The Hierarchical Chinese Postman Problem: the slightest disorder makes it hard, yet disconnectedness is manageable

Vsevolod A. Afanasev\textsuperscript{a}, René van Bevern\textsuperscript{a}, Oxana Yu. Tsidulko\textsuperscript{a,b}

\textsuperscript{a}Department of Mechanics and Mathematics, Novosibirsk State University, Novosibirsk, Russian Federation
\textsuperscript{b}Sobolev Institute of Mathematics of the Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russian Federation

Abstract

The Hierarchical Chinese Postman Problem is finding a shortest traversal of all edges of a graph respecting precedence constraints given by a partial order on classes of edges. We show that the special case with connected classes is NP-hard even on orders decomposable into a chain and an incomparable class. For the case with linearly ordered (possibly disconnected) classes, we get 5/3-approximations and fixed-parameter algorithms by transferring results from the Rural Postman Problem.

Keywords: approximation algorithm, fixed-parameter algorithm, NP-hardness, arc routing, rural postman problem, temporal graphs

1. Introduction

The following NP-hard arc routing problem was formally introduced by Dror et al. [9] and arises in snow plowing, garbage collection, flame and laser cutting [24].

\textbf{Problem 1.1} (Hierarchical Chinese Postman Problem, HCPP). Input: An undirected graph $G = (V, E)$, edge weights $\omega : E \rightarrow \mathbb{N}$, a partition $\mathcal{P}$ of $E$ into $k$ classes, a partial order $\prec$ on $\mathcal{P}$. Find: A least-weight closed walk traversing each edge in $E$ at least once such that each edge $e$ in a class $E' \prec E''$ is traversed only after all edges in all classes $E'' \prec E'$ are traversed.

The case $k = 1$ is the \textit{Chinese Postman Problem} (CPP), which reduces to a minimum-weight perfect matching problem [5, 7, 10, 29]. We study the following special cases of HCPP:

- $\text{HCPP}(l)$: the order $\prec$ is linear,
- $\text{HCPP}(c)$: each edge class induces a connected subgraph,
- $\text{HCPP}(c, l)$: both of the above restrictions.

HCPP(l) and HCPP(c, l) can also be understood as variants of the Travelling Salesman Problem (TSP) in temporal graphs [27], with the difference that it is required to explore all edges instead of all vertices and that edges never disappear from the graph. HCPP(c, l) is polynomial-time solvable [9, 15, 22]. This naturally raises questions about HCPP(c) and HCPP(l):

(a) Is HCPP(c) effectively solvable on other restricted order types, like several scheduling problems on, for example, tree orders [19], interval orders [28], and bounded-width orders [1, 31]? HCPP(c) was even conjectured to be polynomial-time solvable when the number of classes is constant [3].

(b) Is HCPP(l) effectively solvable when the number of connected components in each class is sufficiently small? If the number of connected components is unbounded, HCPP(l) is NP-hard already for $k = 2$ [4].

Our contributions. In Section 3, we show that HCPP(c) is NP-hard even on partial orders that are decomposable into a linear order and one incomparable class, thus negatively answering (a) for all the order types mentioned there, in particular for orders of width two. The remaining sections are dedicated to question (b).

In Section 4, we revisit a construction of Dror et al. [9] that reduces HCPP(c, l) to the $s$-$t$-Rural Postman Path Problem ($s$-$t$-RPP, Problem 4.1). We show that, when applied to HCPP(l), the construction transfers performance guarantees of approximation and randomized algorithms from $s$-$t$-RPP to HCPP(l).

In Section 5, we show a 5/3-approximation algorithm for HCPP(l). This contrasts TSP in temporal graphs, which is not better than 2-approximable unless $P = \mathbb{NP}$ [27]. To get the 5/3-approximations for HCPP(l), we use the construction from Section 4 and show a 5/3-approximation algorithm for $s$-$t$-RPP analogously to Hoogeveen’s [18] adaption of Christofides’ and Serdyukov’s 3/2-approximation from TSP [5, 8, 30] to $s$-$t$-TSP. Any better approximation factor for $s$-$t$-RPP will directly yield a better approximation factor for HCPP(l).

In Section 6, we use the construction from Section 4 to show that HCPP(l) is solvable in polynomial-time if each class induces a constant number $c$ of connected components. When the edge weights are polynomially bounded, one can even obtain randomized fixed-parameter algorithms with respect to $c$, using an algorithm for the Rural Postman Problem due to Gutin et al. [16].

2. Preliminaries

By $\mathbb{N}$, we denote the set of natural numbers, including zero. For two multisets $A$ and $B$, $A \sqcup B$ is the multiset obtained by adding the multiplicities of elements in $A$ and $B$. By $A \setminus B$ we denote the multiset obtained by subtracting the multiplicities of elements in $B$ from the multiplicities of elements in $A$. Finally, given some weight function $\omega : A \rightarrow \mathbb{N}$, the weight of a multiset $A$ is $\omega(A) := \sum_{e \in A} \nu(e) \omega(e)$, where $\nu(e)$ is the multiplicity of $e$ in $A$. 

\textit{Email addresses:} v.afanasev30@nsu.ru (Vsevolod A. Afanasev), rvb@nsu.ru (René van Bevern), o.tsidulko@nsu.ru (Oxana Yu. Tsidulko)
We mostly consider simple undirected graphs $G = (V,E)$ with a set $V(G) := V$ of vertices, a set $E(G) := E \subseteq \{(u,v) \mid u,v \in V, u \neq v\}$ of edges. Unless stated otherwise, $n$ denotes the number of vertices and $m$ denotes the number of edges. Within proofs, there may occur multigraphs, where $E$ is a multiset, and directed graphs $G = (V,A)$ with a set of arcs $A \subseteq V^2$. The degree of a vertex in an undirected multigraph is its number of incident edges. We call a vertex balanced if it has even degree. We call a graph balanced if all its vertices are balanced. For a multiset $R$ of edges, we denote by $V(R)$ the set of their incident vertices. For a multiset $R$ of edges of $G$, $G(R) := (V(R), R)$ is the (multi)graph induced by the edges in $R$.

A walk from $v_0$ to $v_1$ in a graph $G$ is a sequence $w = (v_0, e_1, v_1, e_2, v_2, \ldots, e_i, v_i)$ such that $e_i = [v_{i-1}, v_i]$ (if $G$ is undirected) or $e_i = (v_{i-1}, v_i)$ (if $G$ is directed) for each $i \in [1, \ell]$. When there is no ambiguity (like in simple graphs), we will also specify walks simply as a list of vertices. If $v_0 = v_\ell$, then we call $w$ a closed walk. If all vertices on $w$ are pairwise distinct, then $w$ is a path. If only its first and last vertex coincide, then $w$ is a cycle. A subwalk $w'$ of $w$ is any subsequence of $w$ that is itself a walk. By $E(w)$, we denote the multiset of edges on $w$, that is, each edge appears on $w$ and in $E(w)$ equally often. The weight of a walk $w$ is $\omega(w) := \sum_{e \in E(w)} \omega(e)$. For a walk $w$, we denote $G(w) := G(E(w))$. Note that $G(R)$ and $G(w)$ do not contain isolated vertices yet might contain edges with a higher multiplicity than $G$ and, therefore, are not necessarily sub(multi)graphs of $G$. An Euler walk for $G$ is a walk that traverses each edge or arc of $G$ exactly as often as it is present in $G$. An Euler tour is a closed Euler walk. A graph is Eulerian if it allows for an Euler tour. A connected undirected graph is Eulerian if and only if all its vertices are balanced. For any $\alpha \geq 1$, an $\alpha$-approximate solution for a minimization problem is a feasible solution whose weight does not exceed the weight of an optimal solution by more than a factor of $\alpha$, called the approximation factor [14]. The Exponential Time Hypothesis (ETH) is that 3-SAT (Problem 3.2) with $n$ variables is not solvable in $2^{o(n)}$ time [20].

3. NP-hardness for HCPP(c) with one incomparable class

HCPP(c,l) is polynomial-time solvable [9, 15, 22]. We show that adding an incomparable class makes the problem NP-hard.

**Theorem 3.1.** Even on orders decomposable into a linear order and one incomparable class, and if all edges have weight one, (i) HCPP(c) is NP-hard and (ii) not solvable in $2^{o(n+m+k)}$ time unless ETH fails.

To prove Theorem 3.1, we use a polynomial-time many-one reduction from 3-SAT, which is NP-hard [21] and, unless the ETH fails, is not solvable in $2^{o(n+m)}$ time [20].

**Problem 3.2 (3-SAT).**

**Input:** A formula $\varphi$ in conjunctive normal form with $n$ variables and $m$ clauses, each containing at most three literals.

**Question:** Is there a assignment to the variables satisfying $\varphi$?

The reduction is carried out by the following construction, which is illustrated in Figure 1.

**Construction 3.3.** Let $\varphi$ be an instance of 3-SAT. First, delete each clause containing both $x$ and $\bar{x}$: they are always satisfied. Consider now the variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$. For each $i \in [1, \ldots, n]$, let $u_i$ denote the number of clauses containing either $x_i$ or $\bar{x}_i$. We describe an instance $I_\varphi = (G, \omega, P, \prec)$ of HCPP(c). All edges of $G$ will have weight one.

Graph $G$ contains a path $(c_i, c_i^*, c_i)$ for each $j \in [1, \ldots, m]$ and, for each $i \in [1, \ldots, n]$ and $\ell \in [1, \ldots, 6u_i]$, a path $P_i^\ell := ((t_i^\ell, c_i^*)^\ell, f_i^\ell)$. For each $i \in [1, \ldots, n]$, it contains a cycle

$$X_i := (t_i^0, t_i^1, \ldots, t_i^{6u_i-1}, t_i^6, i^0, t_i^1, \ldots, t_i^1, t_i^0).$$

For each literal $x_i$ in a clause $C_j$, graph $G$ contains a cycle

$$Z_{ij} := (f_i^{\ell_1}, c_j^1, a_i, c_j^1, t_j^1, f_j^{\ell_2}),$$

where $\ell \leq \mu_i$ is such that $C_j$ is the $\ell$-th clause containing $x_i$ or $\bar{x}_i$. For each literal $x_i$ in a clause $C_j$, graph $G$ contains a cycle

$$Z_{ij} := (f_i^{\ell_1}, c_j^1, a_i, c_j^1, t_j^1, f_j^{\ell_2}),$$

where $\ell \leq \mu_i$ is such that $C_j$ is the $\ell$-th clause containing $x_i$ or $\bar{x}_i$. The edges are ordered as follows. For each $i \in [1, \ldots, n]$ and $j \in [1, \ldots, 4\mu_i]$, $E_i^j = E(P_i^j)$ is a connected class. They are lexicographically ordered, that is,

$$E_i^j < E_i^{j'} \iff (i < i') \lor (i = i' \land j \leq j').$$

They are preceded by the connected class

$$E_0 = \bigcup_{i=1}^n E(X_i) \cup \bigcup_{i=1}^n E(Y_{i \mod n+1}) \cup \bigcup_{x \in C_j} E(Z_{i,x}) \cup \bigcup_{x \in C_j} E(\tilde{Z}_{i,x}).$$

Finally, the edge set $E^*$ consists of all edges incident to $c^*$ forms an incomparable connected class.

For convenience, we collect the vertices of $I_\varphi$ of the form $t_i^j$ and $f_i^j$ and of the form $c_i^j$ and $c_j^i$ into sets

$$V_{\text{FT}} := \bigcup_{j=1}^m V_i^j \text{ and } V_C := \bigcup_{j=1}^m \{c_j^1, c_j^2\}.$$
Figure 1: Illustration of Construction 3.3. The graph is generated from the formula $\varphi = (t_1 \lor x_2) \land (s_2 \lor x_3)$, that is, $C_1 = (t_1 \lor x_2)$ and $C_2 = (s_2 \lor x_3)$. The dotted edges form the edge set $E_0$. The solid edges form the paths $P_i$ and $(v, e^*, c^*, c^2)$. The gray areas illustrate the types of cycles introduced in the construction: they consist of the dashed edges enclosed in the gray areas. Note that $E_0$ forms an Eulerian subgraph: it is the union of cycles and thus each vertex has an even number of incident edges in $E_0$. Thus, each vertex of the form $t_i, f_i, c_i^1$, and $c_i^2$ is imbalanced and they are the only imbalanced vertices.

Figure 2: A minimum-weight closed walk for the graph in Figure 1 first visits each edge in $E_0$ exactly once (the dotted edges in Figure 1), and then follows the arrows as shown in this figure. The walk corresponds to $x_1 = 0$ and $x_2 = x_3 = x_4 = 1$. 
Definition 3.5. A closed walk for an HCPP(∈ instance $(G, \omega, \mathcal{P}, <)$ with $G = (V, E)$ is a tight tour if it has weight at most $|E| + b/2$, where $b$ is the number of imbalanced vertices in $G$.

Proposition 3.6. The HCPP(∈ instance $I_{p} = (G, \omega, \mathcal{P}, <)$ created from a 3-SAT instance $\varphi$ by Construction 3.3 allows for a tight tour if and only if $\varphi$ is satisfiable.

In the rest of this section, it remains to prove Proposition 3.6, which, together with Observation 3.4(iii), yields Theorem 3.1.

Satisfiability of $\varphi$ implies a tight tour in $I_p$. Assume that $\varphi$ is satisfiable. We show a tight tour $T$ for $I_p$. Without loss of generality, assume that $x_1$ is “true”; otherwise, we can replace $x_1$ by $\overline{x_1}$ throughout the formula $\varphi$.

The tight walk $T$ for $I_p$ then looks as follows (an example is shown in Figure 2). It starts in $f_1$, first visits each edge of $E_0$ exactly once and returns to $f_1$. This is possible by Observation 3.4(i). Then, it remains to traverse the paths $(f_i, f'_c, f'_j)$ for each $i \in \{1, \ldots, n\}$ and $\ell \in \{1, \ldots, 6\mu_1\}$ and the paths $(c_i^j, c'_i, c'_j)$ for each $j \in \{1, \ldots, m\}$. This is done as follows. For $i$ from 1 to $n$, if $x_i$ is “true”, $T$ visits the vertices $f_1, f_1, f_2, f_2, f_3, f_3, \ldots, f_{6\mu_1}, f_{6\mu_1}, f_{6\mu_1}$, for some $\ell \in \{1, \ldots, \mu_1\}$ taking a detour through the vertices $f_{6\ell-3}^i, c_i, c_i, c'_i, c''_i, f_{6\ell-2}^i$ if clause $C_j$ contains $x_i$ and $(c_i^j, c'_i, c'_j)$ has not been traversed before. If $x_i$ is “false”, then $T$ visits $f_1, f_1, f_2, f_2, f_3, f_3, \ldots, f_{6\mu_1}, f_{6\mu_1}, f_{6\mu_1}$, for some $\ell \in \{1, \ldots, \mu_1\}$ taking a detour through the vertices $f_{6\ell-3}^i, c_i, c_i, c'_i, c''_i, f_{6\ell-2}^i$ if clause $C_j$ contains $\overline{x_i}$ and $(c_i^j, c'_i, c'_j)$ has not been traversed before. Finally, after $f_{6\mu_1}^i$ or $f_{6\mu_1}^i$, the walk $T$ returns to $f_1$. Note that this traversal is possible due to the cycle $Y_{i, \text{mod } n+1}$ for each $i \in \{1, \ldots, n\}$.

Observe that the closed walk $T$ contains all edges and respects precedence constraints: for each edge in $E_0$ and all paths $(f_i, c_i, f_j)$ for $i \in \{1, \ldots, n\}$ and $\ell \in \{1, \ldots, \mu_1\}$, this is obvious. To see that the path $(c_i^j, c'_i, c'_j)$ has been traversed for each $j \in \{1, \ldots, m\}$, observe that each clause $C_j$ contains a true literal, so that a detour via $c_i^j, c'_i, c'_j$ is taken.

To see that $T$ is tight, we check which edges are traversed a second time. When $x_i$ is “true”, the edges $\{f_{2\ell}^i, f_{2\ell+1}^i\} \in E_0$ for $\ell \in \{1, \ldots, \mu_1 - 1\}$ are visited a second time, whereas each edge $\{f_{2\ell-1}^i, f_{2\ell}^i\} \in E_0$ for $\ell \in \{1, \ldots, \mu_1\}$ is traversed a second time or skipped by a detour that traverses the edges $\{f_{2\ell-1}^i, c'_i\}$ and $\{f_{2\ell}^i, c'_i\}$ a second time. Analogously when $x_i$ is “false”. Moreover, for each $i \in \{1, \ldots, n\}$, one edge of the cycle $Y_{i, \text{mod } n+1}$ is visited a second time: it joins the last vertex visited by $T$ in $X_i$ to the first vertex visited by $T$ in $X_i$. We thus see that the edges visited a second time form a matching. Their endpoints are the $b$ imbalanced vertices $V_{FT} \cup V_C$. Thus, $T$ traverses not more than $|E| + b/2$ edges.

Tight tour for $I_p$ implies satisfiability of $\varphi$. Assume that $I_p$ allows for a tight tour $T$. We show that $\varphi$ is satisfiable.

Lemma 3.7. Let $M = E(T) \setminus E$, that is, $M$ is the multiset of the edges that $T$ traverses additionally to $E$ (taking into account the multiplicity of additional visits). Then,

(i) $M \subseteq E_0$ is a perfect matching on the vertices $V_{FT} \cup V_C$, in particular, $M$ contains each edge at most once,

(ii) each edge in $M$ has an endpoint in $V_{FT}$, and

(iii) $T_1$ is an Euler tour for $G(E_0)$, and

(iv) $T_2$ is an Euler tour for $G((E \setminus E_0) \cup M)$.

Proof. (i) Since $T$ is a closed walk, all vertices in $G(T)$ are balanced, whereas its subgraph $G(E) = G$ has $b$ imbalanced vertices. Since $T$ contains at most $|E| + b/2$ edges, the graph $G(T)$ contains at most $b/2$ additional edges. Thus, $G(T)$ contains a set $M$ of at most $b/2$ edges whose endpoints are the $b$ imbalanced vertices of $G(E)$. By Observation 3.4(ii), these are exactly the vertices $V_{FT} \cup V_C$. This is only possible if $M$ is a perfect matching on $V_{FT} \cup V_C$. Since each edge of $G$ that is not in $E_0$ has at least one balanced endpoint (namely, $c^*$ or one of the $c'_j$), we easily get $M \subseteq E_0$.

(ii) The only edges in $E_0$ that have no endpoints in $V_{FT}$ have one of the vertices of the form $a_i$ or $b_j$ as endpoints. Since these are only on the cycle $Z_{ij}$ or $Z_{ij}$, they are balanced. Thus, $M$ cannot contain such edges.

(iii) Let $T_1$ be the minimal prefix of $T$ traversing all edges in $E_0$. Towards a contradiction, assume that $T_1$ is not a closed walk. Then, the last vertex in $T_1$ is balanced in $G(E_0)$ by Observation 3.4(i), yet not balanced in its supergraph $G(T_1)$. Thus, the last edge in $T_1$ does not belong to $E_0$ or is traversed by $T_1$ more than once, which contradicts the minimality of $T_1$. Thus, due to the precedence constraints, $T_1$ is a closed walk in $G(E_0 \cup E^*)$ traversing all of $E_0$.

We show that $T_1$ traverses each edge $e \in E_0 \cup E^*$ at most once. Towards a contradiction, assume that it traverses $e = [u, v]$ twice. Then, $v \in V_{FT}$ by (ii). Thus, $v$ is not incident to any edges in $E^*$ and, because $v$ is balanced in $G(E_0)$ by Observation 3.4(i), it is also balanced in $G(E_0 \cup E^*)$. Since $v$ is balanced in $G(E_0 \cup E^*)$, balanced in $G(T_1)$, and $T_1$ traverses $e$ twice, $T_1$ also traverses another edge incident to $v$ twice, contradicting (i).

We finally show that $T_1$ contains only edges in $E_0$. Towards a contradiction, assume that $T_1$ contains an edge $[c^*, c] \in E^*$. Vertex $c \in V_C$ is balanced in $G(T_1)$, yet not balanced in its subgraph $G(E_0 \cup [c^*, c])$ by Observation 3.4(i). Thus, $G(T_1)$ contains some edge $e \in E_0 \cup E^*$ twice, which is impossible.

(iv) now follows easily from (i) and (ii): $T_2$ has to visit all edges in $E \setminus E_0$ and all edges in $M$, which are not visited by $T_1$. The budget of $|E| + b/2 = |E| + |M|$ does not allow it to visit any edge in $(E \setminus E_0) \cup M$.

In the following, our aim is showing that the matching $M$, which exists by Lemma 3.7, takes only one of two possible forms in each variable cycle $X_i$. This will correspond to setting a variable to “true” or “false”.

Definition 3.8. Let $i \in \{1, \ldots, n\}$ and $\ell \in \{1, \ldots, \mu_1\}$. We call an edge $\{f_{6\ell-3}^i, f_{6\ell-2}^i\}$ covered if $\{f_{6\ell-3}^i, f_{6\ell-2}^i\} \in M$ or if there is a $j \in \{1, \ldots, m\}$ such that both $\{f_{6\ell-3}^i, c_j^i\}$ and $\{f_{6\ell-2}^i, c_j^i\}$ are in $M$.  

We call an edge \( \{f^{6\ell-3}, f^{6\ell-2}\} \) covered if \( \{\ell_i^{6\ell-3}, f_i^{6\ell-2}\} \in M \) or if there is a \( j \in \{1, \ldots, m\} \) such that both \( \{\ell_i^{6\ell-3}, c_j^1\} \) and \( \{f_i^{6\ell-2}, c_j^2\} \) are in \( M \).

**Lemma 3.9.** For each \( i \in \{1, \ldots, n\} \), either all \( \{\ell_i^{6\ell-3}, f_i^{6\ell-2}\} \) are covered or all \( \{f^{6\ell-3}, f^{6\ell-2}\} \) are covered for \( \ell \in \{1, \ldots, \mu\} \).

**Proof.** For an arbitrary \( i \in \{1, \ldots, n\} \) and \( \ell \in \{1, \ldots, \mu\} \), we first show that exactly one of the edges \( \{\ell_i^{6\ell-3}, f_i^{6\ell-2}\} \) and \( \{f_i^{6\ell-3}, f_i^{6\ell-2}\} \) is covered. Note that, by Construction 3.3, at most one of these pairs of vertices is attached to \( \{c_j^1, c_j^2\} \) for any \( j \in \{1, \ldots, m\} \). Without loss of generality, let this be \( \{f_i^{6\ell-3}, f_i^{6\ell-2}\} \). The other case is symmetric.

Denote \( R := E \setminus E_0 \). We exploit that, by Lemma 3.7(iv), \( G(R \cup M) \) is connected. Thus, there is at least one edge of \( M \) leaving any subset of connected components of \( G(R) \). Therefore, for each \( j \in \{6 \ell - 5, \ldots, 6 \ell - 1\} \), only one of \( \{\ell_i^j, f_i^{j+1}\} \) and \( \{f_i^j, f_i^{j+1}\} \) is in \( M \); otherwise, \( M \) could not contain any edge leaving the set of connected components \( \{t_i^j, c_i^1, f_i^{j+1}\} \). We exploit also that, by Lemma 3.7(i), all vertices in \( V_{FT} \) must be incident to an edge of \( M \).

We distinguish two cases, illustrated in Figure 3. First, assume that \( \{f_i^{6\ell-3}, f_i^{6\ell-2}\} \) is covered. That is, it is in \( M \). Then all the other bold edges shown in Figure 3a must also be in \( M \). Thus, the edge \( \{f_i^{6\ell-3}, f_i^{6\ell-2}\} \) is not covered. If, on the contrary, the edge \( \{f_i^{6\ell-3}, f_i^{6\ell-2}\} \) is not covered, that is, not in \( M \), then all the bold edges shown in Figure 3b must be in \( M \). To match the vertices \( t_i^{6\ell-3} \) and \( f_i^{6\ell-2} \), one either has \( \{\ell_i^{6\ell-3}, f_i^{6\ell-2}\} \in M \) or \( \{f_i^{6\ell-3}, c_i^1\} \). We exploit that, by Construction 3.3, since \( \ell < \mu \) and \( \ell \neq \mu \), there is an arc \( (u, v) \). Dror et al. [9] reduce HCPP(c,l) to multiple s-t-RPP instances in which the subgraph \( G(R) \) is connected. Since this case of s-t-RPP is polynomial-time solvable, this yields a polynomial-time algorithm for HCPP(c,I) [9].

We now show that, while applying the same construction to HCPP(l), it does not yield polynomial-time solvable instances of s-t-RPP, it allows to transfer running times, approximation factors, and error probabilities of s-t-RPP algorithms to HCPP(l).\(^1\) We start by describing the construction.

**Definition 4.2.** In this section, we denote the edge classes of HCPP(l) instances \( (G, \omega, \mathcal{P}, \prec) \) by \( E_1, \ldots, E_k \), where \( E_j \prec E_j \) if and only if \( 1 \leq i < j \leq k \).

By \( R[u, v, i] \), we denote the s-t-RPP instance of finding a minimum-weight walk between the vertices \( u \) and \( v \) in \( G(E_1 \cup \cdots \cup E_i) \) traversing all edges in \( E_i \). By \( P[u, v, i] \), we denote an arbitrary optimal solution to \( R[u, v, i] \).

**Construction 4.3.** From a HCPP(l) instance \( (G, \omega, \mathcal{P}, \prec) \), construct a directed arc-weighted graph \( G = (V, F, \omega) \) as illustrated in Figure 4: The vertex set \( V_i = \bigcup_{j=1}^{i-1} V_j \) is a union of layers \( V_j \). Each layer \( V_j \) for \( i \in \{2, \ldots, k\} \) contains a copy of each vertex in \( G \) that is incident to an edge of \( E_j \) and of any ancestor class. Precisely,

\[
V_i = \{u_i \mid u_i \in V(E_1)\} \quad \text{for } i \in \{1, k + 1\}
\]

\[
V_i = \left\{u_i \mid u_i \in V(E_i) \cap \bigcup_{j=1}^{i-1} V(E_j)\right\} \quad \text{for } i \in \{2, \ldots, k\}.
\]

For each pair of vertices \( u_i, v_i + v_i \in V_i, i \in \{1, \ldots, k\} \), there is an arc \( (u_i, v_i + v_i) \in A_F \) of weight \( \omega(v_i, v_i + v_i) = \omega(P[u, v, i]) \). If \( P[u, v, i] \) does not exist, there is no arc \( (u_i, v_i + v_i) \).

The following has been shown by Dror et al. [9].

**Proposition 4.4.** Let \( I := (G, \omega, \mathcal{P}, \prec) \) be an HCPP(l) instance and \( \Gamma \) be constructed from \( I \) by Construction 4.3. Then, the weight of an optimal solution to \( I \) coincides with the least weight of any layer path in \( \Gamma \), where a layer path in \( \Gamma \) is a path from \( v_1 \in V_1 \) to \( v_k \in V_{k+1} \) such that \( v_1 \prec v_2 \prec \cdots \prec v_k \).

\(^1\) Cabral et al. [6] showed a polynomial-time reduction of HCPP(l) to RPP. It, however, does not allow to transfer approximation factors, since it introduces very heavy required edges. Since these always contribute to the goal function, this makes bad approximate solutions “look” good.
In particular, each layer path in $\Gamma$ has the form $J = (v_1, y_2, y_3, \ldots, y_k, v_{k+1})$, where $y_i \in V_i$ for $i \in \{2, \ldots, k\}$ and concatenating the corresponding walks $P[v, y_1, 1], P[y_1, y_2, 2], \ldots, P[y_k, v, k]$ yields a feasible solution $W_I$ of weight $\omega(W_I) = \omega(J)$ for $I$.

Construction 4.3 can be used to solve HCPP($G, l$) in $O(kn^3)$ time: $\Gamma$ has at most $kn^2$ arcs, the weight of each is computed by solving an $s$-$t$-RPP instance $R[u, v, i]$, which works in $O(n^3)$ time since the set $E_i$ of required edges is connected [9]. It remains to find a layer path in $\Gamma$. This can be done in $O(n^3)$ time by $n$ times calling a linear-time single-source shortest-path algorithm for directed acyclic graphs.

However, when applied to HCPP($I$), Construction 4.3 gets to solve $s$-$t$-RPP instances $R[u, v, i]$ where the set of required edges $E_i$ might be disconnected. Since we do not know how to solve them in polynomial time, in Sections 5 and 6, we will solve them using approximation algorithms and randomized fixed-parameter algorithms. We now show how their performance guarantees will carry over to HCPP($I$).

**Lemma 4.5.** Let $I = (G, \omega, \mathcal{P}, <)$ be an HCPP($I$) instance. Assume that there is an algorithm running in $t$ time, that, given any $s$-$t$-RPP instance $R[u, v, i]$ (cf. Definition 4.2), outputs an $\alpha$-approximate solution for $R[u, v, i]$ with probability at least $1 - p$.

Then, there is an algorithm running in $O(kn^3 + kn^2)$ time that returns an $\alpha$-approximate solution for $I$ with probability at least $1 - pk$.

**Proof.** Let $\mathcal{A}$ denote the assumed randomized approximation algorithm for solving $s$-$t$-RPP instances $R[u, v, i]$. Since we can check the feasibility of any solution returned by $\mathcal{A}$ in linear time, we can assume that $\mathcal{A}$ makes only one-sided errors: for a feasible instance $R[u, v, i]$ with probability $p$, it does not find a solution or produces a solution that is not $\alpha$-approximate. Moreover, since feasibility of $I$ is easy to check [9], we will assume that $I$ has a feasible solution and compute a solution to $I$ as follows.

Construct an arc-weighted directed graph $\Gamma = (V_I, \mathcal{A})$ from $G$ as described in Construction 4.3, yet for each $i \in \{1, \ldots, k\}$ and every $u_i \in V_i$ and $v_{i+1} \in V_{i+1}$, the weight $\omega(\mathcal{A}(u_i, v_{i+1})) = \omega(P[u_i, v_{i+1}])$, where $P[u_i, v_{i+1}]$ is computed by applying $\mathcal{A}$ to the $s$-$t$-RPP instance $R[u_i, v_{i+1}]$. Finally, try to compute a least-weight path layer $J$ in $\Gamma$. If it exists, then the corresponding closed walk $W_J$ is a feasible solution of weight $\omega(W_J) = \omega(J)$ for $I$. The running time of the whole procedure is $O(kn^3 + kn^2)$ since the graph $\Gamma$ has $O(kn^2)$ arcs, the weight of each can be computed in $t$ time, and the least-weight layer path in $\Gamma$ can finally be found by $n$ times applying a single-source shortest-path algorithm for directed acyclic graphs. It remains to analyze the probability that the procedure returns an $\alpha$-approximate solution for $I$.

To this end, let $W^*$ be an optimal solution to $I$, $\Gamma = (V_I, \mathcal{A})$ be constructed by Construction 4.3 from $I$, and $J^* = (x_1, y_2, \ldots, y_k, x_{k+1})$ be a least-weight layer path in $\Gamma$. First, assume that $\mathcal{A}$ indeed produced an $\alpha$-approximate solution for each instance $R[u, v, i]$ corresponding to any arc $(u_i, v_{i+1})$ on $J^*$. Then, for each arc $(u_i, v_{i+1})$ on $J^*$,

$$\omega(\mathcal{A}(u_i, v_{i+1})) = \omega(P[u_i, v_{i+1}]) \leq \alpha \omega(R[u_i, v_{i+1}]) = \alpha \omega(J^*) = \alpha \omega(W^{*})$$

and $J^*$ witnesses the existence of the computed least-weight layer path in $\Gamma$. Thus, the weight $\omega(W_J) = \omega(J)$ is at most

$$\omega(J^*) = \omega(x_1, y_2) + \omega(y_2, y_3) + \cdots + \omega(y_k, x_{k+1}) \leq \alpha \omega(x_1, y_2) + \alpha \omega(y_2, y_3) + \cdots + \alpha \omega(y_k, x_{k+1}) = \alpha \omega(J^*) = \alpha \omega(W^*)$$

If the described procedure fails to produce an $\alpha$-approximate solution for $I$, then, by contraposition, $\mathcal{A}$ failed to produce an $\alpha$-approximate solution for at least one $s$-$t$-RPP instance $R[u, v, i]$.
corresponding to an arc \((u_i, v_{i+1})\) on \(J^*\). Since \(J^*\) has \(k\) arcs, this happens with probability at most \(k p\) by the union bound. 

5. A 5/3-approximation algorithm for HCPP(I)

We now show a polynomial-time 5/3-approximation algorithm for \(s \times t\)-RPP. Using Lemma 4.5, this directly yields a polynomial-time 5/3-approximation algorithm for HCPP(I). The algorithm is an adaption of the Christofides-Serdyukov-like 3/2-approximation algorithm from RPP \([4, 11]\) to \(s \times t\)-RPP. It closely follows Hoogeveen’s [18] adaption of the Christofides-Serdyukov 3/2-approximation algorithm from metric TSP \([5, 8, 30]\) to metric \(s \times t\)-TSP.

**Theorem 5.1.** The \(s \times t\)-RPP is 5/3-approximable in \(O(n^3)\) time.

**Proof.** We assume \(s \neq t\) (otherwise, one can add a dummy vertex \(s \neq r\) and an edge \((s, t)\) of zero weight to the initial graph). We only show the 5/3-approximation algorithm for \(s \times t\)-RPP instances \(I := (G, R, \omega, s, t)\) such that \(G = (V, E)\) is a complete graph on the vertex set \(V = (R)\) and such that the weight function \(\omega\) satisfies the triangle inequality. This is enough, since the general case reduces to this special case in \(O(n^3)\) time and any \(\alpha\)-approximation for the special case yields an \(\alpha\)-approximation for the general case [4]. The 5/3-approximation algorithm works in four steps.

**Step 1.** Compute a set \(T \subseteq E'\) of edges of minimum total weight such that \(G(R \cup T)\) is connected (for example, using Kruskal’s algorithm [23]).

**Step 2.** Let \(S \subseteq V'\) be the set of vertices in \(V \setminus \{s, t\}\) that are imbalanced in \(G(R \cup T)\) and of those vertices in \(\{s, t\}\) that are balanced in \(G(R \cup T)\). Note that \(|S|\) is even: Indeed, consider the set \(|S'|\) of all vertices that are imbalanced in \(G(R \cup T)\). Clearly, \(|S'|\) is even. Now, if \(s, t \in S'\), then \(S = S' \setminus \{s, t\}\). If \(s, t \notin S'\), then \(S = S' \cup \{s, t\}\). If \(s \in S'\) and \(t \notin S'\) (or vice versa), then \(S = S' \setminus \{t\}\) (or \(S = S' \setminus \{s\}\)). Thus, \(|S|\) is even.

**Step 3.** Construct a minimum-weight perfect matching \(M \subseteq E\) on the vertices of \(S\) in \(G\) (for example, using Lawler’s algorithm [25, Section 6.10]).

**Step 4.** Return an Euler walk \(P\) in \(G(R \cup T \cup M)\). Note that \(P\) exists (and can be computed using Hierholzer’s algorithm [12, 17]) since \(G(R \cup T \cup M)\) is connected and all its vertices except for \(s\) and \(t\) are balanced. Thus, the endpoints of \(P\) are \(s\) and \(t\) and \(P\) is a feasible solution to \(I\).

All steps can be carried out in \(O(n^3)\) time. It remains to prove that \(P\) is a 5/3-approximation. To this end, let \(P^*\) be an optimal solution for \(I\). Obviously, \(\omega(R \cup T) \leq \omega(P^*)\). Thus, it remains to show \(\omega(M) \leq 2 \cdot 3 \cdot \omega(P^*)\). To this end, consider \(Q = E(P^*) \cup R \cup T\). We will construct three perfect matchings \(M_1, M_2,\) and \(M_3\) on \(S\) in \(G\) such that \(\omega(M_1) + \omega(M_2) + \omega(M_3) \leq \omega(Q)\), and thus \(\omega(M) \leq 1/3 \cdot \omega(Q) \leq 2/3 \cdot \omega(P^*)\).

Since the imbalanced vertices of \(G(P^*)\) are exactly \(s\) and \(t\), the unbalanced vertices in \(G(Q)\) are exactly those in the set \(S\). Let the vertices of \(S = \{v_1, v_2, \ldots, v_{2t}\}\) be numbered in the order of their first occurrence on \(P^*\) and let \(P^*\) be the subwalk of \(P^*\) between the vertices \(v_{2i-1} \in S\) and \(v_{2i} \in S\) for all \(i \in [1, \ldots, t]\). Let

\[ E_1 := \left\{ \frac{t}{i=1} E(P^*) \right\}. \]

By shortcutting each path \(P^*\) to one edge, one gets a perfect matching \(M_1\) on the vertices of \(S\) such that \(\omega(M_1) \leq \omega(E_1)\).

The subgraph \(G(Q \setminus E_1)\) is Eulerian: it is connected since \(R \cup T \subseteq Q \setminus E_1\) and it is balanced since the imbalanced vertices of \(G(E_1)\) are exactly those of \(G(Q)\), that is, \(S\). Its Euler cycle can be shortcut to a simple cycle on \(S\), which can be partitioned into two perfect matchings \(M_2\) and \(M_3\) on \(S\). Thus,

\[ \omega(P) = \omega(R \cup T) + \omega(M) \leq \omega(P^*) + \omega(M_2) + \omega(M_3)/3 \leq \omega(P^*) + \omega(Q)/3 \leq 5/3 \cdot \omega(P^*), \]

where the second inequality is due to the metric weights \(\omega\). 

Plugging Theorem 5.1 into Lemma 4.5, we immediately get:

**Corollary 5.2.** HCPP(I) is 5/3-approximable in \(O(kn^2)\) time.

6. Parameterized algorithms for HCPP(I)

Lemma 4.5 allows us to easily transfer well-known parameterized algorithms from RPP to HCPP(I) to show:

**Theorem 6.1.** Let \(\omega_{\max}\) be the maximum edge weight and \(c\) be the maximum number of connected components in any edge class of an HCPP(I) instance. Then, HCPP(I) is

i) polynomial-time solvable for constant \(c\) and
ii) solvable in \(2^c \cdot \text{poly}(\omega_{\max}, n)\) time with polynomially bounded error probability.

**Proof.** To prove the theorem, it is enough to show that the known RPP algorithms can also be used for \(s \times t\)-RPP. To this end, we reduce \(s \times t\)-RPP to RPP. We assume that \(s \neq t\) and that \(s\) and \(t\) are non-adjacent in \(s \times t\)-RPP instances (otherwise, we can add a new source \(s'\) and a required weight-zero edge \((s', s)\)).

Now, note that an \(s \times t\)-RPP instance \(I := (G, R, \omega, s, t)\) can be reduced to an RPP instance \(I' := (G', R', \omega')\) where \(I'\) is obtained from \(I\) by adding an edge \((s, t)\) of weight \(2\omega(E)\) to both \(E\) and \(R\). Then, an optimal solution \(P\) for \(I\) yields a solution of weight \(\omega(P) + 2\omega(E)\) for \(I'\). Moreover, an optimal solution \(P\) for \(I'\) uses the edge \((s, t)\) exactly once: if \(P\) traversed it multiple times, then it would be cheaper to replace the second traversal of \((s, t)\) by any other \(s \times t\)-path in \(G\). Thus, \(P\) can be turned into a solution of weight \(\omega(P) - 2\omega(E)\) for \(I\). That is, optimal solutions translate between \(I\) and \(I'\) (yet approximate solutions do not).

Moreover, if the number of connected components in \(G(R)\) is \(c'\), then the number of connected components in \(G'(R')\) is at most \(c' + 1\). Thus, since RPP is solvable in polynomial time for constant \(c'\) [4, 13], so is \(s \times t\)-RPP. And since RPP is solvable in \(2^c \cdot \text{poly}(\omega_{\max}, n)\) with polynomially bounded error probability [16], so is \(s \times t\)-RPP. To conclude the proof of the theorem, it is enough to apply Lemma 4.5 and to observe that, for any \(s \times t\)-RPP instance \(P(u, v, i)\) solved, the subgraph \(G(E_i)\) induced by the required edges \(E_i\) has at most \(c\) connected components. 

7. Conclusion

We have shown that HCPP(c) is NP-hard even on orders that decompose into a linear order and an incomparable class. Unless P = NP, this rules out polynomial-time algorithms for HCPP(c) on many order types (tree orders, interval orders, bounded-width orders), yet leaves unproven the conjecture that HCPP(c) is polynomial-time solvable for a constant number of edge classes [4].

We have shown a 5/3-approximation algorithm for HCPP(l) by presenting a simple 5/3-approximation algorithm for s-t-RPP. Any better approximation factor for s-t-RPP immediately carries over to HCPP(l) using Lemma 4.5. Indeed, sure better approximation factors are possible, given that a 3/2-approximation is known for RPP. Analogous approximation algorithms for metric s-t-TSP have gone a long way since the 5/3-approximation due to Goemans [18]: recently, a 2/3-approximation for metric s-t-TSP has been shown, matching the approximation factor of the Christofides-Serdyukov algorithm for metric TSP [8, 30]. It is not obvious whether the used approaches carry over to s-t-RPP.

Finally, transferring fixed-parameter algorithms from RPP to s-t-RPP, we have shown fixed-parameter algorithms for HCPP(l) parameterized by the maximum number c of connected components in any edge class. Thus, it is interesting whether lossy kernelization results for RPP [2] carry over to HCPP(l), and if so, with which performance guarantees.

**Funding.** This work is funded by Mathematical Center in Akademgorodok, agreement No. 075-19-2019-1675 with the Ministry of Science and Higher Education of the Russian Federation.

**References**

[1] R. van Bevern, R. Bredereck, L. Bulitko, C. Komusiewicz, N. Talmon, G. Woeginger, Precedence-constrained scheduling problems parameterized by partial order width, in: Y. Kochetov, M. Khachay, V. Beresnev, E. Nurminski, P. Pardalos (Eds.), DOOR 2016, Springer, 2016, volume 9869 of Lecture Notes in Computer Science, pp. 105–120, doi:10.1007/978-3-319-44914-2_9.

[2] R. van Bevern, T. Fluschnik, O. Yu. Tsidulko, On approximate data reduction for the Rural Postman Problem: Theory and experiments, Networks 76 (2020) 485–508, doi:10.1002/net.21985.

[3] R. van Bevern, S. Hartung, A. Nichterlein, M. Sorge, Constant-factor approximations for capacitated arc routing without triangle inequality, Operations Research Letters 42 (2014) 290–292, doi:10.1016/j.orl.2014.05.002.

[4] R. van Bevern, R. Niedermeier, M. Sorge, M. Weller, Complexity of arc routing problems, in: Á. Corberán, G. Laporte (Eds.), Arc Routing, SIAM, volume 20 of MOS-SIAM Series on Optimization, chapter 2, 2015, pp. 19–52, doi:10.1137/9781611973679.ch2.

[5] R. van Bevern, V. A. Sluga, A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem, Historia Mathematica (in press), doi:10.1016/j.hm.2020.04.003.

[6] E. A. Cabral, M. Gendreau, G. Ghiani, G. Laporte, Solving the hierarchical Chinese postman problem as a rural postman problem, European Journal of Operational Research 155 (2004) 44–50, doi:10.1016/s0377-2217(02)00813-5.

[7] N. Christofides, The optimum traversal of a graph, Omega 1 (1973) 719–732, doi:10.1016/0305-0488(73)90089-3.

[8] N. Christofides, Worst-case analysis of a new heuristic for the traveling salesman problem, Management Science Research Report 388, Carnegie-Mellon University, Pittsburgh, Pennsylvania, USA, 1976.

[9] M. Dror, H. Stern, P. Trudeau, Postman tour on a graph with precedence relationship on arcs, Networks 17 (1987) 283–294, doi:10.1002/net.3230170404.

[10] J. Edmonds, E. L. Johnson, Matching, Euler tours and the Chinese postman, Mathematical Programming 5 (1973) 88–124, doi:10.1007/BF01580113.

[11] H. A. Eiselt, M. Gendreau, G. Laporte, Arc routing problems, part II: The Rural Postman Problem, Operations Research 43 (1995) 399–414, doi:10.1287/opre.43.3.399.

[12] H. Fleischner, Eulerian Graphs and Related Topics: Part I, Volume 2, North-Holland, Amsterdam, The Netherlands, volume 50 of Annals of Discrete Mathematics, chapter X, 1991, pp. X.1–X.14.

[13] G. N. Frederickson, Approximation Algorithms for NP-hard Routing Problems, Ph.D. thesis, University of Maryland Graduate School, College Park, Maryland, USA, 1977.

[14] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, USA, 1979.

[15] G. Ghiani, G. Improta, An algorithm for the hierarchical Chinese postman problem, Operations Research Letters 26 (2000) 27–32, doi:10.1016/S0167-6377(99)00046-2.

[16] G. Gutin, M. Wahlström, A. Yeo, Rural Postman Parameterized by the number of components of required edges, Journal of Computer and System Sciences 83 (2017) 121–131, doi:10.1016/j.jcss.2016.06.001.

[17] C. Hierholzer, C. Wiener, Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren, Mathematische Annalen 6 (1873) 30–32, doi:10.1007/BF01442866.

[18] J. Hoogeveen, Analysis of Christofides’ heuristic: Some paths are more difficult than cycles, Operations Research Letters 10 (1991) 291–295, doi:10.1016/0167-6377(91)90016-I.

[19] T. C. Hu, Parallel sequencing and assembly line problems, Operations Research 9 (1961) 841–848, doi:10.1287/opre.9.6.841.

[20] R. Impagliazzo, R. Paturi, F. Zane, Which problems have strongly exponential complexity?, Journal of Computer and System Sciences 63 (2001) 512–530, doi:10.1006/jcss.2001.1774.

[21] R. M. Karp, Reducibility among combinatorial problems, in: R. E. Miller, J. W. Thatcher, I. D. Bohlinger (Eds.), Complexity of Computer Computations, The IBM Research Symposia Series, Springer, 1972, pp. 85–103, doi:10.1007/978-1-4684-2001-2_9.

[22] P. Korteweg, T. Volgenant, On the hierarchical Chinese postman problem with linear ordered classes, European Journal of Operational Research 169 (2006) 41–52, doi:10.1016/j.ejor.2004.06.003.

[23] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, Proceedings of the American Mathematical Society 7 (1956) 48–48, doi:10.1090/s0002-9939-1956-0078686-7.

[24] G. Laporte, The un-directed Chinese postman problem, in: Á. Corberán, G. Laporte (Eds.), Arc Routing, SIAM, volume 20 of MOS-SIAM Series on Optimization, chapter 2, 2015, pp. 53–64, doi:10.1137/9781611973679.ch3.

[25] E. Lawler, Combinatorial Optimization—Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.

[26] J. K. Lenstra, A. H. G. Rinnooy Kan, On general routing problems, Networks 6 (1976) 273–280, doi:10.1002/net.3230060305.

[27] O. Michail, P. G. Spirakis, Traveling salesman problems in temporal graphs, Theoretical Computer Science 634 (2016) 1–23, doi:10.1016/j.tcs.2016.04.006.

[28] C. Papadimitriou, M. Yannakakis, Scheduling interval-ordered tasks, SIAM Journal on Computing 8 (1979) 405–409, doi:10.1137/0208031.

[29] A. I. Serdyukov, O zadache nakhodzheniya minimal’nogo Eulerova mul’tigrafa dlya svarznogo grafa so svyazhennymi rebrami, Upravlyaemye sistemy 12 (1974) 61–67, URL http://nas1.math.nsc.ru/aim/journals/uzs/uzs12_008.pdf.

[30] A. I. Serdyukov, O nekotorykh ekstremal’nykh obkhodakh v grafakh, Upravlyaemye sistemy 17 (1978) 76–79, URL http://nas1.math.nsc.ru/aim/journals/uzs/uzs17_007.pdf.

[31] V. V. Sverkh, Efektivno razreshimyi sluchai zadachi kalendarnogo planirovaniya s vozobnovyvymi resursami, Diskretnyj Analiz i Issledovaniye Operacij 7 (2000) 75–82, URL http://mi.mathnet.ru/dz294.