An Exact Algorithm for Large-Scale Non-convex Quadratic Facility Location

Carlos Armando Zetina
IVADO Labs and Canada Excellence Research Chair in Data Science for Real-Time Decision-Making, Polytechnique Montréal, Canada, carlos.zetina@ivadolabs.com

Ivan Contreras
Concordia University and Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) Montreal, Canada H3G 1M8, ivan.contreras@cirrelt.ca

Sachin Jayaswal
Indian Institute of Management Ahmedabad, Gujarat, India, sachin@iima.ac.in

We study a general class of quadratic capacitated p-location problems facility location problems with single assignment where a non-separable, non-convex, quadratic term is introduced in the objective function to account for the interaction cost between facilities and customer assignments. This problem has many applications in the field of transportation and logistics where its most well-known special case is the single-allocation hub location problem and its many variants. The non-convex, binary quadratic program is linearized by applying a reformulation-linearization technique and the resulting continuous auxiliary variables are projected out using Benders decomposition. The obtained Benders reformulation is then solved using an exact branch-and-cut algorithm that exploits the underlying network flow structure of the decomposed separation subproblems to efficiently generate strong Pareto-optimal Benders cuts. Additional enhancements such as a matheuristic, a partial enumeration procedure, and variable elimination tests are also embedded in the proposed algorithmic framework. Extensive computational experiments on benchmark instances (with up to 500 nodes) and on a new set of instances (with up to 1,000 nodes) of four variants of single-allocation hub location problems confirm the algorithm’s ability to scale to large-scale instances.

Key words: quadratic facility location; hub location; binary quadratic programs; Benders decomposition;

1. Introduction

Discrete facility location problems (FLPs) constitute a fundamental class of problems in location science. Given a finite set of customers and a finite set of potential locations for the facilities, FLPs consist of determining the location of facilities to open and the assignment of customers to the selected facilities in order to optimize some cost (or service)-oriented objective. FLPs arise naturally in a wide variety of applications such as supply chain management, emergency and health care systems, humanitarian logistics, telecommunications systems, districting, and urban transportation planning, among others (see, Drezner and Hamacher 2001, Correia and da Gama 2013).
Given the wide range of applications, many variants of FLPs have been studied and they differ with respect to the considered objective, type of demand, network topology, and assignment pattern, among other features. Some of the classical problems involving a cost-oriented objective are fixed-charge facility location [Kuehn and Hamburger 1963, Fernandez and Landete 2015, Fischetti et al. 2017], p-median [Hakimi 1964, Daskin and Lee Maass 2015], uncapacitated p-location [Nemhauser et al. 1978, Wolsey 1983, Ortiz-Astorquiza et al. 2019], and single-source capacitated facility location problems [Neebe and Rao 1983, Gadegaard et al. 2018]. FLPs with service-oriented objectives are p-center [Hakimi 1964, Calik et al. 2015] and covering location problems [Toregas et al. 1971, Church and ReVelle 1974, Garcia and Marín 2015]. In such classical FLP variants, the associated objective functions are assumed to be linear functions of the setup cost for opening facilities and/or of the allocation cost of the demand served by the open facilities.

In this paper, we study a general class of quadratic capacitated p-location problems with single assignments (QCpLPs) which can be defined as follows. Let \( N \) be a set of nodes representing both the set of potential facility locations and the set of customers. Let \( f_k \) and \( b_k \) denote the setup cost and capacity, respectively, of potential facility \( k \in N \), and \( c_{ik} \) denote the cost of allocating customer \( i \in N \) to facility \( k \in N \). Let \( q_{ikjm} \) represent the interaction cost between the allocation of customer \( i \in N \) to facility \( k \in N \) and the allocation of customer \( j \in N \) to facility \( m \in N \). When \( i = k \) and \( j = m \), \( q_{kkmm} \) corresponds to the interaction cost between facilities \( k \) and \( m \). The QCpLP consists of selecting a set of facilities to open, such that no more than \( p \) facilities are opened, and of assigning each customer to exactly one open facility while respecting capacity constraints on the amount of demand served by each facility. The objective is to minimize the total setup cost, allocation cost and the (quadratic) interaction cost between facilities and between customer assignments.

The main contributions of this paper are the following. The first is to introduce the QCpLPs, which can be stated as a binary quadratic program (BQP) having a non-separable, non-convex quadratic function. QCpLPs are quite a general class of problems that can be used to model numerous applications in transportation and telecommunications where a cost (or benefit) captures the interaction between facilities and/or customer assignments. It contains several special cases of hub location problems that are of particular interest such as the p-hub median problem with single assignments [O’Kelly 1987, Ernst and Krishnamoorthy 1992, 1996], the uncapacitated hub location problem with single assignments [Campbell 1994, Ghaﬁfarinasab and Kara 2018], the capacitated hub location problem with single assignments [Contreras et al. 2011, Meier and Clausen 2018], and the star-star hub network design problem [Helme and Magnanti 1989, Labbé and Yaman 2008]. It also has as particular cases some BQPs such as the quadratic semi-assignment problem [Saito et al. 2009, Rostami et al. 2018] and its capacitated variant. In addition, the QCpLP can also be used to model situations in which a negative interaction between obnoxious (or undesirable)
facilities exist and as a consequence, it is desirable to locate them far from each other. For instance, dangerous plants producing or handling hazardous materials are a source of potential damage to population and their negative impact can be measured as a function of the distances between them (Carrizosa and Tóth 2015). A negative interaction among facilities also arises when the facilities compete for the same market and locating facilities as far away from each other is beneficial to mitigate cannibalization and to optimize their competitive market advantage (Curtin and Church 2006, Lei and Church 2013).

The second contribution of this paper is to introduce and computationally compare two strong mixed-integer linear programming (MILP) reformulations of QCpLP. The first is obtained by applying the reformulation-linearization technique (RLT) of Adams and Sherali (1986) to the original BQP. The second reformulation is obtained by projecting out a large number of (continuous) variables from a relaxation of the RLT reformulation using Benders decomposition. Although the resulting Benders reformulation is not as strong as the complete RLT reformulation, its separation problem has a network flow structure, which can be exploited to solve QCpLPs efficiently. The third contribution is to present an exact branch-and-cut algorithm based on the Benders reformulation to solve large-scale instances of the QCpLP. We develop several algorithmic refinements to accelerate its convergence. These include: i) an exact separation procedure that efficiently generates Pareto-optimal cuts at fractional and integer points by solving transportation problems, ii) the use of a matheuristic to generate high quality initial feasible solutions, iii) variable elimination tests performed at the root node, and iv) a partial enumeration phase to further reduce the size of the problem before branching is performed. To assess the performance of our algorithm, we performed extensive computational experiments on several sets of benchmark large-scale instances of networks with up to 1,000 nodes for special uncapacitated and capacitated cases previously studied in the literature.

The remainder of the paper is organized as follows. Section 2 provides a succinct review on nonlinear discrete FLPs and positions this paper in view of related work. Section 3 formally defines the problem and presents a BQP formulation. The RLT reformulation and the Benders reformulation are presented in Sections 4 and 5, respectively. Section 6 presents the proposed acceleration techniques and the branch-and-cut algorithm. Section 7 presents the results of extensive computational experiments performed on a wide variety of large-scale instances of variants of single allocation hub location problems. Conclusions follow in Section 8.

2. Literature Review

The study of nonlinear discrete FLPs began with the work of Zangwill (1968) where production and distribution costs are considered to be separable concave functions of the quantities produced
and distributed. In particular, Zangwill (1968) considers an uncapacitated FLP with arbitrary concave cost functions and presents an exact algorithm that uses a characterization of the extreme points of the feasible region and dynamic programming. Soland (1974) extends the previous problem to the capacitated case and develops an exact branch-and-bound algorithm in which dual bounds are obtained at the nodes of the enumeration tree by solving a transportation problem. O’Kelly (1986a,b, 1987) are arguably the first studies dealing with quadratic extensions of FLPs. In particular, O’Kelly (1987) introduce a quadratic extension of the $p$-median problem to design hub-and-spoke networks where a set of hub facilities are used as transshipment, consolidation or sorting points to connect a large set of interacting nodes. A non-convex quadratic term arises from the interaction of hub facilities due to the routing of flows between node pairs. The seminal work of O’Kelly gave rise to a new family of FLPs, denoted as hub location problems (HLPs), that has since evolved into a rich research area (see, Campbell and O’Kelly 2012, Contreras and O’Kelly 2019). HLPs lie at the heart of network design and planning in transportation and telecommunication systems. Application areas of HLPs in transportation include air freight and passenger travel, postal delivery, express package delivery, trucking, liner shipping, public transportation, and rapid transit systems. Applications of HLPs in telecommunications arise in the design of various digital data service networks. For several particular classes of HLPs, exact solution algorithms have been developed using various integer programming techniques: i) branch-and-bound algorithms that use bounding procedures such as combinatorial algorithms (Ernst and Krishnamoorthy 1998), dual-ascent algorithms (Cánovas et al. 2007), and Lagrangean relaxations (Contreras et al. 2011b, Rostami et al. 2016, Tanash et al. 2017, Alibeyg et al. 2018); ii) branch-and-price algorithms (Contreras et al. 2011d, Rothenbcher et al. 2016); iii) branch-and-cut algorithms (Labbé et al. 2005, Contreras and Fernández 2014, Zetina et al. 2013, Meier and Clausen 2018); and iv) Benders decomposition algorithms (Camargo et al. 2011, Contreras et al. 2011a,c, Martins de Sá et al. 2015, Mahéo et al. 2019).

Helme and Magnanti (1989) study a quadratic extension of the uncapacitated fixed-charge facility location problem arising in the design of satellite communication networks. In this case, demand nodes in the same cluster communicate via terrestrial links whereas different clusters communicate with each other via satellite links using earth stations. A (non-convex) quadratic term is considered in the objective to account for both the within and between cluster traffic costs. Desrochers et al. (1995) introduce a congested facility location problem in which congestion takes place at facilities and is characterized by convex delay functions that approximate the mean delay experienced by users waiting to be served at a facility. The authors propose a column generation algorithm as a bounding procedure within a branch-and-price algorithm to obtain optimal solutions of the problem. Fischetti et al. (2016) study a congested capacitated facility location problem in which the
congestion cost is modeled as a convex quadratic function. The authors present a perspective formulation and an associated Benders reformulation in which all continuous variables are projected out. The resulting convex mixed integer nonlinear program is solved with a branch-and-cut algorithm to obtain optimal solutions for the linear and quadratic capacitated variants. Harkness and ReVelle (2003) study a closely related problem in which unit production costs increase once a certain scale of output is reached. Such diseconomies of scale in production are modeled using convex functions to represent situations in which production capacity can be stretched by incurring some additional cost due to overtime, procuring more costly materials, or neglecting equipment maintenance schedules. The authors propose alternative linear integer programming formulations using piecewise linear approximations of the convex functions. Lu et al. (2014) focus on a more comprehensive model in which both economies and diseconomies of scale are jointly considered. In the context of production-distribution systems, unit production costs tend to initially decrease as production volume increases, but after some threshold production level, unit costs start to increase due to over-utilization of resources, overtime and facility congestion. Economies and diseconomies of scale are captured by an inverse S-shaped function which is initially concave and then turns convex. The authors present various column generation heuristics to approximately solve this problem. Günlük et al. (2007) and Günlük and Linderoth (2012) study a quadratic extension of the fixed-charge facility location problem in which a separable convex quadratic term is used. In particular, allocation costs are assumed to be proportional to the square of a customer’s demand served by an open facility. This problem arises in the design of electric power distribution systems where distribution transformers are located close to customer nodes so as to minimize power loss. Because of cable resistance, power loss in a cable is proportional to the square of the current and the resistance of the cable. Fischetti et al. (2017) develop a Benders decomposition algorithm based on the perspective formulation of Günlük and Linderoth (2012) for solving large-scale instances for both the linear and separable convex quadratic variants. Finally, we refer the reader to Elhedhli and Merrick (2012), Lu et al. (2014), and Berman and Krass (2015) for additional references on nonlinear FLPs.

From a methodological perspective, our paper is related to Contreras et al. (2011a) where an exact algorithm based on Benders decomposition is presented to solve the uncapacitated hub location problem with multiple assignments. Similar to Contreras et al. (2011a), we use a Benders reformulation in combination with several algorithmic refinements such as the separation of Pareto-optimal cuts, partial enumeration, and heuristics to obtain optimal solutions to large-scale instances. However, the way these techniques are used in our exact algorithm are significantly different. The Benders reformulation used in Contreras et al. (2011a) is obtained from a (mixed-integer linear) path-based formulation whereas the one we present in this work is obtained from a relaxation of a strong RLT reformulation of a BQP. Contreras et al. (2011a) use a standard
iterative Benders decomposition algorithm in which at every iteration, a relaxed integer master problem is optimally solved to derive a cut. In our work, we solve the Benders reformulation with a branch-and-cut framework in which Benders cuts are separated at fractional and integer solutions in a single enumeration tree. Moreover, the partial enumeration, heuristic, and Benders cuts of our algorithm exploit the information from fractional solutions whereas in Contreras et al. (2011a) only integer solutions are used for these techniques. Finally, the separation procedure for generating (non-stabilized) Pareto-optimal cuts presented in Contreras et al. (2011a) is approximate and can only be used for integer solutions, whereas in our algorithm we use an efficient exact separation routine to generate stabilized Pareto-optimal cuts for both fractional and integer solutions.

3. Problem Definition

We recall that QC\textsubscript{p}LP seeks to select a set of facilities to open, such that no more than \( p \) facilities are opened, and to assign each customer to exactly one open facility while respecting capacity constraints on the amount of demand served by each facility. The objective is to minimize the total setup cost, allocation cost and the (quadratic) interaction cost between facilities and customer assignments. To formulate the problem, for each pair \( i, k \in N \), we define binary location/allocation variables \( z_{ik} \) equal to one if and only if node \( i \) is assigned to facility \( k \). When, \( i = k \), variable \( z_{kk} \) represents whether a facility is installed or not at node \( k \). It is also assumed that when \( z_{kk} \) is equal to one, node \( k \) will be allocated to itself. Hereafter, whenever obvious, the limits on summations and the \( \forall \) symbol associated with the constraints are suppressed for ease of notation. The QC\textsubscript{p}LP can be stated as follows BQP:

\[
(QF) \quad \text{min} \quad \sum_{k \in N} f_k z_{kk} + \sum_{i \in N} \sum_{k \in N} c_{ik} z_{ik} + \sum_{i \in N} \sum_{i < j} \sum_{k \in N} \sum_{m \in N} q_{ikjm} z_{ik} z_{jm} \\
\text{s.t.} \quad \sum_{k \in N} z_{ik} = 1 \quad i \in N \\
\quad \quad z_{ik} \leq z_{kk} \quad i, k \in N, i \neq k \\
\quad \quad \sum_{k \in N} z_{kk} \leq p \\
\quad \quad \sum_{i \neq k} d_i z_{ik} \leq \bar{b}_k z_{kk} \quad k \in N \\
\quad \quad z_{ik} \in \{0,1\} \quad i, k \in N,
\]

where \( \bar{b}_k = (b_k - d_k) \). The three terms in the objective function (1) capture the (linear) setup cost of opening facilities, the (linear) assignment cost of customers to facilities, and the (quadratic) interaction cost between facilities and customer assignments. Constraint sets (2) and (3) ensure that each node is assigned to exactly one facility while constraints (4) ensure that customers are assigned to open facilities. Constraint (4) states that at most \( p \) facilities can be opened. Finally,
constraints (5) are the capacity restrictions on the amount of demand that can be served by each opened facility. Note that these constraints take into account that if facility \( k \) is opened, the demand of node \( k \) will always be served by such facility. The non-separable, non-convex quadratic term in the objective function in combination with the capacity and single assignment constraints make QCpLP a challenging problem to solve. In the following section, we describe the procedure we followed to linearize the quadratic term in order to obtain tight linear MIP reformulations for the QCpLP.

4. RLT-based Reformulations

A standard strategy (Fortet 1959, Glover and Woolsey 1974) to linearizing the bilinear terms \( z_{ik}z_{jm} \) of the objective function (1) is to replace them by new continuous variables \( x_{ikjm} \). The nonlinear relation \( x_{ikjm} = z_{ik}z_{jm} \) is then imposed via the following sets of linear constraints:

\[
\begin{align*}
  x_{ikjm} &\leq z_{ik} & i, j, k, m &\in N, i < j \\
  x_{ikjm} &\leq z_{jm} & i, j, k, m &\in N, i < j \\
  x_{ikjm} &\geq z_{ik} + z_{jm} - 1 & i, j, k, m &\in N, i < j.
\end{align*}
\]

Several improved linearization strategies have been proposed for reformulating zero-one quadratic programs as equivalent linear mixed-integer programs, such as those described in Glover (1975), Adams and Sherali (1986), Sherali and Smith (2007), Liberti (2007), and Caprara (2008), among others. In particular, the linearization strategy introduced in Adams and Sherali (1986) is known to produce tight reformulations and has been generalized and enhanced to design the so-called reformulation linearization technique RLT (Sherali and Adams 1990, 2013). The RLT is a powerful technique whose \( n \)-th hierarchy produces the convex hull of the mixed integer solutions at the expense of having polynomial terms in the resulting reformulation.

In what follows, we apply the level-1 RLT to QF in order to obtain tight linear MIP reformulations for the QCpLP. In the next section, we show how the additional set of continuous variables \( x_{ikjm} \) required in a reduced reformulation can be projected out using Benders decomposition to obtain an equivalent linear MIP reformulation for the QCpLP in the original space of the \( z_{ik} \) variables.

The level-1 RLT applied to QF consists of the following steps:

(i.) Form \( n^3 \) constraints by multiplying the \( n \) equality constraints (2) by each \( z_{jm}, j, m \in N \):

\[
\sum_{k \in N} z_{ik}z_{jm} = z_{jm} \quad i, j, m \in N.
\]
(ii.) Form $n^3(n-1) + n^2 + n^3$ constraints by multiplying the $n(n-1) + 1 + n$ inequality constraints (3), (4), (5) and the bound constraints by each $z_{jm}, j, m \in N$:

$z_{ik}z_{jm} \leq z_{kk}z_{jm}$ \hspace{1cm} i, j, k, m \in N, i \neq k

$\sum_{k \in N} z_{kk}z_{jm} \leq pz_{jm}$ \hspace{1cm} j, m \in N

$\sum_{i \neq k} d_iz_{ik}z_{jm} \leq \bar{b}_kz_{kk}z_{jm}$ \hspace{1cm} j, k, m \in N.

$z_{ik}z_{jm} \leq z_{jm}$ \hspace{1cm} i, j, k, m \in N.

$0 \leq z_{ik}z_{jm}$ \hspace{1cm} i, j, k, m \in N.

(iii.) Form $n^3(n-1) + n^2 + n^3$ constraints by multiplying the $n(n-1) + 1 + n$ inequality constraints (3), (4), and (5) by each $(1 - z_{jm})$, $j, m \in N$. Note the bound constraints are omitted as they give meaningless inequalities:

$z_{ik}(1 - z_{jm}) \leq z_{kk}(1 - z_{jm})$ \hspace{1cm} i, j, k, m \in N, i \neq k

$\sum_{k \in N} z_{kk}(1 - z_{jm}) \leq p(1 - z_{jm})$ \hspace{1cm} j, m \in N

$\sum_{i \in N} \sum_{i \neq k} d_iz_{ik}(1 - z_{jm}) \leq \bar{b}_kz_{kk}(1 - z_{jm})$ \hspace{1cm} j, k, m \in N.

(iv.) Linearize the above sets of constraints and objective (1) by substituting $x_{ikjm} = z_{ik}z_{jm}$, for each $i, j, k, m \in N, i < j$, and add $x_{ikjm} \geq 0$ to obtain the following linear MIP reformulation:

$$\text{(RL$_1$)} \quad \min \sum_{k \in N} f_kz_{kk} + \sum_{i \in N} \sum_{k \in N} c_{ik}z_{ik} + \sum_{i \in N} \sum_{j \in i < j} \sum_{k \in N} \sum_{m \in N} q_{ikjm}x_{ikjm}$$

s.t. \hspace{1cm} (2) - (6)

$$\sum_{k \in N} x_{ikjm} = z_{jm} \hspace{1cm} i, j, m \in N, i < j \hspace{1cm} (10)$$

$$\sum_{k \in N} x_{jmk} = z_{jm} \hspace{1cm} i, j, m \in N, i > j \hspace{1cm} (11)$$

$$x_{ikjm} \leq x_{kkjm} \hspace{1cm} i, j, k, m \in N, i \neq k, i < j, k < j \hspace{1cm} (12)$$

$$x_{ikjm} \leq x_{jmk} \hspace{1cm} i, j, k, m \in N, i \neq k, i < j, k > j \hspace{1cm} (13)$$

$$x_{jmk} \leq x_{kkjm} \hspace{1cm} i, j, k, m \in N, i \neq k, i > j, k < j \hspace{1cm} (14)$$

$$x_{jmk} \leq x_{jmk} \hspace{1cm} i, j, k, m \in N, i \neq k, i > j, k > j \hspace{1cm} (15)$$

$$x_{ikjm} \leq z_{jm} \hspace{1cm} i, j, k, m \in N, i \neq k, i < j, k < j \hspace{1cm} (16)$$

$$\sum_{k < j} x_{kkjm} + \sum_{k > j} x_{jmk} \leq pz_{jm} \hspace{1cm} j, m \in N \hspace{1cm} (17)$$

$$\sum_{i < j} d_ix_{ikjm} + \sum_{i > j} d_ix_{jmk} \leq \bar{b}_kz_{kkjm} \hspace{1cm} j, k, m \in N, k < j \hspace{1cm} (18)$$
\[
\sum_{i<j} d_i x_{ikjm} + \sum_{i>j} d_i x_{jimk} \leq \tilde{b}_k x_{jmkk} \quad j, k, m \in N, k > j
\] (19)

\[
z_{ik} - x_{ikjm} \leq z_{kk} - x_{kkjm} \quad i, j, k, m \in N, i \neq k, i < j, k < j
\] (20)

\[
z_{ik} - x_{ikjm} \leq z_{kk} - x_{jmkk} \quad i, j, k, m \in N, i \neq k, i < j, k > j
\] (21)

\[
z_{ik} - x_{jimk} \leq z_{kk} - x_{kkjm} \quad i, j, k, m \in N, i \neq k, i > j, k < j
\] (22)

\[
z_{ik} - x_{jimk} \leq z_{kk} - x_{jkmk} \quad i, j, k, m \in N, i \neq k, i > j, k > j
\] (23)

\[
\sum_{k \in N} z_{kk} - \sum_{k < j} x_{kkjm} - \sum_{k > j} x_{jmkk} \leq p (1 - z_{jm}) \quad j, m \in N
\] (24)

\[
\sum_{i \neq k} d_i z_{ik} - \sum_{i < j} d_i x_{ikjm} - \sum_{i > j} d_i x_{jimk} \leq \tilde{b}_k (z_{kk} - x_{kkjm}) \quad j, k, m \in N, k < j
\] (25)

\[
\sum_{i \neq k} d_i z_{ik} - \sum_{i < j} d_i x_{ikjm} - \sum_{i > j} d_i x_{jimk} \leq \tilde{b}_k (z_{kk} - x_{jkmk}) \quad j, k, m \in N, k > j
\] (26)

\[
x_{ikjm} \geq 0 \quad i, j, k, m \in N, i < j.
\] (27)

The RLT formulation of Adams and Sherali (1986) contains the standard linearization constraints (7)-(9), which are obtained after multiplying \((1 - z_{jm})\), \(j, m \in N\), by each bound constraint \(0 \leq z_{ik} \leq 1\), \(i, k \in N\). However, these constraints are implied by constraints (10)-(11) and thus, there is no need to add them to \(RL_1\) (Adams and Sherali, 1986). Moreover, an important consequence of this implication is that constraints (10)-(11) are sufficient to provide a valid linear MIP formulation of QCpLP. That is, constraints (12)-(26), although useful for strengthening the linear programming (LP) relaxation, are actually redundant for the description of the set of feasible solutions to QCpLP in the extended space of the \((z, x)\) variables. Therefore, the QCpLP can be stated as the following reduced linear MIP:

\[
(RL_2) \quad \min \sum_{k \in N} f_k z_{kk} + \sum_{i \in N} \sum_{k \in N} c_{ik} z_{ik} + \sum_{i \in N} \sum_{j < k} \sum_{k < m} \sum_{m \in N} q_{ikjm} x_{ikjm}
\]

s.t. (2) - (11), (13), (11), (27).

Although both reformulations \(RL_1\) and \(RL_2\) contain \(n^2\) binary variables and \(n^3(n-1)/2\) continuous variables, the number of additional constraints required in \(RL_1\) is about \(n^3(n-1)+3n^2(n-1)/2+2n^2\), whereas in \(RL_2\) is only \(n^2(n-1)/2\).

We next provide the results of computational experiments to compare the quality of the LP relaxation bounds obtained with \(RL_1\) and \(RL_2\), as well as with six additional relaxations, denoted as \(RL_3, \ldots, RL_8\), to assess the marginal contribution to the improvement of the LP bounds when adding specific subsets of constraints of \(RL_1\) independently. Table II provides the configurations of the considered relaxations of the RLT-based reformulation.

Tables 2 and 3 provide a comparison of the obtained \%LP gap and computational time (in seconds) needed to solve the LP relaxation, respectively, of each of the considered relaxations. For
Table 1  Configuration of considered relaxations of the RLT-based reformulation.

| Formulation | Combination of RLT inequalities |
|-------------|---------------------------------|
| $RL_1$      | Full RLT reformulation          |
| $RL_2$      | Only RLT for assignment constraints: (10)–(11) |
| $RL_3$      | $RL_2$ + RLT for linking constraints: (12)–(15) |
| $RL_4$      | $RL_2$ + RLT for linking constraints: (19)–(22) |
| $RL_5$      | $RL_2$ + RLT for all linking constraints: (12)–(15), (19)–(22) |
| $RL_6$      | $RL_2$ + RLT for capacity constraints: (17)–(18) |
| $RL_7$      | $RL_2$ + RLT for capacity constraints: (24)–(25) |
| $RL_8$      | $RL_2$ + RLT for all capacity constraints: (17)–(18), (24)–(25) |

From Tables 2 and 3, we note that although there is a clear improvement in the quality of the LP bound when solving $RL_1$, the substantial increase in the CPU time required to solve the associated
LP when compared to $RL_2$, does not justify its use. The same is true for the rest of the relaxations ($RL_3$ to $RL_8$). Moreover, adding constraints $[12]$--$[26]$ breaks the decomposability and the network flow structure of the separation problem associated with the Benders reformulation of $RL_2$. For these reasons, we next present a Benders reformulation only for $RL_2$.

5. Benders Reformulation

Benders decomposition is a well-known partitioning method applicable to MILPs (Benders 1962). In particular, it reformulates an MILP by projecting out a set of complicating continuous variables to obtain a formulation with fewer variables but typically with a huge number of constraints, which can be separated efficiently via the solution to an LP subproblem, known as the dual subproblem (DSP). These new constraints are usually referred to as Benders cuts and involve only the variables kept in the reduced problem, plus one additional continuous variable. Given that only a small subset of these constraints are usually active in an optimal solution, a natural relaxation is obtained by dropping most of them and generating them on the fly as needed within a cutting plane algorithm.

Let $Z$ denote the set of vectors $z$ satisfying constraints $[2]$--$[6]$. For any fixed $\bar{z} \in Z$, the primal subproblem (PS) in the space of the $x$ variables is

$$
\begin{align*}
\text{(PS)} & \quad \min & & \sum_{i \in N} \sum_{j > i} \sum_{k \in N} \sum_{m \in N} q_{ikjm} x_{ikjm} \\
& \quad \text{s.t.} & & \sum_{k \in N} x_{ikjm} = \bar{z}_{jm} \quad i, j, m \in N, i < j \tag{28} \\
& & & \sum_{m \in N} x_{ikjm} = \bar{z}_{ik} \quad i, j, k \in N, i < j \tag{29} \\
& & & x_{ikjm} \geq 0 \quad i, j, k, m \in N, i < j. \tag{30}
\end{align*}
$$

Note that the indices $i, j$ and $k, m$ of constraints $[11]$ have been swapped in $[29]$ to better highlight the decomposability of the problem, where subproblems consist of several independent transportation problems. In particular, PS can be decomposed into $n(n - 1)/2$ problems $PS_{ij}$, one for each pair $(i, j) \in N \times N, i < j$. Therefore, we can construct the corresponding dual subproblem (DS$_{ij}$) for each $(i, j)$. Let $\alpha$ and $\beta$ denote the dual variables of constraints $[28]$ and $[29]$, respectively. For each $(i, j) \in N \times N, i < j$, the corresponding DS$_{ij}$ can be stated as follows:

$$
\begin{align*}
\text{(DS$_{ij}$)} & \quad \maximize & & \sum_{m \in N} \bar{z}_{jm} \alpha_{ijm} + \sum_{k \in N} \bar{z}_{ik} \beta_{ijk} \tag{31} \\
& \quad \text{s.t.} & & \alpha_{ijm} + \beta_{ijk} \leq q_{ikjm} \quad k, m \in N. \tag{32}
\end{align*}
$$

Given the network flow structure of each $PS_{ij}$, a sufficient condition for feasibility to $PS_{ij}$ is $
\sum_{k \in N} \bar{z}_{ik} = \sum_{m \in N} \bar{z}_{jm}$, for each $(i, j) \in N \times N, i < j$. This is actually guaranteed by constraints $[2]$ of $Z$ and as a consequence, $PS_{ij}$ is always feasible. Therefore, DS$_{ij}$ has always at least one optimal
solution with a finite objective value. In turn, this implies there is no need to incorporate feasibility cuts to the resulting Benders reformulation. For \( i, j \in \mathbb{N}, i < j \), let \( EP_{ij} \) denote the set of extreme points of the polyhedron of \( DS_{ij} \). The optimal value of each \( DS_{ij} \) is then equal to

\[
\max_{(\alpha, \beta) \in EP_{ij}} \sum_{m \in \mathbb{N}} \bar{z}_{jm} \alpha_{ijm} + \sum_{k \in \mathbb{N}} \bar{z}_{ik} \beta_{ijk}.
\]

Introducing the extra continuous variable \( \eta \) for the overall interaction cost, the Benders reformulation (BR) associated with \( RL_2 \) is

\[
\begin{align*}
(BR) \quad & \min \sum_{k \in \mathbb{N}} f_k z_{kk} + \sum_{i \in \mathbb{N}} \sum_{k \in \mathbb{N}} c_{ik} z_{ik} + \eta \\
\text{subject to} \quad & (2) - (5) \\
\eta \geq \sum_{i \in \mathbb{N}} \sum_{k \in \mathbb{N}} \left( \sum_{j<i} \alpha_{jik} + \sum_{j>i} \beta_{ijk} \right) z_{ik} \quad (\alpha, \beta) \in EP,
\end{align*}
\]

where \( EP \) is the Cartesian product of the sets of extreme points \( EP_{ij}, i, j \in \mathbb{N}, i < j \) and constraints (34) are the so-called Benders optimality cuts. We note that BR contains only the original variables \( z \) and one additional continuous variable.

Whereas it has been empirically shown (Magnanti and Wong 1981) that the disaggregated form of (34), with one for each subproblem, may lead to an improved performance of the Benders decomposition algorithm for some classes of problems, our preliminary computational experiments showed that not to be the case for our problem, since the need to add \( n(n-1)/2 \) constraints at every iteration leads to a problem that increases too rapidly in size, especially for medium to large-size instances with \( n > 200 \). As shown in the next section, the proposed algorithm does however exploit the decomposability of the dual subproblem leading to a reduction of the required CPU time to prove optimality.

6. An Exact Algorithm for QCpLP

In this section, we present an exact branch-and-cut algorithm based on BR to solve QCpLPs. The standard Benders decomposition algorithm is an iterative procedure in which at every iteration a relaxed integer master problem, containing only a small subset of Benders cuts, is optimally solved to obtain a dual bound. The dual subproblem is then solved to obtain a primal bound and to determine whether additional Benders cuts are needed in the relaxed master problem. If needed, these cuts are added to the master problem and solved again. This iterative procedure is repeated until the convergence of the bounds is attained, if an optimal solution exists. One of the major drawbacks of this standard approach is the need to solve an integer master problem at each iteration. To overcome this difficulty, recent implementations of Benders decomposition
have considered the solution of the Benders reformulation with a standard branch-and-cut framework, in which Benders cuts are separated not only at nodes with integer solutions but also at the nodes with fractional solutions of a single enumeration tree (see, for instance Fischetti et al. 2017, Ortiz-Astorquiza et al. 2019, Zetina et al. 2019b). We use this approach to develop an exact algorithm for QCpLP.

In addition, we use the following strategies to speed up the convergence of our branch-and-cut algorithm: i) we exploit the structure of the subproblem to generate Pareto-optimal cuts by efficiently solving network flow problems, ii) we use a simple but effective stabilization procedure for generating cuts, iii) we use a matheuristic to generate high quality solutions, iv) we apply variable elimination tests at the root node, and v) we perform a partial enumeration phase to permanently fix location variables $z_{kk}$ to either one or zero before exploring the enumeration tree.

6.1. Stabilized Pareto-optimal Cuts

It is well known that the selection of Benders cuts plays an important role in the overall convergence of Benders decomposition algorithms. Magnanti and Wong (1981) proposed a procedure for obtaining Pareto-optimal cuts, that is, cuts that are not dominated by any other cut. In particular, Magnanti and Wong (1981) define the notion of cut dominance as follows. Given two cuts defined by dual solutions $u$ and $u'$ of the form $\theta \geq f(u) + zg(u)$ and $\theta \geq f(u') + zg(u')$, respectively, the cut defined by $u$ dominates the cut defined by $u'$ if and only if $f(u) + zg(u) \geq f(u') + zg(u')$ with strict inequality holding for at least one feasible point $z$ of MP. If a cut defined by $u$ is not dominated by any other cut, then it is a Pareto-optimal Benders cut.

To obtain Pareto-optimal cuts, Magnanti and Wong (1981) propose solving an additional linear program similar to DS. It is parameterized by a core point of the set $Z$, which is a point in the relative interior of its convex hull. Let $z^0$, $\bar{z}$, and $\Gamma(\bar{z})$ denote a given core point, the current (possibly fractional) solution, and the optimal solution value of $DS_{ij}$, respectively. To determine whether there exists a Pareto-optimal cut or not that is violated by the point $\bar{z}$, for each $i, j \in N, i < j$, we solve the following Pareto-optimal subproblem:

$$ (DPO_{ij}) \quad \text{maximize} \quad \sum_{m \in N} z^0_{jm} \alpha_{ijm} + \sum_{k \in N} z^0_{ik} \beta_{ijk} $$

$$ \text{s.t.} \quad \sum_{m \in N} \bar{z}_{jm} \alpha_{ijm} + \sum_{k \in N} \bar{z}_{ik} \beta_{ijk} \geq \Gamma(\bar{z}) $$

$$ \alpha_{ijk} + \beta_{ijm} \leq q_{ikjm} \quad k, m \in N, $$

where constraints (36) guarantee that the optimal solution to $DPO_{ij}$ is contained in the set of optimal solutions to the original $DP_{ij}$. 
Let $\delta$ be the dual variable associated with constraints \((36)\) and $x_{ikjm}$ be those of constraints \((37)\). Dualizing DPO$_{ij}$ we obtain the following linear program:

\[
(PPO_{ij}) \quad \min \sum_{k \in N} \sum_{m \in N} q_{ikjm} x_{ikjm} - \Gamma(\bar{z}) \delta 
\]

s.t. \[
\sum_{k \in N} x_{ikjm} = z_{jm}^0 + \bar{z}_{jm} \delta 
\]

\[
\sum_{m \in N} x_{ikjm} = z_{ik}^0 + \bar{z}_{ik} \delta 
\]

\[
x_{ikjm} \geq 0 \quad k, m \in N 
\]

\[
\delta ij \geq 0. 
\]

Given that $\delta$ affects the right-hand-side of flow constraints \((39)\) and \((40)\), this problem corresponds to a parametric transportation problem. Given that $\sum_{m \in N} \bar{z}_{jm} = \sum_{k \in N} \bar{z}_{ik} = 1$, PPO$_{ij}$ can be interpreted as a problem where a rebate of $\Gamma$ is given for each unit of additional supply/demand shipped. Similar to the Pareto-optimal problem presented in Magnanti et al. (1986) for uncapacitated multi-commodity network design problems, rather than performing a parametric analysis on PPO$_{ij}$ to determine its optimal solution, we use the following result.

**Proposition 1.** Any value of $\delta \geq 1$ is optimal to PPO$_{ij}$.

Proof The total demand $\sum_{k \in N} (z_{ik}^0 + \bar{z}_{ik} \delta)$ must flow via some subset of arcs $(k, m) \in N \times N$. At most $\sum_{k \in N} z_{ik}^0 = 1$ of this flow can go through demand nodes $j \in N$ having $\bar{z}_{ik} = 0$. Any additional flow must use arcs incident to demand nodes $j \in N$ such that $\bar{z}_{ik} > 0$, and thus, the marginal flow cost will be precisely $\Gamma(\bar{z})$. Therefore, any value of $\delta \geq 1$ must be optimal for PPO$_{ij}$.

An important consequence of this result is that there is no longer a need to know the value $\Gamma(\bar{z})$ to select $\delta$. Therefore, the complexity for generating a Pareto-optimal cut is the same as generating a standard Benders optimality cut, i.e., solving $n(n-1)/2$ transportation problems, which can be efficiently solved by using the network simplex algorithm.

In our algorithm, we consider the following family of core points of $Z$ for $p \geq 2$:

\[
z_{ik}^0(\bar{H}) = \begin{cases} 
\frac{p}{|\bar{H}|} - \epsilon & i \in \bar{H}, k \in \bar{H}, i = k, \\
1 - \frac{p - \epsilon}{|\bar{H}|} & i \in \bar{H}, k \in \bar{H}, i \neq k, \\
\frac{1}{|\bar{H}|} & i \in N \setminus \bar{H}, k \in \bar{H}, i \neq k,
\end{cases}
\]

where $0 < \epsilon < \min \left\{ \frac{1}{|\bar{H}|}, \frac{p-1}{|\bar{H}|-2} \right\}$, and $\bar{H} \subseteq N$ denote the current set of candidate facilities to open at a given node of the enumeration tree.

Although the solution to the Pareto-optimal subproblem PPO$_{ij}$ guarantees that the obtained cut will be non-dominated, in practice some non-dominated cuts may be more useful than others.
That is, a core point selection strategy is needed to further improve the convergence of the cutting plane algorithm used to solve the LPs at the nodes of the enumeration tree. Ortiz-Astorquiza et al. (2019), Zetina et al. (2019a), and Zetina et al. (2019b) present and computationally compare several static and dynamic core point selection strategies to separate Pareto-optimal cuts for multi-level facility location and multi-commodity network design problems. An interesting observation from these studies is that the strategy providing on average the best results is different in each of these works. These strategies can actually be seen as stabilization procedures for generating effective non-dominated cuts. Stabilization is frequently used to improve the convergence of column generation and cutting plane algorithms needed to solve Dantzig-Wolfe decompositions and Lagrangean relaxation [Ben-Ameur and Neto 2007, Fischetti at al. 2017, Pessoa at al. 2018].

After performing extensive preliminary experiments, we note that an efficient core point selection strategy is to dynamically update the separation point $\hat{z}'$ at each iteration by considering a convex combination of the separation point $\hat{z}$ considered in the previous iteration and the current point $\bar{z}$ as follows:

$$
\hat{z}' = \phi \hat{z} + (1 - \phi)\bar{z},
$$

where $\phi \in \mathbb{R}$, and $0 < \phi < 1$. We initialize the algorithm by using $\hat{z} = z^0(N)$ at the first iteration of the cutting plane algorithm. Each time the reduction tests of Section 6.3.1 modify the current set of candidate facilities to open $\tilde{H} \subseteq N$, we reinitialize $\hat{z}$ as $z^0(\tilde{H})$.

### 6.2. A Matheuristic for QC$p$LP

When using branch-and-cut algorithms, it is important to find high quality feasible solutions early on in the process. This leads to smaller search trees since they provide better bounds for pruning and a guide for selecting variables to branch on. Finding near-optimal solutions in a preprocessing stage can be used to perform variable elimination tests that reduce the size of the formulation to be solved [Contreras et al. 2011a, Alibeyg et al. 2018].

In this section, we present a matheuristic that exploits the information generated during the solution of the root node to effectively explore the solution space. It consists of two phases. The first is a constructive heuristic in which the support of the LP relaxation of the Benders reformulation is used to build a reduced linear integer relaxation of $QF$, which is then solved with a general purpose solver to construct an initial feasible solution. The second is a local search heuristic in which several neighborhoods are systematically explored to improve the initial solution obtained during the constructive phase. One of such neighborhoods is efficiently explored by solving an MILP with CPLEX. We next describe in detail each of these phases.
6.2.1. Constructive Heuristic

Let $\bar{z}^t$ denote the solution at iteration $t$ of the LP relaxation of the Benders reformulation at the root node of the enumeration tree and let $H(\bar{z}^t) = \{k \in N : \bar{z}_{kk}^t > 0\}$ correspond to the set of (partially) opened hub facilities in $\bar{z}^t$. We can obtain a feasible solution to QCpLP by solving the following reduced (linear) relaxation of SQFLP:

$$\begin{align*}
(\text{RLF}) \quad \min & \quad \sum_{k \in H(\bar{z}^t)} f_k z_{kk} + \sum_{i \in N} \sum_{k \in H(\bar{z}^t)} c_{ik} z_{ik} \\
\text{s.t.} & \quad \sum_{k \in H(\bar{z}^t)} z_{ik} = 1 \quad i \in N \\
& \quad z_{ik} \leq z_{kk} \quad i \in N, k \in H(\bar{z}^t), i \neq k \\
& \quad \sum_{k \in H(\bar{z}^t)} z_{kk} \leq p \\
& \quad \sum_{i \neq k} d_i z_{ik} \leq \bar{b}_k z_{kk} \quad k \in H(\bar{z}^t) \\
& \quad z_{ik} \in \{0, 1\} \quad i \in N, k \in H(\bar{z}^t).
\end{align*}$$

Note that the quadratic term of objective function (1) of QF has been relaxed. Whenever $|H(\bar{z}^t)| \ll |N|$, RLF will be substantially smaller than QF. Given that any feasible solution to (45)-(49) is also feasible for (2)-(6), we can compute a valid upper bound on the optimal solution of QCpLP by simply evaluating objective (1) using the optimal solution of RLF.

Noting that any instance of the RLF can be transformed into an instance of the well-known single-source capacitated facility location problem, one can use the state-of-the-art exact algorithm given in Gadegaard et al. (2018) to solve RLF. However, we solve RLF by using a general purpose solver given that the time spent in solving RLF is negligible when compared to the total CPU time needed by our branch-and-cut algorithm. Finally, we only solve RLF when the support of the LP relaxation of the Benders reformulation changes from one iteration to the next at the root node, i.e., whenever $H(\bar{z}^{t-1}) \neq H(\bar{z}^t)$.

6.2.2. Local Search Heuristic

The local search heuristic is used to improve the initial solution obtained from the constructive heuristic. It consists of two phases. The first phase, denoted as the facility modification phase, uses a variable neighborhood descent (VND) method to systematically explore five neighborhoods that modify both the set of hubs and assignment decisions. The second phase, denoted as the assignment modification phase, solves a well-known MILP as an approximation to improve assignment decisions.

In what follows, solutions are represented by pairs of the form $s = (H, a)$ where $H \subseteq N$ denotes the set of opened facilities and $a : N \rightarrow H$ is the assignment mapping, i.e., $a(i) = k$ if customer $i \in N$ is assigned to facility $k \in H$. For any feasible assignment, $h_k$ denotes the available capacity of facility $k$, i.e., $h_k = b_k - \sum_{i : a(i) = k} d_i$. 
During the facility modification phase, we use a VND method (Brimberg and Mladenovic 1996). It is based on a systematic search in a set of \( r \) neighborhoods, \( N_1, N_2, \ldots, N_r \). The VND works by performing a local search in a neighborhood \( N_1 \) until a local optimal solution is found. After that, the algorithm switches to neighborhoods \( N_2, \ldots, N_r \), sequentially, until an improved solution is found. Each time the search improves the best known solution, the procedure restarts using the neighborhood \( N_1 \). Our implementation of the VND algorithm explores two types of neighborhood structures. The first type focuses on the reassignment of customers to open facilities, whereas the second type allows the set of open facilities to change. In all cases, we only consider movements that are feasible with respect to constraints (2)-(6).

The *shift* neighborhood considers all solutions that can be obtained from the current one by changing the assignment of exactly one node, whereas the *swap* neighborhood contains all solutions that differ from the current one in the assignment of two nodes. Let \( s = (H, a) \) be the current solution, then

\[
N_1(s) = \{ s' = (H, a') : \exists i \in N, a'(i) \neq a(i) \},
\]

and

\[
N_2(s) = \{ s' = (H, a') : \exists i_1, i_2, a'(i_1) = a(i_2), a'(i_2) = a(i_1), a'(i) = a(i), \forall i \neq i_1, i_2 \}.
\]

For exploring \( N_1 \), we consider all pairs of the form \((i, j)\) where \( a(j) \neq i \) and \( h_i \geq d_j \). Also, for exploring \( N_2 \) we consider all pairs of the form \((i_1, i_2)\) where \( a(i_1) \neq a(i_2) \), \( h_{a(i_1)} + d_{i_1} \geq d_{i_2} \) and \( h_{a(i_2)} + d_{i_2} \geq d_{i_1} \). In both cases we perform the first improving move.

We explore three additional neighborhood structures of the second type. They affect the current set of open facilities. The first one considers a subset of feasible solutions that are obtained from the current one by opening a new facility and by assigning some customers to it. That is,

\[
N_3(s) \subseteq \left\{ s' = (H', a') : H' = H \cup \{ k \}; \forall j, a'(j) = r \in H', \sum_{j : a'(j) = r} d_j \leq b_r, \forall r \in H' \right\}.
\]

To explore \( N_3(s) \), all nodes \( k \in N \setminus H \) are considered. Again, let \( s = (H, a) \) denote the current solution, \( a'(k) \) the new assignment and \( \hat{h}_r \) the available capacity of hub \( r \). Initially, \( a'(p) = a(p) \) for all \( p \in N \) and \( \hat{h}_r = h_r \) for all \( r \in H \). For each potential facility \( k \in N \setminus H \), we consider nodes by decreasing order of their demand \( d_i \). Node \( j \) is reassigned to facility \( k \) if \( c_{jk} \leq c_{ja(i)} \) and \( \hat{h}_k \geq d_j \). If node \( j \) is reassigned to facility \( k \) we update its assignment and the available capacity of facilities \( k \) and \( a(j) \).

The second neighborhood structure of the second type considers a subset of feasible solutions that are obtained from the current one by closing a facility and reassigning its assigned customers to other open facilities. That is,

\[
N_4(s) \subseteq \left\{ s' = (H', a') : H' = H \setminus k; \forall j, a'(j) = r \in H', \sum_{j : a'(j) = r} d_j \leq b_r, \forall r \in H' \right\}.
\]
To explore $N_4(s)$, all facilities $k \in H$ are considered. Once more, let $a'(k)$ denote the new assignment and $\hat{h}_r$ the available capacity of facility $r$. Initially, $a'(p) = a(p)$ for all $p \in N$ and $\hat{h}_r = h_r$, for all $r \in H$. For each open facility $k \in H$, we consider its assigned customers in decreasing order of their demand $d_i$. Customer $j$ is reassigned to facility $\hat{m}$, where $\hat{m} = \arg\min \left\{ c_{jm} : \hat{h}_m - d_j \geq 0, m \in H \setminus k \right\}$. If node $j$ is reassigned to facility $\hat{m}$, we update its assignment and the available capacity of facilities $\hat{m}$ and $a(j)$.

The last neighborhood of the second type considers a subset of feasible solutions that are obtained by simultaneously closing an open facility and opening a closed one. That is,

$$N_5(s) \subset \left\{ s' = (H', a') : H' = H \setminus \{m\} \cup \{i\}, m \in H, i \in N \setminus H \right\}.$$ 

To explore $N_5(s)$, all nodes $i \in N \setminus H$ are considered, and a set of solutions is obtained from the current one by interchanging an open facility by a closed one while ensuring capacity is enough to meet demand, and reassigning all the customers to their closest open facility with available capacity.

Once the VND search terminates with a local optimal solution with respect to the above five neighborhoods, we proceed with the assignment modification phase. The objective of this second step is to intensify the search on the assignment decisions by solving a well-known combinatorial optimization problem that arises as a subproblem in the QCpLP. In particular, when the location of the facilities is given, the QCpLP reduces to a capacitated variant of the quadratic semi-assignment problem (Saito et al. 2009, Rostami et al. 2018). Moreover, if we relax the quadratic term of (1) for each set of open facilities $H \subseteq N$, we obtain the following generalized assignment problem (GAP):

$$\begin{align*}
\text{(GAP)} \quad \text{min} & \quad \sum_{i \in N} \sum_{k \in H} c_{ik} z_{ik} \\
\text{s.t.} & \quad \sum_{k \in H} z_{ik} = 1, \quad i \in N \quad (50) \\
& \quad \sum_{i \neq k} d_i z_{ik} \leq \bar{b}_k z_{kk}, \quad k \in H \quad (51) \\
& \quad z_{ik} \in \{0, 1\}, \quad i \in N, k \in H. \quad (52)
\end{align*}$$

The solution to the GAP can be seen as an intensification mechanism in which we focus on improving the assignment decisions for all customers simultaneously, while taking explicitly into account the capacity restrictions at facilities but relaxing the quadratic term in the objective. In case the solution to the GAP for a given set $H$ obtained from the local optimal solution of the first step modifies some customers’ assignments, we perform a simple VNS on the obtained solution in which only the shift and swap neighborhoods are explored. Similar to the MILP solved in the
constructive phase, one can use the state-of-the-art exact algorithm given in [Avella et al. (2010)] to solve GAP. However, we solve GAP by using a general purpose solver given that the time spent in solving it is negligible compared to the total CPU time needed by our branch-and-cut algorithm.

6.3. Model Reduction Techniques

When solving large-scale optimization problems, reducing the size of the formulation without compromising optimality is crucial to reduce the computational time. Having a high quality feasible solution suggests the use of elimination tests and variable fixing procedures to significantly reduce the size of the formulation to be solved in the branch-and-cut algorithm.

6.3.1. Elimination Tests

First proposed in [Crowder and Padberg (1980)], elimination tests have been shown to be an effective tool for identifying variables that will not be in the optimal solution and can therefore be discarded. It uses linear programming principles to conclude that a chosen variable will have a value of zero at an optimal solution. The procedure is based on the following well-known proposition written in terms of our Benders reformulation \( BR \).

**Proposition 2.** Let \( UB \) and \( LB \) be an upper bound and an LP relaxation-based lower bound to \( BR \), respectively, and \( r_{ck} \) be the reduced cost coefficient of \( z_{kk} \), \( k \in N \), at an optimal solution of any linear relaxation of \( BR \) in which only a subset of Benders cuts is included. If \( LB + r_{ck} > UB \), then \( z_{kk} = 0 \) at an optimal solution to \( BR \).

This proposition allows the elimination of a variable from the formulation without compromising on optimality. By verifying the condition for all non-basic location variables every time new constraints are added to the LP relaxation of \( BR \), a significant number of variables may be eliminated. Note that we focus only on the location variables \( z_{kk} \), \( k \in H \) since the impact of removing it will lead to the elimination of all other related variables, \( z_{ik} \), \( \forall i \in N \), thereby reducing the size of \( BR \) by \( |N| \).

6.3.2. Partial Enumeration

Similar to the idea of strong branching ([Applegate et al. 1995]), partial enumeration creates a set of what-if scenarios to measure the impact that fixing a variable \( z_{kk} \), \( k \in N \) to an arbitrary value \( \pi \) has on the LP of the formulation. If the optimal solution value of the linear program with fixed variable \( z_{kk} \) is larger than that of a known upper bound, then that variable will not have a value of \( \pi \) at an optimal solution to \( BR \). This method is particularly useful when the variable can only take two possible values as is the case with the binary location variables.

Since there are two possible what-if scenarios, one can create for each variable in the partial enumeration scheme, a simple rule to identify good candidates. If for a given location variable \( z_{kk} \) \( k \in N \) its value is less than or equal to 0.2, we then solve a linear programming problem with the
added constraint $z_{kk} = 1$. If the resulting optimal solution value is greater than the best upper bound known, we can then make $z_{kk}$ and its corresponding assignment variables equal to 0 (i.e., remove them from the formulation). We refer to this procedure as PE$_0$, which stands for partial enumeration to fix at 0.

If the value of $z_{kk}$ is greater than or equal to 0.8, we then solve a linear programming problem with the added constraint $z_{kk} = 0$. If the resulting optimal solution value is greater than the best upper bound known, we then fix $z_{kk} = 1$. We refer to this procedure as PE$_1$, which stands for partial enumeration to fix at 1.

Both the elimination test and partial enumeration procedure are performed as a part of a pre-processing phase in which the linear relaxation of BR is solved. This reduces the size of the mixed binary formulation used in the branch-and-cut algorithm, and of the modified transportation problems PPO$_{ij}$ solved to obtain Benders cuts.

### 6.4. Some Implementation Details of the Branch-and-Cut Algorithm for QC$_p$LP

Having explained each of the algorithmic enhancements of our procedure, we now present the complete exact algorithm which is divided into two parts. The first is the root-processing routine, in which the linear programming relaxation of BR is solved and variables are dynamically eliminated and fixed. The second part is the branch-and-cut algorithm where the mixed binary program is solved to proven optimality. The transportation problem PPO$_{ij}$ is used to separate new Benders cuts at every node of the tree at a depth of multiples of $\gamma$.

The preprocessing phase is an iterative procedure in which the LP relaxation of BR without any Benders cuts is first solved. The mathheuristic is then called using the support (set of variables with non-zero solution value) of the LP relaxation to obtain a feasible solution and to potentially update the incumbent. We then perform the variable elimination procedure and, if the iteration number is divisible by $\phi$, we also perform the partial enumeration to close facilities (PE$_0$). We then proceed to generate a Benders cut by solving our modified transportation problem PPO$_{ij}$ for each $i, j \in N$ and add it to BR if the cut is violated by at least $\epsilon$. We then reoptimize the BR with its newly added constraint. If the resulting optimal solution value is larger than the previous by a margin of at least $\kappa$, the variable elimination procedure is executed again and the entire process is repeated. If the improvement is less than $\kappa$, we then execute the partial enumeration methods PE$_1$ followed by PE$_0$. The resulting formulation is then used in the mixed binary programming phase.

Upon completing the preprocessing phase, the resulting mixed binary program BR is of a significantly smaller size than the original formulation. The reduced BR along with the previously generated Benders cuts are then inputted into a solver that executes a branch-and-cut process and infer additional general mixed integer cuts to strengthen the formulation. Up to $\Upsilon$ additional
Benders cuts are obtained when exploring nodes at a depth of multiples of $\gamma$ of the branch-and-bound tree and are added only if the current fractional solution violates them by at least $\epsilon$. For correctness, the tolerance to add Benders cuts at integer solutions is set to 0. Finally, we also impose that in the branching framework, priority be given to location variables $z_{kk}$, to ensure the impact of opening and closing facilities is evaluated first. The source code can be downloaded from https://sites.google.com/view/carloszetina/Research/Publications.

7. Computational Experiments

We next present the results of extensive computational experiments performed to assess the performance of our branch-and-cut algorithm. Given that the star-star hub network design problem is a special case of hub location problems and that the instances are not publicly available, our experiments are based on the well-studied, well-benchmarked single-allocation hub location problems. In the first part of the experiments, we focus on a comparison between our exact algorithm and other exact algorithms reported in the literature for four particular cases of QCpLP arising in hub location: the uncapacitated hub location problem with single assignments (UHLPSA), the uncapacitated $p$-hub median problem with single assignments (UpHMPSA), the capacitated hub location problem with single assignments (CHLPSA), and the capacitated $p$-hub median problem with single assignments (CpHMPSA). In the second part of the experiments, we test the robustness and limitations of our exact algorithm on large instances involving up to 1,000 nodes. All algorithms were coded in C using the callable library for CPLEX 12.9.0 and run on an Intel Xeon E5 2687W V3 processor at 3.10 GHz with 750 GB of RAM under Linux environment. The separation and addition of Benders optimality cuts within the branch-and-cut algorithm has been implemented via lazy callbacks and user cut callbacks. For a fair comparison, all use of CPLEX was limited to one thread and the traditional MIP search strategy. The following parameter values were, in order of appearance used in our final implementation: $\phi = 0.5$, $\epsilon = 100$, $\kappa = 0.1$, $\Upsilon = 2$, and $\gamma = 10$.

7.1. Comparison with Alternative Solution Algorithms

We now present a comparison between our branch-and-cut algorithm for QCpLP and the most recently proposed exact algorithms for UHLPSA (Meier and Clausen 2018, Ghaffarinasab and Kara 2018), UpHMPSA (Ghaffarinasab and Kara 2018), and CHLPSA (Contreras et al. 2011d, Meier and Clausen 2018). To the best of our knowledge, these are the state-of-the-art algorithms for solving the respecting problems to proven optimality. We also present the results of our algorithm for the CpHMPSA. However, to the best of our knowledge, there is no ad hoc exact algorithm in the literature for the CpHMPSA. All computational experiments in this section are performed using the well-known Australian Post (AP) set of instances. It consists of the Euclidean distances $d_{ij}$ between 200 postal districts in Sydney, Australia, and of the values of
$w_{ij}$ representing postal flows between pairs of postal districts. From this set of instances, we have selected those with $|N| \in \{100, 125, 150, 175, 200\}$, $p \in \{5, 10, 15, 20\}$ and with setup costs and capacities of the types loose (L) and tight (T) (see Contreras et al., 2011d). We can represent any instance of the above mentioned hub location problems as an instance of the QCpLP by setting the costs in the objective as $c_{ik} = (\chi O_i + \delta D_i) d_{ik}$ and $q_{ikjm} = \tau w_{ij} d_{km}$, where $O_i = \sum_{j \in N} w_{ij}$, $D_i = \sum_{j \in N} w_{ji}$, and $\chi$, $\tau$, $\delta$ represent the unit collection, transfer, and distribution costs, respectively. For the AP data set, these unit costs are set to $\chi = 2$, $\tau = 0.75$, and $\delta = 3$.

Our comparisons are done with Meier and Clausen (2018), Ghaffarinasab and Kara (2018), and Contreras et al. (2011d) which are, to the best of our knowledge, the state of the art ad hoc exact solvers for their respective problem variants. Meier and Clausen (2018) use C# to call Gurobi 6.0 running on a processor at 3.4 GHz while Ghaffarinasab and Kara (2018) use Java to call CPLEX 12.6 on a processor at 3.30 GHz. Finally, Contreras et al. (2011d) use C to develop a branch-and-price solver run on a processor at 2.33 GHz.

Our first comparison is for the UHLPSA where there exist fixed set-up costs for locating facilities but both capacity and cardinality constraints are relaxed. We compare our algorithm to the results presented in Meier and Clausen (2018) and Ghaffarinasab and Kara (2018). The detailed results of the comparison between the exact methods using the AP data set are provided in Table 4. The first two columns are the instance name and optimal objective function value. These are then followed by the time in seconds reported by Ghaffarinasab and Kara (2018) and Meier and Clausen (2018) and the information of our algorithm’s performance. The column under the heading $\%$Dev heur reports the percent deviation between the best solution obtained with the matheuristic presented in Section 6.2 and the optimal solution. The column $\%$fixed plants gives the percent of facilities that were closed at the end of the partial enumeration phase. The column $\%$time root provides the percent of time spent by the algorithm solving the root node, including the matheuristic, the elimination tests, and the partial enumeration phase. The last column reports the number of nodes explored in the enumeration tree.

The results in Table 4 indicate that our exact algorithm is, on average, an order of magnitude faster than the cutting plane algorithm of Meier and Clausen (2018), which explicitly requires the assumption of Euclidean distances. On the other hand, our algorithm takes on average half of the computation time required by the iterative Benders decomposition algorithm of Ghaffarinasab and Kara (2018). In addition, the matheuristic is capable of finding an optimal solution in 9 out of 10 instances.

Our second comparison is for the UpHMPSA, where the fixed set-up costs for locating facilities are disregarded but a cardinality constraint on the number of open hubs is imposed. Also, capacity limitations on the facilities are not considered. Once more, we compare our algorithm to the
Table 4  Comparison of state-of-the-art exact algorithms for the uncapacitated hub location problem with single assignments using the AP data set.

| Instance | Opt. | Ghaffarinasab and Kara (2018) | Meier and Clausen (2018) | Branch-and-cut algorithm |
|----------|------|------------------------------|--------------------------|--------------------------|
|          |      | time(s)                      | time(s)                  | time(s)                  |
| 100LT    | 238,016.28 | 6.91                        | 82.42                    | 23.76 0.00 97 68 70 |
| 100TT    | 305,097.95 | 3.24                        | 60.23                    | 2.32 1.52 98 68 0    |
| 125LT    | 227,949.00 | 43.30                       | 411.46                   | 42.89 0.00 98 37 122 |
| 125TT    | 258,839.68 | 16.08                       | 188.05                   | 8.54 0.00 98 21 9   |
| 150LT    | 225,450.09 | 107.24                      | 1,259.22                 | 76.05 0.00 99 39 172 |
| 150TT    | 234,778.74 | 26.30                       | 478.38                   | 59.06 0.00 99 5 227  |
| 175LT    | 227,655.38 | 188.17                      | 2,044.77                 | 89.86 0.00 99 70 51  |
| 175TT    | 247,876.80 | 44.55                       | 1,639.21                 | 69.84 0.00 99 8 180  |
| 200LT    | 233,802.98 | 68.18                       | 5,493.47                 | 290.87 0.00 99 65 276 |
| 200TT    | 272,188.11 | 1,399.53                    | 20,292.35                | 102.27 0.46 97 29 136 |

Geom. mean 45.15 782.85 41.96 0.00 98 31 0
Arith. mean 190.35 3,194.96 76.55 0.20 98 41 124

results presented in Meier and Clausen (2018) and Ghaffarinasab and Kara (2018) for solving the UpHMPSA. The detailed results of the comparison between the exact methods using the AP data set are provided in Table 5.

Table 5  Comparison of state-of-the-art exact algorithms for the uncapacitated p-hub median problem with single assignments using the AP data set.

| Instance | p   | Opt. | Ghaffarinasab and Kara (2018) | Meier and Clausen (2018) | Branch-and-cut algorithm |
|----------|-----|------|------------------------------|--------------------------|--------------------------|
|          |     |      | time(s)                      | time(s)                  | time(s)                  |
| 100      | 5   | 136,929.44 | 313.80                       | 356.39                   | 23.26 0.00 86 82 32 |
| 100      | 10  | 106,469.57 | 109.18                       | 32.22                    | 34.33 0.00 82 84 14  |
| 100      | 15  | 90,533.52  | 144.40                       | 85.95                    | 41.92 0.00 67 66 79  |
| 100      | 20  | 80,270.96  | 61.47                        | 33.54                    | 26.35 0.00 72 82 10  |
| 125      | 5   | 137,175.68 | 1,286.48                     | 1,104.31                 | 79.83 0.04 88 58 111 |
| 125      | 10  | 107,092.09 | 414.08                       | 184.62                   | 69.69 0.00 81 89 13  |
| 125      | 15  | 91,494.56  | 1,271.86                     | 465.08                   | 81.34 0.00 69 65 132 |
| 125      | 20  | 81,471.65  | 213.76                       | 111.03                   | 107.45 0.00 67 63 149 |
| 150      | 5   | 137,425.90 | 2,989.83                     | 1,474.76                 | 153.27 0.00 91 72 178 |
| 150      | 10  | 107,478.12 | 1,148.01                     | 412.53                   | 159.79 0.00 92 82 68  |
| 150      | 15  | 92,050.58  | 1,695.15                     | 405.57                   | 256.22 0.03 68 52 142 |
| 150      | 20  | 82,229.39  | 531.99                       | 185.34                   | 195.07 0.00 78 70 114 |
| 175      | 5   | 139,354.51 | 31,347.15                    | 10,699.58                | 535.70 0.00 92 54 342 |
| 175      | 10  | 109,744.35 | 10,551.64                    | 3,023.02                 | 895.67 0.61 57 41 236 |
| 175      | 15  | 94,123.66  | 19,602.93                    | 8,143.73                 | 627.45 0.04 66 42 402 |
| 175      | 20  | 83,843.59  | 1,778.11                     | 271.03                   | 429.19 0.00 79 61 167 |
| 200      | 5   | 140,062.65 | 127,546.79                   | 17,628.38                | 1,156.65 0.07 79 50 367 |
| 200      | 10  | 110,147.66 | 46,706.90                    | 4,957.31                 | 1,302.66 0.00 83 39 1036 |
| 200      | 15  | 94,459.20  | 26,640.56                    | 1,107.48                 | 769.89 0.21 66 52 151  |
| 200      | 20  | 84,955.36  | 27,224.48                    | 526.08                   | 1,141.73 0.00 75 37 1007 |

Geom. mean 2,195.82 613.97 198.46 0.00 76 60 122.88
Arith. mean 15,078.93 2,560.40 404.37 0.05 77 62 237.5
The results in Table 5 show once more that our exact algorithm is on average three and two orders of magnitude faster than those of Ghaffarinasab and Kara (2018) and Meier and Clausen (2018), respectively. The matheuristic is capable of finding an optimal solution in 14 out of 20 instances, and the average deviation is only 0.05%. It is worth noting that the UpHMPSA seems to be more difficult to solve as compared to the UHLPSA. The largest CPU time for the UHLPSA was about six minutes (200 nodes with loose setup costs) whereas for the UpHMPSA was one hour (200 nodes and $p = 20$).

Our third comparison is for the CHLPSA, where the fixed set-up costs and capacities of facilities are taken into account but the cardinality constraint is disregarded. We compare our algorithm to the results presented in Meier and Clausen (2018) using a branch-and-cut algorithm and Contreras et al. (2011d) using a branch-and-price algorithm for solving the CHLPSA. The detailed results of the comparison between the exact methods using the AP data set are provided in Table 6.

| Instance | Opt. | Meier and Clausen (2018) time(s) | Contreras et al. (2011d) time(s) | Branch-and-cut algorithm time(s) | %Dev | %fixed | %time | BB plants | root |
|----------|------|---------------------------------|---------------------------------|---------------------------------|------|--------|-------|----------|------|
| 100LL    | 246,713.97 | 176.31 | 459.89 | 19.11 | 0.00 | 93 | 54 | 68 |
| 100LT    | 256,155.33 | 141.23 | 347.95 | 27.88 | 0.06 | 95 | 29 | 155 |
| 100TL    | 362,950.09 | 174.03 | 124.92 | 40.49 | 0.59 | 92 | 14 | 242 |
| 100TT    | 474,068.96 | 1,122.14 | 328.11 | 95.35 | 0.37 | 93 | 6 | 943 |
| 125LL    | 239,889.33 | 415.41 | 1,650.57 | 45.15 | 0.02 | 93 | 52 | 101 |
| 125LT    | 251,259.16 | 279.53 | 552.99 | 45.11 | 0.02 | 97 | 41 | 299 |
| 125TL    | 246,486.69 | 53.24 | 41.22 | 59.04 | 0.00 | 98 | 4 | 302 |
| 125TT    | 291,807.35 | 109.19 | 322.73 | 27.52 | 0.00 | 97 | 12 | 194 |
| 150LL    | 234,765.44 | 153.85 | 3,347.21 | 53.01 | 0.35 | 97 | 82 | 8 |
| 150LT    | 249,797.49 | time out | 11,818.19 | 84.61 | 0.00 | 92 | 50 | 293 |
| 150TL    | 262,543.08 | 1,419.80 | 1,114.95 | 73.29 | 0.31 | 98 | 13 | 393 |
| 150TT    | 322,976.47 | 3,243.56 | 4,299.28 | 192.74 | 0.00 | 92 | 7 | 1124 |
| 175LL    | 227,997.58 | 1,010.55 | 3,418.10 | 119.05 | 0.00 | 98 | 56 | 150 |
| 175LT    | 251,540.80 | 2,196.45 | 12,408.05 | 131.39 | 0.00 | 95 | 61 | 138 |
| 175TL    | 244,860.41 | 152.07 | 256.60 | 54.48 | 0.00 | 98 | 11 | 138 |
| 175TT    | 312,193.78 | 17,634.54 | 4,886.88 | 223.70 | 0.07 | 94 | 7 | 764 |
| 200LL    | 231,069.50 | 725.40 | 5,813.00 | 210.83 | 0.00 | 98 | 54 | 241 |
| 200LT    | 267,218.35 | 7,460.18 | 45,874.73 | 980.45 | 0.71 | 86 | 19 | 1441 |
| 200TL    | 273,443.81 | 902.96 | 869.67 | 163.37 | 0.00 | 98 | 8 | 412 |
| 200TT    | 290,582.04 | 4,117.31 | 3,211.04 | 346.75 | 0.11 | 98 | 7 | 804 |
| Geom. mean | 470.16 | 1,376.70 | 88.96 | 0.00 | 95 | 20 | 252 |
| Arith. mean | 2,183.57 | 5,057.30 | 149.67 | 0.13 | 95 | 29 | 411 |

In the case of the CHLPSA, our algorithm is significantly faster than the others. In some instances it is up to three orders-of-magnitude faster, e.g., 150LT which was not solved in 12 hours by
The matheuristic provides an optimal solution for 13 out of 20 instances, while the average deviation is only 0.13%. The most time consuming instance (200LT) was solved in about eight minutes.

Our last series of experiments in this section is for the CpHMPSA, where the fixed set-up costs are disregarded but capacities and cardinality constraints of facilities are taken into account. To the best of our knowledge, there is no ad hoc algorithm for solving this hub location variant in the literature. Therefore, in Table 7, we only report the results obtained with our algorithm using the AP data set.

The results of Table 7 show that our exact algorithm can optimally solve all considered instances in less than one hour. In fact, the most time consuming instance (200 nodes and $p = 20$) took only 30 minutes to be solved. The matheuristic is capable of finding an optimal solution in 13 out of 20 instances, and the average deviation is only 0.06%. It is worth noting that the CpHMPSA seems to be more difficult to solve as compared to the CHLPSA. The largest CPU time for the CHLPSA is about eight minutes (200 nodes with loose setup costs and tight capacities) whereas for the CpHMPSA is 30 minutes (200 nodes and $p = 20$).
7.2. Results for Larger Instances

As seen in Section 7.1, our algorithm scales well to larger instances while either being competitive or significantly faster than tailored exact algorithms that exploit problem-specific structure for all considered problem variants. This leads to the question of what are the size limits of our proposed algorithm. In this section, we present the results of solving the large-scale instances first presented in Contreras et al. (2011a) as a single-assignment variant. During this last set of experiments, we focus on the two variants in which our algorithm performed the best for the AP data set: the UHLPSA and CHLPSA.

Contreras et al. (2011a) introduce a new data set containing three different sets of instances with diverse structural characteristics in the flow network. They consider different levels of magnitude for the amount of flow originating at a given node to obtain three different sets of nodes: low-level (LL) nodes, medium-level (ML) nodes, and high-level (HL) nodes. The total outgoing flow of LL, ML, and HL nodes lies in the interval $[1, 10]$, $[10, 100]$, and $[100, 1,000]$, respectively. In this section, we use the first set of instances, called Set I, in which the number of HL, ML, and LL nodes is 2%, 38%, and 60% of the total number of nodes, respectively. We use the instance generation code of Contreras et al. (2011a) to generate instances with $N = 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950,$ and $1,000$. To generate setup costs and capacities, we use the same procedure as described in Contreras et al. (2011a).

Tables 8 and 9 present the detailed results of our branch-and-cut algorithm for the UHLPSA and CHLPSA, respectively, using the Set I instances. In the case of the UHLPSA, instances with up to 700 nodes can be optimally solved in less than half of a day of CPU time. The largest-size instances with 750 to 1,000 were optimally solved in CPU times of less than three days. From our observations, these results are remarkable given that the number of variables in the largest instance with 1,000 nodes require 499,501,000,000 variables in the RL formulation. Similar results are obtained for the more challenging CHLPSA, in which instances with up to 800 nodes can be solved in less than one day of CPU time. The largest instances from 850 to 1,000 nodes require two to three days to prove optimality. Moreover, the matheuristic is capable of finding the optimal solution to 13 out of 15 uncapacitated instances and in 14 out of 20 for the capacitated case, with an average deviation of only 0.01%.

8. Conclusions

In this paper, we have studied a general class of non-convex quadratic capacitated $p$-location problems with single assignments. The quadratic term in the objective function accounts for the interaction cost between facilities and customer assignments. A binary quadratic program was first linearized by applying a reformulation-linearization technique and the associated additional
variables were then projected out using Benders decomposition. We proposed an exact branch-
and-cut algorithm that incorporated several algorithmic refinements such as stabilized Benders
Pareto-optimal cuts, a matheuristic, and variable elimination and partial enumeration procedures.
Extensive computational experiments on a set of existing and new large-scale instances with up to 1,000 nodes have clearly confirmed the efficiency and robustness of our algorithm in solving four different particular cases of the studied problem. To the best of our knowledge, the new instances are by far the largest and most difficult ever solved for any type of hub location problem.

Acknowledgments
This research was partially funded by the the Canadian Natural Sciences and Engineering research Council under grants 2018-06704, 2017-01746, and by a Research & Publication Grant, Indian Institute of Management Ahmedabad. The second author would also like to acknowledge the support and productive research environment provided by Andrea Lodi at the Canada Excellence Research Chair in Data Science for Real-Time Decision-Making as well as IVADO Labs. This support is gratefully acknowledged.

References
Adams WP, Sherali HD (1986) A tight linearization and an algorithm for zero-one quadratic programming problems. Management Science 32(10):1274–1290.
Alibeyg A, Contreras I, Fernández E (2018) Exact solution of hub network design problems with profits. European Journal of Operational Research 266:57–71.
Applegate D, Bixby R, Chvatal V, Cook B (1995) Finding cuts in the tsp (a preliminary report). Technical report.
Avella P, Boccia M, Vasilyev I (2010) A computational study of exact knapsack separation for the generalized assignment problem. Computational Optimization and Applications 45:543–555.
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 4:238–252.
Berman O, Krass D (2015) Stochastic location models with congestion. Laporte G, Saldanha da Gama F, Nickel S, eds., Location Science, 443–486.
Brimberg J, Mladenovic N (1996) A variable neighbourhood algorithm for solving the continuous location-allocation problem. Studies in Locational Analysis 10:1–12.
Calik H, Labbe M, Yaman H (2015) p-center problems. Laporte G, Saldanha da Gama F, Nickel S, eds., Location Science, 79–92.
Camargo R, Miranda JG, Luna H (2011) Benders decomposition for hub location problems with economies of scale. Transportation Science 43:86–97.
Campbell JF (1994) Integer programming formulations of discrete hub location problems. European Journal of Operational Research 72(2):387–405.
Campbell JF, O’Kelly ME (2012) Twenty-five years of hub location research. *Transportation Science* 46(2):153–169.

Cánovas L, García S, Marín A (2007) Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique. *European Journal of Operational Research* 179:990–1007.

Caprara A (2008) Constrained 0–1 quadratic programming: Basic approaches and extensions. *European Journal of Operational Research* 187(3):1494–1503.

Carrizosa E, Tóth B (2015) Anti-covering problems. Laporte G, Saldanha da Gama F, Nickel S, eds., *Location Science*, 115–132.

Church R, ReVelle C (1974) The maximal Covering Location Problem. *Regional Science Association Papers* 32:101–118.

Contreras I, Cordeau JF, Laporte G (2011a) Benders decomposition for large-scale uncapacitated hub location. *Operations Research* 59(6):1477–1490.

Contreras I, Cordeau JF, Laporte G (2011b) The dynamic uncapacitated hub location problem. *Transportation Science* 45:18–32.

Contreras I, Cordeau JF, Laporte G (2011c) Exact solution of large-scale hub location problems with multiple capacity levels. *Transportation Science* 46:439–459.

Contreras I, Díaz JA, Fernández E (2011d) Branch and price for large-scale capacitated hub location problems with single assignment. *INFORMS Journal on Computing* 23(1):41–55.

Contreras I, Fernández E (2014) Hub location as the minimization of a supermodular set function. *Operations Research* 62(3):557–570.

Contreras I, O’Kelly M (2019) Hub location problems. Laporte G, Saldanha da Gama F, Nickel S, eds., *Location Science*.

Correia I, da Gama FS (2015) *Facility Location Under Uncertainty* (Springer).

Crowder H, Padberg MW (1980) Solving large-scale symmetric travelling salesman problems to optimality. *Management Science* 26(5):495–509.

Curtin KM, Church RL (2006) A family of location models for multiple-type discrete dispersion. *Geographical Analysis* 38(3):248–270.

Daskin M, Lee Maass K (2015) The p-median problem. Laporte G, Saldanha da Gama F, Nickel S, eds., *Location Science*, 21–46.

Desrochers M, Marcotte P, Stan M (1995) The congested facility location problem. *Location Science* 3(1):9–23.

Drezner Z, Hamacher H (2001) *Facility Location* (Springer-Verlag Berlin).

Elhedhli S, Merrick R (2012) Green supply chain network design to reduce carbon emissions. *Transportation Research Part D: Transport and Environment* 17(5):370 – 379.
Ernst AT, Krishnamoorthy M (1996) Efficient algorithms for the uncapacitated single allocation p-hub median problem. Location Science 4(3):139 – 154.

Ernst AT, Krishnamoorthy M (1998) An exact solution approach based on shortest-paths for p-hub median problems. INFORMS Journal on Computing 10:149–162.

Fernandez E, Landete M (2015) Fixed-charge facility location problems. Laporte G, Saldanha da Gama F, Nickel S, eds., Location Science, 47–78.

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.

Fortet R (1959) L’algebre de boole et ses applications en recherche operationnelle. Cahiers du Centre d’Etudes de Recherche Operationelle 1:5–36.

Gadegaard SL, Klose A, Nielsen LR (2018) An improved cut-and-solve algorithm for the single-source capacitated facility location problem. EURO Journal on Computational Optimization 6(1):1–27.

Garcia S, Marin A (2015) Covering location problems. Laporte G, Saldanha da Gama F, Nickel S, eds., Location Science, 93–114.

Ghaffarinasab N, Kara BY (2018) Benders decomposition algorithms for two variants of the single allocation hub location problem. Networks and Spatial Economics .

Glover F (1975) Improved linear integer programming formulations of nonlinear integer problems. Management Science 22(4):455–460.

Glover F, Woolsey E (1974) Converting the 0-1 polynomial programming problem to a 0-1 linear program. Operations Research 22(1):180–182.

Günlük O, Lee J, Weismantel R (2007) MINLP strengthening for separable convex quadratic transportation-cost UFL. Technical Report RC2413 (W0703-042), Research Division, IBM, Yorktown Heights, NY.

Günlük O, Linderoth J (2012) Perspective reformulation and applications. Lee J, Leyffer S, eds., Mixed Integer Nonlinear Programming, 61–92.

Hakimi SL (1964) Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research 12(3):450–459.

Harkness J, ReVelle C (2003) Facility location with increasing production costs. European Journal of Operational Research 145(1):1–13.

Helme MP, Magnanti TL (1989) Designing satellite communication networks by zero—one quadratic programming. Networks 19(4):427–450.

Kuehn A, Hamburger M (1963) A heuristic program for locating warehouses. Management Science 9(4):643–666.
Labbé M, Yaman H (2008) Solving the hub location problem in a star–star network. Networks 51(1):19–33.

Labbé M, Yaman H, Gourdin E (2005) A branch and cut algorithm for hub location problems with single assignment. Mathematical programming 102(2):371–405.

Lei TL, Church RL (2013) A unified model for dispersing facilities. Geographical Analysis 45(4):401–418.

Liberti L (2007) Compact linearization for binary quadratic problems. 4OR 5(3):231–245.

Lu D, Gzara F, Elhedhli S (2014) Facility location with economies and diseconomies of scale: models and column generation heuristics. IIE Transactions 46(6):585–600.

Magnanti T, Mireault P, Wong R (1986) Tailoring Benders decomposition for uncapacitated network design. Gallo G, Sandi C, eds., Network Flow at Pisa, 112–154, Mathematical Programming Studies, volume 26.

Magnanti TL, Wong RT (1981) Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. Operations Research 29(3):464–484.

Mahéo A, Kilby P, Van Hentenryck P (2019) Benders decomposition for the design of a hub and shuttle public transit system. Transportation Science 53(1):77–88.

Martins de Sá E, Contreras I, Cordeau JF, de Camargo R, Miranda G (2015) The hub line location problem. Transportation Science 9:500–518.

Meier JF, Clausen U (2018) Solving single allocation hub location problems on Euclidean data. Transportation Science 52(5):1141–1155.

Neebe AW, Rao MR (1983) An algorithm for the fixed-charge assigning users to sources problem. Journal of the Operational Research Society 34(11):1107–1113.

Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions - I. Mathematical Programming 14:265–294.

O’Kelly ME (1986a) Activity levels at hub facilities in interacting networks. Geographical Analysis 18(4):343–356.

O’Kelly ME (1986b) The location of interacting hub facilities. Transportation Science 20(2):92–106.

O’Kelly ME (1987) A quadratic integer program for the location of interacting hub facilities. European Journal of Operational Research 32(3):393–404.

Ortiz-Astorquiza C, Contreras I, Laporte G (2019) An exact algorithm for multi-level uncapacitated facility location. Transportation Science 53:1085–1106.

Pessoa A, Sadykov R, Uchoa E, Vanderbeck F (2018) Automation and combination of linear-programming based stabilization techniques in column generation. INFORMS Journal on Computing 30(2):339–360.

Rostami B, Errico F, Lodi A (2018) A convex reformulation and an outer approximation for a class of binary quadratic program. Technical Report DS4DM-2018-002.
Rostami B, Strothmann C, Buchheim C (2016) A decomposition approach for single allocation hub location problems with multiple capacity levels. Cerulli R, Fujishige S, Mahjoub AR, eds., Combinatorial Optimization, 237–248.

Rothenbcher AK, Drexl M, Irnich S (2016) Branch-and-price-and-cut for a service network design and hub location problem. European Journal of Operational Research 255:935–947.

Saito H, Fujie T, Matsui T, Matuura S (2009) A study of the quadratic semi-assignment polytope. Discrete Optimization 6(1):37 – 50.

Sherali H, Adams W (2013) Reformulation–linearization techniques for discrete optimization problems. Handbook of Combinatorial Optimization., 2849–2896.

Sherali HD, Adams WP (1990) A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM Journal on Discrete Mathematics 3(3):411–430.

Sherali HD, Smith JC (2007) An improved linearization strategy for zero-one quadratic programming problems. Optimization Letters 1(1):33–47.

Soland RM (1974) Optimal facility location with concave costs. Operations Research 22:373–382.

Tanash M, Contreras I, Vidyarthi N (2017) An exact algorithm for the modular hub location problem with single assignments. Computers & Operations Research 85:32–44.

Toregas C, Swain R, ReVelle C, Bergman L (1971) The location of emergency service facilities. Operations Research 19(6):1363–1373.

Wolsey LA (1983) Fundamental properties of certain discrete location problems. Thisse JF, Zoller HG, eds., Locational Analysis of Public Facilities, 331–356.

Zangwill WI (1968) Minimum concave cost flows in certain networks. Management Science 14:429–450.

Zetina CA, Contreras I, Cordeau JF (2019a) Exact algorithms based on Benders decomposition for multi-commodity uncapacitated fixed-charge network design. Computers & Operations Research 311–324.

Zetina CA, Contreras I, Cordeau JF, Nikbakhsh E (2017) Robust uncapacitated hub location. Transportation Research Part B: Methodological 106:393–410.

Zetina CA, Contreras I, Fernández E, Luna-Mota C (2019b) Solving the optimum communication spanning tree problem. European Journal of Operational Research 273(1):108–117.