



Approximation algorithms for the restricted k -Chinese postman problems with penalties

Pengxiang Pan¹ · Hongtao Zhu²

Received: 4 August 2022 / Accepted: 28 February 2023 / Published online: 13 March 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Given an undirected graph $G = (V, E; w, p)$ with a depot $r \in V$ and an integer $k \in \mathbb{Z}^+$, each edge $e \in E$ has a weight $w(e) \in \mathbb{R}^+$ and a penalty $p(e) \in \mathbb{R}_0^+$, where the weights satisfy the triangle inequality, we consider two types of restricted k -Chinese postman problems with penalties. (1) The restricted min–max k -Chinese postman problem with penalties (MM-RPCPP) is asked to find a set of k tours starting from r and collectively covering all vertices, such that the maximum tour weight plus the total penalty paid for the uncovered edges is minimized. (2) The restricted min–sum k -Chinese postman problem with penalties (MS-RPCPP) is asked to find a set of k tours satisfying the constraint mentioned above and that each edge appears in at most one tour and each tour contains at least one edge, such that the sum of weights of all tours plus the total penalty paid for the uncovered edges is minimized. In the paper, we design a combinatorial $(\frac{7}{2} - \frac{2}{k})$ -approximation algorithm to solve the MM-RPCPP. Furthermore, we present a combinatorial 2-approximation algorithm to solve the MS-RPCPP.

Keywords Approximation algorithm · Penalties · Restricted · Chinese postman · Traveling salesman

1 Introduction

The routing problems are a much-studied family of combinatorial optimization problems, and have a wide range of applications that include network surveillance, household refuse collection, snow removal, street sweeping, school bus routing.

✉ Hongtao Zhu
htZhu86@outlook.com
Pengxiang Pan
panpx@outlook.com

¹ School of Mathematics and Statistics, Yunnan University, Kunming, China

² School of Mathematical Sciences, Universiti Sains Malaysia, George Town, Pulau Pinang, Malaysia

Considering an application on road network rescue, Zhu and Pan [16] proposed the RESTRICTED CHINESE POSTMAN PROBLEM WITH TOTAL PENALTY (RPCPP), which is defined on an undirected graph $G = (V, E; w, p)$ with a weight function $w : E \rightarrow \mathbb{R}^+$ satisfying the triangle inequality, and a penalty function $p : E \rightarrow \mathbb{R}_0^+$. This problem is to find a tour C traversing each vertex $v \in V$ at least once, such that the value $w(C) + p(E \setminus C)$ is minimized, where $w(C) = \sum_{e \in C} w(e)$, and $p(E \setminus C) = \sum_{e \in E \setminus C} p(e)$. In 2021, a 1.5-approximation algorithm was designed by Zhu and Pan [16] to solve the RPCPP. Other related routing problems and their applications can be found in [1–3, 5, 10].

In the case that a given weight function satisfies the triangle inequality, there are two types of special cases of the RPCPP, which are described as follows: The CHINESE POSTMAN problem (CPP) [11] is to find a tour on a given undirected graph traversing each edge at least once, such that the weight of this tour is minimized. Clearly, the CPP with the triangle inequality holds is a special case of the RPCPP, where $p(e) \geq 2w(e)$ for each edge $e \in E$. Edmonds and Johnson [6] designed a polynomial time exact algorithm to solve the CPP.

In comparison, the METRIC TRAVELING SALESMAN problem (metric TSP) [15, p.30] is to find a cycle on a given complete graph traversing each vertex exactly once, such that the weight of this cycle is minimized. The metric TSP is another special case of the RPCPP, where G is a complete graph and $p(\cdot) \equiv 0$. Christofides [15] presented a 1.5-approximation algorithm to solve the metric TSP. In 2021, Karlin et al. [12] reconsidered the metric TSP and designed a $(1.5 - \epsilon)$ -approximation algorithm for some $\epsilon > 10^{-36}$.

In real life, it is common for multiple vehicles to work in parallel. Frederickson et al. [7] addressed the CPP and the metric TSP with multiple vehicles and min–max objective, namely, the MIN–MAX k -CHINESE POSTMAN problem (MM-CPP) and the MIN–MAX k -TRAVELING SALESMAN problem (MM-TSP).

The MM-CPP and the MM-TSP are defined as follows: Given an undirected graph $G = (V, E; w)$ with a depot $r \in V$, an integer $k \in \mathbb{Z}^+$, and a weight function $w : E \rightarrow \mathbb{R}^+$ that satisfies the triangle inequality, the MM-CPP is to find k tours starting at r , and collectively traversing each edge $e \in E$ at least once, such that the maximum of the weights of tours is minimized. Employing a splitting method, Frederickson et al. [7] designed a $(2 - \frac{1}{k})$ -approximation algorithm to solve the MM-CPP. When replacing a given undirected graph with a complete graph, the MM-TSP is to find k cycles starting at r , and collectively covering all vertices in V , such that the maximum of the weights of cycles is minimized. Reemploying a splitting method, Frederickson et al. [7] presented a $(\frac{5}{2} - \frac{1}{k})$ -approximation algorithm to solve the MM-TSP.

Moreover, the CPP and the metric TSP with multiple vehicles and min-sum objective were studied by Frieze [14] and Pearn [8]. There, the two problems are referred to as the MIN-SUM k -CHINESE POSTMAN problem (MS-CPP) and the MIN-SUM k -TRAVELING SALESMAN problem (MS-TSP), which are defined as follows: In a given undirected graph $G = (V, E; w)$ with a depot $r \in V$ and an integer $k \in \mathbb{Z}^+$, by adding two constraints that each edge in G must be serviced by exactly one postman and all the k postmen must be involved in the delivery service, the MS-CPP is to find k tours starting at r , and collectively traversing each edge $e \in E$ at least

once, such that the sum of weights of tours is minimized. Using a lower bound algorithm in [4] for the capacitated arc routing problem, Pearn [14] showed that the MS-CPP can be solved optimally. By replacing a given undirected graph with a complete graph, the MS-TSP is to find k cycles starting at r and collectively covering all vertices in V , and satisfying that each cycle contains at least 3 edges, such that the sum of weights of cycles is minimized. Modifying Christofides' algorithm [15, p.32] for solving the metric TSP, Frieze [8] designed a 1.5-approximation algorithm to solve the MS-TSP.

Motivated by the extensive application in practices of the problems mentioned above, we consider the RESTRICTED MIN-MAX k -CHINESE POSTMAN PROBLEM WITH PENALTIES (MM-RPCPP) and its a variant, which are defined as follows: Given an undirected graph $G = (V, E; w, p)$ with a depot $r \in V$, an integer $k \in \mathbb{Z}^+$, a weight function $w : E \rightarrow \mathbb{R}^+$ that satisfies the triangle inequality, and a penalty function $p : E \rightarrow \mathbb{R}_0^+$, (1) the MM-RPCPP is asked to find a set $\mathcal{C} = \{C_1, \dots, C_k\}$ of k tours starting from r and collectively covering all vertices, such that the value $f(\mathcal{C}) = w_{\max}(\mathcal{C}) + p(E \setminus \mathcal{C})$ is minimized, where $w_{\max}(\mathcal{C}) = \max\{\sum_{e \in C} w(e) \mid C \in \mathcal{C}\}$ and $p(E \setminus \mathcal{C}) = \sum_{e \in E \setminus \mathcal{C}} p(e)$, and (2) The RESTRICTED MIN-SUM k -CHINESE POSTMAN PROBLEM WITH PENALTIES (MS-RPCPP) is asked to find a set $\mathcal{C} = \{C_1, \dots, C_k\}$ of k tours satisfying the constraint mentioned above and that each edge appears in at most one tour and each tour contains at least one edge, such that the value $h(\mathcal{C}) = w(\mathcal{C}) + p(E \setminus \mathcal{C})$ is minimized, where $w(\mathcal{C}) = \sum_{C \in \mathcal{C}} \sum_{e \in C} w(e)$, and $p(E \setminus \mathcal{C})$ is defined in (1). Without loss of generality, we assume $k \leq |E|$ in the paper.

The MM-RPCPP and the MS-RPCPP are multiple-vehicle extensions of the RPCPP respectively. For the MM-RPCPP, the MM-CPP is the special case $p(e) > 2 \cdot w(E)$ for each edge $e \in E$, and the MM-TSP is another special case that G is a complete graph and $p(\cdot) \equiv 0$. For the MS-RPCPP, the CPP with the triangle inequality holds is the special case that $k = 1$ and $p(e) > 2 \cdot w(E)$ for each edge $e \in E$, and the metric TSP is another special case that $k = 1$, G is a complete graph and $p(\cdot) \equiv 0$. In addition, the MS-RPCPP is closely related to the MS-CPP and the MS-TSP, while it is not a generalization of them.

The MM-RPCPP and the MS-RPCPP may have applications in the real world. For example, an extension of the application in [16] is described as follows: Given a road network, k vehicles and some rescue workers, we need to arrange workers to transfer people on some roads to the road ends in advance, and then plan vehicle routings to rescue people who lie on the residual network. It is of interest to know how to plan vehicle routings to minimize the time spent by workers and vehicles, where the time spent on each road by workers corresponds to a penalty and the time spent on each road by any vehicle corresponds to a weight. In order to minimize the rescue time, we can arrange vehicles in parallel to rescue people, this question can be reduced to the MM-RPCPP. In addition, we further consider another variant of the RPCPP, that is the MS-RPCPP.

To the best of our knowledge, these two problems that we propose have not been considered in the literature. The contribution of this paper is to present two approximation algorithms with constant factors to solve the MM-RPCPP and the MS-RPCPP, respectively.

This paper is organized as follows: In Sect. 2, we present some preliminaries to facilitate the descriptions of algorithms. In Sect. 3, we design a $(\frac{7}{2} - \frac{2}{k})$ -approximation algorithm to solve the MM-RPCPP. In Sect. 4, we present a 2-approximation algorithm to solve the MS-RPCPP. In Sect. 5, we provide our conclusion and further work.

2 Preliminaries

Given a graph $G = (V, E)$, a walk P connecting a vertex v_{i_1} and a vertex $v_{i_{k+1}}$ is an alternating sequence $P = v_{i_1} e_{i_1} v_{i_2} e_{i_2} v_{i_3} \cdots v_{i_k} e_{i_k} v_{i_{k+1}}$ such that, $k \geq 1$ and $e_{i_j} = v_{i_j} v_{i_{j+1}} \in E$ for each integer $j \in \{1, \dots, k\}$. If $v_{i_1} = v_{i_{k+1}}$, the walk P is called a *tour*. In addition, a walk P is said to be a *path* if the vertices in P are all distinct. Similarly, a tour P is called a *cycle* if the vertices in P are all distinct.

On a weighted graph $G = (V, E; w)$, the *distance* $d_G(u, v)$ of two vertices u and v is the weight of a shortest path connecting u and v . The *eccentricity* $\delta_G(v)$ of a vertex v is $\max_{u \in V} d_G(u, v)$.

We say that an undirected graph $G = (V, E)$ is *connected* if there is a path connecting u and v for any $u, v \in V$; Otherwise, G is disconnected. An undirected graph without a cycle (as a subgraph) is called a *forest*. We call a connected forest a *tree*.

A *subgraph* of a graph $G = (V, E)$ is a graph $G' = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$. If $E' = \{uv \in E \mid u, v \in V'\}$, then $G' = G[V']$ is the subgraph of G induced by V' . Similarly, if $V' = \{u, v \in V \mid uv \in E'\}$, then $G' = G[E']$ is the subgraph of G induced by E' . We call a subgraph G' of G *spanning* if $V' = V$. A subgraph G' is called a *spanning tree* if G' is a tree and spanning.

Given a graph $G = (V, E)$, an *Euler tour* in G is a tour that traverses each edge $e \in E$ exactly once. In addition, a graph G is *Eulerian* if this graph G admits an Euler tour.

For any weighted graph $G = (V, E; w)$, we say that the weight function $w(\cdot)$ satisfies the *triangle inequality* when $w(xz) \leq d_G(x, y) + d_G(y, z)$ for any $xz \in E$, $x, y, z \in V$. Let $n = |V|$ and $m = |E|$ denote the numbers of vertices and edges in a given graph $G = (V, E)$, respectively. For any tour C of G , we denote E_C the subset of edges in E that is traversed by C .

In our algorithms to solve the MM-RPCPP and the MS-RPCPP, we need the following lemmas.

Lemma 1 [16] *Given an undirected connected graph $G = (V, E; w, p)$ as an instance of the RPCPP, there exists a $\frac{3}{2}$ -approximation algorithm (referred as RPCPP algorithm) for solving the RPCPP in time $O(n^3)$.*

Lemma 2 [9] *Given an undirected connected graph $G = (V, E; w)$ with a depot $r \in V$, a positive integer k and a subset $F \subseteq E$, there exists an algorithm (referred as MST-DC algorithm) which can determine a minimum weight subset $F' \subseteq E$ such that $G[F \cup F']$ is connected and has at least k edges incident to the depot r , or output that there is no such solution, in running time $O(n^2)$.*

3 The restricted min-max k -Chinese postman problem with penalties

In this section, we consider the RESTRICTED MIN-MAX k -CHINESE POSTMAN PROBLEM WITH PENALTIES (MM-RPCPP). Note that the MM-RPCPP is NP-hard since the MM-CPP and the MM-TSP are known to be NP-hard [7]. When $k = 1$, the MM-RPCPP becomes the RPCPP, we can use the RPCPP algorithm to solve the MM-RPCPP to obtain a $\frac{3}{2}$ -approximation solution. Thus, we assume $k \geq 2$ in the section.

The strategy to solve the MM-RPCPP is described as follows:

- (1) Determine a tour in G , satisfying that some objective value as small as possible.
- (2) Partition the above tour into k walks, and then extend walks to k tours satisfying that the maximum tour weight as small as possible.

Using the strategy mentioned above, we design an approximation algorithm (Algorithm 1) in the following.

Algorithm 1 : \mathcal{A}_{MM}

Input: An undirected graph $G = (V, E; w, p)$ with a depot r , a weight function $w : E \rightarrow \mathbb{R}^+$ satisfying the triangle inequality, a penalty function $p : E \rightarrow \mathbb{R}_0^+$ and an integer $k \geq 2$;

Output: A set $\mathcal{C} = \{C_1, \dots, C_k\}$ of tours starting from r , and the value $f(\mathcal{C}) = w_{\max}(\mathcal{C}) + p(E \setminus \mathcal{C})$.

Begin

Step 1: Construct an auxiliary graph $G' = (V, E; w', p)$, where $w'(e) = \frac{w(e)}{k}$ for each edge $e \in E$; Using the RPCPP algorithm, find a feasible solution $C = (r, v_{i_1}, v_{i_2}, \dots, v_{i_t}, r)$ of the RPCPP problem in G' , satisfying that the value $w'(C) + p(E \setminus C)$ as small as possible; If there is no feasible solution for the instance $(G'; w', p)$ of the RPCPP problem, then output "no feasible solution on G ", and stop;

Step 2: According to the weight function $w(\cdot)$, compute the eccentricity of depot r on the graph G , denoted by $\delta_G(r)$;

Step 3: For each $j = 1, 2, \dots, k-1$, let $L_j = \frac{j}{k}(w(C) - 4\delta_G(r)) + 2\delta_G(r)$, and find the last vertex $v_{i_{p'(j)}}$ satisfying $w(C[v_{i_{p'(j)}}]) \leq L_j$, where $w(C[v_{i_{p'(j)}}]) = w(r, v_{i_1}, v_{i_2}, \dots, v_{i_{p'(j)}})$, and then let $s_j = L_j - w(C[v_{i_{p'(j)}}])$;

Step 4: For each $j = 1, 2, \dots, k-1$, if

$$s_j + d_G(r, v_{i_{p'(j)}}) \leq w(v_{i_{p'(j)}} v_{i_{p'(j)+1}}) - s_j + d_G(v_{i_{p'(j)+1}}, r),$$

then let $v_{i_{p(j)}} = v_{i_{p'(j)}}$, otherwise $v_{i_{p(j)}} = v_{i_{p'(j)+1}}$;

Step 5: Let $C_1 = (r, v_{i_1}, v_{i_2}, \dots, v_{i_{p(1)}})$, $C_2 = (v_{i_{p(1)}}, \dots, v_{i_{p(2)}})$, \dots , $C_k = (v_{i_{p(k-1)}}, \dots, r)$;

For each $j \in \{1, 2, \dots, k\}$, transform C_j into a tour by connecting r to the two endpoints of C_j with shortest paths on $G = (V, E; w)$;

Step 6: Output a set $\mathcal{C} = \{C_1, \dots, C_k\}$ of k tours starting from r , and the value $f(\mathcal{C}) = w_{\max}(\mathcal{C}) + p(E \setminus \mathcal{C})$.

End

On an auxiliary graph $G' = (V, E; w', p)$ of a given graph $G = (V, E; w, p)$, it is clear that the function w' satisfies the triangle inequality, because of the weight function w satisfying the triangle inequality. Using the RPCPP algorithm and Lemma 1, we can find a feasible solution for an instance $(G'; w', p)$ of the RPCPP. The specific result is described as follows:

Lemma 3 Given an auxiliary graph $G' = (V, E; w', p)$, the RPCPP algorithm is a $\frac{3}{2}$ -approximation algorithm in time $O(n^3)$ to solve the RPCPP problem.

According to the algorithm \mathcal{A}_{MM} , we first get the following lemma.

Lemma 4 $w_{\max}(C) \leq \frac{w(C)}{k} + 2(2 - \frac{2}{k})\delta_G(r)$.

Proof We first consider any tour C_j , $j \in \{2, \dots, k - 1\}$. Because the weight function $w : E \rightarrow \mathbb{R}^+$ satisfies the triangle inequality, we obtain $w(v_{i_{p'(j)}} v_{i_{p'(j+1)}}) \leq d_G(r, v_{i_{p'(j)}}) + d_G(r, v_{i_{p'(j+1)}})$. By the definition of $\delta_G(r)$ in Step 2 of the algorithm \mathcal{A}_{MM} , we can get $d_G(r, v_{i_{p'(j)}}) + d_G(r, v_{i_{p'(j+1)}}) \leq 2\delta_G(r)$, implying that

$$d_G(r, v_{i_{p'(j)}}) + w(v_{i_{p'(j)}} v_{i_{p'(j+1)}}) + d_G(r, v_{i_{p'(j+1)}}) \leq 4\delta_G(r).$$

Thus, we obtain

$$\min\{s_j + d_G(r, v_{i_{p'(j)}}), w(v_{i_{p'(j)}} v_{i_{p'(j+1)}}) - s_j + d_G(v_{i_{p'(j+1)}}, r)\} \leq 2\delta_G(r).$$

Similarly, the above inequality still holds by replacing j with $j - 1$.

It is easy to see that, the case resulting in the maximum weight of any tour C_j is that the tour starts depot r and follows the shortest path to reach $v_{i_{p'(j-1)}}$, and then follows tour C to $v_{i_{p'(j+1)}}$, and finally follows the shortest path back to r . And in this situation, we have $s_{j-1} + d_G(r, v_{i_{p'(j-1)}}) \leq 2\delta_G(r)$, and $w(v_{i_{p'(j)}} v_{i_{p'(j+1)}}) - s_j + d_G(v_{i_{p'(j+1)}}, r) \leq 2\delta_G(r)$. Hence, we obtain

$$w(C_j) \leq 2\delta_G(r) + \frac{1}{k}(w(C) - 4\delta_G(r)) + 2\delta_G(r). \tag{1}$$

For other cases of tour C_j , i.e., $j \in \{1, k\}$, the above inequality (1) is easily proved. Therefore, for any $j \in \{1, \dots, k\}$, we have $w(C_j) \leq 2\delta_G(r) + \frac{1}{k}(w(C) - 4\delta_G(r)) + 2\delta_G(r)$, implying $w_{\max}(C) \leq \frac{w(C)}{k} + 2(2 - \frac{2}{k})\delta_G(r)$. \square

According to Algorithm 1, Lemma's 3 and 4, we obtain a result for the MM-RPCPP as follows:

Theorem 1 The algorithm \mathcal{A}_{MM} is a $(\frac{7}{2} - \frac{2}{k})$ -approximation algorithm to solve the MM-RPCPP and it runs in time $O(n^3)$.

Proof It is clear that the algorithm can correctly indicate whether there is a feasible solution to the MM-RPCPP. We need only consider the case that there are feasible solutions for the MM-RPCPP in the following.

For any instance $G = (V, E; w, p)$ of the MM-RPCPP and its auxiliary graph $G' = (V, E; w', p)$, let C^* denote an optimal solution for the instance $G = (V, E; w, p)$ of the MM-RPCPP, E_{C^*} denote the subset of edges collectively traversed by all tours

in C^* . And let C^* and $f'(C^*) = w'(C^*) + p(E \setminus C^*)$ denote an optimal solution and optimal value for the instance $(G'; w', p)$ of the RPCPP, respectively.

By the algorithm \mathcal{A}_{MM} , it is clear that $E_C \subseteq E_{C^*}$, so we obtain

$$p(E \setminus C) \leq p(E \setminus C^*).$$

Using Lemma 4, we have

$$w_{\max}(C) \leq \frac{w(C)}{k} + 2\left(2 - \frac{2}{k}\right)\delta_G(r).$$

From Lemma 3, we know that the RPCPP algorithm can generate a feasible solution C for the instance $(G'; w', p)$ of the RPCPP, satisfying $w'(C) + p(E \setminus C) \leq \frac{3}{2}(w'(C^*) + p(E \setminus C^*))$.

Because all tours in C^* have a common depot r , it is easy to merge them into a tour, i.e., a feasible solution for the instance $(G'; w', p)$ of the RPCPP. Hence, we obtain

$$\begin{aligned} w'(C^*) + p(E \setminus C^*) &\leq w'(C^*) + p(E \setminus C^*) \\ &\leq k \cdot w'_{\max}(C^*) + p(E \setminus C^*) \\ &= k \cdot \frac{w_{\max}(C^*)}{k} + p(E \setminus C^*) \\ &= w_{\max}(C^*) + p(E \setminus C^*). \end{aligned}$$

Therefore, we obtain

$$\begin{aligned} w'(C) + p(E \setminus C) &\leq \frac{3}{2}(w'(C^*) + p(E \setminus C^*)) \\ &\leq \frac{3}{2}w_{\max}(C^*) + \frac{3}{2}p(E \setminus C^*). \end{aligned}$$

Due to all tours in C^* collectively traversing each vertex $v \in V$ at least once, we have $\delta_G(r) \leq \frac{1}{2}w_{\max}(C^*)$. Therefore, we obtain

$$\begin{aligned} f(C) &= w_{\max}(C) + p(E \setminus C) \\ &\leq \frac{w(C)}{k} + 2\left(2 - \frac{2}{k}\right)\delta_G(r) + p(E \setminus C) \\ &= w'(C) + p(E \setminus C) + 2\left(2 - \frac{2}{k}\right)\delta_G(r) \\ &\leq \frac{3}{2}w_{\max}(C^*) + \frac{3}{2}p(E \setminus C^*) + \left(2 - \frac{2}{k}\right)w_{\max}(C^*) \\ &\leq \left(\frac{7}{2} - \frac{2}{k}\right)w_{\max}(C^*) + \frac{3}{2}p(E \setminus C^*) \\ &\leq \left(\frac{7}{2} - \frac{2}{k}\right)f(C^*). \end{aligned}$$

This shows that the algorithm \mathcal{A}_{MM} is a $(\frac{7}{2} - \frac{2}{k})$ -approximation algorithm to solve the MM-RPCPP.

The complexity of the algorithm \mathcal{A}_{MM} can be determined as follows: (1) Using the RPCPP algorithm [16], the running time of Step 1 is $O(n^3)$ that determines a tour C . (2) Using Dijkstra's algorithm [13] to compute the eccentricity of depot r , the running time of Step 2 is $O(n^3)$. (3) Steps 4-5 need at most time $O(n^3)$ to convert the tour C to a feasible solution \mathcal{C} . Hence, the running time of the algorithm \mathcal{A}_{MM} is $O(n^3)$. \square

4 The restricted min-sum k -Chinese postman problem with penalties

In this section, we consider the RESTRICTED MIN-SUM k -CHINESE POSTMAN PROBLEM WITH PENALTIES (MS-RPCPP). Note that the MS-RPCPP is NP-hard since it is a generalization of metric TSP, where the metric TSP is known to be NP-hard [15, p.30].

From the description of the MS-RPCPP, we obtain a result in the following.

Lemma 5 *Any optimal solution \mathcal{C}^* to the MS-RPCPP, satisfies that all tours in \mathcal{C}^* collectively traverse each edge at most twice.*

Proof Given any instance $G = (V, E; w, p)$ to the MS-RPCPP, we know that each edge $e \in E$ appears in at most one tour $C \in \mathcal{C}^*$, where \mathcal{C}^* is any optimal solution of the MS-RPCPP. Now, we only need to consider any tour $C \in \mathcal{C}^*$.

Suppose the tour C traversing some edge e at least three times, we can construct a new Eulerian subgraph by simply deleting the two copy edges of the edge e on the Eulerian subgraph corresponding to C . It is clear that any Euler tour C_e on the new Eulerian subgraph also traverses all edges in E_C . As a result, we can replace C with C_e in \mathcal{C}^* , which can reduce the weight of \mathcal{C}^* , to get a better solution. This contradicts the optimality of \mathcal{C}^* . \square

By Lemma 5, the strategy to solve the MS-RPCPP is described as follows:

- (1) Find a connected spanning subgraph which has at least k edges incident to the depot, satisfying that the weight (with respect to function $\max\{0, w(\cdot) - p(\cdot)\}$) of the subgraph is minimized.
- (2) Partition the above subgraph into k subgraphs, and then construct a tour for each subgraph.

According to the above strategy, we design an algorithm (Algorithm 2) in the following.

Algorithm 2 : \mathcal{A}_{MS}

Input: An undirected graph $G = (V, E; w, p)$ with a depot r , a weight function $w : E \rightarrow \mathbb{R}^+$ satisfying the triangle inequality, a penalty function $p : E \rightarrow \mathbb{R}_0^+$ and an integer $k \in \mathbb{Z}^+$;
Output: A set $\mathcal{C} = \{C_1, \dots, C_k\}$ of tours starting from r and satisfying that each edge appears in at most one tour and each tour contains at least one edge, and the value $h(\mathcal{C}) = w(\mathcal{C}) + p(E \setminus \mathcal{C})$.

Begin

- Step 1:** Let $E_1 = \{e \in E \mid w(e) < p(e)\}$, $E_2 = E \setminus E_1 = \{e \in E \mid w(e) \geq p(e)\}$;
 - Step 2:** For the subset E_1 , find a subset $E_S \subseteq E_2$ of minimum weight with respect to w'' by applying the MST-DC algorithm, where $w''(e) = w(e) - p(e)$ for all $e \in E_2$, such that $G_{1,S} = (V, E_1 \cup E_S)$ is connected, and has at least k edges incident to the depot r ; If there is no such subset E_S , then output “no feasible solution on G ”, and stop;
 - Step 3:** Partition the graph $G_{1,S}$ into k subgraphs H_1, \dots, H_k , including depot r , and satisfying $\bigcup_{i=1}^k E(H_i) = E_1 \cup E_S$ and that $E(H_i) \cap E(H_j) = \emptyset$ for any $i, j \in \{1, \dots, k\}$; (A simple partition procedure is to, (1) find a spanning tree T on $G_{1,S}$ and partition T into k subtrees rooted at r , and then (2) allocate each edge in $G_{1,S} \setminus T$ to any associated subtree.)
 - Step 4:** For each $i \in \{1, \dots, k\}$, determine an Euler tour C_i of H_i by “doubling” its edges;
 - Step 5:** Output a set $\mathcal{C} = \{C_1, \dots, C_k\}$ of tours, and the value $h(\mathcal{C}) = w(\mathcal{C}) + p(E \setminus \mathcal{C})$.
- End**
-

By Algorithm 2, we obtain the following result for the MS-RPCPP.

Theorem 2 *The algorithm \mathcal{A}_{MS} is a 2-approximation algorithm to solve the MS-RPCPP and it runs in time $O(n^2)$.*

Proof It is clear to see that the algorithm can correctly indicate whether there is a feasible solution to the MS-RPCPP. We need only consider case that there are feasible solutions for the MS-RPCPP in the following.

For any instance $G = (V, E; w, p)$ of the MS-RPCPP, let \mathcal{C}^* denote an optimal solution of instance G to the MS-RPCPP, $E_{\mathcal{C}^*} \subseteq E$ denote the subset of edges traversed by tours in \mathcal{C}^* .

For an optimal solution \mathcal{C}^* to G as an instance of the MS-RPCPP, by using Lemma 5, we obtain a fact that all tours in \mathcal{C}^* collectively traverse each edge in E at most twice, meaning that the number of edges in $E_{\mathcal{C}^*}$ incident to r is at least k , and $(V, E_1 \cup (E_{\mathcal{C}^*} \setminus E_1))$ is a spanning connected subgraph of G . For the subset E_1 , Step 2 produces a subset $E_S \subseteq E_2$ of minimum weight with respect to w'' so that $G_{1,S} = (V, E_1 \cup E_S)$ is connected, and has at least k edges incident to the depot r . This implies that

$$w''(E_S) \leq w''(E_{\mathcal{C}^*} \setminus E_1).$$

For the set \mathcal{C} of tours produced at Step 5, since $E_1, E_S \subseteq E_{\mathcal{C}}$ and $E_1 \cap E_S = \emptyset$, we have $p(E \setminus E_{\mathcal{C}}) \leq p(E) - p(E_1) - p(E_S) = p(E_2) - p(E_S)$, which means

$$p(E_S) + p(E \setminus E_{\mathcal{C}}) \leq p(E_2).$$

It follows that

$$\begin{aligned} w(E_1) + w(E_S) + p(E \setminus \mathcal{C}) &= w(E_1) + w''(E_S) + p(E_S) + p(E \setminus E_{\mathcal{C}}) \\ &\leq w(E_1) + w''(E_{C^*} \setminus E_1) + p(E_2) \\ &= w(E_1) + w(E_{C^*} \setminus E_1) - p(E_{C^*} \setminus E_1) + p(E_2) \\ &= w(E_{C^*}) + w(E_1 \setminus E_{C^*}) + p(E_2 \setminus E_{C^*}) \\ &< w(E_{C^*}) + p(E \setminus E_{C^*}) \leq h(C^*). \end{aligned}$$

where the penultimate inequality follows from $w(E_1 \setminus E_{C^*}) < p(E_1 \setminus E_{C^*})$ because $w(e) < p(e)$ for all $e \in E_1$ (Fig. 1).

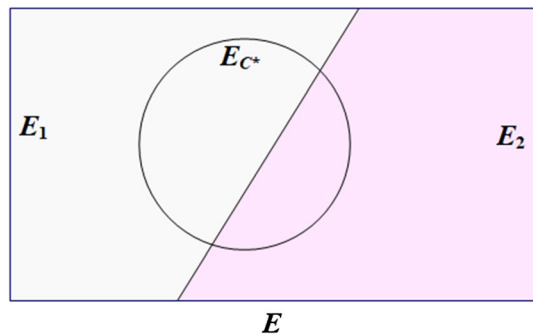
Thus, we obtain

$$\begin{aligned} h(\mathcal{C}) &= w(\mathcal{C}) + p(E \setminus \mathcal{C}) \\ &= 2(w(E_1) + w(E_S)) + p(E \setminus \mathcal{C}) \\ &\leq 2h(C^*). \end{aligned}$$

This shows that the algorithm \mathcal{A}_{MS} is a 2-approximation algorithm to solve the MS-RPCPP.

The complexity of the algorithm \mathcal{A}_{MS} can be determined as follows: (1) Using the MST-DC algorithm [9], the running time of Steps 1-2 is $O(n^2)$ that computes E_1 and E_S . (2) Using the BFS method [13, p.28-29] to partition the graph $G_{1,S}$ into k subgraphs H_1, \dots, H_k , the running time of Step 3 is $O(n^2)$. (3) Applying an algorithm in [6] to determine an Euler tour, the running time of Step 4 is $O(m)$ due to $\bigcup_{i=1}^k E(H_i) = E_1 \cup E_S$. Hence, the running time of the algorithm \mathcal{A}_{MS} is $O(n^2)$. \square

Fig. 1 The interpretation of $w(E_1) + w(E_{C^*} \setminus E_1) = w(E_{C^*}) + w(E_1 \setminus E_{C^*})$ and $-p(E_{C^*} \setminus E_1) + p(E_2) = p(E_2 \setminus E_{C^*})$



5 Conclusion and further work

In this paper, we consider the RESTRICTED MIN–MAX k -CHINESE POSTMAN PROBLEM WITH PENALTIES (MM-RPCPP) and the RESTRICTED MIN–SUM k -CHINESE POSTMAN PROBLEM WITH PENALTIES (MS-RPCPP), respectively. We obtain the following two main results:

- (1) We design a $\left(\frac{7}{2} - \frac{2}{k}\right)$ -approximation algorithm in time $O(n^3)$ to solve the MM-RPCPP.
- (2) We present a 2-approximation algorithm in time $O(n^2)$ to solve the MS-RPCPP.

In further work, we shall study other restricted routing problems with penalties. In particular, we intend to study the following problem (that we call the budgeted restricted Chinese postman problem with penalties).

Given an undirected graph $G = (V, E; w, p)$ with a depot $r \in V$ and a budget $B \in \mathbb{R}^+$, each edge has a weight, a traverse time and a penalty, it is asked to find a set of tours starting from r and collectively covering all vertices, such that the maximum traverse time of tours is at most B . The objective is to minimize the sum of weights of all tours plus the total penalty paid for uncovered edges.

Acknowledgements The authors are all grateful to an associate editor and all reviewers for their insightful comments and for their suggested changes that improve the presentation greatly.

References

1. Aráoz, J., Fernández, E., Meza, O.: Solving the prize-collecting rural postman problem. *Eur. J. Oper. Res.* **196**, 886–896 (2009)
2. Aráoz, J., Fernández, E., Zoltan, C.: Privatized rural postman problems. *Comput. Oper. Res.* **33**, 3432–3449 (2006)
3. Balas, E.: The prize collecting traveling salesman problem. *Networks* **19**(6), 621–636 (1989)
4. Benavent, E., Campos, V., Corberán, A., Mota, E.: The capacitated arc routing problem: lower bounds. *Networks* **22**, 669–690 (1992)
5. Bienstock, D., Goemans, M.X., Simchi-Levi, D., Williamson, D.: A note on the prize collecting traveling salesman problem. *Math. Program.* **59**, 413–420 (1993)
6. Edmonds, J., Johnson, E.L.: Matching, Euler tours and the Chinese postman. *Math. Program.* **5**(1), 88–124 (1973)
7. Frederickson, G.N., Hecht, M.S., Kim, C.E.: Approximation algorithms for some routing problems. *SIAM J. Comput.* **7**(2), 178–193 (1978)
8. Frieze, A.M.: An extension of Christofides heuristic to the k -person travelling salesman problem. *Discret. Appl. Math.* **6**, 79–83 (1983)
9. Glover, F., Klingman, D.: Finding minimum spanning trees with a fixed number of links at a node. In: *Combinatorial Programming: Methods and Applications* (Roy, B., Ed.), pp. 191–201. Dordrecht-Holland (1975)
10. Goemans, M., Williamson, D.: A general approximation technique for constrained forest problems. *SIAM J. Comput.* **24**, 296–317 (1995)
11. Guan, M.G.: Graphic programming using odd or even points (in Chinese). *Acta Math. Sinica* **10**, 263–266 (1960)
12. Karlin, A., Klein, N., Oveis Gharan, S.: A (slightly) improved approximation algorithm for metric TSP. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 32–45. Association for Computing Machinery, New York, USA (2021)

13. Korte, B., Vygen, J.: *Combinatorial Optimization: Theory and Algorithms*. Springer, Berlin (2018)
14. Pearn, W.L.: Solvable cases of the k -person Chinese postman problem. *Oper. Res. Lett.* **16**, 241–244 (1994)
15. Vazirani, V.V.: *Approximation Algorithms*. Springer, Berlin (2001)
16. Zhu, H.T., Pan, P.X.: The restricted Chinese postman problems with penalties. *Oper. Res. Lett.* **49**, 851–854 (2021)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.