A new formulation and Benders decomposition for the multi-period maximal covering facility location problem with server uncertainty

Amit Kumar Vatsa[1,*], Sachin Jayaswal

Production and Quantitative Methods, Indian Institute of Management, Vastrapur, Ahmedabad, Gujarat, 380 015, India

Abstract

Facility location problems reported in the literature generally assume the problem parameter values (like cost, budget, etc.) to be known with complete certainty, even if they change over time (as in multi-period versions). However, in reality, there may be some uncertainty about the exact values of these parameters. Specifically, in the context of locating primary health centers (PHCs) in developing countries, there is generally a high level of uncertainty in the availability of servers (doctors) joining the facilities in different time periods. For transparency and efficient assignment of the doctors to PHCs, it is desirable to decide the facility opening sequence (assigning doctors to unmanned PHCs) at the start of the planning horizon. We present a new formulation for a multi-period maximal covering location problem with server uncertainty. We further demonstrate the superiority of our proposed formulation over the only other formulation reported in the literature. For instances of practical size, we provide a Benders decomposition based solution method, along with several refinements. For instances that the CPLEX MIP solver could solve within a time limit of 20 hours, our proposed solution method turns out to be of the order of 150 - 250 times faster for the problems with complete coverage, and around 1000 times faster for gradual coverage.

Keywords: OR in health services, Location, Primary health centers, Benders decomposition, Uncertainty

1. Introduction and literature review

A discrete facility location problem (FLP) is the problem of finding the optimal (defined with respect to certain objectives) subset within a given set of candidate facility locations. FLPs have been widely used/studied in the context of health care facilities (Rahman and Smith, 2000; Ghaderi and Jabalameli, 2013; Pacheco and Casado, 2005). Within health care, they have been studied in the context of hospitals (Baray and Cliquet, 2013); emergency medical services (Silva and Serra, 2008; Jayaswal, 2014; Cho et al., 2014; Ramirez-Nafarrate et al., 2015);

*Corresponding author, Phone: +91-9974234079

Email addresses: amitkv@iimahd.ernet.in (Amit Kumar Vatsa), sachin@iimahd.ernet.in (Sachin Jayaswal)

and preventive health care (Verter and Lapierre, 2002; Zhang et al., 2012). FLPs have also been studied in other contexts like schools (Antunes and Peeters, 2000, 2001); banks (Wang et al., 2002); jails (Hernández et al., 2012); distribution centers (Geoffrion and Graves, 1974; Balcik and Beamon, 2008); and fire stations (Schilling et al., 1980). These problems mostly assume user demand and facility/transportation cost as given and time independent. However, when the problem parameters like user demand or facility/transportation cost change over time, an optimal facility location decision in one period may become sub-optimal in future periods. In such a situation, the optimal facility location decision needs to be revised over time according to changes in the demand/cost. However, revisiting facility location decisions in future periods may involve relocating/closing facilities opened in earlier periods. Relocating/closing facilities is generally costly, and may even be prohibitive in many cases. So, when the problem parameters are expected to change over time, a better idea is to plan ahead for more than one period. This gives rise to a multi-period FLP (MFLP), with parameter values changing over multiple time periods (Nickel and Saldanha da Gama, 2015).

Several variants of MFLP have been studied in the literature since its introduction by Ballou (1968). Wesolowsky and Truscott (1975); Melo et al. (2006) present the problem with constraints on location and relocation of facilities. Dias et al. (2006, 2007, 2008) consider MFLPs wherein facilities can be closed and reopened. Wesolowsky and Truscott (1975); Saldanha da Gama and Captivo (1998); Canel et al. (2001); Melo et al. (2006) study the problems wherein closing of facility involves capital expenditure. Erlenkotter (1981); Shulman (1991); Canel et al. (2001); Melo et al. (2006); Dias et al. (2007); Thanh et al. (2008), among others, have studied MFLPs wherein facility capacities change over time. MFLPs with budget restrictions have been studied by Antunes and Peeters (2000, 2001); Wang et al. (2003); Melo et al. (2006); Ghaderi and Jabalameli (2013). Antunes and Peeters (2000, 2001); Melo et al. (2006) have studied MFLPs with both budget and capacity constraints. Interested readers are referred to a detailed survey of the literature on MFLPs by Arabani and Farahani (2012) and Nickel and Saldanha da Gama (2015).

Classical versions of FLP/MFLP assume the problem parameter values (like demand, cost, budget, etc.) to be known with complete certainty, even if they change over time (as in MFLP). However, in reality, there may be some uncertainty about the exact values of these parameters. Averbakh and Berman (1997, 2000); Chen and Lin (1998); Vairaktarakis and Kouvelis (1999); Killmer et al. (2001); Burkard and Dollani (2002); Albareda-Sambola et al. (2011); Berman and Wang (2011) have accounted for the uncertainty in demand in FLP. Uncertainty in cost has been considered by Chen and Lin (1998); Vairaktarakis and Kouvelis (1999); Burkard and Dollani (2002). Hernández et al. (2012) study an MFLP with stochastic demands arising in the context of locating a given number of new jails in Chile, besides deciding when and where to expand the existing capacity. Nickel et al. (2012) consider an MFLP with service level and investment decisions, wherein the demand as well as the rate of return on investments are uncertain. Albareda-Sambola et al. (2013) study an MFLP with uncertainty in demand, (facility opening and maintenance) costs and the minimum number of facilities to open in each time period. We refer the readers to Snyder (2006) and Correia and Saldanha da Gama (2015) for a detailed review of the literature on FLPs under uncertainty.

Uncertainty may also arise with respect to the availability of servers/resources. This is generally true in the case of locating Primary Health Centers (PHCs), which are single doctor clinics meant to provide very basic health care in rural areas in developing countries. Due to acute shortage of doctors in rural areas, many of these PHCs remain temporarily inoperative. Moreover, there is a high degree of uncertainty regarding the number of doctors that will be available to join these PHCs in any given period. Such an uncertainty in the availability of servers/resources has not received much attention in the existing MFLP literature. Current et al. (1998) consider a situation wherein the final number of facilities to be sited is uncertain. They use a minimax regret approach to find the initial set of facilities for a p-median FLP. However, their work does not consider decisions for multiple time periods. To the best of our knowledge, Vatsa and Ghosh (2014) is the only paper to have considered such an uncertainty in the context of an MFLP.

In the current paper, we study an MFLP with uncertainty in the number of servers (doctors) available in each period of the planning horizon. Similar to Vatsa and Ghosh (2014), we consider the sequence of opening facilities over time as the decision variable, which makes this problem different from classical MFLPs. Through this paper, we make the following contributions to the scarce literature on MFLP with uncertainty in server availability:

1. We present a formulation of the problem, which we show to be stronger than the only other formulation available in the literature.
2. We present a Benders decomposition based exact solution method, and refinements thereof, to solve realistic problem instances.

The remainder of the paper is organized as follows. Section 2 describes the problem in detail, followed by mathematical models and their comparison with the existing models in the literature. Section 3 presents a Benders decomposition based solution approach, followed by computational experiments in Section 4. The paper concludes with a summary and directions for future research in Section 5.

2. Problem description

The problem described in this section is motivated by the one faced by district administrations in providing primary health care facilities to the rural population in developing countries. The World Health Organization (WHO), through its Alma-Ata declaration (1978), expressed the need for a Primary Health Center (PHC) for every 30,000 population in the plain areas and for every 20,000 in tribal and hilly areas. However, achieving this target (set by the Alma-Ata declaration) has been a challenge in most of the developing countries, largely due to shortage of doctors and increasing population (Walley et al., 2008; Rohde et al., 2008). Consequently, there is generally a shortage of PHCs. In many cases, even if PHCs exist, many of them remain unmanned due to shortage of doctors. When doctors do become available over a period of time, the challenge facing the district administration is to find the best sequence of unmanned PHCs to assign the doctors to, so as to cover the maximum population over the entire planning horizon. For transparency in policy making and implementation, it is essential that this sequence of assigning doctors to unmanned PHCs be pre-decided at the start of the planning horizon. A PHC becomes operational once a doctor is assigned to it. To be consistent with the FLP/MFLP literature, we refer to a PHC with an assigned doctor as an "open facility" wherever required in the rest of the paper.

To describe the problem setting, we assume a planning horizon consisting of discrete time periods $t \in T = \{1, 2, ..., |T|\}$. Furthermore, we consider a district, which is divided into population zones (e.g., villages), each of which is represented as a node $i \in I = \{1, 2, ..., m\}$. Let $j \in J = \{1, 2, ..., n\}$ denote any PHC without an assigned doctor. J^b is the set of PHCs that are manned with doctors at the beginning of the planning horizon, i.e., at $t = 0$. In the rest of the paper, we use the term "candidate facility" to refer to a PHC without an assigned doctor at $t = 0$. Let δ_{ij} be the distance between population zone i and candidate facility j. Opening a facility at j (i.e., assigning a doctor to the PHC at j) covers the entire population at node i if j is within a given distance δ_0 from node i, i.e., $\delta_{ij} \leq \delta_0$. We use a parameter $a_{ij} = 1$ if facility j is within the covering distance δ_0 from demand node i, 0 otherwise. We use N_i to denote the set of candidate facilities that can cover a demand node i, i.e., $N_i = \{j \in J : a_{ij} = 1\}$. Let d_{it} represent the population (demand) at node i in time period t. In general, there will be some amount of uncertainty about the future population. However, our limited study of the population growth rates in various districts of Gujarat, India, indicated a very high correlation between the population growth rates between two successive decades. Hence, we assume the population in any time period to be known with certainty since it can be estimated fairly accurately using the current population and the growth rate.

If the exact number of doctors (henceforth called servers) that will become available to join PHCs in each period of the planning horizon were known with complete certainty, then the district administration would ideally like to assign them to those candidate facilities (i.e., unmanned PHCs) that maximize the total population covered over the planning horizon. This is a classical Multi-period Maximal Covering Location Problem (MMCLP), as introduced by Gunawardane (1982). However, the exact number of doctors that will become available to join PHCs in each period of the planning horizon is generally uncertain. We describe the uncertainty in the server availability using a parameter p_{ts} to represent the number of new servers that become available at time t under scenario $s \in S$. We assume that once a newly available server is assigned to a facility (i.e., a facility is opened), it remains with the assigned facility throughout the planning horizon (in reality, doctors may leave in the middle of the planning horizon, which we leave as a direction for future research). In presence of such an uncertainty, one of the plausible objectives of the district administration would be to find the sequence of opening candidate facilities (assigning doctors to unmanned PHCs) that maximizes the expected coverage over all possible server availability (doctor joining) scenarios. However, this would require assigning probabilities to the different scenarios, which is a challenging exercise, particularly when the decision environment has multiple interdependent uncertain factors (Kouvelis and Yu, 1996). In absence of the knowledge of these probabilities, robust optimization is generally used. One plausible robust objective is to maximize the minimum coverage across all scenarios. This approach results in overly conservative decisions (Kouvelis and Yu, 1996; Snyder, 2006). In this paper, we use a *minimax* regret approach, which is a lesser conservative criterion. The regret from a proposed solution in any scenario s is defined as the difference between the maximum population that could have been covered (denoted by ζ_s^*) and the population actually covered in that scenario using the proposed solution. We refer to the resulting problem as the Multi-period Maximal Covering Location Problem under Server Uncertainty (MMCLPSU). We summarize below the notation used to define the problem:

T: Set of time periods in the planning horizon, $t \in T$

S: Set of all possible server availability scenarios, $s \in S$

p_{ts}: Number of new servers that become available at time t under scenario s

I: Set of demand nodes, $i \in \{1, 2, ..., m\}$

d_{it}: Demand of demand node i in time period t

J: Set of candidate facility locations, $j \in \{1, 2, ..., n\}$

J^b: Set of initially open facilities

δ_{ij}: Distance between demand node i and candidate facility j

δ_0: Covering distance such that candidate facility j is said to cover node i if $\delta_{ij} \leq \delta_0$

a_{ij}: 1 if facility j is within the covering distance δ_0 from demand node i, 0 otherwise

N_i: Set of candidate facilities that can cover a demand node i, i.e., $N_i = \{j \in J : a_{ij} = 1\}$

ζ_s^*: Maximum demand that can be covered in scenario s over the complete planning horizon

To mathematically model the problem, we define the following decision variables:

y_{jts}: 1 if candidate facility j is open in time period t under scenario s, 0 otherwise

x_{its}: 1 if demand node i is covered in period t under scenario s, 0 otherwise

r_{jl}: 1 if facility j is l^{th} ($l \in \{1, 2, ..., n\}$) in the sequence of opening facilities, 0 otherwise

Using these variables, the objective function of the MMCLPSU can be defined as $\min \max_{s \in S} \{\zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} x_{its}\}$. With the above notation, the MMCLPSU, as presented by Vatsa and Ghosh (2014), can be mathematically stated as follows:

[MMCLPSU-V&G:]

$$\text{Min} \quad \theta \tag{1}$$

$$s.t. \quad \theta \geq \zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} x_{its} \qquad \forall s \in S \tag{2}$$

$$x_{its} \leq \sum_{j \in N_i} y_{jts} + \sum_{j \in J^b} a_{ij} \qquad \forall i \in I, t \in T, s \in S \tag{3}$$

$$\sum_{j \in J} y_{jts} = \sum_{t' \leq t} p_{t's} \qquad \forall t \in T, s \in S \tag{4}$$

$$\sum_{l \in \{1,2,...,n\}} r_{jl} = 1 \qquad \forall j \in J \tag{5}$$

$$\sum_{j \in J} r_{jl} = 1 \qquad \forall l \in \{1, 2, ..., n\} \tag{6}$$

$$\sum_{l \in \{1,2,...,n\}} l \; r_{jl} \leq \sum_{t' \leq t} p_{t's} + n\,(1 - y_{jts}) \qquad \forall j \in J, t \in T, s \in S \tag{7}$$

$$0 \leq x_{its} \leq 1 \qquad \forall i \in I, t \in T, s \in S \tag{8}$$

$$\theta \geq 0 \tag{9}$$

$$y_{jts} \in \{0, 1\} \qquad \forall j \in J, t \in T, s \in S \tag{10}$$

$$r_{jl} \in \{0, 1\} \qquad \forall j \in J, l \in \{1, 2, ..., n\} \tag{11}$$

(1) and (2) together help linearize the above described objective function ($\min \max_{s \in S} \{\zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} x_{its}\}$). ζ_s^* is the maximum coverage possible in a given scenario $s \in S$. Its value is obtained by solving (12) - (17), as given below, which is an MMCLP. Constraint set (3) ensures that any demand node is covered in any period and scenario only if at least one open facility exists within its covering distance. The number of open facilities in any period and scenario is specified by (4). Constraint sets (5) and (6) ensure that each facility is given a unique rank in the sequence. Constraint set (7) links the variables r_{jl} and y_{jts} based on the condition that a facility at j will be open in period t and scenario s ($y_{jts} = 1$) only if the rank of the facility j is less than or equal to the total number of new servers that become available till period t in scenario s. Even though x_{its} are binary, Vatsa and Ghosh (2014) show that relaxing them as continuous variables leaves the solution to the MMCLPSU unchanged. Hence, constraint set (8) relaxes x_{its} as continuous variables. Constraints (9)-(11) are the non-negativity

and binary constraints.

$[MMCLP_s$:]

$$\text{Max} \quad \zeta_s = \sum_{i\in I}\sum_{t\in T} d_{it}x_{its} \tag{12}$$

$$s.t. \quad x_{its} \leq \sum_{j\in N_i} y_{jts} + \sum_{j\in J^b} a_{ij} \qquad \forall i \in I, t \in T \tag{13}$$

$$y_{jts} \geq y_{j(t-1)s} \qquad \forall j \in J, t \in T\backslash\{1\} \tag{14}$$

$$\sum_{j\in J} y_{jts} = \sum_{t'\leq t} p_{t's} \qquad \forall t \in T \tag{15}$$

$$0 \leq x_{its} \leq 1 \qquad \forall i \in I, t \in T \tag{16}$$

$$y_{jts} \in \{0,1\} \qquad \forall j \in J, t \in T \tag{17}$$

Constraint set (14) in $MMCLP_s$ ensures that a facility once opened remains open throughout the planning horizon. Such a constraint is also required for the MMCLPSU, but it is already implied by the use of sequence variable r_{jl}.

For a problem with m demand nodes, n candidate facilities, and $|T|$ time periods, the total number of scenarios $|S| = \binom{n+|T|}{n} = \frac{(n+|T|)!}{n!(|T|)!}$. For MMCLPSU-V&G, this results in $n|T||S| + n^2$ binary (for y_{jts}, r_{jl}) and $m|T||S|$ continuous (for x_{its}) variables, and $|S|+(m+n+1)|T||S|+2n$ constraints. For example, $m = 100$, $n = 15$, $|T| = 4$ results in $|S| = 3,876$ scenarios and $232,785$ binary and $1,550,400$ continuous variables, and $1,771,362$ constraints (excluding binary and lower/upper bound constraints). Although using scenario dominance conditions, Vatsa and Ghosh (2014) are able to reduce the problem size considerably, the problem is still difficult to solve, taking around 40 hours (using CPLEX MIP solver) in some instances. We, therefore, present an alternative formulation for the MMCLPSU, which results in fewer variables and constraints. We further show that our formulation is better than MMCLPSU-V&G.

To introduce our formulation, we define a new set of decision variables $z_{jk} = 1$ if candidate facility j is one among the $k \in \{0, 1, ..., n\}$ candidate facilities that have been opened during the planning horizon, 0 otherwise. Clearly, the number of candidate facilities opened depends on the time period t of the planning horizon and the server availability scenario s, given by the relation $k(t,s) = \sum_{t'\leq t} p_{t's}$. It is important to note that different combinations of t and s may result in the same number of open facilities (k). For ease of notation, we explicitly state the dependence of k on t and s as $k(t,s)$ in the rest of the paper only when t or s takes a specific value. The variable z_{jk} is related to the variable y_{jts} and r_{jl} in MMCLPSU-V&G as follows:

$$z_{jk(t,s)} = y_{jts} \qquad \forall j \in J, t \in T, s \in S \tag{18}$$

$$z_{jk} - z_{j(k-1)} = r_{jk} \qquad \forall j \in J, k \in \{1, 2, .., n\} \tag{19}$$

The variables z_{jk}, by definition, should satisfy the following relations:

$$z_{jk} \geq z_{j(k-1)} \qquad \forall j \in J, k \in \{1, 2, ...n\} \tag{20}$$

$$\sum_{j \in J} z_{jk} = k \qquad \forall k \in \{0, 1, 2, ...n\} \tag{21}$$

For example, consider a solution with the z_{jk} values as given in Table 1. The sequence of opening the 5 facilities in this example is B-D-E-A-C. With a server availability scenario s, if 2 new servers become available by the end of time t, i.e., $\sum_{t' \leq t} p_{t's} = 2$, then the two candidate facilities to be opened will be B and D, i.e., $z_{B2} = z_{D2} = 1$, while $z_{A2} = z_{C2} = z_{E2} = 0$.

Table 1: An example with variable z_{jk}

		Total Open (k)					
		k=0	k=1	k=2	k=3	k=4	k=5
Facilities	A	0	0	0	0	1	1
	B	0	1	1	1	1	1
	C	0	0	0	0	0	1
	D	0	0	1	1	1	1
	E	0	0	0	1	1	1

With the above variable definition, the MMCLPSU can be mathematically restated as follows: [MMCLPSU:]

$$(1), (2), (8), (9), (20), (21)$$

$$x_{its} \leq \sum_{j \in N_i} z_{jk(t,s)} + \sum_{j \in J^b} a_{ij} \qquad \forall i \in I, t \in T, s \in S \tag{22}$$

$$z_{jk} \in \{0, 1\} \qquad \forall j \in J, k \in \{0, 1, 2, ...n\} \tag{23}$$

Constraint set (22) ensures that any demand node is covered in any period and scenario only if at least one open facility exists within its covering distance. This combines (3) and (18). Like MMCLPSU-V&G, we are relaxing x_{its} as continuous variables (in (8)) since doing so leaves the solution to the model unchanged. Table 2 provides a comparison of the resulting model size for MMCLPSU versus MMCLPSU-V&G. Taking $m = 100, n = 15, |T| = 4$ results in $|S| = 3,876$ scenarios and only 240 binary variables and $1,554,517$ constraints (excluding binary and lower/upper bound constraints), as opposed to $232,785$ binary variables and $1,771,362$ constraints in the case of MMCLPSU-V&G. The number of continuous variables remains the same. Moreover, constraint set (21) fixes z_{j0} to 0 and z_{jn} to 1 $\forall j \in J$, further reducing the computational effort in MMCLPSU. We now show mathematically that MMCLPSU is better than MMCPLSU-V&G.

Table 2: Comparison between MMCLPSU-V&G and MMCLPSU formulations

	MMCLPSU-V&G	MMCLPSU
No. of binary variables	$n^2 + n\|T\|\|S\|$	$n^2 + n$
No. of continuous variables	$m\|T\|\|S\|$	$m\|T\|\|S\|$
No. of constraints	$\|S\| + (m + n + 1)\|T\|\|S\| + 2n$	$\|S\| + m\|T\|\|S\| + n^2 + n + 1$

Proposition 1. *P_{LP}(MMCLPSU) $\subset$ P_{LP}(MMCLPSU-V&G), where $P_{LP}(\cdot)$ is the polyhedron of the LP relaxation of $(\cdot)$.*

Proof. Given a feasible solution $[\hat{z}, \hat{x}, \hat{\theta}]$ to the LP relaxation of MMCLPSU, we can construct a variable $r_{jk} = \hat{z}_{jk} - \hat{z}_{j(k-1)} \quad \forall j \in J, k \in \{1, 2, .., n\}$. Now,

$$\sum_{j} r_{jk} = \sum_{j} \hat{z}_{jk} - \sum_{j} \hat{z}_{j(k-1)} = k - (k-1) = 1 \qquad [\because \sum_{j \in J} \hat{z}_{jk} = k \text{ using (21)}]$$
$$\sum_{k \geq 1} r_{jk} = \sum_{k \geq 1} \hat{z}_{jk} - \sum_{k \geq 1} \hat{z}_{j(k-1)} = \hat{z}_{jn} - \hat{z}_{j0} = 1 \qquad [\because \hat{z}_{jn} = 1 \text{ and } \hat{z}_{j0} = 0 \text{ using (21) and LP relaxation of (23)}]$$

Hence, r_{jk} satisfies constraint sets (5) and (6) of MMCLPSU-V&G. Now, we substitute y_{jts} with $\hat{z}_{jk}$, where $k = \sum_{t' \leq t} p_{t's}$, and check if $[\hat{z}, \hat{x}, \hat{\theta}]$ satisfies the other constraints of MMCLPSU-V&G. Constraints (7), i.e., $\sum_{l \in \{1,2,...,n\}} l \ r_{jl} \leq \sum_{t' \leq t} p_{t's} + n(1 - y_{jts}) \quad \forall j \in J, t \in T, s \in S$, will be satisfied by $[\hat{z}, \hat{x}, \hat{\theta}]$ if:

$$\sum_{k \geq 1} k \ r_{jk} - n(1 - y_{jts}) \leq \sum_{t' \leq t} p_{t's} \quad \forall j \in J, t \in T, s \in S \tag{24}$$
$$\iff \sum_{k \geq 1} k(\hat{z}_{jk} - \hat{z}_{j(k-1)}) - n(1 - \hat{z}_{jk}) \leq k \quad \forall j \in J, k \in \{0, 1, 2, .., n\} : k = \sum_{t' \leq t} p_{t's} \tag{25}$$
$$\iff [\hat{z}_{j1} - \hat{z}_{j0}] + [2\hat{z}_{j2} - 2\hat{z}_{j1}] + ... + [n\hat{z}_{jn} - n\hat{z}_{j(n-1)}] + n\hat{z}_{jk} \leq k + n \quad \forall j, k \tag{26}$$
$$\iff n\hat{z}_{jn} + n\hat{z}_{jk} - \hat{z}_{j0} - \hat{z}_{j1} - \hat{z}_{j2} - ... - \hat{z}_{j(n-1)} \leq k + n \quad \forall j, k \tag{27}$$
$$\iff n\hat{z}_{jk} - \hat{z}_{j1} - \hat{z}_{j2} - ... - \hat{z}_{j(n-1)} \leq k \quad \forall j, k \qquad [\because \hat{z}_{jn} = 1, \hat{z}_{j0} = 0] \tag{28}$$
$$\iff n\hat{z}_{jk} - \hat{z}_{j1} - \hat{z}_{j2} - ... - \hat{z}_{j(n-1)} - \hat{z}_{jn} \leq k - 1 \quad \forall j, k \qquad [\because \hat{z}_{jn} = 1] \tag{29}$$
$$\iff (\hat{z}_{jk} - \hat{z}_{j1}) + (\hat{z}_{jk} - \hat{z}_{j2}) + .. + (\hat{z}_{jk} - \hat{z}_{j(k-1)}) \leq k - 1 + (\hat{z}_{j(k+1)} - \hat{z}_{jk}) + ... + (\hat{z}_{jn} - \hat{z}_{jk}) \quad \forall j, k \tag{30}$$

Since $\hat{z}_{jk} \geq \hat{z}_{j(k-1)} \quad \forall j, k \geq 1$ (using (20)), each of the terms within parenthesis in the last inequality lies between 0 and 1. Since there are $k - 1$ terms on the left-hand side (LHS) of inequality (30), LHS cannot be greater than $k - 1$. The right-hand side (RHS) of (30) is $k - 1$

+ some non-negative terms. Hence, inequality (30) holds true. Consequently, the inequality $\sum_l lr_{jl} \leq \sum_{t' \leq t} p_{t's} + n(1 - y_{jts}) \quad \forall j \in J, t \in T, s \in S$, is satisfied by $[\hat{z}, \hat{x}, \hat{\theta}]$. Furthermore, constraint set (3) of MMCLPSU-V&G is the same as (22) of MMCLPSU (replacing y_{jts} with z_{jk}, where $k = \sum_{t' \leq t} p_{t's}$). Therefore, $[\hat{z}, \hat{x}, \hat{\theta}]$ is a feasible solution to the LP relaxation of MMCLPSU-V&G.

It follows from (18) that different combinations of scenario s and time t in MMCLPSU that result in the same number $k = \sum_{t' \leq t} p_{t's}$ of open facilities, will always have the same value for the variable y_{jts} $\forall j \in J$. However, this is not true for MMCLPSU-V&G. This implies that a solution that is feasible to the LP relaxation of MMCLPSU-V&G may not be feasible to the LP relaxation of MMCLPSU. We now prove that this is indeed true.

Summing over $j \in J$ the constraint set (7) in MMCLPSU-V&G, we get:

$$\sum_{j \in J} y_{jts} \leq \sum_{t' \leq t} p_{t's} + (n-1)/2 \qquad \forall t \in T, s \in S \tag{31}$$

Comparing (31) with constraint set (4) suggests that there must be at least one j for which constraint (7) will be non-binding. Now, consider a period t_1 in scenario s_1, and a period t_2 in scenario s_2 such that $\sum_{t' \leq t_1} p_{t's_1} = \sum_{t' \leq t_2} p_{t's_2} = k$. Let facility A be a facility under scenario s_1 and period t_1 for which constraint (7) is non-binding. Assume a feasible LP relaxation solution of MMCLPSU-V&G that is also feasible to the LP relaxation of MMCLPSU. This implies $y_{At_1s_1} = y_{At_2s_2} = \hat{z}_{Ak}$ for that solution. Let us generate another solution by increasing $y_{At_1s_1}$ by ϵ (since (7) is non-binding for $y_{At_1s_1}$), where ϵ is an infinitesimal positive number. This will violate constraint (4) of MMCLPSU-V&G. Nonetheless, if we simultaneously decrease $y_{Bt_1s_1}$ by the same amount ϵ, where B is any candidate facility other than A, then the solution remains feasible to MMCLPSU-V&G. However, this solution will not be feasible to MMCLPSU since $y_{At_1s_1} + \epsilon \neq y_{At_2s_2} = \hat{z}_{Ak}$.

Thus, any solution to the LP relaxation of MMCLPSU is also a solution to the LP relaxation of MMCLPSU-V&G. However, the converse is not true. □

Proposition 2. *$Z_{LP}(MMCLPSU) = Z_{LP}(MMCLPSU - V\&G)$, where $Z_{LP}(\cdot)$ is the LP relaxation based lower bound of $(\cdot)$.*

Proof. In Proposition 1, it is shown that the LP feasible region of MMCLPSU is a proper subset of the LP feasible region of MMCLPSU-V&G. We prove Proposition 2 by showing below that an optimal solution to the LP relaxation of MMCLPSU-V&G falls in the LP feasible region of MMCLPSU.

Consider an optimal solution $[\hat{r}^*, \hat{y}^*, \hat{x}^*, \hat{\theta}^*]$ to the LP relaxation of MMCLPSU-V&G. Then, we have:

$$\hat{y}^*_{jts} \leq \min\left\{1, (k + n - \sum_{l} l\ \hat{r}^*_{jl})/n\right\} \qquad \text{(from (7) and (10))} \tag{32}$$

$$\sum_{j} \hat{y}^*_{jts} = k \qquad \text{(from (4))} \tag{33}$$

where $k = \sum_{t' \leq t} p_{t's}$. At optimality, the objective of regret minimization ensures that variables y_{jts} take the maximum possible values. Hence, every combination of scenario s and time t such that $k = \sum_{t' \leq t} p_{t's}$, will have the same value of $\hat{y}^*_{jts}$ (say $= \hat{z}_{jk}$) $\forall j \in J$. Clearly, $\hat{z}_{jk} \geq \hat{z}_{j(k-1)}$ (from (32) and (33)). Furthermore, $\sum_{j} \hat{z}_{jk} = \sum_{j} \hat{y}^*_{jts} = k$ (from (33)). All other constraints in MMCLPSU-V&G and MMCLPSU are similar. Consequently, $[\hat{r}^*, \hat{y}^*, \hat{x}^*, \hat{\theta}^*]$ is a feasible LP solution to MMCLPSU, and therefore, an optimal LP solution to MMCLPSU. $\square$

It follows from Proposition 1 that MMCLPSU is a better formulation compared to MMCLPSU-V&G even though both have the same LP relaxation based lower bound. This is highlighted using the example depicted in Figure 1, which shows the LP feasible regions BC and $OABC$ corresponding to two alternative formulations, let us say $f1$ and $f2$, respectively. Clearly, $P_{LP}(f1) \subset P_{LP}(f2)$, and both $f1$ and $f2$ have the same LP bound at B. However, after branching at the root node in a branch-and-bound tree, the feasible region for $f1$ reduces to CG, while that for $f2$ reduces to $OFGC$ and DAE. Clearly, $f1$ will never take more computational effort in getting to the IP optimal solution G.

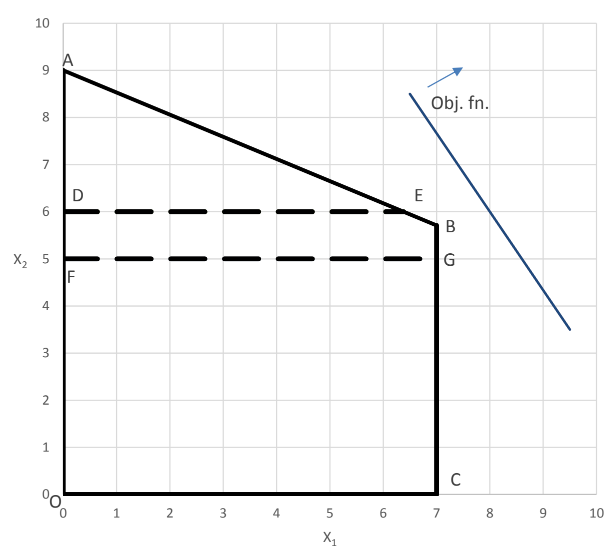

Figure 1: An example of LP feasible region

Table 3 presents a comparison of the computation times taken to obtain the optimal solution (using CPLEX MIP solver) from the two formulations for different instances. Clearly, MMCLPSU solves the problem significantly faster. For example, for instance 2 with $n = 15, m = 100$, and $|T| = 4$, the computation time required by MMCLPSU-V&G is more than 11 times that required by MMCLPSU. For instance 3, MMCLPSU-V&G fails to solve the problem even after a CPU time limit of 20 hours, while MMCLPSU solves it in close to an hour. For larger problem sizes, MMCLPSU-V&G fails to find the optimal solution for any of the 5 instances within the 20 hour time limit. MMCLPSU, on the other hand, is able to solve all the 5 instances within the same time limit. However, the CPU time required to solve instances with $n = 15, m = 200, |T| = 4$ is significantly large even for MMCLPSU, the maximum being close to 8 hours. For larger instances, it will be difficult to solve MMCLPSU to optimality within a reasonable time limit. Therefore, we present in Section 3 a Benders decomposition based solution approach to speed up the solution process.

Table 3: Comparison between MMCLPSU-V&G and MMCLPSU

Problem Size	Instance	F1-CPU(s)	F2-CPU(s)	MIP-opt	LP-gap
$n = 15, m = 100, \|T\| = 4$	1	2054.8	251.4	159.6	9.8%
	2	70710.4	6029.7	3221.5	37.8%
	3	*	3693.3	7631.9	39.4%
	4	*	7891.4	4106.3	38.4%
	5	1810.0	229.3	249.7	4.5%
$n = 15, m = 200, \|T\| = 4$	1	*	28644.5	13335.4	20.8%
	2	*	9062.8	4833.0	20.4%
	3	*	4670.8	904	15.8%
	4	*	27265.0	6263.6	54.0%
	5	*	7397.1	5848.9	34.8%

F1-CPU(s) and F2-CPU(s) refer to the CPU times in seconds taken by MMCLPSU-V&G and MMCLPSU, respectively; MIP-opt refers to the MIP optimal objective function value; LP-gap refers to the gap between MIP-opt and the LP based lower bound; * indicates the instance could not be solved in a 20 hour time limit.

2.1. Gradual coverage

In MMCLPSU, we assumed a particular facility j can either cover or not cover a demand node i depending on whether the node i lies within or outside the covering distance from j. Accordingly, we defined a parameter $a_{ij} = 1$ if facility j can cover demand node i, 0 otherwise. However, in most of the situations, the coverage does not change so abruptly. There is instead a range of distances, between a minimum and a maximum covering distance (δ_{min} and δ_{max}), within which the coverage reduces gradually with distance. Such a gradual coverage is considered by Church and Roberts (1983); Berman et al. (2003); Karasakal and Karasakal

(2004); Berman et al. (2010). However, none of them consider a multi-period planning horizon or server uncertainty. We now generalize MMCLPSU by allowing for gradual/partial coverage of a demand node if it lies between δ_{min} and δ_{max} from an open facility.

For the complete coverage version of MMCLPSU, it was sufficient to know whether a demand node i was covered or not in a given time and scenario. Accordingly, we defined a variable x_{its}. However, such a variable definition is not sufficient to model the gradual coverage since to determine the level of coverage of a node i, it is also important to know which specific facility covers it. Accordingly, we now define a variable $x_{ijts} = 1$ if the demand node i is covered (fully or partially) by facility at j in period t and scenario s, 0 otherwise. In this problem, the coverage function can take fractional values if the demand node i is within δ_{min} and δ_{max} from facility at j, i.e., $a_{ij} \in [0, 1]$. Similarly, we redefine N_i as the set of candidate facilities that are within the maximum covering distance δ_{max} from demand node i. We also define N_i^b as the set of facilities open at the beginning of the planning horizon that lie within δ_{max} of node i. The resulting problem, which we refer to as the Multi-period Maximal Covering Location Problem under Server Uncertainty with Partial coverage (MMCLPSU-P), can be formulated as follows:

[MMCLPSU-P:]

$$\text{Min} \quad \theta \tag{34}$$

$$s.t. \quad \theta \geq \zeta_s^* - \sum_{i \in I} \sum_{j \in N_i \cup N_i^b} \sum_{t \in T} a_{ij} d_{it} x_{ijts} \qquad \forall s \in S \tag{35}$$

$$x_{ijts} \leq z_{jk(t,s)} \qquad \forall i \in I, j \in N_i, \forall t \in T, s \in S \tag{36}$$

$$\sum_{j \in N_i \cup N_i^b} x_{ijts} \leq 1 \qquad \forall i \in I, t \in T, s \in S \tag{37}$$

$$z_{jk} \geq z_{j(k-1)} \qquad \forall j \in J, k \in \{1, 2, ...n\} \tag{38}$$

$$\sum_{j \in J} z_{jk} = k \qquad \forall k \in \{0, 1, 2, ...n\} \tag{39}$$

$$x_{ijts} \geq 0 \qquad \forall i \in I, j \in N_i \cup N_i^b, t \in T, s \in S \tag{40}$$

$$\theta \geq 0 \tag{41}$$

$$z_{jk} \in \{0, 1\} \qquad \forall j \in J, k \in \{0, 1, 2, ...n\} \tag{42}$$

(34) and (35) help linearize the objective of minimizing the maximum regret, similar to the MMCLPSU formulation above. ζ_s^* is the maximum coverage possible in a given scenario $s \in S$. Its value is obtained by solving (43) - (49), as given below, which we call as the Multi-period Maximal Covering Location Problem with partial coverage (MMCLP-P). Constraint set (36) ensures that a demand node is covered by any facility that lies within δ_{max} from the demand node, only if the facility is open. Constraint set (37) ensures that a demand node is covered by at most one facility. Constraints (38) and (39) are the same as (20) and (21) respectively. x_{ijts},

which by definition is a binary variable, can be relaxed as a continuous variable (Vatsa and Ghosh, 2014). Since $x_{ijts} \leq 1$ is already implied by (37), continuous relaxation of binary x_{ijts} is stated as (40). (41) and (42) are non-negativity and binary constraints. Clearly, MMCLPSU is a special case of MMCLPSU-P when $\delta_{min} = \delta_{max}$.

[*MMCLP-P*$_s$:]

$$\text{Max} \quad \zeta_s = \sum_{i \in I} \sum_{j \in J} \sum_{t \in T} a_{ij} d_{it} x_{ijts} \tag{43}$$

$$s.t. \quad x_{ijts} \leq y_{jts} \qquad \forall i \in I, j \in N_i, \forall t \in T \tag{44}$$

$$\sum_{j \in N_i \cup N_i^b} x_{ijts} \leq 1 \qquad \forall i \in I, t \in T \tag{45}$$

$$y_{jts} \geq y_{j(t-1)s} \qquad \forall j \in J, t \in T \backslash \{1\} \tag{46}$$

$$\sum_{j \in J} y_{jts} = \sum_{t' \leq t} p_{t's} \qquad \forall t \in T \tag{47}$$

$$0 \leq x_{ijts} \leq 1 \qquad \forall i \in I, j \in N_i \cup N_i^b, t \in T \tag{48}$$

$$y_{jts} \in \{0, 1\} \qquad \forall j \in J, t \in T \tag{49}$$

All the constraints of *MMCLP-P*$_s$ are also implied in MMCLPSU-P. Here again, as in $MMCLP_s$, constraint set (46) ensures that a facility once opened remains open throughout the planning horizon. Such a constraint is redundant in MMCLPSU-P, as it is already implied by the use of the sequence variable r_{jl}.

Like MMCLPSU for the complete coverage, MMCLPSU-P is also a better formulation compared to the formulation given by Vatsa and Ghosh (2014) for the problem with gradual coverage. This can be proven along similar lines as done for MMCLPSU, and hence we skip the details. We now present the Benders decomposition based solution method for the complete and gradual coverage versions of the problem.

3. Benders decomposition based solution method

Benders decomposition is a partition based solution technique, which has been applied to solve mixed integer programming problems (Benders, 1962). It has been successfully applied to (multicommodity) network design (Geoffrion and Graves, 1974), facility location (Wentges, 1996), and hub location (de Camargo et al., 2009, 2011; Contreras et al., 2011). Costa (2005) provides a detailed review of the application of Benders decomposition to fixed-charge network design problems.

In Benders decomposition method, the original problem is partitioned into a master problem and a sub-problem. The master problem and the sub-problem are solved iteratively by utilizing the solution of one in the other. The master problem contains a set of the complicating (integer) variables and their associated constraints. The sub-problem is obtained by temporarily fixing the integer variables in the original problem using the solution of the master problem. At each iteration, a relaxed master problem is solved to obtain a lower bound. The sub-problem solution generates a Benders cut, which is added back to the master problem. The master problem is completely defined when all possible Benders cuts are added to the problem. However, in practice this is unnecessary, and at each iteration a relaxed master problem is solved, where only a subset of all possible Benders cuts is added to the master problem. For a minimization problem, the relaxed master problem solution at any iteration provides a lower bound to the original problem, while the sub-problem solution generates an upper bound. The Benders algorithm converges to an optimal solution for the original mixed integer programming problem if such a solution exists.

In Section 3.1, we describe how the Benders decomposition based solution method is applied to MMCLPSU. The Benders decomposition for MMCLPSU-P shares a lot of similarity with MMCLPSU. Hence, we relegate its description to Appendix A.

3.1. Complete coverage

As shown by Vatsa and Ghosh (2014), removal of the set J^b of pre-existing facilities, if any, along with the demand nodes that they cover does not affect the optimal objective function value of MMCLPSU. We use this result to eliminate set J^b from further consideration in MMCLPSU. By fixing the binary variables z_{jk} as $\bar{z}_{jk}$ we obtain the following primal sub-problem:

[MMCLPSU-PSP:]

$$\text{Min} \quad \theta \tag{50}$$

$$s.t. \quad \theta + \sum_{i \in I} \sum_{t \in T} d_{it} x_{its} \geq \zeta_s^* \qquad \forall s \in S \tag{51}$$

$$x_{its} \leq \sum_{j \in N_i} \bar{z}_{jk(t,s)} \qquad \forall i \in I, t \in T, s \in S \tag{52}$$

$$x_{its} \leq 1 \qquad \forall i \in I, t \in T, s \in S \tag{53}$$

$$\theta, x_{its} \geq 0 \qquad \forall i \in I, t \in T, s \in S \tag{54}$$

Let α_s, β_{its} and ρ_{its} be the dual variables associated with the constraint set (51), (52) and (53) respectively. The dual of this problem can be formulated as follows:

[MMCLPSU-DSP:]

$$\text{Max} \quad \sum_{s \in S} \zeta_s^* \alpha_s - \sum_{i \in I} \sum_{t \in T} \sum_{s \in S} \rho_{its} - \sum_{i \in I} \sum_{t \in T} \sum_{s \in S} \left(\beta_{its} \sum_{j \in N_i} \bar{z}_{jk(t,s)} \right) \tag{55}$$

$$
\begin{aligned}
s.t.\quad & d_{it}\alpha_s - \beta_{its} - \rho_{its} \leq 0 && \forall i \in I, t \in T, s \in S && (56)\\
& \sum_{s \in S} \alpha_s \leq 1 && && (57)\\
& \alpha_s, \beta_{its}, \rho_{its} \geq 0 && \forall i \in I, t \in T, s \in S && (58)
\end{aligned}
$$

Let H denote the set of all extreme points of MMCLPSU-DSP. For each extreme point $h \in H$, we denote the corresponding values of the dual variables as $\alpha_s^h, \beta_{its}^h$, ρ_{its}^h, and the corresponding values of the primal variables as x_{its}^h, θ^h. The Benders cut generated by the extreme point h to be included in the master problem is given by:

$$
\eta \geq \sum_{s \in S} \zeta_s^* \alpha_s^h - \sum_{i \in I} \sum_{t \in T} \sum_{s \in S} \rho_{its}^h - \sum_{i \in I} \sum_{t \in T} \sum_{s \in S} \left(\beta_{its}^h \sum_{j \in N_i} z_{jk(t,s)} \right) \tag{59}
$$

Since the master problem deals with z_{jk} variables, rearranging the last term of (59) gives the following alternative representation of the Benders cuts:

$$
\eta \geq \sum_{s \in S} \zeta_s^* \alpha_s^h - \sum_{i \in I} \sum_{t \in T} \sum_{s \in S} \rho_{its}^h - \sum_{j \in J} \sum_{t \in T} \sum_{s \in S} \left(\sum_{i \in N_j} \beta_{its}^h \right) z_{jk(t,s)} \tag{60}
$$

where N_j is the set of demand nodes that can be covered by any candidate facility j, i.e., $N_j = \{i \in I : a_{ij} = 1\}$. The master problem can be stated as follows:

[MMCLPSU-MP:]

$$
\begin{aligned}
\text{Min}\quad & \eta && && (61)\\
s.t.\quad & z_{jk} \geq z_{j(k-1)} && \forall j \in J, k \geq 1 && (62)\\
& \sum_{j \in J} z_{jk} = k && \forall k \in \{0, 1, ..., n\} && (63)\\
& \eta \geq \sum_{s \in S} \zeta_s^* \alpha_s^h - \sum_{i \in I} \sum_{t \in T} \sum_{s \in S} \rho_{its}^h - \sum_{j \in J} \sum_{t \in T} \sum_{s \in S} \left(\sum_{i \in N_j} \beta_{its}^h \right) z_{jk(t,s)} && \forall h \in H && (64)\\
& z_{jk} \in \{0, 1\}, \eta \geq 0 && \forall j \in J, k \in \{0, .., n\} && (65)
\end{aligned}
$$

Proposition 3. *The primal sub-problem MMCLPSU-PSP is always feasible and bounded for any feasible solution $\bar{z}_{jk}$ to MMCLPSU-MP.*

Proof. A feasible solution to the master problem at any iteration provides a facility opening sequence, indicated by the values of $\bar{z}_{jk}$. Such a sequence obtained from the master problem also conveys the set of open facilities, and hence the coverage of each demand node (defined by the value of the variable x_{its}), in each time period t and scenario s. Hence, a feasible solution to the master problem always produces a feasible solution to the corresponding sub-problem. This feasible solution can be used to calculate the overall coverage and the regret in each scenario s. Using the regret value in each scenario, the objective function value of the sub-problem, which is the maximum regret across all scenarios, can be obtained. Since the regret in any scenario, and hence the maximum among them, is finite, the optimal solution to the sub-problem is always bounded. $\square$

We now present an algorithm to efficiently solve MMCLPSU-DSP since it needs to be solved iteratively in the Benders decomposition framework.

Algorithm 1 Solution algorithm for MMCLPSU-DSP

1: set $x_{its} \leftarrow \min(1, \sum_{j \in N_i} \bar{z}_{jk}) \quad \forall i \in I, t \in T, s \in S$, where $k = \sum_{t' \leq t} p_{t's}$;

2: $\theta \leftarrow \max_{s \in S} \left(\zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} x_{its} \right), \xi \leftarrow \operatorname{argmax}_{s \in S} \left(\zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} x_{its} \right)$. Ties can be broken arbitrarily;

3: set $\alpha_\xi \leftarrow 1, \alpha_s \leftarrow 0 \quad \forall s \in S \backslash \{\xi\}$;

4: **if** $x_{its} = 0$ **then** set $\rho_{its} \leftarrow 0, \beta_{its} \leftarrow d_{it}\alpha_s \quad \forall i \in I, t \in T, s \in S$;

5: **else** set $\beta_{its} \leftarrow 0$, $\rho_{its} \leftarrow d_{it}\alpha_s \quad \forall i \in I, t \in T, s \in S$;

6: **end if**

7: output $\alpha_s, \beta_{its}, \rho_{its} \quad \forall i \in I, t \in T, s \in S$.

Proposition 4. *For a given solution $\bar{z}_{jk}$ to MMCLPSU-MP, Algorithm 1 gives an optimal solution to MMCLPSU-DSP.*

Proof. First, we prove that Algorithm 1 gives a feasible solution to MMCLPSU-DSP. Clearly, steps 1 and 2 give an optimal solution to MMCLPSU-PSP. The solution to MMCLPSU-DSP is obtained in steps 3 to 6 using the complementary slackness conditions between MMCLPSU-PSP and MMCLPSU-DSP. Applying the complementary slackness condition to (51) gives: $(\theta + \sum_{i \in I} \sum_{t \in T} d_{it} x_{its} - \zeta_s^*)\alpha_s = 0 \quad \forall s \in S$. This, together with (57), gives as feasible solution $\alpha_\xi = 1$, where $\xi = \operatorname{argmax}_{s \in S}(\zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} x_{its})$ and $\alpha_s = 0 \;\; \forall s \in S \backslash \{\xi\}$ in step 3. Furthermore, the complementary slackness condition on constraint set (53) in step 4 gives $\rho_{its} = 0$ when $x_{its} = 0 \quad \forall i \in I, t \in T, s \in S$. β_{its} is obtained in step 4 using the values of α_s and ρ_{its} in (56) and exploiting the fact that (56) is binding at optimality. On the other hand, when $x_{its} \neq 0$, step 5 gives feasible values for β_{its} and ρ_{its} using (56). The intuition behind this step comes from the interpretation of the dual variables.

We now show that this solution is optimal. From steps 3, 4 and 5, $\alpha_s = 0, \beta_{its} = 0, \rho_{its} = 0 \quad \forall s \in S \backslash \{\xi\}$. Hence, with the solution found in Algorithm 1, the objective function of

MMCLPSU-DSP, given by (55), can be expressed as:

$$\zeta_{\xi}^{*}-\sum_{i\in I}\sum_{t\in T}\rho_{it\xi}-\sum_{i\in I}\sum_{t\in T}\left(\beta_{it\xi}\sum_{j\in N_i}\bar{z}_{jk(t,\xi)}\right) \tag{66}$$

It can be seen from steps 4 and 5 that $\beta_{it\xi}$ indicates the demand that is not covered, while $\rho_{it\xi}$ indicates the demand that is covered at demand node i in period t and scenario ξ. Consequently, the first two terms in (66) together give the regret in scenario ξ, which from step 2 is equal to θ. Hence, (66) can be restated as:

$$\theta-\sum_{i\in I}\sum_{t\in T}\left(\beta_{it\xi}\sum_{j\in N_i}\bar{z}_{jk(t,\xi)}\right) \tag{67}$$

Moreover, from step 4, it is evident that $\beta_{it\xi}$ takes a non-zero value only if $x_{it\xi}=0 \implies \sum\limits_{j\in N_i}\bar{z}_{jk}=0$ (from step 1). Consequently, the second term in (67) is equal to zero. Hence, the objective function value of MMCLPSU-DSP is equal to θ, which is also the objective function value of MMCLPSU-PSP. Since this dual solution is feasible, it must be optimal. □

Corollary 4.1. *The Benders cut* (64) *can be expressed as:*

$$\eta\geq\theta^{h}-\sum_{j\in J}\sum_{t\in T}\left(\sum_{i\in N_j}\beta_{it\xi^{h}}^{h}\right)z_{jk(t,\xi^{h})}\quad\forall h\in H \tag{68}$$

where ξ^h *is* $\operatorname*{argmax}\limits_{s\in S}\left(\zeta_s^{*}-\sum\limits_{i\in I}\sum\limits_{t\in T}d_{it}x_{its}\right)$ *associated with the extreme point* h*.*

Proof. This follows directly from substituting the values of the dual variables in (67), using z_{jk} as a variable, and rearranging the terms. □

Proposition 5. *Let* s_1 *and* s_2 *be any two scenarios in step 2 of Algorithm 1 such that* $\theta=\zeta_{s_1}^{*}-\sum\limits_{i\in I}\sum\limits_{t\in T}d_{it}x_{its_1}=\zeta_{s_2}^{*}-\sum\limits_{i\in I}\sum\limits_{t\in T}d_{it}x_{its_2}$ *and* $\sum\limits_{t'\leq t}p_{t's_1}\leq\sum\limits_{t'\leq t}p_{t's_2}\ \ \forall t\in T$*, then the Benders cut provided by* $\xi\leftarrow s_2$ *is at least as strong as that provide by* $\xi\leftarrow s_1$*.*

Proof. Clearly, from step 1, we know that MMCLPSU-PSP has a unique optimal solution for a given solution $\bar{z}_{jk}$ to MMCLPSU-MP. Let that solution to MMCLPSU-PSP be $\bar{x}_{its},\bar{\theta}$. However, MMCLPSU-DSP may have multiple optimal solutions corresponding to this primal optimal solution (depending on the choice of ξ in step 2 of Algorithm 1). Let two such optimal solutions be associated with the extreme points h_1 and h_2 of MMCLPSU-DSP. Let $\alpha_s^{h_1},\beta_{its}^{h_1},\rho_{its}^{h_1}$ and $\alpha_s^{h_2},\beta_{its}^{h_2},\rho_{its}^{h_2}$ be the optimal solutions to MMCLPSU-DSP at the extreme points h_1 and h_2. Furthermore, let $\xi\leftarrow s_1$ at the extreme point h_1 and $\xi\leftarrow s_2$ at the extreme point h_2 (in step 2 of Algorithm 1) such that $\theta^{h_1} = \zeta^*_{s_1} - \sum_{i \in I}\sum_{t \in T} d_{it} x^{h_1}_{its_1} = \zeta^*_{s_2} - \sum_{i \in I}\sum_{t \in T} d_{it} x^{h_2}_{its_2} = \theta^{h_2} = \theta$. Then, the Benders cut $\eta \geq \theta^{h_2} - \sum_{j \in J}\sum_{t \in T}\left(\sum_{i \in N_j} \beta^{h_2}_{it\xi^{h_2}}\right) z_{jk(t,\xi^{h_2})}$ is at least as strong as the Benders cut $\eta \geq \theta^{h_1} - \sum_{j \in J}\sum_{t \in T}\left(\sum_{i \in N_j} \beta^{h_1}_{it\xi^{h_1}}\right) z_{jk(t,\xi^{h_1})}$ if:

$$\theta^{h_1} - \sum_{i \in I}\sum_{t \in T}\left(\beta^{h_1}_{its_1}\sum_{j \in N_i} z_{jk(t,s_1)}\right) \leq \theta^{h_2} - \sum_{i \in I}\sum_{t \in T}\left(\beta^{h_2}_{its_2}\sum_{j \in N_i} z_{jk(t,s_2)}\right) \tag{69}$$

Since, $\theta^{h_1} = \theta^{h_2}$, the above condition reduces to:

$$\sum_{i \in I}\sum_{t \in T}\left(\beta^{h_1}_{its_1}\sum_{j \in N_i} z_{jk(t,s_1)}\right) \geq \sum_{i \in I}\sum_{t \in T}\left(\beta^{h_2}_{its_2}\sum_{j \in N_i} z_{jk(t,s_2)}\right) \tag{70}$$

We now prove that (70) is indeed true. For this, let $\sum_{t' \leq t} p_{t's_1} = k_t$ and $\sum_{t' \leq t} p_{t's_2} = k'_t \;\; \forall t \in T$. It is given that $k_t \leq k'_t \;\; \forall t \in T$. Also, (from (62)), we get $z_{j(k-1)} \leq z_{jk} \;\; \forall j \in J, k \in \{1, 2, ..., n\}$. Hence, any feasible solution to MMCLPSU-MP should satisfy: $\bar{z}_{jk_t} \leq \bar{z}_{jk'_t} \;\; \forall j \in J, t \in T$. Therefore, step 1 of Algorithm 1 gives the following relation: $x_{its_1} \leq x_{its_2} \;\; \forall i \in I, t \in T$. This, together with steps 4 and 5 of Algorithm 1, gives: $\beta^{h_1}_{its_1} \geq \beta^{h_2}_{its_2} \;\; \forall i \in I, t \in T$. This proves that (70) is true, which proves the proposition. □

3.2. Implementation of Benders decomposition

Benders decomposition described above is the classical textbook version. In the classical implementation of Benders decomposition, the master problem is solved to optimality at each iteration, which becomes increasingly difficult with each successive iteration. The modern version of Benders decomposition, therefore, uses an incumbent solution in the branch-and-bound search tree to be passed to the sub-problem for Benders cut generation. This is facilitated by the flexibility provided by commercial solvers (like CPLEX) to the users to intervene in the branch-and-bound tree search process (using callbacks in CPLEX). In this framework, the master problem is solved to optimality only once. Moreover, the generated Benders cuts are added to the master problem as *lazy constraints*. Bai and Rubin (2009); Fortz and Poss (2009); Botton et al. (2013) have found this implementation to be more efficient than the classical version of Benders decomposition. Figure 2 presents a flowchart of this implementation of the Benders decomposition algorithm.

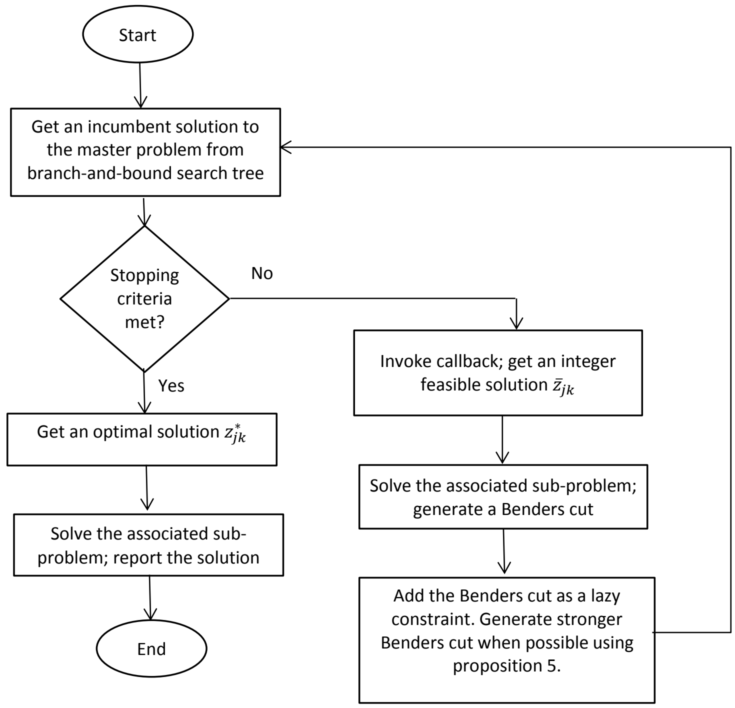

Figure 2: Flowchart for Benders decomposition implementation

4. Computational study

In this section, we describe the data generation scheme used for our computational experiments, followed by a discussion of the computational results.

4.1. Data generation

We use the following scheme to generate the data used in our computational study. The number of demand nodes $m \in \{200, 300, 400, 500\}$. X and Y coordinates of all the demand nodes are generated as $X \sim U[0, 100]$ and $Y \sim U[0, 100]$. The number of candidate facilities $n \in \{10, 15, 20\}$. These candidate facilities are randomly selected as a subset of the m demand nodes. This gives us 12 (=4x3) combinations for both the MMCLPSU and MMCLPSU-P models. In all our experiments, the set J^b of open facilities at the start of the planning horizon is assumed to be empty. Distance δ_{ij} between demand node i and candidate facility location j is taken as the Euclidean distance $\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$. Covering distance in MMCLPSU is fixed as $\delta_0 = 20$ for $n = 10, 15$ and $\delta_0 = 15$ for $n = 20$. Maximum and minimum covering distances in MMCLPSU-P are fixed as $\delta_{max} = 30$ and $\delta_{min} = 20$ for $n = 10, 15$, while $\delta_{max} = 25$ and $\delta_{min} = 15$ for $n = 20$. Coverage is assumed to decrease linearly between δ_{min} and δ_{max}, implying the following coverage function:

$$a_{ij} = \begin{cases} 1 & \text{if } \delta_{ij} \le \delta_{min}, \\ 1 - \frac{\delta_{ij}-\delta_{min}}{\delta_{max}-\delta_{min}} = \frac{\delta_{max}-\delta_{ij}}{\delta_{max}-\delta_{min}} & \text{if } \delta_{min} < \delta_{ij} \le \delta_{max}, \\ 0 & \text{if } \delta_{ij} > \delta_{max}. \end{cases}$$

The first period demand at any demand node i is generated as $d_{i1} \sim U[50, 1500]$. Demand at node i in successive periods of the planning horizon varies as $d_{it} = d_{i(t-1)}(1 + g_i)$, where g_i is the demand growth rate at node i, generated as $g_i \sim U[-0.04, 0.10]$. Please note that our data generation scheme allows for positive as well as negative growth rates. This is based on our observation that some of the villages in the Dangs district of Gujarat, India, exhibit negative growth rates.

Length of the planning horizon ($|T|$) in all experiments is assumed to be 5 periods. We assume that by the end of the planning horizon (i.e., in period $t = 5$), servers are available for all the candidate facilities under any scenario. With this assumption, all facility opening sequences give the same demand coverage in the last period. Hence, the last period $t = 5$ can be excluded from the model since it does not make any contribution to the regret. Clearly, any problem with $|T|$ periods with such an assumption is equivalent to a corresponding problem with $|T| - 1$ periods without this assumption. Thus, for a problem with 5 periods and n candidate facilities, finding the total number of possible scenarios under this assumption is equivalent to the classic problem of arranging n identical balls in 5 different cells, given by $\binom{n+4}{n} = \frac{(n+4)!}{n!4!}$ (see Feller, 1968, pp. 38). We use Algorithm 2 to enumerate all possible scenarios of availability of n servers for $|T| = 5$.

Algorithm 2 Generation of server availability scenarios

$s \leftarrow 0$;
for $t_1 := 0$ **to** n **step** 1 **do**
  for $t_2 := 0$ **to** $n - t_1$ **step** 1 **do**
    for $t_3 := 0$ **to** $n - t_1 - t_2$ **step** 1 **do**
      for $t_4 := 0$ **to** $n - t_1 - t_2 - t_3$ **step** 1 **do**
        $s \leftarrow s + 1$;
        $p_{1s} \leftarrow t_1, p_{2s} \leftarrow t_2, p_{3s} \leftarrow t_3, p_{4s} \leftarrow t_4, p_{5s} \leftarrow n - t_1 - t_2 - t_3 - t_4$;
      end for
    end for
  end for
end for

4.2. Computational results

Computational study is done on the data generated using the scheme described above. All the experiments are run on a personal computer with Intel Core i5 (3.30 GHz) processor; 4 GB RAM; and windows 64-bit operating system. Solution algorithms are coded in C++ (Visual Studio 2010), and IBM ILOG CPLEX 12.4 is used as the MIP solver. In all our experiments, the maximal coverage ζ_s^* for each scenario $s \in S$ is obtained by solving $MMCLP_s$ for complete coverage and $MMCLP\text{-}P_s$ for gradual coverage using the CPLEX MIP solver. The total CPU time taken to obtain ζ_s^* across all scenarios is less than 200 and 4000 seconds for the complete and gradual coverage, respectively, even for the largest problem instance that we solve. These times are much smaller than the CPU times taken by the CPLEX MIP solver to solve MMCLPSU and MMCLPSU-P, respectively. Hence, we do not include these times in the total CPU times reported in all our experiments.

It is clear from Table 2 that the problem size (number of variables and constraints) increases with the number of scenarios considered. Hence, in all our experiments, we use scenario dominance rules given by Vatsa and Ghosh (2014) to reduce the size of the problem. For this, we represent any server availability scenario s as $(b_1, b_2, .., b_t, .., b_{|T|})$ where b_t is the number of new servers that become available in period $t \in T$. A facility opening sequence Π is represented as $\Pi = (\pi_1, \pi_2, ..., \pi_n)$, where π_i is the i^{th} facility in the facility opening sequence. Furthermore, $\bar{d}_{t,\pi_i \cup \pi_{i+1} \cup \ldots \cup \pi_j}$ is the total demand covered by the set of facilities $\{\pi_i, \pi_{i+1}, ..., \pi_j\}$ in period t.

Rule 1:. Scenarios in which all n servers become available in the same period will have zero regret for any facility opening sequence. Hence, the regret associated with these scenarios can never be greater than the regret associated with any other scenario.

Rule 2:. Consider a scenario $s_1 = (0, .., b_t, .., b_{|T|})$ that has no new server available in the first period, and the first new server available in period t. Compare s_1 with another scenario $s_2 = (1, .., b_t - 1, .., b_{|T|})$. For any facility opening sequence Π, the regret associated with s_1 can never be greater than that associated with s_2 if:

$$\bar{d}_{1,\pi_1} + \bar{d}_{2,\pi_1} + .. + \bar{d}_{(t-1),\pi_1} \leq \zeta_{s_2}^* - \zeta_{s_1}^* \tag{71}$$

Rule 3:. Consider a scenario $s_1 = (b_1, .., b_t, .., 0)$ that has no new server available in the last period, and the n^{th} new server available in period t. Compare s_1 with another scenario $s_2 = (b_1, .., b_t - 1, .., 1)$. For any facility opening sequence Π, the regret associated with s_1 can never be greater than that associated with s_2 if:

$$\left(\bar{d}_{(|T|-1),J} - \bar{d}_{(|T|-1),J\setminus\pi_n}\right) + \left(\bar{d}_{(|T|-2),J} - \bar{d}_{(|T|-2),J\setminus\pi_n}\right) + ... + \left(\bar{d}_{t,J} - \bar{d}_{t,J\setminus\pi_n}\right) \geq \zeta_{s_1}^* - \zeta_{s_2}^* \tag{72}$$

Rule 4:. Consider a scenario $s_1 = (1, .., b_t, .., b_{|T|})$ that has 1 new server available in the first period, and the second new server available in period t. Compare s_1 with another scenario $s_2 = (2, .., b_t - 1, .., b_{|T|})$. For any facility opening sequence Π, the regret associated with s_1 can never be greater than that associated with s_2 if:

$$\left(\bar{d}_{1,\pi_1 \cup \pi_2} - \bar{d}_{1,\pi_1}\right) + \left(\bar{d}_{2,\pi_1 \cup \pi_2} - \bar{d}_{2,\pi_1}\right) + .. + \left(\bar{d}_{(t-1),\pi_1 \cup \pi_2} - \bar{d}_{(t-1),\pi_1}\right) \leq \zeta^*_{s_2} - \zeta^*_{s_1} \tag{73}$$

Rule 5:. Consider a scenario $s_1 = (b_1, .., b_t, .., 1)$ that has 1 new server available in the last period, and the $(n-1)^{th}$ new server available in period t. Compare s_1 with another scenario $s_2 = (b_1, .., b_t - 1, .., 2)$. For any facility opening sequence Π, the regret associated with s_1 can never be greater than that associated with s_2 if:

$$\begin{aligned} &\left(\bar{d}_{(|T|-1),J\setminus\pi_n} - \bar{d}_{(|T|-1),J\setminus\{\pi_n,\pi_{n-1}\}}\right) + \left(\bar{d}_{(|T|-2),J\setminus\pi_n} - \bar{d}_{(|T|-2),J\setminus\{\pi_n,\pi_{n-1}\}}\right) + ... \\ &.. + \left(\bar{d}_{t,J\setminus\pi_n} - \bar{d}_{t,J\setminus\{\pi_n,\pi_{n-1}\}}\right) \geq \zeta^*_{s_1} - \zeta^*_{s_2} \end{aligned} \tag{74}$$

Rule 6:. Consider a scenario $s_1 = (2, .., b_t, .., b_{|T|})$ that has 2 new servers available in the first period, and the third new server available in period t. Compare s_1 with another scenario $s_2 = (3, .., b_t - 1, .., b_{|T|})$. For any facility opening sequence Π, the regret associated with s_1 can never be greater than that associated with s_2 if:

$$\begin{aligned} &\left(\bar{d}_{1,\pi_1\cup\pi_2\cup\pi_3} - \bar{d}_{1,\pi_1\cup\pi_2}\right) + \left(\bar{d}_{2,\pi_1\cup\pi_2\cup\pi_3} - \bar{d}_{2,\pi_1\cup\pi_2}\right) + .. \\ &.. + \left(\bar{d}_{(t-1),\pi_1\cup\pi_2\cup\pi_3} - \bar{d}_{(t-1),\pi_1\cup\pi_2}\right) \leq \zeta^*_{s_2} - \zeta^*_{s_1} \end{aligned} \tag{75}$$

Rule 7:. Consider a scenario $s_1 = (b_1, .., b_t, .., 2)$ that has 2 new servers available in the last period, and the $(n-2)^{th}$ new server available in period t. Compare s_1 with another scenario $s_2 = (b_1, .., b_t - 1, .., 3)$. For any facility opening sequence Π, the regret associated with s_1 can never be greater than that associated with s_2 if:

$$\begin{aligned} &\left(\bar{d}_{(|T|-1),J\setminus\{\pi_n,\pi_{n-1}\}} - \bar{d}_{(|T|-1),J\setminus\{\pi_n,\pi_{n-1},\pi_{n-2}\}}\right) + \left(\bar{d}_{(|T|-2),J\setminus\{\pi_n,\pi_{n-1}\}} - \bar{d}_{(|T|-2),J\setminus\{\pi_n,\pi_{n-1},\pi_{n-2}\}}\right) \\ &+ ... + \left(\bar{d}_{t,J\setminus\{\pi_n,\pi_{n-1}\}} - \bar{d}_{t,J\setminus\{\pi_n,\pi_{n-1},\pi_{n-2}\}}\right) \geq \zeta^*_{s_1} - \zeta^*_{s_2} \end{aligned} \tag{76}$$

The above dominance rules resulted in the elimination of 50% to 65% of the total $3,876$ possible scenarios for problem instances with $n = 15$, $|T| = 5$ and $m = 100$. We use $S_0 \subset S$ to denote the remaining set of non-dominated scenarios, and replace S by S_0 in all our experiments with MMCLPSU and MMCLPSU-P.

We conduct our computational experiments with 10 instances for each of the 12 problem sizes described in Section 4.1. For MMCLPSU, Table 4 reports the objective function value (Obj), time taken by the CPLEX MIP solver (CPLEX CPU(s)), time and number of cuts required by the classic and callback versions of the Benders decomposition method (BD-Classic and BD-Callback). Clearly, BD-Classic and BD-Callback outperform the CPLEX MIP solver. For example, the computation time taken by the CPLEX MIP solver for the problem size of n=15, m=300 is on average more than 150 times the time taken by BD-Classic, and more than 250 times that taken by BD-Callback. Furthermore, the CPLEX solver could not solve MMCLSPU instances beyond the problem size n=15, m=300 within the time limit of 20 hours. Benders-Classic, on the other hand, could solve most of the problem instances till n=20, m=400 within the same time limit, while Benders-Callback solved instances of the size n=20, m=500 in close to 2 hours on average. At the same time, we notice that the number of cuts, and hence the CPU time, required by BD-Classic and BD-Callback increases with the problem size. However, the increase in CPU time is more drastic for BD-Classic since it solves a new master problem to optimality at each iteration, whereas BD-Callback solves only one master problem to optimality. Subsequently, for MMCLPSU-P, we perform computational experiments only with the CPLEX MIP solver and BD-Callback.

Table 5 provides a comparison between CPLEX MIP solver and BD-Callback for MMCLPSU-P. Like MMCLPSU, we notice that BD-Callback solves much larger instances of MMCLPSU-P compared to the CPLEX MIP solver within the time limit of 20 hours. For example, the CPLEX MIP solver could not solve 6 out of the 10 instances corresponding to $n = 10, m = 300$ within the 20 hour limit, while BD-Callback could solve all the 10 instances corresponding to the largest size of $n = 20, m = 500$ in close to 1 hour on average. Furthermore, for the instances that the CPLEX MIP solver could solve within the time limit, BD-Callback is of the order of 1,000 times faster. Moreover, comparing the results in Table 4 and 5, we notice that the CPLEX MIP solver could not solve many instances corresponding to $n = 10, m = 300$ for MMCLPSU-P, while it could solve much larger instances for MMCLPSU. This is expected since the size of the mathematical model for MMCLPSU-P is much larger than that for MMCLPSU (for example, MMCLSPU-P has variables x_{ijts} while MMCLPSU has x_{its}). Interestingly, the same does not appear to hold true with respect to BD-Callback. On the contrary, BD-Callback solves MMCLPSU-P instances much faster, on average, compared to MMCLPSU. This is true because, as it can be seen from Tables 4 and 5, the average number of Benders cuts used by BD-Callback is smaller for MMCLPSU-P compared to MMCLPSU. This indicates that the Benders cuts in MMCLPSU-P carry more information compared to the cuts in MMCLPSU (since the two master problems, MMCLPSU-MP and MMCLPSU-P-MP, differ only in Benders cuts). This can be explained by comparing the Benders cuts in the two models, which suggests that for $\delta_{min} = \delta_0$, as used in our experiments, the additional (partial) coverage allowed between δ_{min} and δ_{max} in the case of MMCLPSU-P may result in more information with each Benders cut (consequently fewer cuts). In order to check this assertion, we conducted additional experiments with 20 instances for different values of the covering distance (δ_0) in MMCLPSU. We depict the resulting average number of cuts used for each value of δ_0 in Figure 3, which confirms our assertion.

Figures 4 and 5 show how the number of instances (out of 20) with 0 maximum regret as the objective function value, and the average objective function value vary with δ_0. As obvious from Figure 4, with a sufficiently high δ_0, most of the instances result in 0 as the objective function value. This is expected since with a very high δ_0, all the facility opening sequences provide the same coverage under any given scenario. Hence, the regret associated with an optimal sequence will be zero. Furthermore, as can be seen from Figure 5, the average objective function value first increases and then decreases with an increase in δ_0. This is due to the fact that with very small δ_0, the total demand that can be covered is small in magnitude, and hence the regret is also of similar magnitude. As δ_0 increases, it becomes possible to cover more demand nodes by judiciously selecting candidate facilities to open. Consequently, when the optimal facility opening sequence has a regret, it is generally of a higher magnitude. Furthermore, a very high δ_0 results in too many instances with low or zero objective function value, as observed in Figure 4. Consequently, the average objective function value decreases with very high values of δ_0. This observation provides an interesting insight. The covering distance represents the connectivity of the nodes. A larger covering distance indicates demand nodes are well connected with candidate facilities. The above observation implies that with a good level of connectivity, even the most unfavorable scenario will not have a high regret.

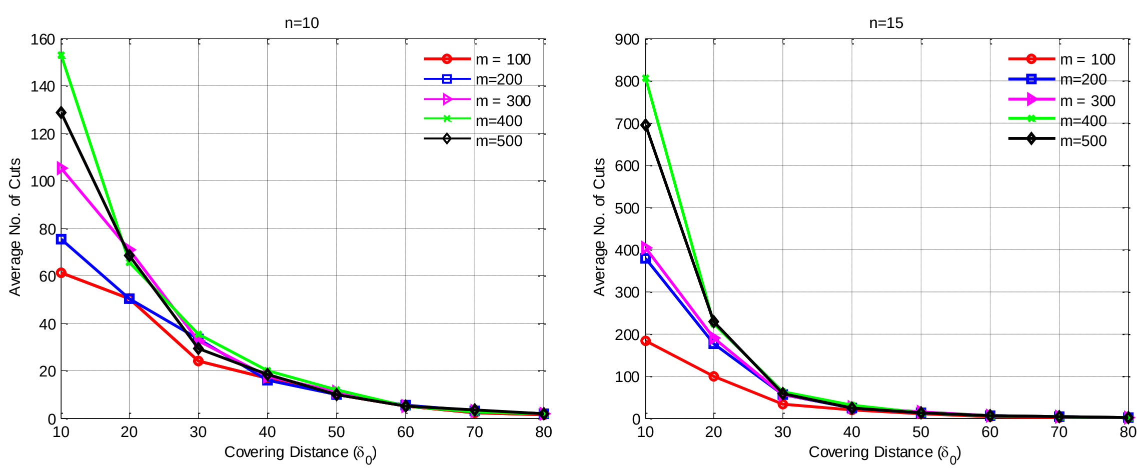

Figure 3: Average No. of Cuts vs. δ_0

5. Conclusion

In this paper, we provided a new formulation of the multi-period maximal covering (both complete and partial) location problem with server uncertainty, motivated by its relevance with respect to primary health centers. We mathematically proved that our formulation is better

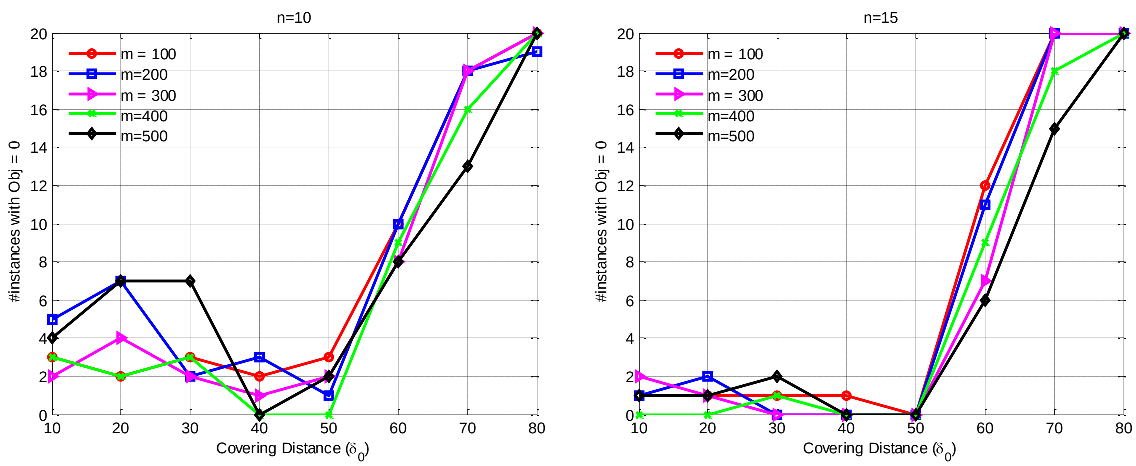

Figure 4: No. of instance with zero objective function value vs. δ_0

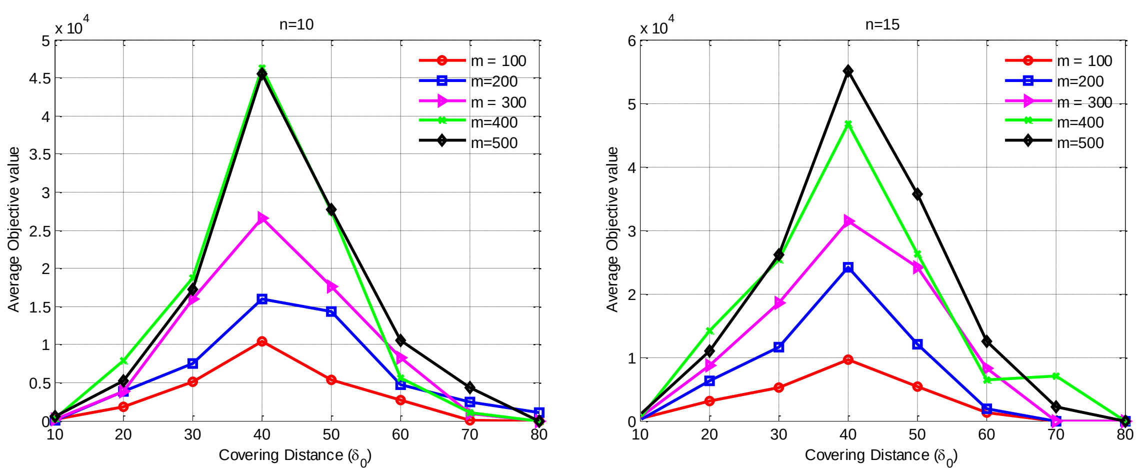

Figure 5: Average objective function value vs. δ_0

Table 4: Computational results with MMCLPSU

Ins.	Obj	CPLEX	BD-Classic		BD-Callback		Obj	CPLEX	BD-Classic		BD-Callback	
		CPU(s)	CPU(s)	Cuts	CPU(s)	Cuts		CPU(s)	CPU(s)	Cuts	CPU(s)	Cuts
	n=10, m=200						n=10, m=300					
1	4673.2	97.9	5.1	88	5.3	86	8557.7	59.7	9.3	84	12.2	114
2	7509.0	184.9	2.6	50	2.4	47	5174.2	79.7	7.1	77	5.5	56
3	19828.6	16.8	4.4	52	6.0	72	7986.4	600.7	13.3	116	12.1	116
4	115.7	26.3	2.8	38	2.9	37	255.9	41.3	6.6	71	7.9	76
5	0.0	70.0	0.7	10	1.8	25	0.0	124.8	2.0	30	2.6	29
6	8758.8	62.2	5.3	72	4.2	62	21763.9	578.0	5.7	69	5.1	61
7	0.0	42.1	1.2	26	1.3	25	4618.1	176.9	18.5	147	13.1	109
8	0.0	112.2	3.1	48	3.5	50	0.0	94.1	4.5	67	6.6	96
9	13179.4	438.9	1.2	24	1.4	32	863.1	107.2	4.1	62	4.2	61
10	0.0	54.6	0.7	10	1.3	16	13.0	60.0	4.2	62	4.4	59
Avg.		110.6	2.7	41.8	3.0	45.2		192.2	7.5	78.5	7.4	77.7
Max.		438.9	5.3	88	6.0	86		600.7	18.5	147	13.1	116
	n=10, m=400						n=10, m=500					
1	10688.6	63.7	10.0	73	9.8	61	0.0	395.0	9.6	58	15.1	87
2	374.6	55.0	12.7	99	10.6	74	1966.1	930.6	17.4	90	24.2	121
3	396.9	65.5	10.8	83	11.5	84	9289.5	2292.5	14.5	74	21.9	103
4	10033.1	85.7	10.7	82	10.0	70	0.0	92.1	10.4	66	11.9	68
5	20531.7	1382.6	7.9	58	9.8	72	5094.7	102.5	4.8	39	6.8	45
6	14744.4	970.5	13.8	74	15.7	83	7181.4	120.5	12.3	54	16.5	68
7	0.0	158.8	3.1	25	10.3	75	0.0	204.1	5.7	35	4.5	24
8	13783.7	75.1	4.1	31	5.0	34	9362.9	1197.0	15.6	71	20.5	88
9	7549.6	972.4	7.7	55	9.3	62	0.0	402.8	6.6	42	13.8	78
10	15144.2	289.1	6.1	62	6.6	64	2680.2	229.4	21.6	111	17.2	76
Avg.		411.8	8.7	64.2	9.9	67.9		596.7	11.8	64	15.2	75.8
Max.		1382.6	13.8	99	15.7	84		2292.5	21.6	111	24.2	121
	n=15, m=200						n=15, m=300					
1	13335.4	25878.3	530.6	435	96.0	401	863.5	7025.8	28.8	82	31.9	87
2	7509.0	7200.8	41.3	121	39.6	156	7991.5	28235.2	147.7	231	107.0	231
3	4833.0	8965.0	66.9	175	71.3	218	18405.0	53327.2	514.2	329	229.2	368
4	9274.4	489.0	51.2	138	34.9	136	0.0	4262.8	72.4	161	89.5	193
5	3608.8	14270.5	33.8	113	36.0	151	1582.7	1218.5	28.5	79	43.6	121
6	904.0	4637.8	88.6	241	88.5	237	20642.3	26878.8	52.8	107	89.1	225
7	3215.5	609.7	31.0	120	38.7	168	13364.7	36511.5	476.6	314	202.0	315
8	6263.6	27162.2	444.5	431	212.1	457	7406.7	1863.2	53.7	126	39.8	106
9	16059.5	1727.6	44.9	112	32.0	102	19254.6	48713.7	105.0	166	73.1	185
10	0.0	1220.4	20.9	80	20.2	82	9712.3	36164.8	107.3	238	58.7	153
Avg.		9216.1	135.4	196.6	67.0	210.8		24420.1	158.7	183.3	96.4	198.4
Max.		27162.2	530.6	435	212.1	457		53327.2	514.2	329	229.2	368
	n=15, m=400						n=15, m=500					
1	7520.3	*	159.4	185	200.3	266	20806.6	*	300.3	248	231.1	290
2	17916.3	*	81.6	118	76.8	124	9711.8	*	191.6	289	177.3	287
3	4162.9	*	247.0	248	183.1	258	12126.9	*	181.1	192	216.1	239
4	12103.4	*	174.2	163	193.1	214	4972.4	*	58.8	95	103.7	147
5	18194.5	*	86.4	125	87.9	148	17206.1	*	65.7	92	89.9	139
6	27138.1	*	207.0	242	170.4	276	9248.3	*	370.0	275	296.1	250
7	617.0	*	59.2	127	72.6	148	732.0	*	85.5	114	147.5	176
8	12120.2	*	43.9	89	49.8	97	4748.5	*	82.3	107	87.3	105
9	11109.7	*	275.2	190	230.1	270	8087.1	*	131.2	132	178.4	172
10	15144.2	*	71.2	123	64.1	133	22524.7	*	2132.3	843	880.0	896
Avg.			140.5	161	132.8	193.4			359.9	238.7	240.8	270.1
Max.			275.2	248	230.1	276			2132.3	843	880.0	896

[*] Could not be solved in 20 hours

Table 4 (continued)

Ins.	Obj	CPLEX CPU(s)	BD-Classic CPU(s)	BD-Classic Cuts	BD-Callback CPU(s)	BD-Callback Cuts	Obj	CPLEX CPU(s)	BD-Classic CPU(s)	BD-Classic Cuts	BD-Callback CPU(s)	BD-Callback Cuts
	n=20, m=200						n=20, m=300					
1	3492.3	*	*	*	5821.9	3396	5378.1	*	4727.3	679	956.9	582
2	3222.5	*	2167.5	444	648.6	493	7633.0	*	26362.8	983	3381.9	1479
3	360.0	*	448.8	379	500.9	392	3723.2	*	*	*	12338.3	3044
4	541.2	*	482.7	404	316.9	268	10221.8	*	4918.1	703	1674.3	1022
5	1806.3	*	809.0	328	389.1	366	4659.3	*	6153.2	500	1159.4	645
6	808.7	*	1165.4	541	696.3	569	0.0	*	582.0	315	528.5	359
7	2464.9	*	450.6	313	360.9	369	9978.9	*	*	*	4232.2	1143
8	561.2	*	63022.0	2095	14399.7	2404	3906.0	*	8166.7	1260	1935.1	1067
9	5813.0	*	6108.6	748	1109.9	700	6518.0	*	3655.2	806	1577.1	856
10	420.7	*	1337.0	788	812.6	635	8671.4	*	34265.6	2063	2956.1	1424
Avg.					2505.7	959.2					3074.0	1162.1
Max.					14399.7	3396					12338.3	3044
	n=20, m=400						n=20, m=500					
1	2257.3	*	11643.8	946	2481.8	1049	16066.7	*	*	*	38432.0	6642
2	8364.5	*	27470.4	990	4381.5	1273	18915.9	*	*	*	10891.1	2447
3	5297.3	*	9396.0	1416	4399.3	1147	10163.3	*	*	*	4040.3	1105
4	8180.6	*	7367.5	724	2815.3	927	5262.4	*	*	*	1812.1	637
5	8351.3	*	2443.0	556	1247.0	617	6761.3	*	*	*	1217.9	465
6	5257.1	*	4070.3	616	1469.9	572	11044.2	*	*	*	4803.8	1556
7	13696.3	*	5057.6	889	4559.9	1467	5918.8	*	*	*	1419.6	433
8	13.0	*	561.0	250	431.1	200	10376.5	*	*	*	2756.5	943
9	5021.5	*	*	*	9420.4	3097	9723.9	*	*	*	5072.3	1528
10	13158.8	*	8537.6	1037	3196.9	1140	4167.9	*	*	*	5968.6	1829
Avg.					3440.3	1148.9					7641.4	1758.5
Max.					9420.4	3097					38432.0	6642

* Could not be solved in 20 hours

Table 5: Computational results with MMCLPSU-P

Ins.	Obj	CPLEX CPU(s)	BD-Callback CPU(s)	Cuts	Obj	CPLEX CPU(s)	BD-Callback CPU(s)	Cuts	Obj	CPLEX CPU(s)	BD-Callback CPU(s)	Cuts
	n=10, m=200				n=10, m=300				n=10, m=400			
1	11272.2	3463.9	6.1	64	13040.9	18044.7	8.4	43	13199.7	*	22.4	60
2	2227.8	5060.3	4.1	34	11574.0	37982.8	9.8	48	13922.7	*	17.1	56
3	11591.5	6569.4	10.2	75	17891.7	*	15.7	79	23551.0	*	17.9	66
4	540.0	1822.0	4.0	39	16340.6	*	9.2	56	10634.6	*	9.9	47
5	823.3	2146.4	3.3	28	0.0	*	2.0	14	9475.6	*	11.6	42
6	6862.4	7209.0	8.6	79	11633.1	*	6.9	50	0.0	*	6.2	24
7	3079.5	854.7	2.4	31	10260.9	*	19.7	94	15026.5	*	25.2	90
8	2901.2	1388.9	5.6	43	12196.8	44738.32	7.9	62	30696.0	*	14.7	51
9	4972.4	931.2	2.3	22	10965.5	*	4.6	39	12520.3	*	17.2	65
10	2832.5	957.8	4.3	43	44.9	34574.86	7.6	45	20226.0	*	10.8	53
Avg.		3040.4	5.1	45.8			9.2	53			15.3	55.4
Max.		7209.0	10.2	79			19.7	94			25.2	90
	n=10, m=500				n=15, m=200				n=15, m=300			
1	55221.4	*	27.5	74	21085.2	*	55.8	142	16563.4	*	109.7	154
2	21287.3	*	22.4	82	23183.0	*	58.5	135	21128.9	*	205.0	235
3	20978.7	*	28.7	77	4635.7	*	68.3	108	13450.2	*	153.4	159
4	0.0	*	12.7	44	6605.7	*	33.9	73	19648.8	*	163.8	166
5	2168.2	*	8.0	31	7287.6	*	25.4	56	2890.1	*	41.3	72
6	5692.6	*	18.5	47	6434.4	*	78.9	137	13262.0	*	78.9	112
7	18005.9	*	7.9	27	11732.4	*	44.4	110	13293.4	*	186.8	162
8	10452.8	*	20.3	71	8372.1	*	122.3	145	15356.1	*	69.4	77
9	0.0	*	14.4	47	7480.7	*	32.5	78	10250.7	*	50.6	82
10	0.0	*	10.5	31	2832.5	*	56.2	137	18249.3	*	131.3	188
Avg.			17.1	53.1			57.6	112.1			119.0	140.7
Max.			28.7	82			122.3	145			205.0	235
	n=15, m=400				n=15, m=500				n=20, m=200			
1	13706.7	*	315.9	222	53670.4	*	149.3	125	16507.5	*	2341.0	1011
2	38421.3	*	204.8	153	9142.0	*	169.3	171	8715.6	*	533.3	285
3	18023.2	*	227.7	181	0.0	*	94.8	66	579.1	*	730.0	348
4	22114.1	*	295.9	176	13426.1	*	125.0	109	9570.6	*	2400.2	927
5	20806.4	*	100.6	96	21214.9	*	158.2	123	12815.5	*	957.8	413
6	12505.6	*	168.8	170	9824.0	*	244.5	127	5520.7	*	790.4	425
7	19116.4	*	113.4	101	33889.3	*	155.8	117	5227.8	*	509.9	310
8	37372.0	*	55.2	51	16802.1	*	152.5	116	5798.4	*	2131.7	623
9	14657.5	*	121.8	88	27899.4	*	261.2	135	10044.3	*	821.8	499
10	20226.0	*	78.0	78	19117.5	*	247.9	194	5548.5	*	596.4	311
Avg.			168.2	131.6			175.8	128.3			1181.3	515.2
Max.			315.9	222			261.2	194			2400.2	1011
	n=20, m=300				n=20, m=400				n=20, m=500			
1	8443.8	*	799.0	294	12058.1	*	4543.4	1085	18216.0	*	6104.7	1159
2	13895.6	*	2619.7	783	23681.4	*	5706.7	1075	13043.3	*	5708.7	885
3	11130.8	*	4752.6	1061	16695.2	*	5127.7	1246	33116.3	*	1590.3	338
4	4132.6	*	1505.7	442	13883.3	*	3815.9	682	38415.5	*	2788.4	640
5	7005.1	*	807.6	237	11886.5	*	1202.7	369	18873.8	*	990.8	250
6	18516.8	*	1141.4	412	16383.1	*	1650.5	412	29209.2	*	5050.1	891
7	9416.2	*	4697.5	1147	14953.3	*	2368.0	528	4355.3	*	1396.5	247
8	7788.2	*	1535.1	459	5772.3	*	526.8	156	10134.0	*	2923.1	514
9	13878.0	*	1290.1	497	17246.0	*	5265.5	1120	29415.5	*	3622.2	571
10	6352.6	*	2231.4	679	9806.0	*	1378.2	339	23821.3	*	7873.3	1446
Avg.			2138.0	601.1			3158.5	701.2			3804.8	694.1
Max.			4752.6	1147			5706.7	1246			7873.3	1446

* Could not be solved in 20 hours

than the only other formulation available in the literature. Using computational experiments, we showed that for large problem instances, our formulation is more than 10 times faster compared to the earlier formulation in the literature. Still, the CPLEX MIP solver was unable to solve large size instances. Consequently, we developed a Benders decomposition based approach, which could solve much larger problem instances within a reasonable time. We further provided refinements to the Benders method, like heuristics for the sub-problems and cut strengthening methods, which drastically reduced the computational time needed to solve problem instances with up to 20 facilities, 500 demand nodes and 5 periods. Furthermore, for the instances that the CPLEX MIP solver could solve within a time limit of 20 hours, our proposed solution method turned out to be of the order of 150 - 250 times faster for the problems with complete coverage, and around 1000 times faster for gradual coverage.

Future research may use other regret measures like maximization of expected coverage when the probabilities of various server availability scenarios can be estimated. Extension of this paper with capacity restrictions at candidate facilities is another interesting avenue for further research.

Appendix A

This appendix describes the application of the Benders decomposition approach to the gradual coverage case.

By fixing the binary variables z_{jk} as $\bar{z}_{jk}$ we obtain the following primal sub-problem:

[MMCLPSU-P-PSP:]

$$\text{Min} \quad \theta \tag{A.1}$$

$$s.t. \quad \theta \geq \zeta_s^* - \sum_{i \in I} \sum_{j \in N_i \cup N_i^b} \sum_{t \in T} a_{ij} d_{it} x_{ijts} \qquad \forall s \in S \tag{A.2}$$

$$x_{ijts} \leq \bar{z}_{jk(t,s)} \qquad \forall i \in I, j \in N_i, \forall t \in T, s \in S \tag{A.3}$$

$$\sum_{j \in N_i \cup N_i^b} x_{ijts} \leq 1 \qquad \forall i \in I, t \in T, s \in S \tag{A.4}$$

$$x_{ijts} \geq 0 \qquad \forall i \in I, j \in N_i \cup N_i^b, t \in T, s \in S \tag{A.5}$$

$$\theta \geq 0 \tag{A.6}$$

Let α_s, β_{ijts} and γ_{its} be the dual variables associated with (A.2), (A.3) and (A.4), respectively. The dual sub-problem is formulated as:

[MMCLPSU-P-DSP:]

$$\text{Max} \quad \sum_{s \in S} \zeta_s^* \alpha_s - \sum_{i \in I} \sum_{t \in T} \sum_{s \in S} \gamma_{its} - \sum_{i \in I} \sum_{j \in N_i} \sum_{t \in T} \sum_{s \in S} \beta_{ijts} \bar{z}_{jk(t,s)} \tag{A.7}$$

$$
\begin{aligned}
s.t.\quad & a_{ij} d_{it} \alpha_s - \beta_{ijts} - \gamma_{its} \leq 0 && \forall i \in I, j \in N_i \cup N_i^b, t \in T, s \in S && \text{(A.8)}\\
& \sum_{s \in S} \alpha_s \leq 1 && && \text{(A.9)}\\
& \alpha_s, \beta_{ijts}, \gamma_{its} \geq 0 && \forall i \in I, j \in N_i \cup N_i^b, t \in T, s \in S && \text{(A.10)}
\end{aligned}
$$

Let N_j be the set of demand nodes that can be covered completely or partially by any candidate facility j, i.e., $N_j = \{i \in I : a_{ij} > 0\}$. The master problem can be formulated as follows:

[MMCLPSU-P-MP:]

$$
\begin{aligned}
\text{Min}\quad & \eta && && \text{(A.11)}\\
s.t.\quad & z_{jk} \geq z_{j(k-1)} && \forall j \in J, k \geq 1 && \text{(A.12)}\\
& \sum_{j \in J} z_{jk} = k && \forall k \in \{0, 1, ..., n\} && \text{(A.13)}\\
& \eta \geq \sum_{s \in S} \zeta_s^* \alpha_s^h - \sum_{i \in I} \sum_{t \in T} \sum_{s \in S} \gamma_{its}^h - \sum_{j \in J} \sum_{t \in T} \sum_{s \in S} \left(\sum_{i \in N_j} \beta_{ijts}^h \right) z_{jk(t,s)} && \forall h \in H && \text{(A.14)}\\
& z_{jk} \in \{0, 1\} && \forall j \in J, k \in \{0, 1, ..., n\} && \text{(A.15)}
\end{aligned}
$$

Proposition A.1. *The primal sub-problem MMCLPSU-P-PSP is always feasible and bounded for any feasible solution $\bar{z}_{jk}$ to MMCLPSU-P-MP.*

Proof. This can be proved along similar lines as the proof for Proposition 3. □

We now present an algorithm to efficiently solve MMCLPSU-P-DSP.

Algorithm A.1 Dual sub-problem solution

1: set $X_{its} \leftarrow \max_{j \in N_i \cup N_i^b} a_{ij} \bar{z}_{jk} \quad \forall i \in I, t \in T, s \in S$, where $k = \sum_{t' \leq t} p_{t's}$;
2: $\theta \leftarrow \max_{s \in S} \left(\zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} X_{its} \right)$, $\xi \leftarrow \operatorname{argmax}_{s \in S} \left(\zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} X_{its} \right)$. Ties are broken using Proposition 5;
3: set $\alpha_\xi \leftarrow 1, \alpha_s \leftarrow 0 \quad \forall s \in S \backslash \{\xi\}, \beta_{ijts} \leftarrow 0 \quad \forall i \in I, j \in N_i^b, t \in T, s \in S$;
4: **if** $X_{its} = 0$ **then** set $\gamma_{its} \leftarrow 0, \beta_{ijts} \leftarrow a_{ij} d_{it} \alpha_s, \quad \forall i \in I, j \in N_i, t \in T, s \in S$;
5: **else** set $\gamma_{its} \leftarrow X_{its} d_{it} \alpha_s$ and $\beta_{ijts} \leftarrow max(0, a_{ij} d_{it} \alpha_s - \gamma_{its}) \quad \forall i \in I, j \in N_i, t \in T, s \in S$;
6: **end if**
7: output $\alpha_s, \beta_{ijts}, \gamma_{its} \quad \forall i \in I, j \in N_i, t \in T, s \in S$.

Proposition A.2. *For a given solution $\bar{z}_{jk}$ to MMCLPSU-P-MP, Algorithm A.1 gives an optimal solution to MMCLPSU-P-DSP.*

Proof. First, we prove that Algorithm A.1 gives a feasible solution to MMCLPSU-P-DSP. Let us define $X_{its} = \max_{j \in N_i \cup N_i^b} a_{ij} x_{ijts} \ \ \forall i \in I, t \in T, s \in S$ as the maximum level (fraction) of coverage possible for node i in time period t and scenario s. X_{its}, by this definition, is also equal to $\max_{j \in N_i \cup N_i^b} a_{ij} \bar{z}_{jk} \ \ \forall i \in I, t \in T, s \in S$, where $k = \sum_{t' \leq t} p_{t's}$, as shown in step 1. Clearly, steps 1 and 2 together solve MMCLPSU-P-PSP optimally. The solution to MMCLPSU-P-DSP is obtained in steps 3 to 6 using complementary slackness conditions between MMCLPSU-P-PSP and MMCLPSU-P-DSP.

Applying the complementary slackness condition to (A.2) gives: $(\theta + \sum_{i \in I} \sum_{j \in N_i \cup N_I^b} \sum_{t \in T} d_{it} x_{ijts} - \zeta_s^*)\alpha_s = 0 \ \ \forall s \in S$. This, together with (A.9), gives as feasible solution $\alpha_\xi = 1$, where $\xi = \operatorname{argmax}_{s \in S}(\zeta_s^* - \sum_{i \in I} \sum_{t \in T} d_{it} X_{its})$ and $\alpha_s = 0 \ \ \forall s \in S \backslash \{\xi\}$ in step 3. $X_{its} = 0$, by its definition, implies $x_{ijts} = 0 \ \ \forall j \in N_i \cup N_i^b$ for any $i \in I, t \in T, s \in S$. In step 4, the above result, together with the complementary slackness condition on constraint set (A.4), gives $\gamma_{its} = 0$ when $X_{its} = 0 \ \ \forall i \in I, t \in T, s \in S$. Furthermore, β_{its} is obtained in step 4 using the values of α_s and γ_{its} in (A.8) and exploiting the fact that (A.8) is binding at optimality. On the other hand, when $X_{its} \neq 0$, step 5 gives feasible values for β_{its} and ρ_{its} using (A.8). The intuition behind this step comes from the interpretation of the dual variables that an increase of one unit in the RHS of (A.4) implies an improvement of $X_{its} d_{it} \alpha_s$ in the objective function value (double counting demand covered at node i in period t and scenario s).

We now show that this solution is the optimal solution to MMCLPSU-P-DSP. With this solution obtained using Algorithm A.1, MMCLPSU-P-DSP objective function (A.7) is expressed as:

$$\theta - \sum_{i \in I} \sum_{j \in N_i} \sum_{t \in T} \beta_{ijt\xi} \bar{z}_{jk(t,\xi)} \tag{A.16}$$

As in the problem with complete coverage, the second term of (A.16) is equal to zero. This is because β_{ijts} takes a positive value only when $z_{jk} = 0$, where $k = \sum_{t' \leq t} p_{t'\xi}$ (follows from steps 1, 4 and 5). Consequently, the dual and primal objective function values are the same (equal to θ), and hence the Algorithm A.1 solves the MULLPSU-P-2-DSP to optimality. $\square$

Although ties in step 2 of Algorithm A.1 can be broken arbitrarily, breaking them using Proposition 5 is guaranteed to generate a Benders cut that is no weaker than any other Benders cut generated by breaking ties arbitrarily. The proof for this is similar to that for Proposition 5, and hence we skip the details.

Corollary A.2.1. *The Benders cut* (A.14) *can be expressed as:*

$$\eta \geq \theta^h - \sum_{j \in J} \sum_{t \in T} \left(\sum_{i \in N_j} \beta^h_{ijt\xi^h} \right) z_{jk(t,\xi^h)} \quad \forall h \in H \tag{A.17}$$

where ξ^h *is* $\underset{s \in S}{\operatorname{argmax}} \left(\zeta^*_s - \sum_{i \in I} \sum_{t \in T} d_{it} X_{its} \right)$ *associated with the extreme point* h*.*

Proof. This follows directly from substituting the values of the dual variables in (A.16), using z_{jk} as a variable, and rearranging the terms. □

Acknowledgements

The authors would like to thank the anonymous referees for their constructive comments on the earlier version of the paper.

Albareda-Sambola, M., Alonso-Ayuso, A., Escudero, L. F., Fernández, E., Pizarro, C., 2013. Fix-and-relax-coordination for a multi-period location–allocation problem under uncertainty. Computers & Operations Research 40 (12), 2878–2892.

Albareda-Sambola, M., Fernández, E., Saldanha da Gama, F., 2011. The facility location problem with Bernoulli demands. Omega 39 (3), 335–345.

Antunes, A., Peeters, D., 2000. A dynamic optimization model for school network planning. Socio-Economic Planning Sciences 34 (2), 101–120.

Antunes, A., Peeters, D., 2001. On solving complex multi-period location models using simulated annealing. European Journal of Operational Research 130 (1), 190–201.

Arabani, A. B., Farahani, R. Z., 2012. Facility location dynamics: An overview of classifications and applications. Computers & Industrial Engineering 62 (1), 408–420.

Averbakh, I., Berman, O., 1997. Minimax regret p-center location on a network with demand uncertainty. Location Science 5 (4), 247–254.

Averbakh, I., Berman, O., 2000. Minmax regret median location on a network under uncertainty. INFORMS Journal on Computing 12 (2), 104–110.

Bai, L., Rubin, P. A., 2009. Combinatorial Benders cuts for the minimum tollbooth problem. Operations Research 57 (6), 1510–1522.

Balcik, B., Beamon, B. M., 2008. Facility location in humanitarian relief. International Journal of Logistics 11 (2), 101–121.

Ballou, R. H., 1968. Dynamic warehouse location analysis. Journal of Marketing Research 5 (3), 271–276.

Baray, J., Cliquet, G., 2013. Optimizing locations through a maximum covering/p-median hierarchical model: Maternity hospitals in France. Journal of Business Research 66 (1), 127–132.

Benders, J. F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4 (1), 238–252.

Berman, O., Drezner, Z., Krass, D., 2010. Generalized coverage: New developments in covering location models. Computers & Operations Research 37 (10), 1675–1687.

Berman, O., Krass, D., Drezner, Z., 2003. The gradual covering decay location problem on a network. European Journal of Operational Research 151 (3), 474–480.

Berman, O., Wang, J., 2011. The minmax regret gradual covering location problem on a network with incomplete information of demand weights. European Journal of Operational Research 208 (3), 233–238.

Botton, Q., Fortz, B., Gouveia, L., Poss, M., 2013. Benders decomposition for the hop-constrained survivable network design problem. INFORMS Journal on Computing 25 (1), 13–26.

Burkard, R. E., Dollani, H., 2002. A note on the robust 1-center problem on trees. Annals of Operations Research 110 (1-4), 69–82.

Canel, C., Khumawala, B., Law, J., Loh, A., 2001. An algorithm for the capacitated, multi-commodity multi-period facility location problem. Computers & Operations Research 28 (5), 411–427.

Chen, B., Lin, C.-S., 1998. Minmax-regret robust 1-median location on a tree. Networks 31 (2), 93–103.

Cho, S.-H., Jang, H., Lee, T., Turner, J., 2014. Simultaneous location of trauma centers and helicopters for emergency medical service planning. Operations Research 62 (4), 751–771.

Church, R., Roberts, K. L., 1983. Generalized coverage models and public facility location. In: Papers of the Regional Science Association. Vol. 53. Springer, pp. 117–135.

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

Correia, I., Saldanha da Gama, F., 2015. Facility location under uncertainty. In: Location Science. Springer, pp. 177–203.

Costa, A. M., 2005. A survey on Benders decomposition applied to fixed-charge network design problems. Computers & Operations Research 32 (6), 1429–1450.

Current, J., Ratick, S., ReVelle, C., 1998. Dynamic facility location when the total number of facilities is uncertain: A decision analysis approach. European Journal of Operational Research 110 (3), 597–609.

de Camargo, R. S., de Miranda, G., Ferreira, R. P., 2011. A hybrid outer-approximation/Benders decomposition algorithm for the single allocation hub location problem under congestion. Operations Research Letters 39 (5), 329–337.

de Camargo, R. S., de Miranda Jr, G., Luna, H. P. L., 2009. Benders decomposition for hub location problems with economies of scale. Transportation Science 43 (1), 86–97.

Dias, J., Captivo, M., Clímaco, J., 2008. A memetic algorithm for multi-objective dynamic location problems. Journal of Global Optimization 42 (2), 221–253.

Dias, J., Captivo, M. E., Clímaco, J., 2006. Capacitated dynamic location problems with opening, closure and reopening of facilities. IMA Journal of Management Mathematics 17 (4), 317–348.

Dias, J., Eugénia Captivo, M., Clímaco, J., 2007. Efficient primal-dual heuristic for a dynamic location problem. Computers & Operations Research 34 (6), 1800–1823.

Erlenkotter, D., 1981. A comparative study of approaches to dynamic location problems. European Journal of Operational Research 6 (2), 133–143.

Feller, V., 1968. An Introduction to Probability Theory and Its Applications: Volume 1. J. Wiley & sons.

Fortz, B., Poss, M., 2009. An improved Benders decomposition applied to a multi-layer network design problem. Operations Research Letters 37 (5), 359–364.

Geoffrion, A. M., Graves, G. W., 1974. Multicommodity distribution system design by Benders decomposition. Management Science 20 (5), 822–844.

Ghaderi, A., Jabalameli, M. S., 2013. Modeling the budget-constrained dynamic uncapacitated facility location–network design problem and solving it via two efficient heuristics: A case study of health care. Mathematical and Computer Modelling 57 (3), 382–400.

Gunawardane, G., 1982. Dynamic versions of set covering type public facility location problems. European Journal of Operational Research 10 (2), 190–195.

Hernández, P., Alonso-Ayuso, A., Bravo, F., Escudero, L. F., Guignard, M., Marianov, V., Weintraub, A., 2012. A branch-and-cluster coordination scheme for selecting prison facility sites under uncertainty. Computers & Operations Research 39 (9), 2232–2241.

Jayaswal, S., 2014. Emergency medical service system design under service level constraints for heterogeneous patients. Working Paper WP2014-11-04, Indian Institute of Management Ahmedabad.

Karasakal, O., Karasakal, E. K., 2004. A maximal covering location model in the presence of partial coverage. Computers & Operations Research 31 (9), 1515–1526.

Killmer, K. A., Anandalingam, G., Malcolm, S. A., 2001. Siting noxious facilities under uncertainty. European Journal of Operational Research 133 (3), 596–607.

Kouvelis, P., Yu, G., 1996. Robust discrete optimization and its applications. Vol. 14. Springer.

Melo, M. T., Nickel, S., Saldanha da Gama, F., 2006. Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. Computers & Operations Research 33 (1), 181–208.

Nickel, S., Saldanha da Gama, F., 2015. Multi-period facility location. In: Location Science. Springer, pp. 289–310.

Nickel, S., Saldanha da Gama, F., Ziegler, H.-P., 2012. A multi-stage stochastic supply network design problem with financial decisions and risk management. Omega 40 (5), 511–524.

Pacheco, J. A., Casado, S., 2005. Solving two location models with few facilities by using a hybrid heuristic: A real health resources case. Computers & Operations Research 32 (12), 3075–3091.

Rahman, S., Smith, D. K., 2000. Use of location-allocation models in health service development planning in developing nations. European Journal of Operational Research 123 (3), 437–452.

Ramirez-Nafarrate, A., Lyon, J. D., Fowler, J. W., Araz, O. M., 2015. Point-of-dispensing location and capacity optimization via a decision support system. Production and Operations Management 24 (8), 1311–1328.

Rohde, J., Cousens, S., Chopra, M., Tangcharoensathien, V., Black, R., Bhutta, Z. A., Lawn, J. E., 2008. 30 years after Alma-Ata: Has primary health care worked in countries? The Lancet 372 (9642), 950–961.

Saldanha da Gama, F., Captivo, M., 1998. A heuristic approach for the discrete dynamic location problem. Location Science 6 (1), 211–223.

Schilling, D. A., Revelle, C., Cohon, J., Elzinga, D. J., 1980. Some models for fire protection locational decisions. European Journal of Operational Research 5 (1), 1–7.

Shulman, A., 1991. An algorithm for solving dynamic capacitated plant location problems with discrete expansion sizes. Operations Research 39 (3), 423–436.

Silva, F., Serra, D., 2008. Locating emergency services with different priorities: the priority queuing covering location problem. Journal of the Operational Research Society 59 (9), 1229–1238.

Snyder, L. V., 2006. Facility location under uncertainty: a review. IIE Transactions 38 (7), 547–564.

Thanh, P., Bostel, N., Péton, O., 2008. A dynamic model for facility location in the design of complex supply chains. International Journal of Production Economics 113 (2), 678–693.

Vairaktarakis, G. L., Kouvelis, P., 1999. Incorporation dynamic aspects and uncertainty in 1-median location problems. Naval Research Logistics 46 (2), 147–168.

Vatsa, A. K., Ghosh, D., 2014. Tabu search for multi-period facility location: Uncapacitated problem with an uncertain number of servers. Working Paper WP2014-11-03, Indian Institute of Management Ahmedabad.

Verter, V., Lapierre, S. D., 2002. Location of preventive health care facilities. Annals of Operations Research 110 (1-4), 123–132.

Walley, J., Lawn, J. E., Tinker, A., De Francisco, A., Chopra, M., Rudan, I., Bhutta, Z. A., Black, R. E., 2008. Primary health care: Making Alma-Ata a reality. The Lancet 372 (9642), 1001–1007.

Wang, Q., Batta, R., Bhadury, J., M Rump, C., 2003. Budget constrained location problem with opening and closing of facilities. Computers & Operations Research 30 (13), 2047–2069.

Wang, Q., Batta, R., Rump, C. M., 2002. Algorithms for a facility location problem with stochastic customer demand and immobile servers. Annals of Operations Research 111 (1-4), 17–34.

Wentges, P., 1996. Accelerating Benders' decomposition for the capacitated facility location problem. Mathematical Methods of Operations Research 44 (2), 267–290.

Wesolowsky, G., Truscott, W., 1975. The multiperiod location-allocation problem with relocation of facilities. Management Science 22 (1), 57–65.

Zhang, Y., Berman, O., Verter, V., 2012. The impact of client choice on preventive healthcare facility network design. OR spectrum 34 (2), 349–370.