A Fast $(2 + \frac{2}{7})$-Approximation Algorithm for Capacitated Cycle Covering [*]

Vera Traub Thorben Tröbst[†]

June 7, 2021

Abstract

We consider the *capacitated cycle covering problem*: given an undirected, complete graph G with metric edge lengths and demands on the vertices, we want to cover the vertices with vertex-disjoint cycles, each serving a demand of at most one. The objective is to minimize a linear combination of the total length and the number of cycles. This problem is closely related to the capacitated vehicle routing problem (CVRP) and other cycle cover problems such as min-max cycle cover and bounded cycle cover. We show that a greedy algorithm followed by a post-processing step yields a $(2 + \frac{2}{7})$-approximation for this problem by comparing the solution to a polymatroid relaxation. We also show that the analysis of our algorithm is tight and provide a $2 + \epsilon$ lower bound for the relaxation.

1 Introduction

Our work is motivated by the classical and well-studied capacitated vehicle routing problem (CVRP) which was introduced by Dantzig and Ramser [8]. In this problem we are given an undirected, complete graph $G = (V, E)$ with metric edge lengths $\ell : E \to \mathbb{R}_{\geq 0}$ and a distinguished vertex $s \in V$ which is called the *depot*. Moreover, every vertex is assigned a demand $b(v)$. The goal is to cover V with cycles $C_1, \ldots, C_k$ such that each cycle visits s, satisfies $b(C_i) \leq 1$ and the total length $\sum_{i=1}^{k} \ell(C_i)$ is minimum. Here $b(C_i) \coloneqq \sum_{v \in V(C_i)} b(v)$ is the total demand of the vertices of C_i and $\ell(C_i) \coloneqq \sum_{e \in E(C_i)} \ell(e)$ is the total length of the edges of C_i.

The CVRP has received a large amount of attention in the last 60 years. While there has been much progress regarding computational results (see e.g. [22, 18, 19]), from the viewpoint of approximation algorithms only small progress has been made. The simple optimal tour partitioning algorithm by Altinkemer and Gavish [1], which achieves an approximation ratio of 3.5, has not been substantially improved in the past 30 years. (In fact, the approximation ratio is $2 + \alpha$ where α is the best known approximation ratio for TSP.) For the so-called unit-demand variant where all vertices have demand $1/Q$ for some $Q \in \mathbb{N}$, the tour partitioning algorithm by Haimovich and Kan [14] from 1985 has approximation ratio $1 + \alpha$, which is currently 2.5.

Significant improvements have been achieved in special cases, such as when the metric is Euclidean [9, 15] or arises from graphs with special structure [16, 2, 4, 3]. In the general case, two improvements have been made. The first is by Bompadre et al. [6] who improved the approximation guarantee by $\Theta(1/Q^3)$ where Q is the least common denominator of the (rational) demands b. Very recently, Blauth et al. [5] have announced $3.5 - \epsilon$ and $2.5 - \epsilon$ algorithms for the general CVRP and the unit-demand CVRP respectively.

In this paper we study a variant of the CVRP, where we do not have a depot vertex that must be visited by every tour, but instead have a fixed opening cost $\gamma > 0$ per tour. Formally, this problem, which we call the *capacitated cycle covering problem* (CCCP), is defined as follows. We are given an undirected, complete graph $G = (V, E)$ with metric edge lengths $\ell : E \to \mathbb{R}_{\geq 0}$, vertex demands $b : V \to [0, 1]$, and an opening cost $\gamma \in \mathbb{R}_{\geq 0}$. The goal is to compute a *capacitated cycle cover*, i.e. cycles $C_1, \ldots, C_k$ in G, such that every $v \in V$ is contained in exactly one cycle and $b(C_i) \leq 1$ for all i, minimizing the total cost $\sum_{i=1}^{k} \ell(C_k) + \gamma k$. Here it is allowed that a cycle contains only one or two vertices.

[*] An extended abstract appeared in the proceedings of IPCO 2020.

[†] Supported in part by NSF grant CCF-1815901

To the best of our knowledge, this precise problem formulation has not appeared in the literature. However, besides the capacitated vehicle routing problem, the CCCP is also closely related to other cycle covering problems. This includes min-max cycle cover and bounded cycle cover which were first studied by Even et al. [12]. In the former problem we are asked to compute a cycle cover $C_1, \dots, C_k$ which minimizes $\max_{i=1}^{k} \ell(C_i)$ where k is part of the input. In the latter we wish to find a cycle cover $C_1, \dots, C_k$ with $\ell(C_i) \leq 1$ for all i with minimum k. Recently, Yu et al. [23, 25] provided new approximation algorithms for the these problems with approximation ratios of 5 and $4 + 4/7$ and running times of $O(n^3)$ and $O(n^5)$ respectively. Here and in the following $n := |V|$.

Even more recently, Das et al. [10] studied the min-max variant of the capacitated cycle covering problem. In this problem we wish to find a capacitated cycle cover $C_1, \dots, C_k$ where k is part of the input such that $\max_{i=1}^{k} \ell(C_i)$ is minimized. They provide a constant factor approximation algorithm (the factor is > 250 and they do not specify it exactly) for min-max capacitated tree cover which implies a constant factor approximation algorithm for the cycle cover variant.

We remark that the simpler problem of finding a minimum cycle cover with at most k cycles admits a straight-forward 2-approximation: simply compute a minimum k-tree cover using Kruskal's algorithm and double the edges to obtain a cycle cover. However, in the case of the CCCP it is NP-hard to solve the analogous capacitated tree covering problem (or even to approximate it within $3/2 - \epsilon$) since it contains the bin packing problem.

1.1 Our Results and Techniques

Note that the capacitated cycle covering problem includes both the TSP (for $b \equiv 0$ and suitably large γ) and bin packing (for $\ell \equiv 0$) and is thus NP-hard to approximate within a factor of $3/2 - \epsilon$. Hence, we are primarily interested in approximation algorithms and relaxations for the problem. Our main result is the following theorem.

Theorem 1. *Given an instance of the capacitated cycle covering problem, we can compute a* $(2 + 2/7)$*-approximate solution in* $O(n^2)$ *time.*

We remark that if the pairwise distances between all vertices are given explicitly, the input has size n^2 and hence the runtime is linear.

The first step of our algorithm is to compute a carefully chosen spanning forest in our input graph. Having such a forest, we turn it into a capacitated cycle cover as follows. We first ensure that every connected component of the forest contains vertices of total demand at most 1. This is done by splitting large components into smaller ones if necessary. Then from every connected component of the forest we can compute a cycle of at most twice the length of the forest component. See Section 2.

The most important part of our algorithm is to choose the initial spanning forest. We do not solve a tree covering problem as a black box but anticipate that we will have to double edges and split up large components. To compute our spanning forest we use a linear programming relaxation, which we call the tree cover LP. This LP is closely related to a natural LP relaxation for the capacitated vehicle routing problem. Moreover, the tree cover LP has the important property that the set of feasible solutions is a polymatroid. This allows us to solve the LP very efficiently using the polymatroid greedy algorithm. See Section 3.

We then analyze a simple randomized rounding algorithm that rounds a fractional LP solution to a spanning forest. For this we exploit that the extreme point solutions of our LP relaxation are highly structured. As a result, we obtain a randomized $(2 + 2/7)$-approximation algorithm for the CCCP and also show that the ratio between our solution for CCCP and the value of the tree cover LP is at most $2 + 2/7$. See Section 4.

Then we show that we can derandomize our algorithm and obtain a simple and deterministic greedy algorithm for computing our spanning forest (Section 5). This will complete the proof of Theorem 1.

We also provide two forms of lower bounds for our analysis: we prove that the analysis of our deterministic algorithm is tight and we show a $2 + \epsilon$ lower bound on the gap between the tree cover LP and the capacitated cycle covering problem (Section 6).

Finally, in Section 7 we discuss the connection between the CCCP and the CVRP, particularly in relation to the tree cover LP from Section 3. Moreover, we mention several open questions.

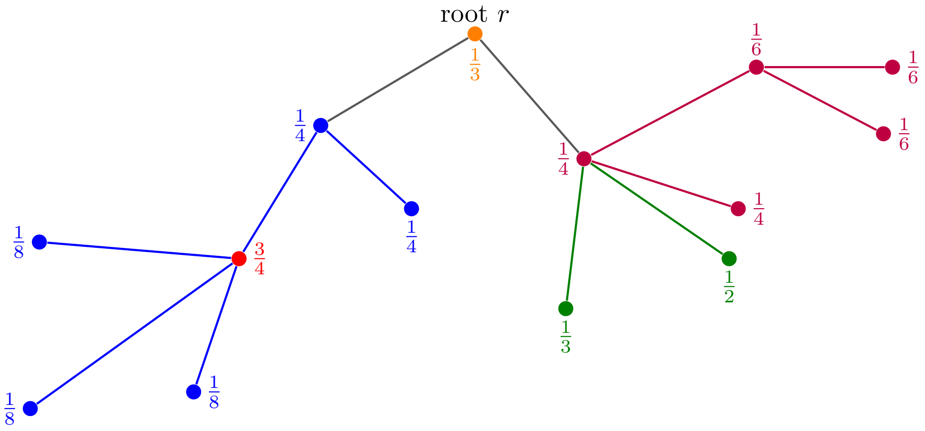

Figure 1: Example of the tree splitting procedure. The number next to a vertex v is its demand $b(v)$. The vertices in red, green, purple, blue, and orange show the sets $R_1, R_2, \dots, R_5$, respectively. The edges of a Steiner tree T_i are shown in the same color as its terminal set R_i.

2 Tree Splitting

In the following we will call a set U of vertices *large* if $b(U) := \sum_{u \in U} b(u) > 1$ and *small* otherwise. A common and useful technique for dealing with capacities in facility location and vehicle routing problems is to cluster vertices into clusters with demands between $1/2$ and 1 (see e.g. [17, 16, 12, 24]). By making sure that the demand in each cluster is at least $1/2$, we can guarantee that we have at most twice as many clusters as necessary. This idea can be used to prove the following lemma. See Figure 1 for an illustration.

Lemma 2 (Tree Splitting)**.** *Let $T = (V, E)$ be a tree and $b : V \to [0, 1]$ some vertex demands with $b(V) > 1$, i.e. V is large. Then we can partition V into $k \le 2b(V)$ many small sets $R_1, \dots, R_k$ and find edge-disjoint connected subgraphs $T_1, \dots, T_k$ of T such that $R_i \subseteq V(T_i)$, i.e. T_i is a Steiner tree with terminal set R_i, for all i. Moreover, this can be done in linear time.*

Proof. Pick an arbitrary root r for T. Then we perform the following splitting-off procedure (similar to Algorithm A in [17]).

As long as the vertex set $V(T)$ of the tree T remains large, we iterate the following. Let v be maximally far away from r with the property that $V(T_v)$ is large (in the sense that no proper subtree is large), where T_v is the subtree rooted at v. Let $w_1, \dots, w_l$ be the children of v. Since $b(V(T_v)) = b(v) + \sum_{i=1}^{l} b(V(T_{w_l}))$, we must have that $b(v) \ge 1/2$ or there exists a set $N \subseteq \{1, \dots, l\}$ with $\sum_{i \in N} b(V(T_{w_i})) \in [1/2, 1]$. In the first case we split off a singleton tree $(\{v\}, \emptyset)$ covering the vertex v and replace v in T by a Steiner vertex, i.e. we set its demand to zero. In the second case we split off a tree covering all vertices contained in the subtrees T_{w_i} for $i \in N$; the Steiner tree for this set of terminals contains v as a Steiner vertex and for $i \in N$ contains the edge $\{v, w_i\}$ and the subtree T_{w_i}. Thus we then remove these subtrees from T.

Let $T_1, \dots, T_{k-1}$ be the Steiner trees split off during this algorithm and let T_k be the remaining tree. Moreover, let $R_1, \dots, R_k$ be the respective terminal sets of these Steiner trees. Then we know that $b(R_i) \ge 1/2$ for all $i \le k-2$ and $b(R_{k-1}) + b(R_k) \ge 1$. Thus $2b(V) = 2\sum_{i=1}^{k} b(R_i) \ge k$.

Finally, to carry this out in linear time one may proceed as follows. We consider the vertices of the tree in a bottom-up order, starting at the leaves. We compute the weight of the subtree rooted at a vertex by summing up the demands of the vertex itself and the subtrees rooted at its children, which have been considered before. We continue this process until we find a vertex v with $V(T_v)$ large. The splitting step requires linear time in the number of vertices that are permanently removed from the tree. We then update the demand of the subtree rooted at v by subtracting the demand $b(R_i)$ covered by the tree T_i we split off and continue. This will compute all (R_i, T_i) in linear time. □

As a corollary, we get a simple construction which turns any forest F in G into a solution to the capacitated cycle covering problem. For an edge set F, we denote by $\mathcal{C}(F)$ the collection of vertex sets of the connected components of (V, F).

Lemma 3. *Let (V, F) be a forest. Then we can compute in linear time a feasible solution $C_1, \dots, C_k$ to the CCCP with cost bounded by*

$$2\ell(F) + \gamma \cdot \sum_{A \in \mathcal{C}(F)} u(A) \tag{1}$$

where $\ell(F) := \sum_{e \in F} \ell(e)$ and $u : 2^V \to \mathbb{R}_{\geq 0}$ is given by

$$u(A) := \begin{cases} 1 & \text{if } A \text{ is small,} \\ 2b(A) & \text{if } A \text{ is large.} \end{cases} \tag{2}$$

Proof. We first apply Lemma 2 to all large connected components of F. Together with the remaining small connected components, this yields a partition of V into k small sets $R_1, \dots, R_k$ and Steiner trees $T_1, \dots, T_k$ with terminal sets $R_1, \dots, R_k$ respectively, where $k \leq \sum_{A \in \mathcal{C}(F)} u(A)$. Then we turn each Steiner tree T_i with terminal set R_i into a cycle C_i with vertex set R_i and $\ell(C_i) \leq 2\ell(T_i)$. This is accomplished by the standard technique of ordering the elements of R_i as they appear in a depth-first search of T_i. Equivalently, one can double all edges of T_i, find an Eulerian walk, and shortcut this walk to a cycle on R_i. Shortcutting does not increase the length since ℓ is metric. □

Thus in the following sections we will discuss how to find a forest F such that (1) is at most $(2 + 2/7)$ times the cost of an optimum capacitated cycle cover.

3 The Tree Cover LP

To obtain a lower bound on the cost of an optimum solution to the CCCP, we use the following linear program.

$$\begin{aligned} \min \quad & \ell(x) + \gamma(|V| - x(E)) \\ \text{s.t.} \quad & x(E[A]) \leq |A| - \max\{1, b(A)\} \quad \forall \emptyset \neq A \subseteq V, \\ & x \geq 0, \end{aligned} \tag{3}$$

where $\ell(x) := \sum_{e \in E} x_e \ell(e)$, $x(E) := \sum_{e \in E} x_e$, and $E[A]$ denotes the set of edges in E that have both endpoints in A.

Note that LP (3) is rather a relaxation of a tree covering problem than of capacitated cycle covering: integral solutions are edge sets of forests in which every connected component contains vertices of total demand at most 1. Nonetheless, it provides a lower bound for the cost of an optimum CCCP solution because every feasible solution to the CCCP contains such a forest. Hence we get the following.

Lemma 4. *Let (G, ℓ, b, γ) be an instance of the CCCP. Then the optimum value of the LP (3) is a lower bound on the cost of an optimum solution of the CCCP.*

Proof. Let $C_1, \dots, C_k$ be an optimum solution of the CCCP. Then we can obtain a spanning forest (V, F) with k connected components and $\ell(F) \leq \sum_{i=1}^k \ell(C_i)$ by removing an arbitrary edge from each cycle. We claim that the incidence vector $x = \chi^F$ of F is a feasible solution to (3).

For every vertex set $\emptyset \neq A \subseteq V$, we have $x(E(A)) \leq |A| - 1$ because (V, F) is a forest. Moreover, $x(E(A)) \leq |A| - b(A)$ since the subgraph $(A, F[A])$ induced by A has at least $b(A)$ connected components because every connected component of (V, F) contains vertices of total demand at most 1. Finally, observe that

$$\ell(x) + \gamma(|V| - x(E)) = \ell(x) + \gamma k \leq \sum_{i=1}^k \ell(C_i) + \gamma k.$$

□

We would like to remark that the tree cover LP (3) is closely related to a natural LP relaxation of the CVRP. We will return to this connection in Section 7.

In the remaining part of this section we explain how one can solve the tree cover LP (3) by a greedy algorithm. The key insight for proving this is that (3) is equivalent to optimizing over a polymatroid. See Chapter 44 of [20] for an introduction to polymatroids and the polymatroid greedy algorithm.

Lemma 5. *Let P be the set of feasible solutions to the LP (3). Then*

$$P = \left\{ x \in \mathbb{R}^E \;\middle|\; \begin{array}{ll} x(F) & \leq r(F) \qquad \forall F \subseteq E, \\ x & \geq 0 \end{array} \right\}$$

where $r(F) \coloneqq \sum_{A \in \mathcal{C}(F)} (|A| - \max\{1, b(A)\})$. Moreover, r is monotone, submodular, and satisfies $r(\emptyset) = 0$. Thus P is a polymatroid.

Proof. First, observe that P is indeed a description of the feasible solutions of (3). Clearly, the constraints in P include all constraints in the LP. For the other direction, note that for any $F \subseteq E$ and any feasible solution x to the LP, we have

$$x(F) \leq \sum_{A \in \mathcal{C}(F)} x(E[A]) \leq \sum_{A \in \mathcal{C}(F)} (|A| - \max\{1, b(A)\}) = r(F).$$

The set $\mathcal{C}(\emptyset)$ contains all singletons $\{v\}$ with $v \in V$. Using $b(v) \leq 1$ for all $v \in V$ this implies $r(\emptyset) = \sum_{v \in V} (1 - \max\{1, b(v)\}) = 0$. Next, we show that r is monotone. Let $F \subseteq E$ be arbitrary and $e \in E \setminus F$. If $\mathcal{C}(F \cup \{e\}) = \mathcal{C}(F)$, we have $r(F \cup \{e\}) = r(F)$. Otherwise, let $A_1, A_2 \in \mathcal{C}(F)$ be the two components of F joined by e. Then

$$\begin{aligned} r(F \cup \{e\}) - r(F) &= |A_1 \cup A_2| - \max\{1, b(A_1 \cup A_2)\} - |A_1| + \max\{1, b(A_1)\} - |A_2| + \max\{1, b(A_2)\} \\ &= \max\{1, b(A_1)\} + \max\{1, b(A_2)\} - \max\{1, b(A_1) + b(A_2)\} \qquad (4) \\ &\geq 0. \end{aligned}$$

It remains to show that r is submodular. To this end let $F' \subseteq F \subseteq E$ be arbitrary and $e \in E \setminus F$. We need to show that

$$r(F' \cup \{e\}) - r(F') \geq r(F \cup \{e\}) - r(F). \qquad (5)$$

If e does not join two different connected components of (V, F), the right-hand side of (5) is 0. Then (5) follows from the monotonicity of r. Otherwise, let $A_1, A_2 \in \mathcal{C}(F)$ be the two components of F joined by e. Since $F' \subseteq F$, the edge e also connects two different connected components of (V, F'). Let $A'_1, A'_2 \in \mathcal{C}(F')$ be the the vertex sets of these components. We may assume that $A'_1 \subseteq A_1$ and $A'_2 \subseteq A_2$ since $F' \subseteq F$. Like in (4) we get

$$r(F \cup \{e\}) - r(F) = \max\{1, b(A_1)\} + \max\{1, b(A_2)\} - \max\{1, b(A_1) + b(A_2)\}$$

and

$$r(F' \cup \{e\}) - r(F') = \max\{1, b(A'_1)\} + \max\{1, b(A'_2)\} - \max\{1, b(A'_1) + b(A'_2)\},$$

where $b(A'_1) \leq b(A_1)$ and $b(A'_2) \leq b(A_2)$. So (5) reduces to the observation that the expression

$$\max\{1, x\} + \max\{1, y\} - \max\{1, x + y\}$$

is non-increasing in x and y for $x, y \geq 0$. □

Algorithm 1 formally describes the polymatroid greedy algorithm for solving (3).

Algorithm 1: Polymatroid greedy algorithm for the tree cover LP

```
Input: An instance $(G, \ell, b, \gamma)$ of the CCCP with $G = (V, E)$.
Output: An optimum solution $x$ to the linear program (3).
Let $x_e \coloneqq 0$ for all $e \in E$.
Let $e_1, \dots, e_m \in E$ be the edges with $\ell(e_i) < \gamma$ sorted such that $\ell(e_1) \leq \dots \leq \ell(e_m)$.
Let $\mathcal{C} \coloneqq \{\{v\} \mid v \in V\}$ .
for $i \coloneqq 1, \dots, m$ do
    if $e_i$ connects two distinct $C, C' \in \mathcal{C}$ then
        $\mathcal{C} \coloneqq \mathcal{C} \setminus \{C, C'\} \cup \{C \cup C'\}$
        $x_{e_i} \coloneqq \begin{cases} 1 & \text{if } C \cup C' \text{ is small} \\ 1 - b(C) & \text{if } C \text{ is small and } C' \text{ is large} \\ 1 - b(C') & \text{if } C' \text{ is small and } C \text{ is large} \\ 2 - b(C) - b(C') & \text{if } C, C' \text{ are small, but } C \cup C' \text{ is large} \\ 0 & \text{else, i.e. } C, C' \text{ large.} \end{cases}$
return $x$.
```

Note that $\mathcal{C}$ remains a partition of the vertex set. At the end of iteration i it contains the vertex sets of the connected components of $(V, \{e_1, \ldots, e_i\})$. Moreover, the support $\{e \in E : x_e > 0\}$ of the returned LP solution x is the edge set of a forest (by the condition in line 5). This structure will be useful in the next section, where we analyze an algorithm for rounding x to an integral vector.

Lemma 6. *Algorithm 1 computes an optimum solution of the LP (3).*

Proof. By Lemma 5 we know that LP (3) optimizes over a polymatroid. Thus the polymatroid greedy algorithm which sets $x_{e_i} := r(\{e_1, \ldots, e_i\}) - r(\{e_1, \ldots, e_{i-1}\})$ for every $i \leq m$ produces an optimal solution. We show that Algorithm 1 outputs the same solution. At the beginning of iteration i of Algorithm 1, the set $\mathcal{C}$ contains the vertex sets of the connected components of $(V, \{e_1, \ldots, e_{i-1}\})$. If an edge e_i joins two different connected components C, C' of $(V, \{e_1, \ldots, e_{i-1}\})$, we have

$$\begin{aligned}
&r(\{e_1, \ldots, e_i\}) - r(\{e_1, \ldots, e_{i-1}\}) \\
&= (|C \cup C'| - \max\{1, b(C \cup C')\}) - (|C| - \max\{1, b(C)\} + |C'| - \max\{1, b(C')\}) \\
&= \max\{1, b(C)\} + \max\{1, b(C')\} - \max\{1, b(C \cup C')\}
\end{aligned}$$

and Algorithm 1 sets x_{e_i} to exactly those values. Otherwise, e_i does not join two different connected components. So we have $r(\{e_1, \ldots, e_i\}) = r(\{e_1, \ldots, e_{i-1}\})$ and Algorithm 1 sets $x_{e_i} := 0$ in line 1. □

4 Randomized Rounding

We will now show how we can round the fractional solution x generated by Algorithm 1 to a forest F while bounding the cost (1) of the resulting CCCP solution. More precisely, we will prove the following theorem.

Theorem 7 (Randomized rounding)**.** *Let x be a solution of the tree cover LP (3) computed by Algorithm 1. Define a random edge set $F \subseteq E$ by independently picking each edge e with probability* $\min\{1, (1 + 1/7)x_e\}$. *Then*

$$\mathbb{E}\left[\sum_{A \in \mathcal{C}(F)} u(A)\right] \leq \left(2 + \frac{2}{7}\right)(|V| - x(E)),$$

where u is defined by (2), and $\mathbb{E}[2\ell(F)] \leq (2 + 2/7)\, \ell(x)$.

Note that this implies that the total cost (1) is at most $2 + 2/7$ times the objective value $\ell(x) + \gamma(|V| - x(E))$ of our optimum LP solution x. The scaling factor $1 + 1/7$ on the probabilities x_e is chosen to decrease the expected number of components of (V, F) (while increasing the expected length) such that we lose the same factor in both cost terms wrt. the LP. By Lemma 3 and Lemma 4, Theorem 7 yields a randomized $(2 + 2/7)$-approximation algorithm for the CCCP.

In the rest of this section we prove Theorem 7. We may assume wlog. that $(V, \{e_1, \ldots, e_m\})$ is connected; otherwise we prove the statement for each connected component. Let E' be the set of edges e_i for which the condition in line 5 of Algorithm 1 was fulfilled. Every such edge $e_i = \{v, w\} \in E'$ connected two sets $C, C' \in \mathcal{C}$ in iteration i of Algorithm 1. Let $C^v_{e_i} \in \{C, C'\}$ be the set containing v and let $C^w_{e_i} \in \{C, C'\}$ be the other set (containing w). By construction of $\mathcal{C}$ in Algorithm 1, (V, E') is a spanning tree. Thus, F is always a forest. Moreover, the subgraphs of $(V, \{e_1, \ldots, e_{i-1}\} \cap E')$ induced by $C^v_{e_i}$ and $C^w_{e_i}$ are connected. The structure of x is illustrated in Figure 2.

Lemma 8. *For every set $F \subseteq E'$, we have*

$$\sum_{A \in \mathcal{C}(F)} u(A) \leq 2 \cdot (|V| - x(E)) + \sum_{e \in E' \setminus F} \sum_{u \in e} \max\{1 - 2b(C^u_e), 0\}.$$

Proof. We first consider the case $\mathcal{C}(F) = \{V\}$ and hence $F = E'$. Then we have $x(E) \leq |V| - \max\{1, b(V)\}$ since x is a feasible solution to (3) and hence $u(V) \leq \max\{1, 2b(V)\} \leq 2(|V| - x(E))$. Now assume

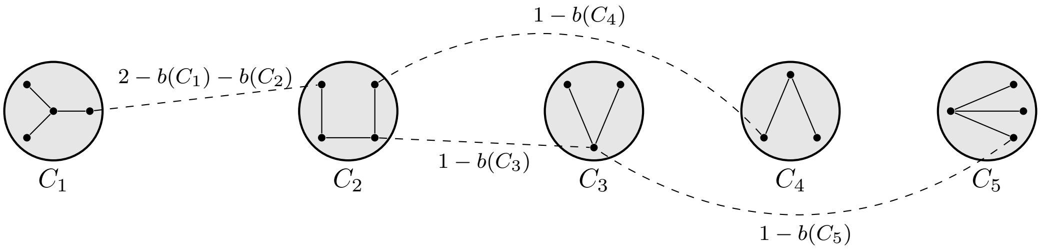

Figure 2: The structure of the connected components of supp(x). Each gray component represents a small tight set inside of which x is the incidence vector of a spanning tree. They are connected by fractional edges (shown as dashed edges). The numbers next to the dashed edges show possible values x_e in an extreme point solution of the tree cover LP; here the vertex sets of the small Gray components are denoted by $C_1, \dots, C_5$.

$\mathcal{C}(F) \neq \{V\}$ and compute

$$\begin{aligned}\sum_{A \in \mathcal{C}(F)} u(A) &= \sum_{\substack{A \in \mathcal{C}(F)\\ A \text{ large}}} 2b(A) + \sum_{\substack{A \in \mathcal{C}(F)\\ A \text{ small}}} 1\\ &= 2b(V) + \sum_{\substack{A \in \mathcal{C}(F)\\ A \text{ small}}} (1 - 2b(A))\\ &\leq 2b(V) + \sum_{A \in \mathcal{C}(F)} \max\{1 - 2b(A), 0\}\\ &\leq 2 \cdot (|V| - x(E)) + \sum_{A \in \mathcal{C}(F)} \max\{1 - 2b(A), 0\},\end{aligned} \tag{6}$$

where we used in the last inequality that x is a feasible solution to (3).

Recall that $\mathcal{C}(F) \neq \{V\}$ and (V, E') is a spanning tree. Consider some $A \in \mathcal{C}(F)$ and let i be minimum such that $e_i = \{v, w\} \in \delta(A) \cap E'$, where wlog. $v \in A$ and $\delta(A)$ is used to denote the set of edges which have exactly one endpoint in A. So $v \in A \cap C^v_{e_i} \neq \emptyset$. Since the subgraphs of $(V, \{e_1, \dots, e_{i-1}\} \cap E')$ induced by $C^v_{e_i}$ and $C^w_{e_i}$ are connected and i was chosen minimal, we have $C^v_{e_i} \subseteq A$. Hence, $\max\{1 - 2b(C^v_{e_i}), 0\} \geq \max\{1 - 2b(A), 0\}$. Note that $e_i \in E' \setminus F$ because $e_i \in \delta(A)$ and $A \in \mathcal{C}(F)$. Thus,

$$\begin{aligned}\sum_{A \in \mathcal{C}(F)} \max\{1 - 2b(A), 0\} &\leq \sum_{A \in \mathcal{C}(F)} \sum_{e \in E' \setminus F} \sum_{\substack{u \in e\\ C^u_e \subseteq A}} \max\{1 - 2b(A), 0\}\\ &\leq \sum_{e \in E' \setminus F} \sum_{u \in e} \max\{1 - 2b(C^u_e), 0\},\end{aligned}$$

because $\mathcal{C}(F)$ is a partition of V. Together with (6) this completes the proof. □

Lemma 9. *Let x be a solution of the tree cover LP (3) computed by Algorithm 1. Define a random edge set $F \subseteq E$ by independently picking each edge e with probability $\min\{1, (1 + 1/7)x_e\}$. Then*

$$\mathbb{E}\left[\sum_{e \in E' \setminus F} \sum_{u \in e} \max\{1 - 2b(C^u_e), 0\}\right] \leq 2/7 \cdot (|V| - x(E)).$$

Proof. We consider an edge $e \in E'$ and a vertex $u \in e$. If $x_e < 1$, by the definition of x_e in Algorithm 1 we have $x_e \geq 1 - b(C^u_e)$ and therefore

$$\begin{aligned}&\mathbb{P}[e \notin F] \cdot \max\{1 - 2b(C^u_e), 0\}\\ =\ &\max\{1 - (1 + 1/7) \cdot x_e,\ 0\} \cdot \max\{1 - 2b(C^u_e), 0\}\\ \leq\ &\max\{1 - (1 + 1/7) \cdot (1 - b(C^u_e)),\ 0\} \cdot \max\{1 - 2b(C^u_e), 0\}\\ \leq\ &2/7 \cdot b(C^u_e).\end{aligned}$$

Hence,

$$\mathbb{E}\left[\sum_{e\in E'\setminus F}\sum_{u\in e}\max\{1-2b(C_e^u),0\}\right] = \sum_{e\in E':x_e<1}\mathbb{P}[e\notin F]\cdot\sum_{u\in e}\max\{1-2b(C_e^u),0\} \le \sum_{e\in E':x_e<1}\;\sum_{u\in e:C_e^u\text{ small}} 2/7\cdot b(C_e^u). \tag{7}$$

Let $1\le i<j\le m$ with $e_i=\{u,v\}, e_j=\{u',v'\}\in E'$ with $x_{e_i},x_{e_j}<1$. We claim that if the vertex sets $C_{e_i}^u$ and $C_{e_j}^{u'}$ are both small, then they are disjoint. In iteration i of Algorithm 1, we merge $C_{e_i}^u$ and $C_{e_i}^v$ into a single component $C_{e_i}^u\cup C_{e_i}^v$. This new component must be large because $x_{e_i}<1$. During the course of the algorithm we only merge components of the partition $\mathcal{C}$ of V. Therefore either $C_{e_i}^u$ and $C_{e_j}^{u'}$ are disjoint, or $C_{e_i}^u\cup C_{e_i}^v\subseteq C_{e_j}^{u'}$ which implies that $C_{e_j}^{u'}$ is large. Hence,

$$\sum_{e\in E':x_e<1}\;\sum_{u\in e:C_e^u\text{ small}} b(C_e^u)\le b(V)\le |V|-x(E),$$

where $b(V)\le|V|-x(E)$ holds because x is a feasible solution to (3). Together with (7) this completes the proof. □

The bound $\mathbb{E}[2\ell(F)]\le(2+2/7)\,\ell(x)$ follows directly from the linearity of expectation. Hence, Lemma 8 and Lemma 9 imply Theorem 7.

5 A Fast and Deterministic Algorithm

In this section we show how one can derandomize our $(2+2/7)$-approximation algorithm. Algorithm 2 formally describes the computation of the forest (V,F). The partition $\mathcal{C}$ is updated exactly as in Algorithm 1. However, now we do not compute the value x_{e_i} but instead directly round it in a deterministic way (lines 7 – 10).

Algorithm 2: Solving the relaxation and rounding deterministically

```
   Input: An instance $(G,\ell,b,\gamma)$ of the CCCP with $G=(V,E)$.
   Output: A forest $(V,F)$.
 1 Let $F:=\emptyset$.
 2 Let $e_1,\dots,e_m\in E$ be the edges with $\ell(e_i)<\gamma$ sorted such that $\ell(e_1)\le\dots\le\ell(e_m)$.
 3 Let $\mathcal{C}:=\{\{v\}\mid v\in V\}$ .
 4 for $i:=1,\dots,m$ do
 5     if $e_i$ connects two distinct $C,C'\in\mathcal{C}$ then
 6         $\mathcal{C}:=\mathcal{C}\setminus\{C,C'\}\cup\{C\cup C'\}$
 7         if $C\cup C'$ is small or $\min\{b(C),b(C')\}=0$ then
 8             $F:=F\cup\{e_i\}$
 9         if $\gamma\cdot(\max\{1-2b(C),0\}+\max\{1-2b(C'),0\})>2\ell(e_i)$ then
10             $F:=F\cup\{e_i\}$
11 return $(V,F)$.
```

The motivation for lines 7 – 10 comes directly from the proof of Theorem 7. There we used that we can sample an edge set $F\subseteq E'$ such that

$$\mathbb{E}\left[2\ell(F)+2\cdot(|V|-x(E))+\sum_{e\in E'\setminus F}\sum_{u\in e}\max\{1-2b(C_e^u),0\}\right]\le\left(2+\frac{2}{7}\right)\cdot\mathrm{LP}$$

which provided an upper bound for the cost of the CCCP solution constructed from F. Lines 7 – 10 in Algorithm 2 are chosen to minimize this quantity deterministically. This allows us to obtain the following.

Lemma 10. *Algorithm 2 computes a forest (V,F) with*

$$2\ell(F)+\gamma\cdot\sum_{A\in\mathcal{C}(F)}u(A)\le\left(2+\frac{2}{7}\right)\cdot\mathrm{LP}, \tag{8}$$

where LP *denotes the value of* (3).

Proof. Note that the partition $\mathcal{C}$ in iteration i of Algorithm 2 is the same as in iteration i of Algorithm 1 assuming wlog. that the edges are sorted in the same order in both algorithms. Hence, we apply lines 7–10 of Algorithm 2 precisely for those edges e_i for which we set x_{e_i} in line 7 of Algorithm 1. Once again let E' be the edges e_i which fulfill the condition in line 5 as in Section 4. For each $e_i \in E'$ and $u \in e_i$ we also define $C^u_{e_i} \subseteq V$ as before: it is the set $C^u_{e_i} \in \{C, C'\}$ in iteration i which contains u.

Let x be the (fractional) output of Algorithm 1 and let (V, F) be the output of Algorithm 2. Then by comparing the two algorithms, we observe that an edge e is always included in F if $x_e = 1$ and it is never included in F if $x_e = 0$. Moreover, F minimizes

$$\sum_{e\in F} 2\ell(e) + \sum_{e\in E'\setminus F} \sum_{u\in e} \gamma \cdot \max\{1 - 2b(C^u_e), 0\} \tag{9}$$

among all sets F with $\{e \in E : x_e = 1\} \subseteq F \subseteq \{e \in E : x_e > 0\}$. This is because for any e with $x_e \in (0, 1)$, we will contribute either $2\ell(e)$ or $\sum_{u\in e} \gamma \cdot \max\{1 - 2b(C^u_e), 0\}$ to (9) depending on whether e was included in F by the algorithm or not. The decision in line 9 is specifically made to minimize this contribution.

Finally, by Lemma 9 there exists such an edge set F where (9) is at most

$$\left(2 + \frac{2}{7}\right) \ell(x) + \frac{2}{7} \cdot \gamma \cdot (|V| - x(E)).$$

Hence, also the edge set F computed by Algorithm 2 fulfills this bound. But then by Lemma 8 we have

$$\begin{aligned} 2\ell(F) + \gamma \cdot \sum_{A\in\mathcal{C}(F)} u(A) &\le 2\ell(F) + 2\gamma \cdot (|V| - x(E)) + \sum_{e\in E'\setminus F} \sum_{u\in e} \gamma \cdot \max\{1 - 2b(C^u_e), 0\} \\ &\le \left(2 + \frac{2}{7}\right) (\ell(x) + \gamma \cdot (|V| - x(E))) \\ &\le \left(2 + \frac{2}{7}\right) \text{LP}. \end{aligned}$$

□

Now it remains to combine the various lemmas and show that we can carry out everything in $O(n^2)$ time as claimed.

Proof of Theorem 1. First we run Algorithm 2 to compute a forest (V, F) with

$$2\ell(F) + \gamma \cdot \sum_{A\in\mathcal{C}(F)} u(A) \le \left(2 + \frac{2}{7}\right) \cdot \text{LP},$$

as established by Lemma 10. If $\mathcal{C}$ is maintained as a union-find data structure, this will take $O((n + m) \log n + m)$ time, where $m = O(n^2)$. However, note that the only edges which connect distinct components, i.e. satisfy the condition of line 5, are edges which appear in a minimum spanning tree. So we may precompute this MST in $O(n^2)$ time and then simply work on the $O(n)$ many edges in this tree. This reduces the total amount of time to $O(n^2 + n \log n) = O(n^2)$.

Finally, we know from Lemma 4 that LP $\le$ OPT and by Lemma 3 we can turn the forest (V, F) into a capacitated cycle cover with cost at most

$$2\ell(F) + \gamma \cdot \sum_{A\in\mathcal{C}(F)} u(A).$$

Since this last step takes $O(n)$ time, we are done. □

6 Lower Bounds

In this section we show that the approximation ratio of Algorithm 2 followed by the Algorithm from Lemma 3 is at least $(2+2/7)$, i.e. we show that our analysis of the deterministic algorithm in the preceding sections is tight. Moreover, we show that the cost of an optimum solution to the CCCP might be more than twice the value of the tree cover LP (3).

Theorem 11. *For any $\epsilon > 0$ there is a CCCP instance where Algorithm 2 computes an edge set $F \subseteq E$, such that there is no capacitated cycle cover $C_1, \ldots, C_k$ with cost at most $(2 + 2/7 - \epsilon)\mathrm{LP}$ where $V(C_i)$ is connected in (V, F) for all $i \in \{1, \ldots, k\}$.*

Proof. For $n \in \mathbb{N}$ with $n \geq 4$, let $G = (V, E)$ be the complete graph on the vertices $v_1, \ldots, v_n$ with the metric ℓ on V given by $\ell(v_i, v_j) := \frac{1}{4}|i - j|$, i.e. (G, ℓ) is the metric closure of a path. Assign uniform demands of $b(v) := 1/4$ to every vertex v and let $\gamma := 1$. Then we observe that $\mathrm{LP}(G, \ell, b, \gamma) = \frac{7}{16}n$. See Figure 3.

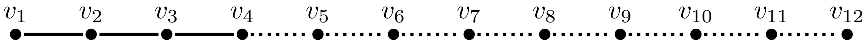

Figure 3: An optimum solution to the tree cover LP (3) for instance from the proof of Theorem 11 for $n = 12$. For every solid edge e we have $x_e = 1$ and for every dotted edge e we have $x_e = 3/4$.

But now consider what Algorithm 2 does on this instance. Assume that the edges are sorted such that $e_i = \{v_i, v_{i+1}\}$ for all $i \in \{1, \ldots, n - 1\}$. The algorithm will then buy the edges e_1 to e_3. But it will not buy any other edge as

$$\gamma \max\{1 - 2b(v_{i+1}), 0\} = \frac{1}{2} = 2\ell(\{v_i, v_{i+1}\})$$

for all $i \in \{1, \ldots, n - 1\}$. So the condition in line 9 is never satisfied except for the first three iterations of the loop. Hence, any CCCP solution which is “contained” in the connected components of F (i.e. it does not contain a cycle C_i where $V(C_i)$ is not connected in (V, F)), must contain at least $n - 4$ singleton cycles.

Finally, we conclude that any such CCCP solution has a cost of at least

$$n - 4 \;=\; \frac{n-4}{\frac{7}{16}n}\mathrm{LP} \;\geq\; \left(\frac{16}{7} - \epsilon\right)\mathrm{LP} \;=\; \left(2 + \frac{2}{7} - \epsilon\right)\mathrm{LP}$$

for n large enough. □

We remark that although Theorem 11 shows that our analysis of Algorithm 2 followed by the Algorithm from Lemma 3 is tight, it might be that the analysis of our randomized rounding algorithm is not.

We now show that the cost of an optimum solution to the CCCP might be more than twice the value of the tree cover LP (3). We define

$$\rho := \sup\left\{ \frac{\mathrm{OPT}(\mathcal{I})}{\mathrm{LP}(\mathcal{I})} \;\middle|\; \mathcal{I} \text{ is a CCCP instance} \right\}.$$

Here we use $\mathrm{OPT}(\mathcal{I})$ to refer to the minimum cost of a CCCP solution on the instance $\mathcal{I} = (G, \ell, b, \gamma)$. Similarly, $\mathrm{LP}(\mathcal{I})$ refers to the solution value of the tree cover LP (3) for the instance $\mathcal{I}$.

Theorem 12. $\rho \geq 2 + \frac{62}{11745} > 2.005$.

To prove Theorem 12 we use the following lemma that can be proven by an argument similar to Goemans [13], and Carr and Vempala [7].

Lemma 13. *Let $G = (V, E)$ a complete graph and $b : V \to [0, 1]$ some vertex demands. Moreover, let x be a feasible solution to the tree cover LP (3) such that the support of x is the edge set of a spanning tree T. Then there are weights $\lambda_1, \ldots, \lambda_k > 0$, small sets $R_1, \ldots, R_k \subseteq V$ and trees $T_1, \ldots, T_k$ in T such that $R_i \subseteq V(T_i)$ for all i and*

- $\sum_{i=1}^{k} \lambda_i \leq \rho(|V| - x(E))$,
- $\sum_{i: e \in T_i} \lambda_i \leq \frac{\rho}{2} x_e$ *for every $e \in E(T)$, and*
- $\sum_{i: v \in R_i} \lambda_i \geq 1$ *for every $v \in V$.*

Proof. Let $(R_i, T_i)_{i=1}^N$ enumerate all pairs of small sets $R_i \subseteq V$ and trees T_i in T with $R_i \subseteq V(T_i)$. Assume for a contradiction that the conclusion of the lemma is false. Then the LP

$$\begin{aligned} \min_{(\lambda_i)_{i=1}^N, \mu} \quad & \mu \\ \text{s.t.} \quad & \sum_{i=1}^N \lambda_i \leq \mu(|V| - x(E)), \\ & \sum_{i: e \in T_i} \lambda_i \leq \frac{\mu}{2} x_e \quad \forall e \in E(T), \\ & \sum_{i: v \in R_i} \lambda_i \geq 1 \qquad \forall v \in V, \\ & \lambda, \mu \geq 0 \end{aligned}$$

has some optimal value $\mu^* > \rho$. Since this LP is both bounded and feasible, it follows from strong LP duality that the dual LP

$$\max_{\gamma, \ell', (\beta_v)_{v \in V}} \quad \sum_{v \in V} \beta_v \tag{10a}$$

$$\text{s.t.} \quad \sum_{e \in E(T)} \frac{1}{2} \ell'(e) x_e + \gamma(|V| - x(E)) \leq 1, \tag{10b}$$

$$\sum_{e \in E(T_i)} \ell'(e) + \gamma \geq \sum_{v \in R_i} \beta_v \quad \forall i \in [N], \tag{10c}$$

$$\gamma, \ell', \beta \geq 0 \tag{10d}$$

has some solution (γ, ℓ', β) with $\sum_{v \in V} \beta_v = \mu^* > \rho$.

Consider now the CCCP instance which is defined on the complete graph on V with demands b, opening cost γ being the value obtained from the dual LP and edge lengths ℓ being the metric closure of $\frac{1}{2} \cdot \ell'$. We want to show that on this particular instance, the gap between the tree cover LP and any CCCP solution is at least $\mu^* > \rho$ which is a contradiction.

Note first that constraint (10b) implies directly that x is a tree cover solution for this new instance with cost at most 1. So now consider any capacitated cycle cover $C_1, \ldots, C_k$. Since ℓ is the metric closure of the $\frac{1}{2} \cdot \ell'$, which is defined on the edges of the tree T, we can find indices $i_1, \ldots, i_k$ such that $R_{i_j} = V(C_j)$ and $\ell(C_j) \geq 2\ell(T_{i_j}) = \ell'(T_{i_j})$ for all $j \leq k$. This is achieved by "projecting" the cycles into T, i.e. replacing each edge by a sequence of edges in T of the same length. But then we can lower bound the cost

$$\begin{aligned} \sum_{j=1}^k \ell(C_j) + \gamma k &\geq \sum_{j=1}^k (\ell'(T_{i_j}) + \gamma) \\ &\geq \sum_{j=1}^k \sum_{v \in R_{i_j}} \beta_v \\ &= \mu^* \end{aligned}$$

by using constraint (10c). But since $\mu^* > \rho$, this contradicts the definition of ρ. □

Proof of Theorem 12. We consider the family of tree cover LP solutions depicted in Figure 4. More precisely, for any $h \geq 2$ we let

$$V := \{r\} \cup \{v_l \mid l \in [h]\} \cup \{w_{l,j} \mid l \in [h], j \in [16]\}$$

where we use the notation $[h] = \{1, \ldots, h\}$ and define

$$x_e := \begin{cases} 1 & \text{if } e = \{r, v_l\} \text{ for } l \in [h], \\ \frac{22}{23} & \text{if } e = \{v_l, w_{l,j}\} \text{ for } l \in [h], j \in [16], \\ 0 & \text{otherwise}, \end{cases}$$

$$b(v) := \begin{cases} \frac{1}{23} & \text{if } v = w_{l,j} \text{ for } l \in [h], j \in [16], \\ 0 & \text{otherwise}. \end{cases}$$

It is easy to check that x is indeed a feasible solution to the tree cover LP with these demands.

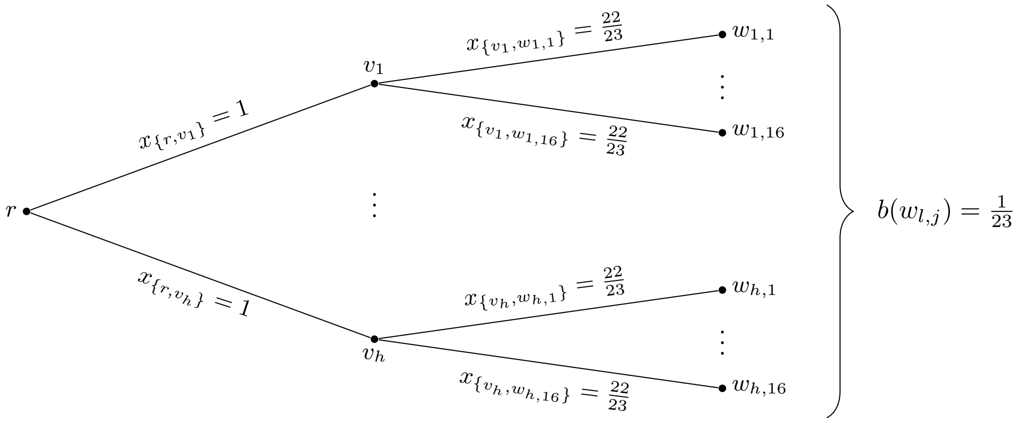
r
v_1
v_h
$x_{\{r,v_1\}} = 1$
$x_{\{r,v_h\}} = 1$
$x_{\{v_1,w_{1,1}\}} = \frac{22}{23}$
$x_{\{v_1,w_{1,16}\}} = \frac{22}{23}$
$x_{\{v_h,w_{h,1}\}} = \frac{22}{23}$
$x_{\{v_h,w_{h,16}\}} = \frac{22}{23}$
$w_{1,1}$
$w_{1,16}$
$w_{h,1}$
$w_{h,16}$
$b(w_{l,j}) = \frac{1}{23}$

Figure 4: A family of LP solutions x that together with Lemma 13 proves Theorem 12. The constants are chosen to maximize the lower bound obtained from this family of instances. In the figure edges e with $x_e > 0$ are shown.

By Lemma 13 we can now obtain weights $\lambda_1, \dots, \lambda_k > 0$, small sets $R_1, \dots, R_k \subseteq V$ and trees $T_1, \dots, T_k$ in T such that $R_i \subseteq V(T_i)$ and

- $\sum_{i=1}^k \lambda_i \leq \rho(|V| - x(E))$,
- $\sum_{i:e \in T_i} \lambda_i \leq \frac{\rho}{2} x_e$ for every $e \in E$, and
- $\sum_{i:v \in R_i} \lambda_i \geq 1$ for every $v \in V$.

Our general strategy will now be as follows. We will first compute how much demand $b(V)$ we have to cover in relation to the total weight $\sum_{i=1}^k \lambda_i$. This tells us how much demand the sets R_i should cover on average. We will then use that if ρ is small, the edges of the type $\{v_l, w_{l,j}\}$ cannot be used in all of the trees and this means that some amount weight must be on singleton sets. Since these sets are very inefficient at covering demand, we must compensate somehow by putting some weight on sets with high demand. But since the trees connecting these sets with high demand must necessarily use two edges in $\delta(r)$, the weight on these trees is bounded and this is what ultimately implies a bound on ρ.

First, compute

$$\sum_{i=1}^k \lambda_i \leq \rho(|V| - x(E)) = \rho\left(1 + \frac{16}{23}h\right)$$

and

$$\sum_{i=1}^k \lambda_i b(R_i) = \sum_{v \in V} b(v) \sum_{i:v \in R_i} \lambda_i \geq b(V) = \frac{16}{23}h.$$

This tells us that the weighted average demand of the sets R_i must be roughly $\frac{1}{\rho}$ for large h.

Next, consider one of the edges $e = \{v_l, w_{l,j}\}$ on which we have $x_e = \frac{22}{23}$. Then

$$\sum_{i:e \in E(T_i)} \lambda_i \leq \frac{\rho}{2} x_e = \frac{11}{23}\rho.$$

But note that if $|R_i| \geq 2$ for some l with $w_{l,j} \in R_i$, then the edge e must be used. However, $w_{l,j}$ still needs to be covered sufficiently often. So there must be some i such that $R_i = \{w_{l,j}\}$ and $\lambda_i \geq 1 - \frac{11}{23}\rho$. Let us define

$$\hat{\lambda}_i := \begin{cases} 1 - \frac{11}{23}\rho & \text{if } R_i = \{w_{l,j}\} \text{ for } l \in [h], j \in [16], \\ 0 & \text{otherwise.} \end{cases}$$

Then we have just shown $\hat{\lambda} \leq \lambda$.

These singleton sets are particularly inefficient at covering demand, however. They only cover

$$\sum_{i=1}^{k} \hat{\lambda}_i b(R_i) = \left(1 - \frac{11}{23}\rho\right) \frac{16}{23}h$$

demand despite the fact that

$$\sum_{i=1}^{k} \hat{\lambda}_i = \left(1 - \frac{11}{23}\rho\right) 16h.$$

This implies that the remaining demand of

$$\sum_{i=1}^{k} (\lambda_i - \hat{\lambda}_i) b(R_i) \geq \frac{16}{23}h - \left(1 - \frac{11}{23}\rho\right) \frac{16}{23}h = \frac{176}{529}\rho h$$

must be covered with sets of weight

$$\sum_{i=1}^{k} (\lambda_i - \hat{\lambda}_i) \leq \rho \left(1 + \frac{16}{23}h\right) - \left(1 - \frac{11}{23}\rho\right) 16h.$$

Now define

$$a := \sum_{i: b(R_i) \leq \frac{16}{23}} (\lambda_i - \hat{\lambda}_i),$$
$$b := \sum_{i: b(R_i) > \frac{16}{23}} (\lambda_i - \hat{\lambda}_i).$$

Then we have just shown that

$$a + b \leq \rho \left(1 + \frac{16}{23}h\right) - \left(1 - \frac{11}{23}\rho\right) 16h.$$

In addition, we can bound

$$\frac{176}{529}\rho h \leq \sum_{i=1}^{k} (\lambda_i - \hat{\lambda}_i) b(R_i) \leq \frac{16}{23}a + b$$

which together with the previous inequality yields

$$\frac{176}{529}\rho h \leq \frac{16}{23}\left(\left(1 + \frac{16}{23}h\right) - \left(1 - \frac{11}{23}\rho\right) 16h\right) + \frac{7}{23}b.$$

Lastly, note that b counts the weight on the sets R_i with $b(R_i) > \frac{16}{23}$. But for any such i, we must have some $w_{j,l}, w_{j',l'} \in R_i$ for $j \neq j'$. So in order for T_i to be connected, it must contain at least two edges in $\delta(r)$. But since any such edge e can be used at most $\frac{\rho}{2}x_e$ many times where $x_e = 1$, this implies that $b \leq \frac{1}{4}\rho h$. Thus we have

$$\frac{176}{529}\rho h \leq \frac{16}{23}\left(\left(1 + \frac{16}{23}h\right) - \left(1 - \frac{11}{23}\rho\right) 16h\right) + \frac{7}{92}\rho h$$

which can be rearranged to

$$\rho \geq \frac{23552h}{11745h + 1472}.$$

Hence, for $h \to \infty$ we obtain $\rho \geq \frac{23552}{11745} = 2 + \frac{62}{11745} > 2.005$ as desired. □

We note that the instances used in this proof almost have uniform demands. In various vehicle routing problems these instances tend to be easier since we do not have to deal with the issue of not being able to pack the demands tightly. It is possible to assign some extra demands on the vertices to show that even on uniform demand instances, we have that the gap between the tree cover LP and the CCCP is strictly greater than 2.

7 Relation to the CVRP and Open Questions

The main open problem which inspired this research is the long-standing problem of improving the approximation guarantee for the CVRP. An integer programming formulation for the CVRP which features prominently in the vehicle routing literature is the following two-index formulation (named this way because there is a variable for every edge / pair of vertices).

$$\begin{aligned} \min \quad & \ell(x) \\ \text{s.t.} \quad & x(\delta(A)) \geq 2l(A) && \forall \emptyset \neq A \subseteq V \setminus \{s\}, \\ & x(\delta(v)) = 2 && \forall v \in V \setminus \{s\}, \\ & x_e \in \{0,1,2\} && \forall e \in E. \end{aligned} \tag{11}$$

Here, $l(A)$ is any valid lower bound for the number of vehicles which are required to serve the demand A. Common choices are $l(A) := b(A)$, $l(A) := \lceil b(A) \rceil$, or the true lower bound which requires solving a bin packing problem. It is easy to show that the integrality gap of the LP relaxation is unbounded for the choice $l(A) := b(A)$. On the other hand, Diarrassouba [11] recently showed that the non-linear lower bound choice $l(A) := \lceil b(A) \rceil$ makes finding an exact solution to the LP relaxation NP-hard. This makes it difficult to exploit the potential structure of extreme point solutions. However, it is still possible to solve the LP approximately by separating the rounded constraints only up to a certain constant demand.

It turns out that the optimal tour partitioning algorithm for the CVRP [1] computes a solution of cost at most 3.5 times the value of the LP relaxation of (11) for the choice $l(A) := \max\{1, b(A)\}$, i.e. the following LP (see e.g. [21]).

$$\begin{aligned} \min \quad & \ell(x) \\ \text{s.t.} \quad & x(\delta(A)) \geq 2 && \forall \emptyset \neq A \subseteq V \setminus \{s\}, \\ & x(\delta(A)) \geq 2b(A) && \forall \emptyset \neq A \subseteq V \setminus \{s\}, \\ & x(\delta(v)) = 2 && \forall v \in V \setminus \{s\}, \\ & x \geq 0. \end{aligned} \tag{12}$$

In particular, the integrality gap of (12) is at most 3.5. Moreover, this LP (12) is closely related to the tree cover LP (3). The following LP is equivalent to (12) in the sense that every feasible solution to one of the LPs is also a feasible solution for the other.

$$\begin{aligned} \min \quad & \ell(x) \\ \text{s.t.} \quad & x(E[A]) \leq |A| - \max\{1, b(A)\} && \forall \emptyset \neq A \subseteq V \setminus \{s\}, \\ & x(\delta(v)) = 2 && \forall v \in V \setminus \{s\}, \\ & x \geq 0. \end{aligned}$$

Therefore, for every feasible solution x to (12), the restriction of x to $G - s$ is a feasible solution to the tree cover LP (3).

In this paper we showed that we can round extreme point solutions of the tree cover LP efficiently to obtain $(2 + \frac{2}{7})$-approximate solutions for the CCCP. Given the close relation of the tree cover LP and LP (12), a natural question is whether related techniques can be applied to the CVRP. For the CVRP we only know of the 3.5 bound on the integrality gap of (12). In particular, the recently announced $3.5 - \epsilon$ algorithm due to Blauth et al. [5] does *not* imply a $3.5 - \epsilon$ bound on the integrality gap. We conjecture that the real integrality gap is much closer to the trivial lower bound of 2 than to 3.5. Both non-trivial lower bounds and of course any upper bound better than 3.5 would be of significant interest, also for the special case of uniform demands.

Lastly, we mention a few more open questions:

- We have shown that ρ, the gap between the tree cover LP and the CCCP, satisfies $2.005 \leq \rho \leq 2+2/7$. What is the precise value of ρ?
- Can we do any better for the CCCP if we restrict ourselves to uniform demand instances or instances in which we allow to split the demand between multiple cycles?
- Is there an LP-based $(3.5 - \epsilon)$-approximation algorithm for the CVRP?

- Given that the CCCP is in some sense a version of the CVRP with uniform opening costs of tours, is there some natural choice of non-uniform opening costs which results in a problem harder than the CCCP but which still has an approximation guarantee close to 2?

References

[1] Altinkemer, K., Gavish, B.: Heuristics for unequal weight delivery problems with a fixed error guarantee. Operations Research Letters **6**(4), 149–158 (1987)

[2] Becker, A.: A tight 4/3 approximation for capacitated vehicle routing in trees. In: Blais, E., Jansen, K., Rolim, J.D.P., Steurer, D. (eds.) Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018). Leibniz International Proceedings in Informatics (LIPIcs), vol. 116, pp. 3:1–3:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018)

[3] Becker, A., Klein, P.N., Saulpic, D.: Polynomial-time approximation schemes for k-center, k-median, and capacitated vehicle routing in bounded highway dimension. In: Azar, Y., Bast, H., Herman, G. (eds.) 26th Annual European Symposium on Algorithms (ESA 2018). Leibniz International Proceedings in Informatics (LIPIcs), vol. 112, pp. 8:1–8:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018)

[4] Becker, A., Klein, P.N., Schild, A.: A PTAS for bounded-capacity vehicle routing in planar graphs. In: Friggstad, Z., Sack, J.R., Salavatipour, M.R. (eds.) Algorithms and Data Structures. pp. 99–111. Springer International Publishing, Cham (2019)

[5] Blauth, J., Traub, V., Vygen, J.: Improving the approximation ratio for capacitated vehicle routing. arXiv:1907.08304 (2020)

[6] Bompadre, A., Dror, M., Orlin, J.B.: Improved bounds for vehicle routing solutions. Discrete Optimization **3**(4), 299–316 (2006)

[7] Carr, R., Vempala, S.: On the Held-Karp relaxation for the asymmetric and symmetric traveling salesman problems. Mathematical Programming **100**(3), 569–587 (2004)

[8] Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Management Science **6**(1), 80–91 (1959)

[9] Das, A., Mathieu, C.: A quasipolynomial time approximation scheme for Euclidean capacitated vehicle routing. Algorithmica **73**(1), 115–142 (2015)

[10] Das, S., Jain, L., Kumar, N.: A constant factor approximation for capacitated min-max tree cover. arXiv:1907.08304 (2019), to appear in APPROX/RANDOM 2020

[11] Diarrassouba, I.: On the complexity of the separation problem for rounded capacity inequalities. Discrete Optimization **25**, 86–104 (2017)

[12] Even, G., Garg, N., Koenemann, J., Ravi, R., Sinha, A.: Min–max tree covers of graphs. Operations Research Letters **32**(4), 309–315 (2004)

[13] Goemans, M.X.: Worst-case comparison of valid inequalities for the TSP. Mathematical Programming **69**(1), 335–349 (1995)

[14] Haimovich, M., Kan, A.H.G.R.: Bounds and heuristics for capacitated routing problems. Mathematics of Operations Research **10**(4), 527–542 (1985)

[15] Khachay, M., Dubinin, R.: PTAS for the Euclidean capacitated vehicle routing problem in $\mathbb{R}^d$. In: Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., Pardalos, P. (eds.) Discrete Optimization and Operations Research. pp. 193–205. Springer International Publishing, Cham (2016)

[16] Labbé, M., Laporte, G., Mercure, H.: Capacitated vehicle routing on trees. Operations Research **39**(4), 616–622 (1991)

[17] Maßberg, J., Vygen, J.: Approximation algorithms for network design and facility location with service capacities. In: Chekuri, C., Jansen, K., Rolim, J.D.P., Trevisan, L. (eds.) Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques. pp. 158–169. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)

[18] Pecin, D., Pessoa, A., Poggi, M., Uchoa, E.: Improved branch-cut-and-price for capacitated vehicle routing. Mathematical Programming Computation **9**(1), 61–100 (2017)

[19] Pessoa, A., Sadykov, R., Uchoa, E., Vanderbeck, F.: A generic exact solver for vehicle routing and related problems. In: Lodi, A., Nagarajan, V. (eds.) Integer Programming and Combinatorial Optimization. pp. 354–369. Springer International Publishing, Cham (2019)

[20] Schrijver, A.: Combinatorial optimization: polyhedra and efficiency, vol. 24. Springer Science & Business Media (2003)

[21] Tröbst, T.: Capacitated vehicle routing and cycle covering problems. Master's Thesis, Research Institute for Discrete Mathematics, University of Bonn (2019)

[22] Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research **234**(3), 658 – 673 (2014)

[23] Yu, W., Liu, Z.: Improved approximation algorithms for some min-max and minimum cycle cover problems. Theoretical Computer Science **654**, 45–58 (2016)

[24] Yu, W., Liu, Z.: Better approximability results for min–max tree/cycle/path cover problems. Journal of Combinatorial Optimization **37**(2), 563–578 (2019)

[25] Yu, W., Liu, Z., Bao, X.: New approximation algorithms for the minimum cycle cover problem. Theoretical Computer Science (2019)