Computers & Operations Research 176 (2025) 106957

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

New models for close enough facility location problems

Alejandro Moya-Martínez [a], Mercedes Landete [a,*], Juan F. Monge [a], Sergio García [b]

[a] Departamento de Estadística e Investigación Operativa, Universidad Miguel Hernández, 03202 Elche, Alicante, Spain
[b] School of Mathematics, University of Edinburgh, United Kingdom

ARTICLE INFO

Keywords:
Close enough
Facility location
Column generation

ABSTRACT

Two integer programming problems are introduced and formulated in this paper, both based on the concepts of *close enough* and facility location. Location problems using the notion of *close enough* allow customers to pick up their demand at pickup points different from the facilities but that are still not too far from the latter.

Given a discrete set of customers, a discrete set of potential facility locations, and a maximum distance that each customer is willing to travel free of charge to pick up their order, the Close Enough Facility Location Problem consists in determining which facilities to open among the candidates, on which points on the plane to install pickup points, and how to assign customers to both facilities and pickup points, in an optimal way taking into account different costs. In this work we propose two generalizations of this problem. The first is to consider that the pickup points have capacities. The second is to consider that the communications network is restricted to a graph, and that therefore the pickup points cannot be installed on any point on the plane but only on the network. These problems are named the Capacitated Close-Enough Facility Location Problem and the Network Capacitated Close-Enough Facility Location Problem, respectively. We propose a column generation algorithm for the two introduced problems that allows us to obtain better results for large-scale problems than the CPLEX solver.

1. Introduction and literature review

Location problems have a very important relevance in numerous contemporary scenarios. Logistics companies and businesses requiring the establishment of facilities or pickup points often need to devise optimization models. The main goal of these models is to obtain an optimal solution that determines both the ideal facility locations and the customers they serve. The p-median problem is one of the most studied problems in discrete optimization, since it was introduced by Hakimi (1964). Years later, the capacitated p-median problem (CPMP) was introduced and shown to be an NP-hard problem (see Hartmanis, 1982).

In the last decade, consumer behavior has undergone significant changes. For instance, many consumers now prefer picking up their orders at a convenient pickup point near their home or workplace, rather than providing their address and waiting for an alert about the delivery time. Large distribution companies are increasingly adopting this approach, utilizing service or collection points in proximity to the customer. An example of this trend is the self-service points (*lockers*) that Amazon is installing in all the cities.

The *close enough* concept used in facility location or routing problems allows for the relocation of customers. A plant is close enough to a customer if there is an intermediate point between the plant and the customer to which the customer is willing to travel to be served at no additional cost. In this case, the location problem consists of deciding where to install the plants so that all customers have a close enough plant. A vehicle is close enough to a customer when they are willing to move to the vehicle route to obtain the vehicle service. Thus, the problem consists of designing routes so that all customers have a close enough route.

Some papers about routing problems that use close enough conditions are Gulczynski et al. (2006), Corberán et al. (2019), Corberán et al. (2021), Hernández-Pérez et al. (2021), Bianchessi et al. (2022a,b), Di Placido et al. (2023) and Reula and Martí (2023). In Gulczynski et al. (2006) and Di Placido et al. (2023) the problem studied is the traveling salesman problem and heuristic approaches are proposed. The close enough traveling salesman problem is a generalization of the traveling salesman problem that requires a salesman to just go close enough to each customer instead of visiting the exact location of each customer. Over the 17 years between the publication of these two works on the close enough traveling salesman problem, more than a dozen works have been published on it. In Corberán et al. (2019), Corberán

* Corresponding author.
E-mail addresses: a.moya@umh.es (A. Moya-Martínez), landete@umh.es (M. Landete), monge@umh.es (J.F. Monge), sergio.garcia-quiles@ed.ac.uk (S. García).

https://doi.org/10.1016/j.cor.2024.106957
Received 2 January 2024; Received in revised form 13 December 2024; Accepted 14 December 2024
Available online 21 December 2024
0305-0548/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

et al. (2021), Bianchessi et al. (2022a,b) and Reula and Martí (2023) the problem studied is the arc routing problem. The close enough arc routing problem models the situation in which customers are not necessarily nodes of a network and the vehicles must traverse the edges that are close enough to the customers. Corberán et al. (2019) introduce the problem and study optimal solutions, whereas the other listed papers on the close enough arc routing problem introduce other generalizations as the profitable or the distance-constrained close enough arc routing problem. In Hernández-Pérez et al. (2021) the problem studied is the pickup and delivery problem and the authors do not explicitly say that the reallocation of customers is due to a close enough condition: they assume that customers are paid for moving to the vehicle route.

Three papers about location problems that include close enough conditions are Landete and Laporte (2019), Corberán et al. (2020) and Moya-Martínez et al. (2021). In the first paper, customers can collect their demand from some cooperative customers which act as intermediate points, whereas in the last paper customers can do it from some general pickup points. In the first paper, the number of cooperative customers is a finite set while in the latter the number of all potential candidate pickup points is an infinite set.

In this paper we assume that there is a finite set of potential plant locations and a finite set of customers who are willing to travel to any point within a radius to pick up their orders at no additional cost. We assume that a pickup point may have a capacity that limits the number of customers allocated to it and we consider two cases: the pickup points can be installed anywhere on a plane or on a graph. The problem is to determine where to install the plants and the pickup points among the potential sets such that all the customers are served, capacities are not exceeded, and the total cost is minimized. The pickup points offer an opportunity to reduce transportation costs, as customers are willing to retrieve goods from a location that is sufficiently close to the facility that they would have to go to by default. This paper builds on Moya-Martínez et al. (2021), where the close enough facility location problem was introduced. The same basic problem is addressed with two added difficulties: (i) capacities are added to the pickup points, i.e., the number of customers that can move to a certain pickup point is restricted, and (ii) pickup points are restricted to be on a graph, i.e., it is analyzed how the space of feasible solutions changes when the problem is solved on a graph. The new problems are named the Capacitated Close-Enough Facility Location Problem (CCEFLP) and the Network Capacitated Close-Enough Facility Location Problem (NCCEFLP), respectively.

The application we mentioned before of the decision on the location of lockers as pickup points fits the description of the problem. It is sensible to assume that the budget manager knows the number of lockers he can afford to install and that a more detailed budget restriction is unnecessary. It also fits the fact that the distance traveled by customers to reach the locker is different in each case, although the price of these rides is the responsibility of the customers and is not in the manager's objective function. It is also appropriate to assign a capacity to each locker. Another application of the problem related to communications rather than to transportation is the wifi router location for internet accessibility. The wifi network and the network cable connection are the two main options to connect to the network on computers and laptops. Computers (customers) can be directly served by the cable connection (facility) or a wifi router (pickup point). It usually happens that the cable service is better than the router service in terms of download/upload speed, which means that the number of devices that can connect to a router is limited. The acceptable distance between customers and pickup points depends on the quality of the router. In the event that all the routers installed are of the same quality, we will assume the best, the system manager must decide where to locate the number of routers that the budget allows.

The main contributions of this work are summarized as follows:

i. Two different extensions of the close enough facility location problem are introduced, namely, the Capacitated Close-Enough Facility Location Problem and the Network Capacitated Close-Enough Facility Location Problem.
ii. An efficient algorithm for discretizing the candidate pickup points set on a graph is proposed.
iii. A column generation algorithm for solving both the CCEFLP and the NCCEFLP is detailed.
iv. Extensive computational experiments are conducted. Instances with a number of nodes ranging from 30 to 100 are solved and the results are discussed.

The remainder of the paper is organized as follows. Section 2 introduces the CCEFLP and proposes a mathematical programming integer linear formulation. Section 3 studies the location problem on a graph. Section 4 proposes a column generation algorithm for both CCEFLP and NCCEFLP. Finally, Section 5 provides a comprehensive computational study that reports the performance of our column generation algorithm both for the CCEFLP on the plane and the CCEFLP on a graph (NCCEFLP). We conclude with the findings and conclusions of this work in Section 6.

2. The capacitated close-enough facility location problem

The Close Enough Facility Location Problem (CEFLP) seeks to minimize the distance between plants and customers or pickup points, as detailed in Moya-Martínez et al. (2021). It is assumed that the customers bear the costs between them and pickup points and therefore do not have to be taken into account in the planning of the distribution network manager. Precisely, the radius associated with each customer establishes the limit of how far he can travel at no cost to the network manager. In this work, we will address the case in which pickup points are constrained by a capacity that cannot be exceeded. This capacity restricts the number of customers that can be assigned to a pickup point. Particularly, the problem here considered involves locating p facilities, determining the location of t pickup points, assigning all customers to open facilities or open pickup points, and finally assigning open pickup points to open facilities.

Set J is the set of potential facilities and set I is the set of customers. For all $i \in I$, h_i is the demand of customer i and R_i is the maximum distance that customer i is willing to travel for picking up their demand. If R_i is the same for all the customers, then we simply represent it as R. As proved in Moya-Martínez et al. (2021), the set of potential pickup points in the CEFLP can be reduced to a finite discrete set. Set K is the finite discrete set of these potential pickup points. When the pickup points can be placed anywhere in the plane, then K is the union of circumference intersections and segment-circumference intersections. In other words, if C_i is the set of points in the circumference with center in customer i and radius R_i for all $i \in I$ and S_{ij} is the set of points in the segment joining customer i with facility j, for all $i \in I, j \in J$, then $K = \left(\bigcup_{i\in I, j\in J}(C_i \cap S_{ij})\right) \cup \left(\bigcup_{i_1,i_2\in I} C_{i_1} \cap C_{i_2}\right)$. When the pickup points must be located on a graph, K is still the finite discrete set of potential pickup points but it is obtained with the algorithm described in Section 3.

Parameter d_{tf} represents the distance between points t and f (customers/facilities / pickup points). For each $i \in I$, K_i is the subset of elements of K that are close enough to i, i.e., $K_i = \{k \in K : d_{ik} \leq R_i\}$. For the capacitated extension, we consider that there is a capacity N_k for each $k \in K$. This value is the maximum number of customers that can be allocated to the pickup point k.

In the literature there are two known mixed integer linear models for the non-capacitated CEFLP, a two-index formulation and a three-index formulation. In Moya-Martínez et al. (2021) it is shown that a branch-and-price algorithm for the three-index formulation is the best way of solving the CEFLP. In the capacitated CEFLP that we introduce in this paper we keep the same notation as in the three index model for the CEFLP and we add the capacity constraint. In particular, we consider three families of binary variables. For each $j \in J$, y_j takes value 1 if and only if facility j is open. For each $k \in K$, v_k takes value 1 if and only if a pickup point is open at location k. For each $i \in I, k \in K, j \in J$, variable w_{ikj} takes value 1 if and only if customer i goes to pickup point k that in turn is allocated to facility j: $w_{iij} = 1$ represents that customer i does not go to any pickup point and is allocated to a facility.

Capacity constraints in terms of these binary variables are:

$$\sum_{i \in I: d_{ik} \le R_i} \sum_{j \in J} w_{ikj} \le N_k v_k \qquad \forall k \in K.$$

We propose to model the CCEFLP by adding capacity constraints to the CEFLP three-index formulation, thus obtaining the following model:

$$\text{(CCEFLP)} \quad \min \sum_{i \in I} \sum_{k \in K_i \cup \{i\}} \sum_{j \in J} h_i d_{kj} w_{ikj} \tag{1}$$

$$s.t. \sum_{j \in J} y_j = p, \tag{2}$$

$$\sum_{k \in K} v_k = t, \tag{3}$$

$$\sum_{j \in J} \sum_{k \in K_i \cup \{i\}} w_{ikj} = 1 \qquad \forall i \in I, \tag{4}$$

$$\sum_{j \in J} w_{ikj} \le v_k \qquad \forall i \in I,\ \forall k \in K_i, \tag{5}$$

$$\sum_{k \in K_i \cup \{i\}} w_{ikj} \le y_j \qquad \forall i \in I,\ \forall j \in J, \tag{6}$$

$$\sum_{i \in I: d_{ik} \le R_i} \sum_{j \in J} w_{ikj} \le N_k v_k \qquad \forall k \in K, \tag{7}$$

$$w_{ikj} \in \{0.1\} \qquad \forall i \in I,\ \forall k \in K_i \cup \{i\},\ \forall j \in J, \tag{8}$$

$$y_j \in \{0.1\} \qquad \forall j \in J, \tag{9}$$

$$v_k \in \{0.1\} \qquad \forall k \in K. \tag{10}$$

The goal of the CCEFLP is to minimize the total cost. Note that the objective function does not consider the distance between a customer and pickup locations (i.e., d_{kj}), but only from a customer or pickup point (i.e., $k \in K_i \cup i$) to a plant (i.e., $j \in J$). Constraints (2) and (3) impose the number of open facilities and open pickup points, respectively. Constraints (4) guarantee that all the customers are served by one facility or go to a pickup point that is close enough. Constraints (5) and (6) enforce that customers only go to close enough open pickup points and are only allocated to open facilities. Capacity constraints (7) limit the number of customers going to an open pickup point. Constraints (8)–(10) are the domain constraints.

3. The CCEFLP on graphs

The CEFLP is a continuous facility location problem since the pickup points can be placed anywhere within a fixed radius from the customers. A useful property of the CEFLP is that the continuous potential pickup location feasible set can be reduced to a discrete pickup location optimal set. In an optimal solution of the CEFLP some customers may have to go the maximum distance that they are willing to go to pick up their demand. In the CEFLP on a graph (or NCCEFLP), the discrete potential pickup location set is not the same as in the CEFLP on the plane. The border set of a customer is not a circumference anymore but a set of points over some edges of the graph. In this section we describe how to discretize the optimal set of pickup points when the problem is solved on a graph.

The formulation for this extension is the formulation for the CEFLP if capacities do not apply, or the formulation in the previous section if dealing with the capacitated case. The difficult part is the calculation of K.

Church and Meadows (1979) deal with the location set-covering problem and the maximal covering location problem when facility placement is allowed anywhere on a graph. The authors prove that for either the first or the second problem at least one optimal solution exists that is composed entirely of points belonging to a finite set of points called the graph intersect point set. However, given a graph the intersect point set does not coincide with the pickup point set.

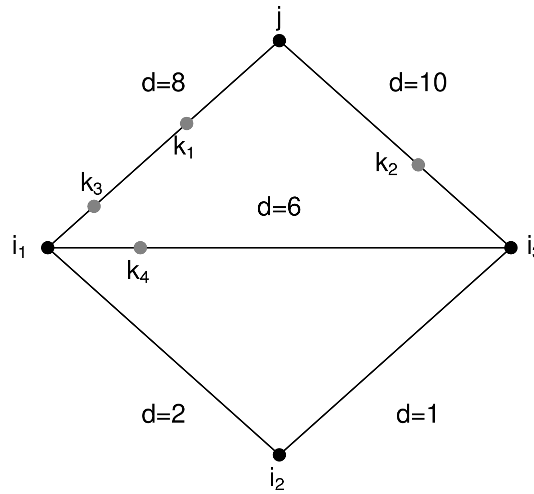


Fig. 1. Different pickup points on a graph.

A good guess in order to decide where to place the optimal pickup points would be to place them on the shortest path between customers and facilities. However, this is not always the correct answer.

Example 1. Let us consider the situation shown in Fig. 1. There are three customers i_1, i_2, and i_3, with unitary demand and one potential facility j. We would like to open one facility ($p = 1$) and one pickup point ($t = 1$) with a total minimum cost when customers agree to travel up to $R = 4$ distance units. We want to illustrate that opening a pickup point that is not in any of the shortest paths connecting a customer with the facility can be better than opening a pickup point in one of these shortest paths. The shortest path from j to i_1 is (i_1, j) and the candidate location for a pickup point in this path (at distance 4 from i_1) is k_1. The shortest path from j to i_2 is $(i_2, i_1), (i_1, j)$ and the candidate location for the pickup point in this path at distance 4 from i_2 is k_3. The shortest path from j to i_3 is (i_3, j) and the candidate location for the pickup point in this path at distance 4 from i_3 is k_2. k_4 is a candidate location for a pickup point that does not belong to any of the mentioned three shortest paths but that can act as a pickup point for the three customers: it is at distance 2 from i_1, at a distance 4 from i_2 and also at a distance 4 from i_3. In fact, $K_{i1} = \{k_1, k_3, k_4\}$, $K_{i_2} = \{k_3, k_4\}$ and $K_{i_3} = \{k_2, k_4\}$. If $N_1 = N_2 = N_3 = 3$, then the manager of the network will open the pickup point at k_4 because then, his costs will reduce to the traveling cost from j to k_4, i.e., 10. Remind that i_1, i_2 and i_3 bear the costs between them and k_4.

Following Example 1, in order to calculate all the candidate pickup points on a network for a customer i, we need to evaluate all the points on a network at distance R_i from customer i. Algorithm 1 is used for calculating all the pickup points on a network. First, $Pickup\_Points(i, l, \hat{R}_i)$ is a recursive function that obtains candidate pickup points from node i, that is, all the points on the graph at a distance $\hat{R}_i$ from node i. Note that node l collects the nodes visited by node i on its path to the pickup point k, and $\hat{R}$ collects the remaining of R_i from l to j on the path from i to j. Finally, function $Create\_Pickup\_Point(i, l, j, \overline{d}_{lk}, \overline{d}_{km})$ creates the pickup point k and saves its location within the network. Note that $\tilde{d}_{lm}$ is the distance on the graph for edge (l, m), and d_{ij} is the distance from i to j in the graph. In addition, $\overline{d}_{lk}$ and $\overline{d}_{km}$ are the distances between the pickup point generated k and the edge nodes (l, m). Once the set of points K and the distances on the graph have been calculated we can solve the CCEFLP. Fig. 2 shows how the algorithm works to obtain

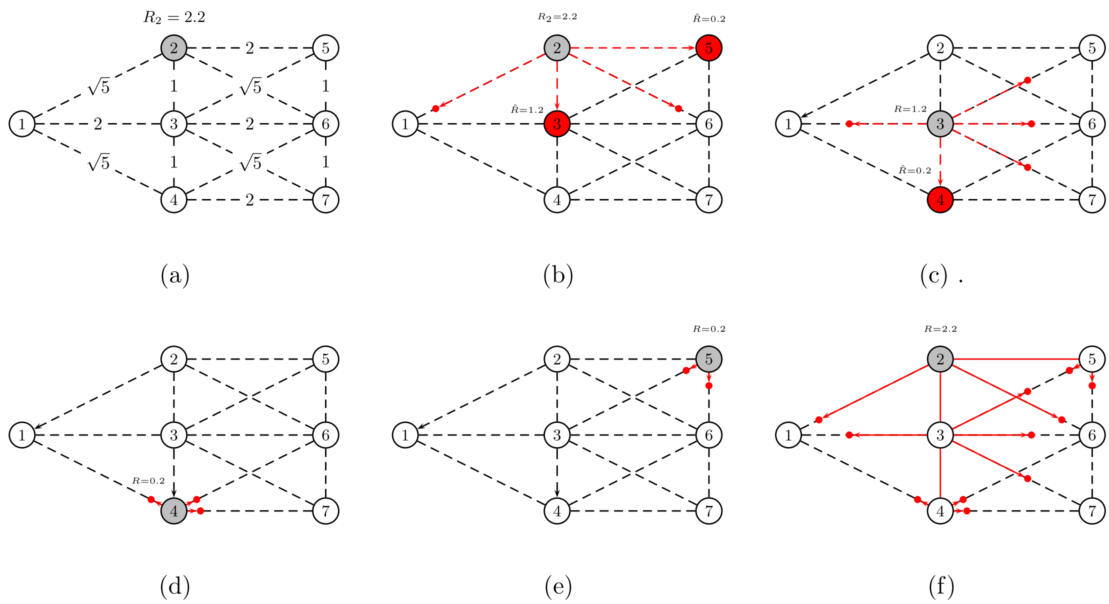


Fig. 2. Small illustrative example for Algorithm 1 .

the candidate pickup points from node 2. The algorithm starts from node 2 (Fig. 2(a)), traversing all edges incident to this node, until it reaches the maximum allowed distance for this node (R_2=2.2) or until it encounters another node (Fig. 2(b)). If a node is reached from node 2 (nodes 3 and 5), then the algorithm is repeated from these nodes, with the maximum distances updated (R_3=1.2 and R_5=0.2) (Figs. 2 (c) and 2 (d) from node 3, and 2 (e) for node 4. Fig. 2(f)) shows all the points generated from node 2 when the algorithm finishes.

Algorithm 1: Calculating pickup points on a network.

$G := (V, E)$;
$K := \emptyset$;
$k = 0$;
forall $i \in V$ **do**
  $K_i := \emptyset$;
  $Pickup\_Points(i, i, R_i)$;
Recursive Function *Pickup_Points(i,l,$\hat{R}$)*
  forall $m \in V$ **do**
    if $(m \neq l$ *and* $m \neq i$ *and* $\tilde{d}_{lm} : (l, m) \in E)$ **then**
      if $\tilde{d}_{lm} \geq \hat{R}$ **then**
        $k++$;
        $\overline{d}_{(m,k)} := \hat{R}$;
        $\overline{d}_{(k,m)} := \tilde{d}_{lm} - \hat{R}$;
        $Create\_Pickup\_Point(i, l, m, \overline{d}_{lk}, \overline{d}_{km})$;
      else
        $Pickup\_Points(i, m, \hat{R} - \tilde{d}_{lm})$
Function *Create_Pickup_Point($i, l, m, \overline{d}_{lk}, \overline{d}_{km}$)*
  $K := K \cup \{k\}$;
  $K_i := K_i \cup \{k\}$;
  pickup point k located in $(l, m) \in E$;
  forall $j \in V$ **do**
    $d_{kj} := \min\{d_{jl} + \overline{d}_{lk}, d_{mj} + \overline{d}_{km}\}$;

Proposition 3.1. *Let (x^*, y^*) be the coordinates for a pickup point k in an optimal solution for the CCEFLP. There exits a customer i which is at distance R_i from (x^*, y^*).*

Proof. Let I_k be the subset of customers in I that are close enough to k, i.e., $I_k = \{i \in I : d_{ik} \leq R_i\}$. $I_k \neq \emptyset$ because k belongs to the optimal solution. We will prove that, if there is not a customer at maximum distance, then there is a feasible solution cheaper than the optimal, which cannot be.

Let us suppose that $d_{ij} < R_i$ for all $i \in I_k$. Let i_1 be the customer with $R_{i_1} - d_{i_1 k} = \min_i\{R_i - d_{ik} : i \in I_k\}$ and let j be the facility to which k is allocated. Let $P_{i_1 j} = (i_1, k) \cup (k, j)$ be the path connecting i_1 and j, where (i_1, k) and (k, j) are the corresponding shortest paths. Let k_1 be the point in the path $P_{i_1 j}$ such that $d_{i_1 k_1} = R_{i_1}$. Then, the solution to CCEFLP that exchanges k by k_1 is feasible and cheaper than the optimal. It is feasible because

$$I_{k_1} = \{i \in I : d_{ik_1} = d_{ik} + R_{i_1} - d_{i_1 k} \leq d_{ik} + R_i - d_{ik} = R_i\} = I_k,$$

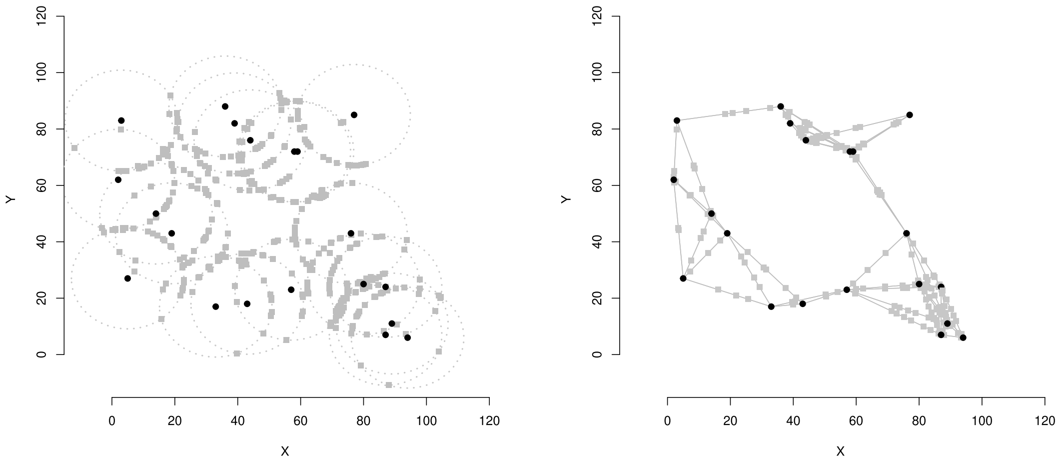
and it is cheaper than the optimal because $d_{jk_1} = d_{jk} - (R_{i_1} - d_{i_1 k}) < d_{jk}$ and the distances between customers in I_k and k_1 are not in the objective function. □

Example 2. Let us consider the complete graph (location on the plane problem) with 20 nodes induced by the example in Moya-Martínez et al. (2021) and the restricted graph (location on the network) when the maximum distance between nodes is limited by 35.13 units (20% of the maximum distance between two nodes on the plane). Fig. 3 illustrates the candidate pickup points on the plane and on the network in this example. The number of pickup points are 442 and 246 for the plane and network, respectively. Note that the number of candidate pickup points on the plane is greater than the number of candidate pickup points on the network in this example, but, as will be shown in Section 5, when the number of customers in the problem increases, this is not necessarily true.

4. Column generation algorithm

The CCEFLP model in Section 2 has a high number of variables with regard to the number of constraints, although it can be checked that the linear relaxation gap is usually small. For this reason, a column generation algorithm is proposed here to solve the problem. The column generation algorithm in this section allows to solve both CCEFLP and NCCEFLP on graphs since this fact only affects the calculation of K.

Let LR-CCEFLP be the linear relaxation of CCEFLP, i.e., the linear model obtained from CCEFLP when replacing the binary domain



(a) On the plane. (b) On the network.

Fig. 3. Candidate pickup points on the plane and on the network.

constraints by bound constraints imposing that the values of variables y, and v are in the interval [0.1]. Note that variables w are restricted in the interval [0.1] by the constraint (6). Let $(\delta, \theta, \alpha_i, \beta_{ik}, \gamma_{ij}, \tau_k)$ be the dual variables of constraints (2), (3),(4), (5), (6) and (7), respectively; and (m_j, n_k) the dual variables of the constraints y, and v less than or equal to one, respectively. The dual problem for LR-CCEFLP is written as follows:

$$
\begin{array}{llll}
\text{(D-LR-CCEFLP)} & \max & p\delta + t\theta + \sum_{i\in I}\alpha_i + \sum_{j\in J} m_j + \sum_{k\in K} n_k & \\
& \text{s.t.} & \alpha_i + \tau_k + \beta_{ik} + \gamma_{ij} \le h_i d_{kj} & \forall j \in J,\ \forall i \in I,\ \forall k \in K_i \cup \{i\}, \\
& & \delta - \sum_{i\in I}\gamma_{ij} + m_j \le 0 & \forall j \in J, \\
& & \theta - N_k\tau_k - \sum_{i\in I_k}\beta_{ik} + n_k \le 0 & \forall k \in K, \\
& & \tau_k \le 0 & \forall k \in K, \\
& & \beta_{ik} \le 0 & \forall k \in K,\ \forall i \in I_k, \\
& & \gamma_{ij} \le 0 & \forall i \in I,\ \forall j \in J, \\
& & m_j \le 0 & \forall j \in J, \\
& & n_k \le 0 & \forall k \in K, \\
& & \alpha_i \text{ free} & \forall i \in I, \\
& & \delta, \theta. \text{ free.} &
\end{array}
$$

Let $\hat{K} \subset K$ be a subset of pickup points and let Re-D-LR-CCEFLP be the above dual model D-LR-CCEFLP restricted to $\hat{K}$. Let also Re-LR-CCEFLP be the primal model LR-CCEFLP restricted to $\hat{K}$. Let $(\delta^*, \theta^*, \alpha^*, \beta^*, \gamma^*, \tau^*, m^*, n^*)$ be an optimal solution to Re-D-LR-CCEFLP. For all $k \in K \setminus \hat{K}$, the reduced cost $\tilde{c}_k$ of column v_k is

$$\tilde{c}_k = N_k\tau_k + \sum_{i\in I_k}\beta_{ik} - \theta^* - n_k,$$

where

$$\tau_k + \beta_{ik} \le \min_{j\in J}\{h_i d_{kj} - \alpha_i^* - \gamma_{ij}^*, 0\} \quad \forall i \in I_k.$$

Note that, for all $k \in K \setminus \hat{K}$, the value of n_k is equal to zero, because v_k is not part of the basic solution from $\hat{K}$.

The pricing sub-problem for obtaining the maximum value for the reduced cost is:

$$
\begin{array}{llll}
\bar{c}_k = & \max & N_k\tau_k + \sum_{i\in I_k}\beta_{ik} - \theta^* & (11) \\
& \text{s.t.} & \tau_k + \beta_{ik} \le \min_{j\in J}\{h_i d_{kj} - \alpha_i^* - \gamma_{ij}^*, 0\} & \forall i \in I_k, \\
& & \tau_k \le 0. & \\
& & \beta_{ik} \le 0 & \forall i \in I_k.
\end{array}
$$

The following proposition gives the optimal value for this pricing problem. The proof of the proposition shows that this optimal value is the sum of the N_k smallest values of a list.

Proposition 4.1. *Let $(\delta^*, \theta^*, \alpha^*, \beta^*, \tau^*, \gamma^*)$ be the optimal solution of Re-D-LR-CCEFLP for $k \in \hat{K} \subset K$. For each $k \in K$, let $I_k = \{i \in I : d_{ik} \le R_i\}$ be the set of customers that could go to the pickup point k. For each $i \in I$, let $a_{ik} = \min_{j\in J}\{h_i d_{kj} - \alpha_i^* - \gamma_{ij}^*, 0\}$. And let $\tilde{a}_{1k} \le \tilde{a}_{2k} \le \ldots \le \tilde{a}_{|I_k|,k}$ be values a_{ik} sorted in nondecreasing order. Then the maximum reduced cost $\bar{c}_k$ of column v_k is $\bar{c}_k = \sum_{i=1}^{\min\{N_k,|I_k|\}} \tilde{a}_{ik} - \theta^*$.*

Proof. The dual problem of

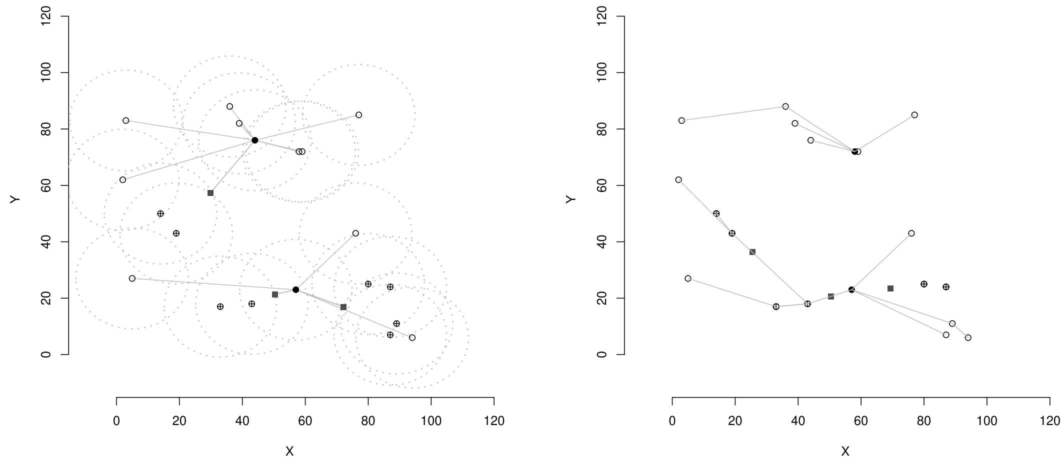
$$
\begin{array}{llll}
(P_1) & \max & N_k\tau_k + \sum_{i\in I_k}\beta_{ik}, & \\
& \text{s.t.} & \tau_k + \beta_{ik} \le \min_{j\in J}\{h_i d_{kj} - \alpha_i^* - \gamma_{ij}^*, 0\} & \forall i \in I_k, \\
& & \tau_k \le 0. & \\
& & \beta_{ik} \le 0 & \forall i \in I_k.
\end{array}
$$

is

$$
\begin{array}{llll}
(P_2) & \min & \sum_{i\in I_k} a_{ik}v_{ik} & \\
& \text{s.t.} & \sum_{i\in I_k} v_{ik} \le N_k, & \\
& & v_{ik} \le 1 & \forall i \in I_k.
\end{array}
$$

The optimal value for the latter is the sum of the N_k smallest a_{ik} values, provided that $N_k \le |I_k|$. In other words, the optimal value of (P_2) is $\sum_{i=1}^{\min\{N_k,|I_k|\}} a_{ik}$. According to the strong duality condition, it implies that the optimal value of (P_1) is the same and thus $\bar{c}_k$ is $\sum_{i=1}^{\min\{N_k,|I_k|\}} a_{ik} - \theta^*$. □

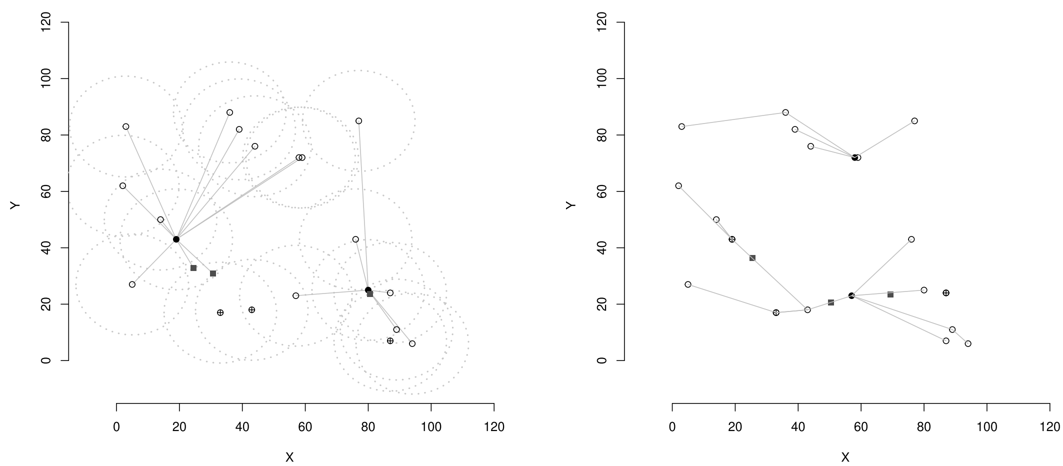
$\bar{c}_k$ is an estimation of the improvement on the objective function if pickup point k is introduced in Re-LR-CCEFLP. If $\bar{c}_k \ge 0$ for all $k \in K \setminus \hat{K}$, then the current solution of Re-LR-CCEFLP is also optimal for LR-CCEFLP and the column generation approach finishes. Otherwise each negative value proposes the addition of a new column (variable). In each iteration the optimal value of Re-LR-CCEFLP not only gives an upper bound of the optimal value of LR-CCEFLP, but also a lower bound of it. The optimal value of LR-CCEFLP cannot be reduced more than the smaller reduced cost $\bar{c}_k$ for each customer i if $k \in K_i$, hence



(a) On the plane. $v^* = 3,629$.

(b) On the network. $v^* = 4,379$.

Fig. 4. Optimal solution for the uncapacitated problem. The black circles and the gray squares represent the location of the plants and pickup points, respectively. The white and cross-out circles represent the location of the customers who are served from the plants and the customers who move to the pick up points, respectively.



(a) On the plane. $v^* = 4,250$.

(b) On the network. $v^* = 4,698$.

Fig. 5. Optimal solution for the capacitated problem. The black circles and the gray squares represent the location of the plants and pickup points, respectively. The white and cross-out circles represent the location of the customers who are served from the plants and the customers who move to the pick up points, respectively.

$$LB = v^*(\text{Re-LR-CCEFLP}) + \sum_{i \in I} \min_{k \in K_i} \bar{c}_k \leq v^*(\text{LR-CCEFLP}).$$

Obviously, the solution of Re-LR-CCEFLP is an upper bound of v^*(LR-CCEFLP).

The column generation algorithm that we propose in this section starts with a small subset of pickup points $\hat{K}$ and adds at most one pickup point for each customer at each iteration. For each customer i, the pickup point within the distance R_i that is considered to be added to $\hat{K}$ is the one with smallest reduced cost. If this minimum reduced cost is non-negative, no pickup point is added. If the same index k is the one with the smallest reduced cost for two different customers, then it is introduced only once. If $\bar{c}_k \geq 0$ for all $k \in K \setminus \hat{K}$, then the optimal solution of Re-D-LR-CCEFLP is an optimal solution of D-LR-CCEFLP. Otherwise, pickup points with negative reduced costs must be included in $\hat{K}$ and the algorithm continues. The details of the column generation algorithm that we propose are given in Algorithm 2.

Algorithm 2 initializes the subset of pickup points as an arbitrary small set, the relative gap and the upper bound (UB) to infinity. The threshold for the gap is set to 0.01 and it is represented with ϵ. The stop criterion is the gap measured as the relative difference between the upper and lower bounds. In each iteration of the algorithm, as long as this gap is larger than ϵ, Re-LR-CCEFLP for $\hat{K} \subset K$ is solved to optimality and we compute its dual optimal variables $(\delta^*, \theta^*, \alpha^*, \beta^*, \gamma^*, \tau^*)$. For all k in $K \setminus \hat{K}$ the reduced cost is computed as indicated in Proposition 4.1. For each customer i, the pickup point k'_i within its radius with smallest reduced cost is added to $\hat{K}$. Finally, lower and upper bounds are updated. The optimal value of Re-LR-CCEFLP, named v^*(Re-LR-CCEFLP) is an upper bound for LR-CCEFLP, and, the value v^*(Re-LR-CCEFLP)+ $\sum_{i \in I} \bar{c}_{k'(i)}$ is a lower bound for the same LR-CCEFLP.

When the algorithm ends, problem CCEFLP restricted to $K = \hat{K}$, which we will refer to as Re-CCEFLP, is solved and the lower bound of LR-CCEFLP is reported as a lower bound of CCEFLP.

Example 2 (*continued*). Figs. 4 and 5 present the optimal solution for CEFLP on the plane or on a graph when the number of facilities and

Table 1
Average dimensions of the three instances for problems CCEFLP and NCCEFLP.

Instances	n	CCEFLP			NCCEFLP 20%			NCCEFLP 30%		
		#K	n01	m	#K	n01	m	#K	n01	m
i30	30	1058.00	128 828.00	6247.00	–	–	–	788.67	88 138.67	4658.33
i35	35	1435.33	227 313.67	9149.00	482.50	79 442.50	4031.50	1700.67	246 210.67	9979.67
i40	40	1882.00	374 028.67	12 825.67	837.00	166 477.00	6656.00	2682.33	481 602.33	16 333.33
i45	45	2407.33	610 012.33	17 979.67	1873.67	475 828.67	14 519.00	4520.33	988 595.33	28 501.67
i50	50	3054.67	1076304.67	27 069.67	3330.00	1046530.00	26 792.00	10 129.67	3207046.33	76 666.00
i55	55	3690.00	1520205.00	34 343.00	4217.50	1468097.50	33 966.50	13 458.33	5105065.00	109 166.00
i60	60	4386.00	2107846.00	43 103.67	6584.50	2747234.50	55 980.00	17 105.38	6995073.08	137 122.85
i65	65	5131.33	2817963.00	52 695.67	11 843.33	6237283.33	111 972.33	24 026.57	11342857.29	202 515.43
i70	70	5952.86	3737092.86	64 224.86	14 492.67	8421212.67	139 626.67	30 296.27	15800990.82	260 629.91
i75	75	6796.00	4683758.50	74 855.50	17 333.93	10832607.14	167 310.57	35 943.78	20082218.78	309 000.44
i80	80	7770.55	6222708.73	91 937.27	24 916.36	17624930.91	251 474.55	51 172.00	32235260.89	460 031.11
i85	85	8783.27	7905661.91	108 997.73	34 731.00	27275335.50	362 601.70	73 211.75	53637321.75	710 770.75
i90	90	9885.00	10376580.00	133 260.50	57 369.80	56171757.80	689 141.00	122 112.00	112363567.71	1377517.29
i95	95	11 040.00	12958067.50	156 444.50	69 124.55	70350884.55	818 145.55	175 195.25	170252167.33	1974692.17
i100	100	12 196.50	15036446.50	172 539.00	130 643.86	151628351.00	1655818.93	198 120.00	183169520.00	2038032.00

Table 2
Average results of the 3 data sets in the p-median problem.

Instance	p-median			p-median 20%			p-median 30%		
	Obj	Time	%GAP	Obj	Time	%GAP	Obj	Time	%GAP
i30	5196,80	1,20	0.00				5292,67	3	0.00
i35	6234,78	25,00	0.03	7500.00	4	0.00	6345,33	4	0.00
i40	7182,95	1,20	0.00	8597,00	1	0.00	7333,33	3	0.00
i45	8165,19	32,60	0.02	8821,67	5	0.00	8436,67	231	0.07
i50	9193,21	1,80	0.00	10403,50	5	0.00	9510.00	12	0.00
i55	10162,58	2,00	0.00	10848,50	5	0.00	10378,00	19	0.00
i60	11117,83	4,00	0.00	11662,00	183	0.24	11322,00	29	0.00
i65	12032,50	4,25	0.00	12291,00	165	0.16	12027,67	42	0.00
i70	12822,83	5,00	0.00	13001,00	32	0.00	12788,00	56	0.00
i75	13807,24	6,67	0.00	14043,67	43	0.00	13883,67	76	0.00
i80	14339,97	8,33	0.00	14751,33	68	0.00	14547,33	134	0.00
i85	15302,05	10.75	0.00	15635,67	114	0.07	15362,33	206	0.00
i90	16149,21	12,67	0.00	16369,33	185	0.00	26979,33	373	0.00
i95	17157,78	15,50	0.00	17278,67	292	0.00	17352,50	295	0.00
i100[a]	18109,44	17,75	0.00	18597,50	227	0.00			

[a] Out of memory for the p-median 30% problems.

pickup points is limited to 2 and 3, respectively. The capacities of the pickup points are infinity in Fig. 4 and 1 in Fig. 5.

Algorithm 2: Column generation algorithm

initialization: $\hat{K} = K_0$, GAP=1. $UB = \infty$, $\epsilon = 0.01$;
while $GAP > \epsilon$ **do**
  Solve Re-LR-CCEFLP (Re-D-LR-CCEFLP) for subset $\hat{K} \subset K$;
  Result: Primal (y^*, v^*, w^*) and dual $(\delta^*, \theta^*, \alpha^*, \beta^*, \gamma^*, \tau^*)$ solutions;
  forall $k \in K \setminus \hat{K}$ **do**
    forall $i \in I$ **do**
      $a_{ik} = \min_{j\in J}\{h_i d_{kj} - \alpha_i^* - \gamma_{ij}^*\}$;
    $\bar{c}_k = \sum_{i=1}^{\min\{N_k, |I_k|\}} \tilde{a}_{ik} - \theta^*$;
  forall $i \in I$ **do**
    $k_i' = \arg\min_{k\in K_i}\{\bar{c}_k : \bar{c}_k < 0\}$;
    Update $\hat{K} := \hat{K} \cup \{k'\}$;
  UB=v^*(Re-LR-CCEFLP);
  LB=v^*(Re-LR-CCEFLP)+ $\sum_{i\in I} \bar{c}_{k_i'}$;
  $GAP = \frac{UB-LB}{LB}$;
Solve(Re-CCEFLP);
$GAP = \frac{v^*(Re-CCEFLP)-LB}{LB}$;

5. Computational results

In this section we present a computational study for CCEFLP and NCCEFLP. The experiment has been coded in C++, using IBM ILOG CPLEX Optimization Studio v20.0. The characteristics of the computer are 2 Intel(R) Xeon(R) 3.10 GHz, and 768 GB RAM.

The data were taken from the instances in p-medcap1.txt of the problem solved in Osman and Christofides (1994), originally generated for the p-median problem. This file includes 60 different problems and we have selected the first three datasets of size 100. Then, we have generated 15 instances of each dataset, varying the size of n from 30 to 100. increasing by 5 in each case; resulting in a total number of 45 instances. Each instance has been solved with 5 different capacities. In all instances $I = J$, the customers are potential sites for locating a facility, and the radius will remain constant. We assume that customers are willing to go up to 15% of the maximum Euclidean distance between pairs of points in the data. In order to generate the instances on a network, the edges with length less than 20% or 30% of the largest distance in the complete graph have been selected. For all the cases p (number of facilities) and t (number of pickup points) are fixed to 4 and 5 respectively.

Table 1 shows the average number of pickup points generated ($\#K$) for each set of instances with the same size, the average number of binary variables ($n01$), and the average number of constraints (m). We rule out the three i30 instances in NCCEFLP 20% because the three are disconnected graphs. The main conclusion we can draw from Table 1 is that the number of pickup points increases significantly when considering the problem on a network compared to the plane, and it also increases significantly with the network density. This increase in the number of pickup points implies an increase in the number of variables and constraints, therefore, an increase in the dimension of the problem.

Table 3
Average results of the 3 data set in CCEFLP.

Inst	Cap	CCEFLP						z_{LB}	CCEFLP Algorithm				
		Obj	Time	%GAP	$P(x_{ikj})$	$P(d_{ikj})$	z*		Iter	Time	#%k	$\%GAP_{LB}$	GR
i30	∞	2675.31	2.67	2.16	0.60	0.48	2675.30	2608.41	4.33	4.67	9.98	2.26	1.00
i30	10	2675.31	4.33	2.16	0.60	0.48	2675.30	2603.75	4.00	4.67	10.01	2.43	1.00
i30	5	2675.31	4.00	2.07	0.60	0.43	2979.25	2886.48	4.00	4.33	9.87	3.12	0.90
i30	4	2694.13	3.67	1.92	0.59	0.45	2816.53	2748.47	4.00	3.00	9.17	2.24	0.96
i30	3	2816.53	3.67	1.96	0.51	0.52	2390.19	2338.97	4.00	3.33	9.78	1.88	1.18
i35	∞	3302.74	9.00	4.22	0.67	0.36	3305.24	3161.05	4.00	16..33	8.12	4.33	1.00
i35	10	3302.74	15.33	4.22	0.67	0.36	3307.84	3177.05	4.33	21.67	8.33	3.88	1.00
i35	5	3307.60	12.00	3.72	0.66	0.36	3302.74	3160.26	4.33	15.67	8.54	4.27	1.00
i35	4	3390.01	11.67	3.80	0.59	0.31	3599.35	3487.69	4.00	9.67	8.17	2.92	0.94
i35	3	3598.59	10.33	2.54	0.47	0.26	3394.00	3250.51	4.00	13.67	8.26	4.06	1.06
i40	∞	3943.33	19.67	4.89	0.65	0.22	3943.80	3720.26	4.00	24.00	7.55	5.31	1.00
i40	10	3943.33	44.33	4.89	0.65	0.21	3990.68	3790.39	4.33	29.00	7.56	4.71	0.99
i40	5	3990.20	38.33	3.83	0.57	0.22	3943.80	3765.84	4.33	29.00	7.56	4.29	1.01
i40	4	4134.44	29.33	3.62	0.53	0.14	4402.19	4305.51	3.67	14.33	6.48	2.19	0.94
i40	3	4397.30	24.00	1.97	0.42	0.24	4135.91	3975.44	4.00	20.67	6.99	3.79	1.06
i45	∞	4497.22	77.33	3.76	0.62	0.14	4513.81	4303.81	4.33	63.33	6.87	4.44	1.00
i45	10	4497.22	217.67	3.76	0.62	0.14	4620.65	4433.70	4.67	67.67	7.34	3.74	0.97
i45	5	4613.13	129.33	2.54	0.53	0.14	4522.88	4347.79	4.33	71.33	6.58	3.71	1.02
i45	4	4797.00	84.33	2.05	0.46	0.12	5161.92	5076.41	4.00	19.33	5.80	1.53	0.93
i45	3	5157.50	64.00	1.43	0.37	0.18	4838.12	4678.41	4.00	33.00	6.19	2.94	1.07
i50	∞	4728.28	193.33	3.82	0.67	0.32	4734.95	4523.43	4.67	149.33	6.64	4.28	1.00
i50	10	4733.28	535.67	3.87	0.66	0.29	5105.11	4928.07	4.33	174.00	5.92	3.26	0.93
i50	5	5143.92	386.00	2.71	0.54	0.15	4767.08	4597.61	4.67	180.00	6.32	3.48	1.08
i50	4	5478.38	390.67	1.42	0.43	0.16	5970.10	5902.85	3.67	62.33	5.37	1.13	0.92
i50	3	5970.04	266.67	0.91	0.33	0.13	5488.83	5356.55	3.67	101.33	5.39	2.35	1.09
i55	∞	5135.23	393.33	2.64	0.68	0.27	5167.10	5004.67	4.33	183.33	5.26	3.19	0.99
i55	10	5166.95	1289.33	2.94	0.66	0.24	5666.01	5485.13	4.33	273.33	5.03	3.19	0.91
i55	5	5842.74	816.00	2.84	0.48	0.20	5319.63	5177.88	4.33	220.00	5.39	2.70	1.10
i55	4	6194.33	740.67	1.27	0.39	0.08	6749.96	6677.67	4.00	66.33	4.46	1.03	0.92
i55	3	6743.82	688.00	0.89	0.29	0.10	6194.32	6111.98	4.00	108.00	4.87	1.31	1.09
i60	∞	5663.14	1090.33	3.47	0.66	0.24	5711.82	5494.22	4.67	509.33	5.39	3.87	0.99
i60	10	5711.83	2647.33	3.81	0.63	0.21	6300.55	6108.38	4.67	377.67	4.94	3.04	0.91
i60	5	6513.32	1463.33	2.32	0.45	0.21	5881.35	5708.71	5.00	439.67	5.13	3.08	1.11
i60	4	7024.71	1129.67	1.86	0.36	0.15	7643.29	7560.82	3.00	103.00	3.68	1.09	0.92
i60	3	7639.73	1046.67	0.64	0.27	0.11	7030.47	6894.86	4.00	146.67	4.01	1.98	1.09
i65	∞	6034.24	3041.67	1.79	0.66	0.23	6083.50	5953.71	4.67	476.00	4.53	2.11	0.99
i65	10	6083.70	7461.00	1.86	0.63	0.19	6861.98	6701.42	4.00	377.33	4.20	2.37	0.89
i65	5	7126.11	3773.67	2.08	0.41	0.20	6315.53	6193.90	4.33	490.33	4.37	1.88	1.13
i65	4	7604.91	3406.00	1.07	0.33	0.10	8307.40	8200.60	3.33	225.00	3.36	1.26	0.92
i65	3	8307.40	3113.00	0.98	0.25	0.09	7604.91	7518.40	3.67	223.00	3.51	1.17	1.09
i70	∞	6480.48	6449.00	1.89	0.65	0.21	6574.82	6382.97	4.67	1265.67	4.55	2.78	0.99
i70	10	6564.54	13 131.67	2.27	0.61	0.18	7450.12	7276.67	4.00	937.67	3.98	2.33	0.88
i70	5	7733.52	7269.33	1.25	0.38	0.13	6793.55	6677.64	4.00	892.67	3.88	1.73	1.14
i70	4	8228.14	6187.67	0.45	0.30	0.09	8967.61	8896.28	3.67	248.33	3.39	0.73	0.92
i70	3	8905.55	4469.00	0.52	0.23	0.08	8228.13	8176.21	3.67	403.67	3.38	0.60	1.08
i75	∞	8622.19	3220.50	14.38	0.63	0.36	7139.95	6879.16	5.00	2373.00	4.36	3.40	1.21
i75	10	7769.98	6594.00	7.27	0.56	0.15	8128.24	7948.37	4.67	1013.67	3.68	2.19	0.96
i75	5	8790.86	3785.00	2.93	0.34	0.10	7443.63	7324.23	5.00	1344.00	4.01	1.44	1.18
i75	4	9144.44	2805.50	0.32	0.28	0.08	9803.59	9732.64	3.00	361.00	2.96	0.74	0.93
i75	3	9833.62	1078.50	0.08	0.21	0.09	9060.60	9015.39	3.67	584.33	3.18	0.51	1.09
i80	∞	8229.74	12 669.00	10.12	0.64	0.32	7233.72	6982.13	5.00	3860.50	4.13	3.47	1.14
i80	10	12 541.87	14 411.50	100.00	0.44	0.25	8692.57	8546.86	4.33	2719.33	3.63	1.71	1.44
i80	5	9144.75	6029.00	0.89	0.33	0.07	9181.13	9064.76	4.00	979.00	3.48	1.25	1.00
i80	4	9666.49	1869.50	0.42	0.26	0.09	10 461.77	10 427.14	3.00	489.33	2.52	0.33	0.92
i80	3	10 461.77	2944.00	0.22	0.20	0.09	9724.90	9678.12	4.00	1007.67	3.30	0.50	1.08
i85	∞	12 881.61	14 417.00	100.00	0.58	0.45	8237.29	7786.92	4.67	6282.00	3.89	5.28	1.56
i85	10	27 464.93	14 415.50	100.00	0.35	0.53	9412.95	9121.81	4.00	3855.33	3.50	3.14	2.92
i85	5	12 709.96	11 117.00	67.40	0.26	0.11	8406.44	8225.15	4.67	4705.67	3.77	2.29	1.51
i85	4	10 491.35	2792.50	0.21	0.25	0.12	10 474.71	10 430.64	3.33	450.67	2.79	0.43	1.00
i85	3	11 231.77	4377.50	0.06	0.19	0.10	11 277.93	11 245.27	3.33	141.33	2.70	0.29	1.00
i90	∞	20 939.77	14 421.00	100.00	0.35	0.61	8428.48	8146.19	5.00	8046.67	3.81	3.27	2.48
i90	10	14 288.74	14 418.50	100.00	0.51	0.23	9869.53	9681.89	4.33	5698.67	3.30	1.94	1.45
i90	5	10 036.99	10 943.00	50.12	0.29	0.07	8675.32	8586.73	5.33	6523.33	3.88	0.98	1.16
i90	4	10 800.42	1733.50	0.25	0.23	0.09	12 025.43	12 001.13	3.67	2121.67	2.64	0.20	0.90
i90	3	11 737.02	7168.00	0.24	0.17	0.06	11 163.03	11 114.36	3.67	2615.00	2.69	0.44	1.05
i95	∞	16 422.09	14 423.50	100.00	0.36	0.14	9084.20	8672.74	5.00	13 001.67	3.51	4.44	1.81
i95	10	18 091.24	14 423.00	100.00	0.48	0.37	10 597.19	10 451.95	4.00	8498.67	2.73	1.42	1.71
i95	5	13 999.86	14 423.50	100.00	0.27	0.20	9142.20	8996.95	5.00	7505.67	3.84	1.61	1.53
i95	4	17 378.09	14 422.50	100.00	0.22	0.29	12 899.95	12 847.30	3.33	1282.00	2.18	0.40	1.35
i95	3	12 698.06	7626.00	50.03	0.16	0.06	12 003.36	11 929.09	3.33	1314.67	2.22	0.62	1.06

(*continued on next page*)

Table 2 shows the average p-median results for the different problems addressed in this paper and for the different data sets. The time limit was set to 4 h although in general the execution time is negligible.

Table 3 (*continued*).

Inst	Cap	CCEFLP						z_{LB}	CCEFLP Algorithm				
		Obj	Time	%GAP	$P(x_{ikj})$	$P(d_{ikj})$	z*		Iter	Time	#%k	$\%GAP_{LB}$	GR
i100	∞						9861.92	9509.05	5.00	13 489.33	3.22	3.52	
i100	10						11 737.19	11 529.13	4.00	13 555.00	2.58	1.83	
i100	5						10 040.89	9865.10	5.00	7036.00	3.29	1.94	
i100	4						14 075.72	13 938.36	3.67	2069.33	2.43	0.97	
i100	3						13 160.10	13 029.60	3.67	3027.00	2.39	1.00	

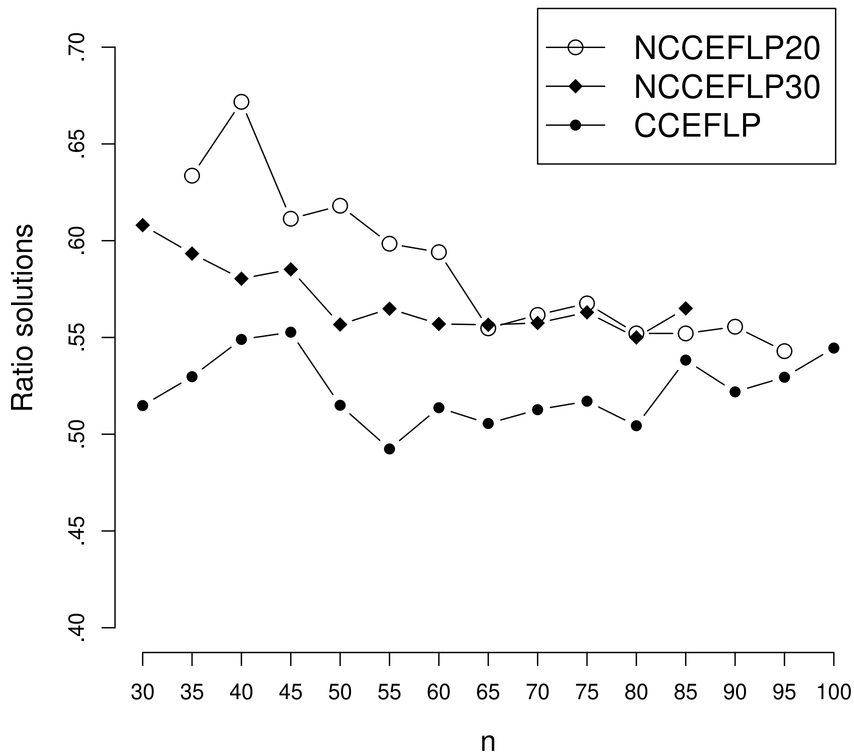


Fig. 6. *p*-median problem versus NCCEFLP and CCEFLP.

The linear relaxation gap (%GAP) is usually small while the largest instances in NCCEFLP 30% could not be solved due to memory overflow. In general, computational time for solving NCCEFLP is slightly higher than for solving CCEFLP. Later we use the values in this table to illustrate the advantages of using the proposed approach compared to employing a more classical *p*-median location problem.

Table 3 shows the average results of the instances presented in Table 1 for the CCEFLP problem, i.e., the location problem on the plane. For each instance different capacities have been considered for each pickup point: 3, 4, 5, 10 and ∞, where a capacity of ∞ means that it is the uncapacitated version. Two different runs have been carried out, plain use of CPLEX v20.1 for the models and the Column Generation Algorithm in Algorithm 2. Columns header with Obj, Time and %GAP give, respectively, the best feasible solution of the CCEFLP problem, the computational time required or the maximum time allowed (14400 s), and the optimality gap between the solution of the CCEFLP and its linear relaxation. Column $P(x_{ikj})$ shows the proportion of customers allocated to pickup points. Note that the proportion of customers allocated to facilities is the difference up to 1. Column $P(d_{ikj})$ shows the proportion of traveled cost due to pickup allocation, the difference up to 1 is the proportion for facility allocation. Columns z^*, z_{LB}, $Iter$, $Time$, #%k, $\%GAP_{LB}$, and GR show the results for the column generation algorithm solution: the solution and the lower bound provided by the algorithm, the number of iterations, the total computational time, the percentage of pickup points considered in the last iteration of the algorithm, the GAP between the solution and its lower bound, and the ratio between the optimal solution Obj and the algorithm solution (Goodness Ratio). The missing values occur when CPLEX is unable to find a feasible solution within the time limit of 14400 s.

Looking at Table 3 it can be concluded that the column generation approach presented in this work is always competitive compared with the plain use of CPLEX. Our column generation algorithm is significantly faster than CPLEX in most instances, especially for the last ones. Moreover, values of GR are always close to 1, and in many cases even greater. Note, for example, that for the instance i85 with capacity=10. the algorithm obtains an objective value three times smaller than the CPLEX objective value, i.e., GR=2.92. It is notorious that the algorithm uses a small number of pickup points, i.e., the algorithm consumes almost all the time to solve the reduced CCEFLP to optimality. In terms of GR values, the algorithm performs better for uncapacitated problems. In all cases, the optimality gap provided by the algorithm is small, with the worst result being for instance i85, which has a gap of 5.28%. However, even in this case, the algorithm finds a solution 1.56 times better than CPLEX. The number of connections between pickup point and facilities decreases with the capacity of the pickup points, as they can serve fewer customers, and the number of direct connections between customers and facilities increases. On the other hand, the relative cost of the two types of connections—direct or through a

Table 4
Average results for the three data sets in the NCCEFLP problem with 20% network density.

Inst	Cap	CCEFLP						z_{LB}	CCEFLP Algorithm				
		Obj	Time	%GAP	$P(x_{ikj})$	$P(d_{ikj})$	z*		Iter	Time	#%k	$\%GAP_{LB}$	GR
i35	∞	4863.00	90.50	0.72	0.52	0.65	4752.20	4697.46	2.33	4.33	11.61	1.15	1.02
i35	10	4863.00	106.00	0.72	0.52	0.65	4752.20	4704.94	3.00	4.00	12.61	0.99	1.02
i35	5	4863.00	111.00	0.72	0.52	0.65	4752.20	4695.76	2.33	4.00	11.87	1.19	1.02
i35	4	4884.00	124.50	0.72	0.48	0.64	4770.87	4711.64	3.33	3.00	13.13	1.24	1.02
i35	3	5025.50	140.00	1.01	0.42	0.62	4908.03	4848.96	3.00	3.00	10.89	1.20	1.02
i40	∞	5775.00	201.00	0.08	0.49	0.59	5775.17	5770.61	2.50	4.00	9.36	0.08	1.00
i40	10	5775.00	199.00	0.08	0.49	0.59	5775.17	5770.12	3.00	5.50	10.11	0.09	1.00
i40	5	5782.50	203.50	0.08	0.47	0.59	5782.46	5775.26	3.00	5.50	10.40	0.12	1.00
i40	4	5827.00	192.00	0.08	0.44	0.58	5827.16	5785.72	2.50	5.50	9.52	0.71	1.00
i40	3	5989.50	186.00	0.00	0.39	0.56	5989.48	5989.47	3.00	5.50	10.13	0.00	1.00
i45	∞	5380.00	550.67	0.90	0.56	0.59	5393.38	5299.71	1.67	8.33	5.59	1.74	0.99
i45	10	5380.00	529.67	0.90	0.56	0.59	5393.38	5319.49	3.33	9.00	5.45	1.37	0.99
i45	5	5493.00	739.33	1.19	0.50	0.58	5506.68	5418.92	2.67	9.67	5.10	1.59	0.99
i45	4	5588.33	683.00	0.71	0.45	0.58	5590.46	5524.18	2.00	8.67	5.00	1.19	1.00
i45	3	5938.67	390.33	0.95	0.36	0.46	5938.84	5865.89	2.67	6.33	4.99	1.23	1.00
i50	∞	6430.00	2224.00	1.13	0.58	0.23	6430.24	6339.02	2.00	263.00	3.37	1.42	1.00
i50	10	6430.00	2576.50	1.13	0.58	0.23	6430.24	6339.01	2.00	447.00	3.41	1.42	1.00
i50	5	6657.50	2803.50	2.16	0.50	0.24	6667.17	6498.90	3.00	133.00	3.51	2.52	0.99
i50	4	6852.00	1440.50	1.22	0.42	0.23	6852.17	6747.80	1.50	48.00	3.47	1.52	1.00
i50	3	7220.00	1042.50	0.27	0.33	0.18	7219.97	7183.05	2.00	154.50	3.26	0.51	1.00
i55	∞	6478.00	2599.00	0.26	0.60	0.32	6492.41	6458.69	2.00	51.00	3.01	0.52	0.99
i55	10	6478.00	2422.00	0.26	0.60	0.32	6492.41	6433.29	1.50	55.00	2.91	0.91	0.99
i55	5	6801.50	2553.50	1.74	0.48	0.30	6801.73	6692.04	3.50	54.00	3.25	1.61	1.00
i55	4	7051.50	1497.50	0.82	0.39	0.15	7051.73	6972.01	2.50	54.50	2.80	1.13	1.00
i55	3	7539.00	1487.50	0.68	0.29	0.10	7538.95	7488.39	1.50	21.00	2.59	0.67	1.00
i60	∞	6929.00	4344.00	2.06	0.61	0.27	6928.72	6778.81	2.00	469.00	2.19	2.16	1.00
i60	10	6942.00	4657.00	2.16	0.60	0.26	6941.80	6783.71	3.50	666.00	2.05	2.28	1.00
i60	5	7413.00	5759.00	2.51	0.43	0.08	7425.08	7230.38	2.00	638.00	1.89	2.62	0.99
i60	4	7769.50	5562.50	1.77	0.36	0.19	7769.31	7619.46	1.50	920.00	1.84	1.93	1.00
i60	3	8320.50	3569.00	1.07	0.27	0.16	8331.27	8220.53	1.50	879.50	1.90	1.33	0.99
i65	∞	6818.00	8444.33	1.13	0.62	0.37	6818.12	6728.67	2.67	622.33	1.23	1.31	1.00
i65	10	6850.00	7909.67	1.30	0.60	0.37	6849.84	6766.29	2.33	335.00	0.94	1.22	1.00
i65	5	7713.00	6855.67	2.30	0.40	0.30	7724.98	7583.00	2.33	513.33	0.97	1.84	0.99
i65	4	8183.67	5592.00	1.75	0.33	0.25	8183.66	8072.48	1.33	1429.33	1.01	1.36	1.00
i65	3	8784.67	5812.00	1.10	0.25	0.31	8784.76	8694.15	1.00	889.00	0.89	1.03	1.00
i70	∞	7363.00	11 383.33	2.98	0.61	0.42	7302.06	7152.52	2.33	1344.67	0.77	2.05	1.01
i70	10	7562.67	12 664.67	8.48	0.57	0.44	7375.78	7239.94	3.00	1438.00	0.83	1.84	1.02
i70	5	8352.33	9407.33	2.10	0.38	0.26	8370.92	8226.93	2.00	1759.33	0.80	1.72	0.99
i70	4	8793.33	8868.00	1.15	0.30	0.28	8793.24	8694.16	2.67	1078.67	0.92	1.13	1.00
i70	3	9359.00	6293.67	0.50	0.23	0.19	9358.77	9329.80	1.00	690.33	0.83	0.31	1.00
i75	∞	14 591.00	13 645.50	2.12	0.34	0.23	7971.26	7862.56	2.00	1818.33	0.74	1.36	1.83
i75	10	18 578.33	9027.00	2.62	0.40	0.36	8062.93	7937.64	1.33	2717.67	0.67	1.55	2.30
i75	5	20 406.00	7071.67	0.61	0.28	0.31	9146.93	9072.31	3.33	1261.67	0.75	0.82	2.23
i75	4	23 932.00	7498.67	19.59	0.16	0.23	9673.08	9607.53	3.33	879.67	0.87	0.68	2.47
i75	3	21 997.33	6269.33	26.10	0.17	0.25	10 315.21	10 289.29	2.00	642.00	0.76	0.25	2.13
i80	∞	21 602.00	14 427.00		0.08	0.02	8145.71	8087.19	2.00	1988.00	0.49	0.72	2.65
i80	10	32 173.50	7508.50		0.13	0.13	8328.30	8231.79	2.00	3328.33	0.46	1.16	3.86
i80	5	33 583.67	5192.67		0.10	0.22	9772.60	9649.60	2.33	2486.33	0.58	1.26	3.43
i80	4	29 683.00	7491.00		0.10	0.12	10 332.97	10 234.80	2.67	3238.67	0.63	0.95	2.87
i80	3	28 730.00	7486.00	43.71	0.09	0.12	10 988.80	10 950.28	2.00	5201.67	0.58	0.35	2.61
i85	∞	58 210.00	510.00		0.02	0.03	8632.50	8517.83	1.00	5204.33	0.42	1.33	6.74
i85	10	51 639.00	633.00		0.09	0.12	8997.57	8755.12	2.67	13 290.33	0.37	2.69	5.73
i85	5	51 517.50	613.50		0.09	0.11	10 844.11	10 703.50	3.00	7520.50	0.43	1.30	4.75
i85	4	52 038.00	372.00		0.06	0.08	11 427.72	11 373.38	3.00	7348.50	0.49	0.48	4.55
i85	3	58 168.00	386.00	78.20	0.01	0.01	12 172.34	12 139.89	2.00	5110.50	0.44	0.27	4.77
i90	∞						9094.33	8999.84	2.00	5505.50	0.24	1.04	
i90	10						9558.88	9239.94	3.00	13 454.00	0.27	3.34	
i90	5						11 498.62	11 225.83	2.50	15 695.00	0.26	2.37	
i90	4						12 075.98	11 976.58	2.00	14 115.50	0.30	0.82	
i90	3						12 890.85	12 778.96	2.00	17 018.00	0.32	0.87	
i95	∞						9381.82	9327.83	4.00	23 243.00	0.27	0.58	
i95	10						9121.39	8875.08	2.00	33 858.00	0.24	2.70	
i95	5						11 423.13	11 408.85	2.00	21 518.00	0.24	0.13	
i95	4						12 131.58	12 114.63	2.00	1901.00	0.25	0.14	
i95	3						13 077.78	13 045.51	2.00	16 241.00	0.23	0.25	
i100	∞												
i100	10						9999.57	9698.67	3.00	3500.00	0.13	1.04	
i100	5						12 545.41	12 433.62	3.00	5484.00	0.14	3.34	
i100	4						13 293.75	13 292.38	2.00	9236.00	0.13	2.37	
i100	3												

pickup point—decreases and increases, respectively, when the capacity of the pickup points decreases.

The next computational experience presented is for the location problem on a network with densities 20% and 30%. Tables 4 and 5

Table 5
Average results for the three data sets in the NCCEFLP problem with 30% network density.

Inst	Cap	CCEFLP						z_{LB}	CCEFLP Algorithm				
		Obj	Time	%GAP	$P(x_{ikj})$	$P(d_{ikj})$	z*		Iter	Time	#%k	%GAP_{LB}	GR
i30	∞	3218.67	101.67	2.94	0.42	0.62	3218.55	3105.15	3.33	0.00	1.71	3.52	1.00
i30	10	3218.67	121.33	2.94	0.49	0.59	3218.55	3096.43	3.33	1.00	1.85	3.79	1.00
i30	5	3218.67	113.00	2.94	0.49	0.59	3218.55	3112.37	3.67	0.33	1.91	3.30	1.00
i30	4	3227.67	118.33	2.94	0.47	0.59	3227.44	3115.63	3.33	0.33	1.91	3.46	1.00
i30	3	3243.00	103.67	2.74	0.44	0.58	3243.15	3138.71	3.33	0.33	1.54	3.22	1.00
i35	∞	3765.33	240.33	0.39	0.39	0.56	3765.24	3747.86	4.00	2.67	1.23	0.46	1.00
i35	10	3765.33	242.33	0.39	0.56	0.59	3765.24	3751.08	4.00	2.00	1.23	0.38	1.00
i35	5	3773.33	220.33	0.53	0.56	0.59	3773.24	3747.15	3.33	2.00	1.28	0.69	1.00
i35	4	3818.67	226.00	1.09	0.50	0.58	3818.43	3776.12	4.00	2.67	1.27	1.11	1.00
i35	3	3952.33	388.67	1.40	0.45	0.58	3958.31	3897.80	4.00	1.33	1.17	1.53	1.00
i40	∞	4257.33	296.00	0.01	0.36	0.46	4256.98	4250.86	3.33	2.00	0.79	0.14	1.00
i40	10	4257.33	315.33	0.01	0.58	0.23	4256.98	4252.97	3.67	2.67	0.83	0.09	1.00
i40	5	4291.00	320.00	0.11	0.58	0.23	4290.82	4278.23	3.67	3.33	0.95	0.29	1.00
i40	4	4381.00	295.00	0.52	0.50	0.24	4380.95	4346.28	3.33	3.00	0.72	0.79	1.00
i40	3	4726.33	300.67	1.25	0.42	0.23	4726.41	4654.66	3.33	3.00	0.71	1.52	1.00
i45	∞	4938.00	756.67	0.72	0.33	0.18	4937.78	4896.46	3.33	8.33	0.57	0.84	1.00
i45	10	4938.00	813.33	0.72	0.60	0.32	4937.78	4893.78	3.67	9.33	0.67	0.89	1.00
i45	5	5041.00	736.00	1.00	0.60	0.32	5040.89	4977.75	3.33	11.00	0.52	1.25	1.00
i45	4	5132.67	588.00	0.49	0.48	0.30	5133.03	5107.37	3.33	13.00	0.47	0.50	1.00
i45	3	5580.33	1010.00	0.96	0.39	0.15	5580.37	5516.21	3.33	9.00	0.58	1.15	1.00
i50	∞	5282.00	2529.00	0.75	0.29	0.10	5294.49	5238.46	3.67	77.00	0.35	1.06	1.00
i50	10	5287.00	2087.33	0.79	0.61	0.27	5287.17	5236.36	3.67	132.00	0.42	0.96	1.00
i50	5	5613.67	2763.00	1.96	0.60	0.26	5613.95	5491.22	3.33	178.00	0.39	2.19	1.00
i50	4	5829.00	2211.67	0.84	0.43	0.08	5834.97	5772.56	3.33	121.33	0.45	1.07	1.00
i50	3	6266.67	1931.33	0.27	0.36	0.19	6266.56	6250.63	3.33	56.67	0.29	0.25	1.00
i55	∞	5862.67	4721.67	0.70	0.27	0.16	5862.55	5794.88	3.33	109.33	0.36	1.15	1.00
i55	10	5894.33	4881.00	1.00	0.62	0.37	5894.27	5833.54	3.67	240.33	0.41	1.03	1.00
i55	5	6302.00	4085.67	1.89	0.60	0.37	6302.09	6179.84	3.33	226.67	0.44	1.94	1.00
i55	4	6546.67	3381.67	0.65	0.40	0.30	6548.17	6471.96	3.00	506.00	0.38	1.16	1.00
i55	3	7098.00	3329.67	0.56	0.33	0.25	7199.68	7138.88	3.00	39.50	0.65	0.84	0.99
i60	∞	6621.00	6213.50	0.39	0.25	0.31	6306.43	6265.62	3.33	1476.67	0.37	0.65	1.05
i60	10	6634.00	6566.00	1.04	0.61	0.42	6346.87	6295.26	3.33	1200.67	0.34	0.81	1.05
i60	5	14 251.67	4520.00	26.97	0.57	0.44	6927.06	6821.29	3.33	1323.67	0.33	1.53	2.06
i60	4	16 420.67	4636.33	27.66	0.38	0.26	7347.64	7246.65	3.33	706.33	0.32	1.37	2.23
i60	3	15 796.33	2700.33	25.69	0.30	0.28	7957.75	7854.67	3.00	229.00	0.30	1.30	1.99
i65	∞	7189.00	13 542.50	5.20	0.23	0.19	6694.15	6568.72	3.00	2082.00	0.21	1.87	1.07
i65	10	15 631.33	9444.67	44.76	0.34	0.23	6725.87	6584.25	3.00	3152.67	0.18	2.11	2.32
i65	5	15 945.00	5712.67	42.47	0.40	0.36	7563.02	7409.12	3.33	5068.67	0.21	2.03	2.11
i65	4	16 548.67	6019.67	41.71	0.28	0.31	8004.36	7859.51	3.00	5269.33	0.22	1.81	2.07
i65	3	17 092.00	5013.33	40.24	0.16	0.23	8572.72	8488.27	2.67	3222.00	0.20	0.99	1.99
i70	∞	40 633.00	373.00	85.01	0.17	0.25	7129.09	7049.63	3.00	6997.67	0.17	1.11	5.70
i70	10	40 705.00	379.00	86.54	0.08	0.02	7202.81	7120.03	3.33	6263.67	0.17	1.15	5.65
i70	5	18 644.67	8551.33	9.64	0.13	0.13	8197.63	8072.23	3.00	5929.67	0.17	1.53	2.27
i70	4	8667.50	12 242.50	5.59	0.10	0.22	8149.86	8089.06	2.50	4355.50	0.26	0.75	1.06
i70	3	19 135.67	5129.33	27.63	0.10	0.12	8708.93	8708.93	2.50	3018.00	0.27	0.00	2.20
i75	∞						7816.06	7715.30	3.33	2086.00	0.25	1.29	
i75	10	42 966.00	615.00		0.02	0.03	7907.77	7772.39	3.00	2651.67	0.23	1.71	5.43
i75	5	41 549.00	459.00		0.09	0.12	9023.93	8925.09	3.33	13 764.67	0.32	1.10	4.60
i75	4	44 847.50	450.00		0.09	0.11	9548.15	9486.71	3.00	6981.00	0.24	0.64	4.70
i75	3	43 660.50	449.50	74.61	0.06	0.08	10 651.12	10 570.51	2.00	7672.00	0.26	0.76	4.10
i80	∞						8001.49	7959.84	3.33	8715.00	0.21	0.52	
i80	10						8388.88	8314.51	3.00	45 538.00	0.21	0.89	
i80	5						9931.96	9809.74	3.00	14 112.50	0.19	1.23	
i80	4	54 538.00	846.00		0.17	0.00	10 525.97	10 480.63	3.00	3055.50	0.19	0.43	5.18
i80	3						11 244.34	11 229.54	3.50	11 333.00	0.20	0.13	
i85	∞						8680.63	8618.42	3.50	37 878.50	0.11	0.72	
i85	10						9032.58	8841.51	3.50	56 828.50	0.10	2.12	
i85	5						10 201.62	10 130.30	2.00	30 570.00	0.10	0.70	
i85	4						10 819.72	10 808.87	2.00	28 249.00	0.10	0.10	
i85	3						11 571.30	11 498.47	2.00	2724.00	0.10	0.63	
i90	∞						8129.20	8062.36	3.00	5900.00	0.10	0.82	
i90	10, 5, 4, 3												
i95	∞												
i95	10						10 774.29		1.00	113 228.00	0.07		
i95	5						12 582.51		1.00	156 409.00	0.07		
i95	4						13 161.10		1.00	79 784.00	0.07		
i95	3												
i100	∞, 10, 5, 4, 3												

show the results for both densities, respectively. The time limit has been set to 60.000 s since it has been seen that the problem on a network is more difficult than its equivalent on the plane. Some instances cannot be solved due to a memory overflow. As a first observation about the results, the complexity of the problem increases when the network density rises. When the network density is 20%, CPLEX plain use does not obtain feasible solutions for instances of size 90 or greater while the column generation algorithm can solve all the cases except two

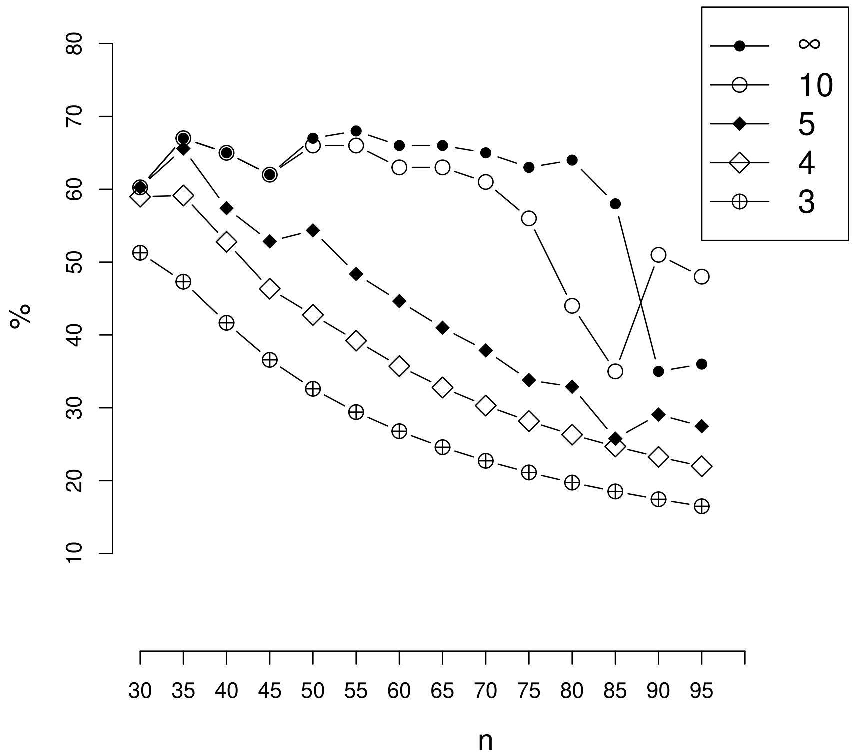


Fig. 7. Percentage of customers allocated to pickup points in CCEFLP and NCCEFLP depending on capacity.

sets of instances with 100 nodes. When the density increases to 30%, CPLEX cannot solve some instances of size 75, while the algorithm fails to provide a feasible solution from a network size of 90. The goodness ratio GR is frequently close to 1 and it is sometimes very large, it is even 6.743 in one case. These numbers illustrate the good performance of our algorithm. The optimality gap $\%GAP_{LB}$ of our algorithm is similar across all solved instances and the number of iterations of our algorithm goes from 2 to 4. The computing time goes from 1 to 60.000 s However, CPLEX cannot solve these more difficult instances, even find a feasible solution. Blank spaces correspond to instances where the computer's memory is exceeded.

Finally, Figs. 6 and 7 gather information that appears in different tables. In particular, Fig. 6 depicts the advantages of using the proposed approach instead of the p-median location problem: it shows the ratio of the optimal value for CCEFLP and NCCEFLP with a capacity equal to ∞ and the solution of the p-median problem. This ratio is lower for the flat problem, i.e. for CCEFLP, and therefore it is in this case that the use of an appropriate model to reduce costs has the greatest impact. In any case, for the rest of the models, the ratio is also satisfactorily small. Fig. 7 depicts the average values in columns $P(x_{ikj})$ in Table 3 multiplied by 100 and distinguishing by capacity. It shows that the percentage decreases with the size of the instance and the line associated with any capacity is above the line associated with another lower capacity.

6. Conclusions

In this paper, we have formulated two new models from an evolving line of research. These models help us to obtain an optimal solution when customers are willing to go to a pickup point. All these models that have been presented here aim to locate both plants and pickup points, always minimizing the company's transportation cost and satisfying customer demands.

The pickup points generated depend on the type of problem we are addressing, either on a plane (CCEFLP) or on a network-based environment (NCCEFLP). In this work we introduce an algorithm to the generation of pickup points within a network. These pickup points are distributed over all possible routes, as they do not need to be limited to the shortest path between two points.

Finally, in order to improve the results obtained with the 3-index integer programming models (CCEFLP and NCCEFLP), a column generation algorithm has been created. Computational results corroborate the good performance of the new algorithm.

CRediT authorship contribution statement

Alejandro Moya-Martínez: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation. **Mercedes Landete:** Writing – review & editing, Supervision, Methodology, Investigation. **Juan F. Monge:** Writing – review & editing, Supervision, Software, Investigation. **Sergio García:** Writing – review & editing, Investigation.

Acknowledgments

This research has been supported through the grants, PID2019-105952GB-I00 and PID2021-122344NB-I00. funded by Ministerio de Ciencia e Innovación, Spain/Agencia Estatal de Investigación, Spain/ 10.130 39/501100011033, and PROMETEO/2021/063, funded by Conselleria d'Innovació, Universitat, Ciencia i Societat Digital, Spain.

Data availability

Data will be made available on request.

References

Bianchessi, N., Corberán, A., Plana, I., Reula, M., Sanchis, J.M., 2022a. The min–max close-enough arc routing problem. European J. Oper. Res. 300 (3), 837–851. http://dx.doi.org/10.1016/j.ejor.2021.10.047.

Bianchessi, N., Corberán, A., Plana, I., Reula, M., Sanchis, J.M., 2022b. The profitable close-enough arc routing problem. Comput. Oper. Res. 140, 105653. http://dx.doi.org/10.1016/j.cor.2021.105653.

Church, R., Meadows, M., 1979. Location modeling utilizing maximum service distance criteria. Geogr. Anal. 11, 358–373. http://dx.doi.org/10.1111/j.1538-4632.1979.tb00702.x, https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1538-4632.1979.tb00702.x.

Corberán, A., Landete, M., Peiró, J., Saldanha-da-Gama, F., 2020. The facility location problem with capacity transfers. Transp. Res. Part E: Logist. Transp. Rev. 138, 1366–5545. http://dx.doi.org/10.1016/j.tre.2020.101943.

Corberán, Á., Plana, I., Reula, M., Sanchis, J., 2019. A matheuristic for the distance-constrained close enough arc routing problem. TOP 27 (2), 312–326. http://dx.doi.org/10.1007/s11750-019-00507-3.

Corberán, Á., Plana, I., Reula, Mi., Sanchis, J.M., 2021. On the distance-constrained close enough arc routing problem. European J. Oper. Res. 291 (1), 32–51. http://dx.doi.org/10.1016/j.ejor.2020.09.012.

Di Placido, A., Archetti, C., Cerrone, C., Golden, B., 2023. The generalized close enough traveling salesman problem. European J. Oper. Res. 310 (3), 974–991. http://dx.doi.org/10.1016/j.ejor.2023.04.010.

Gulczynski, D.J., Heath, J.W., Price, C.C., 2006. The Close Enough Traveling Salesman Problem: A Discussion of Several Heuristics. Springer US, Boston, MA, ISBN: 978-0-387-39934-8, pp. 271–283. http://dx.doi.org/10.1007/978-0-387-39934-8_16.

Hakimi, S.L., 1964. Optimum locations of switching centers and the absolute centers and medians of a graph. Oper. Res. 12 (3), 450–459, (ISSN: 0030364X, 15265463), http://www.jstor.org/stable/168125.

Hartmanis, J., 1982. Computers and intractability: A guide to the theory of NP-completeness (Michael R. Garey and David S. Johnson). SIAM Rev. 24 (1), http://dx.doi.org/10.1137/1024022, 90–2, https://www.proquest.com/scholarly-journals/computers-intractability-guide-theory-np/docview/926173619/se-2?accountid=14777.

Hernández-Pérez, H., Landete, M., Rodríguez-Martín, I., 2021. The single-vehicle two-echelon one-commodity pickup and delivery problem. Comput. Oper. Res. (ISSN: 0305-0548) 127, 105152. http://dx.doi.org/10.1016/j.cor.2020.105152.

Landete, M., Laporte, G., 2019. Facility location problems with user cooperation. TOP 27, 125–145. http://dx.doi.org/10.1007/s11750-018-00496-9.

Moya-Martínez, A., Landete, M., Monge, J.F., 2021. Close-enough facility location. Mathematics (ISSN: 2227-7390) 9 (6), http://dx.doi.org/10.3390/math9060670, https://www.mdpi.com/2227-7390/9/6/670.

Osman, I., Christofides, N., 1994. Capacitated clustering problems by hybrid simulated annealing and tabu search. Int. Trans. Oper. Res. (ISSN: 0969-6016) 317–336. http://dx.doi.org/10.1016/0969-6016(94)90032-9, http://people.brunel.ac.uk/mastjjb/jeb/orlib/pmedcapinfo.html.

Reula, M., Martí, R., 2023. Heuristics for the profitable close-enough arc routing problem. Expert Syst. Appl. (ISSN: 0957-4174) 230, http://dx.doi.org/10.1016/j.eswa.2023.120513.