



An Accelerated Benders Decomposition Algorithm for Solving a Double-Type Double-Standard Maximal Covering Location Problem

Ashkan Fakhri^{1,2} · Antonios Fragkogios³ · Georgios K. D. Saharidis³

Received: 4 August 2020 / Accepted: 17 December 2020 / Published online: 15 March 2021
© The Author(s), under exclusive licence to Springer Nature Switzerland AG part of Springer Nature 2021

Abstract

In this paper, a double-type double-standard model (DtDsM) for maximal covering location problem is proposed which has several applications in determining the location of public emergency facilities. DtDsM includes two types of facilities: normal and backup facilities. Although backup facilities have a greater coverage distance, they offer not a full service but only a primary service. In DtDsM, each demand point must lie within the coverage distance of a backup facility if it does not lie in the coverage distance of a normal facility ensuring it to receive minimal primary services within a predetermined time. Furthermore, an accelerated Benders decomposition algorithm is proposed to solve the model. The speed and accuracy of the algorithm are compared with the commercial solver CPLEX.

Keyword Maximal covering · Public emergency services · Backup-facility location · Accelerated Benders decomposition

1 Introduction

The maximal covering location problem (MCLP) aims to locate a fixed number of facilities to cover, within a given coverage distance, as many demands as possible. MCLP is a very important model to provide social and economic services (such as the placement of health centers, fire stations, emergency warning sirens, chain stores, etc.) in order to maximize certain benefits (see, e.g., [1] and [2]).

✉ Ashkan Fakhri
fakhriashkan@gmail.com; a.fakhri@mazust.ac.ir

¹ Faculty of Science, University of Science and Technology of Mazandaran, Behshahr, Iran

² School of Mathematics, Institute for Research in Fundamental Sciences (IPM), PO Box, 19395-5746 Tehran, Iran

³ Department of Mechanical Engineering, University of Thessaly, Volos, Greece

The classical MCLP, presented in [3], seeks to maximize the number of covered demands within some desirable maximum coverage distance S . However, public services such as emergency medical services (EMS) should be available to all citizens. In fact, even if a patient is in an area where no ambulances can reach there within a reasonable time, the patient must still be responded by an existing ambulance base. Hence, the “Maximal Covering with Mandatory Closeness Constraints (MCMCC)” was proposed in [3] to provide some form of fairness for the demands not served within the desirable coverage distance S . By this model, besides the objective of maximizing the amount of demands served within the distance S , it is ensured that each demand point will find a facility within a less desirable coverage distance T ($T > S$).

Although MCLP is widely studied in the literature, there are a few works on this problem with different coverage distances while the latter is very important and more applicable in many real situations. For example, in the real world, different emergency vehicles with different coverage distances can be applied in an emergency service system. More precisely, it is obvious that in heavily congested streets of a big city such as Tehran, a motorcycle is faster (as well as cheaper) than a van, and it can traverse a longer distance than a van within a standard response time (e.g., 8 min), and hence, the type of vehicles in an emergency base specifies its coverage distance. MCMCC can be referred to as the first MCLP with different coverage distances. As an extension of the MCMCC, Gendreau et al. [4] proposed a so-called double standard model (DSM) in which, similar to (MCMCC), two coverage distances S and T ($S < T$) are considered for a single type of facility and every demand point must be covered by a facility within T while its objective is to maximize the demands that are covered at least *twice* within S . A dynamic version of DSM was developed in [5]. Some other models, such as [6] and [7], incorporated the concept of *backup coverage*. These models try (usually as a second objective) to increment the number of (say backup) facilities that cover a demand point.

Another model which considers different coverage distances is presented in [8]. However, the mentioned model does not consider the concept of backup facilities and just presents a multi-level extension of MCLP which tries to cover each demand by an as close as possible facility. In addition to the above-mentioned articles, Jabalameli et al. [9] supposed that the coverage distance of each facility can be controlled by decision-maker, and they considered the coverage distance as a function of the amount of money invested on the facility.

In the present work, we propose a variant of MCLP entitled double-type double-standard model (DtDsM) to deal with the location-allocation problem of two types of facilities: *normal* and *backup* facilities. Similar to [9], the two types of facilities (can) have different costs and different coverage distances. However, in DtDsM, it is not necessary for a facility with longer coverage distance to be more expensive.

The concept of backup facility in DtDsM is different from before. Unlike the above-mentioned models, here, we do not have only a single type of facility, and we do not aim to cover the demand points by as much as possible facilities. In fact, by backup, we do not mean multiple coverage, but the provision of primary services to the demand points within a standard time by a backup facility (if necessary).

Similar to MCMCC and consequently DSM, the proposed model DtDsM ensures that a less quality service will be available at each demand point. However, the former models translate this as availability of a facility within a less desirable distance while the latter translates it as availability of (at least) a less-quality facility (i.e., a type II or backup facility which offers only some primary services) within a standard time. In other words, each demand point must lie within the coverage distance of a backup facility if it does not lie in the coverage distance of a normal facility. This is another difference between DtDsM and the model of [9] which does not care about the too long distance between a demand point and the facility that serves it but only cares about the relation between the cost of a facility and its coverage distance.

The main objective of DtDsM is to maximize the amount of demands which can receive a full service (from a type 1 or normal facility) within the considered standard time (similar to the classical MCLP). However, as a second objective, it tries to minimize the total distance between the demand points that do not receive a full service on time and the normal facilities closest to them.

DtDsM can be useful in determining the location of facilities in public emergency services such as emergency medical services (EMSs), rescue and firefighting services (RFFSs), etc. Consider two important types of EMS vehicles: van-based ambulances as the normal facilities and motorcycle-based ambulances (motorlances) as the backup facilities. Then, if DtDsM is applied to locate EMS vehicles, it is ensured that each demand will receive at least the primary medical services by a motorlance (backup facility) within a standard waiting time (e.g., 8 min, due to its smaller size, motion abilities, road infrastructures, etc.) before a van ambulance reaches the demand and serves it full emergency services (and takes the injured to a hospital). This fast service extremely reduces the number of casualties in severe accidents before reaching the van ambulances with more complete medical equipment. It is worth knowing, as an example, that in the last few years, 200 motorlances have been purchased and added to Tehran's emergency fleet¹, and DtDsM could be a good suggestion to be applied for determining their locations (See [10], for a survey on optimization models in emergency services).

We now consider the importance of application of an efficient solution algorithm for dealing with the model. Due to the fact that in real-life applications the number of demand points and the facilities could be very large, the solution procedure of an MCLP may be a time-consuming task. On the other hand, the demand for facilities may fluctuate throughout a planning horizon such as a day, depending on the time of day.

A scenario-based model was presented in [11] to dynamically allocate the demands to the facilities in order to minimize the expected total costs under a finite set of scenarios. Also a multi-period model was proposed in [12] which considers some different time intervals (within a time horizon) in which significant changes in demand pattern occur. Their model aims to determine the minimum number of facilities and their locations for each time interval while

¹ <http://dolat.ir/detail/296614>

meeting the minimum expected coverage requirement at each time interval. Another multi-period location and relocation model was presented in [13].

However, apart from predictable changes in the parameters (e.g., the amount of demand at the points during different time intervals), there are some irregular events (such as exhibitions, tournaments, ceremonies, etc.) that may daily or even hourly affect the amount of demands and the distribution of traffic in the network and consequently change travel times and coverage regions of the facilities (i.e., I_i , c_{ij}^v , and c_{ij}^m in DtDsM are subject to temporary changes). Therefore, unpredictable need to relocation of facilities during a time horizon is inevitable.

Actually, the parameters of the facility location problems are very sensitive to the above-mentioned irregular events and can widely vary from one case to another. Hence, developing fast solution algorithms to reset and resolve large-scale facility location problems in the case of irregular changes is very important. As one can see in Sect. 4 (Numerical Results), the time required to directly solve such problems by commonly used solvers (e.g., CPLEX) seriously increases as the problem size increases. Therefore, application of more efficient algorithms such as decomposition techniques is inevitable.

Benders decomposition (BD) method is one of the efficient decomposition methods for solving some complicated mixed-integer problems with a row generation strategy and has been successfully applied for the location problem under certainty ([14–16]) and uncertainty conditions ([17, 18]). In this paper, we propose an accelerated BD algorithm to efficiently solve the DtDsM. For a thorough review of the method, the reader is referred to [19], while the most recent studies in the Benders algorithm are described in [20].

To summarize, the contribution of this paper is twofold: presenting an extension of the maximal covering location model which can handle some practical situations for which the previous models cannot be directly applied and an accelerated version of the well-known Benders decomposition algorithm for solving the proposed model.

The rest of the paper is organized as follows: mathematical formulation of DtDsM is presented in Sect. 2. In Sect. 3, an accelerated Benders decomposition algorithm is proposed to efficiently solve DtDsM. Numerical results are reported in Sect. 4 and finally, Sect. 5 briefly concludes the paper.

2 Mathematical Model

In this section, we propose a 0-1 integer linear programming problem to mathematically express DtDsM.

Parameters of the model:

I : is the set of demand points;	d_{ij} : is the travel distance from potential facility point j to demand point i ;
------------------------------------	---

J : is the set of potential facility points	p_v : is the cost of locating a normal facility
I_i : is the importance level of demand point i	p_m : is the cost of locating a backup facility
c_{ij}^v : equals 1 if the demand point i is in the coverage distance of a normal facility located at point j	c_{ij}^m : equals 1 if the demand point i is in the coverage distance of a backup facility located at point j
w : is an appropriate weight	B : is the total available budget

Variables of the model:

x_j^v : a binary variable which equals 1 iff ² a normal facility is located at potential point j

x_j^m : a binary variable which equals 1 iff a backup facility is located at potential point j
y_{ij}^v : a binary variable which equals 1 iff the demand point i is allocated to a normal facility located at point j
y_{ij}^m : a binary variable which equals 1 iff the demand point i is allocated to a backup facility located at point j

Now, using the above notation, the proposed model (DtDsM) can be presented as follows:

$$(P) : \text{Maximize } \sum_{i \in I} \sum_{j \in J} I_i c_{ij}^v y_{ij}^v - w \sum_{i \in I} \sum_{j \in J} I_i (1 - c_{ij}^v) d_{ij} y_{ij}^v \quad (1)$$

subject to

$$\sum_{j \in J} y_{ij}^v = 1 \quad i \in I \quad (2)$$

$$y_{ij}^v - x_j^v \leq 0 \quad i \in I, j \in J \quad (3)$$

$$y_{ij}^m - x_j^m \leq 0 \quad i \in I, j \in J \quad (4)$$

$$\sum_{j \in J} c_{ij}^v y_{ij}^v + \sum_{j \in J} c_{ij}^m y_{ij}^m = 1 \quad i \in I \quad (5)$$

$$\sum_{j \in J} p_v x_j^v + \sum_{j \in J} p_m x_j^m \leq B \quad (6)$$

$$y_{ij}^v, y_{ij}^m \in \{0, 1\} \quad i \in I, j \in J \quad (7)$$

$$x_j^v, x_j^m \in \{0, 1\} \quad j \in J \quad (8)$$

² If and only if

The above model is a multi-objective model which maximizes the amount of the responded demands, within a standard time, by normal facilities while simultaneously tries to minimize the total distance between the other demands and the normal facilities to which they are allocated. By appropriately choosing the weight w , the first part can be considered as the primary objective, and DtDsM becomes a preemptive priority bi-objective programming problem (a suitable value for w is proposed in the Sect. 4.1). Furthermore, the second part of (1) can be replaced by a latency cost function [21].

Constraints (2) ensure that each demand point should (anyway) be allocated to a normal facility. Constraints (3) and (4) ensure that if some demand points are allocated to a facility located at point j then the facility should be open (i.e. $x_j^v = 1$ or $x_j^m = 1$). Constraints (5) try to support, by a backup facility, the demand points which are not within the coverage distance of any of the normal facilities. Similar to [9], DtDsM considers no restriction on the total number of opened facilities. However, the available budget for opening the facilities is limited by a predefined value B and is presented by constraint (6). Finally, constraints (7) and (8) declare the type of the decision variables.

3 An Accelerated Benders Decomposition Algorithm

$$\sum_{j \in J} P_v x_j^v + \sum_{j \in J} P_m x_j^m \leq B \tag{9}$$

$$\sum_{j \in J} x_j^v \geq 1 \tag{10}$$

In the following, we adjust the problem (P) so that we can apply the Benders decomposition algorithm to solve the problem by iteratively solving finitely many easier subproblems. More precisely, considering x_j^v and x_j^m ($j \in J$) as the complicated variables, we temporarily fix them at some binary values \bar{x}_j^v and \bar{x}_j^m so that the following conditions are satisfied: and we solve the remaining subproblem. Constraint (10) is a valid inequality due to (2) and (3). Then, we check some certain termination criteria. If no criterion is satisfied, then some Benders cuts are generated to more restrict the set of feasible selections of x_j^v and x_j^m , and new feasible values for these complicated variables are selected and this process is repeated⁷

3.1 Primal Subproblem

By temporarily fixing the complicated variables x_j^v and x_j^m at \bar{x}_j^v and \bar{x}_j^m , respectively, the relevant so-called primal subproblem is then

$$\text{(PSP) : Maximize } \sum_{i \in I} \sum_{j \in J} I_i c_{ij}^v y_{ij}^v - w \sum_{i \in I} \sum_{j \in J} I_i (1 - c_{ij}^v) d_{ij} y_{ij}^v \tag{11}$$

subject to

$$\sum_{j \in J} y_{ij}^v = 1 \quad i \in I \tag{12}$$

$$y_{ij}^v \leq \bar{x}_j \quad i \in I, j \in J \tag{13}$$

$$y_{ij}^m \leq \bar{x}_j \quad i \in I, j \in J \tag{14}$$

$$\sum_{j \in J} c_{ij}^v y_{ij}^v + \sum_{j \in J} c_{ij}^m y_{ij}^m = 1 \quad i \in I \tag{15}$$

$$y_{ij}^v, y_{ij}^m \in \{0, 1\} \quad i \in I, j \in J \tag{16}$$

To apply the classical Benders decomposition method, we need the dual of the subproblems (PSP) with no duality gap. However, PSP has binary variables, i.e., the standard duality theory of linear programming could not be directly applied (for previous works on Benders decomposition with integer subproblems, refer to [22] and the related works cited therein). Furthermore, the subproblem should be solved in each iteration of Benders decomposition algorithm, and hence, its solution time is critical in the solution process of the original problem (P). To tackle these difficulties, we first note that the subproblem (PSP) is separable on i , and we can solve it easier and faster by decomposing it into $|I|$ independent subproblems which can be solved in parallel. Then, we consider the relevant subproblem to a demand point i ($i \in I$):

$$(\text{PSP}_i) : \text{Maximize } \sum_{j \in J} I_i c_{ij}^v y_{ij}^v - w \sum_{j \in J} I_i (1 - c_{ij}^v) d_{ij} y_{ij}^v \tag{17}$$

subject to

$$\sum_{j \in J} y_{ij}^v = 1 \tag{18}$$

$$y_{ij}^v \leq \bar{x}_j \quad j \in J \tag{19}$$

$$y_{ij}^m \leq \bar{x}_j \quad j \in J \tag{20}$$

$$\sum_{j \in J} c_{ij}^m y_{ij}^m + \sum_{j \in J} c_{ij}^v y_{ij}^v = 1 \tag{21}$$

$$y_{ij}^v, y_{ij}^m \in \{0, 1\} \quad j \in J \tag{22}$$

Regarding (19) and (20) and since \bar{x}_j^v and \bar{x}_j^m are binary for all $j \in J$, we can replace constraints (22) by the following constraints:

$$y_{ij}^v, y_{ij}^m \in \{0, 1\} \quad j \in J \tag{23}$$

Now, we note that the coefficient matrix of (PSP_i) has the *consecutive ones property*³ for rows, and hence, it is a totally unimodular matrix (TUM⁴). Also, one can simply see that appending an additional unit vector e_k to a TUM yields a TUM, and hence, the coefficient matrix of the standard form of (PSP_i) is also a TUM. But a (standard) linear programming problem with a totally unimodular coefficient matrix yields at least one integer optimal solution for any objective vector and any integer vector on the right-hand side of the constraints (see [23] and [24] for more details).

Therefore, we can relax the integrality restrictions (23) to obtain the following equivalent relaxation of (PSP_i) which is, in fact, a (continuous) linear programming problem:

$$(RPSP_i) : \text{Maximize} \quad (17) \tag{24}$$

subject to

$$(18) - (21) \tag{25}$$

$$y_{ij}^v, y_{ij}^m \geq 0 \quad j \in J \tag{26}$$

According to the above discussion, we actually have shown that we can study the relaxed problem $(RPSP_i)$ instead of the 0-1 integer problem (PSP_i) . In other words, we have shown that instead of the integer subproblem (PSP) , we can work with very smaller as well as easier (continuous) linear programming problems which have nice dual properties.

3.2 Dual Subproblem

We consider dual-variable λ_{i1} for the constraint (18), β_{ij} ($j \in J$) for the constraints (19), γ_{ij} ($j \in J$) for the constraints (20) and λ_{i2} for the constraint (21), and obtain the dual of $(RPSP_i)$ corresponding to the demand site i :

$$(DSP_i) : \text{Minimize} \quad \lambda_{i1} + \lambda_{i2} + \sum_{j \in J} x_j^{-v} \beta_{ij} + \sum_{j \in J} x_j^{-m} \gamma_{ij} \tag{27}$$

Subject to

$$\lambda_{i1} + c_{ij}^v \lambda_{i2} + \beta_{ij} \geq I_i c_{ij}^v - w I_i (1 - c_{ij}^v) d_{ij} \quad j \in J \tag{28}$$

$$c_{ij}^m \lambda_{i2} + \gamma_{ij} \geq 0 \quad j \in J \tag{29}$$

³ A binary matrix A has the consecutive ones property for rows if the columns of A can be permuted so that the 1s in each row appear consecutively.

⁴ A matrix A with elements in $\{0, \pm 1\}$ is TUM if determinant of every square nonsingular submatrix of A is ± 1 .

$$\beta_{ij}, \gamma_{ij} \geq 0 \quad j \in J \tag{30}$$

Then, the following problem can be considered as the dual of (PSP) when x_j^v and x_j^m are fixed at \bar{x}_j^v and \bar{x}_j^m , respectively:

$$\text{(DSP) : Minimize } \sum_{i \in I} \lambda_{i1} + \sum_{i \in I} \lambda_{i2} + \sum_{i \in I} \sum_{j \in J} \bar{x}_j^v \beta_{ij} + \sum_{i \in I} \sum_{j \in J} \bar{x}_j^m \gamma_{ij} \tag{31}$$

Subject to

$$\lambda_{i1} + c_{ij}^v \lambda_{i2} + \beta_{ij} \geq I_i c_{ij}^v - w I_i (1 - c_{ij}^v) d_{ij} \quad i \in I, j \in J \tag{32}$$

$$c_{ij}^m \lambda_{i2} + \gamma_{ij} \geq 0 \quad i \in I, j \in J \tag{33}$$

$$\beta_{ij}, \gamma_{ij} \geq 0 \quad i \in I, j \in J \tag{34}$$

3.3 Master Problem and the Benders Iterations

Let S_i^P and S_i^D be the set of all extreme points and the set of all extreme directions of the feasible region of (DSP_i), respectively, while S^P and S^D are the set of all extreme points and the set of all extreme directions of the feasible region of (DSP), respectively. Then, the original problem (P) introduced by (1)–(8), can be represented as the following so-called Master Problem:

$$\text{(MP) Maximize } \theta \tag{35}$$

Subject to

$$\sum_{j \in J} p_v x_j^v + \sum_{j \in J} p_m x_j^m \leq B \tag{36}$$

$$\sum_{j \in J} x_j^v \geq 1 \tag{37}$$

$$\sum_{i \in I} \sum_{j \in J} \bar{\beta}_{ij} x_j^v + \sum_{i \in I} \sum_{j \in J} \bar{\gamma}_{ij} x_j^m \geq -\sum_{i \in I} (\bar{\lambda}_{i1} + \bar{\lambda}_{i2}), \left(\bar{\lambda}_1, \bar{\lambda}_2, \bar{\beta}, \bar{\gamma} \right)^T \in S^D \tag{38}$$

$$\sum_{i \in I} \sum_{j \in J} \bar{\beta}_{ij} x_j^v + \sum_{i \in I} \sum_{j \in J} \bar{\gamma}_{ij} x_j^m - \theta \geq -\sum_{i \in I} (\bar{\lambda}_{i1} + \bar{\lambda}_{i2}), \left(\bar{\lambda}_1, \bar{\lambda}_2, \bar{\beta}, \bar{\gamma} \right)^T \in S^P \tag{39}$$

$$x_j^v, x_j^m \in \{0, 1\} \quad j \in J \tag{40}$$

where

$$\beta_i = (\beta_{i1}, \dots, \beta_{i|J|}), \quad \beta = (\beta_1, \dots, \beta_{|I|})$$

$$\gamma_i = (\gamma_{i1}, \dots, \gamma_{i|J|}), \gamma = (\gamma_1, \dots, \gamma_{|I|})$$

$$\lambda_1 = (\lambda_{11}, \dots, \lambda_{|I|1}), \lambda_2 = (\lambda_{12}, \dots, \lambda_{|I|2})$$

The master problem involves only the complicated variables x_j^v and x_j^m and an extra auxiliary variable θ . Constraints (38) and (39) are referred to as Benders *feasibility cuts* and *optimality cuts*, respectively. Feasibility cuts ensure that the selected values for x_j^v and x_j^m do not result in an unbounded dual subproblem (DSP) and consequently an infeasible primal subproblem (PSP). On the other hand, optimality cuts ensure that θ takes the minimum value of (DSP) or equivalently the maximum value of (PSP) for each feasible choice of x_j^v and x_j^m .

However, all of the feasibility cuts and optimality cuts are not known beforehand, and they are iteratively generated and appended to the relaxed master problem (RMP) when necessary. (RMP) is supposed to be a relaxed version of the master problem which is defined by (35)–(37), (40) and some (but not all) of the feasibility cuts (38) and the optimality cuts (39). Hence, (RMP) provides an upper bound for the optimal value of the original problem. On the other hand, the optimal value of the primal subproblem (PSP) or equivalently the optimal value of the dual subproblem (DSP) for each feasible values of x_j^v and x_j^m provides a lower bound for the optimal value of the original problem. The Benders iterations are continued until the difference between the best upper bound and the best lower bound is less than a predefined tolerance ϵ .

Note that a vector $(\lambda_1, \lambda_2, \beta, \gamma)^T$ is an extreme feasible point of (DSP) if and only if $(\lambda_{i1}, \lambda_{i2}, \beta_i, \gamma_i)^T$ is an extreme feasible point of (DSP_i) for all $i \in I$. Furthermore, a nonzero vector $(\lambda_1, \lambda_2, \beta, \gamma)^T$ is an extreme recession direction of (DSP) if and only if $(\lambda_{i'1}, \lambda_{i'2}, \beta_{i'}, \gamma_{i'})^T$ is an extreme recession direction of $(DSP_{i'})$ for some $i' \in I$ and

$$(\lambda_{i1}, \lambda_{i2}, \beta_i, \gamma_i)^T = 0, \forall i \neq i'.$$

Therefore, if for some fixed values \bar{x}_j^v and \bar{x}_j^m there exists some $I' \subseteq I$ so that subproblems $(PSP_{i'})$ are infeasible (and consequently, $(DSP_{i'})$ are unbounded) for each $i \in I'$, then we can append to the master problem any arbitrary number of the following feasibility cuts:

$$\bar{\lambda}_{i1} + \bar{\lambda}_{i2} + \sum_{j \in J} \bar{\beta}_{ij} x_j^v + \sum_{j \in J} \bar{\gamma}_{ij} x_j^m \geq 0 \quad i \in I' \tag{41}$$

where $(\bar{\lambda}_{i1}, \bar{\lambda}_{i2}, \bar{\beta}_i, \bar{\gamma}_i)^T$ is an extreme recession direction of $(DSP_{i'})$ for each $i \in I'$. The set of directions of $(DSP_{i'})$ can be precisely characterized by the set of all nonzero vectors $(\lambda_{i1}, \lambda_{i2}, \beta_i, \gamma_i)^T$ satisfying the following constraints (see Sect. 2.4 of [25]):

$$\lambda_{i1} + c_{ij}^v \lambda_{i2} + \beta_{ij} \geq 0 \quad j \in J \tag{42}$$

$$c_{ij}^m \lambda_{i2} + \gamma_{ij} \geq 0 \quad j \in J \tag{43}$$

$$\beta_{ij}, \gamma_{ij} \geq 0 \quad j \in J. \tag{44}$$

3.4 Solution of (PSP_i) and (DSP_i)

In this subsection, we show that the primal subproblems (RPSP_i) and dual subproblems (DSP_i) are quite easy to handle for each $i \in I$. In fact, one of the following cases may occur for this problem:

Case 1 There exists some $\bar{j} \in J$ such that $x_{\bar{j}}^v = c_{\bar{j}i}^v = 1$.

In this case, the demand point i can be optimally allocated to the normal facility located at \bar{j} since i is within its coverage distance. The optimal value is I_i and

$$\begin{cases} \lambda_{i1} = 0, & \lambda_{i2} = I_i, \\ \beta_{ij} = 0 \text{ and } \gamma_{ij} = 0 & \forall j \in J \end{cases} \tag{45}$$

presents an optimal solution of the dual subproblem (DSP_i) in this case. But, if \bar{j} is the unique index with the property that $x_{\bar{j}}^v = c_{\bar{j}i}^v = 1$, then another (alternative) optimal solution of the dual subproblem (DSP_i) can be obtained as follows:

$$\begin{cases} \lambda_{i1} = 0, & \lambda_{i2} = 0, \\ \gamma_{ij} = 0 & \forall j \in J, \\ \beta_{ij} = I_i & \forall j \in J_i^v \text{ and } \beta_{ij} = 0 \quad \forall j \in J \setminus J_i^v \end{cases} \tag{46}$$

where, $J_i^v := \{j \in J : c_{ij}^v = 1\}$.

Proposition 1 If \bar{j} is the unique index with the property that $x_{\bar{j}}^v = c_{\bar{j}i}^v = 1$, then the optimal solutions defined by (46) is an extreme optimal solution of (DSP_i).

Proof By the well-known Shapiro’s variable transformation, the unrestricted (free) variables λ_{i1} and λ_{i2} can be replaced by $\lambda_{i1}^+ - \lambda_{i1}^-$ and $\lambda_{i2}^+ - \lambda_{i2}^-$, respectively, where $\lambda_{i1}^+, \lambda_{i1}^-, \lambda_{i2}^+$, and λ_{i2}^- are non-negative variables. Then, (DSP_i) is a linear programming (LP) problem with $2|J| + 4$ non-negative variables. In fact, $\lambda_{i1} = 0$ is equivalent to $\lambda_{i1}^+ = \lambda_{i1}^- = 0$, and $\lambda_{i2} = 0$ is equivalent to $\lambda_{i2}^+ = \lambda_{i2}^- = 0$. Now, we see that the following constraints provide $2|J| + 4$ linearly independent constraints of (DSP_i) binding at (46):

$$\lambda_{i1}^+, \lambda_{i1}^-, \lambda_{i2}^+, \lambda_{i2}^- \geq 0,$$

$$\gamma_{ij} \geq 0 \quad j \in J,$$

$$\beta_{ij} \geq 0 \quad j \in J \setminus J_i^v,$$

$$\left(\lambda_{i1} + c_{ij}^v \lambda_{i2} + \beta_{ij} \geq I_i c_{ij}^v - w I_i (1 - c_{ij}^v) d_{ij} \quad j \in J, \right) \text{ for } j \in J_i^v$$

This completes the proof according to the simple characterization theorem for extreme points of polyhedral sets.

Remark In our solution algorithm, we prefer (due to preliminary numerical results) to use (46) as an optimal solution of (DSP_i) if possible. However, we use (45) in the case that (46) cannot be used (i.e. whenever j is not the unique index with the property that $x_{ij}^v = c_{ij}^v = 1$). By the following proposition, we show that (45) is also an extreme point.

Proposition 2 The optimal solutions defined by (45) is an extreme optimal solution of (DSP_i) .

Proof Considering the same notation of the proof of Proposition 1, $\lambda_{i1} = 0$ is equivalent to $\lambda_{i1}^+ = \lambda_{i1}^- = 0$ and $\lambda_{i2} = I_i$ is equivalent to $\lambda_{i2}^+ = I_i$ and $\lambda_{i2}^- = 0$. Now, we see that the following constraints provide $2|J| + 4$ linearly independent constraints of (DSP_i) binding at (45):

$$\lambda_{i1}^+, \lambda_{i1}^-, \lambda_{i2}^- \geq 0,$$

$$\gamma_{ij} \geq 0 \quad j \in J,$$

$$\beta_{ij} \geq 0 \quad j \in J,$$

$$\left(\lambda_{i1} + c_{ij}^v \lambda_{i2} + \beta_{ij} \geq I_i c_{ij}^v - w I_i (1 - c_{ij}^v) d_{ij} \quad j \in J, \right) \text{ for } j = \bar{j}.$$

This completes the proof according to the simple characterization theorem for extreme points of polyhedral sets.

Case 2 For each $j \in J$, $c_{ij}^v = 1$ implies $x_j^v = 0$.

In this case, the demand point i is not within the coverage distance of any of normal facilities. This case can itself be divided into the following two subcases:

Subcase 2.1 For each $j \in J$, $c_{ij}^v = 1$ implies $x_j^v = 0$ and $c_{ij}^m = 1$ implies $x_j^m = 0$.

This case occurs when not only the demand point i is not within the coverage distance of any of normal facilities, but also, it is not within the coverage distance of any of backup facilities. Hence, in this case, $(RPSP_i)$ is infeasible. The infeasibility of $(RPSP_i)$ is obvious due to (19)–(21). On the other hand, the dual subproblem (DSP_i) is unbounded in this case and clearly

$$\begin{cases} \lambda_{i1} = 0, & \lambda_{i2} = -1, \\ \beta_{ij} = 1 & \forall j \in J_i^v, \quad \beta_{ij} = 0 \quad \forall j \in J \setminus J_i^v, \\ \gamma_{ij} = 1 & \forall j \in J_i^m, \quad \gamma_{ij} = 0 \quad \forall j \in J \setminus J_i^m \end{cases} \quad (47)$$

is a recession direction for (DSP_i) , where:

$$J_i^v = \left\{ j \in J : c_{ij}^v = 1 \right\} \text{ and } J_i^m = \left\{ j \in J : c_{ij}^m = 1 \right\} \quad (48)$$

Proposition 3 The direction defined by (47) is an extreme recession direction for (DSP_i).

Proof It is sufficient to show that if there exist two directions $(\lambda'_{i1}, \lambda'_{i2}, \beta'_i, \gamma'_i)^T$ and $(\lambda''_{i1}, \lambda''_{i2}, \beta''_i, \gamma''_i)^T$ of the feasible set of (DSP_i) such that

$$(\lambda_{i1}, \lambda_{i2}, \beta_i, \gamma_i)^T = (\lambda'_{i1}, \lambda'_{i2}, \beta'_i, \gamma'_i)^T + (\lambda''_{i1}, \lambda''_{i2}, \beta''_i, \gamma''_i)^T,$$

then these two directions are both positive multiples of $(\lambda_{i1}, \lambda_{i2}, \beta_i, \gamma_i)^T$. First, note that for each $j \in J$, if $c^m_{ij} = 0$, then $\gamma_{ij} = 0$ and since $\gamma'_{ij}, \gamma''_{ij} \geq 0$, we have $\gamma'_{ij} = \gamma''_{ij} = 0$. Also, if $c^m_{ij} = 1$ then $\gamma_{ij} = 1$ and according to (43), we have:

$$\begin{aligned} \begin{cases} \lambda'_{i2} + \gamma'_{ij} \geq 0 \Rightarrow \lambda'_{i2} \geq -\gamma'_{ij} \\ \lambda''_{i2} + \gamma''_{ij} \geq 0 \Rightarrow \lambda''_{i2} \geq -\gamma''_{ij} \end{cases} \begin{matrix} -1 = \lambda_{i2} = \lambda'_{i2} + \lambda''_{i2} \geq -(\gamma'_{ij} + \gamma''_{ij}) = -\gamma_{ij} = -1 \\ \Rightarrow \end{matrix} \begin{cases} \lambda'_{i2} = -\gamma'_{ij} \\ \lambda''_{i2} = -\gamma''_{ij} \end{cases} \\ \lambda_{i2} = -1 \& \gamma_{ij} = 1 \Rightarrow \frac{\lambda'_{i2}}{\lambda_{i2}} = \frac{\gamma'_{ij}}{\gamma_{ij}}, \frac{\lambda''_{i2}}{\lambda_{i2}} = \frac{\gamma''_{ij}}{\gamma_{ij}}. \end{aligned} \tag{49}$$

On the other hand, we know that $J^v_1 \neq J$ because otherwise, according to the assumptions of Subcase 2.1, we have $x^v_j = 0$ for all $j \in J$ which contradicts the valid inequality (37). Therefore, there exists some $j \in J \setminus J^v_1$ where $\beta^-_{ij} = 0$. Hence, we have:

$$\begin{aligned} \beta^-_{ij} = 0 \quad \begin{matrix} \beta^-_{ij} = \beta'_{ij} + \beta''_{ij} \& \beta'_{ij}, \beta''_{ij} \geq 0 \\ \Rightarrow \end{matrix} \quad \beta'_{ij} = \beta''_{ij} = 0 \quad \begin{matrix} c^v_{ij} = 0 \& (\lambda_{i1} + c^v_{ij} \lambda_{i2} + \beta_{ij} \geq 0 \& j \in J) \\ \Rightarrow \end{matrix} \quad \lambda'_{i1}, \lambda''_{i1} \geq 0 \\ \begin{matrix} 0 = \lambda_{i1} = \lambda'_{i1} + \lambda''_{i1} \\ \Rightarrow \end{matrix} \quad \lambda'_{i1} = \lambda''_{i1} = 0. \end{aligned} \tag{50}$$

Now, for each $j \in J$, there exist two cases: $c^v_{ij} = 0$ or $c^v_{ij} = 1$. If $c^v_{ij} = 0$, then $\beta_{ij} = 0$ and since $\beta_{ij} = \beta'_{ij} + \beta''_{ij}$ and $\beta'_{ij}, \beta''_{ij} \geq 0$, we have $\beta'_{ij} = \beta''_{ij} = 0$. Furthermore, if $c^v_{ij} = 1$ then:

$$\begin{aligned} \beta_{ij} = 1 \quad \begin{matrix} (42)\& (50) \\ \Rightarrow \end{matrix} \begin{cases} \lambda'_{i2} \geq -\beta'_{ij} \\ \lambda''_{i2} \geq -\beta''_{ij} \end{cases} \begin{matrix} -1 = \lambda_{i2} = \lambda'_{i2} + \lambda''_{i2} \geq -(\beta'_{ij} + \beta''_{ij}) = -\beta_{ij} = -1 \\ \Rightarrow \end{matrix} \begin{cases} \lambda'_{i2} = -\beta'_{ij} \\ \lambda''_{i2} = -\beta''_{ij} \end{cases} \tag{51} \\ \lambda_{i2} = -1 \& \beta_{ij} = 1 \Rightarrow \frac{\lambda'_{i2}}{\lambda_{i2}} = \frac{\beta'_{ij}}{\beta_{ij}}, \frac{\lambda''_{i2}}{\lambda_{i2}} = \frac{\beta''_{ij}}{\beta_{ij}}. \end{aligned}$$

To summarize, we have shown that $(\lambda'_{i1}, \lambda'_{i2}, \beta'_i, \gamma'_i)^T$ and $(\lambda''_{i1}, \lambda''_{i2}, \beta''_i, \gamma''_i)^T$ are positive multiples of $(\lambda_{i1}, \lambda_{i2}, \beta_i, \gamma_i)^T$ and actually, they are not distinct from $(\lambda_{i1}, \lambda_{i2}, \beta_i, \gamma_i)^T$, and this completes the proof.

Subcase 2.2 For each $j \in J$, $c^v_{ij} = 1$ implies $x^v_j = 0$ but there exists some $j \in J$ such that $x^m_j = c^m_{ij} = 1$

In this case, the demand point i is optimally allocated to a normal facility located at point $j \in \arg \min_{\{j \in J: x_j^v=1\}} d_{ij}$ and also to an arbitrary backup facility located at an arbitrary point j with $c_{ij}^m = 1$. The optimal value of (PSP_i) in this case will be $-wI_i d_{ij}^-$. On the other hand,

$$\begin{cases} \lambda_{i1} = -wI_i d_{ij}^-, & \lambda_{i2} = 0, & \gamma_{ij} = 0 & \forall j \in J, \\ \beta_{ij} = I_i + wI_i d_{ij}^- & \forall j \in J_i^v, & \beta_{ij} = \max(0, wI_i d_{ij}^- - wI_i d_{ij}) & \forall j \in J \setminus J_i^v \end{cases} \quad (52)$$

is an optimal solution of the dual subproblem (DSP_i) .

Proposition 4 *The optimal solution defined by (52) is an extreme optimal solution of (DSP_i) .*

Proof Consider the constraints of the problem (DSP_i) . By the same notation of the proof of Proposition 1, $\lambda_{i1} = -wI_i d_{ij}^-$ is equivalent to $\lambda_{i1}^+ = 0$ and $\lambda_{i1}^- = wI_i d_{ij}^-$, and $\lambda_{i2} = 0$ is equivalent to $\lambda_{i2}^+ = \lambda_{i2}^- = 0$. Hence, $\lambda_{i1}^+, \lambda_{i2}^+, \lambda_{i2}^- \geq 0$ are three constraints binding at (52). These binding constraints and the constraints $\gamma_{ij} \geq 0$ for all $j \in J$ provide $|J| + 3$ constraints binding at (52). On the other hand, for each $j \in J$, either we have a constraint $\beta_{ij} \geq 0$ or a constraint in the bundle (28) which together give $|J|$ more constraints binding at (52). Furthermore, the constraint $\beta_{ij} \geq 0$ and the constraint (28) are both binding at (52) for $j = j$ (where $j \in \arg \min_{\{j \in J: x_j^v=1\}} d_{ij}$), giving one more binding constraint.

All the above-mentioned constraints together provide $2|J| + 4$ linearly independent constraints of (DSP_i) which are binding at (52). This completes the proof according to the simple characterization theorem for extreme points of polyhedral sets.

3.5 Valid Inequalities for the Master Problem

According to the constraints (5), the following inequalities are valid for the original problem (P), and we can add them to the master problem:

$$\sum_{j \in J_i^v} x_j^v + \sum_{j \in J_i^m} x_j^m \geq 1 \quad i \in I \quad (53)$$

where J_i^v and J_i^m are defined by (48). These inequalities prevent infeasibility in the Benders' primal subproblems and equivalently prevent unboundedness in the Benders' dual subproblems. Thus, feasibility cuts are no longer generated.

3.6 Steps of the Algorithm and its Finite Convergence

After the analysis of the different cases of the subproblems (PSP_i) and (DSP_i) , the authors propose the algorithm Alg.1 for solving the original problem (P).

Finite convergence of the proposed algorithm (Alg. 1) can be deduced in a couple of ways. First, as one can easily see, at each iteration of Alg. 1, an optimality Benders cut of type (39) is generated and appended to the relaxed master problem (RMP). Since S^P has a finite number of elements, Alg. 1 terminates in a finite number of iterations for any given ϵ . Furthermore, at each iteration of Alg. 1, at least one master choice is excluded from the

set of feasible master choices which has only a finite number of elements. Hence, finite convergence of Alg. 1 can alternatively be concluded.

Alg. 1. The Proposed Benders Decomposition Algorithm (PBDA) for solving Problem (P)

Find an initial feasible vectors \bar{x}^v, \bar{x}^m and $\bar{\theta}$ for the relaxed master problem defined by (35)-(37), (40) and (53), initialize ϵ and set $Z_{lb} = -\infty$ and $Z_{ub} = \bar{\theta}$.

while $Z_{ub} - Z_{lb} > \epsilon$ **do**

for each $i \in I$

if $\{j \in J : x_j^v = c_{ij}^v = 1\} = \emptyset$

 select $\bar{j} \in \arg \min_{\{j \in J : x_j^v = 1\}} d_{ij},$

$\bar{\lambda}_{i1} \leftarrow -wI_i d_{i\bar{j}},$

$\bar{\lambda}_{i2} \leftarrow 0,$

$\bar{\gamma}_{ij} \leftarrow 0 \quad \forall j \in J,$

$\bar{\beta}_{ij} \leftarrow I_i + wI_i d_{i\bar{j}} \quad \forall j \in J_i^v,$

$\bar{\beta}_{ij} \leftarrow \max(0, wI_i d_{i\bar{j}} - wI_i d_{ij}) \quad \forall j \in J \setminus J_i^v$

else if $|\{j \in J : x_j^v = c_{ij}^v = 1\}| = 1$ % ($|\cdot|$ stands for the cardinal number of a set)

$\bar{\lambda}_{i1} \leftarrow 0,$

$\bar{\lambda}_{i2} \leftarrow 0,$

$\bar{\gamma}_{ij} \leftarrow 0 \quad \forall j \in J,$

$\bar{\beta}_{ij} \leftarrow I_i \quad \forall j \in J_i^v,$

$\bar{\beta}_{ij} \leftarrow 0 \quad \forall j \in J \setminus J_i^v$

else

$\bar{\lambda}_{i1} \leftarrow 0,$

$\bar{\lambda}_{i2} \leftarrow I_i,$

$\bar{\beta}_{ij} \leftarrow 0 \quad \forall j \in J,$

$\bar{\gamma}_{ij} \leftarrow 0 \quad \forall j \in J$

end if

end for

$Z_{lb} \leftarrow \max\{Z_{lb}, \sum_{i \in I} \bar{\lambda}_{i1} + \sum_{i \in I} \bar{\lambda}_{i2} + \sum_{i \in I} \sum_{j \in J} \bar{x}_j^v \bar{\beta}_{ij} + \sum_{i \in I} \sum_{j \in J} \bar{x}_j^m \bar{\gamma}_{ij}\}.$

 Append the following optimality cut to the relaxed master problem:

$$\sum_{i \in I} \sum_{j \in J} \bar{\beta}_{ij} x_j^v + \sum_{i \in I} \sum_{j \in J} \bar{\gamma}_{ij} x_j^m - \theta \geq -\sum_{i \in I} (\bar{\lambda}_{i1} + \bar{\lambda}_{i2}).$$

 Find a new optimal solution $(\bar{x}^v, \bar{x}^m, \bar{\theta})$ of the updated relaxed master problem.

$Z_{ub} \leftarrow \min\{Z_{ub}, \bar{\theta}\}.$

end while

4 Numerical Results

In order to test the efficiency of the proposed Benders decomposition algorithm (PBDA), the authors compared it with both the classical Benders decomposition algorithm (CBDA) and with the state-of-the-art commercial solver CPLEX on a dataset for the model (P) presented in Sect. 3.

4.1 Dataset

In order to better evaluate their whole approach, the authors generated a suitable dataset for the comparison of the three algorithms. This dataset consists of 5 major groups of nodes, each one including different number of nodes (100, 225, 400, 625, and 900 nodes) to simulate the morphology of imaginary cities. For each group, the morphology is similar to the one depicted in Fig. 1. Each square corresponds to a potential point for locating the ambulances (vans and motorlances), and the distance d_{ij} between 2 squares is given by the following Eq. (54), also known as ‘‘Manhattan distance’’ [26]:

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \tag{54}$$

In every group of nodes, 10 different examples are created, each one with different values for parameters I_i and w . For each of the 10 examples, the parameter I_i is randomly generated and takes integer values in the interval $[1, \sqrt{NoOfNodes}]$, and the weight is considered as:

$$w = \frac{1}{\sum_{i \in I} [I_i * \max_{j \in J} \{ (1 - c_{ij}^v) d_{ij} \}]} \tag{55}$$

By using Eq. (55), which contains the maximum values of the distance d_{ij} in the denominator, the weight w takes very small values so that the first term of the objective function (1) is considered of the first priority and the second term (multiplied by w) is considered of the second priority, as stated in Sect. 2

Furthermore, in all groups of examples, the available budget (B), the cost of locating a backup facility (p_m), and the motorcycle coverage distance (MCD) are shown in Table 1.

The authors created the dataset in order to aim at two goals. First, it offers the opportunity to estimate the efficiency of the proposed algorithm in a wide range

Fig. 1 Example city of 100 nodes

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	11	12	13	14	15	16	17	18	19	20
3	21	22	23	24	25	26	27	28	29	30
4	31	32	33	34	35	36	37	38	39	40
5	41	42	43	44	45	46	47	48	49	50
6	51	52	53	54	55	56	57	58	59	60
7	61	62	63	64	65	66	67	68	69	70
8	71	72	73	74	75	76	77	78	79	80
9	81	82	83	84	85	86	87	88	89	90
10	91	92	93	94	95	96	97	98	99	100

Table 1: Data in all examples and cases

	B	p_m	MCD
Value	$\sqrt{NoOfNodes}$	1	5

of the problem sizes. This is the reason why the authors have chosen to solve from a small problem of 100 nodes to a relative large problem of 900 nodes.

The second is to consider some sensitivity analysis on certain parameters to see how these would change the solution. This sensitivity analysis would validate the described model. For this reason, the authors decided to differentiate the ratio between the cost of locating a backup facility (motorlance) and the cost of locating a normal facility (van). Thus, having the backup facility cost fixed for all examples, they considered three cases where the normal facility cost takes different values. Moreover, having a fixed available budget for all examples, the different normal facility costs led to the need to consider proportionally different van coverage distance (VCD) in order to avoid infeasibility of the problem. It makes sense that a more expensive normal facility could cover a larger region. Table 2 depicts the values for these parameters in each of the three cases.

Obviously, the coverage distance of a motorlance is generally larger than that of the van due to the former’s faster arrival to the site of emergency. In the notation of the model, the parameter VCD is translated to c_{ij}^v by $c_{ij}^v = 1$ if $d_{ij} \leq VCD$ and $c_{ij}^v = 0$ otherwise. Parameter MCD is translated to c_{ij}^m , by $c_{ij}^m = 1$ if $d_{ij} \leq MCD$ and $c_{ij}^m = 0$ otherwise. Finally, the cost of a motorlance is lower than a van in all cases.

The above leads to a total number of 150 examples solved (5 groups of nodes \times 3 cases \times 10 examples). The algorithms were implemented using C++ and solved using CPLEX 12.10 in Visual Studio 2019 on a desktop computer with Intel (R) Core(TM) i7-4790 CPU 3.6 GHz, 16 GB RAM, 64-bit.

4.2 Results

The study of the computational results could be divided into two subsections. The first one is the comparison of the three algorithms considering the different problem sizes. The second one would be to evaluate the previously mentioned sensitivity analysis on the three cases of examples.

Table 2 Applied parameters for the different cases

	Case 1	Case 2	Case 3
p_v	2	3	4
VCD	2	3	4

4.2.1 Comparison of the Algorithms

In Table 3, the comparative results for the examples in the groups of 100, 225, 400, 625, and 900 nodes are shown respectively for CPLEX, classical Benders decomposition algorithm (CBDA), and the proposed Benders decomposition algorithm (PBDA). It has to be mentioned that the maximum CPU time for all instances was set at 2 h (7200 s).

Apart from the first two columns, the first three columns show the number of examples solved to optimality in each case by the algorithms. The next three columns show the average CPU time spent to solve these examples to optimality. Following are the average optimality gaps of all examples (either optimal or feasible or infeasible ones) for each case. Note that if an algorithm was unable to find any feasible solution within 2 h, then a gap of 100% was considered for it. Hence, the average gap of 100% for 900 nodes means that CPLEX failed in finding even one feasible solution in all of 10 examples. Finally, the last three columns show the percentage of reduction of the CPU time that an algorithm achieves over the other. For example, in the final column “PBDA vs CBDA”, the negative percentages show how much faster the proposed algorithm is than the classical algorithm. In order to compare the CPU times of two algorithms such as CBDA and CPLEX, the following formula was used:

$$\left(\frac{\left(\frac{10}{n(CBDA)}\right) \times \text{average cpu time of CBDA for examples solved to optimality}}{\left(\frac{10}{n(CPLEX)}\right) \times \text{average cpu time of CPLEX for examples solved to optimality}} - 1 \right) \times 100\% \tag{56}$$

where

$n(CBDA)$ = number of examples solved to optimality by CBDA

Table 3 Comparative results for the dataset (average for 10 examples in each case)

Nodes	Case	Number of examples solved optimally			Average total CPU time (sec) for examples solved optimally			Average % GAP for all examples			Comparison of Average total CPU time		
		CPLEX	CBDA	PBDA	CPLEX	CBDA	PBDA	CPLEX	CBDA	PBDA	CBDA vs CPLEX	PBDA vs CPLEX	PBDA vs CBDA
100	1	10	10	10	71.0	3.9	2.7	0.0%	0.0%	0.0%	-94%	-96%	-32%
100	2	10	10	10	79.7	5.4	4.4	0.0%	0.0%	0.0%	-93%	-94%	-18%
100	3	10	10	10	4.2	1.7	1.2	0.0%	0.0%	0.0%	-60%	-72%	-30%
225	1	10	10	10	60.4	8.4	2.7	0.0%	0.0%	0.0%	-86%	-96%	-68%
225	2	10	10	10	1614.3	16.5	11.3	0.0%	0.0%	0.0%	-99%	-99%	-31%
225	3	4	10	10	4633.3	29.3	27.7	24.7%	0.0%	0.0%	-100%	-100%	-5%
400	1	0	10	10	-	333.4	261.8	5.9%	0.0%	0.0%	-	-	-21%
400	2	5	10	10	2273.6	60.5	36.4	10.3%	0.0%	0.0%	-99%	-99%	-40%
400	3	10	10	10	1914.7	23.3	6.8	0.0%	0.0%	0.0%	-99%	-100%	-71%
625	1	0	10	10	-	2056.5	1812.9	15.5%	0.0%	0.0%	-	-	-12%
625	2	0	10	10	-	476.1	410.8	19.1%	0.0%	0.0%	-	-	-14%
625	3	0	10	10	-	186.7	112.8	6.8%	0.0%	0.0%	-	-	-40%
900	1	0	4	6	-	5416.1	4659.8	100.0%	12.7%	6.8%	-	-	-43%
900	2	0	10	10	-	3123.3	2692.7	100.0%	0.0%	0.0%	-	-	-14%
900	3	0	10	10	-	443.1	327.8	100.0%	0.0%	0.0%	-	-	-26%

*CPLEX, classical Benders decomposition algorithm (CBDA), proposed Benders decomposition algorithm (PBDA)

$$n(\text{CPLEX}) = \text{number of examples solved to optimality by CPLEX}$$

Similar formulas were used for comparing PBDA vs CPLEX and PBDA vs CBDA.

When comparing both CBDA and PBDA with CPLEX, one can notice that the Benders-based algorithms are much more efficient in all examples. Especially, for small to medium instances (100 to 625 nodes), they can solve them all to optimality, while being on average 94% faster than CPLEX. Note that CPLEX results in suboptimal solutions for 225 to 625 nodes in the maximum time of 2 h. As can be seen, in the cases with 900 nodes, CPLEX could not find even one feasible solution in all of 10 examples. This is the reason that its average gap could not be compared with the Benders-based ones.

Moreover, the comparison between the proposed algorithm and the classical one is quite interesting, as it is obvious that PBDA outperforms CBDA. Concerning the average optimality gap, the two algorithms manage to find optimal solutions for almost all instances, while the number of their iterations is exactly the same. The latter fact confirms that the proposed solution method of Sect. 3.4 leads to exactly the same optimality cuts as the classical Benders decomposition algorithm. Focusing on the CPU time, the proposed algorithm is on average 30% faster than the classical one. The good performance of PBDA lies on the fact that there is no need to model and solve the dual subproblem, which is quite computationally expensive. Instead, its solution can be easily computed as presented in Sect. 3.4, which is computationally very fast, as it can be easily coded using “for” and “if” loops. The computational benefit of PBDA seems to be reduced for larger instances such as 900 nodes. However, this happens due to the 2-h maximum limit, to which both algorithms reach for some examples, while PBDA still finds better solutions with lower optimality gaps than CBDA.

4.2.2 Sensitivity Analysis

As previously stated, the authors examined three cases where the cost of locating a normal facility and its coverage distance take different values. For every case, 10 examples were solved, and in Table 4, the average computational times for

Table 4 Average comparative results (for each case in all node groups)

Case	Average Optimal Objective Function	Average % GAP			Average total CPU time (sec) for examples solved optimally			Comparison of Average total CPU time		
		CPLEX	CBDA	PBDA	CPLEX	CBDA	PBDA	CBDA vs CPLEX	PBDA vs CPLEX	PBDA vs CBDA
1	857.1	24.3%	2.6%	1.4%	-	1563.7	1348.0	-	-	-35%
2	1077.4	25.9%	0.0%	0.0%	-	736.3	631.1	-	-	-23%
3	1303.0	26.3%	0.0%	0.0%	-	136.8	95.2	-	-	-34%

Table 5 Average number of vans and motorcycles used in model's solution (average for 10 examples in each case)

Nodes	Case	Total Vans Used	Total Motorcycles Used
100	1	4.0	2.0
100	2	2.0	3.2
100	3	2.0	2.0
225	1	5.0	5.0
225	2	3.0	6.0
225	3	2.0	6.4
400	1	5.3	9.4
400	2	4.0	8.0
400	3	3.0	8.0
625	1	6.0	13.0
625	2	4.0	13.0
625	3	3.0	13.0
900	1	5.3	19.4
900	2	4.0	18.0
900	3	3.0	18.0

these examples in all node groups are depicted. As it is shown, case 1 needs more time to be solved, which can be explained by the fact that the coverage distance of the normal facilities is very small, and thus, it is harder to find a solution that satisfies the master's constraints. To the same direction, case 3 is the easiest one due to the large coverage distance of the normal facility.

Table 5 indicates the average number of vans and motorcycles deployed. As one can see, within the same group of nodes (e.g., 225 nodes), the average number of vans decreases as its cost increases. The coverage gap created due to the use of a small number of expensive vans is covered by increasing the number of motorcycles. This outcome was expected, and its confirmation proves the validity of the model and the proposed solution approach.

However, in the larger examples (e.g., 625 and 900 nodes), one can notice a slight difference in the above conclusion. While the average number of vans decreases as its cost is getting higher, the average number of motorcycles remains the same (large enough) for all three cases. This happens because we have considered a conservative policy for the available budget (Table 1). The relatively small budget for large instances leads to shortage of vans even for

34	34	34	34	34	34	34	27	27	27	27	27	27	27	27
34	34	34	34	34	34	27	27	27	55	27	27	27	27	27
34	34	34	34	34	34	34	27	55	55	55	27	27	27	27
34	34	34	34	34	34	34	55	55	55	55	55	27	27	27
34	92	34	34	34	34	34	98	55	55	55	27	27	27	134
92	92	92	34	34	34	98	98	98	55	27	27	27	134	134
92	92	92	92	34	98	98	98	98	98	134	27	134	134	134
92	92	92	34	154	154	98	98	98	134	146	134	134	134	134
154	92	154	154	154	154	154	98	134	146	146	146	134	134	134
154	154	154	154	154	154	154	154	146	146	146	146	146	134	134
154	154	154	154	154	154	154	154	154	146	146	146	134	134	134
154	197	154	154	154	154	154	154	221	221	146	134	134	134	134
197	197	197	154	154	154	154	221	221	221	221	221	221	134	134
197	197	197	197	154	154	221	221	221	221	221	221	221	221	221
197	197	197	154	154	221	221	221	221	221	221	221	221	221	221

Fig. 2 Solution (225 nodes, case 1, example 1). 5 vans and 5 motorcycles

case 1 (where they are cheaper), thus making it necessary to deploy a large number of motorlances even in this case.

Moreover, it would be interesting to better illustrate the different solutions gained in every case and the coverage of the normal facilities. In the following figures Figs. 2, 3 and 4, the solutions of example 1 of 225 nodes are depicted for all three cases. Each square

76	21	21	21	21	21	21	21	21	21	21	58	58	58	58
21	21	21	21	21	21	21	21	21	21	21	21	58	58	58
76	21	21	21	21	21	21	21	21	21	86	58	58	58	58
76	76	21	21	21	21	21	21	21	86	86	86	86	58	58
76	76	76	21	21	21	21	21	86	86	86	86	86	58	58
76	76	76	76	21	21	21	86	86	86	86	86	86	86	58
76	76	76	76	76	21	94	94	86	86	86	86	86	58	58
76	76	76	76	94	94	94	94	159	86	86	86	58	58	179
76	76	76	94	94	94	94	159	159	159	86	179	58	179	179
76	76	183	94	94	94	159	159	159	159	159	179	179	179	179
76	183	183	183	94	159	159	159	159	159	159	159	179	179	179
183	183	183	183	183	216	159	159	159	159	159	179	179	179	179
183	183	183	183	183	183	216	159	159	159	179	179	179	179	179
183	183	183	183	183	216	216	216	159	216	179	179	179	179	179
216	183	183	183	216	216	216	216	216	216	216	179	179	179	179

Fig. 3 Solution (225 nodes, case 2, example 1). 3 vans and 6 motorcycles

20	20	20	20	20	20	20	20	20	20	43	43	43	43	43	43
20	20	20	20	20	20	20	20	20	20	20	86	43	43	43	43
20	20	20	20	20	20	20	20	20	20	86	86	86	43	43	43
93	20	20	20	20	20	20	20	20	86	86	86	86	86	43	43
93	93	20	20	20	20	20	86	86	86	86	86	86	86	86	43
93	93	93	20	20	20	86	86	86	86	86	86	86	86	86	86
93	93	93	93	20	93	93	86	86	86	86	86	86	86	86	118
106	93	93	93	93	93	172	118	86	86	86	86	118	118	118	118
93	93	93	93	93	172	172	172	118	86	86	86	118	118	118	118
93	93	93	93	172	172	172	172	172	118	86	118	118	118	118	118
106	93	93	172	172	172	172	172	172	172	118	118	118	118	118	118
106	106	172	172	172	172	172	172	172	172	172	118	118	118	118	192
106	215	215	172	172	172	172	172	172	172	192	192	118	192	192	192
215	215	215	215	172	172	172	172	172	192	192	192	192	192	192	192
215	215	215	215	215	172	172	172	192	192	192	192	192	192	192	192

Fig. 4 Solution (225 nodes, case 3, example 1). 2 vans and 7 motorcycles

corresponds to a node of the network. Table 6 offers an explanation of the different shades used in the figures. It should be noted that the number inside every square corresponds to the facility (either normal or back-up) that is the first service provider to that square.

As the cost and the coverage distance of the normal facilities (vans) increases (from case 1 to case 3), their number decreases, while their distribution tends to quite resemble with each other, since the most important nodes (which the normal facilities are motivated to cover) are the same in all cases.

5 Conclusion

A double-type double-standard model (DtDsM) was proposed for maximal covering location problem (MCLP) which is very useful in a wide range of applications like public emergency services in which it is important for each demand point to receive at least a primary service (from a backup facility) within a predefined time before receiving a full service (from a normal facility which cannot serve the demand within the predefined time).

Since the parameters of the problem can widely vary from one case to another, relocation of facilities during a time horizon is inevitable, and developing fast solution

Table 6 Memo for following figures

55	At this location a van is deployed
55	This node is within the coverage distance of a van (denoted by the number)
27	At this location a motorcycle is deployed
27	This node is within the coverage distance of a motorcycle (denoted by the number)

algorithms to reset and resolve real facility location problems is very important. An efficient Benders-based decomposition algorithm was proposed to solve the model which is much faster than the commercial solver CPLEX for solving DtDsM.

It is worth noting that if we assume a standard response time of 5 min and an average motorlance speed of 60 km/h, then the coverage distance of a motorlance is approximately 5 km, and the assumption $MCD=5$ can be interpreted so that the distance between two adjacent points is 1 km. Therefore, a 10×10 network (with 100 nodes) can be considered for the location problem in a region with an area of $9 \times 9 = 81$ square kilometers. Similarly, one can see that a 30×30 network (with 900 nodes) can be considered for the location problem in a region with an area of $29 \times 29 = 841$ square kilometers. This means that our largest test problem can be considered as a simulation of the problem of locating the emergency vehicles in a large city like Tehran (with an area of approximately 730 km^2).

Finally, it is worth noting that although the proposed DtDsM is in fact an incapacitated maximal covering location problem (i.e., no limit is considered on the amount of demands allocated to a single facility), it is still useful and applicable in real emergency service planning if the probability of two demand points (allocated to an identical facility) to be the scene of accidents at the same time is assumed to be very low.

Developing an efficient solution algorithm for a capacitated version of DtDsM can be considered in future studies.

Compliance with Ethical Standards

Conflict of Interest The author declares that there is no conflict of interest.

References

1. Cremer H, De Kerchove AM, Thisse JF (1985) An economic theory of public facilities in space. *Math Soc Sci* 9(3):249–262
2. Yin P, Mu L (2012) Modular capacitated maximal covering location problem for the optimal siting of emergency vehicles. *Appl Geogr* 34:247–254
3. Church R, ReVelle C (1974) The maximal covering location problem. *Pap Reg Sci* 32(1):101–118
4. Gendreau M, Laporte G, Semet F (1997) Solving an ambulance location model by tabu search. *Loc Sci* 5(2):77–88
5. Laporte G, Louveaux F, Semet F, hirion A (2009) Application of the double standard model for ambulance location, In: Nunen J., Speranza M., Bertazzi L. (eds) *Innovations in Distribution Logistics. Lecture Notes in Econom and Math Systems*, vol. 619, Springer, Berlin, Heidelberg
6. Hogan K, ReVelle CS (1986) Concepts and applications of backup coverage. *Manage Sci* 32(11):1434–1444
7. Daskin MS (1983) Maximum expected covering location model: formulation, properties and heuristic solution. *Transp Sci* 17(1):48–70
8. Karaman M (2008) A Genetic algorithm for the multi-level maximal covering ambulance location problem. Master of Science Thesis: Middle East Technical University
9. Jabalameli MS, Bankian Tabrizi B, Moshref Javadi M (2010) Capacitated facility location problem with variable coverage radius in distribution system. *Int J Ind Eng Prod Res* 21:231–237
10. Caunhye AM, Nie X, Pokharel S (2012) Optimization models in emergency logistics: a literature review. *Socio-Econ Plan Sci* 46(1):4–13
11. Marques MdC, Dias JM (2013) Simple dynamic location problem with uncertainty: a primal-dual heuristic approach. *Optimization* 62(10):1379–1397

12. Rajagopalan HK, Saydam C, Xiao J (2008) A multi-period set covering location model for dynamic redeployment of ambulances. *Comput Oper Res* 35(3):814–826
13. Schmid V, Doerner KF (2010) Ambulance location and relocation problems with time-dependent travel times. *Eur J Oper Res* 207(3):1293–1303
14. Tang L, Jiang W, Saharidis G (2013) An improved Benders decomposition algorithm for the logistics facility location problem with capacity expansions. *Ann Oper Res* 210(1):165–190
15. Rohaninejad M, Sahraeian R, Tavakkoli Moghaddam R (2018) An accelerated Benders decomposition algorithm for reliable facility location problems in multi-echelon networks. *Comput Ind Eng* 124:523–534
16. Cordeau JF, Furini F, Ljubic I (2019) Benders decomposition for very large scale partial set covering and maximal covering location problems. *Eur J Oper Res* 275(3):882–896
17. Karamyar F, Sadeghi J, Modares Yazdi M (2018) A Benders decomposition for the location-allocation and scheduling model in a healthcare system regarding robust optimization. *Neural Comput Appl* 29:873–886
18. Hosseininassab A, Gzara F (2020) Effects of feasibility cuts in Lagrangian relaxation for a two-stage stochastic facility location and network flow problem. *Optim Lett* 14:171–193
19. Rahmaniani R, Crainic TG, Gendreau M, Rei W (2017) The Benders decomposition algorithm: a literature review. *Eur J Oper Res* 259(3):801–817
20. Fragkogios A, Saharidis GK (2018) Latest advances on Benders decomposition, in *Encyclopedia of Information Science and Technology, Fourth Edition*, IGI GLOBAL, 5411–5421
21. Agnetis A, Grande E, Pacifici A (2012) Demand allocation with latency cost functions. *Math Program Ser A* 132:277–294
22. Fakhri A, Ghatee M, Fragkogios A, Saharidis G (2017) Benders decomposition with integer subproblem. *Expert Syst Appl* 89:20–30
23. Ghouila-Houri A (1962) Caractérisation des Matrices Totalement Unimodulaires. *C R Acad Sci Paris* 254:1192–1194
24. Camion P (1965) Characterization of totally unimodular matrices. *Proc Am Math Soc* 16(5):1068–1073
25. Bazaraa MS, Jarvis JJ, Sherali HD (2010) *Linear programming and network flows-4th ed.*, Wiley, New Jersey.
26. Black PE (2019) Manhattan distance, in *Dictionary of Algorithms and Data Structures* [online].

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.