The Chinese deliveryman problem

Martijn van Ee¹ · René Sitters²

Received: 1 April 2019 / Revised: 7 October 2019 / Published online: 19 October 2019
© The Author(s) 2019

Abstract

We introduce the Chinese deliveryman problem where the goal of the deliveryman is to visit every house in his neighborhood such that the average time of arrival is minimized. We show that, in contrast to the well-known Chinese postman problem, the Chinese deliveryman problem is APX-hard in general and NP-hard for planar graphs. We give a simple $\sqrt{2}$ -approximation for undirected graphs and a $4/3$ -approximation for 2-edge connected graphs. We observe that there is a PTAS for planar graphs and that depth first search is optimal for trees.

Keywords Chinese deliveryman problem · Approximation algorithms · Computational complexity · Chinese postman problem · Graph search

Mathematics Subject Classification 90C27 · 68Q17 · 90C59

1 Introduction

The famous Chinese Postman Problem was first discussed by the Chinese mathematician (Kwan 1960).

Chinese Postman Problem (CPP):¹

A postman has to deliver letters to a given neighborhood. He needs to walk through all the streets in the neighborhood and back to the post-office. How can he design his route so that he walks the shortest distance?

¹ Citation from Grötschel and Yuan (2012).

✉ René Sitters
r.a.sitters@vu.nl
Martijn van Ee
m.v.ee.01@mindef.nl

¹ Netherlands Defence Academy, Den Helder, The Netherlands

² Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

The objective of the postman is to minimize the length of his tour. A more client oriented objective would be to minimize the average time of delivery. We call this the Chinese Deliveryman Problem (CDP).

CDP:

A postman has to deliver letters to a given neighborhood. He needs to visit all the houses in the neighborhood starting from the post office. If the houses are uniformly distributed over the streets then how can he design his route so that average time of delivery is minimal?

An equivalent definition of the CDP is as a search problem on the edges of a graph. One needs to find an object that is hidden at a random location X somewhere in a street, where the position X is uniformly distributed over the total road network. The goal is to find a path that minimizes the expected time of finding the object. Hence, as a deliveryman problem we assume that a street e of length $l(e)$ has a total of $l(e)$ houses which are uniformly and continuously distributed over the street. Graph search problems have been studied extensively but not in exactly this form. The literature on this is vast and we refer to Bonato and Yang (2013) for an extensive review.

Formally, an instance of the CDP is given by a graph $G = (V, E)$ with an integer length $l(e)$ for every edge $e \in E$ and a root vertex $v_0 \in V$. A feasible solution is a path that starts in v_0 and goes through all edges. An edge e is considered a line segment of length $l(e)$ and the completion time of a point p on a line is the length of the path up to the first moment where it meets p . The goal is to minimize the average completion time of the (infinite) set of points. Seen as a search problem, the goal is to minimize the expected time until reaching an unknown random location X which is uniformly distributed over the line segments. (In Sect. 2 we shall briefly discuss other distributions.) We do allow the graph to have loops and multiple edges since it is easy to find an equivalent formulation without loops and multiple edges just by subdividing edges.

In the *directed* CDP, an instance is given by a directed graph $G = (V, A)$ with an integer length $l(e)$ for every arc $a \in A$. We assume that the graph is strongly connected, i.e., every vertex i can be reached from any vertex j through a directed path. We shall only briefly discuss the directed CDP. From now we restrict to the undirected version unless stated otherwise.

In principle, we do allow that the deliveryman visits part of a line and then turns around and visits the remaining part later on his path. However, such a path can never be optimal as we show in Lemma 1. Hence, we shall restrict to solutions that are postman paths and define the completion time $C(e)$ of an edge e as the length of the path up to the point where the complete edge has been traversed minus half its length. The *average completion time* of a solution now becomes

$$\frac{\sum_{e \in E} l(e)C(e)}{\sum_{e \in E} l(e)}.$$

We refer to the numerator of this expression as the *total completion time*.

Another famous optimization problem is the Traveling Salesman Problem (TSP) in which one needs to find a tour of minimum length visiting all the vertices of an edge-weighted graph at least once. Also well-known is the Deliveryman Problem in which

Table 1 Elementary routing problems

Objective	Tour length	Average arrival time
Traversing all vertices	TSP	TRP
Traversing all edges	CPP	CDP

we have to find a path visiting all vertices so as to minimize the average completion time of the vertices, which is also known as the Traveling Repairman Problem (TRP) or the Minimum Latency Problem. The CDP fits nicely in this list as shown in Table 1. Yet, to the best of our knowledge, no research has been done on precisely this problem.

Edmonds (1965) showed that the CDP can be solved efficiently by reducing it to a weighted matching problem. If all edges in the graph are directed, then the problem is also efficiently solvable, as shown by Edmonds and Johnson (1973). But if the graph contains both directed and undirected edges (mixed CPP) then the problem is NP-hard as shown by Papadimitriou (1976). In contrast, we show that the CDP is already NP-hard for undirected planar graphs and even APX-hard for general graphs. For edge-weighted trees however, the CDP is polynomially solvable.

In Sect. 4, we present a $\sqrt{2}$ -approximation for the undirected CDP in general graphs, a $4/3$ -approximation for 2-edge connected graphs, and a PTAS for planar graphs.

1.1 Relation to other minimum latency problems

In this section we show how the approximability of the CDP compares to the approximability of some other minimum latency problems. We only sketch some easy reductions. Sections 3 and 4 give results for the CDP in detail.

We say that an algorithm A is an α -approximation algorithm for a minimization problem if for any instance of the problem the value of the computed solution is no more than α times the optimal value. We say that a family of algorithms A_ϵ is an α -approximation scheme if for any constant $\epsilon > 0$, algorithm A_ϵ is a $(1 + \epsilon)\alpha$ -approximation algorithm. Here we shall restrict to schemes that run in polynomial time, i.e., A_ϵ runs in polynomial time for all fixed $\epsilon > 0$. When $\alpha = 1$, this is known as a polynomial time approximation scheme (PTAS).

The CDP can be turned into a TRP by replacing each edge with a (polynomially) large number of uniformly distributed points. Any α -approximation scheme for TRP implies an α -approximation scheme for the CDP (for any class of graphs that is closed under the division of edges). The best polynomial time approximation ratio for TRP is currently 3.59, given by Chaudhuri et al. (2003). Note that in this reduction, the TRP instance is on an unweighted graph. Koutsoupias et al. (1996) gave an approximation for TRP on unweighted graphs with ratio strictly less than 1.662. Hence, this implies a 1.662-approximation for CDP. In general, CDP is much easier to approximate than TRP as we show in this paper.

Van Omme (2011) introduced another minimum latency version of CPP and called it the *Cumulative Chinese Postman Problem* (CCPP). The latency is defined as the time at which the edge has been traversed completely and the total latency is the unweighted sum of the edge-latencies. That means, the latency is not weighted by the length of the edge. On tree metrics, CCPP and TRP are actually equivalent. Hence, like TRP, the CCPP is NP-hard for edge-weighted trees (Sitters 2002). In contrast, we show in Sect. 3 that the CDP is easy for tree metrics. In his PhD thesis, Van Omme compared many exact ILP formulations for this problem but did not consider approximation algorithms. It is easy to see that any α -approximation scheme for the CCPP on general graphs implies an α -approximation scheme for metric TRP. Hence, CCPP is at least as hard to approximate as metric TRP. We sketch a proof here: Given an instance of TRP we can model it as a CCPP as follows. We make a complete graph with edge lengths equal to the given distances and to each vertex we add a large number M of edges of length zero. For large enough M , the total completion time of those zero-length edges dominates the objective function (since edges are not weighted by their length). Hence, any α -approximation scheme for the CCPP on general graphs implies an α -approximation scheme for metric TRP.

Comparing CDP and CCPP we note that the CDP on general *weighted* graphs is as hard to approximate as the CCPP on general *unweighted* graphs. A sketch of the proof is follows. Given an unweighted CCPP instance, we use that as an instance of CDP. For both instances, $|E|^2/2$ is a lower bound on the optimal values. The difference in the total completion time of the two optimal values is exactly $|E|/2$. Hence, as $|E|$ increases, the ratio of the two optimal values approaches 1. The argument for the other direction is similar. Given an instance of CDP we subdivide an edge of length w into w edges of length 1 and take that as an instance of CCPP. For both instances, the total completion time is at least $L^2/2$, where L is the total edge length. The difference in the two optimal total completion times is no more than $L/2$. Hence, the ratio of the two optimal values approaches 1 when L increases.

Summarizing: any α -approximation scheme for the unweighted Traveling Repairman problem implies an α -approximation scheme for the CDP. Further, the Cumulative Chinese Postman Problem (CCPP) considered by Van Omme is at least as hard to approximate as the metric Traveling Repairman Problem. Further, any α -approximation scheme for unweighted CCPP implies an α -approximation scheme for CDP and vice versa.

2 No-turn property

Intuitively, an optimal solution to an instance of the CDP does not turn around on an edge since in our model the weight of an edge is uniformly distributed over the edge. A consequence of this property, that we prove in Lemma 1, is that we may restrict to solutions that are postman paths. Clearly, this is not true for any distribution of the weight over the edge but does hold for a uniform distribution.

In the *uniformly distributed* CDP, every edge e is given a length $l(e)$ and a weight $w(e)$. A feasible solution is a path that starts in the root and goes through all edges and the completion time of a point p on an edge is the length of the path up to the first

point where it meets p . Now, the completion time of any point on edge e is weighted by the factor $w(e)/l(e)$ and the goal is to minimize the average weighted completion time of the (infinite) set of points. As a search problem: location X is uniform on every edge and the probability that X is on edge e is $w(e)/w(E)$. In the standard CDP, $w(e) = l(e)$ for all edges.

It is easy to see that in general, *uniformly distributed* CDP is as hard to approximate as the TRP (since TRP can be reduced to uniformly distributed CDP by putting no weight on the edges and adding a zero-length edge of weight one to every vertex). Hence, apart from the next lemma, we shall restrict to the standard CDP in this paper, where the weight of an edge is equal to its length.

Lemma 1 *An optimal solution for the uniformly distributed CDP never turns on an edge or loop.*

Proof Consider an optimal solution for an instance of the uniformly distributed CDP. Lets say that a *server* follows the optimal path. Assume that the server turns around on an edge $e = (v, w)$. (For a loop we have $v = w$.) Say that it moves from v to point p on edge (v, w) that it reaches at time t_1 and then moves back to v . Let t_2 be the first moment after t_1 at which the server returns in p . Further, assume that t_1 is the first time the server turns in the solution. If (v, w) has been traversed before then no new points were visited while moving from v to p and back to v and the path is clearly not optimal. Hence we may assume that none of the points on (v, w) were visited before time $t_1 - d(v, p)$. We consider three cases based on the way the server visits p for the second time. In each case we consider two modified paths and argue that one of the two has a strictly smaller total completion time than the original one.

Case 1 The server reaches p through v .

Case 2 The server reaches p through w and continues to v .

Case 3 The server reaches p through w and moves back to w .

Since the solution is optimal we know that it does not turn at time t_2 in case 1 and 2, but does turn at time t_2 in case 3. Let W be the total weight that is served after t_1 and let W' be the total weight that is served between time t_1 and t_2 . Let $\alpha = w(e)/l(e)$.

Case 1 Let the server turn at time $t_1 - \epsilon$ in the point p^- at distance ϵ from p . The weight of the points between p and p^- is $\alpha\epsilon$ and the average delay for these points is $t_2 - t_1 - 2\epsilon$. Further, the points that were visited after t_1 are completed 2ϵ earlier. The total increase is $\epsilon\alpha(t_2 - t_1 - 2\epsilon) - 2\epsilon W$. If, on the other hand, we modify the path in the other direction, i.e., the server turn ϵ time later in a point p^+ , then the total increase is $-\epsilon\alpha(t_2 - t_1) + 2\epsilon(W - \epsilon\alpha)$. The sum is $-4\epsilon^2\alpha < 0$. Hence, one of the two modified paths is strictly better than the original one.

Case 2 The server reaches p through w , i.e., it is in w at $t_2 - d(w, p)$, and continues to v . Now let the server turn at time $t_1 - \epsilon$ in the point p^- at distance ϵ from p . The completion time of any point that was served after time t_1 is reduced by 2ϵ . On the other hand, the completion time of the points between p^- and p is on average increased by $t_2 - t_1 - \epsilon$. The total increase is $\epsilon\alpha(t_2 - t_1 - \epsilon) - 2\epsilon W$. If we modify the path in the other direction then the total increase is $-\epsilon\alpha(t_2 - t_1 - \epsilon) + 2\epsilon(W - \epsilon\alpha)$. The sum is $-2\epsilon^2\alpha < 0$. Hence, the original path is not optimal.

Fig. 1 All self-loops have the same length, for example length 1. An optimal solution will traverse the loops in the order C_1, C_2, C_3, C_4 and the zero-length edge will be traversed 4 times. Clearly, the example works with any number of loops and the zero can be replaced by sufficiently small $\epsilon > 0$

Case 3 The server reaches p through w , i.e., it is in w at $t_2 - d(w, p)$, and moves back to w . The argument is similar as before but now notice that when we turn earlier then, when the server is back in p at time $t_2 - 2\epsilon$, it moves to p^- and back to p to serve the yet unserved points between p^- and p . Similarly, if we let the server turn later then it doesn't have to return to p but can turn at point p^+ . In both cases there is no increase in completion time for points that were served after t_2 in the original solution. If the server turns earlier, then the total increase is $\epsilon\alpha(t_2 - t_1 - \epsilon) - 2\epsilon W'$. If we modify the path in the other direction then the total increase is $-\epsilon\alpha(t_2 - t_1 - \epsilon) + 2\epsilon(W' - \epsilon\alpha)$. The sum is $-2\epsilon^2\alpha < 0$. Hence, the original path is not optimal. $\square$

In the graph search literature, the question for which probability distributions it is optimal not to turn on an edge has been considered for line and star graphs. For line graphs, Beck and Beck (1984) showed that it is optimal not to turn for the uniform distribution. Later, Kella (1993) gave a sufficient condition for optimality of not turning on star graphs. Van Ee (2017) gave a characterization of symmetric distributions on star graphs for which it is optimal not to turn. There is also research on distributions for which turning on an edge is optimal. In particular, it was shown by Baston and Beck (1995) that for the triangular distribution on a line graph (with its peak at the origin) it is optimal to turn infinitely many times.

Another property of optimal solutions that we would like to mention is that, unlike optimal postman solutions, an optimal solution for the CDP may cross an edge an arbitrary number of times as follows from the example of Fig. 1.

3 Complexity

The CDP has the solution space of the Chinese Postman Problem (each edge needs to be traversed) but it has the objective of the Traveling Repairman problem. Intuitively, the complexity of the CDP lies somewhere between that of the CPP and the TRP. Clearly, when the graph has an Euler path starting in the root then this path is optimal for the CDP. We will show that like TRP, the CDP is NP-hard for planar graphs and is APX-hard in general. Unlike the TRP, it is easy for edge-weighted trees.

The TRP on unweighted trees is solved optimally by following a depth-first search path (Blum et al. 1994; Minieka 1989). An immediate corollary is that depth-first is optimal for the CDP on edge-weighted trees. This is easy to see when all distances are integer since we can replace an edge of length l by a path of l edges of length 1 without changing the value of any CDP solution. Here, we give a simple self-contained

proof similar to the proof in Blum et al. (1994) which also holds for non-integer values.

Lemma 2 *Depth-first search is optimal for the Chinese Deliveryman Problem on trees.*

Proof We use the search formulation of the problem. Let L be the total length of the tree and let p_z be the point on the path where a total length of z has been explored and let $C(p_z)$ be that time. The expected time until finding the target is exactly

$$\frac{1}{L} \int_{z=0}^L C(p_z).$$

For any point p on an edge let $\text{depth}(p)$ be its distance to the root. Note that for any solution we have $C(p_z) \geq 2z - \text{depth}(p_z)$ and equality holds for a depth-first search. Hence, for any solution the expected time until finding the target is at least

$$\frac{1}{L} \int_{z=0}^L 2z - \text{depth}(p_z).$$

and equality holds in case of a depth-first search. This integral is independent of the solution. Hence, depth-first search is optimal. $\square$

The next theorem shows that CDP is NP-hard for planar graphs. The proof follows from NP-completeness of the Hamiltonian path problem on planar graphs (Garey et al. 1976).

Theorem 1 *The undirected and directed CDPs are both NP-hard on planar graphs.*

Proof First, we consider the undirected problem and then show how to modify the proof for the directed case. We reduce from the Hamiltonian path problem: Given a planar graph $G = (V, E)$ and point $v_0 \in V$, the question is whether there exists a path that starts in v_0 and visits all vertices exactly once. Given this instance of the Hamiltonian path problem we define an instance of the CDP as follows (Fig. 2).

Fig. 2 The reduction from planar HC to planar CDP

For each edge (v_i, v_j) we define two points a_{ij} and b_{ij} and replace the edge by four edges (v_i, a_{ij}) , (a_{ij}, v_j) , (v_i, b_{ij}) , and (b_{ij}, v_j) , each of length one. To each of the new points a_{ij} and b_{ij} we add a loop of length $C > \frac{3}{2}(n^2 - n)$. Denote the resulting graph by G' . Note that G' is Eulerian. Next, we add for each point v_i of the original graph G a point v'_i and an edge $f_i = (v_i, v'_i)$ of length 1. Denote these edges by F . Let H be the resulting graph and let L be the total length of the edges, i.e., $L = 4m + 2mC + n$. Note that H is a planar graph.

Assume that G has a Hamiltonian path that starts in v_0 . Consider the following tour T_{HP} for the deliveryman instance. Starting in v_0 we follow an Euler tour in G' and next we use the Hamiltonian path from G to visit all edges (v_i, v'_i) by a path of minimum length. The length of the Euler tour is $L - n$ and the total completion time of the edges on this Euler tour is $K_1 = (L - n)^2/2$. The total completion time of the edges (v_i, v'_i) is $(L - n + \frac{1}{2}) + (L - n + 4 + \frac{1}{2}) + \dots + (L - n + 4(n - 1) + \frac{1}{2}) = n(L - n) + 2n^2 - \frac{3}{2}n$. Denote this value by K_2 . The total completion time of all edges is $K_1 + K_2$.

Now assume that there is a solution to the CDP instance of total completion time at most $K_1 + K_2$. We prove that this implies that G has a Hamiltonian path starting from v_0 . We use that the solution does not turn on an edge or loop (Lemma 1). First, we argue that no edge (v_i, v'_i) is traversed before all loops have been traversed. A lower bound on the total completion time is $L^2/2$. Note that equality only holds if H has an Euler path, which it does not have unless the original graph is a single vertex.

If an edge (v_i, v'_i) is traversed before all loops have been traversed then at least one loop is delayed by the length of edge (v_i, v'_i) . Hence, the total completion time will be at least

$$L^2/2 + C > L^2/2 + \frac{3}{2}(n^2 - n) = (L - n)^2/2 + n(L - n) + 2n^2 - \frac{3}{2}n = K_1 + K_2.$$

Hence, loops are traversed before edges (v_i, v'_i) . But then, the total completion time of all edges (v_i, v'_i) must be at least as large as in the tour T_{HP} , i.e., at least K_2 . Further, the total completion time of all edges in G' is exactly K_1 if T starts with an Euler tour on G' and is strictly larger otherwise. Hence, the total completion time of T is exactly $K_1 + K_2$ if T follows an Euler tour in G' followed by a path in H that corresponds to a Hamiltonian path in G and is strictly larger than $K_1 + K_2$ otherwise.

The reduction for the directed case is similar. We can direct the arcs in G' such that G' is a directed Eulerian graph. Further, instead of one arc (v_i, v'_i) we now have two copies of this arc and one arc (v'_i, v_i) . Assume there is a Hamiltonian path in G . We start with an Euler tour that only leaves one arc (v_i, v'_i) unvisited for all i . Next we take the Hamiltonian path to visit these remaining arcs. Let K be the total completion time. Any solution that is not composed of an Euler tour plus Hamiltonian path will have a total completion time strictly larger than K . $\square$

For general graphs, the CDP turns out to be APX-hard. We reduce from (1, 2)-TSP, which was shown to be Max SNP-hard by Papadimitriou and Yannakakis (1993). It follows from the PCP-theorem (Arora et al. 1998a; Arora and Safra 1998) that Max SNP-hard problems do not have polynomial time approximation schemes unless,

Fig. 3 The APX-hardness reduction

$P = NP$. Hence, there is some small constant δ such that approximating the $(1, 2)$ -TSP within a factor $1 + \delta$ is NP-hard.

Theorem 2 *The CDP is APX-hard for general (un)directed graphs.*

Proof We first prove it for undirected graphs. The adjustment for directed graphs is easy. An instance of the $(1, 2)$ -TSP is given by an unweighted graph $G = (V, E)$. The distance between $i \in V$ and $j \in V$ is 1 when $(i, j) \in E$ and 2 otherwise. Papadimitriou and Yannakakis (1993) showed that for $(1, 2)$ -TSP there is a $\delta > 0$ such that it is NP-complete to decide if the optimal tour has length n or if the optimal tour has length at least $(1 + \delta)n$. This even holds when restricted to graphs of maximum degree at most 6. We shall use the following *gap-property* in our reduction:

Let $w_0, w_1, \dots, w_{n-1}$ be a permutation of the vertices where w_0 is the root and define $w_n = w_0$. If G is not Hamiltonian, then the distance in G between w_i and w_{i+1} is at least 2 for at least δn values $i \in \{0, \dots, n - 1\}$.

Given a graph $G = (V, E)$ of maximum degree at most 6, we define the following instance of the CDP. Double all edges and denote by E' the set of copies. The graph $G' = (V, E \cup E')$ is Eulerian. Next, for each vertex v_i add a vertex v'_i and an edge $f_i = (v_i, v'_i)$. Denote these edges by F . Take an arbitrary vertex from V as the origin r . All edges have length 1. The created instance is illustrated in Fig. 3. We assume $|E| \geq n$ since otherwise the graph is obviously not Hamiltonian.

Consider an optimal solution σ^* for the CDP instance. Let C^* be the optimal value. We will w.l.o.g. assume that it ends in a vertex $v_i \in V$. Consequently, each edge f_i is visited *exactly once* in each direction. Denote $|E| = m$ and let t^* be the time at which exactly $2m$ times an edge of $E \cup E'$ has been visited, including multiple visits to the same edge. Let a be the number of edges from F visited after time t^* . Hence, $n - a$ edges of F are visited before time t^* and we have $t^* = 2m + 2(n - a)$. Summarizing:

- (i) In the first t^* positions in the sequence of edges there is exactly $2m$ times an edge from $E \cup E'$.
- (ii) Each edge from F appears twice, one after the other.

(iii) In the first t^* positions there are exactly $n - a$ edges from F .

Assume that G has a Hamiltonian cycle T visiting the vertices in the order $v_0, v_1, \dots, v_n = v_0$. We shall define a solution σ_a with properties (i), (ii), (iii) and for which (i) is strengthened to

(i') In the first t^* positions, each edge from $E \cup E'$ appears exactly once.

Note that when we remove the edges of T from $G' = (V, E \cup E')$ then the new graph is still connected and Eulerian. Now, follow an Euler tour on this graph followed by tour T . For all $i \in \{a, a + 1, \dots, n - 1\}$, visit vertex v'_i after the visit of vertex v_i on tour T . Next, follow T again up to vertex v_{a-1} and visit vertex v'_i after v_i for all $i \in \{0, 1, \dots, a - 1\}$. Denote this solution by σ_a and let C_a be its total completion time. (One can show that the total completion time for σ_a is minimized for $a \approx n/2$ but we do not prove nor use that here.) If a HC exists then σ_a is well-defined. Note that C_a is a function of $|E|$, n , and a only and $C_a = O(n^2)$.

$$C^* \leq C_a = O(n^2).$$

We will show that if no HC exists then

$$C^* = C_a + \Omega(n^2). \quad (1)$$

If no HC exists then σ_a is not well defined. Instead we now define σ_a as a sequence of edges (not necessarily corresponding to a walk in the graph) with the properties (i'), (ii), (iii) and the following pattern, that we denote as property (iv).

(iv) Pattern: $e, e, \dots, e, f, f, e, f, f, \dots, e, f, f,$

where e stands for an edge from $E \cup E'$ and f for an edge in F . The completion time of an edge is defined as its first position in the sequence (minus 0.5). Then the total completion is exactly C_a as defined.

We will reorder the edges in the sequence σ^* such that the total completion time decreases by $\Omega(n^2)$ and such that the new total completion is equal to C_a . (The new sequence may not be a walk in the graph though.) In the pattern of σ_a there is one edge e between every pair f, f . In the pattern of σ^* we also have exactly n times f, f but if no HC exists in G then at least δn times there are at least two e 's between two consecutive pairs f, f . This follows directly from the gap-property defined above.

Let I_E be the set of positions in sequence σ^* with an edge from $E \cup E'$ and let I_F be the positions with an edge from F . Note that $|I_E \cap \{1, 2, \dots, t^*\}| = 2m$. Now, reorder the edges at positions I_E such that all edges in $E \cup E'$ appear exactly once in the positions $I_E \cap \{1, 2, \dots, t^*\}$. This will not increase the total completion time of the sequence. Let σ^{**} be the new sequence. Next, we will define a sequence of edge swaps that decrease the total completion time by $\Omega(n^2)$. We distinguish between the part before t^* and the part after t^* . If before time t^* , the edges f, f, e_1, e_2 (with $f \in F$ and $e_1, e_2 \in E \cup E'$) appear consecutively in σ^{**} then changing the order to e_1, f, f, e_2 will reduce the total completion time of the sequence by exactly 1 (since the completion time of e_1 decreases by 2 and the completion time of f increases by 1. After time t^* we can do something similar. Note that all edges from $E \cup E'$ have been

completed by time t^* . Hence, if e_1, e_2, f, f (with $e_1, e_2 \in E \cup E'$ and $f \in F$) appear consecutively in the sequence after time t^* then changing the order to e_1, f, f, e_2 will reduce the total completion time of the sequence by exactly 1 (since the completion time of f decreases by 1 and the completion time of e_1 and e_2 is unchanged since these were already visited before time t^*). If no HC exists then (using the gap-property described above) we can make $\Omega(n^2)$ swaps before ending in a sequence that has the same pattern and total completion time as σ_a . Hence, (1) follows.

For directed graphs we orient the edges in E and E' in an opposite way. Instead of one edge (v_i, v'_i) we now take two arcs (v_i, v'_i) and one arc (v'_i, v_i) . The main difference is that the directed Euler tour now also contains one of the two arcs (v_i, v'_i) and the arc (v'_i, v_i) . The rest of the proof is the same. $\square$

4 Approximation algorithms

Since the problem is APX-hard for general graphs there is no hope for a PTAS but we show here that simply following a postman tour or postman path already gives a good approximation ratio.

Algorithm 1 Follow an optimal postman tour.

Algorithm 2 Compute an optimal postman tour and traverse it in the best direction.

Algorithm 3 Follow an optimal postman path.

4.1 General graphs

Figure 4 shows that Algorithm 1 is not better than a 1.5-approximation in general. The circle and the edge both have length 1. Traversing the edge first gives a total completion time of 3. Traversing the circle first gives a total completion time of 2. Algorithm 1 might take either if these solutions.

Figure 5 shows that Algorithm 2 is not better than a $13/9 \approx 1.444$ -approximation. The instance has $3n$ edges length 1 and one edge of length $3n/2$. It has an Euler path starting from the origin and its average completion time is $9n/4$. The length of the optimal postman tour is $7n$ and one solution is to follow the pattern $a_1, a_2, b_2, b_2, a_3, a_4, b_4, b_4, \dots$, followed by the long edge and a similar pattern back. For n even, the length of this tour is $7n$. The $3n$ edges of length 1 have average completion time $7n/2 + O(1)$. The long edge has average completion time $2n + 3n/4$. The weighted average is $(3n(7n/2) + (3n/2)(2n + 3n/4)) / 4.5n + O(1) = 13n/4 + O(1)$.

Figure 6 shows that Algorithm 3 is not better than a $\sqrt{2}$ -approximation. In the example, it is optimal to traverse the cycle first. The total completion time is $2k^2 + \sqrt{2}k^2 + O(k)$. The algorithm will double all leaves except for one. If the $k - 1$ doubled leaves are traversed first followed by the cycle and the last leaf, then the total completion time is $2\sqrt{2}k^2 + 2k^2 + O(k)$. Hence, the ratio approaches $\sqrt{2}$ for increasing k . In fact, this is the approximation ratio of the algorithm as we shall prove next through Lemmas 3 and 4. Hence, Algorithm 3 performs best for general graphs.

Fig. 4 A lower bound of $3/2$ for Algorithm 1 on general graphs

Fig. 5 A lower bound of $13/9 \approx 1.444$ (in the limit) for Algorithm 2 on general graphs

Fig. 6 A tight instance (in the limit) for Algorithm 3. There are k leaves of length 1 and one cycle of length $k\sqrt{2}$

Fig. 7 The functions $\underline{L}(t)$ and $\overline{L}(t)$

Lemma 3 For any instance of total length (sum of edge lengths) L , the path of Algorithm 3 has an average completion time of at most $(2\alpha - \alpha^2/2 - 1)L$, where αL is the length of the path.

Proof Let $L(t)$ be the cumulative length that is served in the first t units of the constructed Euler path. We compute a lower bound $\underline{L}(t)$ on $L(t)$. (See Fig. 7.) The postman path traverses every edge at most twice. Hence, we have $L(t) \geq t/2$ for all $t \in [0, \alpha L]$. Consider the function

$$\underline{L}(t) = \begin{cases} t/2, & \text{for } 0 \leq t \leq 2(\alpha - 1)L, \\ t + (1 - \alpha)L, & \text{for } 2(\alpha - 1)L \leq t \leq \alpha L. \end{cases}$$

If $L(t) < \underline{L}(t)$ for some $t \in [2(\alpha - 1)L, \alpha L]$, then we can never complete a length L before time αL . Therefore we have $L(t) \geq \underline{L}(t)$ for all t . The average completion time of $\underline{L}(t)$ is $(2\alpha - \alpha^2/2 - 1)L$. $\square$

Lemma 4 For an instance of total length L with an optimal postman path of length αL , any solution has average completion time at least $(\alpha^2/2 - \alpha + 1)L$.

Proof Consider an arbitrary solution and let $L(t)$ be the cumulative length that is served in the first t units. The function $\bar{L}(t)$ is defined by

$$\bar{L}(t) = \begin{cases} t, & \text{for } 0 \leq t \leq (2 - \alpha)L, \\ t/2 + (1 - \alpha/2)L, & \text{for } (2 - \alpha)L \leq t \leq \alpha L, \\ L, & \text{for } t \geq \alpha L. \end{cases}$$

The average completion time of $\bar{L}(t)$ is $(\alpha^2/2 - \alpha + 1)L$. We will prove that $L(t) \leq \bar{L}(t)$ for all $t \geq 0$. Assume the opposite. Then there must be a moment x for which $L(x) > \bar{L}(x)$ and $(2 - \alpha)L < x < \alpha L$. Let P be the path up to moment x . Clearly, there exists a path, traversing P once and everything not in P at most twice. The length of this path is at most $x + 2(L - L(x)) < x + 2(L - \bar{L}(x)) = \alpha L$, which contradicts that the minimum postman path has length exactly αL by definition. $\square$

Theorem 3 Algorithm 3 is a $\sqrt{2}$ -approximation for the Chinese Deliveryman Problem.

Proof We combine both lemmas above and optimize over α . The approximation ratio is at most

$$(2\alpha - \alpha^2/2 - 1)/(\alpha^2/2 - \alpha + 1), \quad (2)$$

which attains its maximum of $\sqrt{2}$ for $\alpha = \sqrt{2}$. $\square$

4.2 2-edge connected graphs

Using graph properties one can prove better bounds. A famous theorem by Petersen (1891) states that any 2-edge connected cubic graph has a perfect matching. Moreover, any edge-weighted cubic graph has a perfect matching of total length at most one third of the total length of all edges (Edmonds and Johnson 1973). Any 2-edge connected graph can be made cubic by adding edges and vertices. By giving new edges length zero, it follows that a 2-edge connected graph has a postman tour of length at most $4L/3$, where L is the total length of the graph.

Theorem 4 Algorithm 2 is a $4/3$ -approximation for 2-edge connected graphs.

Proof Let L be the total length of the graph. Then the optimal postman tour has length at most $4L/3$. Hence, when traversed in the best direction, the average completion time is at most $2L/3$. On the other hand, a lower bound on the average completion time is $L/2$. Hence, the approximation ratio of the algorithm is at most $4/3$. $\square$

The example in Fig. 8 shows that the ratio is tight in the limit. It is the same example as in Fig. 5 but without the long edge. The optimal average completion time is $3n/2$. The optimal postman tour has length $4n$ and the optimal postman tour with the pattern described for Fig. 5 has average completion time $2n + O(1)$ in both directions.

The approximation ratio of Algorithms 1 and 3 is strictly larger than that of Algorithm 2 as shown by the example in Fig. 9. The 3 edges incident to vertex i are labeled a_i, b_i, c_i . An optimal postman tour has length $4n$. The minimum average completion time is attained by the path $a_1, b_1, a_2, b_2, \dots, a_n, b_n, c_1, \dots, c_n$.

Fig. 8 A tight instance (in the limit) for Algorithm 2 on $n + 1$ vertices and $3n$ edges of length 1. The optimal average completion time is $3n/2$. The optimal postman tour has length $4n$ and there exists one that has an average completion time $2n + O(1)$ in both directions

Fig. 9 A lower bound of $7/5$ (in the limit) for Algorithms 1 and 3 on 2-edge connected graphs

The average completion time is $5n/3 + O(1)$. Another possible postman tour is $a_1, a_1, a_2, a_2, \dots, a_n, a_n, b_1, c_1, \dots, b_n, c_n$ which has an average completion time of $7n/3 + O(1)$. Hence, the ratio of Algorithm 1 is not better than $7/5$. Clearly, this holds for Algorithm 3 as well since the difference between an optimal postman tour and optimal path is only a single edge of length 1. It is easy to prove that $7/5$ is indeed the ratio of Algorithm 3.

Theorem 5 *Algorithm 3 is a $7/5$ -approximation for 2-edge connected graphs.*

Proof As we observed above, for 2-edge connected graphs, the length of the optimal postman path is no more than $4L/3$, where L is the total length of the graph. Hence $\alpha \leq 4/3$. The ratio in (2) is increasing for $1 \leq \alpha < \sqrt{2}$. Hence, for $\alpha \leq 4/3$, the ratio has its maximum of $7/5$ for $\alpha = 4/3$.

4.3 A PTAS for planar graphs

As observed in Sect. 1.1, any α -approximation scheme for TRP implies an α -approximation scheme for the CDP. In the reduction, we simply replace each edge by a polynomially bounded number of points, proportional to its length. Since this reduction preserves planarity, any PTAS for planar graph TRP implies a PTAS for CDP in planar graphs. A PTAS for planar graph TRP is given in Sitters (2019). That PTAS can be simplified to a large extent for the CDP. In fact, already the quasi-PTAS for planar TRP as suggested by Arora and Karakostas (2003) implies a PTAS for CDP on planar graphs. We sketch a proof here. The crucial observation made in Arora and Karakostas (2003) is that with loss of a factor $(1 + \epsilon)$ in the approximation, one may restrict to solutions that are the concatenation of $O(\log n/\epsilon)$ TSP-paths. (We say that the part of a solution between two time points is a TSP-path, when it is a shortest path among all paths visiting the same points and with the same start and end point.) Then, the PTAS for planar TSP by Arora et al. (1998b) can easily be generalized to this setting, resulting in a quasi-polynomial running time. However, the proof in Arora

Fig. 10 Taking an optimal postman tour for the deliveryman can be arbitrarily bad. An optimal postman tour or path will visit edge e of length $2K$ last and has total completion time $\Omega(K^3)$. The optimal solution here has total completion time $O(K^2)$

and Karakostas (2003) had a critical flaw. In the PTAS for planar TSP (Arora et al. 1998b) edges of total length $O(\epsilon)$ times the optimal tour length are contracted and then uncontracted again in the end. This is fine for TSP but not for TRP since this could increase the total completion time by much more than only a factor $(1 + \epsilon)$. For CDP however, contraction is not a problem and moreover, we may restrict to solutions that are the concatenation of only $O(1/\epsilon^2)$ TSP-paths, which turns the suggested QPTAS of Arora and Karakostas (2003) into a PTAS. We now sketch a proof of this below.

Consider some instance of planar CDP and turn it into a TRP by adding a large number of points on the edges. More precisely, we round the edge lengths to polynomially bounded integer values and consecutively, replace any edge e of length $l(e)$ by $l(e)$ edges of length 1. Let L be the total number of points. Then, L^2 is an upper bound on the total completion time of the optimal TRP solution. Hence, the number of points visited after time $t = L/\epsilon$ is less than ϵL . If we replace the solution after time t by a TSP-path (which has length at most $2L$) then the increase in total completion time is $O(\epsilon L^2)$. The part before time t is partitioned into segments of length ϵL each and each is replaced by a TSP-path. Again, the total increase is no more than ϵL^2 since the increase is at most ϵL for any point. Since $\sum_{i=0}^{L-1} i = L(L-1)/2$ is a lower bound on the total completion time we see that the total increase due to replacing segments by TSP-paths is $O(\epsilon)\text{OPT}$. Hence, the optimal solution can be approximated by a concatenation of $O(1/\epsilon^2)$ TSP-paths and the PTAS for planar TSP by Arora et al. (1998b) can easily be generalized to this setting, exactly as was done in the QPTAS from Arora and Karakostas (2003).

5 Final remarks

We showed that simply following a postman tour or postman path gives a relatively small approximation ratio, considered that the problem is APX-hard in general. A possible improvement might be to take the best of the solutions returned by Algorithms 2 and 3. Another improvement may come from exploiting graph properties as we did for trees, planar graphs, and 2-edge connected graphs.

The approximation of the directed CDP remains open. Following an optimal postman tour or path may be far from optimal as shown in Fig. 10. Hence, to find a constant factor approximation algorithm one needs to do something smarter. An obvious approach is to reduce it to asymmetric TRP in the same way as we reduced undirected CDP to symmetric TRP (in the introduction). So far, the best approximation ratio for asymmetric TRP is $O(\log n)$ (Friggstad et al. 2013).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Arora S, Karakostas G (2003) Approximation schemes for minimum latency problems. *SIAM J Comput* 32:1317–1337
- Arora S, Safra S (1998) Probabilistic checking of proofs: a new characterization of NP. *J ACM* 45:70–122
- Arora S, Lund C, Motwani R, Sudan M, Szegedy M (1998a) Proof verification and the hardness of approximation problems. *J ACM* 45:501–555
- Arora S, Grigni M, Karger D, Klein P, Woloszyn A (1998b) A polynomial time approximation scheme for weighted planar graph TSP. In: 9th ACM–SIAM symposium on discrete algorithms, pp 33–41
- Baston V, Beck A (1995) Generalizations in the linear search problem. *Israel J Math* 90:301–323
- Beck A, Beck M (1984) Son of the linear search problem. *Israel J Math* 48:109–122
- Blum A, Chalasani P, Coppersmith D, Pulleyblank W, Raghavan P, Sudan M (1994) The minimum latency problem. In: Proceedings of 26th ACM symposium on theory of computing, Montreal, Quebec, Canada, pp 163–171
- Bonato A, Yang B (2013) Graph searching and related problems. In: Pardalos PM, Du DZ, Graham RL (eds) Handbook of combinatorial optimization. Springer, New York, pp 1511–1558
- Chaudhuri K, Godfrey B, Rao S, Talwar K (2003) Paths, trees, and minimum latency tours. In: Proceedings of 44th symposium foundations of computer science, Cambridge, MA, pp 36–45
- Edmonds J (1965) The Chinese postman problem. *Oper Res* 13:73–77
- Edmonds J, Johnson E (1973) Matchings, Euler tours and the Chinese postman. *Math Program* 5:88–124
- Friggstad Z, Salavatipour MR, Svitkina Z (2013) Asymmetric traveling salesman path and directed latency problems. *SIAM J Comput* 42:1596–1619
- Garey M, Johnson D, Stockmeyer L (1976) Some simplified NP-complete problems. *Theor Comput Sci* 1:237–267
- Grötschel M, Yuan Y (2012) Euler, Mei-Ko Kwan, Königsberg, and a Chinese postman. *Doc Math* 43–50
- Kella O (1993) Star search—a different show. *Israel J Math* 81:145–159
- Koutsoupias E, Papadimitriou C, Yannakakis M (1996) Searching a fixed graph. In: Proceedings of 23rd international colloquium on automata, languages, and programming, LNCS, vol 1099, Paderborn, Germany, Springer, pp 280–289
- Kwan MK (1960) Programming method using odd or even points. *Acta Math Sin* 10:263–266
- In Chinese
- Minieka E (1989) The delivery man problem on a tree network. *Ann Oper Res* 18:261–266
- Papadimitriou C (1976) On the complexity of edge traversing. *J ACM* 23:544–554
- Papadimitriou C, Yannakakis M (1993) The traveling salesman problem with distances one and two. *Math Oper Res* 18:1–11
- Petersen J (1891) Die theorie der regulären graphen. *Acta Math* 193–220
- Sitters R (2002) The minimum latency problem is NP-hard for weighted trees. In: Cook WJ, Schulz A (eds) Proceedings 9th international conference on integer programming and combinatorial optimization, LNCS, vol 2337. Springer, pp 230–239
- Sitters R (2019) Polynomial time approximation schemes for the traveling repairman and other minimum latency problems. Technical report
- van Ee M (2017) Routing under uncertainty: approximation and complexity. Ph.D. thesis, Vrije Universiteit Amsterdam
- van Omme N (2011) Le problème du postier chinois cumulatif. Ph.D. thesis, University of Montreal

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