

# Covering edges in networks

Nicolas Fröhlich<sup>1</sup> | Andrea Maier | Horst W. Hamacher

Department of Mathematics, Technische Universität Kaiserslautern, Germany

## Correspondence

Nicolas Fröhlich, Technische Universität Kaiserslautern, Germany.  
Email: froehlich@mathematik.uni-kl.de

## Funding information

This research was supported by the Bundesministerium für Bildung und Forschung, 13N14561.

## Abstract

In this paper we consider the covering problem on a network  $G = (V, E)$  with edge demands. The task is to cover a subset  $J \subseteq E$  of the edges with a minimum number of facilities within a predefined coverage radius. We focus on both the nodal and the absolute version of this problem. In the latter, facilities may be placed everywhere in the network. While there already exist polynomial time algorithms to solve the problem on trees, we establish a finite dominating set (i.e., a finite subset of points provably containing an optimal solution) for the absolute version in general graphs. Complexity and approximability results are given and a greedy strategy is proved to be a  $(1 + \ln(|J|))$ -approximate algorithm. Finally, the different approaches are compared in a computational study.

## KEYWORDS

approximability, computational study, edge cover, finite dominating set, network location, NP-hardness

## 1 | INTRODUCTION AND LITERATURE REVIEW

In this paper, we tackle a version of the covering problem in facility location on networks, where demand arises continuously along the edges. This is an extension of the basic covering model, in which only the nodes of the graph are to be covered. The location set cover problem was proposed by Hakimi [10]. His aim was to find the “best” locations in a street network to position the police in order to cover a road network. Although this application calls for demand on the streets, Hakimi solved the problem by shifting the demand to the nodes. However, in many real-world problems, demand originates from the edges, for example garbage collection, or bus stop location.

By incorporating edge demands, various different location models can be tackled within a single framework. First, facilities may only be placed on nodes, and an edge to be covered has to be served by a single facility. Second, facilities may be sited anywhere in the network, and each demand edge has to be covered by a single facility. Third, facilities may only be located on nodes, but several facilities may serve the demand along a single edge (this type of covering is called complementary covering). Fourth, facilities may be placed anywhere in the network, and edges may be covered complementarily. Clearly, in the second and the fourth case, the number of possible locations as well as the points which have to be covered are infinite sets. However, for the second case, it is known that the set of feasible locations can be reduced to a finite dominating set (FDS), that is, a set of candidate points which surely contains an optimal solution (cf. Daskin [7]). This paper considers the two cases where edges may be complementarily covered by multiple facilities.

Important applications for a complementary covering arise in demonstrations, street festivals or other urban events, where the police have to safeguard participants and residents. Potential conflict zones are distributed over the urban area and have to be covered by policemen within a predefined distance, the so-called coverage radius. Since police resources are scarce, the question is how to find the minimum number of policemen in order to cover the entire event sufficiently and where to locate them.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. *Networks* published by Wiley Periodicals, Inc.

In spite of its practical relevance and the scientific challenge in dealing with this problem, literature on network location problems with edge demands is rather sparse. Quite often, demand is aggregated at the nodes to model such problems. This may lead to nonoptimal or often infeasible results. The covering problem in networks with edge demands was first introduced by Reville, Toregas, and Falkson [15]. Motivated by servicing population along street networks, a set of both nodes and edges should be covered by a minimum number of facilities within a maximum service distance. In their model, the candidate sites of the facilities are restricted to the nodes in the network. Complementary and partial coverage of an edge by two facilities are not allowed. In contrast, the model in Sadigh, Mozafari, and Kashan [16] allows complementary coverage of an edge by two facilities. A mixed integer linear programming formulation as well as size reduction rules for the problem is given. The problem is then solved using a modified tabu search algorithm. Again, candidate locations are restricted to the nodes of the network.

In Church and Meadows [6], the idea of edge-covering is applied to the maximal covering location problem. Again, the set of possible facilities is restricted to the nodes of the network. The problem can be stated by computing the so-called network intersection points which mark the border of the maximum coverage distance of the candidate facilities. These points divide the edges of the network into subedges such that all points within such a subedge can be covered by the same nodes. Thus, the problem can be solved by maximizing the total length of subedges covered by the located facilities by integer linear programming. Berman, Verter, and Kara [3] give an alternative formulation for the maximal edge-covering model with fewer number of variables and constraints. The result is a nonlinear mixed integer program. They present a greedy algorithm for solving larger problem instances and conduct a case study addressing the transportation of hazardous materials. In Gurevich, Stockmeyer, and Vishkin [9], a linear time algorithm for solving edge covering problems with complementary coverage on trees is developed. They generalize their algorithm for problems on general graphs leading to an exponential time algorithm. Their idea is to divide the graph into components which are “almost trees.” They also showed that facilities of an optimal solution can be located on multiples of  $1/2$  along the edges when edge lengths are integral. In the most recent paper by Berman, Kalcsics, and Krass [2], the maximum location covering problem with edge-based demand as well as its obnoxious version are introduced. In their work, facilities may also be located along the edges of the network. For the single facility case, a solution algorithm is given which is either exact or approximate depending on the complexity of the demand functions along the edges. For the multifacility problem on trees with constant demand functions, a FDS is established. They present an example establishing that these results are not transferable to general graphs.

The work of Miniéka [14] is the first article incorporating edge demands in median and center problems. In his work, he calculates the (absolute) unweighted centers and medians of a graph such that every point on every edge is served. In Handler and Mirchandani [11], the  $p$ -median problem with discrete nodal and continuous edge demands is formulated. To solve the problem on trees, a variant of the Goldman algorithm (cf. Goldman [8]) is developed. In Brandeau, Chiu, and Batta [4], Cavalier and Sherali [5], and Sherali and Nordai [17], median problems on special graphs such as trees, cycles, and chain graphs are investigated. Some error bounds for the  $p$ -median problem with edge demands when restricting the candidate facilities to the node set are given in Zhao and Batta [19]. They show that in this case, a model with continuous edge demands can be transformed into a model with discrete edge demands.

As we aim to locate facilities everywhere along the edges, the set of possible locations is infinite. In order to reduce this problem to a combinatorial one, one might ask for a finite subset of the solution space in which the optimal solution is contained. Such a set is called a FDS. A survey on such finite sets for different network location problems is given in Hooker, Garfinkel, and Chen [12].

The rest of this paper is structured as follows: in Section 2, the complementary edge covering problem is formally introduced. For general graphs, a FDS is derived. Complexity and approximability results are given in Section 3. A greedy algorithm with approximation guarantee is established in Section 4 using the structural insights of the FDSs. Section 5 presents a mixed-integer linear programming formulation for solving the problem and compares the different solution approaches in a computational study. The paper concludes in Section 6 with a summary and further research interests.

## 2 | PROBLEM SETUP AND STRUCTURAL RESULTS

### 2.1 | Notations and definitions

Let  $G = (V, E)$  be an undirected graph with node set  $V$  and edge set  $E$ . For an edge  $e \in E$ , its length is given by a positive integer  $l_e \in \mathbb{N}$ . (Note the theoretical results presented in this paper carry over to  $l_e \in \mathbb{R}^+$ , mutatis mutandis.) Although  $G$  is undirected, we often refer to an edge as ordered tuple  $e = (u, v)$  connecting nodes  $u, v \in V$ , with an arbitrary but fixed order. This is necessary to define a *point*  $x$  on edge  $e = (u, v) \in E$ , denoted by  $x \in e$  given by a tuple  $x = (e, \alpha)$ ,  $\alpha \in [0, 1]$ , where  $\alpha$  is the relative distance of  $x$  from  $u$  with respect to  $l_e$ . By  $(e, [\alpha_1, \alpha_2])$ , we denote the *subedge* of  $e \in E$  starting at  $(e, \alpha_1)$  and ending at  $(e, \alpha_2)$  for some  $0 \leq \alpha_1 < \alpha_2 \leq 1$ . That is,  $(e, [\alpha_1, \alpha_2]) := \{x \in e \mid x = (e, \alpha) \text{ for } \alpha_1 \leq \alpha \leq \alpha_2\}$ . The *interior of an edge*  $e \in E$  is given by  $\text{int}(e) := \{x \in G \mid x = (e, \alpha) \text{ for } \alpha \in (0, 1)\}$ .

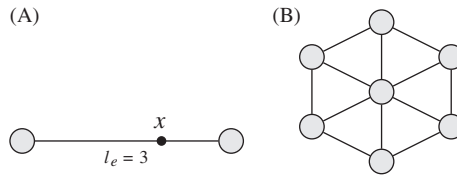


FIGURE 1 Motivating examples. A, An optimal solution. B, Edges are not totally covered

By definition, the equality  $((u, v), \alpha) = ((v, u), 1 - \alpha)$  holds for every  $(u, v) \in E$  and  $\alpha \in [0, 1]$ . Note that each nonsingleton node  $v \in G$  can be considered as a point, since it can be represented by such a tuple by choosing an edge incident to  $v$  and setting  $\alpha = 0$  or  $\alpha = 1$ , respectively. We use the notation  $x \in G$  if we do not want to specify whether  $x$  is a node or a point on an edge. For two points  $x = (e_1, \alpha)$ , and  $y = (e_2, \beta)$  with  $e_1 = (u, v)$ ,  $e_2 = (w, z) \in E$ ,  $e_1 \neq e_2$ , the *distance between  $x$  and  $y$*  is defined as

$$\text{dist}(x, y) = \min\{\alpha l_{e_1} + \text{dist}(u, w) + \beta l_{e_2}, \alpha l_{e_1} + \text{dist}(u, z) + (1 - \beta) l_{e_2}, (1 - \alpha) l_{e_1} + \text{dist}(v, w) + \beta l_{e_2}, (1 - \alpha) l_{e_1} + \text{dist}(v, z) + (1 - \beta) l_{e_2}\}.$$

For two points  $x = (e, \alpha)$ , and  $y = (e, \beta)$  on the same edge  $e = (u, v) \in E$ ,  $\alpha < \beta$ , the *distance between  $x$  and  $y$*  is defined as  $\text{dist}(x, y) = \min\{(\beta - \alpha)l_e, \alpha l_e + \text{dist}(u, v) + (1 - \beta) l_e\}$ .

For a given radius  $\rho \in \mathbb{N}$ , a point  $x \in G$  is called *covered* by a point  $f \in G$  if  $\text{dist}(x, f) \leq \rho$ . Analogously, a point  $x \in G$  is covered by a set  $\mathcal{F} := \{f_1, \dots, f_p\} \subseteq G$  of points, if there is some  $f_i \in \mathcal{F}$  that covers  $x$ . An edge  $e \in E$  is covered if all points  $x \in e$  are covered. For a point  $x \in G$ , its *covered area* is defined by  $\mathcal{A}(x) = \{y \in G \mid \text{dist}(x, y) \leq \rho\}$ . Analogously, for a finite subset  $\mathcal{F} \subseteq G$ , its covered area is defined by  $\mathcal{A}(\mathcal{F}) = \{y \in G \mid \exists x \in \mathcal{F} : \text{dist}(x, y) \leq \rho\}$ . The boundary of the covered area  $\mathcal{A}(\mathcal{F})$  is defined as

$$\text{bd}(\mathcal{A}(\mathcal{F})) := \{x \in \mathcal{A}(\mathcal{F}) \mid \forall \varepsilon > 0 : \exists y \in G \setminus \mathcal{A}(\mathcal{F}) : \text{dist}(x, y) \leq \varepsilon\}.$$

**Definition 2.1.** (Complementary edge cover problem). Given a graph  $G = (V, E)$  with edge lengths  $l_e \in \mathbb{N}$ , a maximum coverage distance  $\rho \in \mathbb{N}$ , and a subset  $J \subseteq E$ , the *nodal complementary edge covering location problem* (NCEC) is to find a set  $\mathcal{F} := \{f_1, \dots, f_p\}$  with  $f_i \in V$  for all  $i \in \{1, \dots, p\}$  of minimal cardinality such that all edges  $e \in J$  are covered by the set  $\mathcal{F}$ . If the elements in  $\mathcal{F}$  are allowed to be anywhere in the network (i.e.,  $f_i \in G$  for all  $i \in \{1, \dots, p\}$ ) the problem is called *absolute complementary edge covering location problem* (ACEC).

Consider the graph given in Figure 1A and a coverage radius  $\rho = 2$ . While at least two facilities are needed in the node restricted version, the point  $x = (e_2, \frac{2}{3})$  covers edge  $e$  completely. On the other hand, ACEC is not approximable by solving a vertex-covering problem where the end nodes of all edges are covered, see Figure 1B: for unit edge lengths and  $\rho = 1$ , a facility located at the center node covers all nodes but none of the points in the interior of the outer edges. By extending the outer cycle, the solution gets arbitrarily bad.

## 2.2 | Structural results

The next results establish that we may assume in the following without loss of generality that the length of edges in the network is bounded by  $\rho$ .

**Lemma 2.2.** *Given an instance of ACEC with an edge  $e \in J$  of length  $l_e > 2\rho$ . Then, the length of  $e$  can be reduced by  $2\rho$  without affecting the optimal solution except for the facility which can be saved in order to cover the deleted part of the edge.*

*Proof.* Let  $I$  be an arbitrary instance of ACEC with  $l_e > 2\rho$  for some  $e = (u, v) \in J$  and let  $I'$  denote the instance where  $l_e$  is reduced by  $2\rho$ . Furthermore, let  $\mathcal{F}^*$  be an optimal solution for instance  $I$ . Clearly, there is at least one facility  $x \in \mathcal{F}^*$  hosted on edge  $e$ . We consider two cases (cf. Figure 2):

Case 1:  $\mathcal{A}(x) \subset e$ , that is, facility  $x$  does not cover any points outside of edge  $e$ . Thus, facility  $x$  splits  $e$  into (at most) three parts: the part  $(e, [t_1, t_2])$  for suitable  $t_1, t_2$  with  $0 \leq t_1 < t_2 \leq 1$  which is covered by  $x$ , and the parts  $(e, [0, t_1])$  and  $(e, [t_2, 1])$  which are covered by other facilities. Note that by definition,  $l(e, [t_1, t_2]) = 2\rho$ . Define  $\varepsilon := l(e, [0, t_1])$  and  $\delta := l(e, [t_2, 1])$ . In instance  $I'$ , the edge corresponding to  $e$  has length  $\varepsilon + \delta$  and thus, the set of facilities  $\mathcal{F}^* \setminus \{x\}$  is feasible for  $I'$ . Assume now that  $\mathcal{F}^* \setminus \{x\}$  is not optimal for  $I'$ . Then there is a solution  $\mathcal{F}'$  of  $I'$  with  $|\mathcal{F}'| < |\mathcal{F}^*| - 1$ . Since we can cover the additional edge length of  $2\rho$  by a single facility,  $\mathcal{F}^*$  is not optimal for  $I$ , which is a contradiction.

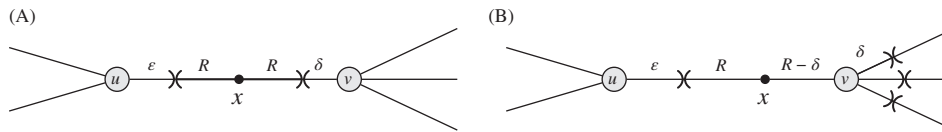


FIGURE 2 Edge with length greater than  $2R$ . A, Case 1:  $\mathcal{A}(x) \subseteq e$ . B, Case 2:  $\mathcal{A}(x) \cap e \neq \mathcal{A}(x)$

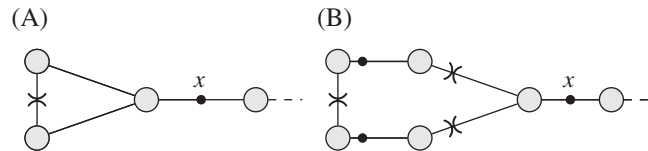


FIGURE 3 Illustration of coverage points. A, An exact coverage point. B, A cycle coverage point

Case 2:  $\mathcal{A}(x) \setminus e \neq \emptyset$ , that is, facility  $x$  covers points outside of edge  $e$ . Without loss of generality, let  $\text{dist}(x, v) = \rho - \delta$  for some  $\delta > 0$  (otherwise change the roles of  $u$  and  $v$ ). Following the argument of the first case, the length of edge  $e$  reduces from  $\varepsilon + 2\rho - \delta$  in instance  $I$  to  $\varepsilon - \delta > 0$  in instance  $I'$  and hence,  $\mathcal{F}^* \setminus \{x\}$  is feasible and optimal for  $I'$ . ■

By iteratively applying the last lemma, without loss of generality, instances of ACEC can be restricted to edge lengths smaller than or equal to  $2\rho$ . The optimal solution for edges with length greater than  $2\rho$  can be reconstructed by locating  $\lceil l_e/2\rho \rceil$  additional facilities at an interval of  $2\rho$ , starting from any neighboring facility. Furthermore, restricting the edge lengths to at most  $\rho$  can be achieved by splitting those edges with length  $R < l_e \leq 2\rho$ , that is, the number of nodes in the modified graph is bounded by  $(n + m)$  and the number of edges is bounded by  $2m$ .

**Assumption 2.3.** In the following, we assume that  $l_e \leq \rho$  for all  $e \in E$ .

For the node restricted version of the problem,  $l_e > \rho$  implies that two facilities have to be placed at both end nodes of edge  $e \in E$ , whereas infeasibility of the problem instance follows directly if  $l_e > 2\rho$ . Thus, Assumption 2.3 also applies to NCEC.

According to Gurevich, Stockmeyer, and Vishkin [9], the set

$$FDS' := \left\{ x \in G \mid x = \left( e, i \cdot \frac{1}{2l_e} \right) \text{ for } e \in E, i = 0, \dots, 2l_e \right\}. \quad (1)$$

is a FDS for ACEC. However, the size of this set depends on the edge lengths of the network. In this section, we establish an FDS for ACEC which is independent of the edge lengths. In particular, the FDS derived below is a subset of  $FDS'$ . On the other hand, the new FDS is more difficult to calculate since it is based on cycles. However, it yields some structural insights which can be extended to real valued edge lengths. In Section 4, we derive a greedy approximation algorithm based on these insights.

In order to compute the FDS, we need to define the network intersection and exact coverage points (see e.g., [2]).

**Definition 2.4.** (Network intersection points). A point  $x \in G$  whose distance  $\text{dist}(v, x)$  is exactly  $\rho$  for some node  $v \in V$  is called *network intersection point* of  $v$ . The set of all network intersection points is denoted by NIP.

**Definition 2.5.** (Exact coverage points). A point  $x \in G$  is called an *exact coverage point* of edge  $e = (u, v)$  if it exactly covers edge  $e$  complementarily through both endpoints of  $e$ , that is, there exists some  $y = (e, \alpha) \in e$  such that  $\text{dist}(x, u) + \alpha l_e = \text{dist}(x, v) + (1 - \alpha)l_e = \rho$  (cf. Figure 3A). The set of all exact coverage points is denoted by EP.

We also need to define a new type of points for the FDS.

**Definition 2.6.** (Cycle coverage points). Let  $C_G$  denote the set of all simple cycles in a graph  $G$ . Furthermore, we define the length of a cycle  $C \in C_G$  as the sum of its edge lengths, denoted by  $l_C$ . A point  $x \in G \setminus C$  is called *cycle coverage point* of  $C$ , if

$$\text{dist}(x, C) = \left( \rho - \frac{l_C}{2} \right) \bmod \rho,$$

where  $\text{dist}(x, C) = \min_{y \in C} \text{dist}(x, y)$ . The set of all cycle coverage points in a graph is denoted by CCP.

Since  $\rho := \lceil l_C/2\rho \rceil$  points are needed to cover a cycle  $C$ , a cycle coverage point  $x$  is reached by shifting one of the facilities as far as possible outside the cycle as illustrated in Figure 3B. Shifting the facility any further would cause a coverage gap on the cycle. Clearly,  $\text{EP} \subseteq \text{CCP}$  by definition.

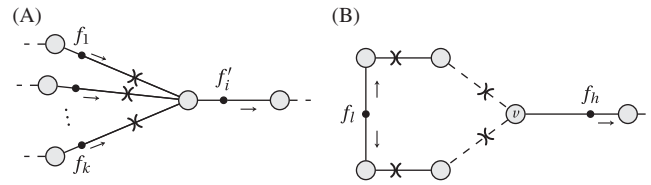


FIGURE 4 Illustration of the proof of Theorem 2.7. A, Facilities  $f_1, \dots, f_k$  have to be shifted. B, Facility  $f_i$  is drawn into two opposite directions

**Theorem 2.7.** (FDS on general graphs). *Define*

$$FDS_{|J|} := V \cup \text{NIP} \cup \text{CCP}$$

and recursively for  $j \geq 1$

$$FDS_{j+1} := FDS_j \cup \{x \in G \mid \exists y \in \text{bd}(\mathcal{A}(FDS_j)) : \text{dist}(x, y) = \rho\}.$$

Then,  $FDS_{|J|}$  is a FDS for ACEC for any graph.

*Proof.* Let  $\mathcal{F}^* = \{f_1, \dots, f_p\}$  be an optimal solution of ACEC. Without loss of generality, we can assume that  $J = E$ , since we could delete all edges  $e \in E$  which are not completely covered by  $\mathcal{F}^*$ , otherwise. Choose any arbitrary facility  $f_i \in \mathcal{F}^*$  that does not belong to  $FDS_{|J|}$ . Then,  $f_i = (e, \alpha)$  for some edge  $e \in E$  and  $\alpha \in (0, 1)$ . By either increasing or decreasing  $\alpha$ , we can move  $f_i$  along edge  $e$  until the optimal solution becomes infeasible. Let  $f_i'$  be the last point that the solution is just feasible or  $\alpha \in \{0, 1\}$ . In the latter case,  $f_i'$  is located on a node and thus,  $f_i' \in FDS_{|J|}$ .

Otherwise we consider three cases:

- The optimal solution becomes infeasible because  $f_i'$  stops covering an edge completely. In this case, by definition,  $f_i' \in \text{NIP}$  and thus,  $f_i' \in FDS_{|J|}$ .
- The optimal solution becomes infeasible because  $f_i'$  stops covering an edge  $c = (u, v) \in J$  complementarily through both end nodes  $u$  and  $v$ . Then, by definition,  $f_i' \in \text{CCP} \subseteq FDS_{|J|}$ .
- There are some critical facilities, say  $f_1 = (e_1, \alpha_1), \dots, f_k = (e_k, \alpha_k)$ , for  $k \in \mathbb{N}$ , such that

$$\text{bd}(\mathcal{A}(f_j)) \cap \text{bd}(\mathcal{A}(f_l)) \neq \emptyset \quad \forall j \in \{1, \dots, k\} \tag{2}$$

and moving  $f_i'$  any further would cause a gap at the boundaries (cf. Figure 4A). If there exists  $\alpha_j$  with  $\alpha_j \in \{0, 1\}$  then  $f_i' \in FDS_{|J|}$ . Otherwise, we continue shifting  $f_i'$ , but to maintain feasibility, we have to either increase or decrease  $\alpha_j$ , for all  $j \in \{1, \dots, k\}$  such that (2) is still satisfied. That is, all neighboring facilities are also moved. Iteratively repeating this process necessarily yields a solution such that the facilities cannot be moved anymore without losing feasibility, because either:

- One of the facilities reaches a vertex. Then, all involved facilities are in  $FDS_{|J|}$  since (2) is still satisfied.
- One facility becomes a network intersection point or a cycle coverage point. By the same arguments as before, all involved facilities are in  $FDS_{|J|}$ .
- There exists a location  $f_l = (e_l, \alpha_l)$  which cannot be moved anymore since it is drawn in two different directions, that is, we would simultaneously have to increase and to decrease  $\alpha_l$  (cf. Figure 4B). But this is only possible if there exists a cycle  $C$  and a vertex  $v \in C$  such that a facility  $f_h \notin C$  covers node  $v$  and a subset of  $C$ . Since  $f_l$  is drawn into two different directions, cycle  $C$  is completely covered by facilities lying on  $C$ . Moreover, each of these facilities covers distinct parts of  $C$ , each of length  $2\rho$ . Since (2) still holds, facility  $f_h$  has to be a cycle coverage point, that is,  $f_h \in \text{CCP}$ . Hence, all involved facilities are included in  $FDS_{|J|}$ .

Fixing the facilities which are already included in  $FDS_{|J|}$  and repeating the whole process yield the desired result. ■

Note that the cardinality of  $FDS_{|J|}$  may be exponential in the input size of ACEC since the number of cycles is exponential for a graph in general. However, the result gives structural insights to the locations of optimal facilities. Note that also,  $FDS_{|J|}$  is a subset of  $FDS'$  and thus, contains less candidates. Furthermore, the result implies a FDS of polynomial size whenever the number of simple cycles in the graph is polynomially bounded. The last property implies immediately the following corollary, as the number of simple cycles in cactus graphs is, by definition, in  $\mathcal{O}(m)$ .

**Corollary 2.8.** *The size of  $FDS_{|J|}$  is polynomially bounded for cactus graphs.*

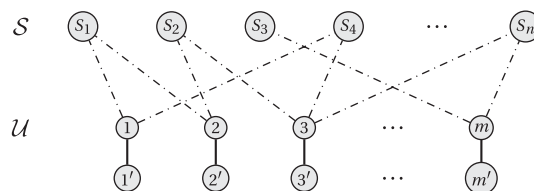


FIGURE 5 Instance of nodal complementary edge covering location problem with unit edge lengths and  $\rho = 2$

### 3 | COMPLEXITY AND APPROXIMABILITY OF COMPLEMENTARY EDGE COVER PROBLEMS

In this section, we analyze the complexity and approximability of complementary edge cover problems. As a main result, we first show bounds on approximability and establish a greedy algorithm yielding a  $(1 + \ln|J|)$ -approximation ratio.

#### Theorem 3.1.

- a NCEC is not in APX, that is, there exists no polynomial time  $r$ -approximate algorithm to solve this problem for any constant  $r \geq 1$  unless  $P = NP$ .
- b NCEC is not approximable in polynomial time within  $(1 - \epsilon)\ln(|J|)$  for  $\epsilon > 0$  unless  $P = NP$ .
- c ACEC is not in APX.

*Proof.*

- a We show that MINIMAL SET COVER is L-reducible to NCEC. The claim then follows since the L-reduction is an approximability preserving reduction, see for example Ausiello et al [1].

Let  $I = (\mathcal{U}, \mathcal{S})$  be an arbitrary instance of MINIMAL SET COVER, where  $\mathcal{U} = \{1, \dots, m\}$  is a universe and  $\mathcal{S} = \{S_1, \dots, S_n\}$  is a collection of nonempty subsets of  $\mathcal{U}$ . The problem is to find a subset  $\mathcal{S}' \subseteq \mathcal{S}$  of minimal cardinality such that  $\mathcal{S}'$  covers  $\mathcal{U}$ .

Construct a graph  $G = (V, E)$  as follows: for every set  $S_i \in \mathcal{S}$ , create a node in  $G$ . Furthermore, add two nodes  $u$  and  $u'$  for every element  $u \in \mathcal{U}$  and connect them by an edge  $(u', u)$ . Finally connect all nodes  $u \in \mathcal{U}$  with all nodes  $S_i$  if and only if  $u \in S_i$ . All edges  $e \in E$  are assumed to have unit edge lengths, and  $J$  is given by  $\{(u, u') \in E | u \in \mathcal{U}\}$  (see Figure 5). By setting the coverage radius  $\rho = 2$  there exists a set cover of cardinality  $p$  if and only if there is a solution of NCEC for instance  $(G, l, J, \rho)$  using  $p$  facilities located at nodes  $S_1, \dots, S_n$ . Observe that if there is an optimal solution  $\mathcal{F}$  where facilities are located at a node corresponding to some  $u \in \mathcal{U}$ , then this solution can be transformed in linear time to a solution within  $\mathcal{F} \subseteq \mathcal{S}$  by relocating all facilities from  $u$  to an arbitrary set  $S_i \in \mathcal{S}$  containing  $u$ .

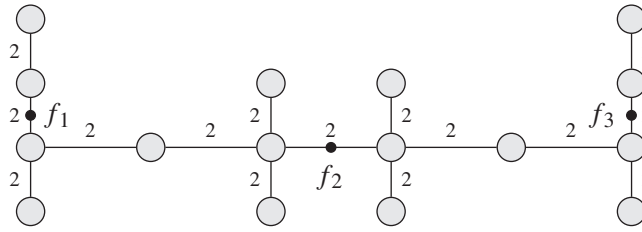
Clearly, for every instance of MINIMAL SET COVER, the instance  $(G, l, J, \rho)$  of NCEC is computable in polynomial time. If the instance of MINIMAL SET COVER has a solution, then the instance of NCEC has a solution, and any (not necessarily optimal) solution of the NCEC-instance can be transformed back in linear time to a solution of the MINIMAL SET COVER instance having the same cardinality.

- b Follows from the L-reduction of Part (a) and the inapproximability result of MINIMAL SET COVER, which is not approximable within  $(1 - \epsilon)\ln(|\mathcal{U}'|)$ , see Ausiello et al [1].
- c The L-reduction is analogous to Part (a) since any optimal facility  $f \notin \{S_1, \dots, S_n\}$  can be shifted to a node  $S_i \in \mathcal{S}$  closest to  $f$  without worsening the optimal objective value. ■

#### Theorem 3.2. NCEC is L-reducible to MINIMAL SET COVER.

*Proof.* Every node  $i \in V$  can divide a given edge  $e = (u, v) \in E$  into at most three parts: one part covers  $(e, [0, t_1])$  for a suitable  $t_1 \in [0, 1]$ , the second part covers  $(e, [t_2, 1])$  for a suitable  $t_2$  with  $t_1 < t_2 \leq 1$  and the part which is not covered  $(e, [t_1, t_2])$ . Thus, iteratively dividing and sorting the intervals on each edge, we can have at most  $2|V| + 1$  intervals on an edge—adding up to  $|E|(2|V| + 1)$  in the worst case. We construct the MINIMAL SET COVER instance as follows: Let  $\mathcal{U}$  be the universe consisting of all the intervals on the edges in  $E$ . The set  $\mathcal{S} = \{S_1, \dots, S_{|V|}\}$  consists of  $|V|$  sets, where  $u \in \mathcal{U}$  is contained in the set  $S_i$  if  $u$  is covered by vertex  $i$ .

Clearly, any feasible solution of the MINIMAL SET COVER instance yields a feasible solution of the NCEC-instance with the same cardinality. Since the intervals on the edges can be calculated using shortest path algorithms, the reduction is polynomial and any solution of the MINIMAL SET COVER instance can be transformed in linear time to a solution of the NCEC-instance. ■



**FIGURE 6** A graph exploiting the gap between nodal complementary edge covering location problem and absolute complementary edge covering location problem.  $\rho = 3$

**Theorem 3.3.** *NCEC is approximable within  $3(1 + \ln|V|)$ .*

*Proof.* By Theorem 3.2 a MINIMAL SET COVER instance with  $|\mathcal{U}'| \leq |E|(2|V| + 1)$  can be computed in polynomial time having the same objective value. In addition any solution of this instance can be transformed back to a solution of the NCEC-instance in linear time. By Johnson [13], the minimal set cover problem can be approximated within  $1 + \ln|\mathcal{U}'|$ . Thus, transforming NCEC into a set cover problem and using the algorithm stated in Johnson [13] returns an approximate solution with factor  $1 + \ln(|E|(2|V| + 1))$ . This implies the desired result by simple calculations. ■

Note that the idea of Theorem 3.3 only works for NCEC since the nodes of  $V$  are a FDS of polynomial size. However, this does not work for ACEC.

**Theorem 3.4.** *Let  $I = (G, l, J, \rho)$  be the underlying graph of an instance of NCEC and ACEC and let  $\mathcal{F}_{\text{NCEC}}^*$  and  $\mathcal{F}_{\text{ACEC}}^*$  denote a corresponding optimal solutions with respect to instance  $I$ , respectively. Then,*

$$|\mathcal{F}_{\text{ACEC}}^*| \leq |\mathcal{F}_{\text{NCEC}}^*| \leq 2 \cdot |\mathcal{F}_{\text{ACEC}}^*|$$

and this bound is tight.

*Proof.* The first inequality is trivial, since ACEC is a relaxation of NCEC. Let  $I$  be an arbitrary instance of ACEC and  $\mathcal{F}_{\text{ACEC}}^* = \{f_1, \dots, f_p\}$  be an optimal solution of ACEC with respect to  $I$ . If  $f_i, i \in \{1, \dots, p\}$  is already a nodal location, then we have nothing to show. So let  $f_i \in \text{int}(e)$  for some  $e = (u, v) \in E$ . Replacing  $f_i$  by two facilities located at  $u$  and  $v$ , respectively, yields a feasible solution of NCEC. Hence,  $2 \cdot |\mathcal{F}_{\text{ACEC}}^*|$  is an upper bound of  $|\mathcal{F}_{\text{NCEC}}^*|$ . To show that this bound is tight we consider the graph given in Figure 6 with edge lengths  $l_e = 2$  for all edges  $e \in E$  and coverage radius  $\rho = 3$ . In this example, we need 3 facilities  $f_1, f_2$  and  $f_3$  in the absolute case whereas 6 facilities have to be installed in the nodal case. By concatenating multiple copies of the graph, arbitrarily large instances can be constructed. ■

**Corollary 3.5.** *ACEC is approximable within a factor  $6(1 + \ln|V|)$ .*

*Proof.* The proofs of Theorems 3.3 and 3.4 yield an approximation ratio of  $2 + 2\ln|E| + 2\ln(2|V| + 1)$  implying the desired result. ■

In the next section we will see that a greedy strategy similar to the greedy algorithm of set cover achieves an even better approximation ratio. We conclude the theoretical results on approximability with the following corollary.

**Corollary 3.6.** *ACEC is at least as hard to solve as its node restricted counterpart NCEC.*

*Proof.* Using a transitivity argument, Theorems 3.2 and 3.1c) imply  $\text{NCEC} \leq_L \text{MINIMAL SET COVER} \leq_L \text{ACEC}$ . ■

## 4 | GREEDY ALGORITHM FOR THE ABSOLUTE COMPLEMENTARY EDGE COVER PROBLEM

In this section, we establish a greedy algorithm extending the procedure introduced in Williamson and Shmoys [18] for approximating the minimal set cover. Their idea to choose in each iteration a set which minimizes the points of uncovered elements is, however, not directly applicable to ACEC. The two main differences are that

- the FDS obtained in the last section is of exponential size and
- we have to cover continuous sets instead of discrete elements.

In order to salvage the greedy idea we will show subsequently that we can reduce the cardinality of the candidates to polynomial size and introduce a measure for evaluating the “size” of the covered area.

Recall that  $\mathcal{A}(x) := \{y \in G \mid \text{dist}(x, y) \leq \rho\}$  is the covered area of a given point  $x \in G$ . To simplify notation, we denote the set of all points on edges of  $J$  by  $G_J := \{x \in e \mid e \in J\}$ . Let  $\mathcal{F}_k^G = \{x_1, \dots, x_k\}$  be the current solution of the greedy algorithm after iteration  $k$  and let  $\mathcal{F}^G = \{x_1, \dots, x_l\}$  denote the solution after termination of the algorithm. For  $k \leq l$ , we define  $C_k := G_J \cap \mathcal{A}(\mathcal{F}_{k-1}^G)$  as the set of all points that are covered by the current solution at the beginning of iteration  $k$ , and analogously, the set of all points which are not covered yet by  $U_k := G_J \setminus C_k$ . For arbitrary  $x \in G$ ,  $\widehat{C}_k(x)$  denotes the set of points which are covered by  $\mathcal{F}_{k-1}^G \cup \{x\}$  for the first time. That is,  $\widehat{C}_k(x) := \mathcal{A}(x) \cap U_k$ .

**Definition 4.1.** (Measure of a subedge). For any subedge  $S = (e, [\alpha_1, \alpha_2])$  of  $e \in E$ ,  $0 \leq \alpha_1 < \alpha_2 \leq 1$ , we define the *measure*  $\mu(S)$  of  $S$  as  $\mu(S) := (\alpha_2 - \alpha_1)l_e$ . Two subedges  $S_1$  and  $S_2$  are *internally disjoint*, if  $\mu(S_1 \cap S_2) = 0$ . For a set of pairwise internally disjoint subedges  $\{S_1, S_2, \dots\}$ , we define  $\mu(\bigcup_{k=1}^{\infty} S_k) := \sum_{k=1}^{\infty} \mu(S_k)$ .

---

**Algorithm 1.** Greedy algorithm for ACEC

---

**Input:** network  $G = (V, E)$  with edge lengths  $l_e$  for  $e \in E$ , a subset  $J \subseteq E$   
**Output:** a set of facilities covering all edges in  $J$

- 1: **procedure** GREEDY
- 2:  $k \leftarrow 0$
- 3:  $\mathcal{F}_k^G \leftarrow \emptyset$
- 4:  $U \leftarrow \{x \mid x \in G_J\}$
- 5: **while**  $U \neq \emptyset$  **do**
- 6:  $k \leftarrow k + 1$
- 7: **choose**  $x_k \in \arg \max_{x \in G} \mu(\widehat{C}_k(x))$
- 8:  $\mathcal{F}_k^G \leftarrow \mathcal{F}_{k-1}^G \cup \{x_k\}$
- 9:  $U \leftarrow U \setminus \widehat{C}_k(x_k)$
- 10: **end while**
- 11: **return**  $\mathcal{F}_k^G$
- 12: **end procedure**

---

The procedure of the greedy algorithm is outlined in Algorithm 1. The next lemma shows that it is only necessary to consider a subset of  $FDS_{|J|}$  which can be computed in polynomial time (cf. line 7).

**Lemma 4.2.** For any  $k \leq l$ , define

$$\text{CAND}_k = V \cup \text{NIP} \cup \text{EP} \cup \{x \in G \mid \exists y \in \text{bd}(C_k) : \text{dist}(x, y) = \rho\},$$

where  $C_k$  is the current covered area. Then,  $\max_{x \in G} \mu(\widehat{C}_k(x)) = \max_{x \in \text{CAND}_k} \mu(\widehat{C}_k(x))$ . Furthermore, the cardinality of  $\text{CAND}_k$  is polynomially bounded in the input size.

*Proof.* For  $x \in e$  where  $e \in E$ , the function  $\mu(\widehat{C}_k(x))$  is a piecewise linear function. The slope of  $\mu(\widehat{C}_k(x))$  changes as the number of (partially) covered edges changes or parts of the edge switch between covered and uncovered. These breakpoints are exactly the points contained in  $\text{CAND}_k$ . It remains to show that the size of  $\text{CAND}_k$  is polynomially bounded.

As  $l_e \leq \rho$  for all  $e \in E$ , in each iteration, at least one additional edge is completely covered that has not been covered before. Thus,  $l \leq |J|$ . We still have to identify the possible candidates for iteration  $k \leq |J|$ .

Since on each edge, there are at most two network intersection points per vertex, the cardinality of **NIP** is bounded by  $2|V| \cdot |E|$ . Moreover, each node  $v \in V$  and each edge  $e \in J$  yield at most  $|E| - 3$  exact coverage points since the smallest cycle contains three edges. Thus,  $|\text{EP}| \leq |J| \cdot |V| \cdot |E|$ .

In order to estimate the size of the set  $\{x \in G \mid \text{dist}(x, C_k) = \rho\}$  we analyze the boundary of  $C_k$ . On any edge  $e \in E$ , there are at most two boundary points of  $C_k$ . Each of those boundary points can cause at most two points on an edge with distance  $\rho$ . Hence,

$$\{x \in G \mid \exists y \in \text{bd}(C_k) : \text{dist}(x, y) = \rho\} \leq 2|E| \cdot 2|E| = 4|E|^2.$$

Note that this number is independent of the current iteration. In total, we have

$$\text{CAND}_k \leq |V| + 2|V| \cdot |E| + |J| \cdot |E| \cdot |V| + 4|E|^2.$$



Since each of these points can be calculated by minimum path algorithms, the greedy algorithm runs in polynomial time. ■

Analogous to the greedy algorithm for the minimal set cover problem (see [18]), we can prove the following theorem.

**Theorem 4.3.** (Approximation-ratio of Algorithm 1). *Algorithm 1 is a  $(1 + \ln(l))$ -approximation algorithm for ACEC, where  $l \leq |J|$  denotes the number of iterations of the algorithm.*

*Proof.* Choose any arbitrary iteration  $k$  of the greedy algorithm. To shorten the notation, we set  $A := \mu(G_J)$  to be the measure of the total area which has to be covered. We establish two bounds used in the proof. The first bound can be derived by simple observation, that is

$$\min_{i \in \{1, \dots, k\} : b_i \neq 0} \frac{a_i}{b_i} \leq \frac{\sum_{i \in \{1, \dots, k\}} a_i}{\sum_{i \in \{1, \dots, k\}} b_i}. \quad (3)$$

The second bound follows from  $\mu(\widehat{C}(x_1)) \geq \dots \geq \mu(\widehat{C}(x_l))$  by the choice of the greedy algorithm and is

$$\mu(U_k) = A - \sum_{i=1}^{k-1} \mu(\widehat{C}(x_i)) \geq A - (k-1) \frac{A}{l} = A \left(1 - \frac{k-1}{l}\right). \quad (4)$$

Also note that  $\mu(U_k) > 0$  for all  $k \leq l$ . Let  $\mathcal{F}^*$  denote an optimal solution with value OPT. Then, it follows

$$\begin{aligned} \frac{1}{\mu(\widehat{C}_k(x_k))} &= \min_{x \in G} \frac{1}{\mu(\widehat{C}_k(x))} \leq \min_{x \in \mathcal{F}^*} \frac{1}{\mu(\widehat{C}_k(x))} \\ &\leq \frac{\sum_{x \in \mathcal{F}^*} 1}{\sum_{x \in \mathcal{F}^*} \mu(\widehat{C}_k(x))} && \text{by (3)} \\ &= \frac{\text{OPT}}{\mu(U_k)}, \end{aligned}$$

where the last equality holds because  $\mathcal{F}^*$  is an optimal cover with  $\mathcal{A}(\mathcal{F}^*) = G_J$ . Since  $\mu(\widehat{C}_k(x_k)) = \mu(U_k) - \mu(U_{k+1})$ , it follows that

$$1 \leq \frac{\mu(U_k) - \mu(U_{k+1})}{\mu(U_k)} \cdot \text{OPT}. \quad (5)$$

In total, we have

$$\begin{aligned} |\mathcal{F}^G| &= \sum_{k=1}^l 1 \leq \left( \sum_{k=1}^l \frac{\mu(U_k) - \mu(U_{k+1})}{\mu(U_k)} \right) \cdot \text{OPT} && \text{by (5)} \\ &= \left( 1 + \sum_{k=1}^{l-1} \frac{\mu(U_k) - \mu(U_{k+1})}{\mu(U_k)} \right) \cdot \text{OPT} \\ &= \left( 1 + \sum_{k=1}^{l-1} \int_0^{\mu(U_k) - \mu(U_{k+1})} \frac{1}{\mu(U_k)} dz \right) \cdot \text{OPT} \\ &\leq \left( 1 + \sum_{k=1}^{l-1} \int_0^{\mu(U_k) - \mu(U_{k+1})} \frac{1}{\mu(U_k) - z} dz \right) \cdot \text{OPT} \\ &= \left( 1 + \sum_{k=1}^{l-1} \ln(\mu(U_{k+1})) - \ln(\mu(U_k)) \right) \cdot \text{OPT} \\ &= (1 + \ln(\mu(U_1)) - \ln(\mu(U_l))) \cdot \text{OPT} \\ &\leq \left( 1 + \ln(A) - \ln \left( A \left( 1 - \frac{l-1}{l} \right) \right) \right) \cdot \text{OPT} && \text{by (4)} \\ &= (1 + \ln(l)) \cdot \text{OPT} \\ &\leq (1 + \ln(|J|)) \cdot \text{OPT}, \end{aligned}$$

where the last inequality follows since  $l \leq |J|$ . ■

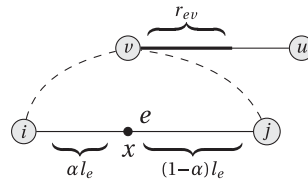


FIGURE 7 The coverage share on edge  $(v, u)$  from point  $x \in e$  via node  $v$

## 5 | MIXED-INTEGER PROGRAMMING FORMULATION AND COMPUTATIONAL STUDY

In the following, we derive a mixed-integer linear programming formulation for ACEC which is based on the following lemma.

**Lemma 5.1.** *There exists an optimal solution of ACEC such that no more than one facility is located on the same edge.*

*Proof.* Assume that we have an optimal solution in which two points are located on the same edge  $e \in E$  with  $e = (u, v)$ . Let these points be given by  $f_1 = (e, \alpha_1)$  and  $f_2 = (e, \alpha_2)$ , where we assume without loss of generality that  $0 \leq \alpha_1 \leq \alpha_2 \leq 1$ . Then we can set  $\alpha_1 = 0$  and  $\alpha_2 = 1$  and the resulting solution still covers all edges in  $J$  and is optimal (since  $l_e \leq \rho$  for all  $e \in E$ ). By optimality, neither  $u$  nor  $v$  is a leaf. Thus, we can set  $f_1 = (e, 0)$  and  $f_2 = (e', 0)$  for a suitable edge  $e' = (v, w) \in E$ . Since we can repeat this process for all edges on which two facilities are located, this yields the desired result. ■

Using Lemma 5.1, we can introduce a single decision variable  $x_e \in \{0, 1\}$  for each edge  $e \in E$  deciding whether this edge hosts a facility or not. Let  $x = (e, \alpha) \in G$  be a facility on edge  $e = (i, j)$  and  $c = (u, v) \in J$  an edge to be covered. Then,  $x$  covers  $r_{ev}^1 = \rho - \alpha_e l_e - \text{dist}(i, v)$  units of length of edge  $c$  through nodes  $i$  and  $v$ , and  $r_{ev}^2 = \rho - (1 - \alpha_e)l_e - \text{dist}(j, v)$  through nodes  $j$  and  $v$ , provided their values are positive. Therefore, the remaining length for covering edge  $c$  through node  $v$  from  $x$  is given by  $r_{ev} = \max\{r_{ev}^1, r_{ev}^2, 0\}$ , cf. Figure 7. Similarly,  $r_{eu}$  is the length that can cover  $c$  through node  $u$ . On the other hand, each edge  $c \in J$  has to be covered by one or two facilities (specified by their hosting edges  $e', e''$ ), which is determined by decision variables

$$z_{e', e''}^c = \begin{cases} 1, & \text{if } c \in J \text{ is covered by two facilities on edges } e', e'' \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Note that if the facility on edge  $e'$  covers edge  $c$  completely, the second facility may be chosen arbitrarily. The mixed-integer program is given by:

$$\min \sum_{e \in E} x_e \quad (6a)$$

$$\text{s.t. } r_{ev} = \max\{r_{ev}^1, r_{ev}^2, 0\} \quad \forall e \in E, v \in V \quad (6b)$$

$$\rho - \alpha_e l_e - \text{dist}(i, v) = r_{ev}^1 \quad \forall e = (i, j) \in E, v \in V \quad (6c)$$

$$\rho - (1 - \alpha_e)l_e - \text{dist}(j, v) = r_{ev}^2 \quad \forall e = (i, j) \in E, v \in V \quad (6d)$$

$$r_{e'i} + r_{e''j} \geq z_{e', e''}^c l_c \quad \forall c = (i, j) \in J, e', e'' \in E \quad (6e)$$

$$\sum_{e' \in E} \sum_{e'' \in E} z_{e', e''}^c = 1 \quad \forall c \in J \quad (6f)$$

$$z_{e', e''}^c \leq x_{e'} \quad \forall c \in J, e', e'' \in E \quad (6g)$$

$$z_{e', e''}^c \leq x_{e''} \quad \forall c \in J, e', e'' \in E \quad (6h)$$

$$x_e \in \{0, 1\} \quad \forall e \in E \quad (6i)$$

$$z_{e', e''}^c \in \{0, 1\} \quad c \in J, \quad \forall e', e'' \in E \quad (6j)$$

$$\alpha_e \in [0, 1] \quad \forall e \in E \quad (6k)$$

$$r_{ev}, r_{ev}^1, r_{ev}^2 \in \mathbb{R}_{\geq 0} \quad \forall e \in E, v \in V \quad (6l)$$

The objective (6a) is to minimize the number of edges chosen to host a facility which is the same as minimizing the number of located facilities. An edge  $e \in E$  is chosen to host a facility if and only if the corresponding binary decision variable  $x_e$  is set to one. The continuous variable  $\alpha_e$  specifies its exact position. Equations (6b)–(6d) ensure the computation of  $r_{ev}$  as deduced before. Inequality (6e) states the covering condition, that is, two facilities are required to cover edge  $c$  at least complementarily. Equality (6f) ensures that exactly one combination of edges  $e', e''$  is assigned for covering edge  $c = (i, j) \in J$ . Of course, only the combination of edges is eligible that are chosen to host a facility which is guaranteed by (6g) and (6h).

Using standard linearization techniques, the mixed integer programming formulation (6) is equivalent for any suitably large  $M \gg \rho$  to

$$\begin{aligned}
 \min \quad & \sum_{e \in E} x_e \\
 \text{s.t.} \quad & r_{ev} \geq r_{ev}^1 \quad \forall e \in E, v \in V \\
 & r_{ev} \geq r_{ev}^2 \quad \forall e \in E, v \in V \\
 & r_{ev} - (1 - \delta_{ev}^1)M \leq r_{ev}^1 \quad \forall e \in E, v \in V \\
 & r_{ev} - (1 - \delta_{ev}^2)M \leq r_{ev}^2 \quad \forall e \in E, v \in V \\
 & r_{ev} - (1 - \delta_{ev})M \leq 0 \quad \forall e \in E, v \in V \\
 & \delta_{ev}^1 + \delta_{ev}^2 + \delta_{ev} \geq 1 \quad \forall e \in E, v \in V \\
 & \delta_{ev}, \delta_{ev}^1, \delta_{ev}^2 \in \{0, 1\} \quad \forall e \in E, v \in V \\
 & \delta_{ev}^2 \in \{0, 1\} \quad \forall e \in E, v \in V \\
 & \delta_{ev} \in \{0, 1\} \quad \forall e \in E, v \in V \\
 & + \text{constraints (6c)–(6i)}.
 \end{aligned}$$

A mixed-integer linear programming formulation for the node restricted version NCEC is given by [16]. Since there are only finitely many candidate sites of the facilities, the structure of the linear programs for ACEC and NCEC differ fundamentally.

We performed computational tests comparing three solution approaches for ACEC. All instances are solved using the mixed-integer program given above, solving a discrete set cover problem on the  $1/2$ -discretization  $FDS'$  stated in Equation (1) by integer programming, and using the greedy approach of Algorithm 1. The three approaches are abbreviated as MILP,  $1/2$ -DISC, and GREEDY, respectively. Note that MILP and  $1/2$ -DISC solve ACEC exactly, whereas GREEDY approximates a solution to the problem.

In total, the analysis comprises 3600 random instances of ACEC. Each random graph is generated according to the Erdős-Rényi random graph model, and for each distinct combinations of the parameters  $|E| \in \{15, 20, 30, 40, 60, 100, 200, 300\}$  and  $\rho \in \{25, 50, 100, 150, 200\}$ , 90 instances are generated. The instances contain both, sparser and denser graphs by varying the number of nodes for each specific number of edges. The edge lengths are chosen randomly, and all edges of the network have to be covered, that is,  $J = E$ . All computations have been done on a compute server.<sup>1</sup> For each instance, a time limit of 600 seconds was set, and each instance was restricted to use a single kernel.

The average runtime results of all instances solved within the time limit are summarized in Table 1. Note that with the exception of GREEDY, some instances could not be solved the time limit. In this case the number of solved instances is given in brackets.

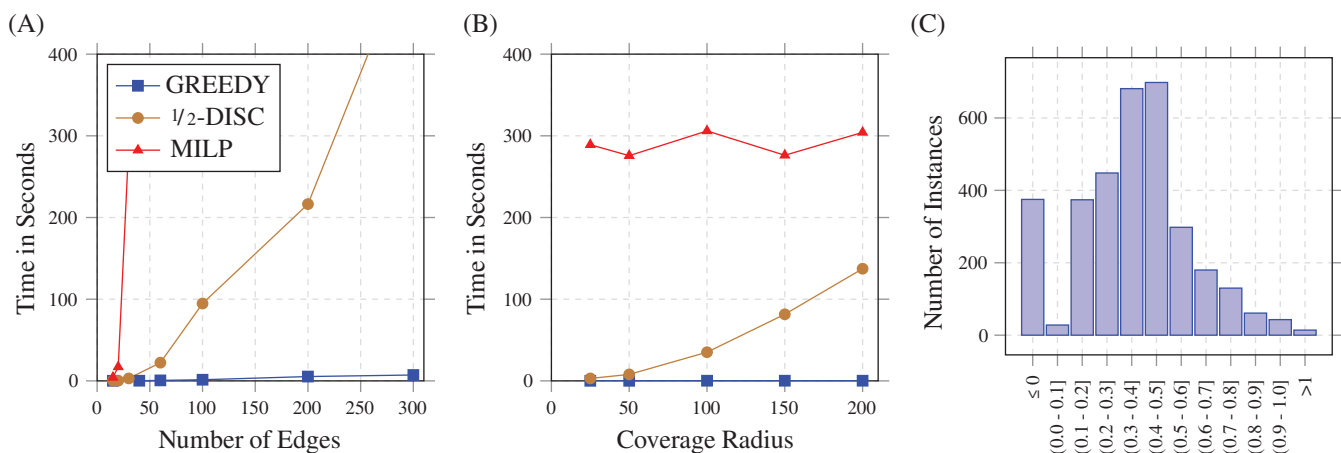
As  $\rho$  increases (and thus the length of the edges), the runtime of  $1/2$ -DISC heavily increases. This is due to the fact that there are much more candidates in the set cover instance. In contrast, the runtimes of GREEDY only increase slightly, and the runtimes of MILP are not affected (see Figure 8A,B and Table 1). However, for larger instances,  $1/2$ -DISC and MILP were not able to solve the problem in the time limit. In terms of runtime, GREEDY clearly outperforms the other approaches. In fact, GREEDY solves the problem even before the Gurobi model is initialized in all 3600 instances. Even if the time for initializing the Gurobi model is omitted and the time limit of Gurobi is set to the runtime of GREEDY, GREEDY returns in 70% of all instances a solution at least as good as the incumbent solution of  $1/2$ -DISC (30% strictly better, 40% equal, 30% strictly worse). The average error of greedy to the best known solution at the time limit of 600 seconds is 38%, where the average error is given by  $\frac{\text{solution}_{\text{greedy}} - \text{solution}_{\text{incumbent}}}{\text{solution}_{\text{incumbent}}}$ . The frequency distribution of the average error is shown in Figure 8C.

<sup>1</sup>Two Intel Xeon E5-2690 (single processor specifications: nominal speed 2.9 GHz, boost up to 3.8 GHz, 16 threads, 8 cores, 16 threads, 20 MB Cache) and 192GB DDR-3 ECC RAM at 1333 MHz. Operating system environment: Linux 64bit, Kernel 3.13.0. Used software: gcc 7.3.0, Python 3.6.8, networkx 2.2 and Gurobi 8.1.0.

**TABLE 1** Average computation time (each 90 instances) in seconds

$\rho$		$ E  = 15$	$ E  = 20$	$ E  = 30$	$ E  = 40$	$ E  = 60$	$ E  = 100$	$ E  = 200$	$ E  = 300$
25	GREEDY	0.1	0.1	0.1	0.2	0.6	1.4	5.3	7.2
	1/2-DISC	0.2	0.3	3.0	2.8	22.3	94.7 <sup>(87)</sup>	216.4 <sup>(44)</sup>	537.9 <sup>(2)</sup>
	MILP	4.2	17.1 <sup>(88)</sup>	289.1 <sup>(49)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>
50	GREEDY	0.1	0.1	0.1	0.3	0.7	1.9	7.8	10.1
	1/2-DISC	0.8	1.5	7.9	15.5	85.3 <sup>(87)</sup>	156.9 <sup>(59)</sup>	357.2 <sup>(7)</sup>	— <sup>(0)</sup>
	MILP	6.4	24.2 <sup>(88)</sup>	275.7 <sup>(45)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>
100	GREEDY	0.1	0.1	0.2	0.3	0.8	2.3	13.6	16.2
	1/2-DISC	3.1	6.4	35.1	82.2 <sup>(89)</sup>	255.6 <sup>(61)</sup>	280.4 <sup>(18)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>
	MILP	6.3	16.8	306.0 <sup>(50)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>
150	GREEDY	0.1	0.1	0.2	0.3	0.8	2.5	15.6	18.2
	1/2-DISC	9.8	17.4	81.5 <sup>(88)</sup>	149.3 <sup>(81)</sup>	274.2 <sup>(53)</sup>	337.2 <sup>(15)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>
	MILP	6.6	16.8 <sup>(88)</sup>	276.2 <sup>(42)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>
200	GREEDY	0.1	0.1	0.2	0.3	0.9	2.7	17.4	19.0
	1/2-DISC	18.8	30.6	137.3 <sup>(84)</sup>	166.7 <sup>(74)</sup>	325.8 <sup>(26)</sup>	435.2 <sup>(16)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>
	MILP	10.2	14.0 <sup>(88)</sup>	304.2 <sup>(39)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>	— <sup>(0)</sup>

Note: If not all instances solved to optimality within time limit of 600 seconds, number of solved instances in brackets.

**FIGURE 8** Computational results A,  $\rho = 25$ . B,  $|E| = 30$ . C, Distribution of average error. [Color figure can be viewed at wileyonlinelibrary.com]

## 6 | CONCLUSION AND FURTHER RESEARCH

In this paper, we considered a generalization of the location set cover problem, where demand arises not only at the nodes but along the edges of a graph. The aim was to determine the minimum number of facilities in order to cover all demanding points within a given coverage radius. In order to cover an edge, multiple facilities may be used, that is, an edge may be covered complementarily. We showed that the problem, which is in general NP-hard to solve, is not even approximable within a constant factor. For general graphs, we derived a finite dominating set of exponential size, that is, a finite set containing at least one optimal solution. Although difficult to calculate, this set is a subset of the known discretization in [9] for general graphs and is a polynomially bounded set for special graph classes such as cactus graphs. Using the new structural results, a greedy algorithm based on a polynomially bounded subset of the FDS was proven to be a  $(1 + \ln(|I|))$ -approximation algorithm of the absolute complementary edge cover problem. We also proposed a mixed-integer linear programming formulation for the problem and tested the performance of greedy in a computational study. An open question is whether there exists a FDS of polynomial size for general graphs. Future research may also address variants of the problem, such as a weighted version of edge cover or the multicover of edges.

### ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments on the first version of the paper. The work on this paper was partially supported by the Bundesministerium für Bildung und Forschung (BMBF) under Grant OPMOPS, FKZ 13N14561.

## ORCID

Nicolas Fröhlich  <https://orcid.org/0000-0002-4556-1798>

## REFERENCES

- [1] G. Ausiello *et al.*, *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*, 1st edn, Springer-Verlag New York Inc., Secaucus, NJ, 1999. ISBN: 3540654313.
- [2] O. Berman, J. Kalcsics, and D. Krass, *On covering location problems on networks with edge demand*, *Comput. Oper. Res.* **74** (2016), 214–227.
- [3] O. Berman, V. Verter, and B.Y. Kara, *Designing emergency response networks for hazardous materials transportation*, *Comput. Oper. Res.* **34**(5) (2007), 1374–1388.
- [4] M.L. Brandeau, S.S. Chiu, and R. Batta, *Locating the two-median of a tree network with continuous link demands*, *Ann. Oper. Res.* **6** (1986), 223–253.
- [5] T.M. Cavalier and H.D. Sherali, *Network location problems with continuous link demands: P-medians on a chain and 2-medians on a tree*, *Eur. J. Oper. Res.* **23** (1986), 246–255.
- [6] R.L. Church and M.E. Meadows, *Location modeling utilizing maximum service distance criteria*, *Geogr. Anal.* **11** (1979), 358–373.
- [7] M.S. Daskin, *Network and Discrete Location*, John Wiley & Sons, Inc., New York, 1995.
- [8] A.J. Goldman, *Optimal center location in simple networks*, *Transport. Sci.* **5** (1971), 212–221.
- [9] Y. Gurevich, L. Stockmeyer, and U. Vishkin, *Solving NP-hard problems on graphs that are almost trees and an application to facility location problems*, *J. ACM* **31** (1984), 459–473.
- [10] S.L. Hakimi, *Optimum distribution of switching centers in a communication network and some related graph theoretic problems*, *Oper. Res.* **13** (1965), 462–475.
- [11] G.Y. Handler and P.B. Mirchandani, *Location on Networks*, MIT Press, Cambridge, MA, 1979.
- [12] J.N. Hooker, R.S. Garfinkel, and C.K. Chen, *Finite dominating sets for network location problems*, *Oper. Res.* **39** (1991), 100–118.
- [13] D.S. Johnson, “*Approximation algorithms for combinatorial problems*,” *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing. STOC '73ACM*, Austin, TX, 1973, pp. 38–49.
- [14] E. Minieka, *The centers and medians of a graph*, *Oper. Res.* **25** (1977), 641–650.
- [15] C. ReVelle, C. Toregas, and L. Falkson, *Applications of the location set-covering problem*, *Geogr. Anal.* **8** (1976), 65–76.
- [16] A.N. Sadigh, M. Mozafari, and A.H. Kashan, *A mixed integer linear program and tabu search approach for the complementary edge covering problem*, *Adv. Eng. Softw.* **41** (2010), 762–768.
- [17] H.D. Sherali and F.L. Nordai, *A capacitated, balanced, 2-median problem on a tree network with a continuum of link demands*, *Transport. Sci.* **22** (1998), 70–73.
- [18] D.P. Williamson and D.B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, Cambridge, MA, 2011.
- [19] P. Zhao and R. Batta, *An aggregation approach to solving the network p-median problem with link demands*, *Networks* **36**(4) (2000), 233–241.

**How to cite this article:** Fröhlich N, Maier A, Hamacher HW. Covering edges in networks. *Networks*. 2020;75:278–290.  
<https://doi.org/10.1002/net.21924>