CIRRELT-2021-21

A Benders Decomposition Algorithm for the Maximum Availability Service Facility Location Problem

Ali Muffak
Okan Arslan

May 2021

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

A Benders Decomposition Algorithm for the Maximum Availability Service Facility Location Problem

Ali Muffak, Okan Arslan*

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and, Department of Decisions Sciences, HEC Montréal, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, QC H3T 2A7, Canada

Abstract. This paper introduces the maximum availability service facility location problem, which integrates the set covering and flow capturing problems to service both stationary and mobile demand in an urban region. The problem has applications in location of government offices, medical facilities and polling stations. We present a mixed-integer linear programming formulation and develop a Benders decomposition algorithm. We implement several acceleration techniques including multi-cut and Pareto-optimal cut generation. We construct these cuts analytically using closed-form expressions for subproblem solutions. Our best algorithm optimally solves instances with up to one thousand nodes, one million commuting customers and one hundred candidate facilities. We also conduct a case study with real data from the city of Chicago and show an application of our model for the location of medical facilities in a pandemic situation. We find that confinement restrictions in a pandemic do not significantly affect the total demand coverage, but facility layout may be significantly different under different confinement levels.

Keywords: location optimization, service facility, maximum availability, Benders decomposition, Pareto-optimal cut.

* Corresponding author: okan.arslan@hec.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2021

1. Introduction

Service facilities such as government offices, medical centers, refueling stations and automatic teller machines offer various types of services to the public, and their location has an impact on their availability and ultimately on their success. Individuals can visit such facilities on an independent trip or as part of their daily commute on their preplanned paths. In this paper, we consider both types of customers when planning for the location of service facilities and maximize their availability by respecting fairness among the individuals.

The location optimization of service facilities aims at maximizing the covered demand. The demand can be stationary, such as in residential areas or work places where the customers are immobile for extended periods of time, or they can be mobile and travel during the period when facilities provide their service. We now review the problems in location science that address the two demand types. Hodgson (1990) introduces the flow capturing location problem (FCLP) where the demand is considered to be mobile and is defined as an origin-destination (OD) pair. The objective is to cover the maximum demand by opening a given number of facilities. Berman et al. (1992) independently introduce the discretionary service facility location problem, which is structurally the same as the FCLP. The authors propose a solution method based on branch-and-bound (B&B) algorithm and present computational results on random networks. Real-world applications of the FCLP are investigated by Hodgson et al. (1996) in a traffic network from the city of Edmonton. In the basic FCLP definition, the drivers are neutral to the location of facilities and do not change their preplanned paths to visit a facility. The problem has witnessed a surge of interest in the literature and one of the touted applications is the the flow refueling location problem (FRLP) by Kuby and Lim (2005), in which covering a demand requires intercepting a vehicle flow possibly for multiple times en route. This ensures the connectivity of the OD pair without running out of the fuel. Upchurch et al. (2009) extended the FRLP by including the capacities of refueling stations. The above FRLP models assume that the vehicles are half-fuelled at the beginning of their trip. The FCLP can be categorized into three groups according to the drivers' behavior (Arslan et al., 2018): the drivers can be neutral, cooperative or non-cooperative to change their paths to visit or avoid the facilities. In the cooperative case, Berman et al. (1995) consider driver deviation from their preplanned trips to visit the service facilities and Kim and Kuby (2012) implement the same idea for the FRLP. Yıldız et al. (2016) develop a branch-and-price algorithm and Arslan et al. (2019) develop a branch-and-cut algorithm for the same problem. When the drivers are noncooperative, as in the case of overweight truck drivers avoiding to encounter weighing stations, the problem we face is then an evasive flow capturing problem (Marković et al., 2015). Other relevant applications include health care facility locations. Taymaz et al. (2020) address the problem of locating health care facilities for mobile workers, who work across cities under severe conditions that debilitate their health. The authors propose a stochastic model to locate walk-in clinics and allocate different services by considering mobile demand.

While the FCLP captures demand flow between OD pairs, covering location problems (CLP) are used to locate facilities by considering the coverage of stationary demand. Depending on the available resources, all the demand can be covered by minimizing the number facilities or a maximum amount of demand can be covered by locating a given number of facilities. Covering problems are extensively used in various applications including humanitarian logistics and health care. Ndiaye and Alfares (2008) introduce a facility location model for healthcare facilities for nomadic or Bedouin populations that move seasonally. They present a facility location model to determine the number and location of the healthcare facilities while minimizing the relocation and operating costs of the facilities. Erkut et al. (2008) propose a model for locating emergency medical service facilities. The authors enhance three classical covering models by including a survival function that puts in relation the response time and survival probability. They show that their model produces more realistic solutions than the classic maximal covering location models. Murali et al. (2012) introduce a capacitated facility location problem to locate medical distribution centers in case of a bio-terror attack. They assume that demand satisfaction is related to distance and they consider demand uncertainty in their model. Recent applications of the CLP include facility location for drones. Chauhan et al. (2019) introduce the problem and propose a 3-stage heuristic approach to solve it. We refer to Snyder (2011) and García and Marín (2019) for various CLP applications and their extensions.

FCLP and CLP are among numerous location models used to optimize facility locations. Turkoglu and Genevois (2020) provide a comprehensive survey of service facility locations and categorize them by application, from banking and health care services to food and refueling applications. While these models can be used in multiple settings, various application contexts require the need to cover the demand at their origins and destinations as well as the need to capture the demand flow on the OD paths. In particular, a regular daily routine of a working individual starts at their home, continues with a commute and finally a journey from work back to home. Depending on their departure and arrival times, they may find a service facility at their origin, on their commute route or at their destination. In the same network, there are also stationary demands that are located at nodes and the service facility planning process needs to take into consideration both types of demand. Such a demand definition is applicable in the location of several facility types:

- *Government offices:* such as immigration services, employment and pension services or health insurance services,
- *Testing and vaccination clinics during a pandemic:* The goal is to provide the population with an opportunity to get tested in close proximity to their homes, schools, workplaces, or on their daily commute,
- *Polling stations for voting.*

Even when the customers are in close proximity to the service facilities, they have limited access to these facilities due to their daily routine and work or school commitments. The aim is to provide customers the opportunity to benefit from these services during their available time. To this end, we introduce *the maximum availability service facility location problem* (MASFLP), which locates service facilities by maximizing their availability to the customers. A demand is defined as an OD pair and it can be covered at the origin, at the destination, or on their way from their origin to the destination. The coverage is defined in terms of the time they have access to at least one facility and is bounded above by a limit. This demand definition allows us to model the customers who commute using their private vehicles or public transportation as well as the stationary customers at nodes. Time dimension is involved in this demand definition. Customers can have available time at their origin (for example, before work or school), on their journey to work, or at their destination (for example, during breaks or after work). To ensure fairness, each demand point can only be covered for a limited number of hours. This guarantees that the service facilities are not concentrated in densely populated regions and all the demand are treated fairly. The MASFLP also handles flow cannibalization (Hodgson, 1990) that arises when facilities are located at strategic intersections with high flow, which leads to covering high flows multiple times and ignoring smaller demand flows. Further elaborations are presented in Section 2.

Solution methods for the FCLP and the CLP have been explored extensively, and Benders decomposition (BD) algorithms (Benders, 1962) are especially suited for the structure of these location problems. Arslan and Karaşan (2016) use BD decomposition and propose different cut implementations to solve the charging station location problem with electric vehicles and plug-in hybrid vehicles. Cordeau et al. (2019) introduce a new approach based on BD to solve large scale partial set covering problems and maximal covering problems. Their algorithm is capable of solving massive data instances. BD methods are also used in other applications. Zetina et al. (2019) present two exact algorithms to solve the multicommmodity uncapacitated fixed-charge network design problem. They show that their algorithm provides an acceleration when compared to CPLEX. We refer to Costa (2005) and Rahmaniani et al. (2017) for a review on the application of BD algorithms in different optimization problems. Other solution methods are covered in the survey by Turkoglu and Genevois (2020).

The location optimization of service facility requires consideration of various factors. The main contributions of this paper are as follows:

- We introduce the maximum availability service facility location problem (MASFLP) by considering both stationary and mobile demand in an urban region. We also consider fairness among the individuals.
- We present a mixed-integer linear programming formulation for the MASFLP.
- We develop a BD algorithm to solve the model and test various acceleration techniques

including multi-cut and Pareto-optimal cut generation. We exploit the subproblem structure to solve them analytically and generate Pareto-optimal cuts as closed-form expressions.

- We conduct extensive computational experiments to evaluate the effectiveness and performance of our proposed solutions compared to CPLEX.
- We demonstrate the efficiency of our algorithm on random networks and present a case study in the city of Chicago for testing center location optimization in a pandemic situation.

The rest of the paper is organized as follows. In Section 2, we introduce the notation and the mathematical model. We present the BD algorithm and the analytical solutions of the subproblems in Section 3, the computational experiments in Section 4, and we conduct a case study in Section 5. Finally, we conclude and offer directions for future work in Section 6.

2. Mathematical model

We now introduce the problem and present a mathematical formulation for the MASFLP.

2.1. Problem definition and notation

Let K be the set of candidate facilities and m be the number of facilities to locate. Let Q be the set of customer demands, and $G = (N, A)$ be the transportation network where N is the set of nodes and $A = \{(i, j) : i, j \in N, i \neq j\}$ is the set of directed arcs. The length of arc $(i, j) \in A$ is l_{ij}. We assume that the distance matrix satisfies the triangular inequality. Each customer demand $q \in Q$ is defined by $\langle o_q, d_q, f_q, t_q^o, t_q^d, t_q^p, \lambda_q \rangle$, where o_q and d_q are the origin and destination nodes, respectively, and f_q is the volume of customers living at o_q, working at d_q and commuting in between. Parameters t_q^o, t_q^d and t_q^p represent the customers' available time during the business hours at the origin, at the destination, and on the OD path, respectively, and λ_q is the customers' deviation tolerance from their preplanned OD path. A non-commuting demand residing at o_q can be represented by setting $d_q = o_q$. Each candidate facility $k \in K$ has a coverage range of r_k. A facility k covers demand at node i if $l_{ik} \leq r_k$, and on path between o_q and d_q if $l_{o_q k} + l_{k d_q} \leq (1 + \lambda_q) l_{o_q d_q}$. For a $q \in Q$, the maximum contribution to the objective function can be at most t_{max}, which ensures that the service is provided fairly to all demand in the network. The service is defined as time availability and not as binary, hence the higher the availability the higher the service provided. Times t_q^o, t_q^d and t_q^p can be regarded as a weight for each demand $q \in Q$ and do not encourage flow cannibalization, which is be handled by binary location variables defined in Section 2.2 as introduced by Hodgson (1990). Let N_p the set of potential locations on path p, N_o the set of potential locations covering origin o, and N_d the set of potential locations covering destination d.

Definition 1. *The MASFLP is defined as selecting a subset of K to open such that the total availability provided to the demand is maximized and the maximum contribution of each demand is at most t_{max}.*

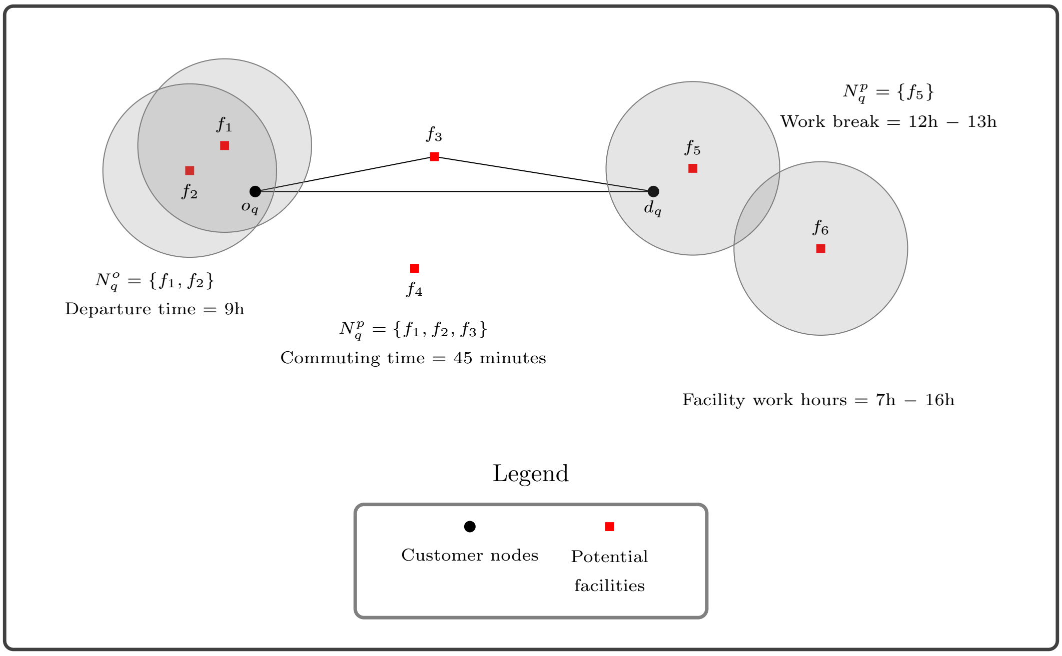


Figure 1: Network model of the MASFLP.

Figure 1 shows an example of a demand on a small network. Observe that demand q can be covered at origin o_q by facilities f_1 and f_2 for $t_q^o = 2$ hours. Availability time t_q^o is the difference between the facility opening hour (7h) and the departure time (9h). Demand q can also be covered on the OD path by facilities f_1, f_2 and f_3 for a fixed amount of time, which is taken as $t_q^p = 45$ minutes and around the destination d_q by facility f_5 for $t_q^d = 1$ hour during the work break. Demand q cannot be covered by f_4 because the driver is intolerant to high deviations and cannot be covered by f_6 either because node d_q is not within the facility's range.

Remark 1. *If $o_q = d_q$, $t_q^p = t_q^d = 0$ for all $q \in Q$, and t_{max} is a very large number, the MASFLP reduces to the MCLP.*

Remark 2. *If $t_q^o = t_q^d = 0$ and $t_q^p = 1$ for all $q \in Q$ and t_{max} is a very large number, the MASFLP reduces to the FCLP.*

Therefore, the MASFLP generalizes the MCLP and the FCLP and is NP-hard.

2.2. Mathematical model

We use the following decision variables to formulate the MASFLP.
$x_k = 1$ if a facility is located at location k, 0 otherwise,
$y_q^o = 1$ if demand q is captured at origin o_q, 0 otherwise,
$y_q^d = 1$ if demand q is captured at destination d_q, 0 otherwise,
$y_q^p = 1$ if demand q is captured on path p_q, 0 otherwise and
θ_q is the time that a service facility is available for demand q.

The MASFLP is then formulated as the following MILP.

$$\text{maximize} \quad \sum_{q \in Q} \theta_q f_q \tag{1}$$

$$\text{subject to} \quad \sum_{k \in K} x_k = m \tag{2}$$

$$\theta_q \leq t_{max} \qquad q \in Q \tag{3}$$

$$\theta_q \leq t_q^o y_q^o + t_q^d y_q^d + t_q^p y_q^p \qquad q \in Q \tag{4}$$

$$\sum_{k \in N_q^o} x_k \geq y_q^o \qquad q \in Q \tag{5}$$

$$\sum_{k \in N_q^d} x_k \geq y_q^d \qquad q \in Q \tag{6}$$

$$\sum_{k \in N_q^p} x_k \geq y_q^p \qquad q \in Q \tag{7}$$

$$\theta_q \geq 0 \qquad q \in Q \tag{8}$$

$$x_k \in \{0, 1\} \qquad k \in K \tag{9}$$

$$y_q^o, y_q^d, y_q^p \in \{0, 1\} \qquad q \in Q. \tag{10}$$

The objective function maximizes total availability of facilities to the customers. Constraints (3) ensure that the available time of demand $q \in Q$ to the service does not exceed a maximum amount of time t_{max}. Constraints (4) ensure that the available time is correctly calculated. Constraints (5), (6), and (7) ensure that a demand is covered only if a facility is open within the coverage range of its origin and destination or on an OD path. Finally, constraints (8), (9), and (10) define the domain of variables. It is straightforward to show that, without loss of generality, we can relax the integrality requirement of the y_q^o, y_q^d, and y_q^p variables. We refer to the formulation with relaxed y variables as P.

3. Benders Decomposition

In this section, we develop a BD algorithm (Benders, 1962) for solving the P model. BD is a decomposition algorithm, in which the continuous variables are projected out from the formulation and an exponential number of constraints are appended instead. These constraints correspond to the extreme points and rays of the linear programming model, which is obtained by fixing the integer variables in the original formulation. Adding all such cuts is impractical, therefore they are added to the formulation as needed. For more information on various implementation details of the algorithm, we refer the reader to Rahmaniani et al. (2017). We now present the reformulation of our problem and introduce analytical solutions of the subproblems and different cut generation strategies to accelerate the algorithm.

3.1. Benders subproblem

By fixing the variable x in the P formulation to $\hat{x} \in \{0,1\}^{|K|}$, we obtain a linear programming (LP) model, which we refer to as subproblem SP($\hat{x}$).

$$
\begin{aligned}
\text{maximize} \quad & \sum_{q \in Q} \theta_q f_q && && (11)\\
\text{subject to} \quad & \theta_q \le t_{max} && q \in Q && (12)\\
& \theta_q - t_q^o y_q^o - t_q^d y_q^d - t_q^p y_q^p \le 0 && q \in Q && (13)\\
& y_q^o \le \sum_{k \in N_q^o} \hat{x}_k && q \in Q && (14)\\
& y_q^d \le \sum_{k \in N_q^d} \hat{x}_k && q \in Q && (15)\\
& y_q^p \le \sum_{k \in N_q^p} \hat{x}_k && q \in Q && (16)\\
& y_q^o \le 1 && q \in Q && (17)\\
& y_q^d \le 1 && q \in Q && (18)\\
& y_q^p \le 1 && q \in Q && (19)\\
& \theta_q, y_q^o, y_q^d, y_q^p \ge 0 && q \in Q. && (20)
\end{aligned}
$$

Note that the subproblem is closed and bounded and therefore feasibility cuts are not required and optimality cuts are enough to ensure convergence of the algorithm. Let $\alpha, \beta, \gamma, \mu, \rho, \phi, \sigma, \delta$ be the dual variables associated with constraints (12)$-$(19), respectively. Then the dual subproblem referred to as DSP($\hat{x}$) is the following:

$$
\text{minimize} \quad \sum_{q \in Q} t_{max}\alpha_q + \sum_{\substack{q \in Q \\ k \in N_q^o}} \hat{x}_k \gamma_q + \sum_{\substack{q \in Q \\ k \in N_q^d}} \hat{x}_k \mu_q + \sum_{\substack{q \in Q \\ k \in N_q^p}} \hat{x}_k \rho_q + \sum_{q \in Q} \phi_q + \sum_{q \in Q} \sigma_q + \sum_{q \in Q} \delta_q \qquad (21)
$$

$$
\text{subject to} \quad \alpha_q + \beta_q \ge f_q \qquad q \in Q \qquad (22)
$$

$$\gamma_q - t_q^o \beta_q + \phi_q \geq 0 \qquad q \in Q \tag{23}$$

$$\mu_q - t_q^d \beta_q + \sigma_q \geq 0 \qquad q \in Q \tag{24}$$

$$\rho_q - t_q^p \beta_q + \delta_q \geq 0 \qquad q \in Q \tag{25}$$

$$\alpha_q, \beta_q, \gamma_q, \mu_q, \rho_q, \phi_q, \sigma_q, \delta_q \geq 0 \qquad q \in Q. \tag{26}$$

3.2. Benders master problem

Let $\mathcal{D}$(DSP) denote the set of extreme points of DSP($\hat{x}$). We construct the master problem referred to as MP as follows:

$$\text{maximize} \quad \eta \tag{27}$$

$$\begin{aligned}\text{subject to} \quad \eta \leq & \sum_{q \in Q} t_{max} \alpha_q + \sum_{\substack{q \in Q \\ k \in N_q^o}} x_k \gamma_q + \sum_{\substack{q \in Q \\ k \in N_q^d}} x_k \mu_q \\ & + \sum_{\substack{q \in Q \\ k \in N_q^p}} x_k \rho_q + \sum_{q \in Q} \phi_q + \sum_{q \in Q} \sigma_q + \sum_{q \in Q} \delta_q \quad (\alpha, \gamma, \mu, \rho, \phi, \sigma, \delta) \in \mathcal{D}(DSP)\end{aligned} \tag{28}$$

$$\sum_{k \in K} x_k = m \tag{29}$$

$$x_k \in \{0, 1\} \qquad k \in K. \tag{30}$$

To handle the exponential number of constraints (28) in MP, we use a branch-and-cut approach. Cuts are added iteratively to the MP by solving the DSP($\hat{x}$). We solve the MP in a single B&B tree similar to Codato and Fischetti (2006).

3.3. Subproblem solution

Given $\hat{x}$, DSP($\hat{x}$) is an LP and can be solved using a solver. In this section, we present analytical solutions of DSP($\hat{x}$) without the need to build and solve an LP. First, note that the dual subproblem DSP($\hat{x}$) can be decomposed based on the demand $q \in Q$. We refer to the following decomposed dual subproblem formulation as DSP$_q$($\hat{x}$).

$$\text{minimize} \quad t_{max} \alpha_q + \sum_{k \in N_q^o} \hat{x}_k \gamma_q + \sum_{k \in N_q^d} \hat{x}_k \mu_q + \sum_{k \in N_q^p} \hat{x}_k \rho_q + \phi_q + \sigma_q + \delta_q \tag{31}$$

$$\text{subject to} \quad \alpha_q + \beta_q \geq f_q \tag{32}$$

$$\gamma_q - t_q^o \beta_q + \phi_q \geq 0 \tag{33}$$

$$\mu_q - t_q^d \beta_q + \sigma_q \geq 0 \tag{34}$$

$$\rho_q - t_q^p \beta_q + \delta_q \geq 0 \tag{35}$$

$$\alpha_q, \beta_q, \gamma_q, \mu_q, \rho_q, \phi_q, \sigma_q, \delta_q \geq 0. \tag{36}$$

We now start constructing a closed-form solution for $\mathrm{DSP}_q(\hat{x})$. For conciseness, let $\hat{x}^o_q = \sum_{k \in N^o_q} \hat{x}_k$, $\hat{x}^d_q = \sum_{k \in N^d_q} \hat{x}_k$, and $\hat{x}^d_q = \sum_{k \in N^p_q} \hat{x}_k$.

Proposition 1. *Constraint* (32) *is always active in an optimal solution of* $DSP_q(\hat{x})$.

PROOF. Observe that $\mathrm{SP}(\hat{x})$ can be decomposed based on q. For a given $\hat{q} \in Q$, consider the primal-dual subproblem pair. Constraint (32) in the $\mathrm{DSP}_q(\hat{x})$ corresponds to the non-negative θ_q variable in the MP. When $\theta_q = 0$, the particular demand cannot be covered and therefore the objective function value is zero. The dual objective function value therefore equals zero due to strong duality. Observe that the solution $\alpha_q = \phi_q = \sigma_q = \delta_q = 0$, $\beta_q = f_q$, $\gamma_q = t_0\beta_q$, $\mu_q = t_d\beta_q$ and $\rho_q = t_p\beta_q$ is feasible and has an objective function value of zero. Therefore it is optimal and satisfies constraint (32) at equality. Now, consider the case with $\theta_q > 0$. Constraint (32) is then always active due to complementary slackness conditions. □

Proposition 1 implies that we can set $\alpha_q = f_q - \beta_q$ and variable α_q can be projected out from the formulation by replacing this term in the objective function. We then obtain the following equivalent model.

$$
\begin{aligned}
\text{maximize} \quad & t_{max}\beta_q - \hat{x}^o_q\gamma_q - \hat{x}^d_q\mu_q - \hat{x}^p_q\rho_q - \phi_q - \sigma_q - \delta_q && (37)\\
\text{subject to} \quad & \gamma_q + \phi_q \geq t^o_q\beta_q && (38)\\
& \mu_q + \sigma_q \geq t^d_q\beta_q && (39)\\
& \rho_q + \delta_q \geq t^p_q\beta_q && (40)\\
& f_q \geq \beta_q && (41)\\
& \beta_q, \gamma_q, \mu_q, \rho_q, \phi_q, \sigma_q, \delta_q \geq 0. && (42)
\end{aligned}
$$

Note that the nonnegativity of α_q induces constraint (41). The objective function is also modified to a maximization to better see the multidimensional knapsack nature of the model.

For fixed $\hat{x}$ and a given $q \in Q$, we define $g_q(\hat{x}) = t_{max} - t_o \min\{\hat{x}^o_q, 1\} - t_d \min\{\hat{x}^d_q, 1\} - t_p \min\{\hat{x}^p_q, 1\}$, which represents the gap between t_{max} and the potential availability of facilities for demand q. If $g_q(\hat{x}) < 0$, the existing network of facilities can provide more availability than t_{max} and If $g_q(\hat{x}) > 0$, the availability of the facilities is less than t_{max} for q. We are now ready to present a closed-form solution for the $\mathrm{DSP}_q(\hat{x})$.

Proposition 2. *Given $\hat{x} \in \{0,1\}^{|K|}$ and a demand $q \in Q$, the following is an optimal solution of the $DSP_q(\hat{x})$:*

$$\beta_q = \begin{cases} f_q & \textit{if } g_q(\hat{x}) \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$

$$\gamma_q = \begin{cases} t_q^o f_q & \textit{if } g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^o \leq 1 \\ 0 & \textit{otherwise} \end{cases}$$

$$\mu_q = \begin{cases} t_q^d f_q & \textit{if } g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^o \leq 1 \\ 0 & \textit{otherwise} \end{cases}$$

$$\rho_q = \begin{cases} t_q^p f_q & \textit{if } g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^o \leq 1 \\ 0 & \textit{otherwise} \end{cases}$$

$$\phi_q = t_q^o f_q - \gamma_q$$

$$\sigma_q = t_q^d f_q - \mu_q$$

$$\delta_q = t_q^p f_q - \rho_q$$

□

PROOF. The constructed solution is feasible. Observe that the objective function coefficients of γ and ϕ variables in model (37)−(42) are negative and these two variables only appear in constraint (38), which implies that there exists an optimal solution satisfying constraint (38) at equality, that is $\gamma_q + \phi_q = t_o\beta_q$. Therefore, the objective function contribution of γ and ϕ is $-\hat{x}_q^o\gamma_q - \phi_q = -\min\{\hat{x}_q^0, 1\}t_0\beta_q$. With a similar reasoning for the other variables, the objective function can be restated as $t_{max}\beta_q - \min\{\hat{x}_q^o, 1\}t_o\beta_q - \min\{\hat{x}_q^d, 1\}t_d\beta_q - \min\{\hat{x}_q^p, 1\}t_p\beta_q$. Hence, the optimal objective function value is zero if $g_q(\hat{x}) := t_{max} - \min\{\hat{x}_q^o, 1\}t_o - \min\{\hat{x}_q^d, 1\}t_d - \min\{\hat{x}_q^p, 1\}t_p \leq 0$. If $g_q(\hat{x}) > 0$, we have $\beta_q = f_q$. Due to the knapsack nature between the remaining variables, the conditions follow. □

In the following, we present three Benders cut selection schemes: single-cut, multi-cut and Pareto-optimal cut.

3.4. Benders cut selection scheme 1: single-cut

For a given $\hat{x}$, the dual values are computed according to Proposition 2, then a cut is added to the MP at each iteration in the form of inequality (28).

3.5. Benders cut selection scheme 2: multi-cut

The Multiple-cut generation scheme is implemented and is proved to be efficient in various studies including de Camargo et al. (2008) and You and Grossmann (2013). To implement the multi-cut scheme, we need to modify the master problem. A new set of variables η_q is introduced to the MP, each of which approximates the cost of a decomposed subproblem. Let η_q be the variable associated with $\text{DSP}_q(\hat{x})$, and $\mathcal{D}(DSP_q)$ the set of extreme points of the dual problems DSP_q. The modified master problem is formulated as follows.

$$\text{maximize} \quad \sum_{q \in Q} \eta_q \tag{43}$$

$$\text{subject to} \quad \eta_q \leq t_{max}\alpha_q + \sum_{k \in N_q^o} x_k \gamma_q + \sum_{k \in N_q^d} x_k \mu_q + \sum_{k \in N_q^p} x_k \rho_q + \phi_q + \sigma_q + \delta_q \qquad (\alpha, \gamma, \mu, \rho, \phi, \sigma, \delta) \in \mathcal{D}(DSP_q),\ q \in Q \tag{44}$$

$$\sum_{k \in K} x_k = m \tag{45}$$

$$x_k \in \{0, 1\} \qquad k \in K. \tag{46}$$

For a given $\hat{x}$ vector, the values of the dual variables for each subproblem are computed according to Proposition 2 and each subproblem solution represents an extreme point used to generate a cut.

3.6. Benders cut selection scheme 3: Pareto-optimal cut

Generating effective cuts that can reduce the number of iterations is highly important. As we observe in Proposition 2, the subproblem can have multiple optimal values. In fact, when $\hat{x}_q^o = 1$, γ_q and ϕ_q values are interchangeable, which implies that infinitely many different cuts can be generated from the same $\hat{x}$ solution. With the objective of generating stronger cuts by selecting one among these alternative cuts, we now investigate the Pareto-optimal cut generation scheme (Magnanti and Wong, 1981). Let $\mathcal{X}$ be the feasible solution set of the master problem, $opt(\text{DSP}(\hat{x}))$ be the optimal objective function value of problem $\text{DSP}(\hat{x})$, and the function $\mathcal{C}(x, d)$ be defined as follows:

$$\mathcal{C}(x, d) = \sum_{q \in Q} t_{max}\alpha_q + \sum_{\substack{q \in Q \\ k \in N_q^o}} x_k \gamma_q + \sum_{\substack{q \in Q \\ k \in N_q^d}} x_k \mu_q + \sum_{\substack{q \in Q \\ k \in N_q^p}} x_k \rho_q + \sum_{q \in Q} \phi_q + \sum_{q \in Q} \sigma_q + \sum_{q \in Q} \delta_q,$$

where we use $d = (\alpha, \gamma, \mu, \rho, \phi, \sigma, \delta)$ to refer to dual variables of DSP(x), for conciseness.

In the Pareto-optimal cut scheme, we only add non-dominated cuts. A cut generated by the dual solution $\bar{d}$ is considered to be dominated by the cut generated by the dual solution $\tilde{d}$, if and only if $\mathcal{C}(x, \bar{d}) \leq \mathcal{C}(x, \tilde{d})$ for all $x \in \mathcal{X}$ with a strict inequality for at least one of the points (Magnanti and Wong, 1981). Solving the following LP allows us to obtain Pareto-optimal cuts.

$$\begin{aligned} \text{minimize} \quad & \mathcal{C}(\bar{x}, d) && (47) \\ \text{subject to} \quad & (22) - (26) \\ & \mathcal{C}(\hat{x}, d) = opt(DSP(\hat{x})). && (48) \end{aligned}$$

We refer to the formulation above as MW$(\hat{x}, \bar{x})$, where $\bar{x}$, referred to as the core point, is an interior point of the convex hull of $\mathcal{X}$. The initial core point is a vector of length $|K|$ and values $\frac{m}{|K|}$ for each dimension. Similar to Papadakos (2008), the core point is updated after each iteration and takes the value of the average of the previous iteration and the current MP solution.

3.7. Magnanti-Wong problem solution

The MW$(\hat{x}, \bar{x})$ can be decomposed based on demand $q \in Q$. Let $\bar{x}^o_q = \sum_{k \in N^o_q} \bar{x}_k$, $\bar{x}^d_q = \sum_{k \in N^d_q} \bar{x}_k$, and $\bar{x}^d_q = \sum_{k \in N^p_q} \bar{x}_k$. We refer to the problem below as MW$_q(\hat{x}, \bar{x})$.

$$\begin{aligned} \text{minimize} \quad & t_{max}\alpha_q + \bar{x}^o_q\gamma_q + \bar{x}^d_q\mu_q + \bar{x}^p_q\rho_q + \phi_q + \sigma_q + \delta_q && (49) \\ \text{subject to} \quad & \alpha_q + \beta_q \geq f_q && (50) \\ & \gamma_q - t^o_q\beta_q + \phi_q \geq 0 && (51) \\ & \mu_q - t^d_q\beta_q + \sigma_q \geq 0 && (52) \\ & \rho_q - t^p_q\beta_q + \delta_q \geq 0 && (53) \\ & \mathcal{C}(\hat{x}, d) = opt(DSP(\hat{x})) && (54) \\ & d_q \geq 0. && (55) \end{aligned}$$

The analytical solution of the MW$_q(\hat{x}, \bar{x})$ is as follows:

Proposition 3. *Given $\hat{x} \in \{0,1\}^{|K|}$ and a demand $q \in Q$, the following is an optimal solution of the $DSP_q(\hat{x})$:*

$$\alpha_q = \begin{cases} 0 & \textit{if } g_q(\hat{x}) \geq 0 \\ f_q & \textit{otherwise} \end{cases}$$

$$\beta_q = \begin{cases} f_q & \textit{if } g_q(\hat{x}) \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$

$$\gamma_q = \begin{cases} t_q^o f_q & \textit{if } (g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^o = 0) \textit{ or } (g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^o = 1 \textit{ and } \bar{x}_q^o \leq 1) \\ 0 & \textit{otherwise} \end{cases}$$

$$\mu_q = \begin{cases} t_q^d f_q & \textit{if } (g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^d = 0) \textit{ or } (g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^d = 1 \textit{ and } \bar{x}_q^d \leq 1) \\ 0 & \textit{otherwise} \end{cases}$$

$$\rho_q = \begin{cases} t_q^p f_q & \textit{if } (g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^p = 0) \textit{ or } (g_q(\hat{x}) \geq 0 \textit{ and } \hat{x}_q^p = 1 \textit{ and } \bar{x}_q^p \leq 1) \\ 0 & \textit{otherwise} \end{cases}$$

$$\phi_q = t_q^o f_q - \gamma_q$$

$$\sigma_q = t_q^d f_q - \mu_q$$

$$\delta_q = t_q^p f_q - \rho_q$$

□

The correctness of Proposition 3 follows from Proposition 2 and similar multi-dimensional knapsack arguments. For sake of conciseness and to avoid repetition, we omit the details here.

4. Computational Study

We now present the data, the design of experiments, and the performance of our algorithms. We present four different cut selection schemes and compare the solution efficiency of the BD algorithm to that of P model solved using CPLEX without any decomposition. We also carry out a case study using our best performing algorithm on a real-world Chicago dataset and discuss managerial findings. All the experiments are performed on a desktop computer with a 2.80 GHz 10-core Intel Core i9-10900F processor and 64GB of RAM, running 64-bit Windows operating system. The algorithms were implemented using Python 3.7.7 and the Python API of CPLEX 12.10.0.

4.1. Design of Experiments

We have randomly generated 10 different networks with a number of nodes n going from 100 to 1000 by increments of 100. The node coordinates are randomly selected in the interval $[0, 25]$. The demands are generated between all pairs of nodes and the customer volume f is assigned randomly in $[1, 100]$ interval and rounded to the nearest integer. We have coverage distance $d \in \{1, 3, 5\}$

and driver tolerance $\lambda \in \{10\%, 30\%, 50\%\}$. Sets N_o, N_d and N_p are constructed accordingly. A work day is divided into 30 minute intervals for a total duration of 8 hours and we generate t_q^o, t_q^d, and t_q^p randomly in $[1, 6], [1, 2]$, and $[1, 4]$, respectively, and rounding to the nearest integer. We set t_{max} to 8, and ensure that for any demand q, $t_o + t_p + t_d \leq 8$. All nodes are considered as potential facility locations and the number of OD pairs in each network is n^2. We set the maximum number of open facilities $m \in \{5, 10, 15, 20, 30, 40, 50, 75, 100\}$. Using nine randomly generated networks and these settings, we test the performance of the BD algorithm and compare different cut implementation schemes. All performance results presented in the tables hereafter represent the average performance on d and λ.

4.2. Computational Results

In this section, we first present the performance of constructing the subproblem solutions analytically as opposed to building an LP model and solving in CPLEX. We then compare different BD implementations using closed-form solutions presented in Section 3 with the implementation on CPLEX without any decomposition. We consider five implementations:

- CPLEX: the MILP is solved using CPLEX directly without any decomposition.
- BD-single: single-cut implementation.
- BD-multi: multi-cut implementation.
- BD-single-Pareto: single-cut implementation where the subproblem is solved analytically and only Pareto-optimal cuts are added.
- BD-multi-Pareto: multi-cut implementation where the subproblem is solved analytically and only Pareto-optimal cuts are added.

In Table 1, we present the results of the performance of BD-single with subproblems solved using LPs and BD-single with subproblems solved analytically. We consider four network sizes from 100 to 400 and the number of facility locations considered is 5, 10, or 15. The three leftmost columns represent the instance setting and solution; n is the number of nodes in the network, m is the number of facility locations, and "Opt. Sol." is the optimal objection function value of the corresponding instance. Columns 4−7 are the total solution time, the subproblem solution time, the number of subproblems, and lastly the average subproblem solution time, respectively, all in seconds (s). The average solution time for the BD-single with subproblems solved using LPs and BD-single with supbroblems solved analytically is 251.9 and 33.8 seconds, respectively. BD-single with supbroblems solved analytically is significantly more efficient and the acceleration increases more for larger networks.

We now present the results of the four BD implementations using analytical solutions and use CPLEX as a benchmark to compare the performance of these algorithms. The computational

Table 1: Comparison of BD subproblem solved analytically and subproblem solved using LPs.

Instance			BD-single (subproblems solved using LPs)				BD-single (subproblems solved analytically)			
n	m	Opt. Sol. ($\times 10^6$)	Total Sol. Time (s)	Subp. Sol. Time (s)	# Subproblems	Avg. Subp. Sol. Time (s)	Total Sol. Time (s)	Subp. Sol. Time (s)	# Subproblems	Avg. Subp. Sol. Time (s)
100	5	3.17	5.0	4.9	5	1.0	1.3	1.2	6	0.2
	10	3.29	13.0	12.9	12	1.0	5.5	5.4	23	0.2
	15	3.39	35.2	35.2	28	1.3	30.2	29.9	121	0.3
200	5	12.35	55.1	55.0	4	13.8	6.7	6.7	4	1.7
	10	12.60	73.7	73.7	6	12.3	22.8	22.7	14	1.6
	15	12.82	238.0	237.9	17	14.0	60.3	60.1	33	1.8
300	5	27.69	166.0	165.9	3	55.3	15.8	15.7	3	5.2
	10	28.13	329.8	329.7	5	66.0	29.3	29.2	5	5.9
	15	28.53	331.1	331.1	5	66.2	31.4	31.3	6	5.2
400	5	49.12	509.9	509.9	3	170.0	37.5	37.4	3	12.5
	10	49.70	759.8	759.8	4	190.0	53.8	53.8	4	13.4
	15	50.18	1010.7	1010.7	5	202.1	111.4	111.3	9	12.4

performances are shown in Table 2. The column "Instance" presents the network size, the number of facility locations and the objective function value. The next column "CPLEX" shows the solution time and the gap (%) using CPLEX implementation. The gap is defined as $(UB - LB)/UB^{*}100$, where UB and LB are upper and lower bounds, respectively. For each of the BD implementations, we present the number of iterations under the column "# Iterations", the total number of cuts under "# Cuts", the total subproblem solution time and the total solution time in the columns "Subp. Sol. Time (s)" and "Total Sol. Time (s)" respectively, and finally the optimality gap (%). Values "TL" in the table refer to the cases that terminated because of the time limit. We consider networks with 100, 200, and 300 nodes.

Table 3 summarizes the performance of the algorithms. The average solution times are 327.6, 288.7, 233.9, 1622.0, and 116.2 seconds for the CPLEX, BD-multi, BD-multi-Pareto, BD-single, and BD-single-Pareto, respectively. The table shows that all the algorithms but the BD-single are computationally more efficient than CPLEX, with the BD-single-Pareto being the most efficient.

This is expected due to the fact that Pareto-optimal cuts are stronger and adding them as single cuts reduces their number significantly compared to multiple cuts. The BD-single algorithm is the least efficient, as shown in Table 2, it fails to solve several instances optimally within a one hour time-limit. BD-multi and BD-multi-Pareto add on average 95365.3 and 49849.3 cuts, respectively as shown in Table 3, a reduction in the number of added cuts of almost half (47%). This shows the importance and efficacy of Pareto-optimal cuts. The average number of iterations required is 18.8, 17.4, 1987.0, and 56.8 for BD-multi, BD-multi-Pareto, BD-single, BD-single-Pareto, respectively. Both multi-cut algorithms require fewer iterations to solve the instances. BD-multi-Pareto and BD-single-Pareto perform the best out of all the algorithms with BD method, therefore, we present the performance of the two algorithms and CPLEX on larger instances to test their limits. Table 4 contains the results.

We considered increasing network sizes from 400 nodes to 1000 nodes with increments of a 100, and the number of facility locations m similar to instances in Table 2. Table 4 also contains the same parameters as Table 2. Values "TL" and "Memory" refer to the termination of the algorithm because of time and memory limits, respectively. The results show that the CPLEX algorithm is unable to solve the majority of instances to proven optimality within a one-hour time limit, meanwhile the BD-multi-Pareto fails due to memory limits. The BD-single-Pareto is able to solve all instances to optimality but 4 instances, those with $n \in \{900, 1000\}$ and $m \in \{75, 100\}$. BD-single-Pareto algorithm is the most efficient, it adds fewer cuts and requires the lowest number of iterations.

Table 2: Performance comparison of the BD algortihms and CPLEX on small instances.

Instance			CPLEX		BD-multi					BD-multi-Pareto					BD-single					BD-single-Pareto				
n	m	Opt. Sol. (10^6)	Total Sol. Time(s)	Gap (%)	# Iterations	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)	Gap (%)	# Iterations	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)	Gap (%)	# Iterations	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)	Gap (%)	# Iterations	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)	Gap (%)
100	5	3.17	4.4	0.0	8	19991	7.0	11.5	0.0	14	10105	11.1	15.2	0.0	6	4	1.3	1.3	0.0	10	3	1.3	2.3	0.0
	10	3.29	7.7	0.0	16	20253	7.0	13.8	0.0	16	10632	12.3	16.9	0.0	23	20	5.4	5.5	0.0	24	8	3.2	5.5	0.0
	15	3.39	10.6	0.0	36	21123	7.5	20.7	0.0	26	11430	18.3	25.0	0.0	121	115	29.9	30.2	0.0	56	22	8.1	13.4	0.0
	20	3.47	12.3	0.0	26	21341	7.7	18.4	0.0	28	11948	19.5	27.0	0.0	1440	1435	372.4	391.7	0.0	60	23	8.8	14.4	0.0
	30	3.60	11.1	0.0	16	22021	7.9	16.3	0.0	18	12706	14.2	19.5	0.0	8712	8702	2496.5	3007.5	0.0	114	45	16.6	27.3	0.0
	40	3.68	8.8	0.0	18	22719	8.1	16.1	0.0	18	12884	14.3	19.7	0.0	9440	9432	2612.7	3103.8	0.0	64	24	9.2	15.2	0.0
	50	3.74	5.5	0.0	22	22200	8.0	17.8	0.0	18	12631	14.3	19.6	0.0	8637	8632	2315.6	2942.8	0.0	46	15	6.3	10.5	0.0
	75	3.83	3.0	0.0	16	20439	7.3	14.3	0.0	10	10081	8.9	12.1	0.0	9335	9327	2528.6	3102.3	0.0	16	6	2.3	3.8	0.0
	100	3.87	0.8	0.0	6	19800	7.4	10.4	0.0	4	9900	5.8	7.4	0.0	3	2	0.6	0.6	0.0	6	2	0.8	1.4	0.0
200	5	12.35	65.5	0.0	8	79629	57.0	124.0	0.0	8	39921	59.8	84.0	0.0	4	3	6.7	6.7	0.0	10	4	9.4	14.8	0.0
	10	12.60	95.9	0.0	10	79906	57.4	130.5	0.0	12	40417	76.8	111.0	0.0	14	10	22.6	22.7	0.0	16	6	14.6	23.2	0.0
	15	12.83	116.7	0.0	22	80250	57.2	164.5	0.0	18	41003	101.9	151.1	0.0	33	29	60.1	60.3	0.0	24	9	21.6	34.5	0.0
	20	13.04	161.2	0.0	12	80567	57.6	137.6	0.0	12	41441	77.9	112.6	0.0	159	149	302.5	303.4	0.0	38	15	35.2	55.7	0.0
	30	13.42	190.0	0.0	24	81847	58.7	172.5	0.0	18	43444	103.3	151.3	0.0	520	513	1017.2	1024.5	0.0	62	20	52.5	86.0	0.0
	40	13.74	223.2	0.0	20	82848	59.2	165.6	0.0	20	44988	113.2	165.9	0.0	1815	1801	3564.3	TL	0.20	134	58	129.3	202.0	0.0
	50	14.01	207.1	0.0	34	84024	59.9	208.3	0.0	24	46059	130.0	192.2	0.0	1810	1802	3568.5	TL	0.38	152	62	141.1	223.6	0.0
	75	14.56	230.9	0.0	42	86130	61.2	234.6	0.0	34	49124	173.2	260.6	0.0	1803	1796	3567.1	TL	0.69	264	116	254.8	397.5	0.0
	100	14.94	266.9	0.0	22	86498	61.6	184.3	0.0	20	49673	116.0	170.2	0.0	1808	1799	3560.3	TL	0.93	88	34	83.1	130.3	0.0
300	5	27.69	337.0	0.0	6	179400	214.6	514.1	0.0	6	89741	187.4	265.8	0.0	4	2	15.7	15.8	0.0	8	3	27.6	43.0	0.0
	10	28.13	477.3	0.0	10	179514	214.6	568.7	0.0	8	89842	213.8	310.9	0.0	14	4	29.2	29.3	0.0	14	6	51.0	77.9	0.0
	15	28.53	621.7	0.0	10	179612	214.9	568.9	0.0	10	90232	241.2	358.8	0.0	33	4	31.3	31.4	0.0	12	5	43.6	66.8	0.0
	20	28.88	739.2	0.0	18	179958	214.9	675.6	0.0	18	91017	348.6	546.9	0.0	159	23	162.7	162.9	0.0	20	7	70.4	108.7	0.0
	30	29.50	1043.7	0.0	16	181236	215.4	653.0	0.0	20	92287	378.4	596.9	0.0	520	119	799.8	800.9	0.0	26	9	89.9	139.8	0.0
	40	30.03	1171.6	0.0	22	182898	219.7	754.6	0.0	22	94612	408.1	650.6	0.0	1815	543	3595.8	TL	0.10	54	20	182.3	286.2	0.0
	50	30.52	987.6	0.0	18	183176	219.6	709.1	0.0	20	95579	380.1	602.4	0.0	1810	545	3597.6	TL	0.22	46	16	152.5	240.8	0.0
	75	31.53	1003.5	0.0	28	187311	225.4	882.5	0.0	26	100699	468.2	760.0	0.0	1803	461	3041.6	TL	0.40	88	33	298.9	468.5	0.0
	100	32.33	840.9	0.0	22	190171	227.8	806.2	0.0	22	103535	419.0	663.9	0.0	1808	542	3595.7	TL	0.48	82	33	287.5	445.4	0.0

Table 3: Summary of computational results on small instances.

Parameter	CPLEX	BD-multi	BD-multi-Pareto	BD-single	BD-single-Pareto
Avg. # Iterartions	–	18.8	17.4	1987.0	56.8
Avg. # Cuts	–	95365.3	49849.3	1770.9	22.4
Avg. Subp. Sol. Time (s)	–	94.9	152.4	1505.3	74.2
Avg. Total Sol. Time (s)	327.6	288.7	233.9	1622.0	116.2

Table 4: Performance comparison on large instances.

Problem			CPLEX		BD-multi-Pareto					BD-single-Pareto				
n	m	Opt. Sol. (10^6)	Total Sol. Time (s)	Gap (%)	# Iterations	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)	Gap (%)	# Iterations	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)	Gap (%)
	5	49.12	949.7	0.0	6	159647	450.6	746.0	0.0	6	2	50.0	78.7	0.0
	10	49.70	1138.9	0.0	8	160046	520.6	899.9	0.0	8	3	70.1	108.6	0.0
	15	50.18	1649.7	0.0	12	160387	646.3	1197.0	0.0	10	4	90.6	139.4	0.0
	20	50.63	1990.3	0.0	10	160977	583.2	1044.1	0.0	10	4	100.6	148.4	0.0
400	30	51.42	3088.5	0.0	20	162994	917.8	1818.8	0.0	30	11	257.5	401.5	0.0
	40	52.15	3015.3	0.0	22	164351	976.2	1948.7	0.0	34	12	285.4	448.3	0.0
	50	52.83	TL	N/A*	24	166024	1036.9	2109.6	0.0	52	18	433.9	685.0	0.0
	75	54.32	3203.5	0.0	16	169745	796.2	1526.0	0.0	40	14	335.2	526.6	0.0
	100	55.55	2942.1	0.0	28	174626	1181.6	2448.0	0.0	120	48	1086.5	1659.1	0.0
	5	76.58	2018.0	0.0			Memory		N/A†	6	2	98.3	154.2	0.0
	10	77.39	3156.3	0.0			Memory		N/A†	6	2	100.2	156.8	0.0
	15	78.08	TL	N/A*			Memory		N/A†	10	4	178.2	272.2	0.0
	20	78.68	TL	N/A*			Memory		N/A†	16	6	279.6	431.2	0.0
500	30	79.75	TL	N/A*			Memory		N/A†	28	9	449.7	714.2	0.0
	40	80.70	TL	N/A*			Memory		N/A†	32	11	545.2	847.8	0.0
	50	81.56	TL	N/A*			Memory		N/A†	22	8	393.1	601.2	0.0
	75	83.42	TL	N/A*			Memory		N/A†	36	12	611.1	949.9	0.0
	100	85.02	TL	N/A*			Memory		N/A†	32	12	553.8	858.0	0.0
	5	110.09	TL	N/A*			Memory		N/A†	6	2	173.8	272.5	0.0
	10	110.96	TL	N/A*			Memory		N/A†	6	2	173.6	272.0	0.0
	15	111.74	TL	N/A*			Memory		N/A†	10	4	311.2	475.6	0.0
	20	112.45	TL	N/A*			Memory		N/A†	12	5	419.2	615.0	0.0
600	30	113.74	TL	N/A*			Memory		N/A†	16	6	482.8	745.8	0.0
	40	114.92	TL	N/A*			Memory		N/A†	14	5	414.5	644.7	0.0
	50	116.00	TL	N/A*			Memory		N/A†	18	6	513.1	807.9	0.0
	75	118.33	TL	N/A*			Memory		N/A†	30	10	857.2	1351.0	0.0
	100	120.38	TL	N/A*			Memory		N/A†	36	12	1023.0	1615.7	0.0
	5	149.69	TL	N/A*			Memory		N/A†	6	2	280.9	440.7	0.0
	10	150.75	TL	N/A*			Memory		N/A†	6	2	280.6	438.9	0.0
	15	151.64	TL	N/A*			Memory		N/A†	8	3	392.6	603.6	0.0
	20	152.45	TL	N/A*			Memory		N/A†	10	4	502.2	764.7	0.0

* The gap is not available because the root node relaxation cannot be solved.
† The gap is not available because of the memory limit.

(Continued on next page)

Table 4 – Performance comparison on large instances (continued)

Problem			CPLEX		Pareto multi-cut					Pareto single-cut				
n	m	Opt. Sol. (10^6)	Total Sol. Time (s)	Gap (%)	# Iterations	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)	Gap (%)	# Iterations	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)	Gap (%)
700	30	153.95	TL	N/A*			Memory		N/A†	12	5	612.0	928.6	0.0
	40	155.34	TL	N/A*			Memory		N/A†	14	5	667.2	1037.8	0.0
	50	156.60	TL	N/A*			Memory		N/A†	16	7	900.8	1327.5	0.0
	75	159.42	TL	N/A*			Memory		N/A†	28	10	1338.8	2086.8	0.0
	100	161.95	TL	N/A*			Memory		N/A†	30	9	1319.6	2116.3	0.0
800	5	195.35	TL	N/A*			Memory		N/A†	6	2	433.5	677.4	0.0
	10	196.56	TL	N/A*			Memory		N/A†	6	2	436.4	681.9	0.0
	15	197.63	TL	N/A*			Memory		N/A†	6	2	434.5	678.4	0.0
	20	198.61	TL	N/A*			Memory		N/A†	14	6	1124.1	1693.4	0.0
	30	200.40	TL	N/A*			Memory		N/A†	10	4	781.6	1194.7	0.0
	40	202.02	TL	N/A*			Memory		N/A†	18	8	1572.9	2309.9	0.0
	50	203.43	TL	N/A*			Memory		N/A†	16	6	1228.4	1895.0	0.0
	75	206.63	TL	N/A*			Memory		N/A†	28	9	2080.5	3247.3	0.0
	100	209.48	TL	N/A*			Memory		N/A†	14	6	1132.8	1715.8	0.0
900	5	246.65	TL	N/A*			Memory		N/A†	6	2	620.8	967.3	0.0
	10	247.91	TL	N/A*			Memory		N/A†	8	3	879.4	1348.5	0.0
	15	249.03	TL	N/A*			Memory		N/A†	8	3	870.1	1335.1	0.0
	20	250.10	TL	N/A*			Memory		N/A†	12	5	1366.3	2082.8	0.0
	30	252.08	TL	N/A*			Memory		N/A†	10	4	1118.0	1709.4	0.0
	40	253.92	TL	N/A*			Memory		N/A†	10	4	1117.8	1713.4	0.0
	50	255.65	TL	N/A*			Memory		N/A†	8	3	871.2	1342.8	0.0
	75	259.42	TL	N/A*			Memory		N/A†	22	8	TL	TL	0.01
	100	262.84	TL	N/A*			Memory		N/A†	20	7	TL	TL	0.01
1000	5	304.36	TL	N/A*			Memory		N/A†	6	2	888.7	1387.7	0.0
	10	305.75	TL	N/A*			Memory		N/A†	6	2	884.5	1388.2	0.0
	15	307.04	TL	N/A*			Memory		N/A†	8	3	1239.6	1915.0	0.0
	20	308.22	TL	N/A*			Memory		N/A†	10	4	1614.8	2463.6	0.0
	30	310.41	TL	N/A*			Memory		N/A†	10	4	1597.6	2460.0	0.0
	40	312.44	TL	N/A*			Memory		N/A†	10	4	1582.8	2420.0	0.0
	50	314.31	TL	N/A*			Memory		N/A†	16	7	2648.1	3451.6	0.0
	75	318.57	TL	N/A*			Memory		N/A†	28	10	TL	TL	0.01
	100	322.38	TL	N/A*			Memory		N/A†	24	9	TL	TL	0.01

* The gap is not available because the root node relaxation cannot be solved.

† The gap is not available because of the memory limit.

5. Chicago Case Study

In this section, we present the results of a case study on a real-world Chicago network. We use publicly available census data from 2010 for population counts, commuting patterns using personal vehicles and OD pair information (Dash Nelson and Rae, 2016), and data from Chicago Transit

Table 5: Chicago dataset characteristics.

Population (million)	# Nodes	# OD pairs	Node Density (#/km^2)	Distance (km) minimum	Distance (km) maximum
2.701	797	35501	1.353	0.19	62.1

Authority for public transportation commuting counts (City of Chicago, 2020). The network representing this dataset is shown in Figure 2 and its characteristics are summarized in Table 5. The network contains 2.701 million customers aggregated into 797 nodes, 35501 OD pairs with a node density of 1.353 nodes/km^2. The minimum and maximum direct distance between a pair of nodes is 0.19 and 52.1 km, respectively. We consider coverage distance $d \in \{1, 2, 3\}$. We use Manhattan distance due to the grid structure of the road network of the city of Chicago. We consider driving deviation tolerances $\lambda \in \{10\%, 30\%, 50\%\}$. Availability time, maximum availability time and numbers of potential locations are selected similar to the random dataset. We consider personal vehicle commuting counts, representing 34% of the total customer demand, as the demand that can be captured on OD pairs or at origin and destination, the public transportation counts, representing 45% of the total demand, as the demand that can be captured at the origin and destination, and the rest 21% of the population count as demand that can only covered at the origin (Dash Nelson and Rae, 2016).

5.1. Computational Performance

We use BD-single-Pareto implementation as the solution method and present the computational efficiency on solving the problem instances of the Chicago case study. Table 6 presents the computational results on instances with varying number of facility locations m and coverage distances d. The driver tolerance λ has an insignificant impact on the performance of the algorithm. Therefore, we report the average performance of $\lambda \in \{10\%, 30\%, 50\%\}$ in Table 6. The two leftmost columns show the coverage distance d and the number of facilities m, columns 3$-$6 display the number of subproblems, the number of cuts added, the total subproblem solution time and the total solution time. Although the total solution time increases significantly for larger d and m values, the BD-single-Pareto implementation solves all the instances within the one-hour time limit.

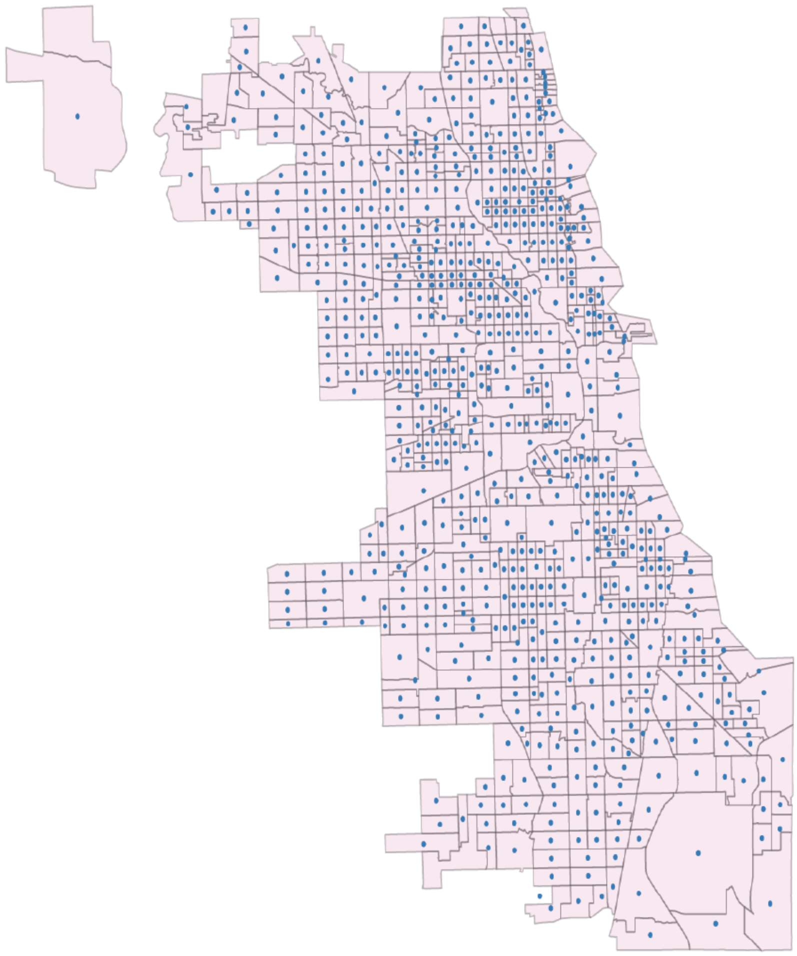


Figure 2: Chicago network and 797 nodes representing the census tracts (City of Chicago, 2018).

5.2. Discussion

In this section, we present the results of the Chicago dataset, compare the impacts of different settings on the demand coverage. We first investigate the change in coverage with the number of facilities m. Figure 3 represents the percentage of coverage for increasing number of facilities in the city of Chicago. Each red dot represents an open facility and the highlighted areas on the maps represent the areas covered. Areas that are highlighted but do not include any facilities at their center are areas housing customers that can be covered along their OD paths. As can be seen from the figure, increasing the number of facilities, expands the coverage significantly. We can only cover 32.08% with 20 facilities but more than 75% with 80 open facilities. Figures 4(a) and (b) confirm the findings that the demand coverage increases with greater number of facilities m regardless of the coverage distance d and driving tolerance λ. Figure 4(a) also shows that we are able to cover

Table 6: Performance of BD-single-Pareto on Chicago dataset.

d	m	# Subproblems	# Cuts	Subp. Sol. Time (s)	Total Sol. Time (s)
1	10	40	14	5.9	12.0
	20	116	48	18.9	36.2
	40	324	188	68.1	119.4
	60	774	336	114.8	347.2
	80	1796	874	319.7	1372.2
	100	2442	1196	438.2	2944.2
3	10	46	18	7.3	14.2
	20	121	52	22.8	56.5
	40	386	212	86.4	185.6
	60	806	362	128.6	393.9
	80	1944	928	362.1	1716.9
	100	2662	1232	473.8	3103.6
5	10	78	32	15.3	34.0
	20	278	208	78.5	155.9
	40	604	296	178.3	795.4
	60	1216	572	285.6	1003.5
	80	2087	1003	402.9	2093.0
	100	2940	1445	512.8	3520.0

93% of the demand with 40 facilities when $d = 3$ but only 49% with the same number of facilities with $d = 1$. We can also cover nearly the same percentage of the population with 20 facilities with $d = 3$ (73%) as 80 facilities with $d = 1$ (75%). These results show that the coverage distance d has a bigger impact on the percentage of demand served compared to the number of facilities m, and increasing the coverage distance of facilities allows us to cover more customer demand compared to increasing the number of facilities m. The coverage distance also affects where we capture the demand, as can be seen from Figure 5(a). Increasing the value of d means that we are able to cover more demand around origin and destination nodes rather than on OD paths.

We have also tested the impact of the driver tolerance λ on the demand coverage. Figure 4(b) shows that increasing the driving tolerance positively impacts the percentage of covered demand.

Nevertheless, the increase is not as significant as the one induced by larger coverage distances. On average the number of facilities decreases by 20 from $\lambda = 0\%$ to $\lambda = 50\%$ for the same percentage of coverage, while the number of facilities decreases by 60 from $d = 1$ to $d = 5$. The driving tolerance impacts where the demand is captured. Figure 5(b) shows that for the instance with $d = 1$ and $m = 60$, the higher the value of λ the more demand is captured on paths. The driver tolerance can have more impact in less dense and more sparse areas where personal vehicles are more widely used.

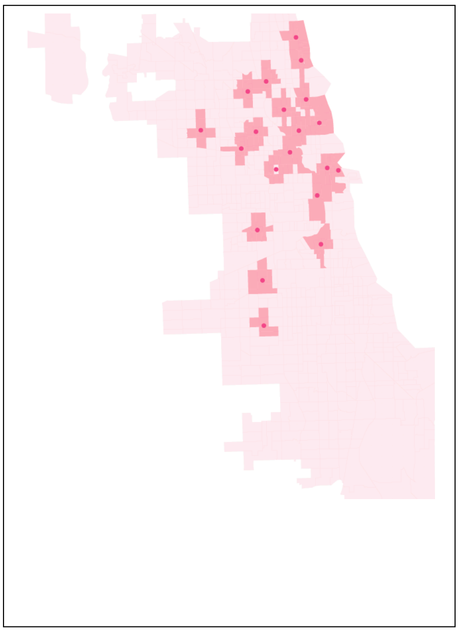
An important application of the model presented is the location of testing facilities during a pandemic as discussed in Section 1. In the following, we present the impact of a confinement during a pandemic on the demand coverage. To simulate a pandemic situation, we create three different datasets from the Chicago dataset. We use public transportation total counts during the period of April 2020, June 2020, and July 2020 as a representation of the drop in commuting counts during the different stages of the pandemic and project them on the rest of the data. The 3 months represent three different levels of confinement: high, moderate, and low, and each level is associated with a confinement percentage representing the decrease in travel. The high level represents an 80% decrease in travel, the moderate level a 60% decrease and the low level a 40% decrease. We refer to the levels as level 0 for no travel restrictions, and level 1, 2, and 3 for the low, moderate and high confinement stages, respectively.

Figure 4(c) shows that the percentage of coverage is not affected by a confinement situation, we can still cover approximately the same percentage of the population regardless of the travel restrictions. The way in which the demand is captured changes drastically, Figure 5(c) shows that higher levels of confinement signify capturing more demand around origins rather than around destinations or on OD paths. This result is in accordance with a real-life pandemic situation, as less people are able to travel and work from home is more prevalent. We also investigate the effects of existing facility set-ups on coverage. We measure the percentage of coverage change of a facility set-up optimized for level u compared to a facility set-up optimized for level v, we refer to this change as the cost of immobility (CI). $\mathrm{CI}(u, v)$ represents the loss of coverage of a set-up optimized using data from level u applied to data from level v. For instance, the facility set-up illustrated in Figure 3(a) optimized with data from level 0, applied to data from level 3 induces an 8.1% loss in coverage. To define $\mathrm{CI}(u, v)$, let $z_v(u)$ the objective function value of the optimal solution of level

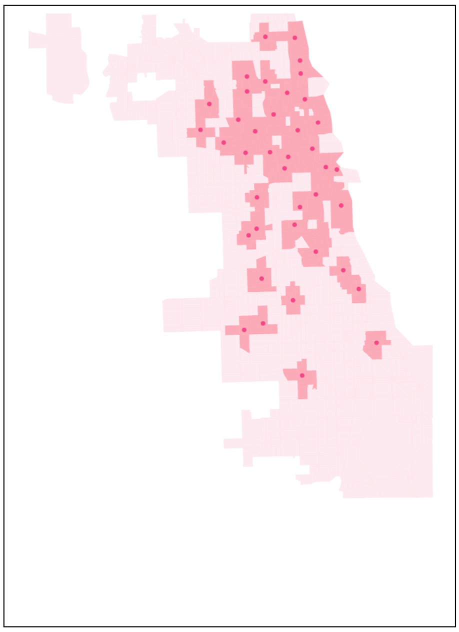
v evaluated using the demand data from level u. The cost of immobility is then defined as follows:

$$CI(u,v) = \frac{z_v(u) - z_u(u)}{z_v(v)}.$$

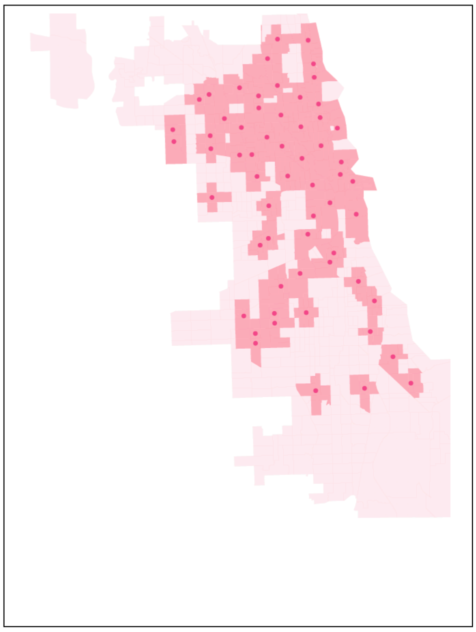
Figures 6 present the cost of immobility matrices for all levels of confinement and different number of facilities. Each matrix entry contains the cost of immobility induced by the transition from the optimization level to the current level. For instance, a set-up with 20 facilities optimized with data from level 0 costs up to 8% in coverage when level 3 travel restrictions come into effect. As can be seen from Figure 6, the cost of immobility from level 0 to 3 (CI(0,3)) is higher than the cost of immobility from level 3 to level 0 (CI(3,0)) (8.1>4.2). For $m = 20$, on average, the cost of immobility when phasing out confinement is 2.46%, while the cost of immobility when entering confinement is 4.55%. Similar findings are true for different m values. This signifies that the optimization is more effective when confinement and travel restriction data is taken into account. Figures 6 also show that the cost of immobility decreases with a higher number of facilities. For example, CI(0,3) $= -8.1\%$ with $m = 20$ but only -2.3% with $m = 80$. The experiments mark the importance of mobile facilities for providing testing services during such rapidly changing environments as pandemics. Indeed, buses were converted into mobile testing facilities in Montreal during the Covid-19 pandemic (Lalonde, 2020).



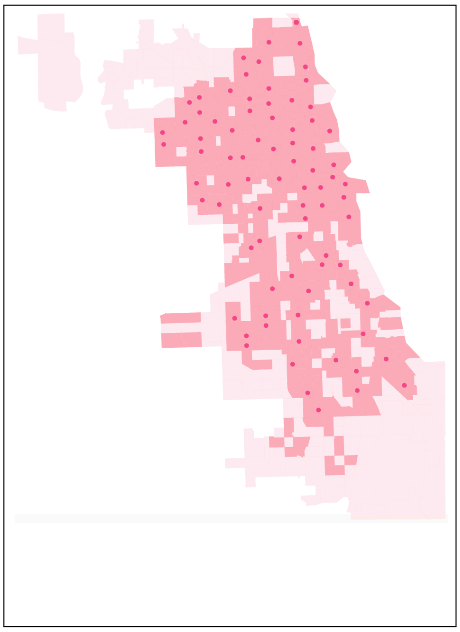
(a) Facilities = 20, Coverage = 32.08%



(b) Facilities = 40, Coverage = 49.70%

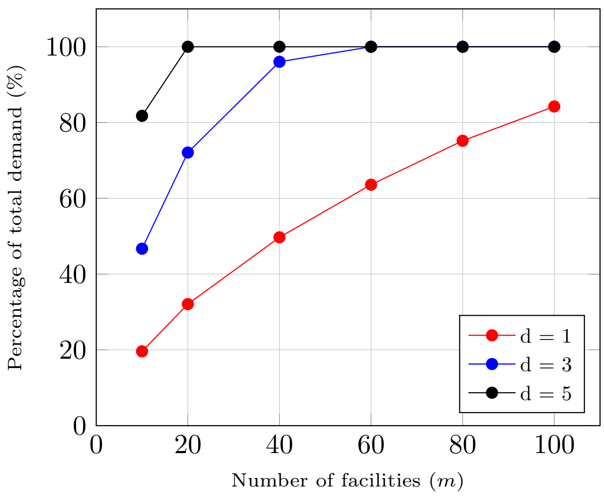


(c) Facilities = 60, Coverage = 63.59%

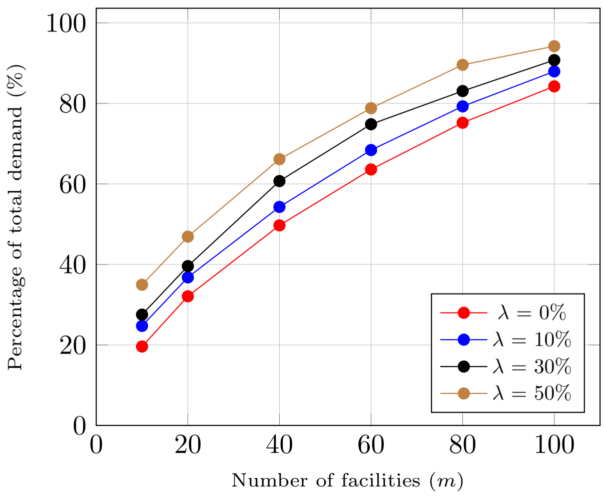


(d) Facilities = 80, Coverage = 75.19%

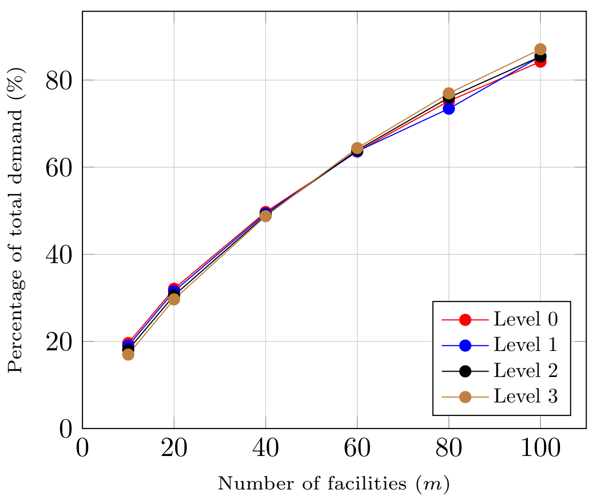
Figure 3: Facility locations and their coverage for Chicago network for level 0, $d = 1$ and $\lambda = 0\%$ with $m \in \{20, 40, 60, 100\}$ in figures (a), (b), (c), and (d), respectively.



(a) Percentage of total demand covered for Level 0 and $\lambda = 0$

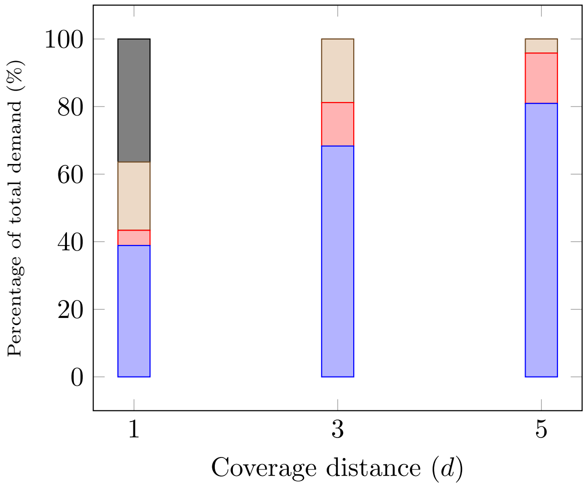


(b) Percentage of total demand covered for Level 0 and $d = 1$

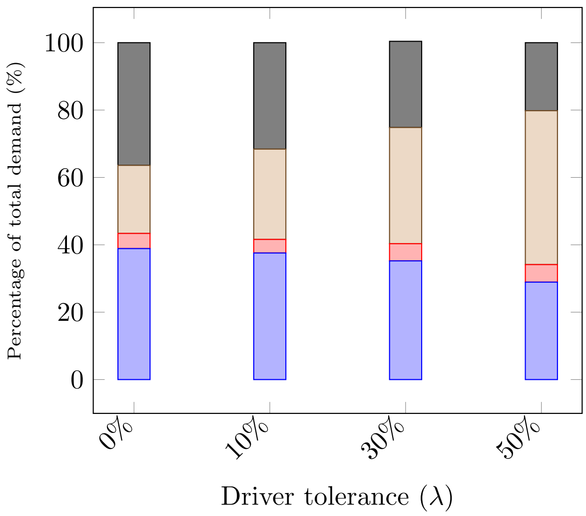


(c) Percentage of total demand covered for $\lambda = 0\%$ and $d = 1$

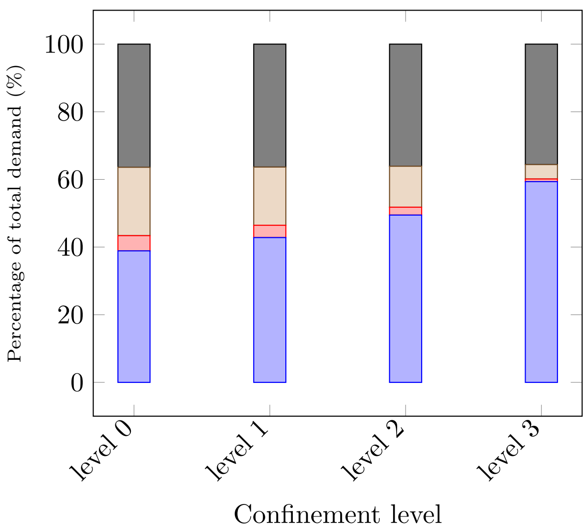
Figure 4: Percentage of demand coverage with $m \in \{10, 20, 40, 60, 80, 100\}$ for different d, λ , and confinement level in figures (a), (b), and (c), respectively.



(a) Type of coverage for $m = 60$, Level 0, and $\lambda = 0\%$



(b) Type of coverage for $m = 60$, Level 0, and $d = 1$



(c) Type of coverage for $m = 60$, $\lambda = 0\%$, and $d = 1$

Coverage at Origin Coverage at destinantion Coverage on path Not covered

Figure 5: Type of coverage for varying d, λ, and confinement level in figures (a), (b), and (c), respectively.

Optimization level		Current level 0	1	2	3
	0	0	-3.1	-5.7	-8.1
	1	-1.7	0	-2.9	-4.8
	2	-2.8	-2.1	0	-2.7
	3	-4.2	-2.9	-1.1	0

(a) CI (%) with $m = 20$

Optimization level		Current level 0	1	2	3
	0	0	-2.3	-4.1	-6.9
	1	-1.1	0	-2.0	-3.9
	2	-2.1	-1.7	0	-1.5
	3	-3.8	-1.9	-0.9	0

(b) CI (%) with $m = 40$

Optimization level		Current level 0	1	2	3
	0	0	-1.3	-2.7	-4.1
	1	-0.8	0	-1.4	-3.1
	2	-1.7	-1.2	0	-1.1
	3	-3.0	-1.2	-0.6	0

(c) CI (%) with $m = 60$

Optimization level		Current level 0	1	2	3
	0	0	-1.0	-2.1	-2.9
	1	-0.5	0	-1.0	-1.7
	2	-1.1	-0.8	0	-1.1
	3	-2.3	-0.8	-0.3	0

(d) CI (%) with $m = 80$

Figure 6: Cost of immobility (CI) for all levels of confinement with m = 20, 40, 60, 80 facilities, in figures (a), (b), (c), (d), respectively.

6. Conclusion

We have presented the maximum availability service facility location problem, in which facility locations are optimized by taking into account the stationary and mobile demand in an urban region. Stationary customers are covered at their origins, the population using the public transportation are covered at their origins or destinations, and personal vehicle users are covered at their origins, destinations or on their commute paths. This problem has applications in location of government offices, medical facilities for testing and vaccination purposes, or polling stations. We have presented a MILP formulation to the problem and have developed a Benders decomposition algorithm. We have proposed an analytical solution to the BD subproblems as well as four different cut implementations to solve problem instances: single-cut, multi-cut, Pareto-optimal single-cut and Pareto-optimal multi-cut. We have conducted extensive experiments and showed on randomly generated datasets. CPLEX without any decomposition is unable to solve large instances of the problem within a one-hour time limit and the Pareto multiple cut implementation struggles due to memory limits. Our Pareto-optimal single-cut implementation of the BD algorithm, on the other hand, performs better than all other implementations and solves large scale instances efficiently. We also have conducted a case study on the city of Chicago and simulated a confinement situation during a pandemic. We have found that with our model, we are capable of capturing the similar demand volumes for all levels of confinement and that locating facilities is more efficient when travel restrictions are taken into account during the optimization.

For future research directions, facility capacities, resource constraints, and a time dimension can be added to the model to make the problem more realistic. Adding uncertainty for the travel flows and demand would make the problem more applicable in the real-world. A robust optimization approach can also help in providing a minimum level of coverage during different phases of confinement for fixed facilities.

References

Arslan, O., Jabali, O., Laporte, G., 2018. Exact solution of the evasive flow capturing problem. Operations Research 66, 1625–1640.

Arslan, O., Karaşan, O.E., 2016. A Benders decomposition approach for the charging station location problem with plug-in hybrid electric vehicles. Transportation Research Part B: Methodological 93, 670–695.

Arslan, O., Karaşan, O.E., Mahjoub, A.R., Yaman, H., 2019. A branch-and-cut algorithm for the alternative fuel refueling station location problem with routing. Transportation Science 53, 1107–1125.

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

Berman, O., Bertsimas, D., Larson, R.C., 1995. Locating discretionary service facilities, II: Maximizing market size, minimizing inconvenience. Operations Research 43, 623–632.

Berman, O., Larson, R., Fouska, N., 1992. Optimal location of discretionary service facilities. Transportation Science 26, 161–260.

Chauhan, D., Unnikrishnan, A., Figliozzi, M., 2019. Maximum coverage capacitated facility location problem with range constrained drones. Transportation Research Part C: Emerging Technologies 99, 1–18.

City of Chicago, 2018. Boundaries - census tracts - 2010. URL: `https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Census-Tracts-2010/5jrd-6zik`. accessed: 2020-08-30.

City of Chicago, 2020. CTA - ridership - 'L' station entries - daily totals. URL: `https://data.cityofchicago.org/Transportation/CTA-Ridership-L-Station-Entries-Daily-Totals/5neh-572f`. accessed: 2020-08-30.

Codato, G., Fischetti, M., 2006. Combinatorial Benders cuts for mixed-integer linear programming. Operations Research 54, 756–766.

Cordeau, J.-F., Furini, F., Ljubić, I., 2019. Benders decomposition for very large scale partial set covering and maximal covering location problems. European Journal of Operational Research 275, 882–896.

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

Dash Nelson, G., Rae, A., 2016. An economic geography of the United States: From commutes to megaregions. PLOS ONE 11, 1–23.

de Camargo, R., Miranda, G., Luna, H., 2008. Benders decomposition for the uncapacitated multiple allocation hub location problem. Computers & Operations Research 35, 1047–1064.

Erkut, E., Ingolfsson, A., Erdoğan, G., 2008. Ambulance location for maximum survival. Naval Research Logistics 55, 42–58.

García, S., Marín, A., 2019. Covering location problems, in: Laporte, G., Nickel, S., Saldanha da Gama, F. (Eds.), Location Science. Springer. chapter 5, pp. 99–119.

Hodgson, M.J., 1990. A flow-capturing location-allocation model. Geographical Analysis 22, 270–279.

Hodgson, M.J., Rosing, K.E., Leontien, A., Storrier, G., 1996. Applying the flow-capturing location-allocation model to an authentic network: Edmonton, Canada. European Journal of Operational Research 90, 427–443.

Kim, J.G., Kuby, M., 2012. The deviation-flow refueling location model for optimizing a network of refueling stations. International Journal of Hydrogen Energy 37, 5406–5420.

Kuby, M., Lim, S., 2005. The flow-refueling location problem for alternative-fuel vehicles. Socio-Economic Planning Sciences 39, 125–145.

Lalonde, M., 2020. Montreal city buses to be used as mobile testing clinics. URL: `https://montrealgazette.com/news/montreal-reports-78-new-covid-19-deaths-385-new-cases`. accessed: 2020-08-30.

Magnanti, T.L., Wong, R.T., 1981. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. Operations Research 29, 464–484.

Marković, N., Ryzhov, I.O., Schonfeld, P., 2015. Evasive flow capture: Optimal location of weigh-in-motion systems, tollbooths, and security checkpoints. Networks 65, 22–42.

Murali, P., Ordóñez, F., Dessouky, M.M., 2012. Facility location under demand uncertainty: Response to a large-scale bio-terror attack. Socio-Economic Planning Sciences 46, 78–87.

Ndiaye, M., Alfares, H., 2008. Modeling health care facility location for moving population groups. Computers & Operations Research 35, 2154–2161.

Papadakos, N., 2008. Practical enhancements to the Magnanti–Wong method. Operations Research Letters 36, 444–449.

Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2017. The Benders decomposition algorithm: A literature review. European Journal of Operational Research 259, 801–817.

Snyder, L.V., 2011. Covering problems, in: Eiselt, H.A., Marianov, V. (Eds.), Foundations of Location Analysis. Springer. chapter 6, pp. 109–135.

Taymaz, S., Iyigun, C., Bayindir, Z., Dellaert, N., 2020. A healthcare facility location problem for a multi-disease, multi-service environment under risk aversion. Socio-Economic Planning Sciences 71, 100755.

Turkoglu, D.C., Genevois, M.E., 2020. A comparative survey of service facility location problems. Annals of Operations Research 292, 399–468.

Upchurch, C., Kuby, M., Lim, S., 2009. A model for location of capacitated alternative-fuel stations. Geographical Analysis 41, 85–106.

Yıldız, B., Arslan, O., Karaşan, O.E., 2016. A branch and price approach for routing and refueling station location model. European Journal of Operational Research 248, 815–826.

You, F., Grossmann, I.E., 2013. Multicut Benders decomposition algorithm for process supply chain planning under uncertainty. Annals of Operations Research 210, 191–211.

Zetina, C.A., Contreras, I., Cordeau, J.-F., 2019. Exact algorithms based on Benders decomposition for multicommodity uncapacitated fixed-charge network design. Computers & Operations Research 111, 311–324.