A Tailored Branch-and-Benders-Cut Approach for Backup Covering Problems

informs
https://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0899-1499, EISSN 1526-5528

Submitted to *INFORMS Journal on Computing*

A tailored Branch-and-Benders-Cut approach for Backup Covering Problems

Edoardo Fadda
Dept. of Mathematical Sciences, Politecnico di Torino, Turin (Italy), edoardo.fadda@polito.it

Ivana Ljubić
ESSEC Business School, Cergy-Pontoise (France), ivana.ljubic@essec.edu

Daniele Manerba
Dept. of Information Engineering, Università degli Studi di Brescia, Brescia (Italy), daniele.manerba@unibs.it

Abstract. Backup covering problems represent a class of covering location models that are extremely important in applications involving decisions that need to comply with service redundancy. Among these models, the most used are the BACOP1 and the BACOP2, which can be seen as different generalizations of the well-known Maximal Covering Location Problem. Given an available budget in terms of facilities to open, BACOP1 aims at maximizing the demand covered twice while ensuring that the total demand is covered at least once, whereas BACOP2 aims at maximizing a weighted combination of the demand covered once and twice without any guarantee on the single coverage. Despite their practical importance, little attention has been given to the development of tailored solution algorithms for backup covering problems of such types. In this work, we exploit the special structure of BACOP1 and BACOP2 to derive a branch-and-cut approach based on the separation of different Benders cuts. Differently from classical Benders decomposition approaches, the cut separation is performed by leveraging combinatorial properties of the resulting subproblems and not through the solution of a linear program. We analyze the dominance properties of the generated cuts and present an empirical comparison of our approach against a state-of-the-art solver on a comprehensive set of large-scale instances. Notably, although the two problems share strong similarities and are addressed within a unified solution framework, the characteristics of the solution procedures and their computational results differ substantially.

Key words: Backup covering problems, Redundant coverage, BACOP1, BACOP2, Benders decomposition

1. Introduction

Covering Location Problems (CLPs) are among the most studied facility location problems in the combinatorial optimization literature (García and Marín 2019), with countless applications in spatial planning, logistics, transportation, healthcare, and many more. A typical CLP example is the well-known Maximal Covering Location Problem, or MCLP (Church and ReVelle 1974). Given a set of demand centers, a set of locations where to potentially place a facility, and the *covering set* of each location, i.e., the set of all the demand centers that would be covered in terms of service by a facility placed in such a location, the main goal of the MCLP is to find a location plan that maximizes the demand covered by the placed facilities, while complying with a certain budget restriction (which, in general, allows to install only up to a certain number of facilities).

Several generalizations or variants of the MCLP have been proposed to accommodate many realistic requirements in different application domains (the interested reader is referred to Farahani et al. 2012 or Snyder and Haight 2016). Some extensions of particular importance focus on redundant coverage aspects, generally aiming at maximizing the demand covered more than once. Such problems gain paramount importance in decisional settings for critical services, such as those linked to healthcare or essential infrastructures, where multiple coverages are planned to face facilities' congestion or disruption. The so-called *Backup Covering Problem of type 1* (BACOP1) and *of type 2* (BACOP2), introduced a long time ago in Hogan and ReVelle (1986), represent the most used models to address coverage redundancy. Given a certain number of facilities to open, BACOP1 is solely focused on maximizing the demand covered twice while ensuring that all the centers' demand is covered at least once, whereas BACOP2 pursues the maximization of a weighted (convex) combination of the demand covered once and twice without imposing any other requirement on the single coverage (i.e., if convenient, some demand centers might remain uncovered). Despite their practical relevance, relatively few studies have been conducted on both models, and the most related contributions concern their application in specific case studies. In particular, no ad-hoc solution algorithms have been proposed in the literature. This is probably because, despite their $\mathcal{NP}$-hardness, state-of-the-art Mixed-Integer Linear Programming (MILP) solvers can address most CLPs very fast, exploiting their powerful treatment of binary variables both at preprocessing and during the branch-and-bound search. BACOP1 and BACOP2 are no exceptions in this regard. However, as for most MILP problems, the solvers' performance degrades rapidly when the instance dimension grows, limiting their application in many realistic settings based on today's technologies. In fact, the development of social networks, large databases, and infrastructures able to collect millions of service requests frequently poses the necessity to solve in a bunch of seconds CLPs where the number of demand centers is many orders of magnitude greater than the number of potential facilities to install (Cordeau et al. 2019, Coniglio et al. 2022, Kahr et al. 2024).

In this paper, we propose a tailored Branch-and-Benders-Cut approach to solve backup covering problems of both types, in particular on the large-scale instances that remain out of reach for the standard MILP solvers. Our method exploits the specific structure of BACOP1 and BACOP2 that allows, once the location decisions have been fixed, to obtain a Linear Programming (LP) subproblem for deciding the value of the remaining variables. Actually, different from the classical Benders decomposition approach, we further exploit the combinatorial structure of the subproblem to derive a closed formula for its optimal solutions and, therefore, a separation procedure for the involved cuts that is faster than solving an LP problem. Moreover, exploiting the degree of freedom in the closed formula, we are able to derive several versions of Benders optimality cuts and prove dominance relations among them. The resulting approach, improved through several acceleration strategies (such as stabilization and a sparse matrix-based reformulation of the cuts), shows its superiority over available state-of-the-art solvers and technologies. In particular, over large instances with different topological properties, we demonstrate that our tailored methods can be used to obtain optimal solutions and consistently return feasible near-optimal solutions when little time is available for the resolution. Notably, even though we propose an overall unified framework for the two backup covering problems addressed, their peculiarities are reflected in tailored features of the solution approaches developed. We also remark that, to the best of our knowledge, this paper presents the first tailored *exact* solution approach for BACOP1 and BACOP2.

The manuscript is organized as follows. Section 2 reviews the relevant literature on backup covering problems and on similar Benders decomposition-based approaches. Section 3 presents the studied problems, their mathematical formulations, and the assumptions made. Sections 4 and 5 describe our tailored approaches for BACOP1 and BACOP2, respectively, while Section 6 presents the general Branch-and-Benders-Cut procedure and some accelerating strategies applied. Section 7 describes our experimental setting and the obtained results. Finally, Section 8 concludes the paper and sketches possible future research.

2. Literature review

In the following, we focus on CLPs addressing redundancy aspects (Section 2.1), and we review works constituting the methodological background behind our approach (Section 2.2).

2.1. Redundant covering location problems

Models that consider multiple coverages can be used to pursue sophisticated decision-making goals such as those related to the robustness or resiliency of the underlying service, i.e., the capacity to maintain a reasonable level of service against possible disruptions or congestion. Many practical applications are sensitive to redundancy features, and the specialized literature has focused in particular on those having a critical impact on the population. For example, particular attention has been given to the performance of the location solutions of healthcare facilities and services, such as ambulances, hospitals, and triage mobile centers (Brotcorne et al. 2003, Li et al. 2011, Zhang et al. 2017, Wang et al. 2021). Moreover, energy facilities or any type of capacity-limited services must be carefully installed to meet the demand using a fault-tolerance perspective, thus enlarging for the final users their possibility to be served by more covering facilities. Studies within this field can be found, e.g., in Li et al. (2013) or Chowdhury et al. (2020). Other applications of covering location models explicitly taking into account redundancy can be also found in the location of emergency warning sirens (Current and O'Kelly 1992), in the design of hierarchical healthcare organizations (Moore and ReVelle 1982, Mitropoulos et al. 2006), firefighting systems (Marianov 2017) and military operations (Armacost 1992). Finally, even for facilities rarely affected by disruptions but ensuring important services to the communities (schools, stations, green areas, markets), a redundant coverage is always preferred since it provides the users with more different alternatives, thus increasing social inclusion and equitable opportunities (Fadda et al. 2019, 2021a,b).

To explicitly address redundancy aspects, many models and perspectives have been proposed. The *Redundant Covering Location Problem* (RCLP) proposed by Daskin and Stern (1981) is a hierarchical optimization setting that seeks, among the optimal solutions of a classical minimum set covering problem, the one maximizing the demand covered at least twice. A similar model had been used by Plane and Hendrick (1977) for locating fire stations. Later, Hogan and ReVelle (1986) introduced and studied the BACOP1 and BACOP2 models, those exactly addressed in our paper. As already seen, with some differences, both models try to maximize the number of times a location is covered at least twice. Another well-known backup model is the *Double Standard Model* (DSM) firstly presented in Gendreau et al. (1997) linked to ambulances location. Here, two different covering radii (called *standards*) are considered for each demand center. The total demand must be covered at least once within the inner standard, while a certain proportion of the demand must also be within the outer radius. Moreover, the demand covered within the inner radius is maximized. Notice that, differently from RCLP, BACOP1, and BACOP2, which are binary problems, the DSM allows to install an integer number of facilities in each location (e.g., the number of ambulances to place at an emergency center), thus being particularly tailored for settings that need higher granularity. Extensions of DSM can be found in Doerner et al. (2005) and Doerner and Hartl (2008) for the minimization of the ambulance workload. Very recently, Fadda et al. (2024) generalized BACOP1, BACOP2, and DSM to their K-th coverage level. The authors obtained more sophisticated models able to address multiple covering needs in different ways and present theoretical and empirical properties of the resulting models against different topologies, covering radii, and budget availabilities.

Finally, despite being deterministic, some models are based on probabilistic principles, such as the *Maximum Expected Covering Location Problem* (see, e.g., Daskin 1983 or Sorensen and Church 2010). Here, each facility has a probability ρ of being unavailable and, therefore, a demand center covered by n facilities has a probability equal to $1-\rho^n$ of receiving the service. The goal is to install facilities, limited by a budget constraint, so as to maximize the sum of the demands of the points weighted by their probabilities of being served (see also the recent work of Wang et al. 2025 and further references therein).

2.2. Benders decomposition-based exact methods

Since the seminal work of its inventor (Benders 1962), the Benders decomposition and the corresponding derivation of cuts have become pillars in mathematical programming for addressing well-structured models. The method is particularly well suited for two-stage stochastic programming models (which exhibit a block-diagonal constraint structure in their scenario-based explicit form), but it has also been successfully applied to a variety of combinatorial optimization problems, including facility location (Wentges 1996) and multi-period production planning (Adulyasak et al. 2015). Two recent surveys by Rahmaniani et al. (2017) and Clautiaux and Ljubić (2025) collect and discuss many variants and applications of the Benders decomposition algorithm. The use of this technique is also very promising in the context of (real or virtual) social networks, where the input data dimension asks for some tailored decomposition approach (Kahr et al. 2024, Tanınmış et al. 2024).

Unfortunately, the Benders decomposition algorithm presents several limitations (e.g., slow convergence and numerical instability) and, therefore, it has been frequently substituted during the last decades by other general procedures. However, the method has experienced renewed interest in recent years because of the introduction of several improvements (Codato and Fischetti 2006 or Fischetti et al. 2010) and the possibility of hybridizing its peculiar features with other approaches. A promising option (implemented in our paper) is the so-called *Branch-and-Benders-Cut* approach, in which Benders cuts are separated within a branch-and-bound tree search. Interestingly, the importance of this methodology has not been perceived only within academic research environments. In fact, starting from version 12.7 onward, a partially automatic Benders algorithm has also been implemented in CPLEX, one of the most used state-of-the-art commercial MILP solvers (Bonami et al. 2020), and in SCIP, one of the most popular non-commercial solvers (Maher 2021).

Another important aspect exploited by our approach is the possibility of separating Benders cuts without resorting to the explicit resolution of LP problems. Cordeau et al. (2019) adopted this strategy to tackle very large-scale instances of both the Partial Set Covering Problem and the MCLP. The work has been extended in Coniglio et al. (2022) to the optimization of concave utility functions composed with a set-union operator and several links to submodular cuts are shown and discussed. Similar approaches appeared concerning other facility location problems: Lamontagne et al. (2024) tackled the Dynamic MCLP through an accelerated Benders decomposition approach, including local branching strategies; Duran-Mateluna et al. (2023) studied the p-median problem; Ljubić et al. (2024) solved the Discrete Ordered Median Problem as a parametric class of location models able to address covering as well as equity and fairness measures; Gaar and Sinnl (2022) proposed a projection-based branch-and-cut algorithm for the p-center problem. To the best of our knowledge, no similar works have appeared in relation to backup covering problems. Actually, we have not even been able to find in the literature any tailored exact approach for the backup covering problems addressed in this paper.

3. Formulations, properties, and assumptions

In this section, we formally define the considered backup covering problems (BACOP1 and BACOP2) and present their standard Integer Linear Programming (ILP) formulations, together with some useful properties and assumptions. Then, we propose a unified Benders reformulation for both problems. Throughout the paper, we will use the following notation:

- $\mathcal{I}$: set of locations where it is possible to install a facility;
- $\mathcal{J}$: set of demand centers;
- $h_j > 0$: demand associated with demand center $j \in \mathcal{J}$;
- p: predefined number of facilities to install (with $p \leq |\mathcal{I}|$);
- $\mathcal{I}(j)$: set of locations (*covering set*) in which installing a facility can cover the demand of center $j \in \mathcal{J}$;
- y_i: binary variable taking value 1 if a facility is installed in location $i \in \mathcal{I}$, and 0 otherwise;
- $u_j^{(1)}, u_j^{(2)}$: binary variable taking value 1 if the demand of center $j \in \mathcal{J}$ is covered at least once and at least twice, respectively, and 0 otherwise.

Finally, for the sake of conciseness, we will use the notation $v(\mathcal{S})$ to represent the summation over the generic set $\mathcal{S}$ of the elements belonging to the generic vector v, i.e., $v(\mathcal{S}) = \sum_{s \in \mathcal{S}} v_s$.

3.1. BACOP1

The BACOP1 aims at locating a specific number p of facilities so as to maximize the total demand covered twice while ensuring that the demand of all the centers is covered at least once. W.l.o.g., we assume:

ASSUMPTION 1. *For the BACOP1, we assume that* $|\mathcal{I}(j)| \geq 2$ *for each* $j \in \mathcal{J}$.

In fact, if there exists a demand center $\bar{j} \in \mathcal{J}$ for which $|\mathcal{I}(\bar{j})| = 0$, then the BACOP1 is infeasible since it is not possible to cover the entire demand at least once. Otherwise, if $|\mathcal{I}(j)| \geq 1, \forall j \in \mathcal{J}$, two cases exist: when $\left|\bigcup_{j\in\mathcal{J}:|\mathcal{I}(j)|=1} \mathcal{I}(j)\right| > p$, then the BACOP1 is infeasible because the available number of facilities is not enough to cover all the demand at least once; instead, when $\left|\bigcup_{j\in\mathcal{J}:|\mathcal{I}(j)|=1} \mathcal{I}(j)\right| \leq p$, then the optimization can be simplified since a facility must be installed in each location $i \in \bigcup_{j\in\mathcal{J}:|\mathcal{I}(j)|=1} \mathcal{I}(j)$ to ensure the coverage. So, any BACOP1 instance can be easily pre-processed to comply with Assumption 1.

A classical ILP formulation for the BACOP1 (Hogan and ReVelle 1986) is as follows:

$$\max_{\boldsymbol{y}\in\{0,1\}^{|\mathcal{I}|},\ \boldsymbol{u}^{(2)}\in\{0,1\}^{|\mathcal{J}|}} \quad \sum_{j\in\mathcal{J}} h_j u_j^{(2)} \tag{1a}$$

$$\text{s.t.} \quad y(\mathcal{I}) = p \tag{1b}$$

$$u_j^{(2)} + 1 \leq y(\mathcal{I}(j)) \qquad \forall j \in \mathcal{J}. \tag{1c}$$

The objective function (1a) aims at maximizing the total demand covered twice. Constraint (1b) ensures to install exactly p facilities while constraints (1c) ensure that the demand of center j contributes to the objective function (i.e., $u_j^{(2)} = 1$) only when at least two facilities are installed within the corresponding covering set $\mathcal{I}(j)$. Since constraints (1c) cannot be satisfied when their right-hand sides are strictly less than 1, then the model returns feasible solutions only if each demand center is covered at least once. All the variables are required to be binary, however, this requirement can be relaxed according to the following property.

PROPERTY 1. *The optimal solution of model* (1) *does not change if* $u^{(2)}$ *variables are relaxed to be continuous in* $[0,1]$*, i.e., if requirement* $\boldsymbol{u}^{(2)} \in \{0,1\}^{|\mathcal{J}|}$ *is substituted by* $\boldsymbol{u}^{(2)} \in [0,1]^{|\mathcal{J}|}$.

Proof. Let us consider the relaxed model proposed in the statement and any vector $\bar{\boldsymbol{y}}$ of values satisfying $\bar{y}(\mathcal{I}) = p, \bar{\boldsymbol{y}} \in \{0,1\}^{|\mathcal{I}|}$, and $\bar{y}(\mathcal{I}(j)) \geq 1, \forall j \in \mathcal{J}$. The problem can now be solved independently for each $j \in \mathcal{J}$. For a given $j \in \mathcal{J}$, variables $u_j^{(2)}$ are constrained by the system $\left\{0 \leq u_j^{(2)} \leq 1,\ u_j^{(2)} \leq UB_j\right\}$, where $UB_j \geq 0$ is an integer value. When $UB_j = 0$, the only possible value for $u_j^{(2)}$ is zero. Instead, when $UB_j \geq 1$, the objective function pushes $u_j^{(2)}$ (which has a strictly positive coefficient h_j by assumption) to take the maximum value in $[0,1]$. The optimal solution of the relaxed model is therefore integer as well. □

3.2. BACOP2

The BACOP2 aims at locating a specific number p of facilities so as to maximize a convex combination of the total demand covered once and twice. Let λ^1 and λ^2 be the weight in such a convex combination, of the single and the double coverage, respectively, thus ensuring $\lambda^1 > 0$, $\lambda^2 > 0$, and $\lambda^1 + \lambda^2 = 1$. Note that the convex combination is strict since it is not meaningful to entirely eliminate one of the two coverage contributions. W.l.o.g., we can make the following assumption:

ASSUMPTION 2. *For the BACOP2, we assume that* $|\mathcal{I}(j)| \geq 1$ *for each* $j \in \mathcal{J}$.

In fact, a demand center $\bar{j} \in \mathcal{J}$ for which $|\mathcal{I}(\bar{j})| = 0$ cannot be covered by any location plan and therefore it can be removed from $\mathcal{J}$. So, any BACOP2 instance can be easily pre-processed to comply with Assumption 2.

A classical ILP formulation for the BACOP2 (Hogan and ReVelle 1986) is as follows:

$$\max_{\boldsymbol{y}\in\{0,1\}^{|\mathcal{I}|},\ \boldsymbol{u^{(1)}},\boldsymbol{u^{(2)}}\in\{0,1\}^{|\mathcal{J}|}} \quad \lambda^1 \sum_{j\in\mathcal{J}} h_j u_j^{(1)} + \lambda^2 \sum_{j\in\mathcal{J}} h_j u_j^{(2)} \tag{2a}$$

$$\text{s.t.} \quad y(\mathcal{I}) = p \tag{2b}$$

$$u_j^{(2)} \leq u_j^{(1)} \qquad \forall j \in \mathcal{J} \tag{2c}$$

$$u_j^{(2)} + u_j^{(1)} \leq y(\mathcal{I}(j)) \qquad \forall j \in \mathcal{J}. \tag{2d}$$

The objective function (2a) aims at maximizing a convex combination of the total demand of the centers covered once and twice, weighted by λ^1 and λ^2, respectively. Constraint (2b) ensures to install exactly p facilities. Constraints (2c) enforce that a demand center is covered at least twice only if it is covered at least once. Constraints (2d) ensure that the number of times a demand center j is covered by facilities installed in $\mathcal{I}(j)$ is enough to obtain a single coverage ($u_j^{(1)} = 1$) or a double one ($u_j^{(2)} = 1$). All the variables are required to be binary, however, the following property holds.

PROPERTY 2. *When* $\lambda^1 > \lambda^2$*, the optimal solution of model* (2) *does not change if* $u^{(1)}$ *and* $u^{(2)}$ *variables are relaxed to be continuous in* $[0,1]$*, i.e., if requirements* $\boldsymbol{u^{(1)}} \in \{0,1\}^{|\mathcal{J}|}$ *and* $\boldsymbol{u^{(2)}} \in \{0,1\}^{|\mathcal{J}|}$ *are substituted by* $\boldsymbol{u^{(1)}} \in [0,1]^{|\mathcal{J}|}$ *and* $\boldsymbol{u^{(2)}} \in [0,1]^{|\mathcal{J}|}$*. When* $\lambda^1 \leq \lambda^2$*, the optimal solution may change.*

Proof. Let us consider the relaxed model proposed in the statement and any vector $\bar{\boldsymbol{y}}$ of values satisfying constraints $\bar{y}(\mathcal{I}) = p$ and $\bar{\boldsymbol{y}} \in \{0,1\}^{|\mathcal{I}|}$. The problem can now be solved independently for each $j \in \mathcal{J}$. For a given $j \in \mathcal{J}$, variables $u_j^{(1)}$ and $u_j^{(2)}$ are constrained by the system $\left\{0 \leq u_j^{(2)} \leq u_j^{(1)} \leq 1,\ u_j^{(2)} + u_j^{(1)} \leq UB_j\right\}$, where $UB_j \geq 0$ and integer. If $UB_j = 0$, then $u_j^{(1)} = u_j^{(2)} = 0$. If $UB_j \geq 2$, the objective function pushes both $u_j^{(1)}$ and $u_j^{(2)}$ (which have strictly positive coefficients $\lambda^1 h_j$ and $\lambda^2 h_j$ by assumption) to take their maximum value in $[0,1]$. Instead, if $UB_j = 1$, the optimal solution depends on the angular coefficient $m = -\frac{\lambda^1}{\lambda^2}$ of the objective function: when $m < -1$ (i.e., when $\lambda^1 > \lambda^2$), the optimal solution is attained at the integer point $P_j^1 : (u_j^{(1)}, u_j^{(2)}) = (1,0)$; when $m > -1$ (i.e., when $\lambda^1 < \lambda^2$), the optimal solution is attained at the fractional point $P_j^2 : (u_j^{(1)}, u_j^{(2)}) = (\frac{1}{2}, \frac{1}{2})$; when $m = -1$ (i.e., when $\lambda^1 = \lambda^2 = \frac{1}{2}$), we have infinitely many optimal solutions on the convex combination between the points P_j^1 and P_j^2 (P_j^1 being the only integer one). □

Hence, to use the relaxed model instead of the binary one, in the rest of the work, we will stick with:

ASSUMPTION 3. *For the BACOP2, we assume that* $\lambda^1 > \lambda^2$.

Nevertheless, this condition is totally realistic since, in practical applications, it always makes sense to give more value to the single coverage of a demand center with respect to the backup one.

3.3. A unified Benders reformulation

Exploiting Property 1 and 2, it is possible to apply a Benders decomposition to the relaxed models of BACOP1 and BACOP2. The Benders reformulation of both models is

$$\max_{\theta\in\mathbb{R},\boldsymbol{y}\in\{0,1\}^{|\mathcal{I}|}} \quad \theta \tag{3a}$$

$$\text{s.t.} \quad y(\mathcal{I}) = p \tag{3b}$$

$$B_r^{feas}(\boldsymbol{y}) \geq 0 \qquad \forall r \in \mathcal{R}(\mathcal{P}) \tag{3c}$$

$$B_v^{opt}(\boldsymbol{y}) \geq \theta \qquad \forall v \in \mathcal{V}(\mathcal{P}), \tag{3d}$$

where variable $\theta \geq 0$ represents the upper bound on the objective function of the corresponding problem, while $\mathcal{R}(\mathcal{P})$ and $\mathcal{V}(\mathcal{P})$ is the set of extreme rays and extreme points, respectively, of $\mathcal{P}$, i.e., the polyhedron representing the feasible region of the respective dual Benders subproblem. Constraints (3c) and (3d) are referred to as *feasibility cuts* and *optimality cuts*, respectively. The model ensures the cardinality constraint (3b) as well. It is important to note that our decomposition, and therefore the subsequent derivations of cuts and their properties, are not affected by the specific nature of the constraint (3b). Actually, any BACOP1 or BACOP2 extension that requires additional constraints depending solely on y-variables can be handled in the same way. When only a (possibly empty) subset of constraints (3c) and (3d) is included in model (3), we will refer to it as the Benders *relaxed master problem* (RMP). In the following, $\tilde{\boldsymbol{y}}$ will denote a solution vector of the RMP or of its LP relaxation. The specific subproblems for BACOP1 and BACOP2, together with their properties, are presented in the following section so as to derive a tailored approach for each problem.

4. A tailored approach for BACOP1

We present hereafter the subproblem of the Benders decomposition for BACOP1, proposing a closed formula to solve its dual. We then derive feasibility and optimality cuts and show dominance properties among them.

4.1. Benders subproblem

For any given vector $\tilde{\boldsymbol{y}} \in [0,1]^{|\mathcal{I}|}$, the primal Benders subproblem for BACOP1 is:

$$\max_{\boldsymbol{u}^{(2)}\in\mathbb{R}_+^{|\mathcal{J}|}} \quad \sum_{j\in\mathcal{J}} h_j u_j^{(2)} \tag{4a}$$

$$\text{s.t.} \quad u_j^{(2)} \leq \tilde{y}(\mathcal{I}(j)) - 1 \qquad \forall j \in \mathcal{J} \tag{4b}$$

$$u_j^{(2)} \leq 1 \qquad \forall j \in \mathcal{J}. \tag{4c}$$

Let π_j and σ_j be the dual variables associated with each of the constraints (4b) and (4c), respectively. Then, the dual of the Benders subproblem (4a)–(4c) is:

$$\min_{\boldsymbol{\pi},\boldsymbol{\sigma}\in\mathbb{R}_+^{|\mathcal{J}|}} \quad \sum_{j\in\mathcal{J}} \left[\left(\tilde{y}(\mathcal{I}(j)) - 1\right)\pi_j + \sigma_j \right] \tag{5a}$$

$$\text{s.t.} \quad \pi_j + \sigma_j \geq h_j \qquad \forall j \in \mathcal{J}. \tag{5b}$$

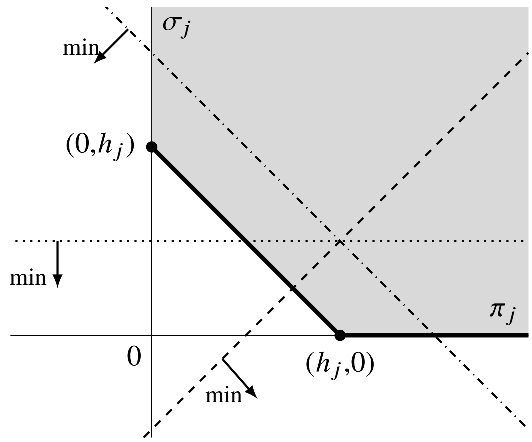


Figure 1 **The feasible region is shaded in grey. The objective function, together with the minimization direction, is drawn as a dashed line for $\tilde{y}(\mathcal{I}(j)) = 0$, as a dotted line for $\tilde{y}(\mathcal{I}(j)) = 1$, and as a dash-dotted line for $\tilde{y}(\mathcal{I}(j)) = 2$. All the (finite) optimal solutions, when varying the value of $\tilde{y}(\mathcal{I}(j)) - 1$, are the points in bold. Among them, exactly 2 vertices appear.**

4.2. Linear-time separation of Benders cuts

Despite being a linear program, and thus easy-to-solve, the dual subproblem (5) can be better approached by using the result stated in the following proposition.

PROPOSITION 1. *If there exists some $\bar{j} \in \mathcal{J}$ such that $-1 \le \tilde{y}(\mathcal{I}(\bar{j})) - 1 < 0$, then problem* (5) *is unbounded along the direction of the extreme rays*

$$(\pi_j^{\rightarrow}, \sigma_j^{\rightarrow}) = \begin{cases} (1,0) & \textit{if } j = \bar{j} \\ (0,0) & \textit{otherwise} \end{cases} \quad j \in \mathcal{J}. \tag{6}$$

Otherwise, the full set of its optimal finite solutions $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*)$ is represented by

$$(\pi_j^*, \sigma_j^*) = \begin{cases} (\epsilon_j^1, 0), \textit{with } \epsilon_j^1 \ge h_j & \textit{if} \quad \tilde{y}(\mathcal{I}(j)) - 1 = 0 \\ (h_j, 0) & \textit{if } 0 < \tilde{y}(\mathcal{I}(j)) - 1 < 1 \\ (\epsilon_j^2, h_j - \epsilon_j^2), \textit{with } 0 \le \epsilon_j^2 \le h_j & \textit{if} \quad \tilde{y}(\mathcal{I}(j)) - 1 = 1 \\ (0, h_j) & \textit{if} \quad \tilde{y}(\mathcal{I}(j)) - 1 > 1 \end{cases} \quad j \in \mathcal{J}. \tag{7}$$

Proof. Problem (5) can be solved independently for each $j \in \mathcal{J}$. For any given j, the feasible region of (5) never changes (see Figure 1), while its objective function has a single parameter $\tilde{y}(\mathcal{I}(j)) - 1$, i.e., the coefficient of π_j, that can vary in $[-1, |\mathcal{I}(j)| - 1]$. As such a coefficient grows with respect to its minimum value, the objective function rotates clockwise. Hence, the problem may be unbounded, may admit a unique optimal solution, or infinitely many, according to the cases presented in the statement. □

Thanks to Proposition 1, we can separate a feasibility or optimality cut for BACOP1 in linear time, which is theoretically and practically faster than using any LP-based approach.

4.3. Benders cuts

Given any extreme ray with direction $(\pi_j^{\rightarrow}, \sigma_j^{\rightarrow})$ of the dual subproblem (5), it is possible to derive feasibility cuts of the form $\pi_j^{\rightarrow}\,(y(\mathcal{I}(j)) - 1) + \sigma_j^{\rightarrow} \ge 0, \quad \forall j \in \mathcal{J}$. Hence, due to the result shown in (6) of Proposition 1, for any vector $\tilde{\boldsymbol{y}}$, we obtain the Benders cuts

$$y(\mathcal{I}(j)) \geq 1, \quad \forall j \in \mathcal{J} : -1 \leq \tilde{y}(\mathcal{I}(j)) - 1 < 0, \tag{B$_1$-f$_1$}$$

which states that a demand center has to be covered at least once. Actually, to guarantee the BACOP1 feasibility, this requirement holds for any $j \in \mathcal{J}$, not just for those satisfying the stated condition on $\tilde{\boldsymbol{y}}$. This opens the possibility, instead of separating (B$_1$-f$_1$) dynamically upon verifying the condition, to add the entire set of constraints $y(\mathcal{I}(j)) \geq 1, \forall j \in \mathcal{J}$, in the RMP since the beginning. Both alternatives will be evaluated later.

Instead, given any finite optimal solution $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*)$ of the dual subproblem (5), the corresponding optimality cut takes the form $\sum_{j \in \mathcal{J}} \left[\pi_j^* \left(y(\mathcal{I}(j)) - 1\right) + \sigma_j^*\right] \geq \theta$. To achieve optimality, it is sufficient to consider those $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*)$ corresponding to vertices of the feasible region of the dual subproblem, which are two for each $j \in \mathcal{J}$ according to Proposition 1. Such vertices are obtainable by fixing $\epsilon_j^1 = h_j$, while there exists the freedom to set ϵ_j^2 to h_j or 0 for any $j \in \mathcal{J}$ such that $\tilde{y}(\mathcal{I}(j)) = 2$. Therefore, when generating the cuts, we explore all the different possibilities in the following. The solution $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*)$ where, for any $j \in \mathcal{J}$, $(\pi_j^*, \sigma_j^*) = \begin{cases} (h_j, 0) & \text{if } \tilde{y}(\mathcal{I}(j)) < 2 \\ (0, h_j) & \text{if } \tilde{y}(\mathcal{I}(j)) \geq 2 \end{cases}$ is a special case of (7) obtainable by setting $\epsilon_j^2 = 0, \forall j \in \mathcal{J}$. So, for any $\tilde{\boldsymbol{y}}$, we obtain the Benders cut

$$\sum_{j \in \mathcal{J} : \tilde{y}(\mathcal{I}(j)) < 2} h_j \left(y(\mathcal{I}(j)) - 1\right) + \sum_{j \in \mathcal{J} : \tilde{y}(\mathcal{I}(j)) \geq 2} h_j \geq \theta. \tag{B$_1$-o$_1$}$$

The solution $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*)$ where, for any $j \in \mathcal{J}$, $(\pi_j^*, \sigma_j^*) = \begin{cases} (h_j, 0) & \text{if } \tilde{y}(\mathcal{I}(j)) \leq 2 \\ (0, h_j) & \text{if } \tilde{y}(\mathcal{I}(j)) > 2 \end{cases}$ is a special case of (7) obtainable by setting $\epsilon_j^2 = h_j, \forall j \in \mathcal{J}$. So, for any $\tilde{\boldsymbol{y}}$, we obtain the Benders cut

$$\sum_{j \in \mathcal{J} : \tilde{y}(\mathcal{I}(j)) \leq 2} h_j \left(y(\mathcal{I}(j)) - 1\right) + \sum_{j \in \mathcal{J} : \tilde{y}(\mathcal{I}(j)) > 2} h_j \geq \theta. \tag{B$_1$-o$_2$}$$

Finally, exploiting the specific cardinality of covering sets, the solution $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*)$ where, for any $j \in \mathcal{J}$, $(\pi_j^*, \sigma_j^*) = \begin{cases} (h_j, 0) & \text{if } \tilde{y}(\mathcal{I}(j)) < 2 \ \vee \ |\mathcal{I}(j)| = 2 \\ (0, h_j) & \text{if } \tilde{y}(\mathcal{I}(j)) \geq 2 \ \wedge \ |\mathcal{I}(j)| > 2 \end{cases}$ is a special case of (7) obtainable by setting $\epsilon_j^2 = \begin{cases} h_j, & \text{if } |\mathcal{I}(j)| = 2 \\ 0, & \text{if } |\mathcal{I}(j)| > 2 \end{cases}$, $\forall j \in \mathcal{J}$. So, for any $\tilde{\boldsymbol{y}}$, we obtain the Benders cut

$$\sum_{j \in \mathcal{J} : \tilde{y}(\mathcal{I}(j)) < 2} h_j \left(y(\mathcal{I}(j)) - 1\right) + \sum_{\substack{j \in \mathcal{J} : \tilde{y}(\mathcal{I}(j)) = 2 \\ \wedge \ |\mathcal{I}(j)| = 2}} h_j \left(y(\mathcal{I}(j)) - 1\right) + \sum_{\substack{j \in \mathcal{J} : \tilde{y}(\mathcal{I}(j)) \geq 2 \\ \wedge \ |\mathcal{I}(j)| > 2}} h_j \geq \theta. \tag{B$_1$-o$_3$}$$

A more intuitive interpretation of the derived Benders cuts can be obtained if we consider a *binary* vector $\tilde{\boldsymbol{y}}$. For example, (B$_1$-o$_1$) can be rewritten as $\sum_{j \notin \mathcal{J}^{(2)}(\tilde{\mathcal{I}})} h_j \left(y(\mathcal{I}(j)) - 1\right) \geq \theta - \sum_{j \in \mathcal{J}^{(2)}(\tilde{\mathcal{I}})} h_j$, where $\tilde{\mathcal{I}} = \{i \in \mathcal{I} : \tilde{y}_i = 1\}$ denotes the subset of locations in which a facility is installed according to $\tilde{\boldsymbol{y}}$ and $\mathcal{J}^{(k)}(\tilde{\mathcal{I}}) \subseteq \mathcal{J}$ denotes the subset of demand centers that are covered at least k times by the facilities installed in $\tilde{\mathcal{I}}$. The inequality states that the potential increase of the demand covered twice by $\tilde{\boldsymbol{y}}$ (the rhs) is upper bounded by all the demand corresponding to demand centers that can still be covered at least twice by opening some additional facilities outside of $\tilde{\mathcal{I}}$. Similar interpretations can be obtained for (B$_1$-o$_2$) and (B$_1$-o$_3$) as well.

4.4. Dominance properties of the derived cuts

By closely analyzing the three proposed optimality cuts, we can identify several properties related to their relative strength. These results are gathered in the following theorem, in which we say that a cut derived from a point $\boldsymbol{y} \in \{0,1\}^{|\mathcal{I}|}$ *dominates* another cut derived from the same point, if the former provides an upper bound on θ at least as strict as the latter one.

THEOREM 3. *Given any solution $\tilde{\boldsymbol{y}}$ of the LP relaxation of the RMP, the dominance relationship between each pair of the optimality cuts derived from $\tilde{\boldsymbol{y}}$ for BACOP1 is shown in the following scheme:*

$$\begin{tikzcd}
 & \boxed{(B_1\text{-}o_3)} \arrow[dl] \arrow[dr, dash, dashed] & \\
\boxed{(B_1\text{-}o_1)} \arrow[rr, dash, dashed] & & \boxed{(B_1\text{-}o_2)}
\end{tikzcd}$$

$A \longrightarrow B$: A dominates B

$A - - - B$: neither A nor B dominates the other

The dominance relationships are proved in Propositions **??**–**??** (see Appendix **??**). These results state that (B_1-o_1) is redundant when implementing a cutting-plane method in which (B_1-o_3) is also separated.

5. A tailored approach for BACOP2

Hereafter, we propose a similar approach for the Benders decomposition of BACOP2. Interestingly enough, the peculiarities of this problem lead to quite different dominance results, which are worth a detailed description.

5.1. Benders subproblem

For any given vector $\tilde{\boldsymbol{y}} \in [0,1]^{|\mathcal{I}|}$, the primal Benders subproblem for BACOP2 is:

$$\max_{\boldsymbol{u}^{(1)},\boldsymbol{u}^{(2)} \in \mathbb{R}_+^{|\mathcal{J}|}} \quad \lambda^2 \sum_{j\in\mathcal{J}} h_j u_j^{(2)} + \lambda^1 \sum_{j\in\mathcal{J}} h_j u_j^{(1)} \tag{8a}$$

$$\text{s.t.} \quad u_j^{(2)} - u_j^{(1)} \le 0 \qquad \forall j \in \mathcal{J} \tag{8b}$$

$$u_j^{(2)} + u_j^{(1)} \le \tilde{y}(\mathcal{I}(j)) \qquad \forall j \in \mathcal{J} \tag{8c}$$

$$u_j^{(1)} \le 1 \qquad \forall j \in \mathcal{J}. \tag{8d}$$

Note that the upper restriction $u_j^{(2)} \le 1$ is not necessary since it is implied by constraints (8b) and (8d). Now, by defining μ_j, π_j, and σ_j as the dual variables associated with each of the constraints (8b), (8c), and (8d), respectively, the dual of the Benders subproblem becomes:

$$\min_{\boldsymbol{\mu},\boldsymbol{\pi},\boldsymbol{\sigma} \in \mathbb{R}_+^{|\mathcal{J}|}} \quad \sum_{j\in\mathcal{J}} \left[\tilde{y}(\mathcal{I}(j))\pi_j + \sigma_j\right] \tag{9a}$$

$$\text{s.t.} \quad \mu_j + \pi_j \ge \lambda^2 h_j \qquad \forall j \in \mathcal{J} \tag{9b}$$

$$-\mu_j + \pi_j + \sigma_j \ge \lambda^1 h_j \qquad \forall j \in \mathcal{J}. \tag{9c}$$

Since μ_j variables do not contribute to the objective function of the above dual subproblem, we can simplify the analysis by optimizing the same function over the projection of the feasible region of (9) on the $(\boldsymbol{\pi},\boldsymbol{\sigma})$-space. By applying the *Fourier-Motzkin* elimination for μ_j variables, the feasible region of (9) is equivalent to

$$\left\{\mu_j \ge -\pi_j + \lambda^2 h_j;\ \pi_j + \sigma_j - \lambda^1 h_j \ge \mu_j;\ \mu_j, \pi_j, \sigma_j \ge 0;\ \forall j \in \mathcal{J}\right\} \equiv$$

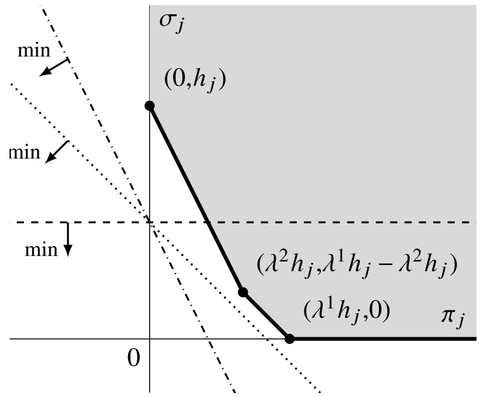


Figure 2 **The feasible region is shaded in grey. The objective function, together with the minimization direction, is drawn as a dashed line for $\tilde{y}(\mathcal{I}(j)) = 0$, as a dotted line for $\tilde{y}(\mathcal{I}(j)) = 1$, and as a dash-dotted line for $\tilde{y}(\mathcal{I}(j)) = 2$. All the (finite) optimal solutions, when varying the value of $\tilde{y}(\mathcal{I}(j)) > 0$, are the points in bold. Among them, exactly 3 vertices appear.**

$$\equiv \left\{\pi_j + \sigma_j - \lambda^1 h_j \geq \mu_j \geq -\pi_j + \lambda^2 h_j;\ \pi_j + \sigma_j - \lambda^1 h_j \geq \mu_j \geq 0;\ \pi_j, \sigma_j \geq 0;\ \forall j \in \mathcal{J}\right\},$$

which has the same solutions in the $(\boldsymbol{\pi}, \boldsymbol{\sigma})$-variables space of the system

$$\left\{\pi_j + \sigma_j - \lambda^1 h_j \geq -\pi_j + \lambda^2 h_j;\ \pi_j + \sigma_j - \lambda^1 h_j \geq 0;\ \pi_j, \sigma_j \geq 0;\ \forall j \in \mathcal{J}\right\}.$$

Remembering that $\lambda^1 + \lambda^2 = 1$, the resulting simplified dual subproblem is then

$$\min_{\boldsymbol{\pi},\boldsymbol{\sigma} \in \mathbb{R}_+^{|\mathcal{J}|}} \quad \sum_{j \in \mathcal{J}} \left[\tilde{y}(\mathcal{I}(j))\pi_j + \sigma_j\right] \tag{10a}$$

$$\text{s.t.} \quad 2\pi_j + \sigma_j \geq h_j \qquad \forall j \in \mathcal{J} \tag{10b}$$

$$\pi_j + \sigma_j \geq \lambda^1 h_j \qquad \forall j \in \mathcal{J}. \tag{10c}$$

5.2. Linear-time separation of Benders cuts

Despite being a linear program, and thus easy-to-solve, the dual subproblem (10) can be better approached by using the closed formula stated in the following proposition.

PROPOSITION 2. *Problem* (10) *always admits a finite optimal solution, and the full set of its optimal solutions* $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*)$ *is represented by*

$$(\pi_j^*, \sigma_j^*) = \begin{cases} (\epsilon_j^1 + \lambda^1 h_j, 0), \textit{with } \epsilon_j^1 \geq 0 & \textit{if} \quad\ \ \tilde{y}(\mathcal{I}(j)) = 0 \\ (\lambda^1 h_j, 0) & \textit{if} \ \ 0 < \tilde{y}(\mathcal{I}(j)) < 1 \\ (\epsilon_j^2, \lambda^1 h_j - \epsilon_j^2), \textit{with } \lambda^2 h_j \leq \epsilon_j^2 \leq \lambda^1 h_j & \textit{if} \quad\ \ \tilde{y}(\mathcal{I}(j)) = 1 \\ (\lambda^2 h_j, \lambda^1 h_j - \lambda^2 h_j) & \textit{if} \ \ 1 < \tilde{y}(\mathcal{I}(j)) < 2 \\ (\epsilon_j^3, h_j - 2\epsilon_j^3), \textit{with } 0 \leq \epsilon_j^3 \leq \lambda^2 h_j & \textit{if} \quad\ \ \tilde{y}(\mathcal{I}(j)) = 2 \\ (0, h_j) & \textit{if} \quad\ \ \tilde{y}(\mathcal{I}(j)) > 2 \end{cases} \quad j \in \mathcal{J}. \tag{11}$$

Proof. Problem (10) can be solved independently for each $j \in \mathcal{J}$. Given any specific j, the feasible region of (10) never changes (see Figure 2), while its objective function has a single parameter $\tilde{y}(\mathcal{I}(j))$, i.e., the coefficient of π_j, that can vary in $[0, |\mathcal{I}(j)|]$. As such a coefficient grows with respect to its minimum value, the objective function rotates clockwise. Hence, the problem yields a unique optimal solution or infinitely many according to the cases presented in the statement. □

Thanks to Proposition 2, we do not need to separate any feasibility cut for BACOP2 (since the corresponding dual subproblem always admits a finite optimal solution), while optimality cuts can be separated in linear time.

5.3. Benders cuts

Given any optimal solution $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*)$ of (10), the corresponding optimality cut takes the form $\sum_{j\in\mathcal{J}} \left[\pi^*_j y(\mathcal{I}(j)) + \sigma^*_j\right] \geq \theta$. Again, we can consider only the vertices of the feasible region of (10), which are three for each $j \in \mathcal{J}$ according to Proposition 2. Such vertices are obtained by fixing $\epsilon^1_j = 0$, while there exists the freedom to set ϵ^2_j to either $\lambda^1 h_j$ or $\lambda^2 h_j$ for any $j \in \mathcal{J}$ such that $\tilde{y}(\mathcal{I}(j)) = 1$ and to set ϵ^3_j to either $\lambda^2 h_j$ or 0 for any $j \in \mathcal{J}$ such that $\tilde{y}(\mathcal{I}(j)) = 2$. Therefore, by considering for each $j \in \mathcal{J}$ all the four above-mentioned $(\epsilon^2_j, \epsilon^3_j)$ combinations we can derive the following optimality cuts:

$$\sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))<1} \lambda^1 h_j y(\mathcal{I}(j)) + \sum_{j\in\mathcal{J}:1\leq\tilde{y}(\mathcal{I}(j))<2} h_j \left[(y(\mathcal{I}(j)) - 1)\,\lambda^2 + \lambda^1\right] + \sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))\geq 2} h_j \geq \theta \tag{B$_2$-o$_1$}$$

$$\sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))<1} \lambda^1 h_j y(\mathcal{I}(j)) + \sum_{j\in\mathcal{J}:1\leq\tilde{y}(\mathcal{I}(j))\leq 2} h_j \left[(y(\mathcal{I}(j)) - 1)\,\lambda^2 + \lambda^1\right] + \sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))> 2} h_j \geq \theta \tag{B$_2$-o$_2$}$$

$$\sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))\leq 1} \lambda^1 h_j y(\mathcal{I}(j)) + \sum_{j\in\mathcal{J}:1<\tilde{y}(\mathcal{I}(j))<2} h_j \left[(y(\mathcal{I}(j)) - 1)\,\lambda^2 + \lambda^1\right] + \sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))\geq 2} h_j \geq \theta \tag{B$_2$-o$_3$}$$

$$\sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))\leq 1} \lambda^1 h_j y(\mathcal{I}(j)) + \sum_{j\in\mathcal{J}:1<\tilde{y}(\mathcal{I}(j))\leq 2} h_j \left[(y(\mathcal{I}(j)) - 1)\,\lambda^2 + \lambda^1\right] + \sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))> 2} h_j \geq \theta. \tag{B$_2$-o$_4$}$$

Moreover, considering $(\epsilon^2_j, \epsilon^3_j)$ equal to $\begin{cases}(\lambda^2 h_j, 0) & \text{if } |\mathcal{I}(j)| > 1\\ (\lambda^1 h_j, 0) & \text{if } |\mathcal{I}(j)| = 1\end{cases}$ or $\begin{cases}(\lambda^2 h_j, \lambda^2 h_j) & \text{if } |\mathcal{I}(j)| > 1\\ (\lambda^1 h_j, \lambda^2 h_j) & \text{if } |\mathcal{I}(j)| = 1\end{cases}$, for each $j \in \mathcal{J}$, we can also derive, respectively, the two following optimality cuts:

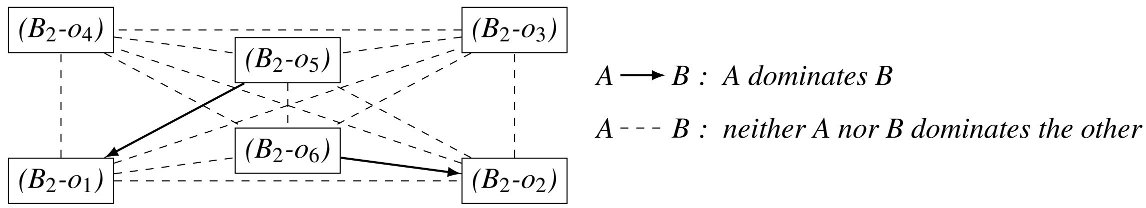
$$\sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))<1} \lambda^1 h_j y(\mathcal{I}(j)) + \sum_{\substack{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))=1\\ \wedge\ |\mathcal{I}(j)|=1}} \lambda^1 h_j y(\mathcal{I}(j)) + \sum_{\substack{j\in\mathcal{J}:1\leq\tilde{y}(\mathcal{I}(j))<2\\ \wedge\ |\mathcal{I}(j)|>1}} h_j \left[(y(\mathcal{I}(j)) - 1)\,\lambda^2 + \lambda^1\right] + \sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))\geq 2} h_j \geq \theta \tag{B$_2$-o$_5$}$$

$$\sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))<1} \lambda^1 h_j y(\mathcal{I}(j)) + \sum_{\substack{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))=1\\ \wedge\ |\mathcal{I}(j)|=1}} \lambda^1 h_j y(\mathcal{I}(j)) + \sum_{\substack{j\in\mathcal{J}:1\leq\tilde{y}(\mathcal{I}(j))\leq 2\\ \wedge\ |\mathcal{I}(j)|>1}} h_j \left[(y(\mathcal{I}(j)) - 1)\,\lambda^2 + \lambda^1\right] + \sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))> 2} h_j \geq \theta. \tag{B$_2$-o$_6$}$$

5.4. Dominance properties of the derived cuts

By closely analyzing the six different optimality cuts derived, we can obtain several properties concerning their relative strength. The results are summarized in the following theorem, where the underlying concept of *dominance* is the same as described for BACOP1 in Section 4.4.

THEOREM 4. *Given any solution* $\tilde{\mathbf{y}}$ *of the LP relaxation of the RMP, the dominance relationship between each pair of the optimality cuts derived from* $\tilde{\mathbf{y}}$ *for BACOP2 is shown in the following scheme:*



These dominance relationships are proved in Propositions ??–?? (see Appendix ??). These results state that (B_2-o_1) and (B_2-o_2) are redundant when implementing a cutting-plane method in which (B_2-o_5) and (B_2-o_6), respectively, are also separated.

6. General framework and implementation details

Hereafter, we describe the general algorithmic framework implemented, together with some strategies helping our approach to be computationally competitive in practice.

Branch-and-Benders-Cut approach. In our method, we embedded the separation of the obtained cuts within a branch-and-bound framework, thus creating a so-called *Branch-and-Benders-Cut* (BBC) approach. This practice, which has become popular in recent literature (Ljubić et al. 2012, Fischetti et al. 2016, 2017), brings at least two advantages: a) the RMP is not solved at each iteration as an integer program; b) more flexibility is given to the algorithm designer concerning when to perform the separation and how many cuts to add (various BBC versions will be compared in Section 7).

Matrix-based cuts reformulation. Given the large number of cuts to generate during the algorithm and their mathematical complexity, the access to the data structures and the execution of their inter-operations behind such a generation become computationally critical. In the following, we detail the cuts reformulation adopted in our implementation to exploit matrix calculus, which is particularly tailored for very sparse data (as the covering information when the number of demand centers is much larger than that of the locations). Let us define the *covering matrix* $C \in \{0,1\}^{|\mathcal{J}|\times|\mathcal{I}|}$, in which each component c_{ji} takes value 1 if demand center $j \in \mathcal{J}$ can be covered by a facility placed in location $i \in \mathcal{I}$, and zero otherwise (this matrix can be pre-computed by starting from the covering sets, even if in many practical settings it is the actual input), the vector $\boldsymbol{h} \in \mathbb{R}^{|\mathcal{J}|}$ containing the demand h_j in the j-th position, and the vector $\boldsymbol{n} = C\mathbf{1}$ (where $\mathbf{1}$ is a vector of length $|\mathcal{I}|$ composed by all ones), so that the value of n_j corresponds to the cardinality of $\mathcal{I}(j)$ for each $j \in \mathcal{J}$. Hence, given any solution vector $\tilde{\boldsymbol{y}}$ of the RMP, we have $\tilde{y}(\mathcal{I}(j)) = \sum_{i\in\mathcal{I}(j)} \tilde{y}_i = [C\tilde{\boldsymbol{y}}]_j, \forall j \in \mathcal{J}$. Moreover, given a generic vector $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{J}|}$, the summation of its components over all the demand centers (possibly characterized by a condition ψ on their coverage in the solution considered) can be rewritten as $\sum_{j\in\mathcal{J}:\,\psi} v_j = \boldsymbol{e}_\psi^T \boldsymbol{v}$, where $\boldsymbol{e}_\psi \in \{0,1\}^{|\mathcal{J}|}$ is a vector having the j-th component equal to 1 if the condition ψ holds, and 0 otherwise. With the equivalences introduced, all the cuts derived can be reformulated in a matrix-based form. As an example, we show hereafter the reformulation of cut (B_1-o_3) for BACOP1, which is non-dominated. First, we rearrange its terms as follows

$$\sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))<2} h_j y(\mathcal{I}(j)) - \sum_{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))<2} h_j + \sum_{\substack{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))=2\\ \wedge\ |\mathcal{I}(j)|=2}} h_j y(\mathcal{I}(j)) - \sum_{\substack{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))=2\\ \wedge\ |\mathcal{I}(j)|=2}} h_j + \sum_{\substack{j\in\mathcal{J}:\tilde{y}(\mathcal{I}(j))\geq 2\\ \wedge\ |\mathcal{I}(j)|>2}} h_j \geq \theta.$$

Then, using the equivalences introduced above, the cut can be written as

$$\boldsymbol{e}^T_{[C\bar{\boldsymbol{y}}]_j<2}(\boldsymbol{h}\odot C\boldsymbol{y}) - \boldsymbol{e}^T_{[C\bar{\boldsymbol{y}}]_j<2}\boldsymbol{h} + \boldsymbol{e}^T_{\substack{[C\bar{\boldsymbol{y}}]_j=2\\ \wedge\, n_j=2}}(\boldsymbol{h}\odot C\boldsymbol{y}) - \boldsymbol{e}^T_{\substack{[C\bar{\boldsymbol{y}}]_j=2\\ \wedge\, n_j=2}}\boldsymbol{h} + \boldsymbol{e}^T_{\substack{[C\bar{\boldsymbol{y}}]_j\geq 2\\ \wedge\, n_j>2}}\boldsymbol{h} \geq \theta,$$

where $\odot$ represents the standard *component-wise* product between two vectors of the same length.

Stabilization. It is known that the solutions coming from the RMP, which affect the subproblem and the cuts derived, tend to wildly oscillate from good points (i.e., close to optimality) to much worse ones, and this instability is reflected in a possibly slower convergence of the method. Hence, we implemented a computationally cheap stabilization routine based on the *in-out* paradigm (Ben-Ameur and Neto 2007, Fischetti et al. 2017) and on a binary search of the parameters involved, which have shown good performances according to Ljubić et al. (2024). More precisely, at any iteration returning the RMP optimal point $\boldsymbol{y}^*$, we compute a *core point* $\boldsymbol{y}^C$, i.e., a solution inside the current RMP relaxation, and separate the Benders cuts with respect to a new point $\bar{\boldsymbol{y}}$ that is a convex combination of the two above solutions, i.e., $\bar{\boldsymbol{y}} := \alpha\boldsymbol{y}^* + (1-\alpha)\boldsymbol{y}^C$, with $0 \leq \alpha \leq 1$. Since it is not possible to know in advance which value of α would return the better cuts, we apply a dichotomic search that generates S different points by starting from $\alpha = 1$ and halving its value at each iteration (since the initial value of α is 1, the very first point is always $\boldsymbol{y}^*$). Note that, if the cut generated from a point $\bar{\boldsymbol{y}}$ is not violated, i.e., if the value of θ isn't improved by a sufficient amount, then the procedure stops, since points closer to $\boldsymbol{y}^C$ would generate ineffective cuts as well. After some preliminary tests, we found a good compromise between computational speed and solution quality in setting $S = 5$ and in applying the stabilization method only at the root node of the branch-and-cut tree. Finally, for the BACOP1, the core point $\boldsymbol{y}^C$ is chosen as the analytical center of the RMP feasible region (easily obtainable by applying a barrier method to the RMP with no objective function), while, for the BACOP2, we can define a single core point by setting $y_i^C = \frac{p}{|\mathcal{I}|}, \forall i \in \mathcal{I}$ (which is even cheaper to compute).

7. Numerical experiments

In this section, we present the numerical experiments conducted to evaluate the performance and practical impact of our BBC approach.

All the experiments have been performed on a 2.30 GHz 6-core Intel Xeon Gold computer with 32 GB of RAM and running Linux v22.04.4. The approach has been implemented in Python using the branch-and-cut framework of CPLEX 22.1.0.0 through its Concert Technology callbacks. The code is available on GitHub (Fadda et al. 2026). CPLEX was executed in single-thread mode with all parameters set to their default values, except for `Preprocessing_Linear` $= 0$ to ensure that our cutting-planes generation is not disrupted by non-linear reformulations at the preprocessing stage (IBM ILOG CPLEX v22.1.0 Documentation 2022). Finally, to avoid tailing-off and numerical approximation issues, the optimality cuts generated should be added only when they are violated by only a sensible amount (Bonami et al. 2020). Hence, in our method, we add a cut only when the ratio between its lhs and rhs exceeds 10^{-6}.

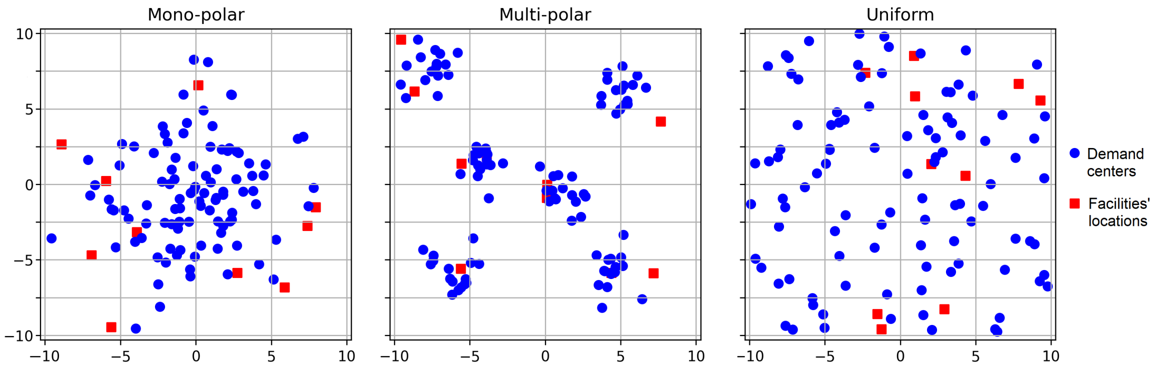


Figure 3 **Examples of the different topologies for 100 demand centers and 10 potential facilities' locations.**

7.1. Instance generation

For the experiments, we generated benchmark instances following Fadda et al. (2021b). $|\mathcal{I}|$ is fixed to 100, while $|\mathcal{J}|$ is several orders of magnitude higher than $|\mathcal{I}|$. More precisely, for each instance, the coordinates of the facilities' location and the demand centers are drawn in the $[-10, 10] \times [-10, 10]$ square and follow three different topologies, namely, *mono-polar*, *multi-polar*, and *uniform*, to simulate different polarization or dispersion aspects of the underlying service coverage. The mono-polar topology is obtained by sampling the 80% of the coordinates according to a multivariate *Student's t* distribution with 3 degrees of freedom (Li and Nadarajah 2020), and the remaining 20% from a Uniform distribution. The multi-polar topology is obtained by considering 6 multinomial distributions with the mean picked from a uniformly random realization and an identity covariance matrix. Finally, in the uniform topology, all the coordinates are sampled according to a pure uniform distribution. A graphical representation of the three topologies is reported in Figure 3. To define the covering sets, we assumed that a facility covers a center if the Euclidean distance between the two positions is smaller than a covering radius r equal to 10% of the square's diagonal. This choice ensures obtaining sensible solutions. Smaller values for r would lead the BACOP1 to be infeasible too frequently and the BACOP2 to degenerate into an MCLP just focusing on the first coverage. Instead, when r is too large, it is easily possible to cover all the centers twice by installing few facilities, thus making the problem uninteresting.

The values of p are set within a specific interval $[p_{\min}, p_{\max}]$. For both problems, $p_{\max}$ represents the minimum number of facilities that ensure to cover at least twice all the centers, i.e., $p_{\max} = \min\{\sum_{i \in \mathcal{I}} y_i \,:\, y(\mathcal{I}(j)) \geq 2, y_i \in \{0, 1\}, i \in \mathcal{I}\}$. Clearly, for values of $p > p_{\max}$, the objective function value does not change. The value of $p_{\min}$, instead, depends on the problem tackled. For BACOP1, $p_{\min}$ represents the minimum value that enables the model to be feasible, i.e., $p_{\min} = \min\{\sum_{i \in \mathcal{I}} y_i :\ y(\mathcal{I}(j)) \geq 1, y_i \in \{0, 1\}, i \in \mathcal{I}\}$, while for BACOP2 (which has no feasibility issues), we simply set $p_{\min} = 1$. Then, for both problems we consider 4 values of p calculated as $p = p_{\min} + \gamma \frac{(p_{\max} - 1) - p_{\min}}{3}$, with $\gamma \in \{0, 1, 2, 3\}$.

Finally, for each demand node j, the associated demand h_j is generated as a uniform integer random number in $[1, 100]$. To comply with Assumption 3, in the BACOP2, we set $\lambda^1 = 0.7$ and $\lambda^2 = 0.3$.

7.2. Performance comparison among different Branch-and-Benders-Cut variants

The implementation of a branch-and-cut method generally involves many trade-off choices. Hereafter, we compare different versions of our BBC to find the best configuration for BACOP1 and BACOP2. The comparison is done over 600 instances, i.e., 10 random repetitions for each combination of the 3 topologies, 4 values of p, and $|\mathcal{J}| \in \{10\text{K}, 50\text{K}, 100\text{K}, 500\text{K}, 1\text{M}\}$. According to the application discussed in the introduction, a time limit of 10 seconds is imposed in each run. For each BBC version v, we consider three performance indicators:

- the relative percentage gap $g_{rel} = 100 \times (z^* - z_v)/z^*$ of v, having the objective value z_v, with respect to the best objective value z^* among all the versions (for each instance);
- the CPU time t in seconds to solve the problem (for each instance);
- the percentage htl of instances for which the method hits the time limit without proving optimality.

7.2.1. BBC variants for BACOP1. For BACOP1, we tested 24 different BBC versions, one for each combination of the following choices:

- all the non-dominated optimality cuts, namely, (B_1-o_2) and (B_1-o_3), can be separated together (`All`) or each of them alone (`Opt`$_2$ and `Opt`$_3$, respectively);
- feasibility cuts (B_1-f_1) can be separated dynamically (`FeasCutSep`) or, as already noted, they can be added all together at the beginning into the RMP (`NoFeasCutSep`);
- all the cuts can be enforced only after a new feasible integer solution is found (`Lazy`), or also applied to any LP relaxation in the branch-and-bound tree (`Lazy+Frac`);
- the stabilization method can be invoked (`Stab`) or skipped to save time (`NoStab`).

For each BBC variant, along with its $htl\,[\%]$, we report in Table 1 the average and the standard deviation of g_{rel} over all the instances and of t over those that are solved to optimality within the time limit. The stabilized variants always exhibit worse performance compared to the corresponding non-stabilized ones, while the separation mode does not impact much the overall performance. Moreover, the variants that separate feasibility cuts generally achieve better results than those where all these constraints are imposed in the RMP. Exceptions are the variants combining `NoFeasCutSep`, `Lazy+Frac`, and `NoStab`, which show the best average g_{rel}. However, their average time and htl are considerably larger than the `FeasCutSep` variants. On the other hand, the fastest method (`Opt`$_2$|`FeasCutSep`|`Lazy`|`NoStab`) exhibits average gaps that are 2-3 times larger than the best ones. Hence, for the rest of the experiments on BACOP1, we selected the `Opt`$_3$|`FeasCutSep`|`Lazy`|`NoStab` variant (in bold), which is a good compromise between quality and CPU time.

7.2.2. BBC variants for BACOP2. For BACOP2, we tested 20 different BBC versions, one for each combination of the following choices (note that, in this case, feasibility cuts are not required):

- all the non-dominated optimality cuts, namely, (B_2-o_3), (B_2-o_4), (B_2-o_5), and (B_2-o_6), can be separated together (`All`) or each of them alone (`Opt`$_3$, `Opt`$_4$,`Opt`$_5$, and `Opt`$_6$, respectively);
- `Lazy` and `Lazy+Frac`, as described above;
- `Stab` and `NoStab`, as described above.

Table 1 Comparison of different BBC versions for BACOP1.

Optimality cuts	Feasibility cuts	Separation mode	Stabilization	g_{rel} [%] Avg.	g_{rel} [%] St.Dev.	$t[s]$ Avg.	$t[s]$ St.Dev.	htl[%]
`All`	`FeasCutSep`	`Lazy`	`NoStab`	0.14	0.50	0.34	0.90	16.92
			`Stab`	0.54	2.48	0.54	1.29	22.98
		`Lazy+Frac`	`NoStab`	0.27	1.47	0.53	1.39	13.46
			`Stab`	0.43	1.86	0.72	1.69	19.62
	`NoFeasCutSep`	`Lazy`	`NoStab`	0.22	0.66	1.16	1.84	25.00
			`Stab`	1.22	4.23	1.91	2.58	27.96
		`Lazy+Frac`	`NoStab`	0.12	0.50	1.26	1.93	21.83
			`Stab`	1.14	4.24	2.03	2.66	24.01
$\texttt{Opt}_2$	`FeasCutSep`	`Lazy`	`NoStab`	0.24	0.63	0.30	0.61	18.46
			`Stab`	0.58	1.75	0.42	0.93	26.50
		`Lazy+Frac`	`NoStab`	0.23	1.04	0.41	1.11	16.15
			`Stab`	0.44	1.57	0.72	1.67	21.27
	`NoFeasCutSep`	`Lazy`	`NoStab`	0.42	0.92	1.14	1.74	26.33
			`Stab`	1.36	4.23	1.91	2.56	29.57
		`Lazy+Frac`	`NoStab`	0.09	0.40	1.20	1.82	23.67
			`Stab`	1.13	4.23	2.02	2.66	26.16
$\texttt{Opt}_3$	**`FeasCutSep`**	**`Lazy`**	**`NoStab`**	**0.14**	**0.49**	**0.34**	**0.89**	**16.92**
			`Stab`	0.51	2.43	0.55	1.34	22.17
		`Lazy+Frac`	`NoStab`	0.27	1.47	0.53	1.40	13.46
			`Stab`	0.34	1.97	0.68	1.58	19.00
	`NoFeasCutSep`	`Lazy`	`NoStab`	0.16	1.34	1.13	1.78	25.17
			`Stab`	1.13	4.34	1.78	2.43	25.50
		`Lazy+Frac`	`NoStab`	0.11	0.49	1.26	1.94	21.83
			`Stab`	1.06	4.39	1.90	2.51	21.50

For each BBC variant, along with its $htl[\%]$, we report in Table 2 the average and the standard deviation of g_{rel} over all the instances and of t over those that are solved to optimality within the time limit. As for BACOP1, the separation mode does not impact the performance much. Instead, here the stabilized variants consistently show better performance in terms of quality, CPU time, and htl. We select $\texttt{Opt}_4$|`Lazy`|`Stab` as the best configuration since it achieves the best values in two indicators and the second-best value (with a 0.2% margin from the best) for htl. Actually, the use of (B_2-o_4) seems to work quite well everywhere. Notably, at least two other variants, i.e., $\texttt{Opt}_3$|`Lazy`|`Stab` and $\texttt{Opt}_4$|`Lazy+Frac`|`Stab`, perform very closely.

Table 2 Comparison of different BBC versions for BACOP2.

Optimality cuts	Separation mode	Stabilization	g_{rel} [%] Avg.	g_{rel} [%] St.Dev.	$t[s]$ Avg.	$t[s]$ St.Dev.	htl[%]
`All`	`Lazy`	`NoStab`	0.52	0.69	1.39	2.05	52.96
		`Stab`	0.15	0.05	0.63	1.09	13.33
	`Lazy+Frac`	`NoStab`	0.53	0.70	1.39	2.01	53.51
		`Stab`	0.03	0.06	0.51	1.01	13.70
$\texttt{Opt}_3$	`Lazy`	`NoStab`	0.42	0.66	0.62	1.47	50.74
		`Stab`	0.01	0.06	0.70	1.24	13.14
	`Lazy+Frac`	`NoStab`	0.44	0.68	0.67	1.65	51.48
		`Stab`	0.01	0.06	0.65	1.30	13.33
$\texttt{Opt}_4$	**`Lazy`**	`NoStab`	0.29	0.38	0.89	1.83	39.81
		`Stab`	**0.01**	**0.06**	**0.54**	**1.17**	**13.33**
	`Lazy+Frac`	`NoStab`	0.29	0.39	0.93	1.91	41.66
		`Stab`	0.01	0.06	0.55	1.22	13.51
$\texttt{Opt}_5$	`Lazy`	`NoStab`	0.48	0.80	1.46	2.19	61.85
		`Stab`	0.01	0.08	0.72	1.16	13.14
	`Lazy+Frac`	`NoStab`	0.49	0.79	1.49	2.21	62.78
		`Stab`	0.01	0.08	0.51	1.07	13.70
$\texttt{Opt}_6$	`Lazy`	`NoStab`	0.52	0.72	1.32	1.94	52.96
		`Stab`	0.02	0.06	0.63	1.08	13.33
	`Lazy+Frac`	`NoStab`	0.54	0.71	1.44	2.11	53.33
		`Stab`	0.02	0.06	0.52	1.04	13.70

Table 3 Computational comparison for BACOP1.

Topology	CPLEX			CPLEX Benders			BBC (Opt$_3$|FeasCutSep|Lazy|NoStab)		
	$t[s]$	$htl[\%]$	$noSol[\%]$	$t[s]$	$htl[\%]$	$noSol[\%]$	$t[s]$	$htl[\%]$	$noSol[\%]$
Mono-polar	4.53	0.00	0.00	2.49	0.00	0.00	0.19	0.00	0.00
Multi-polar	1.19	0.00	0.00	0.61	0.00	0.00	0.13	0.00	0.00
Uniform	0.20	60.00	60.00	8.23	92.50	85.83	0.70	50.76	0.00
Avg.:	2.06	20.00	20.00	3.78	30.83	28.61	0.34	16.92	0.00

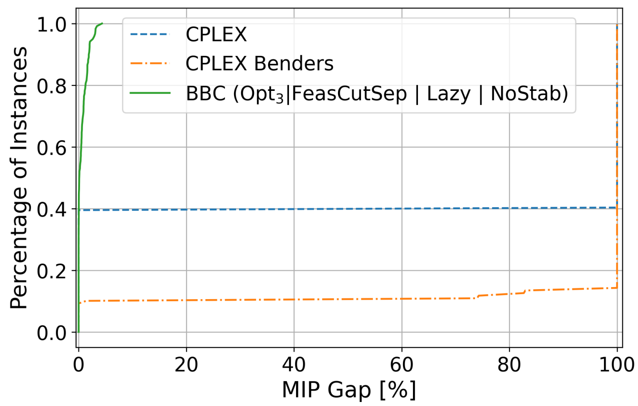


Figure 4 Cumulative percentage of instances solved within the time limit up to any MIP gap percentage for BACOP1.

7.3. Performance comparison against state-of-the-art methods

In this section, we compare the best BBC variants found in Sections 7.2.1–7.2.2, namely, BBC(Opt$_3$|FeasCutSep|Lazy|NoStab) for BACOP1 and BBC(Opt$_4$|Lazy|Stab) for BACOP2, against the CPLEX branch-and-cut framework solving the ILP models reported in Section 3.1 and 3.2 (CPLEX) and the automatic Benders decomposition provided by the CPLEX solver on the same reformulation presented in Section 3.3 (CPLEX Benders). The comparison is done on the same 600 instances introduced before, and a time limit of 10 seconds is imposed on each method.

Concerning BACOP1, we report in Table 3 the indicators t (averaged over all instances of each topology solved to optimality) and htl, along with the percentage $noSol$ of times in which no feasible solutions are found. More details are reported in Table **??**. Our BBC exhibits the smallest average percentage of instances not solved to optimality (around 17%) and always finds a feasible solution. It is worth analyzing the results obtained for each topology. Mono-polar and multi-polar instances are always solved to optimality within the time limit by all three methods; however, for these topologies, our BBC is on average 11 times faster than CPLEX and about 7 times faster than CPLEX Benders. Uniform instances are, instead, more difficult to solve. For a large percentage of these instances, the time limit is reached (50.76% for our BBC, 60% for CPLEX, and 92.50% for CPLEX Benders). Given the large number of instances reaching the time limit, we complete the comparison by plotting in Figure 4, only for the uniform instances, the empirical cumulative distribution function (ECDF) of the percentage of instances solved (y-axis) by each method up to a certain value of percentage MIP gap (on x-axis) within the time limit. The percentage MIP gap is calculated as $MIP_g = 100 \times (z^{UB} - z^{BEST})/z^{BEST}$, where z^{UB} is the best upper bound and z^{BEST} is the value of the best integer feasible solution found during

Table 4 Computational comparison for BACOP2.

Topology	CPLEX			CPLEX Benders			BBC(Opt$_4$|Lazy|Stab)			BBC(Opt$_4$|Lazy|NoStab)		
	$t[s]$	$htl[\%]$	$noSol[\%]$	$t[s]$	$htl[\%]$	$noSol[\%]$	$t[s]$	$htl[\%]$	$noSol[\%]$	$t[s]$	$htl[\%]$	$noSol[\%]$
Mono-polar	9.74	90.74	90.74	8.54	99.99	64.44	0.74	29.43	0.00	0.55	66.67	0.00
Multi-polar	9.23	78.72	78.72	8.81	99.80	99.80	0.59	8.89	0.00	0.88	33.89	0.00
Uniform	9.01	75.00	75.00	9.87	97.78	60.00	0.35	1.67	0.00	1.03	18.89	0.00
Avg.:	9.32	81.49	81.49	9.07	99.19	74.74	0.54	13.33	0.00	0.89	39.81	0.00

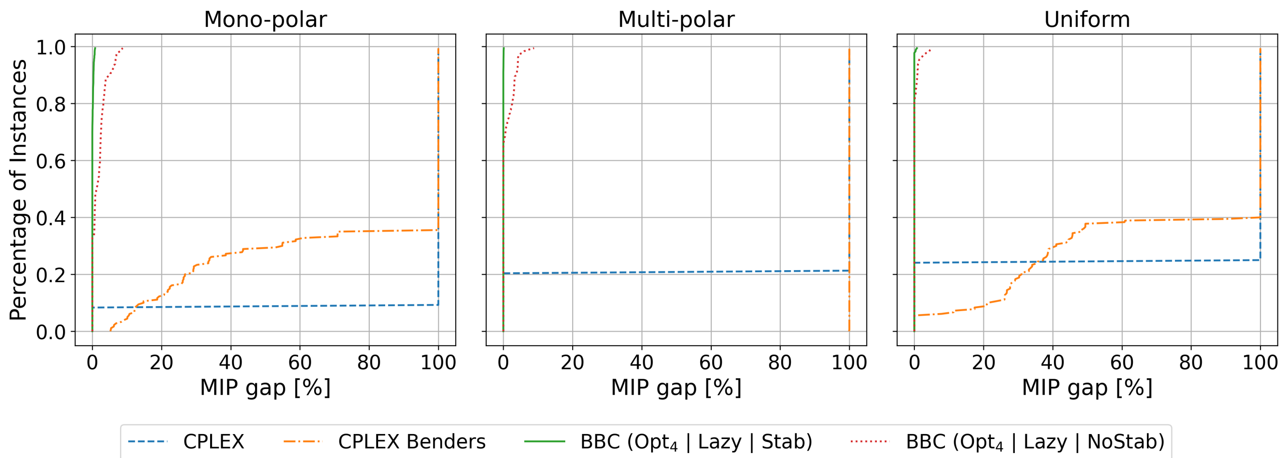


Figure 5 Cumulative percentage of instances solved within the time limit up to any MIP gap percentage for BACOP2.

the optimization process. When the time limit is reached and no feasible solution is found, we report the gap of 100%. We can see that `CPLEX` and `CPLEX Benders` are particularly unreliable on instances not solved to optimality, obtaining 60% and 85% of instances with the worst possible MIP gap, while for our BBC the worst gap does not exceed 4-5% in all the cases.

Similarly, concerning the three methods applied to BACOP2, we report the t, htl, and $noSol$ indicators in Table 4, while we plot in Figure 5 the ECDFs for all the topologies (given the always non-null values of htl). Details are reported in Table **??**. To better understand the effect of the stabilization on our BBC, these results will also include the version BBC(Opt$_4$|Lazy|NoStab). BACOP2 appears to be a much more difficult problem to tackle by `CPLEX` and `CPLEX Benders`, regardless of the topology, with an average time for instances solved to optimality greater than 9 seconds. `CPLEX` hits the timelimit in more than 80% of the cases. Moreover, for all of these unsolved instances, not even a feasible solution can be found. Instead, `CPLEX Benders` hits the time limit in 99% of the cases and, on average, is not able to produce a feasible solution in about 75% of the times. On the other hand, our best BBC setting always finds a feasible solution with an average solution time of 0.25 seconds (uniform instances appear here the easiest to solve, while mono-polar are the hardest ones) and the average number of non-optimal solutions is around 13% (again, mono-polar instances are the most difficult ones). The solution time boost is achieved thanks to the implemented stabilization method that, on average, speeds-up the solution time by 4 seconds and proves optimality for additional 25% of instances. Figure 5 demonstrates that the state-of-the-art solver is largely outperformed by our BBC implementations. For all three considered topologies, our best BBC always returns solutions within 1-2% of the MIP gap, whereas the two CPLEX implementations

Table 5 Computational comparison for BACOP1 with various time limits.

Topology	TL$[s]$	CPLEX				BBC (`Opt`$_3$|`FeasCutSep`|`Lazy`|`NoStab`)			
		$t[s]$	$htl[\%]$	$noSol[\%]$	$MIPgap[\%]$	$t[s]$	$htl[\%]$	$noSol[\%]$	$MIPgap[\%]$
Uniform	10	0.20	60.00	60.00	NA	0.70	50.76	0.00	0.32
	60	41.21	59.50	59.50	NA	39.77	46.50	0.00	0.03
	300	166.62	49.00	49.00	NA	151.48	40.00	0.00	0.02
	600	317.37	47.50	47.50	NA	258.18	31.00	0.00	0.01

Table 6 Computational comparison for BACOP2 with various time limits.

Topology	TL$[s]$	CPLEX				BBC(`Opt`$_4$|`Lazy`|`Stab`)			
		$t[s]$	$htl[\%]$	$noSol[\%]$	$MIPgap[\%]$	$t[s]$	$htl[\%]$	$noSol[\%]$	$MIPgap[\%]$
Mono-polar	10	9.74	90.74	90.74	NA	0.74	29.43	0.00	0.28
	60	15.78	63.88	63.88	NA	1.79	26.11	0.00	0.22
	300	52.23	52.77	52.77	NA	6.06	22.78	0.00	0.17
	600	110.00	43.51	43.51	NA	18.07	20.00	0.00	0.16
Multi-polar	10	9.23	78.72	78.72	NA	0.59	8.89	0.00	0.05
	60	10.40	64.81	63.88	0.21	1.59	6.67	0.00	0.04
	300	65.23	46.29	46.29	NA	6.67	2.78	0.00	0.03
	600	157.00	25.00	25.00	NA	13.10	1.11	0.00	0.03
Uniform	10	9.01	75.00	75.00	NA	0.35	1.67	0.00	0.51
	60	7.49	66.66	66.66	NA	0.43	1.67	0.00	0.12
	300	64.67	50.00	50.00	NA	1.72	0.00	0.00	0.00
	600	155.13	34.25	34.25	NA				

reach the time limit in most cases. We also notice a difference between the stabilized and the non-stabilized BBC implementation, with the latter obtaining some MIP gaps peaks around 8-10%.

7.4. BBC behavior for longer computations

Aiming to generalize the results obtained within 10 seconds to a broader set of applications, in this section we compare our best BBC techniques (namely, `BBC(Opt`$_3$`|FeasCutSep|Lazy|NoStab)` for BACOP1 and `BBC(Opt`$_4$`|Lazy|Stab)` for BACOP2) with the CPLEX branch-and-cut (`CPLEX`) also considering a much longer time limit (TL), i.e., 60, 300, and 600 seconds. `CPLEX Benders` is not reported because of the very bad performance in both problems, with a lot of out-of-memory occurrences. For BACOP1 and BACOP2, respectively, Table 5 and 6 show, along with t, htl, and $noSol$ indicators, also the $MIPgap$, which is the percentage optimality gap averaged over all the instances for which the method hits the time limit.

In Table 5, which reports the results only for uniform BACOP1 instances (as all the mono-polar and multi-polar instances can be solved in less than 10 seconds), we observe that the average computational time t tends to increase with TL, since more instances are solved within a longer runtime, while both htl and $noSol$ decrease. Nevertheless, apart from the case TL$= 10$, both t and htl remain lower for BBC, indicating that the proposed approach can solve a larger number of instances with a smaller average computational time. In particular, the relative saving in time increases as TL increases, stating that our method is robust and is not outperformed by `CPLEX` if more time is available. Finally, note that the MIP gap is always not available (NA) for `CPLEX` since all the instances hitting the TL actually have no feasible solutions. The same indicator instead is very good for our BBC, becoming negligible (below 0.03%) for TL equal to 60 seconds or more.

Table 6 reports the corresponding results for BACOP2, where we can see that the advantage of our method becomes even more pronounced as additional computational time is allowed. In fact, we can see that for BBC, t never exceeds 18.07 seconds, while `CPLEX` reached sometimes more than 150 seconds. Moreover, for `CPLEX`, the number of instances hitting the time limit decreases as TL increases, but for a large portion of these instances, `CPLEX` does not even find a feasible solution. Indeed, even after increasing the time limit to 10 minutes, no feasible solution is found for 43.5%, 25%, and 34.3% of mono-polar, multi-polar, and uniform instances, respectively. Instead, our BBC always finds a feasible solution and is able to reduce htl even further. Only 1.1% of multi-polar instances remain unsolved after 10 minutes, whereas all uniform instances can be solved to optimality in less than 5 minutes. Still, mono-polar instances remain the most difficult ones for our BBC, with an htl of 20%. However, the MIP gap for these instances not solved to optimality is very low, always below 0.51% and below 0.22% for TL equal to 60 seconds or more. Note that we did not run BBC for uniform instances for TL= 600, since all these instances have been solved to optimality within TL= 300, with an average time of 1.72 seconds.

Finally, we again emphasize that our BBC always finds at least a feasible solution for both BACOP1 and BACOP2 instances, which is extremely important for practical application.

8. Conclusions

In this paper, we present a Branch-and-Benders-Cut approach for BACOP1 and BACOP2 that demonstrates strong empirical performance across extensive computational tests. BACOP1 and BACOP2 are two important maximal covering location problems that explicitly address service redundancy. Our method leverages key solution properties to derive multiple families of Benders cuts. On the theoretical side, we study pairwise dominance relationships among these cuts. On the practical side, we implement several acceleration techniques that enable our approach to outperform state-of-the-art solvers. Furthermore, our method demonstrates much more robust performance across varying instance sizes and topologies.

We want to highlight some possible directions for future research related to this topic. First, the developed approach can be seen as a quite general framework that can exploit the properties of many other covering location problems where redundancy of service plays an important role. Second, the two problems studied could be addressed by including stochastic data, perhaps under a two-stage stochastic programming paradigm. In that context, the proposed Benders cuts could be combined with the *L-shaped* method, typically used in stochastic programming. Finally, many recent advances in Benders decomposition (see, e.g., Trukhanov et al. 2010, Ramírez-Pico et al. 2023 and a recent survey in Clautiaux and Ljubić 2025) could be exploited to further improve these implementations.

References

Adulyasak Y, Cordeau JF, Jans R (2015) Benders decomposition for production routing under demand uncertainty. *Operations Research* 63:851–867.

Armacost RL (1992) A 0-1 nonlinear programming model for coast guard fisheries law enforcement aircraft patrols. *European Journal of Operational Research* 56:134–145.

Ben-Ameur W, Neto J (2007) Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks* 49(1):3–17.

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4(1):238–252.

Bonami P, Salvagnin D, Tramontani A (2020) Implementing automatic benders decomposition in a modern mip solver. Bienstock D, Zambelli G, eds., *Integer Programming and Combinatorial Optimization*, 78–90.

Brotcorne L, Laporte G, Semet F (2003) Ambulance location and relocation models. *European Journal of Operational Research* 147:451–463.

Chowdhury N, Pilo F, Pisano G (2020) Optimal energy storage system positioning and sizing with robust optimization. *Energies* 13(3).

Church RL, ReVelle CS (1974) The maximal covering location problem. *Papers of the Regional Science Association* 32:101–118.

Clautiaux F, Ljubić I (2025) Last fifty years of integer linear programming: A focus on recent practical advances. *European Journal of Operational Research* 324:707–731.

Codato G, Fischetti M (2006) Combinatorial benders' cuts for mixed-integer linear programming. *Operations Research* 54(4):756–766.

Coniglio S, Furini F, Ljubić I (2022) Submodular maximization of concave utility functions composed with a set-union operator with applications to maximal covering location problems. *Mathematical Programming* 196(1):9–56.

Cordeau JF, Furini F, Ljubić I (2019) Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research* 275(3):882–896.

Current J, O'Kelly M (1992) Locating emergency warning sirens. *Decision Sciences* 23:221–234.

Daskin MS (1983) A maximum expected covering location model: Formulation, properties and heuristic solution. *Transporation Science* 17:1–113.

Daskin MS, Stern EH (1981) A hierarchical objective set covering model for emergency medical service vehicle deployment. *Transportation Science* 15:137–152.

Doerner K, Hartl R (2008) *Health Care Logistics, Emergency Preparedness, and Disaster Relief: New Challenges for Routing Problems with a Focus on the Austrian Situation*, volume 43 (Springer, Boston, MA).

Doerner KF, Gutjahr WJ, Hartl RF, Karall M, Reimann M (2005) Heuristic solution of an extended double-coverage ambulance location problem for Austria. *Central European Journal of Operations Research* 13:325–340.

Duran-Mateluna C, Ales Z, Elloumi S (2023) An efficient Benders decomposition for the p-median problem. *European Journal of Operational Research* 308(1):84–96.

Fadda E, Ljubić I, Manerba D (2026) A tailored branch-and-benders-cut approach for backup covering problems. URL http://dx.doi.org/10.1287/ijoc.2025.1394.cd, available for download at https://github.com/INFORMSJoC/2025.1394.

Fadda E, Manerba D, Cabodi G, Camurati P, Tadei R (2021a) Evaluation of optimal charging station location for electric vehicles: An italian case-study. Fidanova S, ed., *Recent Advances in Computational Optimization. Springer Studies in Computational Intelligence, vol. 920*, 71–87 (Springer).

Fadda E, Manerba D, Cabodi G, Camurati PE, Tadei R (2021b) Comparative analysis of models and performance indicators for optimal service facility location. *Transportation Research Part E: Logistics and Transportation Review* 145:102174.

Fadda E, Manerba D, Tadei R (2024) How to locate services optimizing redundancy: A comparative analysis of k-covering facility location models. *Socio-Economic Planning Sciences* 94:101938.

Fadda E, Manerba D, Tadei R, Camurati P, Cabodi G (2019) KPIs for Optimal Location of charging stations for Electric Vehicles: the Biella case-study. Ganzha M, Maciaszek L, Paprzycki M, eds., *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*, volume 18 of *ACSIS*, 123–126.

Farahani RZ, Asgari N, Heidari N, Hosseininia M, Goh M (2012) Covering problems in facility location: A review. *Computers & Industrial Engineering* 62(1):368–407.

Fischetti M, Ljubić I, Sinnl M (2017) Redesigning benders decomposition for large-scale facility location. *Management Science* 63(7):2146–2162.

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, Salvagnin D, Zanette A (2010) A note on the selection of benders' cuts. *Mathematical Programming* 124:175–182.

Gaar E, Sinnl M (2022) A scaleable projection-based branch-and-cut algorithm for the p-center problem. *European Journal of Operational Research* 303(1):78–98.

García S, Marín A (2019) Covering location problems. Laporte G, Nickel S, Saldanha da Gama F, eds., *Location Science*, 99–119.

Gendreau M, Laporte G, Semet F (1997) Solving an ambulance location model by tabu search. *Location Science* 5:75–88.

Hogan K, ReVelle C (1986) Concepts and applications of backup coverage. *Management Science* 32:1434–1444.

IBM ILOG CPLEX v2210 Documentation (2022) Linear reduction switch. URL `https://www.ibm.com/docs/en/icos/22.1.0?topic=parameters-linear-reduction-switch-deprecated`.

Kahr M, Leitner M, Ljubić I (2024) The impact of passive social media viewers in influence maximization. *INFORMS Journal on Computing* 36(6):1362–1381.

Lamontagne S, Carvalho M, Atallah R (2024) Accelerated Benders decomposition and local branching for dynamic maximum covering location problems. *Computers & Operations Research* 167:106673.

Li R, Nadarajah S (2020) A review of student's t distribution and its generalizations. *Empirical Economics* 58(3):1461–1490.

Li WWL, Shen Y, Zhang YJ, Win MZ (2013) Robust power allocation for energy-efficient location-aware networks. *IEEE/ACM Transactions on Networking* 21(6):1918–1930.

Li X, Zhao Z, Zhu X, Wyatt T (2011) Covering models and optimization techniques for emergency response facility location and planning: a review. *Mathematical Methods of Operations Research* 74:281–310.

Ljubić I, Pozo MA, Puerto J, Torrejón A (2024) Benders decomposition for the discrete ordered median problem. *European Journal of Operational Research* 317:858–874.

Ljubić I, Putz P, Salazar-González JJ (2012) Exact approaches to the single-source network loading problem. *Networks* 59(1):89–106.

Maher SJ (2021) Implementing the branch-and-cut approach for a general purpose benders' decomposition framework. *European Journal of Operational Research* 290(2):479–498.

Marianov V (2017) *Location Models for Emergency Service Applications*, chapter 11, 237–262.

Mitropoulos P, Mitropoulos I, Giannikos I, Sissouras A (2006) A biobjective model for the locational planning of hospitals and health centers. *Health Care Management Science* 9:171–179.

Moore GC, ReVelle C (1982) The hierarchical service location problem. *Management Science* 28:775–780.

Plane DR, Hendrick TE (1977) Mathematical programming and the location of fire companies for the denver fire department. *Operations Research* 25(4):563–578.

Rahmaniani R, Crainic TG, Gendreau M, Rei W (2017) The benders decomposition algorithm: A literature review. *European Journal of Operational Research* 259(3):801–817.

Ramírez-Pico C, Ljubić I, Moreno E (2023) Benders adaptive-cuts method for two-stage stochastic programs. *Transportation Science* 57(5):1252–1275.

Snyder SA, Haight RG (2016) Application of the maximal covering location problem to habitat reserve site selection: A review. *International Regional Science Review* 39(1):28–47.

Sorensen P, Church R (2010) Integrating expected coverage and local reliability for emergency medical services location problems. *Socio-Economic Planning Sciences* 44:8–18.

Tanınmış K, Aras N, Güney E, Sinnl M (2024) Benders decomposition algorithms for minimizing the spread of harmful contagions in networks. *Computers & Operations Research* 167:106675.

Trukhanov S, Ntaimo L, Schaefer A (2010) Adaptive multicut aggregation for two-stage stochastic linear programs with recourse. *European Journal of Operational Research* 206(2):395–406.

Wang W, Wu S, Wang S, Zhen L, Qu X (2021) Emergency facility location problems in logistics: Status and perspectives. *Transportation Research Part E: Logistics and Transportation Review* 154:102465.

Wang YR, Chen WK, Ljubić I (2025) An efficient branch-and-cut algorithm for the multiple probabilistic covering location problem. URL https://arxiv.org/abs/2511.17128.

Wentges P (1996) Accelerating benders' decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research* 44:267–290.

Zhang B, Peng J, Li S (2017) Covering location problem of emergency service facilities in an uncertain environment. *Applied Mathematical Modelling* 51:429–447.