European Journal of Operational Research 266 (2018) 46–56

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Outer approximation and submodular cuts for maximum capture facility location problems with random utilities

Ivana Ljubić[a,*], Eduardo Moreno[b]

[a] ESSEC Business School, 3 Av. Bernard Hirsch, B.P. 50105, Cergy Pontoise Cedex 95021, France
[b] Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Avda. Diagonal Las Torres 2640, Peñalolen 7941169, Santiago, Chile

ARTICLE INFO

Article history:
Received 28 March 2017
Accepted 13 September 2017
Available online 21 September 2017

Keywords:
Combinatorial optimization
Branch-and-cut
Maximum capture
Random utility model
Competitive facility location

ABSTRACT

We consider a family of competitive facility location problems in which a "newcomer" company enters the market and has to decide where to locate a set of new facilities so as to maximize its market share. The multinomial logit model is used to estimate the captured customer demand. We propose a first branch-and-cut approach for this family of difficult mixed-integer non-linear problems. Our approach combines two types of cutting planes that exploit particular properties of the objective function: the first one are the outer-approximation cuts and the second one are the submodular cuts.

The approach is computationally evaluated on three datasets from the recent literature. The obtained results show that our new branch-and-cut drastically outperforms state-of-the-art exact approaches, both in terms of the computing times, and in terms of the number of instances solved to optimality.

1. Introduction

We propose a methodological and algorithmic framework for a family of facility location problems in which customer behavior is integrated into the optimization model. Facility location problems play a fundamental role in modeling important managerial decisions concerning infrastructure planning, such as placement of new retail or service facilities, placement of new products on the market, or development of optimal customer segmentation policies. Integration of random choice models into optimization models allows companies to make optimal decisions while taking the preferences of their customers into account. One of the frequently used choice models in practice is the multinomial logit model (MNL) which is studied in this paper.

In this article we focus on *Maximum Capture Facility Location Problems with Random Utilities* (MCFLRU). In these problems, we are given a company that is entering the market in which a set of incumbent competitors already operates. The company has to decide where to locate a set of new facilities, so as to maximize the captured demand. Facilities in our setting may correspond to bank offices, warehouses, shopping malls, park-and-ride car parks, and many more. Both, the decision maker and the competitor(s) offer the same product, so that the major decision concerns the location of the new facilities, after which the customers choose the facilities to be served from. Customers act as independent decision makers and it is assumed that their choices are modeled according to the multinomial logit model. One of the first problems of this type studied in the competitive facility location literature, which was also the motivating application for this article, was proposed by Benati and Hansen (2002). In this problem, called the *Maximum Capture Problem with Random Utilities* (MCRU), the goal is to locate exactly r new facilities so as to maximize the market share. The MCRU generalizes well-known and well-studied Maximum Capture Facility Location Problem on a network (see, e.g., ReVelle, 1986) in which customers deterministically choose the closest facility. Since the early work of Benati and Hansen (2002), the MCRU and its variants became an important topic of research, both from the methodological and application perspective. In the existing literature, many exact approaches can be found (cf. Section 2.2) along with case studies on real-world instances (Aros-Vera, Marianov, & Mitchell, 2013; Freire, Moreno, & Yushimito, 2016b; Haase & Müller, 2012, 2015; Müller, Haase, & Kless, 2009). These studies successfully demonstrate that random-choice models can be computationally efficient as far as small and medium size instances are concerned. However, large-scale instances of practical relevance, like those from the case study on placing new park-and-ride facilities in the New York City proposed in Freire, Moreno, and Yushimito (2016a), Freire et al. (2016b) with more than 80 K customer locations, remain out of reach of the existing exact approaches.

* Corresponding author.
E-mail addresses: ivana.ljubic@essec.edu, ivana.ljubic@univie.ac.at, ljubic@essec.edu (I. Ljubić), eduardo.moreno@uai.cl (E. Moreno).

https://doi.org/10.1016/j.ejor.2017.09.023
0377-2217/

Our contribution. In an attempt to linearize the objective function, various mixed integer linear programming (MILP) models were studied in the literature, see Haase and Müller (2014) for an overview. Unfortunately, the proposed linearized counterparts come at the cost of a drastic increase of the number of decision variables, which makes these models prohibitive for large-scale instances. In this article, we consider two sparse MILP models with an exponential number of constraints. The first model relies on the outer-approximation of the continuous relaxation of the objective function, and the second one exploits the submodularity of the objective function. We also investigate a third viable option of combining the two families of cuts in a single MILP model. The latter turns out to be the most promising option from the computational perspective.

We implement and computationally evaluate these branch-and-cut (B&C) approaches against the state-of-the-art exact approaches. Results are compared using three large datasets for the MCRU, recently evaluated in Freire et al. (2016a). Our results show that the proposed methodology outperforms all previously studied approaches from the literature by a large margin. Speed-ups of up to two to three orders of magnitude are reported for small and medium size instances. Furthermore, for all previously unsolved instances from the literature, optimal solutions are found.

The paper is organized as follows: in Section 2, we provide a formal problem definition and a basic mixed-integer non-linear (MINLP) formulation along with an overview of the recent literature. In Section 3, we describe a mixed integer linear programming (MILP) formulation that is based on outer-approximation cuts and provide details of our branch-and-cut implementation. In Section 4, we propose an alternative MILP model that exploits the submodularity of the objective function. Extensive computational results are presented in Section 5, and final conclusions are drawn in Section 6.

2. Problem description

In classical (deterministic) facility location problems, see, e.g., Fischetti, Ljubić, and Sinnl (2017), decision makers search for optimal locations to open new facilities while assuming that the customers always patronize the closest among the open facilities. In many applications, however, customers prefer to be served by facilities according to their own personal preferences, which are not always known to decision makers. Consequently, for decision makers it may be very difficult (if not impossible) to control customer decisions. This is why random utility models are frequently used to forecast the customer behavior and to predict the market share that can be achieved by attracting them. In the underlying optimization models, utility maximization theory (keeping the hypothesis that customers behave rationally) is combined with a random choice model (allowing to model uncertainty in customer behavior).

Multinomial logit model. In the following, we first explain the major idea of the *multinomial logit* (MNL) model that is used to forecast the captured demand for a given company, given its set of available facilities. Let us assume that we are given a set of customers $S = \{1, \dots, |S|\}$ with demands $d_s > 0$. Without loss of generality, each $s \in S$ can also be seen as a group of individuals with a homogeneous behavior. Let the set of available facilities be denoted by $\bar{L}$. Each customer $s \in S$ choses the facilities from $L_s \subseteq \bar{L}$, which are the facilities offered/available to s. One may assume that $L_s \neq \emptyset$ for all $s \in S$, since, otherwise, customers leaving the market will be assigned to an artificial "opt-out" facility that captures their demand.

The customer s splits the demand d_s based on the *utilities* u_{sl} perceived by s for selecting each facility $l \in L_s$. Unobservable variables modeling customer behavior are treated as random variables so that the utility u_{sl} consists of two parts: a measurable utility value v_{sl} (e.g., distance, costs, availability of parking space) and its non-observable part ϵ_{sl}: $u_{sl} = v_{sl} + \epsilon_{sl}$. In the multinomial logit model, it is assumed that the values of ϵ_{sl} are identically independently distributed with the log-Weibull (also known as Gumbel) distribution, which allows to express the probabilities of customer s to select facility l as:

$$p_{sl} = \begin{cases} \frac{e^{v_{sl}}}{\sum_{l' \in L_s} e^{v_{sl'}}}, & l \in L_s \\ 0, & l \notin L_s \end{cases} \quad s \in S, l \in \bar{L}.$$

The value p_{sl} practically corresponds to the expected fraction of the customer's demand d_s to be served by facility l.

2.1. Maximum Capture Facility Location Problems with Random Utilities (MCFLRU)

In competitive facility location problems we consider an environment in which the customers are already served by existing competitors. A "newcomer" company wants to enter the market and searches for the subset of facility locations to open, so as to maximize the forecasted market share achieved by attracting the new customers. Without loss of generality one can assume that there is a single incumbent competitor, and that all competing facilities are aggregated into a *super-facility* a (a can also include the "opt-out" facility). Let $L = \bar{L} \setminus \{a\}$ denote the set of potential facility locations where new facilities can be opened. For a given set of newly open facilities $L^* \subseteq L$, the customer demand is split based on the utilities u_{sl} perceived by customer s for selecting each facility $l \in L^*$, and the utility u_{sa} perceived for choosing the incumbent competitor. So, according to the MNL model (see Freire et al., 2016a), the probability that customer s will select facility $l \in L$ is now given as:

$$p_{sl} = \begin{cases} \frac{a_{sl}}{1 + \sum_{l' \in L^*} a_{sl'}}, & l \in L^* \\ 0, & l \notin L^* \end{cases} \quad s \in S, l \in L,$$

where $a_{sl} = \exp(v_{sl} - v_{sa})$ and v_{sl} are measurable utility values described above.

An MINLP formulation. Let x_l be a binary variable which is set to one if and only if the company decides to locate a facility at $l \in L$. The fraction of demand d_s for $s \in S$, assigned to facility $l \in L$ can then be calculated as:

$$\hat{p}_{sl}(x) = \frac{a_{sl} x_l}{1 + \sum_{l' \in L} a_{sl'} x_{l'}}. \tag{1}$$

Consequently, the fraction of demand the company can capture from the customer s can be described as a function of x:

$$\hat{w}_s(x) = \sum_{l \in L} \hat{p}_{sl}(x) = \frac{\sum_{l \in L} a_{sl} x_l}{1 + \sum_{l \in L} a_{sl} x_l}, \tag{2}$$

and the total market share is given as:

$$\sum_{s \in S} d_s \hat{w}_s(x) = \sum_{s \in S} d_s \frac{\sum_{l \in L} a_{sl} x_l}{1 + \sum_{l \in L} a_{sl} x_l}. \tag{3}$$

For the continuous relaxation of variables x, function $\hat{w}_s(x)$ is continuously differentiable and concave. In fact, $\hat{w}_s(x)$ is the composition of the unidimensional concave increasing function $g(z) = \frac{z}{1+z}$ (for $z > -1$) with the linear function $\Sigma_{l \in L} a_{sl} x_l$.

The family of MCFLRU problems can now be modeled using the following simple MINLP:

$$\max_{x \in X} \quad \sum_{s \in S} d_s \frac{\sum_{l \in L} a_{sl} x_l}{1 + \sum_{l \in L} a_{sl} x_l}. \tag{4}$$

The objective function in (4) maximizes the market share, whereas the set $X \subseteq \{0,1\}^{|L|}$ describes all feasible *facility configurations*. In case of the MCRU introduced in Benati and Hansen (2002), the company is locating a *fixed* number of r facilities, so as to maximize the overall captured customer demand. Consequently, the set X is given as:

$$X = \{x \in \{0,1\}^{|L|} : \sum_{l \in L} x_l = r\}.$$

The methodology proposed in this paper can be applied to many other competitive facility location problems, in which additional constraints on the feasible facility configurations are imposed. These constraints may be related to the investment budget and/or the resulting infrastructure. So, for example, one can simultaneously optimize location and design decisions for the set of newly opened facilities, considering various design characteristics of each facility (e.g., size, appearance, accessibility, layout, etc). Imagine that for each facility $l \in L$, design decisions are encoded from a set of options $t \in T$ (for simplicity, assume there is a single design characteristic to be optimized), and that a fixed opening cost $\tilde{f}_l \geq 0$ is associated to each $l \in L$. Additional cost $\tilde{c}_{lt} \geq 0$ are to be paid for the design characteristic $t \in T$ of a facility l. Given the total available budget $\tilde{B} > 0$, the set X of all feasible facility configurations is encoded by the following constraints:

$$\{x \in \{0,1\}^{|L||T|} : \sum_{l \in L} \sum_{t \in T} (\tilde{f}_l + \tilde{c}_{lt}) x_{lt} \leq \tilde{B}$$
$$\sum_{t \in T} x_{lt} \leq 1 \quad l \in L\}.$$

Customer utilities are then defined for each facility $l \in L$ and each design decision $t \in T$ as u_{slt}, and the objective function turns into $\sum_{s \in S} d_s \frac{\sum_{l \in L} \sum_{t \in T} a_{slt} x_{lt}}{1 + \sum_{l \in L} \sum_{t \in T} a_{slt} x_{lt}}$.

Furthermore, the set X could encode even more complicated network-design decisions. The relevant deterministic counterparts are the connected facility location (Gollowitzer & Ljubić, 2011), in which the set of open facilities has to be connected through a tree, or the traveling purchaser problem in which open facilities are connected in a tour (Laporte, Ledesma, & González, 2003). So, in a general setting one could have

$$X = \{x \in \{0,1\}^{|L|} : Ax + By \leq b, y \in Y\},$$

where variables y are used to model additional constraints imposed on the set of open facilities (e.g., connectivity). The set Y is assumed to be a polyhedral set, which, together with linking constraints $Ax + By \leq b$ guarantees feasibility of the solution x.

2.2. Previous work

Among the problems from the MCFLRU literature, the most prominent and the most studied one is the MCRU problem, introduced in Benati and Hansen (2002). In their article, the authors propose the first exact approach based on a branch-and-bound (B&B) procedure in which the concave NLP relaxation is solved at every node of the B&B tree. In addition, the authors use fractional programming techniques to linearize the model by introducing an additional set of decision variables, and they discuss submodularity of the objective function. Since then, many approaches are proposed in the literature to solve this difficult problem. Most of them focus on developing MILP models that linearize the objective function (Aros-Vera et al., 2013; Haase, 2009; Zhang, Berman, & Verter, 2012). Haase and Müller (2014) benchmark these different MILP reformulations over a set of randomly generated instances. In Freire et al. (2016a), the authors extend this comparison by including the concave relaxation proposed by Benati and Hansen (2002) and a new relaxation of the problem that can be solved using a greedy algorithm, both embedded in a branch-and-bound algorithm. In this very extensive computational study, two additional datasets are considered: one set is derived from ORLIB, and the other corresponds to the real-world instances from a park-and-ride application of the city of New York (see Aros-Vera et al., 2013; Freire et al., 2016b). The obtained results are inconclusive, showing that different approaches perform dissimilarly depending on the dataset utilized. Furthermore, none of the existing approaches was capable of solving the largest instances from the ORLIB and New York dataset to provable optimality.

3. A B&C approach based on outer-approximation

The main idea behind our first approach is to exploit the fact that for the continuous relaxation of the problem, the (maximization) objective function given in (4) is concave and differentiable. Hence, one can replace the non-linear function by its first-order approximation at any given point. This linear approximation is applied within a cutting plane procedure and repeated at every node of the branch-and-bound tree. The proposed approach is a branch-and-cut algorithm that relies on the outer-approximation decomposition method. The Outer Approximation (OA) decomposition approach was introduced by Duran and Grossmann (1986) and it was later improved by Fletcher and Leyffer (1994). A branch-and-cut algorithm in which outer-approximation cuts are separated at every node of the branch-and-bound tree was proposed by Quesada and Grossmann (1992). In general, the outer-approximation algorithm does not necessarily produce a good performance for generic non-linear problems (Bonami, Biegler, Conn, Cornuéjols, Grossmann et al., 2008), but it can provide good results for some families of convex MINLP problems (Mittelmann, 2014; Vielma, Dunning, Huchette, & Lubin, 2017). Outer approximation resembles the generalized Benders decomposition approach originally proposed by Geoffrion (1972). The latter algorithm, which was successfully applied to other (convex) facility location problems in a deterministic setting (see Fischetti, Ljubić, & Sinnl, 2016; Fischetti et al., 2017) was our main motivation to analyze the efficacy of an outer-approximation-based branch-and-cut algorithm applied to this difficult MINLP.

To derive an appropriate outer-approximation-based MILP formulation, we first consider the following equivalent (extended) MINLP formulation for the problem

$$\max \sum_{s \in S} d_s w_s \tag{5a}$$

$$w_s \leq \hat{w}_s(x) \quad s \in S \tag{5b}$$

$$x \in X, \tag{5c}$$

where new continuous variables w_s represent the fraction of the total demand of customers captured by the facilities given by x and where the function $\hat{w}_s(x)$ is defined according to (2). Due to the maximization nature of the problem, at optimum we will have $w_s = \hat{w}_s(x)$, for all $s \in S$.

Given a vector $x^* \in [0,1]^L$, since $\hat{w}_s(x)$ is a concave function, we can bound the value of $\hat{w}_s(x)$ from above by its first-order approximation on x^*, obtaining the valid constraint

$$\hat{w}_s(x) \leq \hat{w}_s(x^*) + \sum_{l \in L} \frac{\partial \hat{w}_s}{\partial x_l}(x^*) \cdot (x_l - x_l^*). \tag{6}$$

Note that

$$\frac{\partial \hat{w}_s}{\partial x_l}(x^*) = \frac{a_{sl}}{(1 + \sum_{l \in L} x_l^* a_{sl})^2},$$

so

$$\hat{w}_s(x^*) + \sum_{l\in L} \frac{\partial \hat{w}_s}{\partial x_l}(x^*)\cdot(x_l - x_l^*)$$
$$= \hat{w}_s(x^*) + \sum_{l\in L} x_l \cdot \frac{a_{sl}}{(1+\sum_{l\in L} x_l^* a_{sl})^2} - \frac{\sum_{l\in L} a_{sl}x_l^*}{(1+\sum_{l\in L} x_l^* a_{sl})^2}$$
$$= \hat{w}_s(x^*) + \sum_{l\in L} x_l \cdot \frac{a_{sl}}{(1+\sum_{l\in L} x_l^* a_{sl})^2} - \hat{w}_s(x^*)\cdot\frac{1}{1+\sum_{l\in L} x_l^* a_{sl}}.$$

Regrouping terms, inequality (6) can be rewritten as

$$\hat{w}_s(x) \le \hat{w}_s(x^*)^2 + \sum_{l\in L} x_l \cdot \frac{a_{sl}}{(1+\sum_{l\in L} x_l^* a_{sl})^2}. \tag{7}$$

Hence, we have:

Proposition 1. *The MCFLRU can be modeled using the following (sparse) MILP formulation with $|S|+|L|$ variables only, and with an exponential number of constraints:*

$$\max \sum_{s\in S} d_s w_s \tag{8a}$$

$$w_s \le \hat{w}_s(x^*)^2 + \sum_{l\in L} x_l \cdot \frac{a_{sl}}{(1+\sum_{l\in L} x_l^* a_{sl})^2} \qquad s\in S, x^*\in X \tag{8b}$$

$$x \in X. \tag{8c}$$

Validity of the latter model follows from the fact that it is sufficient to outer-approximate functions $\hat{w}_s(x)$ only in a finite number of discrete points $x^* \in X$ (in which we observe that the approximation is tight).

In the following, we will refer to constraints (8b) as *outer-approximation cuts* or *OA cuts*. Even though these cuts do not always lead to particularly strong LP-relaxation bounds, in combination with a branch-and-bound machinery of modern MILP solvers, we will demonstrate that this model can lead to a quite effective branch-and-cut procedure.

Branch-and-cut implementation. In order to solve model (8), we rely on usual branching rules and general-purpose cutting planes embedded in modern MILP solvers. Only when the solution x^* of the current LP-relaxation turns out to be integer, we check if constraints (8b) are violated, in which case we add them to the current LP. OA cuts are globally valid and they are implemented using the `lazy-cut callback` procedure within a MILP solver. We point out that convergence of the branch-and-cut algorithm is guaranteed even if the separation is applied to integer points only. This follows from the fact that there are at most $2^{|L|}$ cuts of type (8b) (one for each possible subset of located facilities), hence, in the worst case, all these subsets would be enumerated throughout the branch-and-bound framework and all relevant cuts would be included. Clearly, tight lower and upper bounds normally help in pruning the tree much earlier and avoiding a complete enumeration.

For a given integer or continuous LP-solution x^*, separation of constraints (8b) can be performed in $O(|S||L|)$ time, since, for each $s\in S$, calculation of $\hat{w}_s(x^*)$ and calculation of coefficients next to x_l variables require $O(|L|)$ time.

The quality of the LP-relaxation can be strengthened by inserting violated cuts (8b) associated to (a finite number) of fractional points $x^*\in\bar{X}$, where $\bar{X} = \{x\in[0,1]^{|L|} : Ax + By \le b, y\in Y\}$. As explained above, the latter cuts (implemented as `user-cut callback`) are not needed for the convergence and correctness of the model, and therefore, they can be controlled by the user and can be applied only if they prove to be useful for improving the LP-relaxation bound (for example, at the root node of the branch-and-bound tree).

4. A B&C approach based on submodular cuts

In previous section, we proposed to tackle the non-linearity by solving an outer approximation of the objective function, and by using branch-and-cut to force integrality constraints. A possible drawback of this approach is that the LP-relaxation at the root node of the branch-and-cut tree can result in a relatively weak upper bound. By exploiting submodularity properties of the objective function, one could instead obtain upper bounds that could be tighter than the ones captured by black-box outer-approximation procedure.

Therefore, in this section, we consider an alternative B&C procedure that exploits submodularity and separability of the objective function. In Benati and Hansen (2002), submodular cuts for the MCRU were proposed and computationally investigated. Unfortunately, only a heuristic procedure for the separation of these cuts was implemented and separability of the objective function was not exploited. The obtained results were not particularly promising, which is why the submodular cuts remained forgotten in the later MCRU literature. Our article is the first attempt to provide a more efficient implementation of submodular cuts in the branch-and-cut frameworks of modern MILP solvers.

In the following, we first recall the basic MILP reformulation for maximizing submodular functions, before we present details of our implementation.

4.1. Maximization of submodular functions

Given a function f: $2^L \mapsto \mathrm{R}$, the difference $f(K+l) - f(K)$ is called *marginal contribution* of element l with respect to the set K. For the sake of better readability, we use the notation $K+l$ and $K-l$ to denote the sets $K\cup\{l\}$ and $K\setminus\{l\}$, respectively. The function f is said to be *non-decreasing* if and only if

$$\rho_l(K) := f(K+l) - f(K) \ge 0, \quad K\subset L, l\notin K$$

holds, in which case marginal contributions $\rho_l(K)$ are also referred to as *marginal gains*. We say that f is *submodular* if and only if

$$f(K+l) - f(K) \ge f(\hat{K}+l) - f(\hat{K}), \quad K\subset\hat{K}\subset L, l\notin\hat{K}$$

holds, i.e., marginal gains of adding an element l diminish with the size of the set.

For a given set $X\subseteq\{0,1\}^{|L|}$, let $K_X = \{K\subseteq L : \exists x\in X \text{ s.t. } x_l = 1 \text{ iff } l\in K\}$ be the superset of all sets indexed by a vector $x\in X$. The following result allows us to formulate a MILP problem for maximizing a submodular function.

Lemma 2 (Nemhauser & Wolsey, 1981)**.** *Given a submodular function f: $2^L\mapsto\mathrm{R}$, the maximization problem of the form*

$$\max\{f(K) : K\in K_X\}$$

can be equivalently reformulated as:

$$\max \quad \nu \tag{9a}$$

$$\nu \le f(K) + \sum_{l\in L\setminus K} \rho_l(K)x_l - \sum_{l\in K}\rho_l(L-l)(1-x_l) \qquad K\subseteq L \tag{9b}$$

$$x\in X. \tag{9c}$$

Constraints (9b) are referred to as submodular cuts.

We show that it is sufficient to impose the submodular cuts (9b) only to the set of points $x\in X$:

Proposition 3. *Given a submodular function f: $2^L\mapsto\mathrm{R}$, the maximization problem of the form $\max\{f(K): K\in K_X\}$ can be equivalently reformulated as:*

$$\max \quad \nu \tag{10a}$$

$$\nu \le f(K) + \sum_{l\in L\setminus K} \rho_l(K)x_l - \sum_{l\in K} \rho_l(L-l)(1-x_l) \qquad K \in K_X \tag{10b}$$

$$x \in X \tag{10c}$$

Proof. To show this result, we prove that for any point $x^* \in X$, the tightest submodular cut (9b) is obtained for the associated set $K^* = \{l \in L : x^*_l = 1\}$. Observe first, that the cut (9b) imposed at the set K^* boils down to

$$\nu \le f(K^*).$$

Consider now the submodular cut (9b) associated to an arbitrary set $K \subseteq L$, possibly $K \notin K_X$ and evaluated at the point x^*. We have:

$$\begin{aligned}
& f(K) + \sum_{l\in L\setminus K} \rho_l(K)x^*_l - \sum_{l\in K} \rho_l(L-l)(1-x^*_l) \\
&= f(K) + \sum_{l\in K^*\setminus K} \rho_l(K) - \sum_{l\in K\setminus K^*} \rho_l(L-l) \\
&= f(K+l') + \sum_{l\in K^*\setminus (K+l')} \rho_l(K) - \sum_{l\in K\setminus K^*} \rho_l(L-l) \ge \dots \\
\dots &\ge f(K+K^*) - \sum_{l\in K\setminus K^*} \rho_l(L-l) \\
&\ge f(K+K^*) - \sum_{l\in K\setminus K^*} \rho_l(K+K^*-l) \\
&\ge f(K+K^*-l') - \sum_{l\in K\setminus (K^*+l')} \rho_l(K+K^*-l) \ge \dots \\
\dots &\ge f(K^*),
\end{aligned}$$

where the above inequalities exploit the submodularity property of f. Hence, the tightest cut at x^* is the one associated to K^*, which concludes the proof. □

4.2. Submodular cuts for the MCFLRU

Let us now consider the function $\hat{v}_s : 2^L \mapsto \mathrm{R}$ defined for each $s \in S$ as follows:

$$\hat{v}_s(K) = \frac{\sum_{l\in K} a_{sl}}{1 + \sum_{l\in K} a_{sl}} = \frac{Z^s_K}{1 + Z^s_K} \tag{11}$$

where

$$Z^s_K = \sum_{l\in K} a_{sl}.$$

For each $K \subseteq L$, and each $s \in S$, the function $\hat{v}_s(K)$ calculates the probability that customer s chooses a facility from the subset K.

Moreover, for a customer $s \in S$, a set $K \subset L$, and a facility $l \in L$, let

$$\rho_{sl}(K) = \hat{v}_s(K+l) - \hat{v}_s(K)$$

denote the marginal contribution of adding l to $K \subset L$ for the function $\hat{v}_s$. The following Lemma was proven in Benati (1997):

Lemma 4. *For each $s \in S$, the function $\hat{v}_s(\cdot)$ is submodular and non-decreasing.*

The latter property can be exploited to derive an alternative MILP formulation for the MCFLRU.

Proposition 5. *The MCFLRU can be equivalently stated as the following (extended, but sparse) MILP formulation with $|L| + |S|$ variables:*

$$\max \sum_{s\in S} d_s \nu_s \tag{12a}$$

$$\nu_s \le \hat{v}_s(K) + \sum_{l\in L\setminus K} \frac{a_{sl}x_l}{(1+Z^s_K)(1+Z^s_{K+l})} - \frac{1}{1+Z_L} \sum_{l\in K} \frac{a_{sl}(1-x_l)}{1+Z_{L-l}} \quad s \in S, K \in K_X \tag{12b}$$

$$x \in X, \tag{12c}$$

where the function $\hat{v}_s(\cdot)$ is defined by (11).

Proof. Lemma 4, together with the separability of the objective function and Proposition 3, implies that the objective function can be stated as $\Sigma d_s \nu_s$ where, for each $s \in S$, the value of ν_s is upper bounded by submodular cuts as follows:

$$\nu_s \le \hat{v}_s(K) + \sum_{l\in L\setminus K} \rho_{sl}(K)x_l - \sum_{l\in K} \rho_{sl}(L-l)(1-x_l) \; s \in S, K \subseteq K_X. \tag{13}$$

For each $s \in S$, $l \in L$ and $K \subseteq L$, marginal contributions $\rho_{sl}(K)$ are calculated as:

$$\rho_{sl}(K) = \hat{v}_s(K+l) - \hat{v}_s(K) = \frac{a_{sl}}{(1+Z^s_{K+l})(1+Z^s_K)}.$$

After replacing the values for ρ_{sl} in (13), we obtain the submodular cuts (12b). □

The intuition behind the cuts (12b) is as follows: given a set $K \subset L$, and the value $\hat{v}_s(K)$, if we include an element from $L \setminus K$, the value of $\hat{v}_s(K)$ increases by at most $\rho_{sl}(K)$. Alternatively, if we exclude an element from K, the value of $\hat{v}_s(K)$ decreases by at least $\rho_{sl}(L-l)$, which is the marginal contribution assuming that all facilities but l have been selected. Due to the submodularity of the function $\hat{v}_s(\cdot)$, we have $\rho_{sl}(L-l) \le \rho_{sl}(K-l)$, hence the right-hand side provides a valid upper bound on the value of ν_s, for all $s \in S$ and all $K \in K_X$.

In a similar way, one can consider an additional family of submodular cuts, namely:

$$\nu_s \le \hat{v}_s(K) + \sum_{l\in L\setminus K} \rho_{sl}(\emptyset)x_l - \sum_{l\in K} \rho_{sl}(K-l)(1-x_l) \quad s \in S, K \subseteq L. \tag{14}$$

In these cuts, the marginal contribution of elements $l \in K$ is taken as it is, but the contribution of adding an $l \notin K$ is overestimated assuming that no facility has been selected (i.e., we have $\rho_{sl}(\emptyset) \ge \rho_{sl}(K)$). In Nemhauser and Wolsey (1981), the authors show that one can equivalently replace (9b) by (14), to derive another valid MILP reformulation of the problem. As in Proposition 3, one can easily show that also these cuts do not need to be imposed for every $K \subseteq L$, and that it is sufficient to consider $K \subseteq K_X$.

Branch-and-cut implementation. Separation of submodular cuts (12b) and (14) imposed at integer feasible points $x \in X$ can be performed in polynomial time. Similarly to the OA cuts, separating them on the fly and integrating them within a branch-and-cut framework leads to a viable exact procedure.

Given an integer candidate solution $x^* \in X$ and the current vector ν^*, according to the result of Proposition 5, it is sufficient to check if there exists $s \in S$ such that

$$\nu^*_s > \hat{v}_s(K^*),$$

where $K^* = \{l \in L : x^*_l = 1\}$. If such s is found, the corresponding submodular cuts (12b) and (14) associated to the set K^* (which are globally valid) are inserted into the model.

For the MCRU, it is sufficient to consider submodular cuts of the form $\nu_s \le \hat{v}_s(K) + \sum_{l\in L\setminus K} \rho_{sl}(K)x_l$, as the set X contains only cardinality constraints (see, e.g., Nemhauser & Wolsey, 1981). However,

cuts (12b) and (14) may still be useful in improving the value of the LP-relaxation and cutting off fractional infeasible points. This is why in our default implementation we always separate (12b) and (14).

We remark that separating violated cuts of the form (9b) for the MCRU is an NP-hard problem. This is why Proposition 3 is relevant, because it allows us to separate these cuts only in the integer points of X, which can be done efficiently. Similarly, separation of fractional points $x^* \in X$ is NP-hard. In Benati and Hansen (2002), a heuristic procedure was considered instead. The obtained results indicate that the heuristic generation of submodular cuts is non-efficient and time consuming. In our default implementation we therefore refrain from the separation of fractional points.

4.3. A combined approach: Outer-approximation and submodular cuts

Finally, a natural question arises: would it be useful to combine OA and submodular cuts within the same branch-and-cut procedure? Assuming that separation oracle is applied to integer points only, Remark 1 given below shows that the two B&C approaches, one based on OA cuts and the other based on submodular cuts, do not dominate each other. This is why in our computational study we also investigate the third B&C procedure, which is a combined approach in which OA constraints (8b) are enhanced by submodular cuts (12b) and (14).

Remark 1. Consider a set $\bar{K} \in K_X$. The associated outer-approximation cut and the submodular cut do not dominate each other.

To see this, let us denote by R_{OA} and R_{SC} the right-hand-side of the OA and the submodular cut, respectively, evaluated in $\bar{x}$ where $\bar{x}_l = 1$ if and only if $l \in \bar{K}$. In that case, the R_{OA} and R_{SC} have the same value, which is $\hat{v}_s(\bar{K}) = \hat{w}_s(\bar{x})$. Consider now $l' \notin \bar{K}$ such that $a_{sl'} > 0$. By evaluating the right-hand-side of the two cuts in the point x^* such that $x^*_l = 1$ if and only if $l \in \bar{K} + l'$, we obtain

$$R_{OA} := \hat{w}_s(\bar{x}) + \frac{a_{sl'}}{(1 + Z^s_{\bar{K}})^2},$$

$$R_{SC} := \hat{v}_s(\bar{K}) + \frac{a_{sl'}}{(1 + Z^s_{\bar{K}})(1 + Z^s_{\bar{K}+l'})},$$

and since $Z^s_{\bar{K}+l'} > Z^s_{\bar{K}}$, we have $R_{OA} > R_{SC}$. Finally, let $l' \in \bar{K}$ such that $a_{sl'} > 0$. By taking a point x^* such that $x^*_l = 1$ if and only if $l \in \bar{K} - l'$, we obtain

$$R_{OA} := \hat{w}_s(\bar{x}) - \frac{a_{sl'}}{(1 + Z^s_{\bar{K}})^2},$$

$$R_{SC} := \hat{v}_s(\bar{K}) - \frac{a_{sl'}}{(1 + Z^s_{L})(1 + Z^s_{L-l'})},$$

that is, $R_{OA} < R_{SC}$ unless $\bar{K} = L$.

5. Computational study

5.1. Description of the experiments

The purpose of this computational study is to provide a comparison of the proposed branch-and-cuts against the state-of-the-art exact approaches for the MCRU that have been recently computationally investigated in Freire et al. (2016a). The best performing approaches from the literature, according to Freire et al. (2016a), are:

- **CP** A concave programing approach proposed by Benati and Hansen (2002), that solves the continuous relaxation of problem (4) using a gradient algorithm and embeds this calculation into a B&B procedure.
- **Lin** A linearization technique presented in Haase (2009), that yields a compact MILP formulation with additional $|L| \times |S|$ continuous variables. In our experiments, we used a strengthened variant of this formulation presented in Freire et al. (2016a).
- **MUG** A greedy algorithm presented in Freire et al. (2016a) for computing valid upper bounds, embedded into a B&B procedure.

These three approaches are compared against our three proposed branch-and-cuts:

- **OA** The B&C based on outer-approximation cuts (8b) (cf. Section 3).
- **SC** The B&C based on submodular cuts (12b) and (14) (cf. Section 4).
- **OA+SC** The B&C based on a mix between **OA** and **SC**, i.e., violated OA and submodular cuts are inserted on the fly, as long such cuts can be found.

Implementation details. For solving MILPs, we used IBM-ILOG CPLEX 12.6 as our MILP solver (under default settings). The cuts in **OA** and **SC** are implemented using the `lazy-cut callback` routine and they are applied globally in the B&B tree each time that an integer solution is found. For all approaches, an initial feasible solution is provided by running a greedy algorithm that adds in each step the facility that results in the highest increment of the objective function. All computations were made on machines running Linux 2.6.32 under x86_64 architecture, with two quad-core Intel Xeon E5-2650 processors and 146 Gigabytes of RAM. Each run was performed on a single-core. As a non-linear solver (required for solving **CP**), we used NLopt (see http://ab-initio.mit.edu/wiki/index.php/NLopt) with the Method of Moving Asymptotes algorithm (Svanberg, 2002), which had the best performance among different local-gradient based algorithms implemented in that library.

Benchmark instances. The six approaches listed above are benchmarked using the following three datasets:

- ORLIB dataset, which consists of 11 problems taken from ORLIB's uncapacitated facility location benchmark set (Beasley, 1990; 2005) by introducing an incumbent competitor. Eight problems with $|S| = 50$, $|L| \in \{25, 50\}$ and three problems with $|S| = 1000$, $|L| = 100$ are considered.
- HM14 dataset, which includes randomly generated instances on a plane, proposed by Haase and Müller (2014). For this dataset we have $|S| \in \{50, 100, 200, 400\}$ and $|L| \in \{25, 50, 100\}$.
- P&R-NYC dataset, which comes from a large-scale park-and-ride location problem in New York City described in Freire et al. (2016b), originating from a work of Aros-Vera et al. (2013). These are the largest and the most challenging instances from the MCRU literature, with $|S| = 82,341$, $|L| = 59$, see Fig. 1.

Each problem from the above datasets results in 81 different MCRU instances: a fixed number of selected facilities r is varied between 2 and 10, and different scaling factors for the utility functions v_{sa} and v_{sl} are considered. The total number of instances in each dataset is 891, 972 and 81, respectively. For a more detailed description of each dataset, see Freire et al. (2016a).

5.2. Results on small and medium size instances

We first focus on small and medium size instances, namely those from datasets ORLIB and HM14. For each of the six

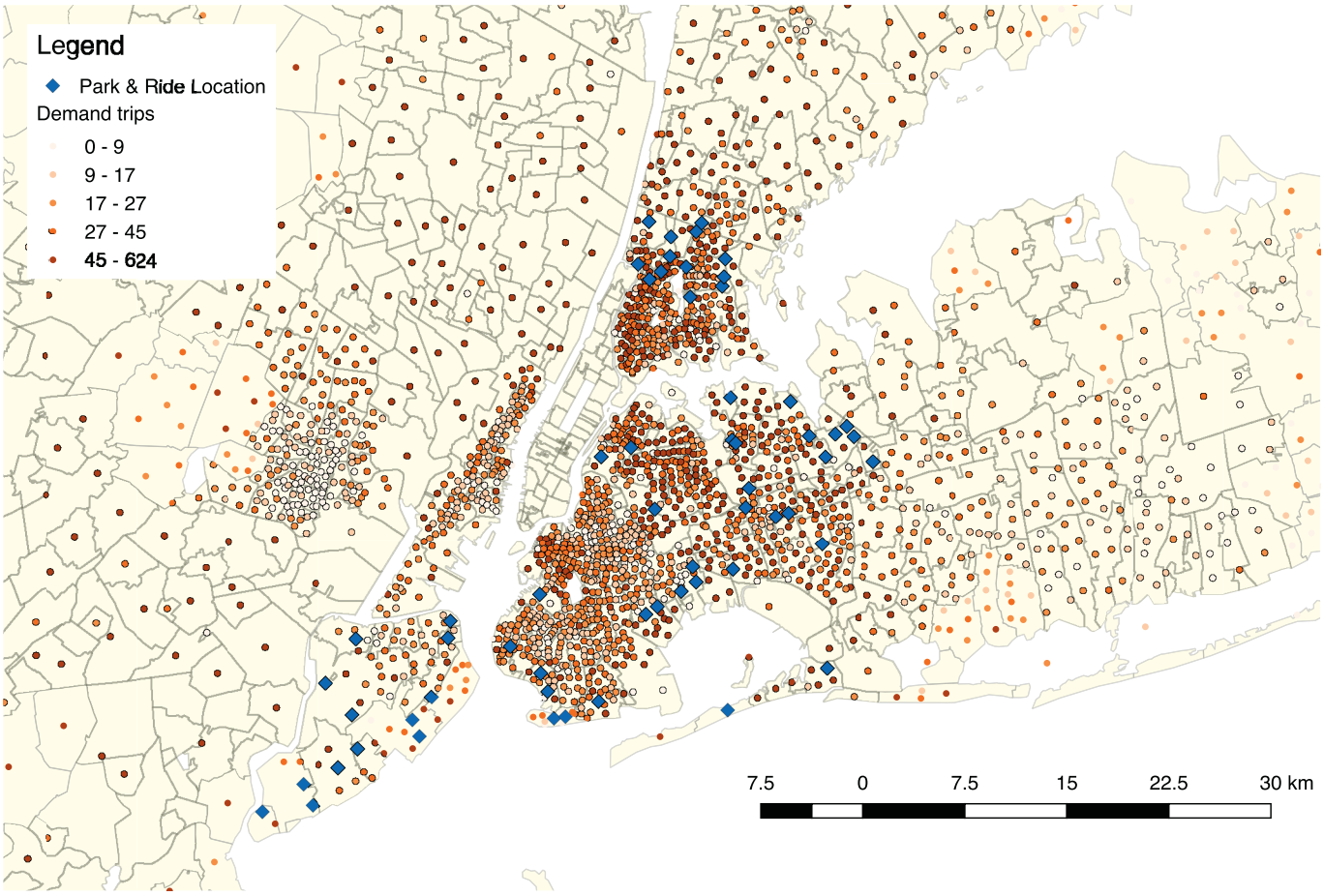

Fig. 1. Diagram of NYC instance. Each circle represents a trip origin to Manhattan, colored according to its demand. There are 3184 origins outside Manhattan and 317 destinations in Manhattan, making 82,341 trips in total. Blue diamonds represent the 59 potential Park-and-Ride facilities. Customers (represented by each trip) decide between an option of taking a direct auto trip from the trip's origin to its destination (the incumbent competitor) and the option of going from the trip's origin to one of the newly opened P&R facilities and then using public transportation to its final destination. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

approaches, Table 1 reports: the number of instances solved to optimality within a time limit of one hour, the average CPU time (in seconds) *among those instances solved to optimality*, the number of nodes in the B&B tree, and the initial gap at the root node. This gap is calculated between the initial greedy solution and the upper bound reported by the MILP solver obtained at the root node after applying all cuts and before starting the B&B procedure. The largest number of instances solved to optimality is indicated in bold, as well as the best computing time(s).

Table 1 indicates that **OA**, resp. **SC** is the fastest approach for ORLIB, resp. HM14 instances, capable of solving (almost) all instances to optimality in the respective dataset. When comparing the computing times, we also notice that the performance of **OA+SC** is very similar to the one of the best performing approach, and only sometimes it is slightly worse (which is not surprising, given that **OA+SC** generates more cuts than **OA** or **SC** alone).

On the smallest instances from the ORLIB dataset (*cap101-104* and *cap131-134*), **OA** and **OA+SC** approaches outperform **CP**, state-of-the-art approach for this dataset (prior to our study), by more than an order of magnitude. These two approaches take less than 0.1 second to prove the optimality, by visiting less than a dozen of B&B nodes. On the contrary, **SC** is not able to solve all instances from this data set. Nevertheless, combination of OA and submodular cuts leads to the tightest root node bounds.

Medium size instances (*capa, capb, capc*) appear to be more challenging. **Lin** is not able to solve any of these instances within an hour, and **SC** solves only a single one. **MUG** manages to prove optimality in about 25% of the cases, wheres this rate for **CP** is about 65%. On the contrary, **OA** and **OA+SC** solve all but one, respectively, 16 instances to optimality. Root gaps obtained by **OA** are very small and comparable to those from **CP**, which explains its excellent performance on these instances. On the contrary, the gaps produced by **SC** are considerably higher, which leads to larger B&B trees, resulting in a poor performance of **SC** for this dataset. Also for these instances, combining both types of cuts (**OA+SC**) turns out to be beneficial, resulting in the root gaps and the sizes of the B&B tree being even smaller than those obtained by **OA**.

To have a closer look at the performance of our approaches on the ORLIB dataset, we also ran the experiments with a time limit set to eight hours. The obtained results are reported in Table 2. It can be seen that optimal solutions for all medium size instances are obtained by **OA**, with average CPU times lying between two and five minutes. Focusing on the performance of our three B&C procedures, we notice that all three approaches enumerate a similar number of branch-and-bound nodes. However, the number of submodular cuts is two orders of magnitude higher than the respective number of OA cuts, which explains the poor performance of **SC**, and the weaker performance of **OA+SC** on this dataset. Observe that the quality of the approximation for **SC** is similar to that of **Lin**, and considerably worse than the one of **OA** and **CP**. Due to this fact, **OA+SC** uses larger number of cuts than **OA** without an important reduction on its gaps.

Table 1
Results for ORlib (up) and HM14 (down) datasets, grouped by problem name (81 instances per row). Time limit set to one hour. Best values (the largest number of instances solved, or the lowest computing time) are shown in bold. Average values are calculated by taking into account only those instances solved to optimality by the respective approach.

	#(Solved instances)						Computing time (s)*						B&B nodes*						Root gap*					
Name	Lin	CP	MUG	OA	SC	OA+SC	Lin	CP	MUG	OA	SC	OA+SC	Lin	CP	MUG	OA	SC	OA+SC	Lin	CP	MUG	OA	SC	OA+SC
cap101	**81**	**81**	**81**	**81**	75	**81**	13.8	0.4	0.2	**0.0**	100.9	**0.0**	4111	34	9057	6	1279	2	10.2	0.3	8.4	0.5	5.1	0.1
cap102	**81**	**81**	**81**	**81**	75	**81**	14.1	0.9	0.2	**0.0**	116.3	**0.0**	4840	170	11,596	6	1460	2	10.3	0.4	8.6	0.7	5.1	0.1
cap103	**81**	**81**	**81**	**81**	**81**	**81**	6.6	0.7	0.1	**0.0**	199.7	**0.0**	1387	86	7559	4	1845	1	10.3	0.5	8.9	0.7	5.0	0.1
cap104	**81**	**81**	**81**	**81**	78	**81**	8.3	0.1	0.2	**0.0**	151.2	**0.0**	1862	7	10,026	4	1495	1	10.2	0.2	8.5	0.5	5.1	0.1
cap131	78	**81**	**81**	**81**	61	**81**	253.1	1.6	7.5	**0.1**	94.6	**0.1**	39303	59	296,281	7	997	2	11.9	0.5	10.5	0.9	6.5	0.2
cap132	79	**81**	**81**	**81**	62	**81**	213.2	0.5	5.9	**0.1**	145.0	**0.1**	37362	15	225,039	4	855	2	12.0	0.5	11.0	0.8	6.4	0.1
cap133	78	**81**	**81**	**81**	62	**81**	199.6	0.3	14.2	**0.1**	219.2	**0.1**	34694	8	543,304	2	1404	1	12.3	0.3	10.9	0.7	6.4	0.1
cap134	79	**81**	**81**	**81**	60	**81**	218.3	0.9	13.9	**0.1**	97.2	**0.1**	39494	36	525,487	3	982	2	12.2	0.5	11.1	0.7	6.3	0.1
capa	–	48	21	**81**	–	74	–	737.5	356.3	**298.4**	–	229.9	–	112	308,778	2016	–	888	–	0.2	31.0	1.4	–	0.9
capb	–	49	23	**81**	1	78	–	665.6	471.4	**120.8**	3039.9	193.4	–	110	393,240	1143	1245	760	–	0.1	30.2	1.3	16.3	0.9
capc	–	53	21	**80**	–	75	–	477.0	296.5	**271.8**	–	413.3	–	61	225,427	1812	–	1051	–	0.1	30.1	1.4	–	1.0

		#(Solved instances)						Computing time (s)*						B&B nodes*						Root gap*					
$\|S\|$	$\|L\|$	**Lin**	**CP**	**MUG**	**OA**	**SC**	**OA+SC**	**Lin**	**CP**	**MUG**	**OA**	**SC**	**OA+SC**	**Lin**	**CP**	**MUG**	**OA**	**SC**	**OA+SC**	**Lin**	**CP**	**MUG**	**OA**	**SC**	**OA+SC**
50	25	**81**	69	**81**	**81**	**81**	**81**	28.1	13.8	0.2	0.4	**0.0**	**0.0**	1	451	11,070	1761	1	0	0.6	9.4	19.3	12.2	0.2	0.1
50	50	**81**	67	79	**81**	**81**	**81**	26.6	211.1	106.3	0.7	**0.1**	**0.1**	7	5375	4,141,023	3219	3	1	0.8	9.1	22.4	12.4	0.2	0.1
50	100	**81**	48	61	70	**81**	**81**	270.3	272.5	167.1	94.1	**0.1**	**0.1**	33	461	4,480,988	262,864	8	7	0.5	5.0	27.7	15.0	0.5	0.3
100	25	**81**	67	**81**	**81**	**81**	**81**	19.8	55.3	1.7	3.8	**0.0**	**0.0**	0	2573	40,582	12,740	1	0	0.5	8.4	22.8	13.7	0.6	0.5
100	50	**81**	58	72	80	**81**	**81**	22.9	162.6	207.9	127.4	**0.1**	**0.1**	5	1696	4,778,935	208,127	1	1	0.3	8.6	32.9	17.0	0.3	0.2
100	100	**81**	49	58	68	**81**	**81**	162.7	289.1	200.3	60.4	0.7	**0.5**	70	368	2,959,530	86,653	70	18	1.1	5.3	28.6	14.0	0.7	0.5
200	25	**81**	74	**81**	**81**	**81**	**81**	14.3	142.7	9.4	1.4	**0.1**	**0.1**	2	2922	110,327	3175	1	0	0.4	11.4	27.9	13.6	0.2	0.1
200	50	**81**	57	67	73	**81**	**81**	39.6	254.7	211.1	57.8	**0.2**	**0.2**	2	1316	2,400,039	86,289	2	2	0.4	10.2	33.1	16.5	0.5	0.4
200	100	**81**	46	46	63	**81**	**81**	663.2	404.5	112.7	74.4	2.0	**1.2**	154	228	686,808	35,141	56	26	0.9	5.4	32.2	17.7	0.5	0.3
400	25	**81**	77	**81**	**81**	**81**	**81**	34.3	133.0	11.7	2.8	**0.1**	0.2	1	1367	49,637	4808	2	1	0.4	10.9	29.3	13.4	0.2	0.2
400	50	**81**	52	62	72	**81**	**81**	284.4	388.3	259.9	116.9	**0.5**	0.6	11	970	1,044,952	72,659	3	2	0.5	9.7	35.1	17.7	0.3	0.4
400	100	76	36	45	60	**81**	**81**	552.2	355.7	299.2	34.4	4.0	**2.5**	114	172	758,270	6168	62	21	0.7	5.7	32.7	15.4	0.6	0.5

* Average among solved instances.

Table 2
Results for mid-sized instances of ORLIB. Time limit set to eight hours.

	Method	# Solved Instances	Computing Time (s)*	Root gap*	B&B nodes*	Number of cuts*
capa	**Lin**	12	10341.3	15.080	9142.9	
	CP	69	3400.9	0.417	680.6	
	MUG	30	2347.1	30.344	2487094.1	
	OA	**81**	286.8	1.393	2016.3	5673.5
	SC	3	5862.4	14.227	1493.3	646488.0
	OA+SC	**81**	980.5	1.050	2029.8	14406.9
capb	**Lin**	13	12330.5	15.074	7472.5	
	CP	65	2388.8	0.314	589.4	
	MUG	33	3372.3	27.310	3338846.3	
	OA	**81**	118.4	1.339	1143.1	5361.5
	SC	3	4001.2	15.522	1246.7	586074.7
	OA+SC	80	303.8	0.955	1002.7	12895.5
capc	**Lin**	12	11448.0	15.575	7361.4	
	CP	62	1302.3	0.284	283.4	
	MUG	30	1701.8	29.415	1721441.1	
	OA	**81**	316.8	1.410	1993.1	6034.7
	SC	3	5014.9	16.270	1469.7	682288.7
	OA+SC	**81**	1086.1	1.101	2001.1	15074.6

* Average among solved instances.

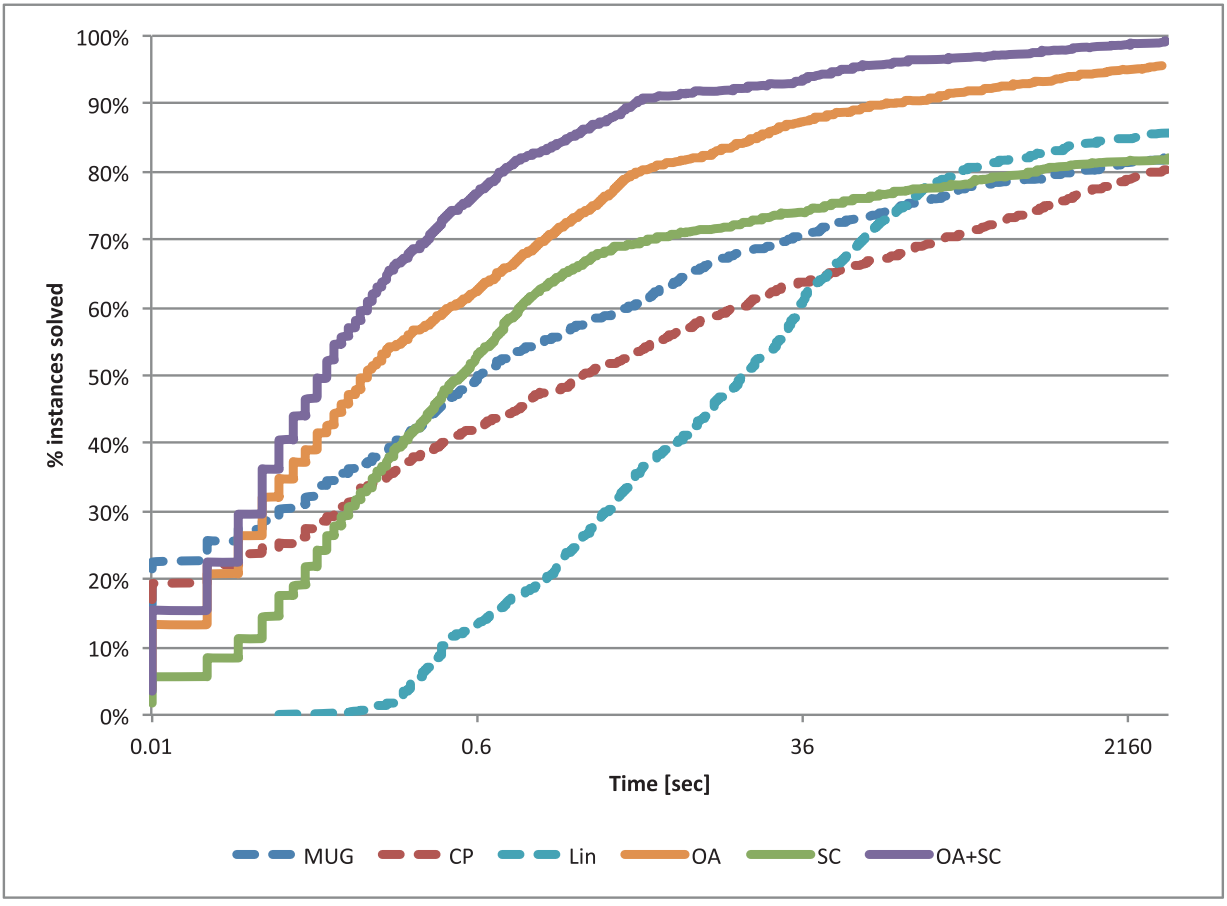

Fig. 2. Performance profile of each approach for HM14 and ORLIB instances.

A slightly different behavior can be observed for HM14 instances. As detailed in Freire et al. (2016a), the linear reformulation **Lin** allows to obtain root gaps smaller than 1% for most of the instances, allowing it to solve all but four instances to optimality, with average computing times ranging between 15 seconds and 10 minutes. Approaches **CP**, **MUG** and **OA** suffer from the very weak root relaxation bounds and do not manage to solve some of the smallest among these instances within one hour. The tightest root gaps are obtained by **SC**. Given that each **SC** subproblem of the B&B tree can be solved much faster than for **Lin**, the computing times of **SC** are two to three orders of magnitude faster than the respective CPU times for **Lin**. Similarly to the ORLIB dataset, **OA+SC** combines the best of the two families of cuts and allows to solve all HM14 instances within fractions of a second, providing even better root gaps than **SC**.

Summary of results on ORLIB and HM14 datasets. The performance chart presented in Fig. 2 summarizes our results over these two

Table 3
Results for NYC dataset, grouped by r (9 instances per row).

	#(Solved instances)					Computing time (s)*					B&B nodes*					Root gap*				
r	CP	MUG	OA	SC	OA+SC	CP	MUG	OA	SC	OA+SC	CP	MUG	OA	SC	OA+SC	CP	MUG	OA	SC	OA+SC
2	6	9	9	9	9	3727.1	69.8	1363.5	455.9	970.6	13	111	139	1	3	6.4	14.9	11.8	0.3	0.4
3	6	9	9	9	9	2485.1	170.8	2177.5	514.3	573.0	11	271	462	4	3	2.5	9.0	6.9	0.5	0.5
4	5	9	9	9	9	2338.2	411.6	2950.7	603.1	674.1	7	725	583	16	2	1.8	6.1	3.1	0.9	0.9
5	5	9	9	9	9	1813.5	1303.0	783.5	504.4	570.6	7	2204	201	2	2	1.1	4.2	2.0	1.1	1.1
6	7	9	9	9	9	4707.5	3187.6	464.7	429.9	595.7	7	6753	80	1	1	0.4	2.8	1.1	1.3	1.1
7	6	9	9	9	9	1169.4	6562.4	417.6	422.3	509.9	5	13,826	103	0	0	0.3	1.9	0.6	0.8	0.8
8	6	9	9	9	9	2441.5	10157.9	391.1	602.7	537.7	8	32,078	74	1	1	0.2	1.5	0.4	0.9	0.9
9	6	6	9	9	9	4025.6	2995.2	397.0	429.0	511.7	12	6015	28	1	0	0.1	0.4	0.2	0.9	0.9
10	5	6	9	9	9	1469.8	3843.9	414.0	412.1	503.4	7	7370	21	0	1	0.1	0.3	0.1	0.0	0.9

* Average among solved instances.

datasets, showing the percentage of instances solved to optimality (given on the y-axis) within a given computing time (given on the x-axis). A point with coordinates (x, y) in this chart indicates that for y% of the instances, the computing time is $\leq x$ seconds. Notice that computing time (which is given in seconds) on the x-axis is shown using logarithmic scale. Fig. 2 demonstrates that two of the three B&C approaches proposed in this paper drastically outperform the state-of-the art exact methods. In particular, by combining outer approximation with submodular cuts (**OA+SC**) we manage to derive a robust B&C framework with a relatively stable performance over different types of benchmark instances. **OA+SC** draws advantage of the strength of the two families of cuts in different settings. It significantly outperforms all the remaining approaches, allowing to solve more instances to optimality and in a much shorter computing time. In fact, the time required by the former approaches to solve 70% of the instances is between 31 seconds (**MUG**) and 207 seconds (**CP**). Our proposed **OA+SC** approach requires 0.33 seconds to solve the same percentage of instances. Similar numbers are obtained for solving 80% of the instances: the former approaches require between 272 seconds (**Lin**) and 2110 seconds (**CP**), versus 0.83 seconds (**OA+SC**). Finally, for solving 90% of the instances all the former approaches require more that one-hour, whereas **OA+SC** solves the same number of instances in 4.37 seconds. In general, the excellent performance of **OA+SC** can be explained by a good balance between the size of the model (in terms of the number of variables) and the quality of the root node relaxation (which is similar to **CP**, and considerably smaller than **MUG**) resulting in smaller B&B trees.

5.3. Results on large scale instances

For P&R-NYC dataset, Table 3 reports the results of **CP**, **MUG**, **OA**, **SC** and **OA+SC** obtained by setting the time limit to 8 hours. Recall that **Lin** can not be applied to this dataset due to the prohibitive size of the resulting MILP formulation. There are 9 instances per row, grouped by the value of r. Our three B&C approaches are the only ones able to solve all instances to optimality, whereas **MUG** and **CP** fail to do so in 6, respectively 29, cases. As before, the gaps at the root node are close to 1%, and only a few nodes of the B&B tree are required to find the optimal solution. Interestingly, the performance of the B&C approaches is not particularly affected by the number of chosen facilities r, which is a serious drawback of **MUG**, the previously best-known approach for this dataset.

6. Conclusions and future work

In this article a new methodology for solving the maximum capture problem with random utilities and related problems is presented. This methodology is based on the first-order approximation of the concave non-linear function, which can be applied using a cutting plane framework. The approach is enhanced by submodular cuts that very often provide good linear approximation of the original problem. Compared to the existing models from the literature, our approach does not considerably increase the size of the MILP reformulation. At the same time, combination of outer-approximation and submodular cuts results in a branch-and-cut procedure with a relatively stable and robust performance over various types of benchmark instances. Extensive computational experiments show that our approach significantly outperforms the state-of-the-art approaches, with obtained speed-ups of two to three orders of magnitude.

Our methodology does not require any particular structure on the set X of feasible facility configurations, which also makes it suitable for more general competitive facility location problems. Possible examples include situations in which (i) budget constraints are imposed on the set of open facilities, (ii) simultaneous facility location and design decisions have to be made, or (iii) some infrastructure requirements (such as connectivity) are imposed on the set of open facilities.

Furthermore, our exact approach is not restricted to competitive facility location problems with multinomial logit models only. The algorithmic framework could be useful for any other type of customer utility functions which can be represented as $f_s(\Sigma_{l\in L}\alpha_{sl}x_l)$ where f_s is strictly concave and increasing function used to capture the effect of diminishing marginal gains by opening additional facilities, and $\alpha_{sl}\geq 0$ are utility values (see, e.g., Ben-Akiva & Bierlaire, 1999). Relevant examples from the literature include the Huff-type utilities, frequently used in marketing and location theory, where the values of α_{sl} are directly proportional to the attractiveness and indirectly proportional to the distance of facility l to customer s (see, e.g., Aboolian, Berman, & Krass, 2007).

Finally, along the lines of research proposed in Ahmed and Atamtürk (2011), Yu and Ahmed (2017), further enhancements of submodular cuts are possible. It would be interesting to study possible lifting procedures of submodular cuts for more general facility configurations X, and their effect on the branch-and-cut performance.

Acknowledgments

Eduardo Moreno acknowledges the financial support of the FONDECYT Grant 1161064.

References

Aboolian, R., Berman, O., & Krass, D. (2007). Competitive facility location model with concave demand. *European Journal of Operational Research, 181*(2), 598–619.
Ahmed, S., & Atamtürk, A. (2011). Maximizing a class of submodular utility functions. *Mathematical Programming, 128*(1–2), 149–169.
Aros-Vera, F., Marianov, V., & Mitchell, J. E. (2013). p-hub approach for the optimal park-and-ride facility location problem. *European Journal of Operational Research, 226*(2), 277–285.

Beasley, J. (1990). Or-library: distributing test problems by electronic mail. *Journal of the Operational Research Society, 40*(11), 1069–1072.
Beasley, J. (2005). Orlib: Operations research library. http://people.brunel.ac.uk/~mastjjb/jeb/info.html.
Ben-Akiva, M., & Bierlaire, M. (1999). Discrete choice methods and their applications to short term travel decisions. In R. W. Hall (Ed.), *Handbook of transportation science* (pp. 5–33)). Boston, MA: Springer US.
Benati, S. (1997). Submodularity in competitive location problems. *Ricerca Operativa, 26*, 3–34.
Benati, S., & Hansen, P. (2002). The maximum capture problem with random utilities: Problem formulation and algorithms. *European Journal of Operational Research, 143*(3), 518–530.
Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., et al. (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization, 5*(2), 186–204.
Duran, M. A., & Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming, 36*(3), 307–339.
Fischetti, M., Ljubić, I., & Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research, 253*(3), 557–569.
Fischetti, M., Ljubić, I., & Sinnl, M. (2017). Redesigning Benders Decomposition for Large Scale Facility Location. *Management Science, 63*(7), 2146–2162.
Fletcher, R., & Leyffer, S. (1994). Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming, 66*(1), 327–349.
Freire, A. S., Moreno, E., & Yushimito, W. F. (2016a). A branch-and-bound algorithm for the maximum capture problem with random utilities. *European Journal of Operational Research, 252*(1), 204–212.
Freire, A. S., Moreno, E., & Yushimito, W. F. (2016b). A column generation approach for the optimal selection of park-and-ride facilities. In *Proceedings of the ninth triennial symposium on transportation analysis (Tristan IX)*.
Geoffrion, A. (1972). Generalized benders decomposition. *Journal of Optimization Theory and Applications, 10*, 237–260.
Gollowitzer, S., & Ljubić, I. (2011). MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research, 38*(2), 435–449.
Haase, K. (2009). Discrete location planning. *Technical Report, WP-09-07*. Institute for Transport and Logistics Studies, University of Sydney.
Haase, K., & Müller, S. (2012). Management of school locations allowing for free school choice. *Omega, 41*(5), 847–855.
Haase, K., & Müller, S. (2014). A comparison of linear reformulations for multinomial logit choice probabilities in facility location models. *European Journal of Operational Research, 232*, 689–691.
Haase, K., & Müller, S. (2015). Insights into clients' choice in preventive health care facility location planning. *OR Spectrum, 37*(1), 273–291.
Laporte, G., Ledesma, J. R., & González, J. S. (2003). A branch-and-cut algorithm for the undirected traveling purchaser problem. *Operations Research, 51*(66), 940–951.
Mittelmann, H. (2014). MINLP benchmark. http://plato.asu.edu/ftp/minlp_old.html.
Müller, S., Haase, K., & Kless, S. (2009). A multiperiod school location planning approach with free school choice. *Environment and Planning A, 41*(12), 2929–2945.
Nemhauser, G., & Wolsey, L. (1981). Maximizing submodular set functions: Formulations and analysis of algorithms. In P. Hansen (Ed.), *Annals of discrete mathematics (11) studies on graphs and discrete programming*. In *North-Holland Mathematics Studies: vol. 59* (pp. 279–301). North-Holland.
Quesada, I., & Grossmann, I. (1992). An LP/NLP based branch-and-bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering, 16*, 937–947.
ReVelle, C. (1986). The maximum capture or "sphere of influence" location problem: Hotelling revisited on a network. *Journal of Regional Science, 26*(2), 343–358.
Svanberg, K. (2002). A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal of Optimization, 12*(2), 555–573.
Vielma, J. P., Dunning, I., Huchette, J., & Lubin, M. (2017). Extended formulations in mixed integer conic quadratic programming. *Mathematical Programming Computation, 9*(3), 369–418.
Yu, J., & Ahmed, S. (2017). Maximizing a class of submodular utility functions with constraints. *Mathematical Programming, 162*(1), 145–164.
Zhang, Y., Berman, O., & Verter, V. (2012). The impact of client choice on preventive healthcare facility network design. *OR Spectrum, 34*(2), 349–370.