



HAL
open science

Robust MILP formulations for the two-stage weighted vertex p -center problem

Cristian Duran-Mateluna, Zacharie Ales, Sourour Elloumi, Natalia Jorquera-Bravo

► **To cite this version:**

Cristian Duran-Mateluna, Zacharie Ales, Sourour Elloumi, Natalia Jorquera-Bravo. Robust MILP formulations for the two-stage weighted vertex p -center problem. *Computers and Operations Research*, 2023, pp.106334. 10.1016/j.cor.2023.106334 . hal-04146260

HAL Id: hal-04146260

<https://hal.science/hal-04146260>

Submitted on 29 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Robust MILP formulations for the two-stage weighted vertex p -center problem

Cristian DURAN-MATELUNA^{a,b,c}, Zacharie ALES^{a,b}, Sourour ELLOUMI^{a,b}, Natalia
JORQUERA-BRAVO^{a,b}

^a UMA, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France.

^b CEDRIC, Conservatoire National des Arts et Métiers, 75003 Paris, France.

^c PDSPPS, Department of Industrial Engineering, University of Santiago of Chile, Santiago, Chile.

Abstract

The weighted vertex p -center problem (PCP) consists of locating p facilities among a set of available sites such that the maximum weighted distance (or travel time) from any demand node to its closest located facility is minimized. This paper studies the exact solution of the two-stage robust weighted vertex p -center problem ($RPCP_2$). In this problem, the location of the facilities is fixed in the first stage while the demand node allocations are recourse decisions fixed once the uncertainty is revealed. The problem is modeled by box uncertainty sets on both the demands and the distances. We introduce five different robust reformulations based on MILP formulations of (PCP) from the literature. We prove that considering a finite subset of scenarios is sufficient to obtain an optimal solution of ($RPCP_2$). We leverage this result to introduce a *column-and-constraint generation* algorithm and a *branch-and-cut* algorithm to efficiently solve this problem optimally. We highlight how these algorithms can also be adapted to solve the single-stage problem ($RPCP_1$) which is obtained when no recourse is considered. We present a numerical study to compare the performances of these formulations on randomly generated instances and on a case study from the literature.

Keywords: discrete location; p -center problem; robust MILP formulations; *column-and-constraint generation* algorithm; *branch-and-cut* algorithm

1 Introduction

The (vertex) p -center problem (PCP) is one of the most studied facility location problems in the literature. It consists of locating p facilities out of m available sites, and allocating n demand nodes

to them, in order to minimize the *radius* which corresponds to the maximum distance (or travel time) between a demand node and its closest located facility. These p located facilities are called *centers*.

The p -center problem under uncertainty is also well studied in the literature ([Çalik et al., 2019]). This problem arises when parameters, such as demands or distances between the demand nodes and the available sites, vary across time or when their exact value is uncertain. Uncertainty is generally represented by parameters which can take any value in an uncertainty set. Each realization of the uncertainty set is called a *scenario*. The most classical sets are the box, the ellipsoidal and the budgeted uncertainty sets (see e.g., [Ben-Tal et al., 2009, Bertsimas and Sim, 2004, Du and Zhou, 2018, Paul and Wang, 2019]).

Two major approaches have been developed to address uncertainty: stochastic optimization and robust optimization. Stochastic optimization requires that a discrete or continuous probabilistic distribution of the uncertain parameters is known, and tries to have the best value on average. Robust optimization tries to protect itself against the worst case ([Ben-Tal et al., 2009]). In this sense, stochastic optimization is more relevant in the context of repeated experiments while robust optimization is more suitable when one wishes at all costs to avoid the worst case, for example, when human lives are at stake.

The incorporation of uncertainty in (PCP) has important applications in emergency logistics problems, where a prompt response to the urgent need for relief is required in affected areas immediately following a disaster (such as earthquakes, tsunamis, landslides, among others). The consequences of these disasters make it challenging to precisely estimate the demand for relief materials or the travel times between relief centers and affected locations ([Sheu, 2007]). Two approaches can be considered depending on whether the demand node allocations to the centers are made before (see e.g, [Averbakh and Berman, 1997], [Lu, 2013]) or after (see e.g, [Du et al., 2020, Demange et al., 2020]) the uncertainty is revealed. The first case corresponds to single-stage problems while the second case leads to two-stage problems in which the demand node allocations are recourse variables. In this context, most of the works have been focused on the investigation of integer programming modeling and heuristic solution approaches (see e.g., [Baron et al., 2011, Hasani and Mokhtari, 2018, Paul and Wang, 2015, Trivedi and Singh, 2017, Trivedi and Singh, 2019]).

1.1 Contribution and Outline

To the best of our knowledge this work is the first one to study the exact solution of the two-stage robust weighted vertex p -center problem ($RPCP_2$), with $p > 1$, in which the uncertainty on the node demands and distances are modeled by box uncertainty sets. We present robust reformulations of this problem based on five MILP formulations of (PCP). We prove that a finite subset of scenarios from the infinite box uncertainty set can be considered without losing optimality. We use this result to propose a *column-and-constraint generation* algorithm ($C\&CG$) and a *branch-and-cut* algorithm ($B\&C$) for the exact solution of ($RPCP_2$). We highlight how these algorithms can be adapted to the single-stage problem ($RPCP_1$). Finally, we show their efficiency on randomly generated instances. To illustrate the usefulness of our proposal we also consider the case study presented in [Lu, 2013] which is inspired from an earthquake that hit central Taiwan in 1999.

The rest of the paper is organized as follows. Section 2 presents the literature review of the deterministic and robust versions of (PCP). Section 3 describes the robust two-stage problem ($RPCP_2$), proves how to reduce the number of considered scenarios, and introduces our five MILP formulations as well as the ($C\&CG$) and ($B\&C$) algorithms. Section 4 presents the computational results. In Section 5 we draw conclusions together with research perspectives.

2 Literature review

The p -center problem was introduced by [Hakimi, 1965], who presented and solved the absolute 1-center problem on a graph. In the *absolute* p -center problem, the centers can be located either on the edges or the vertices of the graph. Later, [Minieka, 1970] extended the problem to the case $p > 1$ and proposed a method to restrict the continuous set of candidate centers to a discrete set of points, without losing optimality. Since then, the problem was commonly referred to as the *vertex* p -center problem or directly as the p -center problem. Several formulations, solution methods, and variants of this problem have been presented. We refer to [Çalik et al., 2019] for a more exhaustive review of applications and solution methods of the p -center problem. In this section, we first focus on the deterministic p -center problem formulations and then on its robust counterparts.

2.1 MILP formulations of the deterministic weighted vertex p -center

Let U be the set of available sites, and V be the set of demand nodes. The distance (or travel time) between any possible pair of demand node $i \in V$ and site $j \in U$ is denoted by t_{ij} . Each demand node

$i \in V$ faces a demand d_i and must be allocated to a single center j . In the following formulations, no demands were originally considered. They directly considered a distance l_{ij} between each demand node i and site j . Nevertheless, to model the weighted vertex p -center, l_{ij} can be replaced by the product of demand d_i and distance t_{ij} .

The classical formulation of the p -center problem was presented in [Daskin, 1996]. This model considers binary variables x_j equal to 1 if and only if site $j \in U$ is located, binary variables y_{ij} equal to 1 if and only if demand node i is allocated to site j , and a variable z equals to the radius:

$$\begin{array}{l}
 \left. \begin{array}{l}
 \text{s.t.} \\
 (F1)
 \end{array} \right\} \begin{array}{ll}
 \min z, & (1) \\
 z \geq \sum_{j \in U} d_i t_{ij} y_{ij}, & i \in V, \quad (2) \\
 \sum_{j \in U} y_{ij} = 1, & i \in V, \quad (3) \\
 y_{ij} \leq x_j, & i \in V, j \in U, \quad (4) \\
 \sum_{j \in U} x_j = p, & (5) \\
 x_j \in \{0, 1\}, & j \in U, \quad (6) \\
 y_{ij} \in \{0, 1\}, & i \in V, j \in U. \quad (7)
 \end{array}
 \end{array}$$

Constraints (3) ensure that each demand node is allocated to only one center and Constraints (4) ensure that no demand node is allocated to a site that is not located. Constraint (5) fixes the number of centers to p . Constraints (2) indicate that the distances between each demand node and its nearest center are less than or equal to the radius. We minimize the radius through Objective function (1).

An alternative formulation was introduced in [Elloumi et al., 2004]. This formulation proposes to associate one variable to each weighted distance in the considered instance. Let $D^0 < D^1 < \dots < D^K$ be the different values in $\{d_i t_{ij}, \forall i \in V, \forall j \in U\}$ and let \mathcal{K} be the set $\{1, 2, \dots, K\}$. The radius variable z and the allocation variables y are replaced by variables z^k with $k \in \mathcal{K}$ equal to 1 if and only if the radius is greater than or equal to D^k . The *initial* formulation they obtain is as follows:

which represents the index of the optimal radius:

$$(F3) \left\{ \begin{array}{ll} \min r, & (13) \\ \text{s.t.} & r + k \sum_{j:d_it_{ij} < D^k} x_j \geq k, \quad i \in V, k \in \mathcal{K}, \quad (14) \\ & \sum_{j \in U} x_j = p, \\ & x_j \in \{0, 1\}, \quad j \in U, \\ & r \geq 0. \quad (15) \end{array} \right.$$

Constraints (14) play a similar role to that of Constraints (9). This formulation (F3) provides a weaker linear relaxation than the previous formulations.

[Calik and Tansel, 2013] introduce two mathematical formulations. The second, denoted by (F4) in this article, is a tighter version of the first. They consider binary variables x of (F1) and a binary variable u^k for all $k \in \mathcal{K} \cup \{0\}$ which is equal to 1 if and only if the optimal radius is equal to D^k . Thus, exactly one of these new binary variables is equal to one. The solutions of these formulations can be mapped to those of (F2_{init}) by setting $u^0 = 1 - z^1$, $u^k = z^k - z^{k+1}$, and $z^k = 1 - \sum_{k=0}^K u^k$.

$$(F4) \left\{ \begin{array}{ll} \min \sum_{k=0}^K D^k u^k, & (16) \\ \text{s.t.} & \sum_{j:d_it_{ij} \leq D^k} x_j \geq \sum_{q=0}^k u^q, \quad i \in V, k \in \mathcal{K} \cup \{0\}, \quad (17) \\ & \sum_{k=0}^K u^k = 1, \quad (18) \\ & \sum_{j \in U} x_j = p, \\ & x_j \in \{0, 1\}, \quad j \in U, \\ & u^k \in \{0, 1\}, \quad k \in \mathcal{K} \cup \{0\}. \quad (19) \end{array} \right.$$

Finally, [Gaar and Sinnl, 2022] recently presented a formulation obtained from a Benders decomposition of (F1) which is closely related to a formulation of the *uncapacitated facility location problem (UFL)* from [Cornuejols et al., 1980] and [Magnanti and Wong, 1981]. This formulation also considers binary variables x from (F1) and a single continuous variable θ to represent the radius which is minimized in the objective function.

$$\begin{aligned}
& \min \theta, & (20) \\
\text{s.t.} & \theta \geq d_i t_{ij} - \sum_{j': d_i t_{ij'} < d_i t_{ij}} (d_i t_{ij} - d_i t_{ij'}) x_{j'}, & i \in V, j \in U, & (21) \\
(F5) & \sum_{j \in U} x_j = p, \\
& x_j \in \{0, 1\}, & j \in \mathcal{U} \\
& \theta \in \mathbb{R}. & (22)
\end{aligned}$$

Constraints (21) ensure that the radius is at least the weighted travel time to the nearest center for each demand node.

Table 1 presents the CPU times of the 5 formulations on the ORLIB instances. We have considered the default setting of CPLEX 20.1 and a time limit of 3 hours. These instances and the characteristics of our computer are detailed in Section 4. Since (F1) cannot solve all the instances, the best solution value found and the corresponding optimality gap are represented in the last two columns.

(F4) is the fastest formulation in average closely followed by (F2). However they can both be significantly faster than the other depending on the instances. Note that when CPLEX pre-solve is disabled only formulations (F2) and (F4) solve most of the instances in less than 30 minutes, with (F2) performing better than (F4).

We will see in Section 4 that the best formulations for the deterministic problem are not necessarily the best for the robust problem.

2.2 Uncertainty representation and solution methods

The location of facilities is a long-term decision which takes into account parameters such as demands or distances between demand nodes and facilities. Since these parameters are likely to vary, several models have been developed to study facility location problems under uncertainty. Stochastic optimization and robust optimization are the two main approaches to address uncertainty. We refer to [Snyder, 2006] and [Correia and Saldanha-da Gama, 2019] for a review of the literature on stochastic and robust facility location problems.

In robust optimization, box, budgeted, ellipsoidal and discrete uncertainty sets are commonly considered (see e.g. [Ben-Tal et al., 2009, Baron et al., 2011, Du and Zhou, 2018, Paul and Wang, 2019, Paul and Wang, 2015, Snyder, 2006]). Since most robust facility location problems are generally harder to solve than their deterministic counterparts, heuristic approaches have taken precedence

Instance				Time (s)					F1	
<i>name</i>	$ V = U $	p	opt	F1	F2	F3	F4	F5	<i>obj</i>	<i>gap</i>
pmed01	100	5	127	13	3	10	25	15	127	0%
pmed02	100	10	98	8	2	16	25	15	98	0%
pmed03	100	10	93	7	3	18	26	19	93	0%
pmed04	100	20	74	4	2	194	25	20	74	0%
pmed05	100	33	48	2	2	11	21	16	48	0%
pmed06	200	5	84	106	18	68	64	61	84	0%
pmed07	200	10	64	81	20	56	64	65	64	0%
pmed08	200	20	55	51	11	421	83	108	55	0%
pmed09	200	40	37	27	21	72	73	108	37	0%
pmed10	200	67	20	12	12	57	49	60	20	0%
pmed11	300	5	59	282	19	134	61	45	59	0%
pmed12	300	10	51	248	25	117	75	142	51	0%
pmed13	300	30	36	218	104	513	97	217	36	0%
pmed14	300	60	26	127	101	429	117	390	26	0%
pmed15	300	100	18	54	55	127	82	139	18	0%
pmed16	400	5	47	875	17	194	63	141	47	0%
pmed17	400	10	39	4652	113	298	64	224	39	0%
pmed18	400	40	28	965	688	3601	243	445	28	0%
pmed19	400	80	18	679	255	780	173	292	18	0%
pmed20	400	133	13	165	44	194	29	238	13	0%
pmed21	500	5	40	3001	29	65	69	68	40	0%
pmed22	500	10	38	TL	381	770	106	775	42	11%
pmed23	500	50	22	2729	456	1768	310	759	22	0%
pmed24	500	100	15	1538	322	906	130	384	15	0%
pmed25	500	167	11	574	70	378	54	687	11	0%
pmed26	600	5	38	10426	514	105	508	333	38	0%
pmed27	600	10	32	TL	41	1140	86	138	35	9%
pmed28	600	60	18	10289	550	2906	129	883	18	0%
pmed29	600	120	13	5195	134	1373	545	602	13	0%
pmed30	600	200	9	1234	87	544	60	426	9	0%
pmed31	700	5	30	TL	34	93	78	180	47	57%
pmed32	700	10	29	TL	1007	4285	1514	1600	116	300%
pmed33	700	70	15	TL	1086	3894	397	857	60	300%
pmed34	700	140	11	8794	761	1026	171	818	11	0%
pmed35	800	5	30	TL	2116	262	129	201	74	147%
pmed36	800	10	27	TL	4243	4889	1875	314	87	222%
pmed37	800	80	15	TL	1681	3922	1595	3251	63	320%
pmed38	900	5	29	TL	89	1235	183	609	84	190%
pmed39	900	10	23	TL	1932	521	1687	1932	115	400%
pmed40	900	90	13	TL	1419	6397	1570	2341	53	308%
Average				4279	462	1095	316	498		

Table 1: Performance comparison of (*PCP*) formulations. TL=7200s.

over exact solution methods ([Correia and Saldanha-da Gama, 2019]). Most robust facility location problems based on discrete uncertainty sets deal with generalizations of the p -median problem, focusing exclusively on analytical results or approximated polynomial-time algorithms (see e.g. [Serra and Marianov, 2018], [Hasani and Mokhtari, 2018]).

The presence or absence of recourse variables, the variables which are fixed once the uncertainty is revealed, has a great influence on the mathematical formulation of the problem. A single-stage problem can be considered when there is no recourse variables while a two-stage is required otherwise. Two-stage models are usually very difficult to solve ([Ben-Tal et al., 2009]). When the second stage problem is a linear program, Benders decomposition method can be used to seek optimal solutions ([Bertsimas et al., 2013, Rahmaniani et al., 2017]).

[Zeng and Zhao, 2013] develop another exact solution method, the ($C\&CG$) generation algorithm (also called row-and-column or scenario generation), which has performed better on different problems including facility location problems (see e.g. [An et al., 2014, Chan et al., 2018]).

Several robust variants of (PCP) with either a single stage or two stages have been considered. For example, [Averbakh and Berman, 1997] consider the weighted p -center problem on a transportation network with uncertain node weights. They minimize the regret of the worst-case scenario and show that the problem can be solved through a number of particular weighted p -center problems. [Averbakh and Berman, 2000], consider a box uncertainty set for the weighted 1-center problem on a network with uncertainty node weights and edge lengths. Each uncertain parameter is assumed to be random with an unknown distribution. They present a polynomial algorithm to find the robust solution for the problem on a tree. [Lu and Sheu, 2013] consider the single-stage weighted vertex p -center with uncertain edge lengths using box uncertainty sets. They consider the single-stage robust problem ($RPCP_1$), prove that it is sufficient to consider a discrete subset of scenarios, and propose a simulated annealing heuristic to solve the problem, then in [Lu, 2013] they extend this research to the weighted vertex p -center with uncertain nodal weights and edge lengths using also box uncertainty sets. [Du and Zhou, 2018] apply a single-stage approach to a p -center problem based on a multiple allocation strategy, i.e, they allowed the allocation of a client to several sites, and three types of symmetric uncertainty sets over units costs: box uncertainty, ellipsoidal uncertainty, and cardinality-constrained uncertainty, where a symmetric interval is defined as an interval where the lower and upper bounds are equidistant from the center. [Du et al., 2020] propose a two-stage robust model for a reliable facility location problem, i.e., when some facilities can be disrupted and the demand nodes can be reallocated to another available facility. They consider uncertain demand

and cost, and propose three solution methods: a linear reformulation, a Benders dual cutting planes method, and a *column-and-constraint generation* method. [Demange et al., 2020] introduce the robust p -center problem under pressure motivated by the context of locating shelters for evacuation in case of wildfires, where the uncertainty is in the available network connections. They present a MILP formulation and a decomposition scheme to solve it.

Article	Allocation level		Decision level		Number of centers		Uncertainty			Measure of robustness		Solution approach	
	Single	Multiple	Single-stage	Two-stage	p=1	p>1	Distance	Demand	Centers	Regret	Worst Case Value	Heuristic	Exact
[Averbakh and Berman, 1997]	x		x			x		x		x			x
[Averbakh and Berman, 2000]	x		x		x		x	x		x			x
[Lu and Sheu, 2013]	x		x			x	x			x		x	
[Lu, 2013]	x		x			x	x	x		x		x	
[Du and Zhou, 2018]		x	x			x	x			x			x
[Du et al., 2020]	x			x		x			x	x			x
[Demange et al., 2020]	x		x			x			x		x		x
Our Proposal	x			x		x	x	x		x			x

Table 2: A summary of related work

Other robust facility location problems have been studied in the literature, mostly considering uncertainty in customer demand and/or facility disruption. Their solution methods are carried out by dualization techniques or column and constraint generation algorithms ([Nikoofal and Sadjadi, 2010, An et al., 2014, Cheng et al., 2021]). For a more detailed literature review we refer you to [Snyder, 2006]. Our research focuses on the exact solution of the two stage weighted vertex p -center problem ($RPCP_2$), with $p > 1$, with uncertainty in both nodal weights and edge lengths, as shown in the last row of Table 2. We also show how these algorithms can be adapted to solve ($RPCP_1$) exactly.

3 Robust weighted vertex p -center problem

We first define ($RPCP_2$). We then prove that it is sufficient to consider a subset of the infinite scenarios in the box uncertainty set to obtain an optimal solution of the problem. Finally, we present ($C\&CG$) and ($B\&C$) algorithms for the exact solution of both ($RPCP_2$) and ($RPCP_1$).

3.1 Problem definition

Following [Lu, 2013], we consider that the node demands and distances can take any value in a box uncertainty set. More precisely, the demand d_i of demand node $i \in V$ is assumed to be in $[d_i^-, d_i^+]$ where $0 \leq d_i^- \leq d_i^+$, while the distance t_{ij} between station $i \in V$ and site $j \in U$ takes its value in $[t_{ij}^-, t_{ij}^+]$ where $0 \leq t_{ij}^- \leq t_{ij}^+$.

Let $W \subset \mathbb{R}^{|V|+|U| \times |V|}$ be the Cartesian product of intervals $[d_i^-, d_i^+]$ and $[t_{ij}^-, t_{ij}^+]$ for each $i \in V$ and $j \in U$. Let $\Omega = \{x \in \{0, 1\}^{|U|} \mid \sum_{j \in U} x_j = p\}$ be the set of vectors representing p located facilities and let $J_x = \{j \in U \mid x_j = 1\}$ be the set of located facilities for vector $x \in \Omega$. For a given scenario $w \in W$ let d_i^w and t_{ij}^w respectively be the demand of node $i \in V$ and the distance between demand node i and site $j \in U$ in scenario w .

In $(RPCP_2)$, the demand nodes are allocated after the uncertainty is revealed. This corresponds to a two-stage approach in which the location of centers is fixed at the first stage and the demand node allocations are the recourse decisions of the second stage. Consequently, the optimal radius associated with $x \in \Omega$ when scenario $w \in W$ occurs is:

$$Z(w, x) = \max_{i \in V} \left\{ \min_{j \in J_x} d_i^w t_{ij}^w \right\} \quad (23)$$

which represents an optimal allocation of demand nodes in scenario w when sites J_x are located. Let $x^*(w) \in \Omega$ be a vector such that the location of $J_{x^*(w)}$ leads to an optimal radius for the deterministic p -center problem in which the uncertain data takes value $w \in W$. We define the *robust deviation* of $x \in \Omega$ for scenario w as:

$$DEV(w, x) = Z(w, x) - Z(w, x^*(w)) \quad (24)$$

It is a non-negative value corresponding to the increase in radius incurred when locating centers J_x rather than $J_{x^*(w)}$ in scenario w . The following lemma allows us to characterize situations where the deviation is zero.

Lemma 1 *Let $x \in \Omega$ and let w be a scenario. Let $(i, j) \in V \times J_x$ be such that $Z(w, x) = d_i^w t_{ij}^w$, i.e., (i, j) allow to reach the optimal radius $Z(w, x)$. Let $\bar{j} = \operatorname{argmin}_{j \in J_{x^*(w)}} t_{ij}^w$, i.e., node i is optimally assigned to site \bar{j} when sites set $J_{x^*(w)}$ of solution $x^*(w)$ are opened. It holds that if $\bar{j} \in J_x$ then $DEV(w, x) = 0$.*

Proof $DEV(w, x) \geq 0$ by definition. Suppose $DEV(w, x) > 0$ i.e. $Z(w, x) > Z(w, x^*(w))$. As i is assigned to \bar{j} for solution $x^*(w)$, $Z(w, x^*(w)) \geq d_i^w t_{i\bar{j}}^w$. Therefore, $Z(w, x) = d_i^w t_{ij}^w > d_i^w t_{i\bar{j}}^w$ and then $t_{ij}^w > t_{i\bar{j}}^w$. This contradicts the fact that $t_{ij}^w \leq t_{i\bar{j}}^w$ that follows from $\bar{j} \in J_x$. \square

The robustness cost of solution $x \in \Omega$ corresponds to the maximal possible robust deviation if sites J_x are located:

$$RC(x) = \max_{w \in W} DEV(w, x) \quad (25)$$

We denote by *worst-case scenario* a scenario which solves (25). $(RPCP_2)$ aims to minimize the regret in the worst-case scenario for all feasible solution $x \in \Omega$:

$$(RPCP_2) : \min_{x \in \Omega} RC(x) \quad (26)$$

We now show that it is not necessary to consider the whole uncertainty set W to optimally solve $(RPCP_2)$.

3.2 Reducing the number of scenarios

Since a box-uncertainty set contains an infinite number of scenarios for a given solution $x \in \Omega$, the evaluation of the robustness cost (25) is a major challenge when solving $(RPCP_2)$. Following [Lu, 2013], we prove that it is sufficient to consider $n = |V|$ scenarios per solution $x \in \Omega$ to optimally solve $(RPCP_2)$, i.e., $|V| \cdot \binom{|U|}{p}$ scenarios, instead of the $p^{|V|} \cdot |V| \cdot \binom{|U|}{p}$ scenarios considered in [Lu, 2013].

Definition 1 Let $x \in \Omega$ be a feasible solution and let $\bar{i} \in V$ be any demand node. We define $w_{\bar{i}}(x)$ as the following scenario:

$$\begin{aligned} \bullet \quad d_i^{w_{\bar{i}}(x)} &= \begin{cases} d_i^+ & \text{if } i = \bar{i} \\ d_i^- & \text{otherwise} \end{cases} \\ \bullet \quad t_{ij}^{w_{\bar{i}}(x)} &= \begin{cases} t_{ij}^+ & \text{if } i = \bar{i} \text{ and } x_j = 1 \\ t_{ij}^- & \text{otherwise} \end{cases} \end{aligned}$$

Hence, in scenario $w_{\bar{i}}(x)$, node \bar{i} is at its maximal demand value while the other nodes are at their minimal demand value. Also, the traveling time of any node-site pair is set to its minimal value except for node \bar{i} for which its travel times are maximal.

We now prove that at least one of the scenarios in $\{w_i(x)\}_{i \in V}$ leads to a maximal deviation for $x \in \Omega$.

Theorem 1 Let $x \in \Omega$ be a first-stage solution of $(RPCP_2)$. There exists $i \in V$ such that $w_i(x)$ is an optimal solution for $RC(x)$.

Proof: Let $w^0 \in W$ be an optimal solution to $RC(x)$ (i.e., $RC(x) = DEV(w^0, x)$). We build a sequence of three other optimal scenarios w^1, w^2, w^3 and prove that w^3 is equal to $w_i(x)$ for some $i \in V$.

Let us consider, for any $k \in \{0, 1, 2, 3\}$, the (node, site) pair $(i_k, j_k) \in V \times J_x$ such that $Z(w^k, x) = d_{i_k}^{w^k} t_{i_k j_k}^{w^k}$ (i.e., (i_k, j_k) allows to reach the optimal radius $Z(w^k, x)$).

Scenario w^1

Let w^1 be a scenario identical to w^0 except for $d_{i_0}^{w^1}$ which is equal to $d_{i_0}^+$. This amounts to multiplying any weighted travel time from i_0 by the same constant $\frac{d_{i_0}^+}{d_{i_0}^{w^0}}$. This ensures that an optimal radius in w^1 is still obtained for (i_0, j_0) and that one can set $(i_1, j_1) = (i_0, j_0)$. We below prove that the deviation of w^1 is not lower than that of w^0 .

$$DEV(w^1, x) - DEV(w^0, x) = Z(w^1, x) - Z(w^0, x) - [Z(w^1, x^*(w^1)) - Z(w^0, x^*(w^0))] \quad (27)$$

It holds that:

$$Z(w^1, x) - Z(w^0, x) = d_{i_1}^{w^1} t_{i_1 j_1}^{w^1} - d_{i_0}^{w^0} t_{i_0 j_0}^{w^0} = d_{i_0}^+ t_{i_0 j_0}^{w^0} - d_{i_0}^{w^0} t_{i_0 j_0}^{w^0} = (d_{i_0}^+ - d_{i_0}^{w^0}) t_{i_0 j_0}^{w^0} \quad (28)$$

Now, let $\tilde{j} = \operatorname{argmin}_{j \in J_{x^*(w^0)}} t_{i_0 j}^{w^0}$, i.e., node i_0 is optimally assigned to site \tilde{j} when sites set $J_{x^*(w^0)}$ of solution $x^*(w^0)$ are opened. We aim at proving the following inequality:

$$Z(w^1, x^*(w^1)) - Z(w^0, x^*(w^0)) \leq (d_{i_0}^+ - d_{i_0}^{w^0}) t_{i_0 \tilde{j}}^{w^0} \quad (29)$$

First, $Z(w^1, x^*(w^1)) \leq Z(w^1, x^*(w^0))$ holds from the definition of $x^*(w)$. Second, let us consider solution $x^*(w^0)$ and its opened sites set $J_{x^*(w^0)}$. In scenario w^0 , node i_0 is assigned to site \tilde{j} at distance $d_{i_0}^{w^0} t_{i_0 \tilde{j}}^{w^0}$. In scenario w^1 , node i_0 is optimally assigned to the same site \tilde{j} , at distance $d_{i_0}^+ t_{i_0 \tilde{j}}^{w^0}$. All the other nodes are assigned in the same way in w^0 and in w^1 . Therefore, Inequality (29) is satisfied.

From (27), (28), and (29) we deduce the following inequality:

$$DEV(w^1, x) - DEV(w^0, x) \geq (d_{i_0}^+ - d_{i_0}^{w^0})(t_{i_0 j_0}^{w^0} - t_{i_0 \tilde{j}}^{w^0}) \quad (30)$$

It remains to prove that $t_{i_0 j_0}^{w^0} - t_{i_0 \tilde{j}}^{w^0} \geq 0$. It comes from the definition of \tilde{j} and Z that $d_{i_0}^{w^0} t_{i_0 \tilde{j}}^{w^0} \leq Z(w^0, x^*(w^0))$. It also comes from the definition of $x^*(w)$ that $Z(w^0, x^*(w^0)) \leq Z(w^0, x) = d_{i_0}^{w^0} t_{i_0 j_0}^{w^0}$. Consequently, $DEV(w^1, x)$ is at least as large as $DEV(w^0, x)$, and scenario w^1 is optimal for $RC(x)$.

Scenario w^2

Let w^2 be a scenario identical to w^1 except for $t_{i_1 j}^{w^1}$ which is equal to $t_{i_1 j}^+$ for all $j \in J_x$. This ensures that an optimal radius in w^2 is still obtained for (i_1, j_1) and that one can set $(i_2, j_2) = (i_1, j_1)$ since i_1 was already leading to the largest weighted travel time in w^1 and some of its travel times have been increased in w^2 . Let $\bar{j} = \operatorname{argmin}_{j \in J_{x^*(w^1)}} t_{i_1 j}^{w^1}$, i.e., node i_1 is optimally assigned to site \bar{j} when sites set $J_{x^*(w^1)}$ of solution $x^*(w^1)$ are opened. We discuss two cases:

- Case 1: $\bar{j} \in J_x$, i.e., site \bar{j} is open in x . In this case, we can deduce from Lemma 1 that $DEV(w^1, x) = 0$. As w^1 is an optimal scenario for $RC(x)$ we can deduce that $DEV(w^2, x) \leq 0$ and, as deviations are non-negative, $DEV(w^2, x) = 0$. Scenario w^2 is also optimal.
- Case 2: $\bar{j} \notin J_x$. Here, solution $x^*(w^1)$ is also optimal for scenario w^2 since node i_1 is assigned to $\bar{j} \notin J_x$ when sites $J_{x^*(w^1)}$ are located, and since the only difference between w^1 and w^2 is an increase of the travel time between i_1 and the sites in J_x . Consequently, $Z(w^1, x^*(w^1)) = Z(w^2, x^*(w^2))$. Moreover, the increase of travel times leads to $Z(w^2, x) \geq Z(w^1, x)$ which ensures that $DEV(w^2, x) \geq DEV(w^1, x)$. Therefore, scenario w^2 is optimal for $RC(x)$.

Scenario w^3

Let w^3 be a scenario identical to w^2 except for $d_i^{w^3}$ which is equal to d_i^- for all $i \in V \setminus \{i_2\}$ and $t_{ij}^{w^3}$ which is equal to t_{ij}^- for all $i \in V \setminus \{i_2\}$ and $j \in U$. This ensures that we can have $(i_3, j_3) = (i_2, j_2)$ since the demand and travel times of i_2 are not modified and the others are reduced. This also ensures that $Z(w^3, x) = Z(w^2, x)$ and that $Z(w^3, x^*(w^3))$ is not greater than $Z(w^2, x^*(w^2))$. Consequently, $DEV(w^3, x)$ is not lower than $DEV(w^2, x)$ which proves that $w_{i_3}(x)$ is an optimal scenario.

Finally, since $i_3 = i_2 = i_1 = i_0$, we can sum up the changes that were progressively made on w^0 to reach w^3 , and observe that w^3 is precisely $w_{i_0}(x)$. Therefore, scenario $w_{i_0}(x)$ is optimal for $RC(x)$. \square

Since Ω is finite, Theorem 1 enables to only consider a finite set of scenarios

$\bar{W} = \{w_i(x) \mid x \in \Omega, i \in V\}$ without losing the optimality:

$$(RPCP_2) : \quad \min_{x \in \Omega} \left\{ \max_{w \in \bar{W}} DEV(w, x) \right\} \quad (31)$$

3.3 MILP formulations of the robust weighted vertex p -center problem

We present how the five formulations presented in Section (2.1) can be adapted to solve $(RPCP_2)$.

Let Z_w^* be the optimal value of (PCP) when the uncertain parameters take value $w \in \bar{W}$ (i.e.,

$Z_w^* = Z(w, x^*(w))$). Note that in the following formulations, the value of Z_w^* is assumed to be known for all $w \in \overline{W}$. We present in Section 3.6 how these values can be computed efficiently in our algorithm when required.

Our robust formulation based on (F1) uses one set of allocation variables y_{ij}^w for each scenario $w \in \overline{W}$ to allow different demand node allocations depending on the scenario:

$$(RF1) \left\{ \begin{array}{ll} \min RC & (32) \\ \text{s.t.} & RC \geq \sum_{j \in U} d_i^w t_{ij}^w \cdot y_{ij}^w - Z_w^*, \quad i \in V, w \in \overline{W}, \quad (33) \\ & \sum_{j \in U} y_{ij}^w = 1, \quad i \in V, w \in \overline{W}, \quad (34) \\ & y_{ij}^w \leq x_j, \quad i \in V, j \in U, w \in \overline{W}, \quad (35) \\ & \sum_{j \in U} x_j = p, \\ & x_j \in \{0, 1\}, \quad j \in U, \\ & y_{ij}^w \in \{0, 1\}, \quad i \in V, j \in U, w \in \overline{W}. \quad (36) \end{array} \right.$$

Constraints (33) set a lower bound on the value of the robustness cost (RC) for each scenario. Objective (32) provides a solution with the lowest maximal deviation. This formulation contains an exponential number of variables and constraints as the size of \overline{W} is proportional to $|\Omega|$.

We now adapt the other formulations to (*RPCP*₂). For each scenario $w \in \overline{W}$, let K_w be the number of different values $\{d_i^w t_{ij}^w\}_{i \in V, j \in U}$ and let \mathcal{K}_w be the set of indices $\{1, \dots, K_w\}$. One needs to sort these values in increasing order and obtain a set of distinct distances D_w^k for $k \in \mathcal{K}_w$. We also consider one set of radius variables z_w^k :

Finally, we can also directly adapt (F5) to (RPCP₂).

$$(RF5) \left\{ \begin{array}{l} \min RC \\ \text{s.t.} \quad RC \geq d_i^w t_{ij}^w - \sum_{j': d_i^w t_{ij'}^w < d_i^w t_{ij}^w} (d_i^w t_{ij}^w - d_i^w t_{ij'}^w) x_{j'} - Z_w^*, \quad w \in \overline{W}, i \in V, j \in U, \\ \sum_{j \in U} x_j = p, \\ x_j \in \{0, 1\}, \quad j \in U. \end{array} \right. \quad (46)$$

Note that both (RF3) and (RF5) do not require an exponential number of variables, which is a significant advantage compared to (RF1), (RF2), and (RF4). However, the five reformulations have an exponential number of constraints.

3.4 Column-and-constraint generation algorithm

To solve these robust MILP formulations, we first propose the *column-and-constraint generation* algorithm (C&CG) represented in Algorithm 1 in which (\overline{RF}) can be any of our five robust formulations (RF1), (RF2), (RF3), (RF4), or (RF5) with \overline{W} initially empty.

At each iteration, Algorithm 1 generates a solution (x, RC) which satisfies all the scenarios currently in \overline{W} by solving (\overline{RF}) (Step 3). If the solution does not satisfy one of the scenarios $\{w_i(x)\}_{i \in V}$ (Step 10), the most violated scenario is added to \overline{W} (Step 14). When no violated scenario is found, an optimal solution is returned.

The value of the optimal radius considering a scenario $w_i(x)$ can be calculated by solving a deterministic (PCP) (Step 7). Note that the radius associated with a feasible solution x in a scenario w_i can be obtained quickly as it only requires to determine the distance between each demand node and its closest center in J_x (Step 8).

Algorithm 1: *column-and-constraint generation* algorithm

input :

- Instance data $(V, U, p, [d_i^-, d_i^+]$ and $[t_{ij}^-, t_{ij}^+]$ for each $i \in V$ and $j \in U$).
- A robust formulation (\overline{RF}) for (PCP) .

output :

- An optimal solution x of (\overline{RF}) and its robustness cost RC .

```
1  $RC \leftarrow 0, \overline{W} \leftarrow \emptyset, isOptimal \leftarrow false$ 
2 while  $isOptimal = false$  do
3    $(x, RC) \leftarrow$  solve  $(\overline{RF})$  with scenarios  $\overline{W}$ 
4    $isOptimal \leftarrow true$ 
5    $\overline{w} \leftarrow \emptyset$ 
6   for  $i \in V$  do
7      $Z^* \leftarrow$  optimal radius of the deterministic  $(PCP)$  for scenario  $w_i(x)$ 
8      $Z \leftarrow$  radius for  $x$  in scenario  $w_i(x)$ 
9      $DEV \leftarrow Z - Z^*$ 
10    if  $DEV > RC$  then
11       $isOptimal \leftarrow false$ 
12       $RC \leftarrow DEV$ 
13       $\overline{w} \leftarrow w_i(x)$ 
14     $\overline{W} \leftarrow \overline{W} \cup \{\overline{w}\}$ 
15 return  $x$  and  $RC$ 
```

3.5 Branch-and-cut algorithm

The main advantage of $(RF3)$, and $(RF5)$ over $(RF1)$, $(RF2)$, and $(RF4)$ is that no new variable is required when a scenario is added to \overline{W} . Consequently, using $(RF3)$ or $(RF5)$, we can define a *branch-and-cut* algorithm ($B\&C$) which checks, at each node of the search tree, if each obtained integer solution (x, RC) satisfies all the scenarios $\{w_i(x)\}_{i \in V}$. If it does not, we generate the corresponding violated inequalities and the solution is ignored by the solver. This can be performed through callbacks which is a feature provided by mixed integer programming solvers. Consequently, Steps 4-14 of Algorithm 1 are performed within a callback. This modification allows us to only generate a single search tree instead of solving (\overline{RF}) from scratch at each iteration of the while loop.

3.6 Solving the deterministic p-center problems

One of the most time consuming steps in these two algorithms is solving the deterministic (*PCP*) associated with the current feasible solution $x \in \Omega$ and scenario $w_i(x)$ in order to obtain $Z_{w_i(x)}^*$ for each $i \in V$ (Step 7 in Algorithm 1). Any of the formulations presented in Section 2.1 could be considered to solve these deterministic problems. However, to improve the performances, we consider the two following improvements.

3.6.1 Reducing the number of deterministic problems solved

For a given $i \in V$, if we know beforehand that a solution x satisfies scenario $w_i(x)$ (i.e., that $Z(w_i(x), x) - Z_{w_i(x)}^* \leq RC$), the solution of the associated deterministic (*PCP*) can be avoided. In particular, this is the case if we know a lower bound Z_{lb}^{*i} on $Z_{w_i(x)}^*$ such that $Z(w_i(x), x) - Z_{lb}^{*i} \leq RC$. Indeed, in that case we have $Z(w_i(x), x) - Z_{w_i(x)}^* \leq Z(w_i(x), x) - Z_{lb}^{*i} \leq RC$. Now, let us consider the scenarios defined as follows.

Definition 2 Let $\bar{i} \in V$. We define $w_{lb}^{\bar{i}}$ as the following scenario:

- $d_i^{w_{lb}^{\bar{i}}} = \begin{cases} d_i^+ & \text{if } i = \bar{i} \\ d_i^- & \text{otherwise} \end{cases}$
- $t_{ij}^{w_{lb}^{\bar{i}}} = t_{ij}^- \quad \forall j \in U$

The following lemma shows that the optimal value of the deterministic (*PCP*) associated to w_{lb}^i provides a lower bound on $Z_{w_i(x)}^*$.

Lemma 2 For any $i \in V$ and $x \in \Omega$, $Z_{w_{lb}^i}^* \leq Z_{w_i(x)}^*$.

Proof:

We know that $Z_{w_{lb}^i}^* \leq Z(w_{lb}^i, x^*(w_i(x)))$. Moreover, since the only difference between scenarios $w_i(x)$ and w_{lb}^i is a reduction of travel times $Z(w_{lb}^i, x^*(w_i(x))) \leq Z_{w_i(x)}^*$.

□

Note that scenarios $\{w_{lb}^i\}_{i \in V}$ do not depend on any feasible solution $x \in \Omega$ and therefore the $|V|$ corresponding lower bounds can all be computed in a pre-processing step. We observed that this improvement enables a significant reduction of the solving time.

3.6.2 Binary search

Solving deterministic (PCP) through MILP solvers is not the most efficient approach. To the best of our knowledge, the best methods are those of [Contardo et al., 2019] and [Gaar and Sinnl, 2022]. The former initially considers a subset of clients, solves the (PCP) associated to this subset, and adds new clients until an optimal solution is obtained. The latter method is based on a branch-and-cut algorithm, considering Benders cuts strengthened with a lifting procedure, and a specialized separation scheme. Both methods can solve very large-scale instances.

These two methods are sophisticated. In order to reach a good compromise between performance and ease of implementation, we solve the deterministic (PCP) through a binary search algorithm well known in the literature ([Toregas et al., 1971]). This algorithm is based on the fact that we can determine if the radius of a deterministic (PCP) is at most D^k by solving a set covering problem (SCP). In this context, the objective is to open as few centers as possible such that for each demand node $i \in V$ a site at distance at most D^k of i is opened:

$$(\mathit{SCP}(V, U, D^k)) \left\{ \begin{array}{ll} \min \sum_{j=1}^m x_j & \\ \sum_{j: d_i t_{ij} \leq D^k} x_j \geq 1 & i \in V \\ x_j \in \{0, 1\} & j \in U \end{array} \right. \text{ s.t.}$$

The radius of (PCP) is at most D^k if and only if the solution value of (SCP) is at most p . The smallest distance D^k that needs at most p centers is the optimal solution of the corresponding (PCP). To find the minimum distance D^k , a binary search is used so that at most $\log_2(K + 1)$ set cover problems need to be solved. This improvement can significantly reduce the execution time of our exact algorithms for ($RPCP_2$).

3.7 Adaptation to the single-stage problem

The ($C\&CG$) and ($B\&C$) algorithms previously presented can be adapted to solve the single-stage problem ($RPCP_1$) using ($RF1$). For this purpose, we only consider one single set of allocation variables y_{ij} which ensures that the demand node allocations are the same regardless of the scenario. These adaptations are not possible for the ($C\&CG$) and ($B\&C$) algorithms based on the others formulations. Indeed, in these formulations it is not possible to ensure that the demand node allocations are the same in all scenarios. Only considering one set of variables z_k in ($RF2$) or u_k

in (RF4) would only ensure that the distance index of the radius is the same in each scenario, not that the demand node allocations are. For the algorithms based on (RF3) or (RF5) the adaptation seems even less possible as these formulations do not contain scenario variables and as the demand node allocations are determined implicitly in the corresponding constraints.

Note that for (RPCP₁) the finite set of scenarios that a solution must satisfy to ensure its optimality is different from $\{w_i(x)\}_{i \in V}$ as proved in Theorem 1 of [Lu, 2013].

4 Computational study

We evaluate the efficiency of our (C&CG) and (B&C) algorithms on a case study presented in [Lu, 2013] and on randomly generated instances. It was not possible to make a direct comparison with [Lu, 2013], because the solution values presented in [Lu, 2013] are not consistent with the ones obtained by our exact solution approach. This is illustrated in A.

Our study was carried out on an Intel(R) Xeon(R) Gold 6144 processor 3,5 GHz, with 32 threads, but only 1 was used, and 378 GB of RAM. IBM ILOG CPLEX 20.1 was used as solver. For the (B&C) algorithm, we use the GenericCallback of CPLEX, which gets called whenever a feasible integer solution is found. We set the absolute tolerance to the best integer objective (EpGap) to 10^{-10} . All times presented in the tables are CPU times in seconds. All our instances files are available online¹.

4.1 Case Study

[Lu, 2013] presents a case study on the location of urgent relief distribution centers (URDCs) in a relief supply distribution network responding to the massive earthquake which hit central Taiwan on September 21, 1999. Specifically, relief supplies were collected from six unaffected counties transported to two URDCs, and then delivered to the 51 relief stations in the 11 townships. Five other candidate sites for URDCs were considered. They divided the number of survivors by the number of relief stations of each township to estimate the relief demand faced by each relief station. They use the data collected in previous research for the travel time between a URDC and a relief station.

Following [Lu, 2013], we consider a distance uncertainty box $[t_{ij}; t_{ij}(1 + \alpha_1)]$, and the demand uncertainty box $[d_i(1 - \alpha_2); d_i(1 + \alpha_2)]$ with $\alpha_1 \in \{0.5, 1.5, 2.5\}$ and $\alpha_2 \in \{0.2, 0.4, 0.6\}$. Thus, we have 9 combinations of parameter values α_1 and α_2 considering 51 demand nodes, 7 possible

¹<https://osf.io/87u6f/>

sites, and the selection of 2 centers. Table 3 shows the results obtained when solving ($RPCP_2$) by applying our algorithms to the case study.

Instance						Iterations							Time(s)						
V	U	p	α_1	α_2	RC	C&CG					B&C		C&CG					B&C	
						RF1	RF2	RF3	RF4	RF5	RF3	RF5	RF1	RF2	RF3	RF4	RF5	RF3	RF5
51	7	2	0.5	0.2	495,000	5	5	5	5	5	5	5	1.0	1.0	0.9	10.0	0.9	1.2	1.2
51	7	2	0.5	0.4	838,500	6	6	6	6	6	10	6	1.1	1.0	1.0	14.1	0.9	1.2	1.1
51	7	2	0.5	0.6	1,159,200	7	6	6	6	6	6	6	1.3	1.0	0.8	11.7	0.9	1.1	1.1
51	7	2	1.5	0.2	1,238,400	14	16	16	15	16	16	15	5.7	4.3	1.9	153.5	1.9	1.9	1.9
51	7	2	1.5	0.4	1,705,800	16	15	15	14	16	16	15	8.0	3.0	1.8	109.1	1.8	1.9	1.8
51	7	2	1.5	0.6	2,150,400	16	15	15	14	15	14	15	7.2	2.8	1.7	100.3	1.6	1.8	1.8
51	7	2	2.5	0.2	1,981,800	18	19	22	19	21	18	18	11.6	5.4	2.5	238.9	2.5	2.2	2.2
51	7	2	2.5	0.4	2,573,100	19	19	21	19	20	18	19	12.4	4.7	2.3	209.9	2.2	2.2	2.2
51	7	2	2.5	0.6	3,141,600	20	19	19	19	19	19	19	13.4	4.3	2.1	181.7	2.1	2.1	2.1
Average						13	13	14	13	14	14	13	6.9	3.0	1.6	114.3	1.6	1.7	1.7

Table 3: Results of the $C&CG$ and $B&C$ algorithms for ($RPCP_2$) on the case study instances.

One can observe in Table 3 that the optimal value of ($RPCP_2$) increases with α_1 and α_2 . The fastest algorithm is ($C&CG$) using the ($RF3$) and ($RF5$) formulations, closely followed by the two ($B&C$). Unlike the deterministic problem, ($RF2$) and ($RF4$) are not the most efficient. Moreover, ($RF4$) requires many more variables and constraints than the other formulations, resulting in a much longer solving time per iteration of the ($C&CG$) algorithm.

We also have observed that the robust solution of all these instances also is the deterministic solution which consists of opening sites 3 and 4. This may be due to the fact that the bounds of the uncertainty boxes are all increased in the same proportion for all the nodes. We will avoid this in the following by considering random intervals for all uncertain parameters.

4.2 Randomly generated instances

Following [Lu, 2013], two dimensional coordinates were uniformly drawn from $[0, 100] \times [40; 60]$ for the two set of demand nodes and available sites. The distances t_{ij} between demand nodes and sites were set to the nearest integer of their Euclidean distance. The demand d_i of each demand node $i \in V$ was uniformly drawn from the interval $[1,000; 2,000]$. Nevertheless, we do not consider the same uncertainty sets as in [Lu, 2013]. For all $i \in V$ and $j \in U$, t_{ij} and d_i can take any value in $[t_{ij}; t_{ij}(1 + \alpha_1^i)]$ and $[d_i(1 - \alpha_2^i); d_i(1 + \alpha_2^i)]$. We consider three cases depending on whether α_1^i and α_2^i are randomly generated in $[0.1, 0.3]$, $[0.4, 0.6]$, or $[0.7, 0.9]$.

Firstly, we generate 18 instances for ($RPCP_2$) with the following parameters: $|V| = |U| \in \{15, 25, 40\}$, and $p = \{2, 3\}$. The results obtained for these instances are presented in Tables 4, and 5 respectively.

The last columns represent the robust solution and the deterministic solutions. Since both problems may have several optimal solutions, we include in the tables the solutions that are the most frequent among those of the seven methods considered. For these instances, we consider a time-limit of 600 seconds (TL2).

Similarly to Table 3, both RC and the solving time increase with α_1 and α_2 in this first set of instances. (RF3) and (RF5) are also the fastest (C&CG) algorithm. This could be explained by the fact that the addition of a scenario does not lead to the addition of any variable. The (B&C) algorithms are much faster even though they perform more iterations. The better performances of (RF5) with the (B&C) algorithm is due to its smaller number of iterations.

We observe that our box uncertainty sets with random bounds lead to robust solutions which are different from the deterministic one, in contrast to the case study. Note that the CPU time increases with p . Indeed, none of the (C&CG) algorithms is able to solve all the instances for $p = 3$. This could be explained by the number of feasible solutions which is proportional to $\binom{m}{p}$.

Instance					Iterations								Time (s)								Robust Solution		Nominal Solution					
V	U	p	(α_1, α_2)	RC	C&CG				B&C				C&CG				B&C				Solution	Solution						
					RF1	RF2	RF3	RF4	RF5	RF3	RF5	RF1	RF2	RF3	RF4	RF5	RF3	RF5										
15	15	2	[0.1, 0.3]	45,870	5	8	6	5	5	8	8	0.5	1.4	0.4	8.7	0.4	0.4	0.5	10	14								
15	15	2	[0.4, 0.6]	136,587	17	16	15	14	17	21	19	6.1	3.2	0.6	31.1	0.7	0.5	0.5	11	14	11	14						
15	15	2	[0.7, 0.9]	228,429	34	28	33	33	33	37	35	26.4	3.4	1.3	144.5	2.0	1.6	1.1	11	14								
25	25	2	[0.1, 0.3]	40,205	8	8	8	8	8	16	12	3.1	6.6	1.0	184.9	1.0	1.2	1.1	21	22								
25	25	2	[0.4, 0.6]	142,096	21	20	21	16	22	32	30	41.6	33.3	3.0	TL	3.7	2.1	1.7	7	22	13	21						
25	25	2	[0.7, 0.9]	228,195	36	34	38	20	38	47	35	230.9	14.5	4.5	TL	5.3	2.2	1.9	13	15								
40	40	2	[0.1, 0.3]	54,082	12	12	13	3	11	19	14	51.4	58.8	4.1	TL	7.4	8.0	6.9	10	26								
40	40	2	[0.4, 0.6]	143,310	23	23	34	3	33	45	36	TL	TL	94.5	TL	63.0	18.0	14.7	25	26	8	26						
40	40	2	[0.7, 0.9]	283,119	30	56	89	6	96	122	108	TL	TL	67.6	TL	91.2	17.6	15.4	23	32								
Average					29				29				39				33				19.7		19.4		5.7		4.9	

Table 4: Results of the C&CG and B&C algorithms on randomly generated instances for (RPCP₂) with $|V| = |U| \in \{15, 25, 40\}$ and $p = 2$. TL2=600s.

Instance					Iterations								Time (s)								Robust Solution			Nominal Solution		
V	U	p	(α_1, α_2)	RC	C&CG				B&C				C&CG				B&C				Solution	Solution	Solution			
					RF1	RF2	RF3	RF4	RF5	RF3	RF5	RF1	RF2	RF3	RF4	RF5	RF3	RF5								
15	15	3	[0.1, 0.3]	56,689	8	7	8	8	8	10	11	0.7	0.7	0.4	15.5	0.5	0.4	0.4	1	6	8					
15	15	3	[0.4, 0.6]	97,360	16	15	16	15	17	24	16	3.6	5.1	0.8	59.7	1.0	0.7	0.5	1	6	12	1	6	12		
15	15	3	[0.7, 0.9]	166,252	33	36	35	37	36	44	43	19.6	14.4	1.5	200.3	2.0	1.0	1.0	1	6	12					
25	25	3	[0.1, 0.3]	40,950	12	8	7	7	10	16	8	6.7	4.7	0.9	157.3	1.7	1.1	0.7	2	12	14					
25	25	3	[0.4, 0.6]	84,827	16	16	18	13	18	45	21	18.6	27.3	6.3	TL	5.5	3.2	2.0	2	14	50	3	7	25		
25	25	3	[0.7, 0.9]	165,978	50	51	54	20	57	66	64	305.8	94.8	14.6	TL	17.9	3.9	4.1	6	7	17					
40	40	3	[0.1, 0.3]	48,123	18	17	15	3	18	40	33	213.1	TL	20.6	TL	22.7	9.4	8.5	25	26	40					
40	40	3	[0.4, 0.6]	125,024	23	19	54	4	52	197	103	TL	TL	389.7	TL	161.7	26.9	18.1	18	25	35	8	26	40		
40	40	3	[0.7, 0.9]	236,529	33	44	152	6	132	537	589	TL	TL	TL	TL	TL	71.2	74.9	25	29	40					
Average					109				99				13.1				12.2									

Table 5: Results of the C&CG and B&C algorithms on randomly generated instances for (RPCP₂) with $|V| = |U| \in \{15, 25, 40\}$ and $p = 3$. TL2=600s.

Since the limit of the (C&CG) algorithms is reached, we only focus on the (B&C) algorithms in the

following. We now consider a time limit of 9000 seconds and instances with the following parameters: $|V| = |U| \in \{60, 80, 100\}$, and $p = \{2, 3, 4, 5\}$. The results are presented in Tables 6, 7, 8, and 9.

Both algorithms were able to solve most instances. However, due to memory limitations, the optimal solution is not obtained for all instances with 80 and 100 nodes in particular for those with the largest uncertainty sets. The results of the two formulations are similar, with (*RF3*) performing better on average for $p = \{3, 4, 5\}$, and worse for $p = 2$. Their solving time is still closely related to the number of iterations even if for a few instances, (*RF3*) performs more iterations but is faster.

Instance					B&C				Robust Solution				Nominal Solution	
$ V $	$ U $	p	(α_1, α_2)	RC	Iterations		Time (s)		RF3		RF5			
					RF3	RF5	RF3	RF5						
60	60	2	[0.1, 0.3]	45,075	31	15	16	9	9	32	9	32		
60	60	2	[0.4, 0.6]	112,704	81	38	37	21	31	32	31	32	4	32
60	60	2	[0.7, 0.9]	216,104	176	73	66	37	32	45	32	45		
80	80	2	[0.1, 0.3]	37,245	21	18	25	26	2	61	2	61		
80	80	2	[0.4, 0.6]	147,825	117	62	113	74	27	61	27	61	54	61
80	80	2	[0.7, 0.9]	243,722	338	296	251	275	25	27	25	27		
100	100	2	[0.1, 0.3]	53,119	55	26	109	66	33	65	65	84		
100	100	2	[0.4, 0.6]	146,640	183	154	265	241	9	64	33	64	33	64
100	100	2	[0.7, 0.9]	246,597	460	497	630	716	64	84	64	84		
Average					162	131	168	163						

Table 6: Results of the *B&C* algorithm on randomly generated instances for (*RPCP*₂) with $|V| = |U| \in \{60, 80, 100\}$ and $p = 2$.

Instance					B&C				Robust Solution				Nominal Solution	
$ V $	$ U $	p	(α_1, α_2)	RC	Iterations		Time (s)		RF3		RF5			
					RF3	RF5	RF3	RF5						
60	60	3	[0.1, 0.3]	45,430	92	79	29	27	24	32	36	24	32	56
60	60	3	[0.4, 0.6]	117,659	354	310	127	127	10	42	44	10	33	38
60	60	3	[0.7, 0.9]	209,352	1015	1024	444	507	20	31	42	14	31	42
80	80	3	[0.1, 0.3]	33,439	55	48	83	95	32	41	58	32	41	58
80	80	3	[0.4, 0.6]	128,900	386	332	399	336	32	63	74	32	63	74
80	80	3	[0.7, 0.9]	206,001	980	1033	1295	1647	10	58	61	10	58	61
100	100	3	[0.1, 0.3]	35,780	52	58	122	175	12	76	84	12	76	84
100	100	3	[0.4, 0.6]	124,704	498	323	1403	1192	19	88	89	12	19	88
Average					429	401	488	513						

Table 7: Results of the *B&C* algorithm on randomly generated instances for (*RPCP*₂) with $|V| = |U| \in \{60, 80, 100\}$ and $p = 3$.

Instance					B&C				Robust Solution								Nominal Solution							
V	U	p	(α_1, α_2)	RC	Iterations		Time (s)		RF3				RF5											
					RF3	RF5	RF3	RF5																
60	60	4	[0.1, 0.3]	36,737	50	42	44	36	3	8	18	55	2	8	18	55	2 8 32 43							
60	60	4	[0.4, 0.6]	99,544	326	293	379	408	2	8	26	32	2	26	32	35								
60	60	4	[0.7, 0.9]	171,749	1684	1519	2918	3371	26	32	48	53	12	26	48	53								
80	80	4	[0.1, 0.3]	31,788	92	58	482	325	3	32	58	69	21	32	58	60	1 3 32 58							
80	80	4	[0.4, 0.6]	108,312	736	633	2152	3259	1	32	48	55	17	32	39	55								
100	100	4	[0.1, 0.3]	32,841	203	142	941	1816	12	14	35	44	12	14	35	73	1 4 12 35							
Average					515	448	1153	1536																

Table 8: Results of the $B\&C$ algorithm on randomly generated instances for $(RPCP_2)$ with $|V| = |U| \in \{60, 80, 100\}$ and $p = 4$.

Instance					B&C				Robust Solution								Nominal Solution							
V	U	p	(α_1, α_2)	RC / BKV	Iterations		Time (s)		RF3				RF5											
					RF3	RF5	RF3	RF5																
60	60	5	[0.1, 0.3]	29,709	52	28	51	31	2	8	12	18	37	3	8	18	18	37	2 8 12 18 40					
60	60	5	[0.4, 0.6]	86,466	386	185	970	774	2	8	18	28	38	2	8	28	28	38						
60	60	5	[0.7, 0.9]	138,852	888	1323	5596	6461	2	8	21	38	39	2	21	38	38	39						
80	80	5	[0.1, 0.3]	29,259	147	129	3086	2366	1	44	48	58	59	1	44	58	58	59	11 15 16 18 40					
80	80	5	[0.4, 0.6]	99,449	643	547	TL3	TL3	2	11	42	47	73	2	11	47	31	73						
100	100	5	[0.1, 0.3]	31,107	142	133	803	6066	13	14	90	95	100	13	14	95	95	100	3 12 14 50 85					
Average					372	395	2011	2747																

Table 9: Results of the $B\&C$ algorithm on randomly generated instances for $(RPCP_2)$ with $|V| = |U| \in \{60, 80, 100\}$ and $p = 5$. TL3=9000s.

4.3 ORLIB instances

We consider the deterministic ORLIB instances ([Beasley, 1990]), which are symmetrical instances, since the set of clients is also the set of candidate sites. In these instances the travel times t_{ij} are provided. To create ($RPCP_2$) instances, we randomly generate demand values in $[1, 100]$ and we consider for the travel times the same box uncertainty set than in the previous section, with α_i and α_{ij} randomly generated in $[0.1, 0.9]$ and such that $\alpha_{ij} = \alpha_{ji}$ for $i \in V$ and $j \in U$. The results are presented in Tables 10, 11, and 12. For these instances, we consider a time-limit of 9000 seconds (TL3).

Starting from $p = 2$, we can see that the solving time increases rapidly with p . Particularly, instance pmed3 is only solved for $p = 2$ and instance pmed1 could not be solved for $p = 4$ due to memory issues. As for the random instances, the formulations have similar performances which depend strongly on the number of iterations. We were unable to solve larger ORLIB instances with $|V| \geq 200$ due to memory limitations.

Instance					B&C				Robust Solution				Nominal Solution	
Name	$ V = U $	p	(α_1, α_2)	RC	Iterations		Time (s)		RF3		RF5			
					RF3	RF5	RF3	RF5	RF3	RF5	RF3	RF5		
pmed1	100	2	[0.1, 0.9]	14,706	392	494	511	663	8	57	8	57	60	95
pmed2	100	2	[0.1, 0.9]	16,138	595	626	672	738	23	64	23	64	7	21
pmed3	100	2	[0.1, 0.9]	25,443	717	633	825	758	15	88	15	88	89	99
pmed4	100	2	[0.1, 0.9]	19,293	761	778	1070	1134	32	100	32	99	2	98
pmed5	100	2	[0.1, 0.9]	13,336	267	186	274	193	52	68	52	68	15	52
Average					546	543	670	697						

Table 10: Results of the $B&C$ algorithm on adapted ORLIB instances for ($RPCP_2$) with $|V| = |U| = 100$ and $p = 2$.

Instance					B&C				Robust Solution				Nominal Solution				
Name	$ V = U $	p	(α_1, α_2)	RC	Iterations		Time (s)		RF3		RF5						
					RF3	RF5	RF3	RF5	RF3	RF5	RF3	RF5					
pmed1	100	3	[0.1, 0.9]	14,094	675	587	2876	2188	9	42	94	9	42	64	9	13	94
pmed2	100	3	[0.1, 0.9]	17,718	567	615	1137	1245	35	65	88	35	68	88	2	9	22
pmed4	100	3	[0.1, 0.9]	15,423	395	459	1303	1413	32	88	99	32	51	99	1	69	87
pmed5	100	3	[0.1, 0.9]	12,228	601	536	1348	1346	51	68	97	51	68	97	7	32	46
Average					560	549	1666	1548									

Table 11: Results of the $B&C$ algorithm on adapted ORLIB instances for ($RPCP_2$) with $|V| = |U| = 100$ and $p = 3$.

Instance					B&C				Robust Solution				Nominal Solution							
<i>name</i>	$ V = U $	p	(α_1, α_2)	RC	Iterations		Time (s)		RF3				RF5							
					RF3	RF5	RF3	RF5												
pmed2	100	4	[0.1, 0.9]	12,600	1032	1134	3457	4980	23	58	77	97	23	58	76	97	2	8	22	40
pmed4	100	4	[0.1, 0.9]	13,812	507	648	6299	4463	7	26	52	99	7	26	52	99	8	23	51	55
pmed5	100	4	[0.1, 0.9]	11,650	1150	-	5563	-	60	71	90	97	-	-	-	-	23	51	69	97

Table 12: Results of the $B\&C$ algorithm on adapted ORLIB instances for $(RPCP_2)$ with $|V| = |U| = 100$ and $p = 4$. TL3=9000s.

5 Conclusions

The weighted vertex p -center problem consists of locating p facilities among a set of potential sites such that the maximum weighted distance from any demand node to its closest located facility is minimized. The incorporation of uncertain information within a robust optimization approach allows us to solve practical emergency logistics problems. However, the robust counterpart of such problems is even more difficult. Therefore, most studies propose a heuristic approach instead of an exact solution approach.

We studied the solution of a robust weighted vertex p -center problem, considering uncertain nodal weights demand and edge lengths using box uncertainty sets. Two variants of this problem are possible depending on whether the demand node allocations to the centers are made after the uncertainty is revealed ($RPCP_2$) or not ($RPCP_1$).

Similarly to ($RPCP_1$), we prove that for ($RPCP_2$) a finite subset of scenarios from the box uncertainty set can be considered without losing optimality. We use this result to propose five robust reformulations based on different MILP formulations of the vertex p -center problem. To optimally solve these reformulations, we introduce a *column-and-constraint generation* algorithm and a *branch-and-cut* algorithm. Moreover, we identify a lower bound on the optimal value of the deterministic p -center problem associated with the identified finite subset of scenarios. We use this result to significantly reduce the solving time of our algorithms. Finally, we highlight how our methods can be adapted to optimally solve ($RPCP_1$).

We present a numerical study to compare the performances of the algorithms on a case study, on randomly generated instances, and on a few instances from ORLIB. We are able to solve optimally the 68 considered instances. The *column-and-constraint generation* algorithm based on formulation ($RF3$) and ($RF5$) is more efficient than the one based on ($RF1$), ($RF2$) and ($RF4$). This is because adding a scenario does not require the addition of any variable. This formulation enables the implementation of a *branch-and-cut* algorithm which significantly reduces the solving time.

In future work, analysis of larger instances with other random box uncertainty sets could be considered. To further improve the performances of the *branch-and-cut* algorithm, alternative branching strategies could be evaluated and integrality cuts (UserCuts) could be dynamically generated. The algorithms could also be improved by solving the deterministic (PCP) at each iteration with some other exact method such as that of [Contardo et al., 2019] or [Gaar and Sinnl, 2022].

Acknowledgments

This work was funded by the National Agency for Research and Development of Chile - ANID (Scholarship Phd. Program 2019-72200492).

The authors would like to thank our anonymous reviewers for their helpful comments.

A Comparison of results with Lu (2013)

We were not able to compare the performances of our exact algorithms and the one of the heuristic in [Lu, 2013] as its results do not seem to be correct. We prove that several robustness costs obtained with this heuristic and reported in [Lu, 2013] are undervalued.

Since we consider problem ($RPCP_1$), the demand node allocations are fixed before the uncertainty is revealed. Consequently, unlike ($RPCP_2$), allocation variables y are necessary to compute the radius and the robustness cost of a solution. Given a feasible solution (x, y) , let t_i^w be the distance (or travel time) in scenario $w \in W$ between a demand node $i \in V$ and its allocated center (i.e., $t_i^w = t_{ij}^w$ with $j \in U$ the only center such that $y_{ij} = 1$). In this single-stage problem, the radius of solution (x, y) is $\max_{i \in V} d_i^w t_i^w$ and its robustness cost is $RC(x, y) = \max_{w \in W} \left\{ \max_{i \in V} d_i^w t_i^w - Z_w^* \right\}$, where Z_w^* is the optimal solution of the deterministic p -center problem in which the uncertain parameters take value w .

A difficulty to evaluate the robustness costs reported in [Lu, 2013] is that for each solution only one demand node allocation is provided. Let us consider the instance described in Subsection 4.1 in which $p = 2$, $\alpha_1 = 0.5$, and $\alpha_2 = 0.2$ and let (x^h, y^h) be its associated solution in [Lu, 2013]. The only information available on (x^h, y^h) is that centers 1 and 2 are located ($x_1^h = x_2^h = 1$) and that demand node 21 is allocated to center 1 ($y_{21,1}^h = 1$). Nevertheless, this is sufficient to obtain a lower bound on the robustness cost as for any scenario $w \in W$ and any demand node $i \in V$, the expression $d_i^w t_i^w - Z_w^*$ constitutes a lower bound of $RC(x^h, y^h)$. Let us consider a scenario w_{21} in which the demand of node 21 is $d_{21}^w = d_{21}^+ = 34,800$ and its distance to center 1 is $t_{21,1}^w = t_{21,1}^+ = 41$. The optimal radius $Z_{w_{21}}^* = 495,600$ is obtained by solving a deterministic p -center problem. Consequently, a lower bound on the robustness cost of value $34,800 \times 41 - 495,600 = 931,200$ is obtained, which is higher than the value 93,619 reported in [Lu, 2013].

Table 13 present similar results on nine instances. The third column contains the robustness costs reported in [Lu, 2013] which are almost all undervalued. Indeed, they are significantly lower than their associated lower bounds presented in Column 4. Column 5 contains the robustness cost of optimal solutions obtained by our ($B\&C$) algorithm adapted to ($RPCP_1$) (see Section 3.7). Note that the branch-and-cut always returns a solution which robustness cost is always lower than the lower bound of the heuristic solution.

Instance		Robustness cost		
		Heuristic solution from [Lu, 2013]		Optimal solution
α_1	α_2	Results from [Lu, 2013]	According to this article	
0.5	0.2	= 93,619	\geq 931,200	= 495,000
0.5	0.4	= 587,837	\geq 1,292,900	= 838,500
0.5	0.6	= 1,477,709	\geq 1,906,600	= 1,159,200
1.5	0.2	= 940,858	\geq 1,870,800	= 1,238,400
1.5	0.4	= 1,859,069	\geq 2,389,100	= 1,705,800
1.5	0.6	= 2,934,605	\geq 3,362,600	= 2,150,400
2.5	0.2	= 1,883,309	\geq 2,810,400	= 1,981,800
2.5	0.4	= 3,400,291	\geq 4,029,900	= 2,573,100
2.5	0.6	= 4,356,480	\geq 4,129,600	= 3,141,600

Table 13: Comparison of the robustness cost of solutions obtained by the heuristic presented in [Lu, 2013] and optimal solutions obtained by the branch-and-cut algorithm adapted for ($RPCP_1$).

References

- [Ales and Elloumi, 2018] Ales, Z. and Elloumi, S. (2018). Compact milp formulations for the p-center problem. In *International Symposium on Combinatorial Optimization*, pages 14–25. Springer.
- [An et al., 2014] An, Y., Zeng, B., Zhang, Y., and Zhao, L. (2014). Reliable p-median facility location problem: two-stage robust models and algorithms. *Transportation Research Part B: Methodological*, 64:54–72.
- [Averbakh and Berman, 1997] Averbakh, I. and Berman, O. (1997). Minimax regret p-center location on a network with demand uncertainty. *Location Science*, 5(4):247–254.
- [Averbakh and Berman, 2000] Averbakh, I. and Berman, O. (2000). Algorithms for the robust 1-center problem on a tree. *European Journal of Operational Research*, 123(2):292–302.
- [Baron et al., 2011] Baron, O., Milner, J., and Naseraldin, H. (2011). Facility location: A robust optimization approach. *Production and Operations Management*, 20.
- [Beasley, 1990] Beasley, J. E. (1990). Or-library: Distributing test problems by electronic mail. *The Journal of the Operational Research Society*, 41(11):1069–1072.
- [Ben-Tal et al., 2009] Ben-Tal, A., Ghaoui, L., and Nemirovski, A. (2009). *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press.

- [Bertsimas et al., 2013] Bertsimas, D., Litvinov, E., Sun, X. A., Zhao, J., and Zheng, T. (2013). Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63.
- [Bertsimas and Sim, 2004] Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.
- [Çalik et al., 2019] Çalik, H., Labbé, M., and Yaman, H. (2019). p-center problems. In *Location science*, pages 51–65. Springer.
- [Calik and Tansel, 2013] Calik, H. and Tansel, B. C. (2013). Double bound method for solving the p-center location problem. *Computers & Operations Research*, 40(12):2991–2999.
- [Chan et al., 2018] Chan, T. C. Y., Shen, Z.-J. M., and Siddiq, A. (2018). Robust defibrillator deployment under cardiac arrest location uncertainty via row-and-column generation. *Operations Research*, 66(2):358–379.
- [Cheng et al., 2021] Cheng, C., Adulyasak, Y., and Rousseau, L.-M. (2021). Robust facility location under demand uncertainty and facility disruptions. *Omega*, 103:102429.
- [Contardo et al., 2019] Contardo, C., Iori, M., and Kramer, R. (2019). A scalable exact algorithm for the vertex p-center problem. *Computers and Operations Research*, 103:211–220.
- [Cornuejols et al., 1980] Cornuejols, G., Nemhauser, G. L., and Wolsey, L. A. (1980). A canonical representation of simple plant location problems and its applications. *SIAM Journal on Algebraic Discrete Methods*, 1(3):261–272.
- [Correia and Saldanha-da Gama, 2019] Correia, I. and Saldanha-da Gama, F. (2019). *Facility Location Under Uncertainty*, pages 185–213. Springer International Publishing, Cham.
- [Daskin, 1996] Daskin, M. (1996). Network and discrete location: Models, algorithms and applications. *Journal of the Operational Research Society*, 48.
- [Demange et al., 2020] Demange, M., Gabrel, V., Haddad, M. A., and Murat, C. (2020). A robust p-center problem under pressure to locate shelters in wildfire context. *EURO Journal on Computational Optimization*, 8:103–139.
- [Du and Zhou, 2018] Du, B. and Zhou, H. (2018). A robust optimization approach to the multiple allocation p-center facility location problem. *Symmetry*, 10(11):588.

- [Du et al., 2020] Du, B., Zhou, H., and Leus, R. (2020). A two-stage robust model for a reliable p-center facility location problem. *Applied Mathematical Modelling*, 77:99–114.
- [Elloumi et al., 2004] Elloumi, S., Labbé, M., and Pochet, Y. (2004). A New Formulation and Resolution Method for the p-Center Problem. *INFORMS Journal on Computing*, 16(1):84–94.
- [Gaar and Sinnl, 2022] Gaar, E. and Sinnl, M. (2022). A scaleable projection-based branch-and-cut algorithm for the p-center problem. *European Journal of Operational Research*.
- [Hakimi, 1965] Hakimi, S. L. (1965). Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475.
- [Hasani and Mokhtari, 2018] Hasani, A. and Mokhtari, H. (2018). Redesign strategies of a comprehensive robust relief network for disaster management. *Socio-Economic Planning Sciences*, 64:92–102.
- [Lu, 2013] Lu, C.-C. (2013). Robust weighted vertex p-center model considering uncertain data: An application to emergency management. *European Journal of Operational Research*, 230(1):113–121.
- [Lu and Sheu, 2013] Lu, C.-C. and Sheu, J.-B. (2013). Robust vertex p-center model for locating urgent relief distribution centers. *Computers & Operations Research*, 40(8):2128–2137.
- [Magnanti and Wong, 1981] Magnanti, T. L. and Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484.
- [Minieka, 1970] Minieka, E. (1970). The m-center problem. *SIAM Review*, 12(1):138–139.
- [Nikoofal and Sadjadi, 2010] Nikoofal, M. E. and Sadjadi, S. J. (2010). A robust optimization model for p-median problem with uncertain edge lengths. *The International Journal of Advanced Manufacturing Technology*, 50:391–397.
- [Paul and Wang, 2015] Paul, J. A. and Wang, X. J. (2015). Robust optimization for united states department of agriculture food aid bid allocations. *Transportation Research Part E: Logistics and Transportation Review*, 82:129–146.
- [Paul and Wang, 2019] Paul, J. A. and Wang, X. J. (2019). Robust location-allocation network design for earthquake preparedness. *Transportation research part B: methodological*, 119:139–155.

- [Rahmaniani et al., 2017] Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801 – 817.
- [Serra and Marianov, 1998] Serra, D. and Marianov, V. (1998). The p-median problem in a changing network: the case of barcelona. *Location Science*, 6(1-4):383–394.
- [Sheu, 2007] Sheu, J.-B. (2007). An emergency logistics distribution approach for quick response to urgent relief demand in disasters. *Transportation Research Part E: Logistics and Transportation Review*, 43(6):687–709.
- [Snyder, 2006] Snyder, L. V. (2006). Facility location under uncertainty: a review. *IIE transactions*, 38(7):547–564.
- [Toregas et al., 1971] Toregas, C., Swain, R., ReVelle, C., and Bergman, L. (1971). The location of emergency service facilities. *Operations research*, 19(6):1363–1373.
- [Trivedi and Singh, 2017] Trivedi, A. and Singh, A. (2017). A hybrid multi-objective decision model for emergency shelter location-relocation projects using fuzzy analytic hierarchy process and goal programming approach. *International Journal of Project Management*, 35(5):827–840.
- [Trivedi and Singh, 2019] Trivedi, A. and Singh, A. (2019). Shelter planning for uncertain seismic hazards using multicriteria decision approach: a case of nepal earthquake. *Journal of Multi-Criteria Decision Analysis*, 26(3-4):99–111.
- [Zeng and Zhao, 2013] Zeng, B. and Zhao, L. (2013). Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461.