Arc-routing for winter road maintenance[*]

Jiří Fink
Department of Theoretical Computer Science and Mathematical Logic,
Charles University, Czech Republic
Martin Loebl
Department of Applied Mathematics, Charles University, Czech Republic

Abstract

The arc-routing problems are known to be notoriously hard. We study here a natural arc-routing problem on trees and more generally on bounded tree-width graphs and surprisingly show that it can be solved in a polynomial time. This implies a sub-exponential algorithm for the planar graphs and small number of maintaining cars, which is of practical relevance.

1 Introduction

In the arc-routing problems, one needs to cover the graph of the road network by subgraphs such that every subgraph would be maintained by one vehicle and then one needs to design routing for a given subgraph by one vehicle.

1.1 Motivation

The problem we study here is motivated by the design of tours for vehicles in winter road maintenance. The standard length of the working shift of a maintenance driver is eight hours in the Czech Republic. Moreover, the Czech legislation requires multiple security brakes for drivers during the working shift. It is natural to expand the time for the security breaks and for all other non-driving manipulations of a vehicle to two hours per shift; this reduces the total time of driving to six hours. For simplicity, we will assume in our model that the working shift lasts six hours during which there are no breaks for the drivers and also the time to load the maintenance material is negligible.

Road network has a service hierarchy defined by the legislation based on traffic volume which partitions the roads into classes. In the Czech republic,

---

there are three such classes: Arterial roads through regions have the highest level of service priority (1). Priority (2) is assigned to bus routes and other important routs. Third priority of service is assigned to local roads. Each class of roads is associated with maximum time of maintenance completion. For instance, in our simplified model of road-maintenance in the Czech Republic we assume that the edges in the first priority level have to be cleaned by a vehicle every two hours, in the second level every four hours and in the third level every six hours.

We will also assume that there are no one-way roads and that for each road, the same vehicle maintains both its sides.

Definition 1. [Graph of road network] Let $G = (V, E)$ be a graph representing a *road network*. Vertices represent crossroads (and dead ends) and edges represent roads among them. Let $z \geq 1$ denote the number of priority classes of roads and let M denote the set of types of maintenance, e.g., $M = \{c, i, s\}$ in the Czech Republic. We associate several functions with G:

- $\alpha : E \to R^+$ gives to every edge a non-negative length,

- $p : E \to \{1, \ldots, z\}$ priority level,

- $m : E \to M$ type of maintenance.

Let $D \subset V$ be a set of depots. For $d \in D$ we denote by $m(v) \in M$ the stored material at depot d.

1.2 Our model

The problem is to design tours for vehicles of winter road maintenance. We want to find a solution with the minimum number of tours and with minimum number of deadheading. In the language of graph theory we want to find how to optimally cover the graph of the road network by subgraphs such that every subgraph would be maintained by one vehicle.

Then we need to design routing for a given subgraph by one vehicle. We need to have in mind that the vehicle may traverse also edges which are maintained by another vehicle (deadheading).

Let G^s denote the symmetric orientation of graph G, i.e., each edge of G appears twice, oppositely oriented, in G^s. The solution of the model has the following parts.

- We construct a partition $P = \{P_1, \ldots, P_r\}$ of the set of arcs of G^s into sets $P_1, \ldots, P_r$ and for each i we assign vertex (depot) $d_i \in D$. We assume the type of maintenance m constant in P_i. We also assume that the oppositely oriented edges belong to the same P_i.

- We construct, for each i, set R_i so that $P_i \subset R_i$ and each arc of P_i may be reached from d_i by a directed closed walk of R_i.

- For each i, we design a schedule of maintenance of the edges of P_i by a single vehicle starting and terminating at d_i and using only arcs of R_i, with the period of six hours. The schedule must meet

  (1) the requirements given by priorities $p(e), e \in P_i$,

  (2) The capacity c_m describes the maximum length of the route which can be maintained with only one loading of the material $m \in M$. The capacity condition requires that during each spreading material m on road-length c_m the vehicle must pass its depot d_i at least once.

  It turns out that it is more convenient to define c_m as a fraction of the maximum route-length rather than as a 'number'.

  If the maintenance method is snowplowing for P_i, i.e., no spreading material is needed, then clearly c_m is formally defined as a large enough number.

- In the actual algorithm described in Section 3, we perform the tree steps above simultaneously.

1.3 State of the art

Winter road maintenance is recognised as a notoriously hard problem (not only) from the algorithmic point of view. As far as we know, most of the literature in the algorithmic winter road maintenance concentrates in designing algorithms, which are typically based on Integer Linear Programming (LP), Constrain Programming (CP) and a local heuristics. The complexity of such algorithms is at least exponential.

An overview of literature on the problem of winter road maintenance and its solutions is [10, 11, 12, 13]. An excellent recent overview illustrating main works on the General Routing Problem can be found in [5] where the authors design a new branch-and-cut algorithm for the capacitated general routing problem. In [9], the authors also consider road priorities and a precedence relation between roads of different priority. In [15], the authors aim at constructing the routes schedule minimising the longest route; the network may have one-way streets and is modelled as a mixed graph.

Kinable et.al. [15] study a real-world snow plow routing problem (in the USA) and they compare three methods based on Integer Linear Programming (LP), Constrain Programming (CP) and a local heuristic. Ciancio et.al. [16] applied Branch-price-and-cut method for the Mixed Capacitated General Routing Problem with Time Windows. Other heuristic algorithms can be found e.g. in [4, 8, 7].

In [1] we introduced an heuristic approach with a very competitive implementation.

1.4 Main contribution

We realised that our approach in [1] can be formalised into a realistic model of winter road maintenance which admits algorithms with a sub-exponential complexity for some classes of graphs; and some of these classes, like the class of the planar graphs, do not add unrealistic conditions to the actual road networks to be maintained.

In this paper, we show that a natural formalisation of the routing problem (introduced above) admits a polynomial algorithm on trees and more generally on bounded tree-width graphs. This implies a sub-exponential algorithm for the practically relevant planar graphs and a small $(o(\sqrt{n}))$ number of maintenance vehicles.

Acknowledgement. We would like to thank to Petra Pelikánová for fruitful discussions.

2 Case of One Route: Admissible plans

Definition 2 (Pre-maintaining plan). *Pre-maintaining plan* is a tuple (G, P, d, α, z, p) where

(1) $G = (V, E)$ is a graph,
(2) $P \subset E$ is the set of maintained edges,
(2) $d \in V$ is called the *depot*,
(3) $\alpha : E \to Z^+$ gives to every edge a non-negative integer length,
(4) $p : E \to \{1, \ldots, z\}$ gives to each edge its priority level.

Next we define the *vehicle route* on T which models the route of one maintaining vehicle of one whole working shift, e.g. six hours in our simplified model of the road maintenance of the Czech Republic.

Definition 3 (Vehicle route). Let (G, P, d, α, z, p) be a pre-maintaining plan. A $L, c, t, f-vehicle\ route$ on G is a closed walk defined by a sequence of arcs $w = (e_1, \ldots, e_l)$ from G^s satisfying:

1. w starts and ends at d and each arc of P^s is traversed at least once by w and at most $f(e)$ times,

2. (total length) $\sum_{q \leq l} \alpha(e_q) \leq L$,

3. (priority) For all pairs $i < j$ and y such that e_i is appears exactly $y - 1$ times among $e_1, \ldots, e_{i-1}$, $e_i = e_j$ and $e_i \neq e_k$ for $i < k < j$, $\sum_{q \geq i}^{j-1} \alpha(e_q) \leq Lt(e_i, y))$, taken cyclically,

4. (capacity) For all pairs $i < j$ such that e_i starts at d, e_j ends at d and e_k is not incident with d for $i < k < j$, $\sum_{i \leq q \leq j, e_q \in P} \alpha(e_q) \leq cL$.

Even if we assume that G is a tree, $z = 3$ and $t(e, y) = p(e)/3$, the existence of a maintaining route is a hard problem:

Observation 1. It is an NP-complete problem to decide if a given pre-maintaining plan admits a $L, c, t, f-$vehicle route.

Proof. The problem is hard even if G is a star rooted at its vertex of largest degree, $P = E$, $z = 3$, $p(e) = 3$ and $t(e, y) = 1/3$ are constant functions and $c = 1/2$. Such input tree admits a vehicle route if and only if the edges can be divided into two parts with the two sums of lengths equal. ☐

Practically we can assume that more conditions hold for maintaining trees.

Definition 4 (Maintaining plan). *Maintaining plan* is a pre-maintaining plan which in addition satisfies that

(1) the length-function α gives to each edge a positive integer bounded by a constant k; we will further assume w.l.o.g. that $\alpha(e) = 1$ for each $e \in E$.

(2) Further we assume that $P = E$: this assumption is supported by practical experiments which indicate that in practical situations deadheading is negligible with respect to the capacity constraints.

(3) We will also assume that each vertex-degree in T is bounded by a constant; we will denote it by Δ. This condition holds in all road networks.

Definition 5 (Admissible plan). Maintaining plan is *admissible* if and only if it admits a $L, c-$vehicle route.

2.1 Examples

First, let G consist of a path P of $L/8$ vertices rooted at an end-vertex where each edge has priority 1, and $L/8$ leaves where each leaf is attached to a different vertex of P by an edge of priority 3. It is not difficult to see that this maintaining tree is admissible.

Secondly let G be a path of $L/4$ edges rooted at an end-vertex, where the initial (from the root) part of length $L/8$ has priority 1 and the remaining part has priority 3. It is not difficult to see that this maintaining tree is not admissible.

We note that in both examples, twice the number of edges of G is $L/2$.

2.2 Deciding Admissibility for trees

We will show in this section that there is a polynomial algorithm based on dynamic programming which can decide if a given maintaining plan (T, d, z, p), T tree, is admissible.

Theorem 1. Fixed integers F, Δ. There exists a polynomial time algorithm which for a tree $T = (V, E)$ rooted in d with maximal degree at most Δ, function $f : E \to N$ such that $f(e) \leq F$ for all $e \in E$ and function $g : E \times \{1, \ldots, F\} \to N$ decides whether there exists a closed walk w starting at r satisfying

- Every edge e of T^s is traversed $f(e)$-times in both directions.

5

- For every edge e of T^s and $y \leq f(e)$, there are at most $g(e, y)$ steps between $y-$th and $(y+1)-$st traverses of e, taken cyclically.

Proof. The length of the route has to be $l = 2 \sum_{e \in E} f(e)$ and let $I = \{1, \ldots, l\}$ be the set of all indices on the route. For every $A \subseteq I$ and $v \in V$ let $M_v[A]$ be true if there exists route satisfying all conditions on $T[v]$ using exactly indices of A on $T[v]$. Similarly we define $M'_v[A]$ for $T'[v]$. Let $z(A)$ for the set of ordered pairs of starting and ending indices of subsequencies of A, i.e. $z(A) = \{(a_1, b_1), \ldots, (a_q, b_q)\}$ such that $A = \{a_1, \ldots, b_1\} \cup \cdots \cup \{a_q, \ldots, b_q\}$ and $a_1 \leq b_1 < b_1 + 1 < a_2 \leq b_2 < b_2 + 1 < \cdots < b_{q-1} + 1 < a_q \leq b_q$. Let $|z(a)| = q$ be the number of subsequences.

Let v be a non-root vertex and $e = vp(v)$. If $M'_v[A] = true$, then $|z(A)| \leq f(e)$ since e has to be traversed $f(e)$ (some traverses may be consecutive). Therefore, there are at most $f(e) \cdot l^{2f(e)}$ sets A such that $M'_v[A] = true$, so we can store all such sets A instead of whole table M'_v to ensure polynomial space. Similarly, if $M_v[A] = true$ then $|z(A)| \leq f(e)$ since $T[v]$ can be entered at most $f(e)$-times.

We determine M_v using the following dynamic programming. If v is a leaf, then $M_v[A] = true$ only for $A = \emptyset$. Consider that $u_1, \ldots, u_s$ are all children of v. Recall that $1 \leq s \leq \Delta$. First, we set $M_v[A] := false$ for all A and then we consider all combination A_i for $i = 1, \ldots, s$ such that $M'_{u_i}[A_i] = true$. Note that there are at most $F^\Delta \cdot l^{2F\Delta} \leq F^d \cdot (2Fn)^{2F\Delta}$ such combinations, so the algorithm is polynomial. Let $A = A_1 \cup \cdots \cup A_s$. We apply the following function for every combination.

- If any two sets of $A_1, \ldots, A_s$ have a common member, then the function terminates, since every index has to be used for exactly once on the route.

- If $z(A) > f(e)$, then the function terminates, since $T[v]$ can be entered at most $f(e)$-times where $e = vp(v)$.

- In the end, we set $M_v[A] := true$.

Now, we determine M'_v. Let $e = \{v, p(v)\}$. First, we set $M'_v[A] := false$ for all A and then we apply the following function for every A with $M_v[A] = true$.

- Let $z(A) = \{(a_1, b_1), \ldots, (a_q, b_q)\}$. If $q > f(e)$ then stop.

- Let $X_1 = \{a'_i; 1 \leq i \leq f(e)\}$, $X_2 = \{b'_i; 1 \leq i \leq f(e)\}$ and $X = X_1 \cup X_2$ be such that (1) $X \cap A = \emptyset$ (2) $X_1 \cap X_2 = \emptyset$ and (3) for each $(a_i, b_i) \in z(A)$, $a_i - 1 \in X_1$ and $b_i + 1 \in X_2$. For each such X_1, X_2 we let $A' = A \cup X$.

- We check if X_1, X_2 satisfy the conditions for $g(e)$: if not, we stop.

- We set $M'_v[A'] := true$.

Finally, the algorithm returns $M_d[I]$. $\qquad\square$

We note that the same proof works if we require that every edge e is traversed *at most* $f(e)$-times in both directions.

Theorem 2. Let z, Δ, F be integer constants and let (T, d, z, p) be a maintaining plan where T is a tree with maximum degree Δ. Let $f : E \to N$ satisfies for each $e \in E$, $f(e) \leq F$. Then there is a polynomial algorithm to decide if a $L, c, t, f-$vehicle route on T exists.

Proof. We use Theorem 1 and note that we can require that every edge e is traversed *at most* $f(e)$-times in both directions, function t can be modelled by g and the capacity constraint can be modelled by connecting the depot to a new vertex of degree one and setting the proper value on $g(e)$ for the new edge. $\qquad\square$

2.3 Graphs of bounded tree-width

A *tree decomposition* of a graph G is a pair (W, b) where W is a tree and $b : V(W) \to 2^{V(G)}$ assigns a *bag* $b(v)$ to each vertex v of W such that

- every vertex is in some bag,

- every edge is a subset of some bag,

- every vertex of G appears in a connected subtree of the decomposition.

The *width* of the tree decomposition is defined as the size of the largest bag, minus one. The *tree-width* of graph G is the minimum width of a tree decomposition of G.

We will need a simple basic property connecting the cuts of G with tree decompositions. Let (W, b) be a tree decomposition of G and let $e = \{u, v\}$ be an edge of W. Let $W_{u,v}$ denote the the component of $W \setminus e$ containing v and let $G_{u,v}$ be the induced subgraph $G[\cup_{z \in V(W_{u,v})} b(z)]$.

Observation 2. Let (W, b) be a tree decomposition of G and let $e = \{u, v\}$ be an edge of W. Then $G = G_{u,v} \cup G_{v,u}$ and $V(G_{u,v}) \cap V(G_{v,u}) = b(v) \cap b(u)$. In particular, G has no edge with one end in $V(G_{u,v}) \setminus V(G_{v,u})$ and the other end in $V(G_{v,u}) \setminus V(G_{u,v})$.

Let $G = (V, E)$ have a distinguished vertex, denoted by d. It is useful to simplify the decomposition. A tree decomposition (W, b) is **canonical** if

- T is rooted, and the root r satisfies $d \in b(r)$.

- Each leaf u satisfies $|b(u)| = 1$.

- Each non-leaf vertex u satisfies one of the following conditions:

  u has exactly one son u' and $b(u) = b(u') \cup \{v\}$ for some vertex $v \in V$.

  u has exactly one son u' and $b(u) = b(u') \setminus \{v\}$ for some vertex $v \in V$.

  u has exactly two sons u', u'' and $b(u) = b(u') = b(u'')$.

It is straightforward to verify that every graph G of tree-width at most k has a canonical tree decomposition of width at most k, of polynomial size.

Theorem 3. Fixed integers F, Δ, k. There exists a polynomial time algorithm which for a graph $G = (V, E)$ rooted in d and with maximal degree at most Δ, given along with its canonical tree decomposition (W, b) of width $k - 1$ and functions $f : E^s \to N$ such that $f(e) \leq F$ for all $e \in E^s$ and $g : E^s \times \{1, \ldots, F\} \to N$ decides whether there exists a closed walk w starting at d satisfying

- Every edge e is traversed $f(e)$-times in both directions.

- For every edge e and $y \leq f(e)$, there are at most $g(e, y)$ steps between the $y-$th and $(y + 1)-$th traverses of e, taken cyclically.

Proof. We assume that for each bag $b(v)$, the edges of G^s incident to a vertex of $b(v)$ (there are at most $2k\Delta$ of them) are linearly ordered. The ordering may differ in different bags.

The length of the route has to be $l = 2\sum_{e \in E} f(e)$ and let $I = \{1, \ldots, l\}$ be the set of all indices on the route. Let $I' = \{(x, i); x \in I, 0 \leq i \leq 2k\Delta\}$. For every $A \subseteq I'$, and $u \in V(W)$ let $M_u[A]$ be true if there exists route $w = (e_1, \ldots, e_l)$ satisfying all conditions on $G_{p(u), u}$ so that:

- If $A_0 = \{x;$ there is i such that $(x, i) \in A\}$ then w uses exactly indices of A_0 on $G_{p(u), u}$,

- For each $(x, i) \in A$, $i = 0$ iff e_x is not incident to a vertex of $b(u)$.

- Let $S(A) = \{x \in A_0; e_x$ is incident with a vertex of $b(u)\}$. For each $x \in S(A)$, if $(x, i) \in A$ then the edge e_x of w is the $i-$the edge of the fixed linear order of the edgers incident with a vertex of $b(w)$.

Let $z(A) = \{(a_1, b_1), \ldots, (a_q, b_q)\}$ such that $A_0 = \{a_1, \ldots, b_1\} \cup \cdots \cup \{a_q, \ldots, b_q\}$ and $a_1 \leq b_1 < b_1 + 1 < a_2 \leq b_2 < b_2 + 1 < \cdots < b_{q-1} + 1 < a_q \leq b_q$. Clearly, $|z(a)| = q$ be the number of subsequences in A_0.

If $M_u[A] = true$ then $|z(A)| \leq |S(A)| \leq 2Fk\Delta$ since $G_{p(u), u}$ can only be entered from a vertex of $b(u)$ which is incident with at most Δ edges in $G_{p(u), u}$ and each such edge can be used at most $f(e)-$times. Therefore, there are at most $(2kl\Delta)^{2Fk\Delta}$ sets A such that $M_u[A] = true$, so we can store all such sets A instead of whole table M_u to ensure polynomial space.

We determine M_u using the following dynamic programming. Let u be a non-root vertex of W.

If u is a leaf, then $M_u[A] = true$ only for $A = \emptyset$.

If u is unique son of $p(u)$ and $b(p(u)) = b(u) \setminus \{v\}$ for some vertex v of G then $M_{p(u)}[A] = true$ iff $M_u[A'] = true$ where A' obtained from A but correcting the contribution of the linear order of edges associated with $b(p(u))$.

Let u be the unique son of $p(u)$ and $b(p(u)) = b(u) \cup \{v\}$ for some vertex v of G. We notice that no edge incident with v belongs to $G_{p(u), u}$. We construct sets A for which $M_{p(u)}[A] = true$ by considering edges from v to $b(u)$ one by one and for each such edge e we perform the same construction as the one of $M'_v[A']$ from $M_v[A]$ in the proof of Theorem 1.

Finally let $p(u)$ have two sons $u = u_1, u_2$. We know $b(p(u)) = b(u_1) = b(u_2)$. Let $S = E^s(G_{p(p(u_1)),u_1} \cap E^s(G_{p(p(u_2)),u_2})$. We observe: if $e \in S$ then e is incident with a vertex of $b(p(u_1))$.

We let again $M_{p(u)}[A] = false$ for each A and do the following:

Consider all pairs A_1, A_2 such that $M_{u_1}[A_1] = true$ and $M_{u_2}[A_2] = true$. We first modify the elements of both A_1, A_2 to reflect the fixed linear order of the edges incident with a vertex of $b(p(v))$; this linear order may be different from the linear order (of thesame set) fixed for $b(u_1)$ or for $b(u_2)$. Let the resulting sets be denoted by A'_1, A'_2.

For $e \in S$ let $i(e)$ denote its index in the fixed linear order of the edges incident with a vertex in $b(p(u_1))$. For each such $i(e)$ let $S_1 = \{x; (x, i(e)) \in A'_1\}$ and $S_2 = \{x; (x, i(e)) \in A'_2\}$. If $S_1 \neq S_2$ then stop.

If $[A'_1]_0 \cap [A'_2]_0$ contain any other element then stop.

If $A' = A'_1 \cup A'_2$ does not satisfy the requirements given by function g on the edges incident with a vertex of $b(p(u))$ then stop.

Let $M_{p(u)}[A'] = true$.

Finally, the algorithm returns $M_d[I]$.

$\square$

Analysing the proof of the above theorem, we get the following:

Theorem 4. Let z, Δ, F be integer constants and let (G, d, z, p) be a maintaining plan where $G = (V, E)$ is a graph rooted in d and with maximal degree at most Δ, given along with its canonical tree decomposition (W, b) of width $k - 1$ and functions $f : E^s \to N$ such that $f(e) \leq F$ for all $e \in E^s$ and $t : E^s \times \{1, \ldots, F\} \to N$. Then there is an algorithm to decide if a $L, c, t, f-$vehicle route on G exists of complexity at most $pol(|G|, |t|) \times (4kF|E|)^{4Fk\Delta}$.

2.4 Deciding Admissibility: negative results

Question: It is necessary to fix F and Δ? We first show that the admissibility is hard for unbounded f even if G is a tree and g depends only on the edge.

Theorem 5. It is NP-complete to decide whether a given binary tree $T = (V, E)$ rooted in d and function $f, g : E \to N$ there exists a closed walk w starting at r satisfying

- Every edge e is traversed $f(e)$-times in both directions.

- For every edge e, there are at most $g(e)$ steps between two consecutive traverses of e in both direction, taken cyclically.

The problem is NP-complete even if we restrict f to be non-increasing on all paths from the depot.

Proof. The 3-partition problems ask to decide whether a given integers $a_1, \ldots, a_{3n}$ can be split into n groups with the same sum. The problem is strongly NP-complete even if it is restricted to integers strictly between $S/2$ and $S/4$ where

9

S is the target sum. Note that in this case, every group has to contains exactly 3 integers.

Let $h = \lceil \log_2 3n \rceil$ and $B = 3h$ and $b_i = Ba_i$ for all $i = 1, \ldots, 3n$ and $S' = B(S+1)$.

Let T_i be a binary tree on b_i edges rooted in r_i. Let T' be a binary tree rooted in r' with leaves $r_1, \ldots, r_{3n}$ such that all leaves are in depth h. Let T be a binary tree such that

- d is the root of T

- d has only one child d'

- T' is attached to the node d'

- trees $T_1, \ldots, T_{3n}$ are attached to leaves of T'.

Note that the size of T is $O(\log n \sum_i a_i)$, so it is only $O(\log n)$-times larger than the size of the instance of 3-partition problem.

Next, $f(e) = 1$ for all edges e on trees $T_1, \ldots, T_{3n}$. For an edge e of T', $f(e)$ is the number of trees of $T_1, \ldots, T_{3n}$ in the subtree of e. Finally, $f(dd') = n$. The goal is to ensure there that the route can be split into n parts by passing dd' and each part traverses from d' to some r_i, whole tree T_i, returns to d' and then traverse two other trees of $T_1, \ldots, T_{3n}$. In order to ensure the proper sum, we set $g(dd') = 2S' + 2$ and $g(e)$ is a sufficiently larger number for all other edges e. Clearly, if integers can be split into n groups, there exists a route.

Consider a walk w. Clearly, every tree $T_1, \ldots, T_{3n}$ has to be traversed by w completely once it is entered. Traverses of dd' split w into n parts and every tree $T_1, \ldots, T_{3n}$ is completely traversed in one part. Note that $g(dd')$ ensures that one part traverses at most S' edges in both directions (excluding dd').

We prove that every part traverse exactly tree trees. For contradiction, assume that trees T_i, T_j, T_k, T_l are traversed in one part. Then, the number of edges in the part is at least $b_i + b_j + b_k + b_l + h = B(a_i + a_j + a_k + a_l) + h \geq 4B\frac{S+1}{4} + h = B(S+1) + h > S'$ which is a contradiction. Since all parts contain at most 3 trees, every part must contains exactly 3 trees.

Now, consider a part traversing trees T_i, T_j, T_k. For contradiction, assume that $a_i + a_k + a_j > S$. The number of traversed edges in the part is at least $b_i + b_j + b_k + h = B(a_i + a_j + a_k) + h \geq B(S+1) + h > S'$ which is a contradiction. Hence, the sum of integers corresponding to each group is exactly S. □

Next theorem treats the case unbounded degrees.

Theorem 6. Fixed integer F. It is NP-complete to decide whether a given tree $T = (V, E)$ rooted in d, function $f : E \to N$ such that $f(e) \leq F$ for every edge e and function $g : E \times \{1, \ldots, F\} \to N$ there exists a closed walk w starting at d satisfying

- Every edge e is traversed $f(e)$-times in both directions.

- For every edge e and $y \leq f(e)$, there are at most $g(e)$ steps between two $y-$the and $(y+1)-$st traverses of e in both direction, taken cyclically.

Proof. Consider an instance of 3-partition consisting of $3n$ integers $a_1, \ldots, a_{3n}$ and let S be the target sum. Let $B = n$ and $b_i = Ba_i$ for $i = 1, \ldots, 3n$ and $S' = BS + 1$ and $S'' = (n-1)S'$. Let T be tree which consists of

- a depo d, and

- n vertices $u_1, \ldots, u_n$ incident only to d where $f(du_i) = 2$ and $g(du_i, 1) = S'$ and $g(du_i, 2) = S''$, and

- $3n$ paths $P_1, \ldots, P_{3n}$ on $b_1, \ldots, b_{3n}$ edges such that one end-vertex of each path is d and these paths have to be traversed only once.

Note that the only possible length of a route is $S' + S''$ and the short and long distances between tranverses of edges du_i have to be exactly S' and S'', respectively. If $a_1, \ldots, a_{3n}$ can be partitioned into n groups of equal sum S then we construct a route as follows: Starts by traversing du_1, tranverse 3 paths of the first group, tranverse du_1, traverse du_2, etc.

Observe that there is no route such that two close traverses of an edge du_i which is interleaved by a traverse of an edge du_j since the sum of lengths of any subset of paths is divisible by B and even all edges edges $du_1, \ldots, du_n$ cannot contribute to a multiple of B. Hence, if there exists a route then it looks like as the one constructed above. $\square$

Note that the problem is NP-complete even if $F = 2$ and all vertices except one have degree 2.

Open problem. We post here an open problem: how hard is it to decide if a maintaining plan which is a planar square grid is admissible?

It follows from Theorem 10 that there is a sub-exponential algorithm, but the existence of a polynomial algorithm is still in the game.

3   Case of More Routes

The graph of road network (see Definition 1) has each edge maintained by one of elements of the set M; we will assume that M has a fixed size, e.g., $M = \{c, i, s\}$. It is natural to assume that each maintaining vehicle can snowplow and thus we can include deadheading in our model.

Definition 6. Let G be a graph of road network and G^s its symmetric orientation. Let $p : E \to \{1, \ldots, z\}$ be its priority function and $m : E \to M$ be its maintaining type function; we assume that the length of each edge is 1. Let D be the set of the depots. We say that a tuple (H, P, d, z, p) where H is a subgraph of G and $d \in D \cap V(H)$ is a *maintaining plan of G* if

- m is constant on H,

- (H, d, z, p) is an admissible maintaining plan (see Definition 4) and $P \subset E(H)$.

Definition 7 (Feasible and Optimal Solution). Feasible solution of a road network G is a set O of admissible plans of G so that the union of their $P-$sets covers $E(G)$. A feasible solution O is *optimal* if $|O|$ is as small as possible.

We remark that the requirements given by the functions f, g concern the individual routes only, not the whole set of the routes of a feasible solution.

In our paper [1] we describe a heuristic for finding an optimal solution. Here we concentrate on classes of networks which admit a subexponential algorithm.

3.1   Finding an optimal solution

First we note that finding an optimal solution is an NP-complete problem even for G a tree and $|D| = 1$.

Theorem 7. Fixed integers F, Δ. It is NP-complete to decide for a given tree $T = (V, E)$ rooted in d with maximal degree at most Δ and function $f, g : E \to N$, f bounded by $|F|$, and integer o bounded by a polynomial in $|T|$, if there exist closed walks $w_1, \ldots, w_o$ starting at d and subsets $P_1, \ldots, P_o$ of $E(T)$ satisfying

- $P_1 \cup \cdots \cup P_o = E(T)$,

- Every edge $e \in P_i$ is traversed at most $f(e)$-times in both directions in each w_i.

- For every edge $e \in P_i$, there are at most $g(e)$ steps between two consecutive traverses of e by w_i in both direction, taken cyclically.

Proof. A straightforward variant of the construction of the proof of Theorem 5 works.

□

Hence from now on let o be a fixed integer and we consider the optimization problem **Roadnet(o):** find out if there is a feasible solution of a road network consisting of at most o admissible plans.

Theorem 8. Fix integers F, Δ, k, o. There exists a polynomial time algorithm which for a graph $G = (V, E)$ with maximal degree at most Δ, given along with its canonical tree decomposition (W, b) of width $k - 1$ and $D \subset V(G)$, and functions $f : E^s \to N$ such that $f(e) \leq F$ for all $e \in E^s$ and $g : E^s \times \{1, \ldots, F\} \to N$ decides whether there exist closed walks $w_1, \ldots, w_o$ starting at a vertex of D and subsets $P_1, \ldots, P_o$ of $E(G)$ satisfying

- $E(G) = P_1 \cup \cdots \cup P_o$,

- For each i, each edge of P_i^s belongs to w_i,

- Every edge e is traversed at most $f(e)$-times in both directions, by each w_i,

- For every edge $e \in P_i$ and $y \leq f(e)$, there are at most $g(e, y)$ steps between the $y-$th and $(y+1)-$th traverses of e, taken cyclically.

Proof. This follows by an analogous dynamics as Theorem 3. □

We arrive at a result analogous to Theorem 4.

Theorem 9. Let z, Δ, F be integer constants and let (G, D, z, p, m) be a road network where $G = (V, E)$ is a graph with maximal degree at most Δ and $D \subset V$, given along with its canonical tree decomposition (W, b) of width $k - 1$ and functions $f : E^s \to N$ such that $f(e) \leq F$ for all $e \in E^s$ and $t : E^s \times \{1, \ldots, F\} \to N$. Then there is an algorithm for **Roadnet(o)** of complexity at most $pol(|G|, |t|) \times (4koF|E|)^{4Fko\Delta}$.

4 Planar networks

Most of the realistic medium size road networks are planar bounded degree, with at most ten thousand edges (road-segments) and around one hundred of the maintaining vehicles.

This naturally leads to studying planar road networks with n vertices and $O(\sqrt{n})$ maintenance cars. We do not know if this problem admits a sub-exponential algorithm.

However, if the number of cars is smaller, the previous theorem implies a sub-exponential algorithm.

Theorem 10. Let G be a planar road network on n vertices and of bounded degree and let an optimum solution has size $o(\sqrt{n})$. Then an optimal solution of G can be found in time $2^{O(\sqrt{n} \log(n))}$.

Proof. This follows from the fact that every planar graph of n vertices has tree width at most $\sqrt{n}$. □

We note that this is close to optimal even for a constant number of cars by the following result of [2]: Assuming the Exponential Time Hypothesis, Steiner Tree problem on planar graphs cannot be solved in time $2^{o(k)} n^{O(1)}$ where k is the number of terminals.

References

[1] J. Fink, M. Loebl, P. Pelikanova, A New Arc-Routing Algorithm Applied to Winter Road Maintenance, preprint 2019.

[2] D. Marx, M. Pilipczuk, M. Pilipczuk, On sub-exponential parameterised algorithms for Steiner Tree and Directed Subset TSP on planar graphs, FOCS 59 (2018).

[3] T. Tantau, A Gentle Introduction to Application of Algorithmic Metatheorems for Space and Cicuit classes, Algorithms 9, 44 (2016).

[4] Binglei Xie, Ying Li, Lei Jin: *Vehicle routing optimization for deicing salt spreading in winter highway maintenance.* Procedia - Social and Behavioral Sciences 96 (2013), 945–953.

[5] A. Bosco, D. Lagana, R. Musmanno, F. Vocaturo: *A Matheuristic Algorithm for the Mixed Capacitated General Routing Problem.* Networks 64(4) (2014), 262–281.

[6] M.R. Garey, D.S.Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co. New York, USA (1979), ISBN:0716710447.

[7] L. Gaspar, Z. Bencze: *Salting route optimization in Hungary.* Transportation Research Procedia 14 (2016), 2421–2430.

[8] I. Gudac, I. Marovic, T. Hanak: *Sustainable optimization of winter road maintenance services under real-time information.* Procedia Engineering 85 (2014), 183–192.

[9] N. Perrier, A. Langevin, C.A. Amaya: *Vehicle routing for urban snow plowing operations.* Transp.Sci. 42 (2008), 44–56.

[10] N. Perrier, A. Langevin, J. F. Campbell: *A survey of models and algorithms for winter road maintenance. Part I: system design for spreading and plowing.* Computers & Operations Research 33 (2006), 209–238.

[11] N. Perrier, A. Langevin, J. F. Campbell: *A survey of models and algorithms for winter road maintenance. Part II: system design for snow disposal.* Computers & Operations Research 33 (2006), 239–262.

[12] N. Perrier, A. Langevin, J. F. Campbell: *A survey of models and algorithms for winter road maintenance. Part III: Vehicle routing and depot location for spreading.* Computers & Operations Research 34 (2007), 211–257.

[13] N. Perrier, A. Langevin, J. F. Campbell: *A survey of models and algorithms for winter road maintenance. Part IV: Vehicle routing and fleet sizing for plowing and snow disposal.* Computers & Operations Research 34 (2007), 258–294.

[14] Sándor F. Tóth, Marc E. McDill, Nóra Könnyü, Sonney George: *Testing the Use of Lazy Constraints in Solving Area-Based Adjacency Formulations of Harvest Scheduling Models.* Forest Science, Volume 59, Issue 2, (2013) 157–176

[15] J. Kinable, Joris, W-J. van Hoeve, and S. F. Smith: *Optimization models for a real-world snow plow routing problem.* International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Springer, Cham, 2016.

[16] C. Ciancio, D. Laganá, and F. Vocaturo: *Branch-price-and-cut for the mixed capacitated general routing problem with time windows.* European Journal of Operational Research 267.1 (2018): 187–199.