SOLUTION iteratively attempts to improve a solution T by exchanging or reallocating edges between different tours such that the longest tour is shortened. It terminates when no improvement is obtained in the current iteration.

3.3.3. Stage 3: SIMPLE-TABU-SEARCH

In this stage, SIMPLE-TABU-SEARCH is called to find new solutions by reallocating the edges in the current solution. SIMPLE-TABU-SEARCH used here is different from the TABU-SEARCH algorithm in TABU-GUARD on two aspects. First, SIMPLE-TABU-SEARCH is simpler and easier to use than TABU-GUARD, as it includes only one tabu list strategy (SIMPLE-TABU) and only one exchange neighbourhood procedure (EN). SIMPLE-TABU and EN are selected, as they perform the best in the tests by Willemse and Joubert

(2012) and our pilot tests. Second, parameters of SIMPLE-TABU-SEARCH are selected by an automatic algorithm, which would lead to a non-biased parameter configuration and solutions of higher quality. The manual parameter selection in TABU-GUARD is biased, as it is usually guided by the personal experience of the users and is influenced by the random factors in the experiments.

SIMPLE-TABU-SEARCH begins with constructing neighbourhood solutions by modifying the initial solution. Then the neighbourhood solutions are examined and the best is chosen for the next iteration. The chosen new solution may be better or worse than the previous solution. However, the search may cycle by continuously moving back to the visited solutions such as the local optimum, which hinders further progress. To tackle this problem, SIMPLE-TABU-SEARCH uses a tabu list to keep track of the recent neighbourhood moves, which are called tabu moves. Each tabu move is stored for a certain number of iterations and would 'expire' and be removed from the list. If a new neighbourhood solution comes from a move that coincides with any tabu move, it cannot be chosen. However, if no non-tabu neighbourhood solutions exist, the algorithm may move to a neighbourhood solution that comes from a tabu move but has better performance than the best solution ever found. The best solution found is called the incumbent solution, and is tracked in the search. SIMPLE-TABU-SEARCH ends when no non-tabu neighbourhood solution is available and no neighbourhood solution that outperforms the incumbent solution exists. Finally, SIMPLE-TABU-SEARCH returns the incumbent solution. Later, we present details on essential components of SIMPLE-TABU-SEARCH.

Constructing neighbourhood solutions. SIMPLE-TABU-SEARCH constructs neighbourhood solutions using EN, which removes an edge from one tour and adding it to another tour. EN is applied to two tours at a time: the longest tour C_i , and any other tour C_j . It removes an edge u from C_i and another edge v from C_j , and then inserts u into C_j and v into C_i by calling INSERT-ARC (Willemse and Joubert 2012) to form a new solution. This process is repeated for any two-edge combination with u of C_i and v of C_j . Then, the overall best solution and the best non-tabu solution obtained from this process is improved by IMPROVE-SOLUTION and returned. SIMPLE-TABU-SEARCH will move to the overall best solution if it outperforms the incumbent solution, or otherwise move to the best non-tabu neighbouring solution.

Tabu list and stopping criteria. The tabu list is updated in each iteration. After moving to a neighbourhood solution T_{new} , SIMPLE-TABU-SEARCH adds to the tabu list the edges u and v that have been moved to create T_{new} . Any move that involves edge u or v leads to a tabu solution. This tabu list strategy is called TABU-SIMPLE (Willemse and Joubert 2012). The edges u and v will be removed from the tabu list after a predefined number of iterations (tabu tenure) have passed.

SIMPLE-TABU-SEARCH has two stopping criteria. First, it terminates if all neighbourhood solutions are tabu and the overall best neighbourhood solution does not outperform the incumbent solution. Second, it terminates if the incumbent solution has not improved after a predefined number of iterations. After SIMPLE-TABU-SEARCH terminates, the incumbent solution is returned as the improved solution.

Parameter configuration. SIMPLE-TABU-SEARCH has two parameters: the maximum number of iterations with improvement t_{max} , and the tabu tenure α . As the initial solutions involve randomness, it is challenging to find the best parameters of SIMPLE-TABU-SEARCH to improve different initial solutions. Therefore, the parameter tuning requires repeated experiments and statistical testing. This task is an instance of the algorithm configuration problem, and can be solved by iterated racing (Balaprakash *et al.* 2007, López-Ibáñez *et al.* 2016).

Iterated racing is a method for automatic configuration, which consists of three steps: sampling new configurations according to a particular distribution; selecting the best configurations from the newly sampled ones via racing; and updating the sampling distribution in order to bias the sample towards the best configurations. These steps are repeated until a termination criterion is met: reaching a minimum number of surviving configurations, a maximum number of used instances or a predetermined computational budget. More details of iterated racing can be found in López-Ibáñez *et al.* (2016).

4. Algorithm evaluation for MMMDRPP

Since MMMDRPP is NP-hard, we propose three lower bounds to approximate the minimum longest route length. Then, a solution for MMMDRPP is evaluated in terms of its lower bound gap (LBG), in comparison with the tightest lower bound. Given TABU-PATROL and similar algorithms involving randomness, its efficiency should be evaluated based on different solutions from it. We will run the algorithm multiple times and use the Best lower bound gap (LBP) and Mean LBP of the solutions, and the execution time to evaluate the efficiency of the algorithm.

Here, we introduce the lower bounds. The first lower bound is called the SPT-LB, which is adapted from STP-LB for MMRPP. For any required edge e , there must be a nearest depot from D and a shortest path length, $SPT(e) = \min_{d \in D} w(SP(d, e), e, SP(e, d))$. In an optimal solution, each required edge has to be traversed, and the route to traverse e is no shorter than $SPT(e)$. Thus, the longest route in an optimal solution must be no shorter than the $\max_{e \in E} SPT(e)$.

The second lower bound, the IPR-LB is obtained by solving the relaxed integer programming problem of MMMDRPP without the constraint (7). This lower bound is the same as IPR-LB for MMRPP, as the different depot nodes are not considered.

The third lower bound is called the Rural Postman Tour Lower Bound (RPT-LB). The idea is to create a rural postman tour using the optimal solution $T^* = \{C_1, \dots, C_k\}$ to the MMMDRPP. Let $w_{max}(T^*)$ be the length of the longest tour of T^* . Then, we can generate a shortest cycle to traverse all depot nodes at least once, which is the well-known Travelling Salesman Problem, and can be near-optimally computed by an approximation algorithm (JGraphT 2017), with the approximation ratio of 2. That is, if the shortest cycle and the calculated cycle are C^* and C' , respectively, we can prove that $w(C^*) \leq w(C') \leq 2 \cdot w(C^*)$. Then, we can combine the tours of T^* and C^* to construct a connected and larger tour T' , with $w(T') = w(T^*) + w(C^*) \leq k \cdot w_{max}(T^*) + w(C')$. As T' traverses all required edges of E , it is a feasible solution to the RPP on $G = (V, E, R)$ with one postman (referred to as 1-RPP). Moreover, this 1-RPP can be near-optimally solved by the algorithm of Christofides *et al.* (1981), with an approximation factor of 3/2. Let $RPPT^*$ be the optimal RPP tour and $RPPT'$ the obtained RPP tour, and it

exists that $w(\text{RPPT}^*) \leq w(\text{RPPT}') \leq 3/2 * w(\text{RPPT}^*)$. As T' is a feasible solution to the 1-RPP, then we have $w(T') \geq w(\text{RPPT}^*) \geq 2/3 * w(\text{RPPT}')$. It is not difficult to prove that $k * w_{\max}(T^*) \geq w(T') - w(C') \geq 2/3 * w(\text{RPPT}') - w(C')$, hence $[2/3 * w(\text{RPPT}') - w(C')]/k$ is a new lower bound for the MMMDRPP.

To evaluate the quality of a solution T , we use the LBG, which is defined as

$$\text{LBG}(T) = \frac{w_{\max}(T) - \text{maxLB}}{w_{\max}(T)} \quad (8)$$

where $w_{\max}(T)$ is the length of the longest route in T , maxLB is the tightest lower bound for the given problem. The range of $\text{LBG}(T)$ is $(0,1)$, and a lower value indicates higher solution quality.

Based on different solutions generated by TABU-PATROL or similar algorithms, we can calculate the Best LBG and Mean LBG. An algorithm with lower Best LBG and lower mean LBG is more efficient than others. Moreover, as time constraint is essential in applications, an efficient algorithm should generate quality solutions within acceptable time.

5. Case study: balanced routes for police patrolling in Camden

In Sections 5 and 6, we use different MMMDRPP cases to test and evaluate the TABU-PATROL with three constructive methods (TSE, CRS and PI). Here, the PI method is adjusted and used as a baseline method, as follows: for each depot $d \in D$, it creates a route $\{d, \text{SPR}(d,r), r, \text{SPR}(r,d), d\}$ by finding the untraversed required edge r that is furthest from d . Then, it adds the remaining untraversed required edges to an existing tour that results in the minimum route length increase.

Here, we illustrate using TABU-PATROL to develop patrol routes from multiple stations in Borough Camden in northwest Inner London. Camden is policed by the Metropolitan Police Service, and during 2011 and 2012 it had five major police stations. As is indicated by MPS, there are different types of officers, including foot patrol officers, community support officers and senior officers. The responsibility and tasks of officers vary with their types, and we select foot patrol officers for this study, as their main task is to patrol and cover the area. The patrol activity in a day is divided into three shifts (i.e. early, late and night shift), each lasting for about eight hours. In each shift, foot patrollers are dispatched to patrol the environment and to cover some important locations.

The purpose of this case is to generate routes for patrolling Camden with a group of patrollers from different stations, in order to balance the route length amongst patrollers. To illustrate the balanced patrol routes, we assume the number of patrollers as five and ten, which means one and two patrollers from each police station, respectively. We also assume that the police patrollers are dispatched to cover the crime hotspots in Camden (see Figure 4). Here, 69 street segments have been identified as crime hotspots that have the highest crime density and covering 1% of the total road length. The time duration of crimes used is from March of 2011 to March of 2012, and the types include theft, burglary, homicide, battery, arson, motor vehicle theft, assault and robbery.

All algorithms described in Sections 3 and 4 were coded in Java version 1.8.0_66 and run in Java SE Runtime Environment and on two 3.60 GHz Intel® Core™ i7 CPUs with 32.0

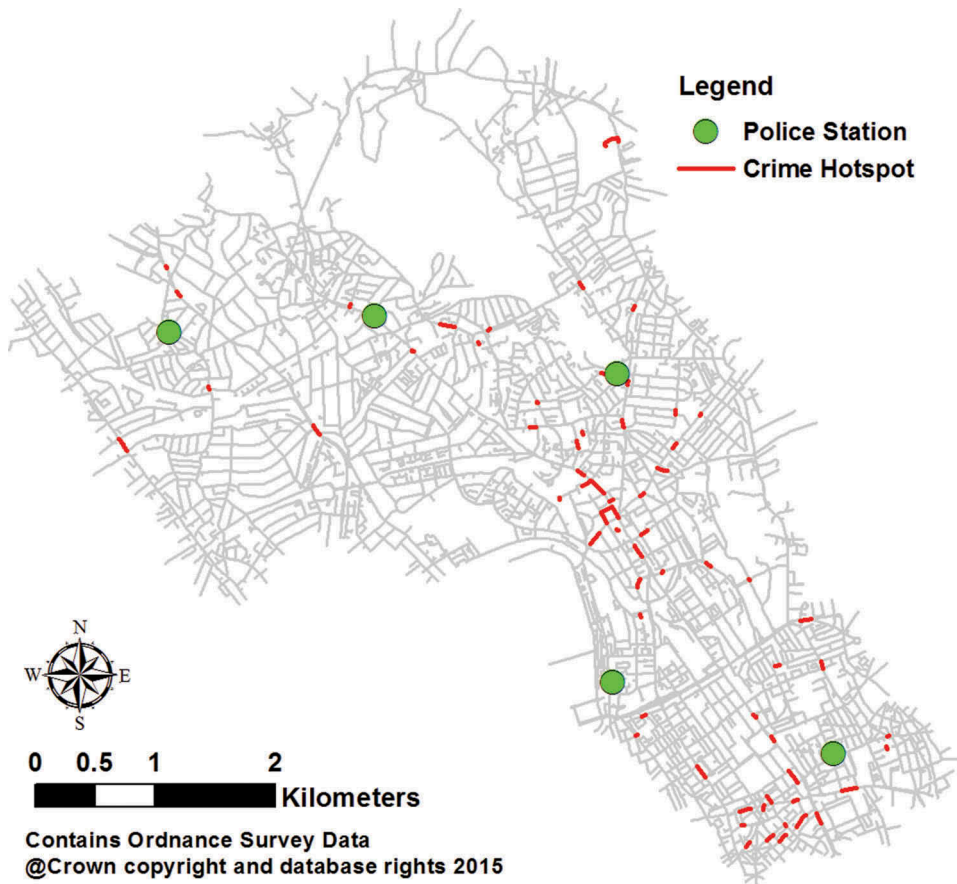


Figure 4. Crime hotspot map of Camden, London.

GB of RAM. The integer programming of IPR-LB was solved by 64-bit Gurobi Optimizer version 6.5.2 (Gurobi Optimization Inc. 2016). Parameter configuration was conducted in R environment version 3.2.2 (R Core Team 2015) and the irace package version 2.1.1662 (López-Ibáñez *et al.* 2016). Data analysis was also performed in R environment.

5.1. Computational results

Here, we first discuss how to use TABU-PATROL to generate patrolling routes, and then evaluate the solutions using the Best LGB, Mean LGB and execution time. Afterwards, we demonstrate the routes from TABU-PATROL on the road map.

TABU-PATROL generated ten different initial solutions for 5 and 10 patrollers by using three constructive methods (TSE, CRS and PI), and then improved the solutions by calling IMPROVED-SOLUTION and SIMPLE-TABU-SEARCH.

The parameter configuration of SIMPLE-TABU-SEARCH was selected by the iterating-racing algorithm from the IRACE package. Both parameters, t_{max} and α are treated as categorical variables, with the range of {100,200,300,400,500,600,700} and {1, 2, ..., 15}, respectively. These ranges are selected based on the TABU-GUARD algorithm and some

pilot experiments. The evaluation measure used by iterated racing is LBG. Up to 300 experiments are performed in the iterated racing. The parameter configuration was conducted on each combination of constructive method and k . Table 1 shows the best configuration for each algorithm and k . The best configurations differ from case to case. Overall, the best performance of SIMPLE-TABU-SEARCH is obtained using a high value of t_{max} and medium value of a .

Before evaluation, we calculate the three MMMDRPP lower bounds for patrolling Camden with the patroller number k as 5 and 10, as is reported in Table 2. All results are given in metres and the tightest bounds are in bold, and the column *Tightest* in Table 2 indicates the tightest lower bound of the three. SPT-LB does not consider k and hence does not change with k , whereas IPR-LB and RPT-LB decrease as k increases. For both k , SPT-LB is the tightest lower bound, probably because of an edge that is very far away from all depots. For clarity, in the remainder of this paper Lower Bound or LB refers to the tightest lower bound for the given k .

Tables 3 and 4 show the results of Best LBG and Mean LBG in Camden using three methods and after three stages. For all methods, large LBGs were observed in the initial solutions. The solution quality was slightly improved after using IMPROVE-SOLUTION, and then considerably improved after SIMPLE-TABU-SEARCH. From Table 2, when $k = 5$, in the initial solutions, the mean LBG was the lowest with TSE, and the highest with PI. In the final solutions, TSE performed the best, in terms of both Best LBG and Mean LBG, while PI produced the worst solutions with the highest Best LBG and Mean LBG. The performance of CRS is slightly better than PI. Large lower bound gaps were still observed

Table 1. Best parameter configuration for each combination of (method, k).

(Method, k)	(TSE, 5)	(TSE, 10)	(CRS, 5)	(CRS, 10)	(PI, 5)	(PI, 10)
a	9	7	12	6	9	5
t	500	700	700	600	600	700

Table 2. Lower bounds for Camden hotspot map (in metres).

k	SPT-LB	IPR-LB	RPT-LB	<i>Tightest</i>
5	4748	1554	2259	4748
10	4748	784	1129	4748

Table 3. Summary of 10 solutions for $k = 5$ in Camden case using three methods.

	TSE		CRS		PI	
	Best LBG	Mean LBG	Best LBG	Mean LBG	Best LBG	Mean LBG
Initial	0.600	0.631	0.643	0.648	0.765	0.781
Improved	0.579	0.611	0.642	0.645	0.764	0.779
Final	0.437	0.453	0.443	0.508	0.446	0.567

Table 4. Summary of 10 solutions for $k = 10$ in Camden using three methods.

	TSE		CRS		PI	
	Best LBG	Mean LBG	Best LBG	Mean LBG	Best LBG	Mean LBG
Initial	0.292	0.395	0.420	0.526	0.710	0.711
Improved	0.231	0.367	0.393	0.513	0.710	0.711
Final	0.128	0.188	0.086	0.121	0.196	0.513

in the final solutions. This may be partly explained by the weak lower bound in this case, which may be much lower than the true value.

When k increases to 10, we observe a different comparison of three algorithms. Overall, both TSE and CRS significantly outperformed PI. The final solutions of CRS were better than TSE, although the initial solutions of CRS were not the best.

Table 5 compares the execution time for three algorithms in the Camden case. Both TSE and CRS are capable of generating 10 different solutions within 200 s, which is very time-efficient. Although PI is quicker than TSE and CRS, PI performs the worst in terms of solution quality.

To visualise the patrolling routes, we show the initial and final solutions by TSE and CRS with 10 patrollers on the road map (Figure 5). Each colour represents the route of a patroller, and the colour of the route overlap is randomly selected from the overlapped routes. When using TSE, the routes in the initial solution constitute a giant tour to traverse all required edges, and the routes have considerable overlaps. In the final solution, the overlap of routes has been significantly reduced, and the route lengths have become much more balanced. Moreover, each new route is compact and is restricted to a small area, which is suitable for patrolling operation. In contrast, in the initial solution by CRS, each route only traverses the required edges that are adjacent to the station, and visually the route lengths are distinct and unbalanced. In the final solution, the new routes become far more balanced and less overlapped, while the compactness of routes is retained. The results reveal the difference of TSE and CRS in generating initial routes, and also acknowledges the effectiveness and efficiency of the SIMPLE-TABU-SEARCH algorithm in balancing the routes, alleviating route overlaps and keeping the routes compact.

5.2. Applying TABU-PATROL in practice

Here, we discuss how to apply TABU-PATROL in practice. In practice, route planners of police can use TABU-PATROL to generate efficient patrol routes. First, route planners decide the available patrollers, and then use TABU-PATROL with TSE and CRS to generate a sufficiently large set of solutions. The number of solutions depends on the time constraint, and as is shown in the case study, generating 10 solutions is a good trade-off between execution time and solution quality. Second, the generated solutions are evaluated by the LBG, and a subset of solutions with the lower LBG is chosen. Third, these chosen solutions can be visualised on the geographical information system (GIS) to view the spatial extent and the traversed area of each route. Route planners can evaluate the feasibility of these routes, based on their knowledge and experience of policing and the local area. Fourth, route planners can now choose one or several feasible solutions, the sequence in which they are implemented, and which routes

Table 5. Summary of execution time for generating 10 solutions on Camden case with three methods (in seconds).

	TSE	CRS	PI
$k = 5$	193	191	131
$k = 10$	84	49	42

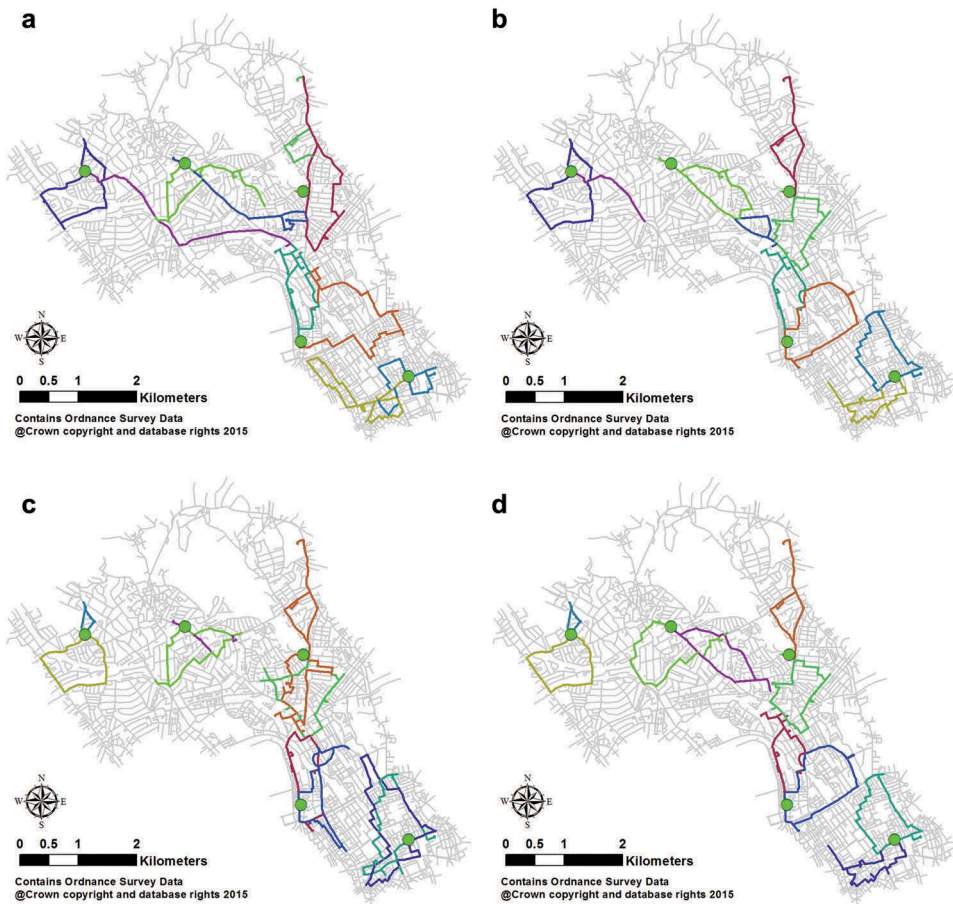


Figure 5. Comparison of the initial and final solutions of TSE and CRS with $k = 10$ in Camden.

should follow a clockwise or anti-clockwise direction. If there are multiple patrolling shifts in the day, route planners can choose a different solution for each shift, which makes the patrolling routes unpredictable and flexible (Willemse and Joubert 2012).

6. Another case study using benchmark problems

For further testing of TABU-PATROL, we tested the algorithm on eight problem instances that are originally proposed for CARP by Li and Eglese (1996). These benchmark instances have been intensively adopted in CARP (Belenguer *et al.* 2006, Brandão and Eglese 2008) and MMCP/MMRPP (Ahr 2004, Ahr and Reinelt 2006, Willemse and Joubert 2012). The original instances entail demands on edges and only one depot, and are adapted for MMDRPP as follows. First, as MMDRPP do not model edge demand, the edge demands are omitted. Moreover, a list of 10 depots is added to each instance to form the new MMDRPP problems, which are called md-egl-**. The depot list of md-egl-e* and md-egl-s* instances is {1, 51, 31, 42, 12, 1, 51, 31, 42, 12}, and {1, 40, 32, 98, 3, 1, 40, 32, 98, 3}, respectively. The new depots are chosen to be evenly distributed in the

network. To our knowledge, this is the first work on the MMMDRPP benchmark problems.

For each instance, TABU-PATROL generated and improved five different solutions for each $k = 2, \dots, 10$. The initial solutions are generated using TSE, CRS and PI, respectively. Each method is coupled with the best SIMPLE-TABU-SEARCH parameter configuration for the Camden case and $k = 10$ (see Table 2). Table 6 reports the average of the mean lower bound gap and execution time over different k . For the complete results, please contact the authors. For the benchmark problems, either TSE or CRS produced the best-quality solutions, except for the md-egl-e1 instance. The worst LBG for TSE and CRS was 0.150 and 0.161, respectively, whereas the worst LBG for PI was 0.440. Overall, these methods have similar execution time on md-egl-e* problems, whereas on md-egl-s* problems, TSE is slower than CRS and PI. As CRS produces high-quality solutions within reasonable execution time, this method is ideal when good solutions have to be generated under time constraints. If there is no time constraint, it is recommended to test TSE and CRS to solve the MMMDRPP and choose the better results.

7. Conclusions and future work

This research presented MMMDRPP, which seeks balanced routes from different depots to cover a road network. This problem is challenging because of the constraint of multiple depots and the objective to minimise the length of the longest route. To solve this problem, we proposed a novel TABU-PATROL algorithm, which follows three stages to generate and improve the routes using tabu search. To approximate the optimal solution value, we devised three lower bounds for this problem, namely SPT-LB, IPR-LB and RPT-LB. Then, we measure the solution quality by using the LBP, and measure the algorithm efficiency by using the best LBP and the mean LBP of its multiple solutions.

In the case study, we demonstrated how to model the route design of police patrolling as the MMMDRPP, and used TABU-PATROL to generate balanced routes. The results showed that TABU-PATROL can generate quality solutions within reasonable time. In TABU-PATROL, we used three methods (TSE, CRS and PI) to generate the initial routes. With 5 patrollers, TSE generates better solutions (with shorter longest routes) than CRS and PI, while with 10 patrollers CRS outperforms TSE and PI. We conclude that TSE and CRS can generate better solutions than PI. In terms of lower bounds, for both 5

Table 6. Results of TABU-PATROL with three constructive methods on MMMDRPP of eight adapted benchmark problems.

	V	E	R	TSE		CRS		PI	
				Mean LBG	Time(s)	Mean LBG	Time(s)	Mean LBG	Time(s)
md-egl-e1	77	98	47	0.145	13.3	0.161	14.6	0.140	16.6
md-egl-e2	77	98	72	0.140	39.4	0.111	37.9	0.193	39.4
md-egl-e3	77	98	87	0.133	78.1	0.118	72.1	0.215	73.1
md-egl-e4	77	98	98	0.123	159	0.134	144	0.440	148
md-egl-s1	140	190	115	0.150	56.1	0.143	44.3	0.247	48.1
md-egl-s2	140	190	147	0.122	414	0.130	326	0.254	317
md-egl-s3	140	190	159	0.118	520	0.123	416	0.261	402
md-egl-s4	140	190	190	0.062	847	0.074	683	0.202	667

and 10 patrollers, the SPT-LB is the tightest of the three. We also discussed how to apply TABU-PATROL in practice.

Furthermore, the TABU-PATROL algorithm was tested on several adapted benchmark problems, and results showed that TABU-PATROL was robust enough to generate quality solutions on different instances.

The TABU-PATROL algorithm proposed here is applicable to applications that involve multiple vehicles from different depots in order to service a set of edges in a road network with balanced routes. Whilst we opted to use the case of patrolling crime hotspots to demonstrate its utility, the algorithm can be used to generate routes in fields such as delivery from multiple warehouses, snow removal work (Li and Eglese 1996) and routing of estate security guards (Willemse and Joubert 2012).

In police patrolling, there exist opportunities to improve patrolling of an area even further. Most notable are increasing the visibility of patrolling and dynamic routing. Patrolling may have different and multiple goals. Besides covering the required locations, patrollers are required to increase their visibility to the public in order to increase public confidence in policing. The task of generating patrol routes to satisfy multiple goals can be modelled as a multiple-objective optimisation problem. This could be a direction for further research.

The patrolling process is complex and dynamic. Patrollers would have to stop patrolling and respond to calls for service if necessary. In this situation, the patrolling tasks should be reallocated among patrollers and the planned routes should be adjusted within a short time. Therefore, this new problem needs to be formulated and solved, which is also a direction for future research.

Acknowledgement

This work is part of the Crime, Policing and Citizenship (CPC): Space-Time Interactions of Dynamic Networks Project (www.ucl.ac.uk/cpc), supported by the UK Engineering and Physical Sciences Research Council (EP/J004197/1). This work is supported in part by the PhD scholarship from the China Scholarship Council (CSC) under the Grant CSC No. 201406010296 and the BEAMS Dean's Prize from University College London.

The authors would like to acknowledge the Metropolitan Police Service (MPS) for provision of the crime data in the London Borough of Camden. They are also grateful to Trevor Adams for many valuable discussions about the manuscript and related work. The results presented and views expressed in this manuscript are the responsibility of the authors alone and do not represent the views of Trevor Adams or the MPS.

The authors gratefully acknowledge team members of the CPC projects, in particular, Gabriel Rosser, for their contribution and feedback on this work. The authors also acknowledge Mr. Rui Sha (UCL-CEGE) for his help on this work.

The authors would like to thank the editors Dr. May Yuan and Prof. Bo Huang, and two anonymous referees for their valuable comments and suggestions.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the UK Engineering and Physical Sciences Research Council [EP/J004197/1]. This work is supported in part by the PhD scholarship from the China Scholarship Council (CSC) under the Grant CSC [No. 201406010296] and the BEAMS Dean's Prize from University College London.

ORCID

Tao Cheng  <http://orcid.org/0000-0002-5503-9813>

John Shawe-Taylor  <http://orcid.org/0000-0002-2030-0073>

References

- Ahr, D., 2004. *Contributions to multiple postmen problems*. Heidelberg: Ruprecht-Karls-Universität.
- Ahr, D. and Reinelt, G., 2002. New heuristics and lower bounds for the min-max k-Chinese postman problem. In: R. Mohring and R. Raman, eds. *Algorithms — ESA 2002*. Heidelberg: Springer Berlin, 64–74.
- Ahr, D. and Reinelt, G., 2006. A tabu search algorithm for the min-max k-Chinese postman problem. *Computers and Operations Research*, 33 (12), 3403–3422. doi:10.1016/j.cor.2005.02.011
- Akyurt, I.Z., Keskinturk, T., and Kalkanci, C., 2015. Using genetic algorithm for winter maintenance operations: multi depot K-Chinese postman problem. *Emerging Markets Journal*, 5 (1), 50. doi:10.5195/EMAJ.2015.69
- Amberg, A., Domschke, W., and Voß, S., 2000. Multiple center capacitated arc routing problems: a tabu search algorithm using capacitated trees. *European Journal of Operational Research*, 124 (2), 360–376. doi:10.1016/S0377-2217(99)00170-8
- Arkin, E.M., Hassin, R., and Levin, A., 2006. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59 (1), 1–18. doi:10.1016/j.jalgor.2005.01.007
- Balaprakash, P., Birattari, M., and Stuetzle, T., 2007. Improvement strategies for the f-race algorithm: sampling design and iterative refinement. *Hybrid Metaheuristics, Proceedings*, 108–122.
- Belenguer, J.-M., et al., 2006. Lower and upper bounds for the mixed capacitated arc routing problem. *Computers & Operations Research*, 33 (12), 3363–3383. doi:10.1016/j.cor.2005.02.009
- Benavent, E., et al., 2009. Min-Max K-vehicles windy rural postman problem. *Networks*, 54 (4), 216–226. doi:10.1002/net.v54:4
- Bertsimas, D. and Tsitsiklis, J.N., 1997. *Introduction to linear optimization*. Belmont, MA: Athena Scientific.
- Brandão, J. and Eglese, R., 2008. A deterministic tabu search algorithm for the capacitated arc routing problem. *Computers & Operations Research*, 35 (4), 1112–1126. doi:10.1016/j.cor.2006.07.007
- Carlsson, J., Ge, D., and Subramaniam, A., 2009. Solving min-max multi-depot vehicle routing problem. *Lectures on Global Optimization*, 55, 31–46.
- Christofides, N. et al., 1981. An algorithm for the rural postman problem. *Report IC.OR.81.5*, 81. Available at: https://www.researchgate.net/publication/239385320_An_algorithm_for_the_Rural_Postman_Problem [Accessed 22 April 2015].
- Corberan, A. and Laporte, G., 2014. *Arc routing: problems, methods, and applications*. Philadelphia, PA: SIAM.
- Degenhardt, J., 2004. An ant-algorithm for the balanced k-Chinese postmen problem. In: *Operations research conference*. Tilburg University.
- Dijkstra, E.W., 1959. A note on two problems in connection with graphs. *Numerische Mathematik*, 1, 269–271. doi:10.1007/BF01386390
- Dror, M., ed., 2000. *Arc routing: theory, solutions and applications*. 1st ed. New York: Springer Science & Business Media.
- Edmonds, J. and Johnson, E.L., 1973. Matching, Euler tours and the Chinese postman. *Mathematical Programming*, 5 (1), 88–124. doi:10.1007/BF01580113

- Eiselt, H.A., Gendreau, M., and Laporte, G., 1995. Arc routing problems, part I: the Chinese postman problem. *Operations Research*, 43 (2), 231–242. doi:10.1287/opre.43.2.231
- Euler, L., 1736. Solutio problematis ad geometriam situs pertinentis. *Commentarii Academiae Scientiarum Petropolitanae*, 8, 128–140.
- Frederickson, G.N., Hecht, M.S., and Kim, C.E., 1976. Approximation algorithms for some routing problems. In: *17th annual symposium on foundations of computer science (sfcs 1976)*. IEEE. 216–227.
- Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13 (5), 533–549. doi:10.1016/0305-0548(86)90048-1
- Glover, F. and Laguna, M., 1997. *Tabu search*. Boston: Kluwer academic publishers.
- Greistorfer, P., 2003. A tabu scatter search metaheuristic for the arc routing problem. *Computers and Industrial Engineering*, 44 (2), 249–266. doi:10.1016/S0360-8352(02)00178-X
- Gurobi Optimization Inc. 2016. *Gurobi optimizer reference manual*. Available at: <http://www.gurobi.com> [Accessed 1 September 2017].
- Hertz, A., Laporte, G., and Mittaz, M., 2000. A tabu search heuristic for the capacitated arc routing problem. *Operations Research*, 48 (1), 129–135. doi:10.1287/opre.48.1.129.12455
- JGraphT, 2017. JGraphT Library. Available from: <http://jgrapht.org/> [Accessed 1 September 2017].
- Lenstra, J.K. and Kan, A.H.G.R., 1976. On general routing problems. *Networks*, 6 (3), 273–280. doi:10.1002/(ISSN)1097-0037
- Li, L.Y.O. and Eglese, R.W., 1996. An interactive algorithm for vehicle routeing for winter gritting. *Journal of the Operational Research Society*, 47 (2), 217–228.
- López-Ibáñez, M., et al., 2016. The irace package: iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3, 43–58. doi:10.1016/j.orp.2016.09.002
- Osterhues, A. and Mariak, F., 2005. On variants of the k-Chinese postman problem. *Operations Research Und Wirtschaftsinformatik* (30). Available at: <http://www.wiso.tu-dortmund.de/wiso/or/Medienpool/publikationen/dispap30.pdf> [Accessed 1 September 2017].
- Prins, C., Lacomme, P., and Prodhon, C., 2014. Order-first split-second methods for vehicle routing problems: a review. *Transportation Research Part C: Emerging Technologies*, 40, 179–200. doi:10.1016/j.trc.2014.01.011
- R Core Team, 2015. R: a language and environment for statistical computing. Available from: <https://www.r-project.org/> [Accessed 1 September 2017].
- Ulusoy, G., 1985. The fleet size and mix problem for capacitated arc routing. *European Journal of Operational Research*, 22 (3), 329–337. doi:10.1016/0377-2217(85)90252-8
- Willemse, E.J. and Joubert, J.W., 2012. Applying min-max k postmen problems to the routing of security guards. *Journal of the Operational Research Society*, 63 (2), 245–260. doi:10.1057/jors.2011.26