Stratified p-Center Problem with Capacity Constraints and Failure Foresight

Antonin Carpentier[1], Laure Brisoux-Devendeville[1], Corinne Lucet[1], Rui Sá Shibasaki[1] and Sami Cherif[1]

***Abstract*— This paper presents a new variant of the p-center problem that incorporates capacity, stratification and failure prediction constraints in the context of optimizing the location of healthcare centers. Here, the demand points are assigned to a main center but can be served by a backup center if the main one fails. Additionally, different types of demand, organized into services, share the same set of p centers. The objective is to minimize the maximum distance between demand points and their assigned centers. A mixed integer linear programming model is proposed and used to solve this problem with the CPLEX solver. Three objective functions are explored separately. The first minimizes the maximum distance to the backup center. The second, minimizes the distances to both the main and backup centers. The last minimizes the maximum distance to the main center. The aim of this study is to compare the structure of the solutions developed and the computation times between these objectives.**

I. Introduction

The p-center is a well-known NP-hard problem [1] that aims to locate p centers so as to minimize the maximum distance between the demand points and their nearest center [2]. This is a classical facility location problem which arises in various domains, but the area of application, with our primary focus being the location of healthcare centers. However, in real-world contexts this problem usually presents additional features. Several studies in the literature introduce variants [3] that incorporate some of these features, such as capacity constraints [4], or the inclusion of existing centers [5]. It is worth noting that these variants typically incorporate only one additional feature at a time, remaining far removed from practical needs. In an attempt to get closer to the complexity found in real-world applications, we propose a new variant of the p-center problem combining capacity constraints [6], stratification aspects [7] and failure foresight [8] ensuring that each demand point is assigned both a main center and a backup center.

The contributions of this paper is twofold. The first contribution is a Mixed-Integer Linear Programming (MILP) formulation of this new problem. The second contribution is a comparison between different objective functions: the first, inspired by the work of Espejo *et al.* [8], aims to minimize the maximal distances between the backup centers and demand points; the second is new to the literature and aims to minimize the sum of the maximum distances between demand points and both main and backup centers; while the last serves as a reference and simply aims to minimize, for each service, the maximal distance between a city and its main service center. Our comparative analysis focuses on two major aspects: the structural differences of solutions obtained by the introduced approaches and their computational performances.

The paper is organized as follows. In Section II, we define the extended variant of p-center, accounting for stratification, capacity constraints and failure foresight. In section III, we introduce a MILP formulation of the problem based on the Daskin [9] model for the classic p-center problem. In Section IV, we report the experimental results comparing the objective functions in terms of structure of the solutions and computational performance. Finally, we conclude and discuss future work in Section V.

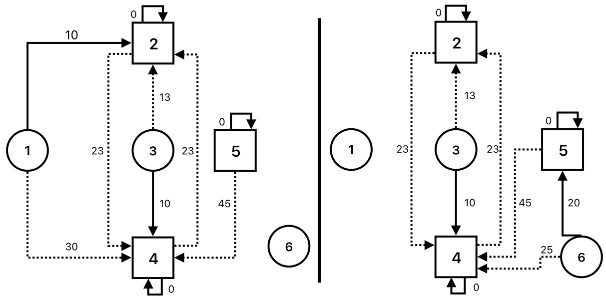
II. Problem Definition and Notations

We formalize the problem using the context of locating healthcare centers within a given region. The region is modeled as a graph $G = (N, E)$, where the set of nodes N (with $n = |N|$) represents cities. Each city has stratified healthcare demands. These demand strata correspond to different types of healthcare services, such as pediatrics, geriatrics, or gynecology. The demand in each city varies across these services, reflecting diverse population needs. The set S contains all the possible services (strata). We denote $q_{is} \geq 0$ the amount of service $s \in S$ demanded by city $i \in N$ and $S_i \subseteq S$ the subset of services such that $q_{is} \neq 0$.

Each city is also a potential healthcare center location with varying capacities for each service. We denote $u_{is} \geq 0$ the potential capacity of the city i to provide the service s and $C_i \subseteq S$ the subset of services such that $u_{is} \neq 0$. The set $N^s \subseteq N$ contains the cities $i \in N$ such that $q_{is} \neq 0$, designating the set of cities demanding the service s. The set of edges E represents connections between cities, each edge being assigned a weight representing the distance or cost of travel. The minimum distance between any two cities i and j is denoted by d_{ij}. Each city i must be associated with two centers to satisfy each of their demands: a main center and a backup center for each of service $s \in S$. Both the main and backup centers must be able to completely satisfy the demand of their assigned demand points. Moreover, for each city, the main center must be closer than the backup center for all services.

The objective is to locate p healthcare centers within the cities (i.e., the nodes of the graph) so as to minimize the maximum distance between a city and its main and/or backup centers, while respecting the constraints. We thus define:

[1], MIS UR 4290, Computer Science, University of Picardy Jules Verne, 80000 Amiens, France
antonin.carpentier1@etud.u-picardie.fr, {laure.devendeville, corinne.lucet, rui.sa.shibasaki, sami.cherif}@u-picardie.fr



(a) Service X (b) Service Y

Fig. 1: Example with 6 cities and 2 services

- $A^s = \max_{i \in N}\{d_{ik} \mid k$ is the main center of city i for service $s\}$, the maximum distance between cities and their main center for healthcare service $s \in S$;
- $B^s = \max_{i \in N}\{d_{ik} \mid k$ is the backup care center of city i for service $s\}$, the maximum distance between cities and their backup center for healthcare service $s \in S$;

Example 1. We consider a graph G composed of 6 cities and 2 services X and Y. Cities $1,2,3,4$ and 5 demand service X (see Fig. 1a) while service Y is only requested by cities $2,3,4,5$ and 6 (see Fig. 1b). The demand for each city in both services is set to 1, while the capacity of healthcare center is set to 20 per service. Assuming $p = 3$, a feasible solution is shown in Fig. 1 where centers are drawn as squares, an assignment to a main center is designated by a plain arc and an assignment to a backup center is designated by a dotted arc. For instance, for service Y, node 5 is the main center for city 6 and itself and node 4 is the backup center for cities 2, 5 and 6. In this example, the largest assigned distance between a city and its main center is 10 for service X and 20 for Y. Besides, the largest assigned distance between a city and its backup center is 45 for both services.

We consider different objective functions. The first one, denoted f and defined in equation (1), is inspired from [8] and aims to minimize, for each service, the maximum distance assigned between a city and its backup center. Espejo *et al.* justify this choice of aiming to minimize the distance to the second closest center (*i.e.*, backup center) by stating that *"in case of an accident which produces a damage in one of the emergency services, all the sites should still be inside the smallest possible radius around an available center"* [8].

$$f = \sum_{s \in S}(B^s) \tag{1}$$

The second objective function g is introduced in equation (2) and aims to minimize not only the maximal distance between a city and its backup center, but also the maximal distance between a city and its main center, for each service. This objective is motivated by the fact that, while the distance to the backup center is important in the event of failure of the main center, the distance to the main center also plays a very important role in everyday life. Therefore, we introduce the terms A^s in the objective function to account for both aspects.

$$g = \sum_{s \in S}(A^s + B^s) \tag{2}$$

Finally, we introduce a third objective function h, defined in equation (3), which aims to minimize, for each service, the maximal distance between a city and its main service center. The function h is introduced solely to provide an additional reference for comparing the functions f and g, particularly concerning the lower bounds of the variables A^s for each service $s \in S$.

$$h = \sum_{s \in S}(A^s) \tag{3}$$

Example 2. We consider the same graph G as in Example 1 showcased in Fig. 1. Computing the optimal solution with the objective function f gives us the following values: $\sum_{s \in S}(A^s) = 40$, $\sum_{s \in S}(B^s) = 90$ and $\sum_{s \in S}(A^s + B^s) = 130$. For the function g, we obtain: $\sum_{s \in S}(A^s) = 30$, $\sum_{s \in S}(B^s) = 90$ and $\sum_{s \in S}(A^s + B^s) = 120$. As for the the objective function h, the results are: $\sum_{s \in S}(A^s) = 30$, $\sum_{s \in S}(B^s) = 105$ and $\sum_{s \in S}(A^s + B^s) = 135$.

III. Formulation

The problem described in Section II can be formulated by Mixed Integer Linear Programming (MILP), based on the classic p-center problem model of Daskin [9]. With this purpose, the following decision variables are defined:

$$x_{ijs} = \begin{cases} 1 & \text{if } j \text{ is the main center for city } i \text{, for service } s, \\ 0 & \text{otherwise. } \forall i, j \in N,\ s \in S \end{cases}$$

$$w_{ijs} = \begin{cases} 1 & \text{if } j \text{ is the backup center for city } i \text{, for service } s, \\ 0 & \text{otherwise. } \forall i, j \in N,\ s \in S \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if a center is opened in city } j \\ 0 & \text{otherwise. } \forall j \in N \end{cases}$$

Besides, we have the secondary integer variables A^s and B^s for each service $s \in S$, described previously. Using these variables, the MILP model is defined as follows:

$$\text{Min } F \tag{4}$$

$s.c$

$$\sum_{j \in N} y_j = p \tag{5}$$

$$\sum_{\substack{j \in N \\ s \in C_j}} x_{ijs} = 1 \qquad \forall i \in N,\ \forall s \in S_i \tag{6}$$

$$\sum_{\substack{j \in N \\ s \in C_j}} w_{ijs} = 1 \qquad \forall i \in N,\ \forall s \in S_i \tag{7}$$

$$x_{ijs} \leq y_j \qquad \forall i, j \in N,\ s \in S_i \cap C_j \tag{8}$$

$$w_{ijs} \leq y_j \qquad \forall i, j \in N,\ s \in S_i \cap C_j \tag{9}$$

$$x_{ijs} + w_{ijs} \leq 1 \qquad \forall i, j \in N,\ s \in S_i \cap C_j \tag{10}$$

$$\sum_{\substack{j \in N \\ s \in C_j}} d_{ij} \cdot x_{ijs} \leq A^s \qquad \forall i \in N,\ s \in S_i \tag{11}$$

$$\sum_{\substack{j\in N\\ s\in C_j}} d_{ij}\cdot w_{ijs} \le B^s \qquad \forall i\in N,\ s\in S_i \quad (12)$$

$$\sum_{\substack{j\in N\\ s\in C_j}} d_{ij}\cdot x_{ijs} \le \sum_{\substack{j\in N\\ s\in C_j}} d_{ij}\cdot w_{ijs} \qquad \forall i\in N,\ s\in S_i \quad (13)$$

$$\sum_{\substack{i\in N\\ s\in S_i}} q_{is}\cdot x_{ijs} + \sum_{\substack{i\in N\\ s\in S_i}} q_{is}\cdot w_{ijs} \le u_{js}\cdot y_{js} \qquad \forall j\in N,\ s\in C_j \quad (14)$$

$$x_{ijs}\in\{0,1\} \qquad \forall i,j\in N,\ s\in S_i\cap C_j \quad (15)$$

$$w_{ijs}\in\{0,1\} \qquad \forall i,j\in N,\ s\in S_i\cap C_j \quad (16)$$

$$y_j\in\{0,1\} \qquad \forall j\in N \quad (17)$$

$$A^s, B^s \ge 0 \qquad \forall s\in S \quad (18)$$

The objective function F in (4) can be either f, g or h, respectively defined in equations (1), (2) and (3). Constraint (5) limits the number of open centers to p. To ensure that cities are assigned to only one main center (resp. backup center) for each service, we introduce the constraint (6) (resp. constraint (7)). Constraints (8) and (9) require cities to connect only to centers that are necessarily open. Constraint (10) prevents center j from being both a main center and a backup center for the same city i for a given service s. Constraints (11) and (12) determine the upper bounds of the distances between a city and its associated main center (respectively backup center). Constraint (13) requires the distance to the backup center to be greater than or equal to the distance to the main center. Finally, constraint (14) stipulates that the total demand of service s must not exceed the capacity of a center for this service. We note that the capacities of the main and backup centers are combined into a single constraint, otherwise we would have over-capacities. In this model the number of variables and constraints is in $O(|S|\cdot|N|^2)$.

IV. Experimental Evaluation

We conducted computational experiments to evaluate both solutions' structure and time performances for each objective function (i.e., f, g and h). The program is coded in C++ and executed on a single thread of an Intel Xeon E5-2620 v4 2.10GHz processor with 64 GB of RAM, running under Debian 12 (bookworm, 64 bit). IBM ILOG Cplex 22.1.1 was used to solve the MILP models, with the feasibility-before-optimality setting enabled. A time limit of 3600 seconds was set for each test.

A. Benchmark Instances

We considered 5 sets of instances: the first four (D_1, D_2, D_3 and D_4) are adapted from the literature, and the fifth D_5 is a set of new small instances. Following the literature, in all the test sets, each vertex is both a demand point and a candidate location for a center. We have enriched the D_1, D_2, D_3 and D_4 instance sets, which initially comprised a single service s_0, by varying the number of services $|S|$ within $\{3,5,10,15,20\}$.

To guarantee service diversity, capacities and demands are taken from s_0 data, respectively denoted q^0_{is} and u^0_{is}. For each city $i\in N$ and each service $s\in S$, $q_{is}=q^0_{is}.l_{is}$ and $u_{is}=u^0_{is}.m_{is}$, where l_{is} and m_{is} are random numbers in $[0.8,1.2]$.

The presence of demand and/or capacity for service s in city $i\in N$ is randomly set. Let $a_i\in]0,1[$ be a real random number associated to city i. We independently draw two random numbers b_{is} and $b'_{is}\in[0,1[$. If $b_{is}<a_i$, demand on service s is present. In a same way, if $b'_{is}<a_i$ capacity on service s is present. Hereafter, we summarize the providence and features of each benchmark set:

- **Set** D_1: Proposed by Lorena and Senne [10] for the Capacitated p-median problem. This set contains six instances with $|N|$ varying from 100 to 402 and p varying from 10 to 40. The capacities of these instances are homogeneous and the distances are euclidean. In total, we have 7 feasible stratified instances.
- **Set** D_2: This set comprises eight instances by Scaparra *et al.* [6], derived from two distinct graphs containing either $|N|=100$ or $|N|=150$ vertices (i.e., cities) and with non-Euclidean integer distances. From each graph, four instances with homogeneous capacities and four with heterogeneous capacities were created by selecting p in $\{5,15\}$. We have a total of 31 feasible stratified instances.
- **Set** D_3: Set from the OR-Lib (pmed) for the capacitated p-median problem. The set contains 40 instances ranging from $|N|=100$ vertices (i.e., cities) to $|N|=800$ and p ranging from 5 to $|N|/3$. These instances have edges weighted according to geometric distances (triangular inequalities). We have 54 realizable stratified instances.
- **Set** D_4: Set from the OR-Lib (pmedcap1) for the p-median problem. This set is composed of 2 groups of 10 instances with equal capacities. The first group has $|N|=50$ and $p=5$ while the second has $|N|=100$ and $p=10$. We have a total of 99 feasible stratified instances.
- **Set** D_5: We generate a new set of 42 statified instances based on the instances in Scaparra *et al.* [6]. These instances contain $|N|\in\{20,25,30,35,40,45,50\}$ nodes and $p\in\{\frac{|N|}{10},\frac{|N|}{5},\frac{|N|}{3}\}$ centers. In addition, for each instance, we consider $|S|\in\{\frac{|N|}{10},\frac{|N|}{5}\}$ number of different strata. For each node $i\in N$ and for each stratum $s\in S$, the demands q_{is} are equal to 1 and are the same for every node. The capacities have different values ranging from $|N|$ to $2\times|N|$.

In our experiments, we solve all the instances described above with the different objective functions f, g and h (respectively defined in equations (1), (2) and (3)) and we compare the computational time and the solution structures between them. In our following analysis, we distinguish between instances for which the three objective functions simultaneously find optimality in less than 3600 seconds and instances for which they do not (see Tables I, II and IV). For the sake of simplicity, we will denote D_i^{sim} ($i\in\{1,\ldots,5\}$) the instances in the set D_i for which f, g and h have simultaneously found optimality and $D_i^{dif}=D_i\setminus D_i^{sim}$ the set of instances for which they have not. We note that

$D_1^{sim} = D_2^{sim} = D_5^{dif} = \emptyset$, i.e., the three objective functions did not manage to solve simultaneously to optimality any instance in D_1 and D_2 whereas they managed to do so for all the instances in D_5.

B. Comparison of Solution Structures

First, we focus on the differences in terms of solution structure. The results are presented in Tables I, II and III for each benchmark set and objective function f, g and h. For these tables, the h function managed to find 2 optimums while f and g did not for D_1^{dif} family. For D_2^{dif}, no objective function found an optimum in 3600 seconds. For instances in D_3^{dif}, the objective function h managed to find 19 optimal instances while the function g found 8 and f only one. As for D_4^{dif}, the objective function h found 20 optimal instances, the function g 10 and the function f none.

In Table I, we firstly focus on analyzing the objective values, with columns $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{A}+\mathscr{B}$ denoting the average of the best objective function values per instance family, i.e., $\mathscr{A} = \frac{1}{|D|}\sum_D \sum_{s\in S} A^s$; $\mathscr{B} = \frac{1}{|D|}\sum_D \sum_{s\in S} B^s$; $\mathscr{A}+\mathscr{B} = \frac{1}{|D|}\sum_D \sum_{s\in S}(A^s+B^s)$ where $|D|$ is the number of instances in the considered instance set D. For example, for the objective function g, the 42 instances of D_5^{sim} obtain on average, for each service, a value of 144.3 as the greatest distance between a city and its assigned main center.

Table I shows the results when the functions f, g and h simultaneously achieve optimality in 21, 39, 42 instances for the respective datasets D_3^{sim}, D_4^{sim} and D_5^{sim}. We can see that f obtains the best values of $\mathscr{B}$, g the best values of $\mathscr{A}+\mathscr{B}$ and h the best values of $\mathscr{A}$. This behavior can be explained by the fact that for all instances, in the case of optimality, f, g and h are respectively lower bounds for the values of $\sum_{s\in S} B^s$, $\sum_{s\in S}(A^s+B^s)$ and $\sum_{s\in S} A^s$. If we go further, the objective function g is on average 2.91% higher on the value of $\mathscr{A}$ than the function h (21.26% for the function f) and is on average 2.19% higher on the value of $\mathscr{B}$ than the function f (82.69% for the function h). The function g gets the best values for $\mathscr{A}+\mathscr{B}$, while f obtains values that are 6.25% higher than those of g and h gets values that are 43.65% higher than those of g. The case where the objective functions do not obtain optimality simultaneously concerns 7, 31, 33, 60 instances for the respective datasets D_1^{dif}, D_2^{dif}, D_3^{dif} and D_4^{dif}. We can observe that the objective function h generally obtains the best values for $\mathscr{A}$, except for D_2^{dif} where g obtains slightly better results.

	Instances	Size	F	$\mathscr{A}$	$\mathscr{B}$	$\mathscr{A}+\mathscr{B}$
Simultaenous	D_3^{sim}	21	f	421.2	**436.7**	858.0
			g	357.5	438.4	**795.9**
			h	**354.0**	951.3	1305.3
	D_4^{sim}	39	f	386.2	**447.1**	833.3
			g	304.6	465.6	**770.6**
			h	**289.8**	821.5	1111.3
	D_5^{sim}	42	f	156.8	**171.6**	328.4
			g	144.3	175.1	**319.4**
			h	**140.6**	251.4	392.0
Non simultaenous	D_1^{dif}	7	f	9441.8	12872.2	22313.9
			g	4609.3	**7399.1**	**12008.4**
			h	**4030.0**	19983.3	24013.3
	D_2^{dif}	31	f	1005.7	1098.1	2103.8
			g	**866.5**	**1029.6**	**1896.1**
			h	868.9	1435.0	2303.9
	D_3^{dif}	33	f	1125.7	1242.7	2368.5
			g	972.2	**1211.2**	**2183.4**
			h	**944.9**	2088.2	3033.1
	D_4^{dif}	60	f	542.1	651.3	1193.5
			g	378.7	**583.6**	**962.2**
			h	**350.2**	1262.8	1613.0

TABLE I: Comparison of the objective functions for simultaneous and non-simultaneous optimal solutions

If we now look at the values of $\mathscr{B}$ and $\mathscr{A}+\mathscr{B}$, the function g always obtains better results than the objective functions f and h. Going further, the objective function g exceeds the function h by an average of 6.35% on the value of $\mathscr{A}$ (and 56.07% compared to function f). However, for the values of $\mathscr{B}$ and $\mathscr{A}+\mathscr{B}$, the function g consistently yields the best values. In contrast, the function f exhibits a 23.71% increase for the value of $\mathscr{B}$ and a 32.32% increase for the value of $\mathscr{A}+\mathscr{B}$. For the values $\mathscr{B}$ of the function h, the function h is 99.56% higher, while for the values $\mathscr{A}+\mathscr{B}$ it is 57.01% higher on average. In both cases of simultaneous and non-simultaneous optimality, it is obvious that the function h always obtains the worst results on the values of $\mathscr{B}$ and $\mathscr{A}+\mathscr{B}$. This outcome naturally arises for this objective function since the values B^s are not constrained.

Next, we focus on the number of times the objective functions f, g and h obtain the best objective values for each set. To this end, we study the objective values $A = \sum_{s\in S} A^s$, $B = \sum_{s\in S} B^s$ and $(A+B) = \sum_{s\in S}(A^s+B^s)$ and we particularly introduce the measure $\#Best(Z)$ which denotes the number of times where an objective function gets the best value of Z, with $Z \in \{A, B, A+B\}$. Table II shows the $\#Best(Y)$ values for each objective function f, g and h in both cases of simultaneous and non-simultaneous optimality. We can observe in the case of simultaneous optimality that the performance of the objective functions varies significantly. The function h consistently dominates the $\#Best(A)$ category, particularly in D_5^{sim}, where it achieves 42 occurrences. This indicates that h is highly effective when focusing on A^s variables. In contrast, f does not perform well in this category, achieving a count of zero in almost all simultaneous cases. However, f performs strongly on $\#Best(B)$, particularly for D_4^{sim} and D_5^{sim}, where it respectively achieves 39 and 42 occurrences. In the case of $\#Best(A+B)$, g always manages to get the best results, notably in D_3^{sim} and D_4^{sim} with 21 and 39 occurrences respectively. regarding $\#Best(B)$, g performs comparatively worse, but these results remain reasonable, especially for the D_3^{sim} and D_5^{sim} benchmark sets with 16 and 12 occurrences respectively.

For non simultaneous instances, a different trend emerges. The function h remains dominant in $\#Best(A)$, particularly in D_3^{dif} (30 occurrences) and D_4^{dif} (58 occurrences), showing its effectiveness in these case, even though g managed to have better results for the D_2^{dif} family. On the other hand, g achieves significant performance in $\#Best(B)$, with particularly strong results in D_2^{dif} (28 occurrences), D_3^{dif} (26 occurrences) and D_5^{dif} (49 occurrences). In terms of #Best(A+B),

	Instances	Size	F	$\#Best(A)$	$\#Best(B)$	$\#Best(A+B)$
Simultaenous	D_3^{sim}	21	f	0	**21**	0
			g	14	16	**21**
			h	**21**	0	0
	D_4^{sim}	39	f	0	**39**	0
			g	6	4	**39**
			h	**39**	0	0
	D_5^{sim}	42	f	1	**42**	5
			g	6	12	**42**
			h	**42**	1	2
Non simultaenous	D_1^{dif}	7	f	0	1	1
			g	2	**6**	**6**
			h	**5**	0	0
	D_2^{dif}	31	f	0	4	1
			g	**15**	**28**	**30**
			h	0	0	16
	D_3^{dif}	33	f	0	9	1
			g	5	**26**	**32**
			h	**30**	0	0
	D_4^{dif}	60	f	0	11	0
			g	4	**49**	**60**
			h	**58**	0	0

TABLE II: Comparison of the number of times an objective function obtains the best objective value for simultaneous and non simultaneous optimal solutions

	Instances (size)	D_3^{sim} (21)			D_4^{sim} (39)			D_5^{sim} (42)		
	F	f	g	h	f	g	h	f	g	h
Maximal	$\#Best(\max_{s\in S} A^s)$	4	**21**	**21**	5	19	**29**	9	22	**29**
	$\#Best(avg_{s\in S} A^s)$	0	16	**21**	0	6	**39**	1	16	**29**
	$\#Best(\max_{s\in S} B^s)$	**21**	20	0	**34**	14	0	**39**	24	2
	$\#Best(avg_{s\in S} B^s)$	**21**	16	0	**39**	4	0	**42**	12	1
Average	$\#Best(\max_{s\in S} MA^s)$	**10**	9	2	1	16	**22**	7	**19**	**19**
	$\#Best(avg_{s\in S} MA^s)$	5	**13**	3	1	**20**	18	5	17	**23**
	$\#Best(\max_{s\in S} MB^s)$	**11**	10	0	**31**	8	0	**28**	16	2
	$\#Best(avg_{s\in S} MB^s)$	10	**11**	0	**31**	8	0	**27**	18	1

TABLE III: Comparison of performance on the maximum and mean service center distances for functions f, g and h

g remains the most effective function, especially for the sets D_2^{dif} (30 occurrences), D_3^{dif} (32 occurrences) and D_4^{dif} (60 occurrences), demonstrating its capacity to optimize both A^s and B^s simultaneously. Furthermore, if we focus exclusively on the objective functions f and g and compare the number of times they obtain the best results on Best(A), we can observe that the objective function g totally dominates the function f. In fact, the g function obtains the best value 233 times out of a total of 233 instances, whereas the f function only manages to match g 8 times.

In the following, we will look at how assignments between cities and their centers are organized. In the *Maximal* section of Table III, we focus on the largest assignments between each city and its centers. In particular, we study, for each instance, the following measures: $\max_{s\in S} A^s$, $\max_{s\in S} B^s$ and $avg_{s\in S} A^s$, $avg_{s\in S} B^s$. As for the *Average* section of Table III, we present the results associated to the average assignments between each city and their centers for every service. So, for each instance, we define the following MA^s and MB^s values for each service $s \in S$:

- $MA^s = \frac{1}{|N^s|} \sum_{(i,k)\in N^s \times N} d_{ik} \cdot x_{iks}$
- $MB^s = \frac{1}{|N^s|} \sum_{(i,k)\in N^s \times N} d_{ik} \cdot w_{iks}$

As before, we consider the maximum ($\max_{s\in S} MA^s$, $\max_{s\in S} MB^s$) and average ($avg_{s\in S} MA^s$, $avg_{s\in S} MB^s$) of these values.

Example 3. We consider the same graph G as in Example 1, showcased in Fig. 1. We thus have $\max_{s\in S} A^s = 20$ and $avg_{s\in S} A^s = 15$. Furthermore, knowing that $(0,0,0,10,10)$ are the assigned distances between each city and its main center for service X, and $(0,0,0,10,20)$ are the assigned distances for service Y, we thus have $MA^X = 4$ and $MA^Y = 6$. As for the assigned distances between each city and its backup center, we have the distances $(13,23,23,30,45)$ for service X and therefore $MB^X = 26.8$, while the distances $(13,23,23,25,45)$ for service Y entail $MB^Y = 25.8$. Therefore, we have $\max_{s\in S} MA^s = 6$ and $avg_{s\in S} MA^s = 5$ while $\max_{s\in S} MB^s = 26.8$ and $avg_{s\in S} MB^s = 26.3$.

The *Maximal* section of Table III shows $\#Best(Z)$ values, where $Z \in \{\max_{s\in S} A^s, \max_{s\in S} B^s, avg_{s\in S} A^s, avg_{s\in S} B^s\}$, for instance families where the three objective functions f, g and h simultaneously achieved optimality. We can observe significant differences for the objective functions f, g and h. Firstly, the function f achieves the best results on all measures concerning B^s values, while h stands out particularly on the measure associated with A^s values. However, f performs poorly on A^s's measures, while h shows very bad performance on B^s's measures. As for the objective function g, although it dominates neither measures on A^s nor measures on B^s, it proves to be the most balanced and shows very satisfying performance in both A^s and B^s values.

The *Average* section of Table III shows $\#Best(Z)$ values for each function on the simultaneously solved instance sets, with $Z \in \{\max_{s\in S} MA^s, avg_{s\in S} MA^s, \max_{s\in S} MB^s, avg_{s\in S} MB^s\}$. The observed differences are much less pronounced than in the previous table. The f function, as expected, shows lower performance on the MA^s values, although these results are better than expected, reaching in particular the best score for $\#Best(\max_{s\in S} MA^s)$ with a value of 10. At the same time, the same function scores best overall on the MB^s values, particularly on the D_4^{sim} family of instances. Concerning the h function, it shows significant weaknesses on the three criteria associated with values of MA^s for the D_3^{sim} instance family, although it remains a good option for the other instance families on the values of MA^s. Moreover, it performs comparatively worse on criteria related to the MB^s values. The objective function g, on the other hand, presents a balanced profile. Without being systematically dominant, it repeatedly achieves the best values for the indicators $\#Best(\min_{s\in S} MA^s)$ and $\#Best(avg_{s\in S} MA^s)$ in the D_3^{sim} and D_4^{sim} families of instances. For criteria relating to the MB^s values, the g function also maintains satisfying performance.

C. Comparison of Computational Times

Finally, we focus on differences in terms of computation time. The solving times in seconds are presented in Table IV for each benchmark set and objective function. We report the average, minimum and maximum times respectively in the colomuns T_{avg}, T_{min} and T_{max}. For example, the 39 instances

	Instances	Size	F	$T_{avg}(s)$	$T_{min}(s)$	$T_{max}(s)$	$Node_{avg}$
Simultaneous	D_3^{sim}	21	f	1196.1	14.6	3516.5	836.3
			g	664.6	**8.5**	2890.0	461.4
			h	**467.1**	10.2	**1913.4**	**34.3**
	D_4^{sim}	39	f	910.5	120.8	3442.9	16757.3
			g	166.2	11.2	487.8	5244.0
			h	**43.3**	**4.5**	**209.3**	**592.7**
	D_5^{sim}	42	f	135.0	0.1	3102.1	4429.3
			g	48.8	0.1	783.4	2525.2
			h	**9.7**	0.1	**163.9**	**746.0**
Non simultaenous	D_1^{dif}	7	f	3600	3600	3600	3434.1
			g	3600	3600	3600	4582.9
			h	**2744.4**	**329.2**	3600	**2126.6**
	D_2^{dif}	31	f	3600	3600	3600	2852.8
			g	3600	3600	3600	3420.8
			h	3600	3600	3600	**1876.8**
	D_3^{dif}	33	f	3564.4	2420.3	3600	6191.2
			g	3084.2	376.3	3600	11868.0
			h	**2295.7**	**87.6**	3600	**3321.7**
	D_4^{dif}	60	f	3600	3600	3600	5789.1
			g	3277.9	860.0	3600	6282.2
			h	**2702.2**	**42.3**	3600	**3573.2**

TABLE IV: Comparison in terms of solving time for simultaneous and non-simultaneous optimal solutions

of the D_4^{sim} family take an average of 166.24 seconds to solve for the objective function g. Column $Node_{avg}$ indicates the average number of nodes in the solving tree, developed by CPLEX over all instances.

In Table IV, in both cases of simultaneous and non-simultaneous optimality, function h reaches the best solution faster on average than f and g. This is obviously related to the number of nodes developed by CPLEX to reach these best solutions and to the fact that the distances to the backup centers are not constrained in h unlike the functions f and g. In case of simultaneous optimality, compared to the function h, the acceleration factors on average time with function f are 2.6 for the set D_3^{sim}, 21 for D_4^{sim} and 13.9 for D_5^{sim}, while with function g, the acceleration factors are 1.42 for D_3^{sim}, 3.8 for D_4^{sim} and 5 for D_5^{sim}. It is also noticeable that the sum of solving times are really lower for h than both f and g. Furthermore, when comparing only the functions f and g, it is surprising to observe that, even though g directly imposes constraints on the A^s variables unlike f, it solves the problem faster than f, with average acceleration factors of 1.8, 5.5, and 2.8 for the D_3^{sim}, D_4^{sim}, and D_5^{sim} families, respectively. In case of non-simultaneous optimality, we can observe that function h obtains better solving times in terms of average and minimum for the sets D_3^{sim} and D_4^{sim}. These results are due to the fact that function h obtained more optimal solutions than f and g. Besides, we can observe in Table IV that, in the case of simultaneous optimality, function g always obtains fewer developed CPLEX nodes than f. As for the opposite case, we can observe that g has more CPLEX nodes developed than f, showing that g traverses the solution space faster than f in the same limited time.

V. Conclusion

In this paper, we presented a new variant of the p-center problem that incorporates capacity constraints, stratification, and failure foresight in the context of optimizing the location of healthcare centers. We proposed a MILP formulation in which we compared three objective functions both in terms of solution structure and computational efficiency. The experimental analysis shows that the simultaneous minimization of distances to main and backup centers (i.e., function g) is more effective in our context than the single minimization of distances to main or backup centers performed by f and h. Moreover, the g function allows us to have both satisfying assignments of cities to their main center and also to their backup center, which the f and h functions do not allow. In addition to having better quality solutions, g also achieves better solving times compared to the objective function f. Furthermore, g enables to better reflect real-world scenarios by emphasizing proximity to main centers, while also taking serious account of the placement of backup centers in the event of failure, which is not the case with f and h. As future work, we would like to take into account an overload coefficient for the backup centers instead of considering that they must take on all demands when main centers fail. We also plan to give different priorities to services.

References

[1] O. Kariv and S. L. Hakimi, "An Algorithmic Approach to Network Location Problems. I: The p -Centers," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 513–538, Dec. 1979. [Online]. Available: http://epubs.siam.org/doi/10.1137/0137040

[2] R. S. Garfinkel, A. W. Neebe, and M. R. Rao, "The m - Center Problem: Minimax Facility Location," *Management Science*, vol. 23, no. 10, pp. 1133–1142, June 1977. [Online]. Available: https://pubsonline.informs.org/doi/10.1287/mnsc.23.10.1133

[3] G. Laporte, S. Nickel, and F. Saldanha Da Gama, Eds., *Location Science*. Cham: Springer International Publishing, 2019. [Online]. Available: http://link.springer.com/10.1007/978-3-030-32177-2

[4] S. Khuller and Y. J. Sussmann, "The Capacitated K - Center Problem," *SIAM Journal on Discrete Mathematics*, vol. 13, no. 3, pp. 403–418, Jan. 2000. [Online]. Available: http://epubs.siam.org/doi/10.1137/S0895480197329776

[5] O. Berman and Z. Drezner, "A new formulation for the conditional p-median and p-center problems," *Operations Research Letters*, vol. 36, no. 4, pp. 481–483, July 2008. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016763770800028X

[6] M. P. Scaparra, S. Pallottino, and M. G. Scutellà, "Large-scale local search heuristics for the capacitated vertex p -center problem," *Networks*, vol. 43, no. 4, pp. 241–255, July 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/net.20000

[7] M. Albareda-Sambola, L. I. Martínez-Merino, and A. M. Rodríguez-Chía, "The stratified p-center problem," *Computers & Operations Research*, vol. 108, pp. 213–225, Aug. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0305054819300966

[8] I. Espejo, A. Marín, and A. M. Rodríguez-Chía, "Capacitated p -center problem with failure foresight," *European Journal of Operational Research*, vol. 247, no. 1, pp. 229–244, Nov. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0377221715004841

[9] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*, 1st ed. Wiley, Apr. 1995. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118032343

[10] L. A. Lorena and E. L. Senne, "A column generation approach to capacitated p-median problems," *Computers & Operations Research*, vol. 31, no. 6, pp. 863–876, May 2004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S030505480300039X