Université de Montréal

Algorithmic contributions to bilevel location problems with queueing and user equilibrium : exact and semi-exact approaches

par

Teodora Dan

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures et postdoctorales
en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en Informatique

31 Août 2018

Résumé

Bien que la littérature sur le problème d'emplacement soit vaste, la plupart des publications considèrent des modèles simples, dans lesquels une autorité centrale assigne les utilisateurs aux installations les plus proches. Des caractéristiques plus réalistes, telles que le comportement des usagers, la compétition et la congestion, sont souvent négligées, peut-être en raison de leur nature hautement non-linéaire «compliquée». Quelques articles ont incorporé ces traits, mais uniquement de facon séparée, et seulement des approches heuristiques ont été proposées comme méthodes de résolution.

Le problème d'emplacement d'installations consiste à localiser un ensemble d'installations de manière optimale afin de répondre à une demande donnée. Dans un environnement congestioné où les usagers ont le choix, les installations sont généralement modélisées sous la forme de files d'attente. Les utilisateurs sélectionnent les installations à fréquenter en fonction de leur utilité perçue, qui est généralement écrite comme une combinaison linéaire de la distance de déplacement, du temps d'attente dans les installations, etc. En résulte un modèle dit "à deux niveaux" appartenant à la classe des programmes mathématiques à contraintes d'équilibre (MPEC en anglais), où l'équilibre peut être exprimé sous la forme d'une inéquation variationnelle.

Notre travail est axé sur le problème d'emplacement d'installations où les usagers ont le choix (CC–FLP en anglais) et nous fournissons un certain nombre de contributions importantes. Du point de vue de la modélisation, nous proposons différents modèles qui capturent les principales caractéristiques du CC–FLP. Pour ces programmes non-linéaires, discrets, et NP-difficiles, nous avons conçu des algorithmes exactes et d'approximation, ainsi que des heuristiques sur-mesure. Notre travail couvre trois articles. Dans le premier article, nous considérons différents modèles qui intègrent l'abandon aux centres de services, en raison des places limitées dans la file d'attente, tandis que le comportement des utilisateurs peut être

déterministe ou stochastique. Dans ce dernier cas, le comportement des usagers correspond au principe d'équilibre de Wardrop, tandis que dans le premier cas, les clients se distribuent entre les établissements selon un modèle de choix d'utilité aléatoire Logit. Au-delà de l'analyse des propriétés théoriques du modèle, nous concevons une heuristique menée par les usagers et un algorithme d'approximation linéaire pour lequel nous prouvons une borne d'erreur de l'approximation, dans le cas d'une file d'attente $M/M/1$.

Le second article est consacré à la conception d'un nouvel algorithme de 'Branch and Bound' (B&B) pour résoudre une sous-classe plus générale des MPEC. L'algorithme est implémenté et évalué sur un CC–FLP. L'idée est de traiter virtuellement chaque nœud de l'arbre B& B comme un problème d'optimisation distinct, afin de tirer parti de la puissance des solveurs MILP et de leur prétraitement fort au niveau de la racine. Notre approche algorithmique est basée sur une combinaison de programmation linéaire à nombres entiers et mixtes (MILP en anglais), de techniques de linéarisation et de la resolution itérative de sous-problèmes convexes, et nécessite une gestion d'arbre sophistiquée.

Dans le troisième article, nous incorporons les prix dans le CC–FLP. Le prix est une variable de décision continue, tout comme la localisation et le niveaux et de service, et les utilisateurs l'intègrent dans leur utilité. Les concepts de tarification du réseaux et de CC–FLP étant fusionnés en un seul modèle, le problème devient extrêmement difficile, également en raison de la présence de variables de localisation et de niveau de service, ainsi que de délais d'attente bidimensionnels. Pour ce programme à deux niveaux non-convexe, nous avons conçu un algorithme basé sur des approximations linéaires emprunté à la fois à la littérature sur la localisation et à la tarification du réseau.

Mots clés : probleme d'emplacement d'installations, programmation à deux niveaux, programmation à nombres entiers et mixtes, équilibre, file d'attente, non convexe, optimisation globale, tarification.

Abstract

While the location literature is vast, most papers consider simpler models, in which a central authority assigns users to the closest facilities. More realistic traits, such as user behaviour, competition, and congestion are often overlooked, perhaps due to their 'complicating' highly non-linear nature. A few papers did incorporate them, but separately, and only heuristic approaches have been proposed as solution methods.

The facility location problem consists in optimally locating a set of facilities in order to satisfy a given demand. In a congested user-choice environment, facilities are typically modeled as queues, and users select the facilities to patronize based on their perceived utility, which is, in general, written as linear combination of travel distance, waiting time at facilities, etc. The resulting bilevel model belongs to the class of mathematical programs with equilibrium constraints (MPECs), where the equilibrium can be expressed as a variational inequality.

Our work is focused on the *competitive congested user-choice facility location problem* (CC–FLP), and we provide a number of strong contributions. From the modeling point of view, we propose various models that capture the key features of CC–FLP. For these NP-hard discrete nonlinear programs we designed exact and approximated algorithms, as well as tailored heuristics. Our work spans three papers. In the first article we consider different models that incorporate balking at facilities, due to limited places in the queue, while user behaviour can be either deterministic or stochastic. In the latter case, user behaviour fits Wardrop's equilibrium principle, while in the former case, customers distribute among facilities according to a Logit random utility choice model. Beyond the analysis of the model's theoretical properties, we design a user-driven heuristic and a linear approximation algorithm, for which we prove a bound on the approximation error, for the $M/M/1$ queue.

The second paper is dedicated to the design of a novel exact branch-and-bound (B&B) algorithm for solving a more general subclass of MPECs, which is implemented and evaluated on a CC–FLP. The idea is to virtually treat each node of the B&B tree as a separate optimization problem, in oder to leverage the strength of the MILP solvers and their strong preprocessing at the root node. Our algorithmic approach is based on a combination of Mixed-Integer Linear Programming (MILP), linearization techniques and the iterative solution of convex subproblems, and requires a sophisticated tree management.

In the third paper we incorporate mill pricing into the CC–FLP. Price is a continuous decision variable, along with the location and service levels, and user incorporate it into their utility. Since concepts from network pricing and CC–FLP are merged into a single model, the problem becomes extremely challenging, also due to the presence of facility location and service level decision variables, as well as bivariate queueing delays. For this non-convex bilevel program we devise an algorithm based on linear approximations, that borrows from both location and network pricing literature.

Keywords: location, bilevel programming, mixed integer programming, equilibrium, queueing, nonconvex, global optimization, pricing.

Contents

List of Tables

List of Figures

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Professor Patrice Marcotte, for the opportunity to work with him. I am grateful for his patience, guidance and valuable advice, along the way. I would like to equally thank my co-supervisor, Professor Andrea Lodi, for his help and advice. Without their knowledge and constant support, this thesis would not have been possible.

I want to thank my parents, Daniela and Octavian, and my grandmother, Tudorița, for their love and encouragements throughout my life, and especially during my graduate studies. They have cultivated my interest towards math and science, and for that I am very grateful. Special thanks go to my sister Georgia. She is an inspiration to me, with her critical mind and her loving soul.

Last but not least, I want to thank my dear husband, Laurent Jakubina, for his unconditional love and support, since the day we met.

Chapter 1

Introduction

People have faced the facility location problem since the early days of human civilization, when they decided where to build their households, their villages, etc. A more formal formulation of the location problem was perhaps first introduced by the mathematician Pierre de Fermat (1601 – 1665). In his work entitled '*Methodus ad disquirendam maximam et minima*' he proposed the following puzzle: 'Given three points in a plane, find a fourth point such that the sum of its distances to the three given points is as small as possible.' [**Eiselt and Marianov, 2011**]. The earliest known (geometrical) solution belongs to Fermat's pupil, Evangelista Torricelli (1598 – 1647). Although he found it around 1640, it was published much later, in 1919 in *'Opere di Evangelista Torricelli'*. This puzzle would later constitute the foundation of an entire class of problems, called the **Facility Location Problem** (FLP).

In Operations Research the class of *Facility Location Problems* (FLP) can be stated as: locate one or more facilities within a set of possible sites [**Dantrakul et al., 2014**], under different assumptions, with the goal to serve a set of demand points. The candidate sites can be represented either by a discrete set of points, or by continuous space. In the former case we have *a discrete location problem* whilst in the latter we have a *continuous location problem* [**Dantrakul et al., 2014**]. From a mathematical point of view, FLP is a combinatorial optimization problem that has numerous practical applications such as location of service centers (health clinics, communication centers, banks, etc.), warehouses, production facilities, plants and shops.

FLP has been extensively studied in the literature. The existent formulations cover a plethora of models, from basic to more elaborate models. However, most papers are making

simplifying assumptions. In this thesis we focus on more realistic models, that incorporate competition among several existing firms, congestion and user behaviour.

1.1. FLP: Basic ingredients

Although an FLP can be modelled in many ways, some components are common to all models. The set of locations, the customers, and the facilities represent the basic ingredients for even the simplest formulation [**Azarmand and Neishabouri, 2009**]. While the **location** subproblem corresponds to deciding the sites where the facilities are to be opened, the **allocation** represents the assignment of the customers to their respective facilities. Of course, they are part of a larger problem, in which other parameters are to be decided, for instance the capacity of the facilities, the number of servers in each facility, etc.

Customers are assigned to facilities either by a central authority, or they have the choice themselves. Typically, when the allocation is centralized, the objective function minimizes the total weighted sum of distances, waiting time etc. When the model takes into account users' preferences, a discrete choice model is used, where a random term enters the users' utility function. In the FLP case, the discrete options are associated to the set of open facilities to patronize. Every client must choose one and only one location from the set.

The main elements of the mathematical model are:

— the agents (users, competitors, etc.);

— the interactions between the agents;

— the optimization objectives, goals, etc.

In the case of Facility Location Problem several issues are being raised. From what point of view are we looking at the problem (users, competitors, etc.)? How many facilities are there to be placed in the network? Do facilities have a limited capacity? Are we considering a deterministic or a probabilistic model? Do users have a choice or is there a central coordination? Are there other competitors already in the market? If yes, how do they interact with the entering firm? Do they engage in a sequential or simultaneous game, or do they remain unresponsive to the actions of the newly entered? The answers to all these questions written in a mathematical form, dictate the main characteristics of the model.

The aspects of the problem that can be incorporated are supposedly unlimited, due to a great variety of real-life situations that can be modelled. Nevertheless, we can identify a few features that tend to appear very often in the literature.

— **Capacity**. We define the *capacity* of a facility as the quantity of product or service offered, etc. When this quantity is limited the problem is *capacitated*, as opposed to the *uncapacitated* case where the capacity is 'unlimited'. The capacity can be represented by the number of servers, the service rate, the number of places available in the waiting line, etc.

— **Nature of objective function**. Depending on the agent that is optimizing, and on the real-life context the objective function can have a social interpretation (for instance improve a public service), or it could represent a profit, a total cost, etc.

— **Nature of choice**. We can identify two types of problems depending on who is the decision maker. If the facilities to be patronized are chosen by the users we talk about a *user choice* environment. On the other hand, the choice can be made by a central authority, in which case we have a centralized decision.

— **Randomness**. The model could contain stochastic elements, for instance, the population from a demand node will choose a facility to patronize with a certain probability that depends on numerous factors: distance, price, congestion, etc. In contrast, in simpler models, user choice is deterministic, and the facilities to patronize are always the closest, the ones offering the lowest price, etc.

— **Congestion**. When modelling a real-life situation it is natural to consider that some sort of congestion will occur at the facilities, due to limited resources. Either the quantity of the product is restricted, or users have to wait in a queue in order to get served. Sometimes congestion takes place on the edges of the network due to heavy traffic. This aspect is captured in models by explicitly incorporating users behaviour in queueing equations, captured queueing delay costs, etc.

— **Competition**. Competitors play an important role, as the way they interact with each other can change the nature of the problem. In some formulations they remain unresponsive, whilst in others they can engage into a Stackelberg or Nash game.

The simplest location problem is the uncapacitated FLP, which involves locating an undetermined number of facilities on a network, in order to satisfy the demand

for a certain commodity, while minimizing the total cost (fixed setup costs + travel time) [**Averbakh et al., 2007, Boffey et al., 2007**]. Setup and travel time costs are represented as a function of the number of open facilities as well as their respective locations. While the capacity is considered 'unlimited' or 'infinite', in practice it means that is sufficiently large to satisfy any possible demand [**Boffey et al., 2007**]. For a given set of parameters

d_i: demand originating from user node $i \in I$,

t_{ij}: travel time from user node $i \in I$ to site $j \in J$,

and decision variables

x_{ij}: fraction of individuals originating from node $i \in I$ that choose facility $j \in J$;

y_j: takes the value of 1 if a facility is located at site j, 0 otherwise.

its formulation is:

$$\text{UFLP:} \quad \min_{x,y} \quad \sum_{i\in I}\sum_{j\in J} d_i t_{ij} x_{ij} + \sum_{j\in J} f_j y_j \tag{1.1.1}$$

$$\text{s.t.} \quad \sum_{j\in J} x_{ij} = 1, \qquad \forall i \in I \tag{1.1.2}$$

$$\sum_{i\in I} x_{ij} \leq y_j, \qquad \forall j \in J \tag{1.1.3}$$

$$\sum_{j\in J} y_j = p, \tag{1.1.4}$$

$$x_{ij} \geq 0, \qquad \forall i \in I,\ \forall j \in J \tag{1.1.5}$$

$$y_j \in \{0,1\}, \qquad \forall j \in J. \tag{1.1.6}$$

The objective (1.1.1) is to minimize the total travel time plus the fixed costs associated with locating all facilities. Constraint (1.1.2) guarantees that the demand of any user is met by one or more facilities. Demands can be satisfied (partially or entirely) only by sites where facilities are open (constraint (1.1.3)). Equations (1.1.4) and (1.1.6) are the non-negativity and binary constraints, respectively.

1.2. Congested Facility Location

The Congested Facility Location Problem (CFLP) is a modification of FLP, where congestion is also taken into consideration. We are interested in the CFLP, since it is a more accurate representation of real-life situations.

Congestion occurs naturally in a context of limited resources and capacitated facilities. If too many customers arrive at the same time at a service center, they will have to wait in line in order to be served. Also, if too many cars travel on the same street, traffic congestion becomes inevitable, hence increasing the travel time/costs.

Congestion can either arise at facilities (thus at nodes, in the case of network representation) or on the links between nodes, or on both. Thus, in a network we can distinguish three types of congestion:

— Node congestion (waiting in line to be served, etc.);

— Link congestion (traffic congestion on roads, etc.);

— Network congestion (both node and link congestion).

Depending on the context of the problem, in most cases congestion on links corresponds to traffic in the network, which is generated only partially by the users taken into account in the FLP. For instance, if we were to decide a location for a hospital in a city, the amount of people travelling to the hospital is insignificant compared to the total traffic in the city. Therefore, it is generally considered that congestion on the links is not influenced by users constituting the demand, and it is modelled by predefined constant travel costs. Hence, the underlying network becomes irrelevant, and the problems can be modelled as a bipartite graph. The sets of nodes represent the demand nodes and the facility nodes, respectively. In all our three article we consider node congestion, i.e., facilities are modelled as $M/M/s/K$ queues, or variants thereof. Then, the delay functions are derived from queueing theory (e.g., waiting time, probability of balking, etc.).

An important aspect when modelling congestion regards the allocation problem. The question to be asked is who is the decision maker? Do users decide which facilities to patronize or is there a central authority in charge of the decision? If the decision maker is a central authority, does it protect the users' interest? If yes, typically the the congestion and/or travel costs are minimized [**Desrochers et al., 1995, Marianov, 2003, Castillo et al., 2009, Marianov and Serra, 2011, Vidyarthi and Jayaswal, 2014, Fischetti et al., 2016**].

If users decide, what are the parameters of their decision? What is their purpose and what makes them prefer a specific facility to another one? A widely used assumption is that users select facilities based solely on proximity [**Marianov, 2003, Berman and Drezner, 2006, Abouee-Mehrizi et al., 2011, Marić et al., 2012, Camacho-Vallejo et al., 2014**]. More complex models, incorporate congestion at facilities, modelled as waiting queues.

Based on these aspects we can identify two environments:

— A user-choice environment, typically modelled by lower-level problem containing equilibrium constraints or a Logit function of utility.

— A centralized authority environment can be modelled by both single and bilevel formulations.

1.2.1. User-choice environment

When clients choose the facilities to patronize, we talk about a user-choice environment. The assumption here is that clients are selfish and they minimize their individual disutility, which is typically expressed as a linear combination of different parameters (e.g., distance, congestion, price, etc.). We distinguish here two types of behaviour, namely deterministic and stochastic.

The most common rule when describing the patronizing behaviour is the deterministic one, which we have considered in all our three articles. The disutilities that we have used are, respectively

$$u_{i,j} = \begin{cases} t_{i,j} + \alpha w_j + \beta p_{K,j} & (1^{\text{st}} \text{ article}) \\ t_{i,j} + \alpha w_j & (2^{\text{nd}} \text{ article}) \\ t_{i,j} + \alpha w_j + \beta p_j & (3^{\text{rd}} \text{ article}), \end{cases} \tag{1.2.1}$$

where w_j represents the waiting time at facilities, $p_{K,j}$ the probability of balking, and p_j the price. Each customer will choose the facility that she finds most attractive. In this case, customers are assigned to facilities according to Wardrop's equilibrium principle. According to it, the disutility of unchosen paths (facilities) is higher than the disutility of the chosen ones. Let γ_i denote the minimum disutility for users originating from node i. Then, the optimal solution x^* is characterized by the complementarity system

$$\begin{aligned} u_{i,j}(x^*) &= \gamma_i, \quad \text{if } x^*_{i,j} > 0 \\ u_{i,j}(x^*) &\geq \gamma_i, \quad \text{if } x^*_{i,j} = 0, \end{aligned} \tag{1.2.2}$$

and the users' problem is written as

$$0 \leq x_{i,j} \perp u_{ij} - \gamma_i \geq 0, \quad i \in I;\ j \in J. \tag{1.2.3}$$

Alternatively, we can use vector-matrix notation and write Eq. (1.2.3) as a variational inequality problem. We group $x_{i,j}$ and $u_{i,j}$ into column vectors $x, u \in \mathbb{R}^{|I|\cdot|J|}$, and given upper-level variables y and μ, the variational inequality $VI\left(u(x), X\right)$ is to find $x^* \in X(y, \mu)$, such that

$$\langle u(x^*), x^* - x\rangle \leq 0\ \forall x \in X, \tag{1.2.4}$$

where $X = \left\{x : y_j \geq x_{i,j} \geq 0;\ \sum_{j\in J} = d_i\right\}$. Then, a solution of the variational inequality will correspond to an equilibrium solution.

If u is a gradient, we define the function $U : \mathbb{R} \longrightarrow \mathbb{R}^{|I|\cdot|J|}$, such that $\nabla U(x) = u(x)$. If U is twice continuously differentiable and convex, $VI\left(u(x), X\right)$ can be written as the following convex optimization problem [**Beckmann et al., 1956**]

$$\text{LL1:} \qquad \min_{x} \quad U(x) \tag{1.2.5}$$

$$\text{s.t.} \quad \sum_{j\in J^*} x_{ij} = d_i \qquad \forall i \in I \tag{1.2.6}$$

$$x_{ij} \geq 0. \qquad \forall i \in I; \forall j \in J^* \tag{1.2.7}$$

Some papers consider the choice as probabilistic, and they use a random utility model. In our first article we also consider probabilistic behaviours. In this framework, the utility of facility j for a customer issued from demand node i is given by

$$\tilde{u}_{ij} = -u_{ij} + \varepsilon_{ij}$$

where ε_{ij} are independent Gumbel variates with common scale parameter θ and variance $\pi^2/(6\theta^2)$. In this multinomial logit framework (see [**McFadden, 1974**]), the demand generated at node i that patronize an open facility j is given by the expression

$$x_{ij} = d_i \frac{e^{-\theta u_{i,j}}}{\sum_{l\in J^*} e^{-\theta u_{i,l}}}, \tag{1.2.8}$$

where J^* represents the set of open facilities. For small values of θ, users are spread more or less evenly between facilities while, when θ is large, the assignment approaches that of a Wardrop equilibrium. Similar to its deterministic counterpart, the solution of Eq. (1.2.8) can be obtained by solving the program (see [**Fisk, 1980**]).

$$\text{LL2:} \quad \min_{x,\lambda,\rho,w,p} \quad \sum_{i\in I}\sum_{j\in J^*}\left[\frac{1}{\theta}x_{ij}\ln x_{ij}\right] + U(x)$$

$$\text{s.t.} \quad \sum_{j\in J^*} x_{ij} = d_i, \qquad \forall i \in I$$

$$x_{ij} \geq 0, \qquad \forall i \in I; \forall j \in J^* \ .$$

Due to the presence of the equilibrium constraints, the problem belongs to a larger class of problems, named MPEC (mathematical programs with equilibrium constraints). The original problem can be reformulated as NP-hard bilevel program, where at the upper level we have the firm's optimization problem (e.g., maximizing the market share or profit, minimizing the costs, etc.). and at the lower level we have users' problem (LL1) or (LL2).

1.3. Competitive Facility Location

The *Competitive Facility Location Problem* (ComFLP) can be defined as an FLP that involves several rival firms, already present in, or joining the market in the future. Competitors strive to capture maximum market share, or to maximize their profit. On the other hand, users choose the facility to patronize based on certain parameters: facility attractiveness, price of the commodity, distance from the customer to the facility, etc.

ComFLP involve a number of interacting agents that interact with one another and is a part of Game Theory, which is a branch of mathematics that studies the strategic human behaviour in a competitive situation. The participants involved in the game (i.e. decision-makers) are the *players*. Each player is faced with one or more alternatives or *strategies*. The outcome of the game depends on the decisions of all players, but uncertainty elements can also be present in the game.

Economists have developed a number of models based on different types of behaviour. When only two players compete in the market over a homogeneous product, we have a

duopoly. Three quintessential examples of duopoly models are Cournot-Nash, Bertrand and Stackelberg.

(1) **Cournot-Nash**

If the quantity competition is simultaneous, we have a Cournot-Nash duopoly. The selling price is represented by a decreasing function of the total demand, called the *inverse demand function*, that we note $p(x)$. Let $c_1(x_1)$ and $c_2(x_2)$ be the production costs of the two firms, respectively. Then, the profit function of firm i is:

$$f_i(x_1, x_2) = x_i p(x_2 + x_2) - c_i(x_i).$$

If the objective of every firm is to maximize their profit, then the quantities are given by the Nash equilibrium.

(2) **Bertrand**

Bertrand's model is similar to Cournot-Nash, but the players compete in price, rather than in quantities, and it assumes that clients want to buy from the lowest priced firm. If the firms charge the same price, the demand is split evenly between the two. Then, at equilibrium, prices will equal the marginal cost.

(3) **Stackelberg**

Let us consider the same duopoly scenario, but the companies are moving sequentially rather than simultaneously. Firm 1 chooses the quantity to produce. Next, firm 2 takes note of the quantity produced by firm 1, and produces its quantity accordingly. We call firm 1 the *leader* and firm 2 the *follower*. The price is set once both quantities are decided.

In order to have a Stackelberg equilibrium, some assumptions are being made. First, the follower observes the leader's action, and the leader must know it 'ex ante' . Second, the follower must commit to a future Stackelberg action, and the leader must be aware of it, as well. In practice firms engage into a Stackelberg game only if one has the possibility to move first.

The main difference between a Cournot-Nash or Bertrand, and a Stackelberg model is that in the case of the former two model, no player is at disadvantage due to the simultaneity of the game. In a Stackelberg game, one player must move first, which could be an advantage or disadvantage.

Acording to [**Plastria, 2001**], when modelling a ComFLP, several aspects are taken into account: competition type, features of the market and decision variables.

The simplest type of competition is the *static* one, in which competitors already present in the market are unresponsive to the actions of the newly entered firm, thus their characteristics remain fixed. When the competitors react to the actions of the firm emerging into the market (competition with foresight [**Plastria, 2001**]), we have a sequential model. The firms engage in a Stackelberg game, in which the newcomer plays the role of the leader and the follower is played by the existing firms. A full overview on the sequential models is provided by [**Eiselt and Laporte, 1997**].

Interactions between competitors can also be described by a *dynamic model*, where later players have some knowledge about earlier actions. The challenge of such a model is to find the equilibrium solution, if one exists.

The Competitive Facility Location Problem was first introduced by Hotelling in 1929, when he studied the competition between two firms under inelastic demand [**Hotelling, 1929**]. In his work he considered a very simple scenario: a demand population is uniformly distributed along a line segment, and two firms locate their service points at distances a and b from the two segment ends (A and B), respectively. Customers choose one of the firms based on the sum of commodity price and transportation costs, in an attempt to minimize their total costs. The objective of each firm is to maximize their profit. The decision variables of ComFLP are common to the generic FLP: the number of facilities to locate, as well as their locations, their properties (capacity, service rate, etc.), the objective of the firms, etc. Later, Labbé and Hakimi [**Labbé and Hakimi, 1991**], described a two-stage location–allocation game in which two firms are engaging in a Nash game. The two firms are striving to maximize their profit. First, they decide on location of their facilities, and at the second stage on the quantities to produce. A more recent contribution is proposed by [**Ljubić and Moreno, 2018**] who address a market share-maximization competitive FLP, where captured customer demand is represented by a multinomial logit model. The authors solve this problem using two branch-and-cut techniques, namely outer approximation cuts and submodular cuts.

More elaborate developments of ComFLP have been proposed in the literature. They involve queueing at facilities, and interactions between agents. Most of them are based on

Stackelberg games, and we can identify two main trends, when it comes to who are the players. In one, the game is played between two firms, and they can react to each others' actions, while clients are also taken into account, although their behaviour is modelled by making simplifying assumptions (e.g. the gravity-based rule) [**Küçükaydin et al., 2011, Saidani et al., 2012, Beresnev, 2013**].

Another possibility is that one player is represented by the newly entered firm, and the second one by the users. One or more competitors are also present in the game, but they remain unresponsive to the other firm action [**Marianov et al., 2008**].

To the best of our knowledge, [**Marianov et al., 2008**], are the first to study the CC–FLP. They consider a problem where a new firm is making location decisions in a market where competitors are already operating, but they remain unresponsive to the actions of the newly arrived. The objective of the entering firm is to maximize market share. Users patronize a facility with a probability given by a Logit function of distance and waiting time. Congestion is captured as waiting time in facilities. As the new firm attracts clients, its facilities become more and more congested and the waiting time at its facilities increases, which can deter some customers. An equilibrium is reached when no client has incentive to deviate from her path. Facilities are considered as $M/M/s/K$ queues, which means arrivals and service are Poisson processes with the mean rate λ and μ, respectively. There are s servers available, and the queue length is limited to K customers.

1.4. Pricing

Pricing is a key component in market competition. On the one hand, lowering the prices might attract more customers, but their presence creates congestion, which, finally, might deter customers. On the other hand, a higher price might attract less customers, but the firm could still increase its profit.

Three types of pricing strategies are typically considered in the literature ([**Hanjoul et al., 1990**]):

— mill pricing: prices can vary between facilities;

— uniform pricing: all facilities charge the same price;

— discriminative pricing: customers patronizing the same facility can be charged different prices.

While the location pricing literature is extensive, most papers study hierarchical models in which the locations are decided first, then the price competition is defined by the Bertrand model [**Pérez et al., 2004, Panin et al., 2014**]. Several authors [**Hwang and Mai, 1990, Cheung and Wang, 1995, Aboolian et al., 2008**] argue that this strategy is suboptimal. A joint decision is more suited for practical applications and can provide valuable insight into whether or not is profitable for a firm to enter a new market.

Some papers consider simultaneous decisions on location, price and capacity, but they omit competition [**Dobson and Stavrulaki, 2007, Abouee-Mehrizi et al., 2011, Tong, 2011, Hajipour et al., 2016, Tavakkoli-Moghaddam et al., 2017**]. [**Pahlavani and Saidi-Mehrabad, 2011**] include competition, but they consider location as fixed.

Other papers consider congestion and competition and pricing, but locations are fixed [**Sattinger, 2002, Chen and Wan, 2003**]. A full review of the literature concerning competition in queueing systems is provided in [**Hassin, 2016**].

1.5. Contribution

The aim of this thesis is twofold. From a modelling standpoint we extend and improve the existing models by incorporating location, service levels and pricing decisions into a competitive, congested user-choice market. Users select facilities based not only on proximity, but on more realistic traits, such as congestion, price, probability of service denial, etc. **In our three papers** we analyze various CC–FLP models that fit the MPEC framework, which have not been previously investigated.

Another significant **novelty** of our work is from an algorithmic point of view. The bilevel literature is very rich, but the variants of CC–FLP that we consider are extremely challenging, due to their combinatorial, highly non-linear, non-differentiable, and nonconvex nature, even when location variables are fixed.

The lower level is non-linear, the upper-level is non-convex, and the KKT optimality conditions of the lower-level can not be reformulated into an MILP. Since the performance of the exact and quasi-exact bilevel algorithms typically rests on these conditions, they cannot be successfully applied.

Metaheuristics could be used in our case (Tabu search, genetic algorithms, etc.), but they are not ideal, for several reasons. First, the solution space would increase tremendously when modelling the upper-level non-binary variables, like service level and price. Additionally they only yield local optimums, which is not desirable, as we are interested in global (or close to global) optimal solutions. For these difficult problems, we propose exact and approximated algorithms, as well as tailored heuristics.

What distinguishes our papers from one another is

i) The presence of different elements in the models. In the first paper we consider continuous service levels, finite queues, and both stochastic and deterministic patronizing behaviour. In the second article service level is fixed and the number of servers is the decision variable. The third article incorporates pricing, and the service levels are continuous.

ii) The algorithms employed are significantly different. In the first article we propose a piece-wise linear approximation (matheuristic) and a tailored heuristic. While the second article is dedicated to a Branch-and-Bound exact method, the third paper proposes an approximated method inspired from the toll pricing network methods.

Chapter 2

Competitive facility location with selfish users and queues

In the first article we consider the problem faced by a service firm locating new facilities in a competitive market. A customer traveling from node i to facility j incurs a fixed travel time t_{ij}. Arriving at facility j, she observes the queue and joins it, provided there are no more than $K-1$ customers in the system. If there are no vacancies, she is denied access, and leaves the system as a lost customer.

The objective of the emerging firm is to maximize the total throughput rate at its facilities, rather than the arrival rate, which was previously considered in the literature [**Marianov et al., 2008**]. We demonstrate why, when balking is present, the throughput rate is preferable to the arrival rate, and how the maximization of the latter can lead to paradoxical and unrealistic situations. The firm decides on location (binary) and a continuous service rate, has a limited budget that can be spent on building facilities or service rates.

For the sake of computational tractability, and for 'cleaner analytical results' [**Berman and Krass, 2015**], we use single-server queues in this paper. But our model is flexible and can accommodate a number of situations, that include or not balking. Other types of queues can be considered (e.g., $M/M/s/K$ or $M/M/s$) provided that the number of server s is fixed, and the decision variable is the service rate μ.

In a random utility model, clients patronize the facility that minimizes their individual disutility, expressed as a linear combination of travel time, queueing, and probability of service denial. Then, the utility of facility j for a customer originating from demand node i

is

$$\tilde{u}_{ij} = -u_{ij} + \varepsilon_{ij} = -(t_{ij} + \alpha w_j + \beta p_{Kj}) + \varepsilon_{ij},$$

where ε_{ij} are independent Gumbel variates with common scale parameter θ and variance $\pi^2/(6 \cdot \theta^2)$.

In this multinomial Logit framework, flows between demand nodes i and open facilities j are determined according to the formula (see **[McFadden, 1974]**)

$$x_{ij} = d_i \frac{e^{-\theta\,(t_{ij} + \alpha w_j + \beta p_{Kj})}}{\sum_{l \in J^*} e^{-\theta\,(t_{il} + \alpha w_l + \beta p_{Kl})}}, \tag{2.0.1}$$

where J^* represents the set of open facilities. For large values of θ, the assignment approaches that of a Wardrop equilibrium.

Eq. (2.0.1) can be reformulated as a convex optimization problem. Then, the original program is a non-linear bilevel problem (and an MPEC), involving a leader and a follower (users). Beyond the analysis of the theoretical properties of our model, we propose two resolution techniques.

The first technique is based on the bilevel reformulation. We write a piecewise linear approximation of the lower-level nonlinear terms and constraints, followed by the optimality conditions of the obtained program. Then, the resulting bilinear terms, and the upper-level objective function are linearized in order to reduce the model to an MILP, which we solve using an off-the-shelf software, such as CPLEX. The algorithm is a matheuristic for which no formal bound on the error is guaranteed, in the presence of balking. In its absence, this approach is asymptotically exact.

The second approach is a heuristic method based on a surrogate single-level problem that automatically yields user-optimized flows.

Author contributions

— The general research ideas were proposed by my supervisor, Patrice Marcotte.

— The research (including proofs, code, experiments, etc.) was carried out by me.

— The article was written by me, and it was revised and corrected by Patrice Marcotte.

Submitted to Operations Research June 2016. Accepted May 2018.

Competitive facility location with selfish users and queues

Teodora Dan, Patrice Marcotte

ABSTRACT

In a competitive environment, we consider the problem faced by a service firm that makes decisions with respect to both the location and service levels of its facilities, taking into account that users patronize the facility that maximizes their individual utility, expressed as the sum of travel time, queueing delay, and a random term. This situation can be modelled as a bilevel program that involves discrete and continuous variables, as well as linear and nonlinear (convex and nonconvex) functions. We design for its solution an algorithm based on piecewise linear approximation, as well as a matheuristic that exploits the very structure of the problem.

Keywords: location, bilevel programming, equilibrium, queueing, nonconvex

2.1. Introduction

2.1.1. Contribution of this paper

While the literature concerning discrete facility location is vast, few studies have focused on user choice, where the latter frequently involves congestion, either along the paths leading to a facility, or at the facility itself. The aim of this paper is to analyze a model that captures

the key features of congestion within a user choice environment, yielding a bilevel program where the leader firm's objective function integrates the stochastic equilibrium resulting from the choice of locations and the associated service levels. Beyond the analysis of the model's theoretical properties, the paper is devoted to the design and analysis of efficient algorithms, whose nature is either based on approximations or heuristic.

Our model is closely related to that of [**Marianov et al., 2008**], who propose a location model where queueing (and balking) is explicitly taken into account, while users are assigned to facilities according to a logit discrete choice model, yielding a mathematical program involving user-equilibrium constraints. Their model is well suited to a variety of applications, such as location of shops, restaurants, walk-in clinics, etc., where user flows are not in direct control of the optimizer, but are dictated by utility maximization principles. One aim of this paper is to extend and improve their model, both from the modelling and algorithmic standpoints. Our main contributions are the following:

— The introduction of service rate as endogenous variables, as well as the correct modelling of the balking process, by integrating within a user's utility the probability of service denial.

— The consideration of competing facilities.

— The explicit treatment of the deterministic (Wardrop) case, which corresponds to a zero-variance logit model.

— The reformulation of the model as a standard bilevel model, thus allowing an approximate reformulation as a mixed integer linear program MILP.

— The design of a heuristic algorithm and its validation against the MILP solution.

The remainder of this paper is organized as follows. Section 2.1.2 is devoted to the literature review, and Section 2.2 to a description of the model, together with a study of its theoretical properties. Section 2.3 is dedicated to algorithms: a linear approximation algorithm in Subsection 2.3.1, and a user-driven heuristic in Subsection 2.3.2. Numerical experiments, discussion of our results, as well as an illustrative case are detailed in Section 2.4. Extensions of the current framework are mentioned in the concluding Section 2.5.

2.1.2. Literature review

Location problems have been widely studied, due to their simple structure and numerous real-life applications. Most literature is concerned with versions of the problem where users are simply assigned to shortest paths, and thus sidesteps the nonlinearities associated with the important issue of user behaviour, including congestion. In our model, customers select their own path and whenever congestion occurs, customers leaving from the same origin may travel along different paths or patronize different facilities. This user behavior principle fits the framework of a Wardrop equilibrium in the deterministic case, and of stochastic user equilibrium when a random utility model of delay is assumed. The overall bilevel model belongs to the class of mathematical programs with equilibrium constraints (MPEC), where the equilibrium can be expressed as a variational inequality. It can be reformulated as an NP-hard discrete nonlinear bilevel program which, it goes without saying, poses formidable challenges from the computational point of view.

Competitive location models were introduced by [**Hotelling, 1929**]. In his seminal paper, the author addresses the simple situation where two firms engage in spatial competition, with the purpose of maximizing individual profit through the location of a point along a segment located at respective distances a and b from the endpoints. It is assumed that demand is uniformly distributed along the line segment, and customers patronize the closest facility. This work represents the cornerstone for a plethora of articles concerned with the topic of competitive facility location. The environment considered therein was generalized to a network by [**Hakimi, 1983**], who studied variants of the weighted p-median problem involving competition. [**Labbé and Hakimi, 1991**] address a two-stage location-allocation game, where location is decided at the first stage while, at the second stage, two firms engage in a Cournot game with respect to quantities. An interesting development is considered by [**Küçükaydin et al., 2011**], where one firm decides the sites and attractiveness for new facilities in order to maximize its profit. In this Stackelberg (leader-follower) setting, the competitor responds to the leader's action and adjusts its attractiveness level to maximize its profit, while user behavior is characterized by Huff's gravity law. In the work of [**Beresnev, 2013**], two competing firms strive to maximize profit as well, but user preferences are provided by a linear order relation. The model is then solved by branch-and-bound techniques. [**Drezner et al., 2015**] address a leader-follower competitive coverage model,

where the attractiveness of a facility is related to an attraction radius, and customers are spread evenly among facilities that fall within this radius. The leader can open new facilities or adjust the attractiveness of existing ones, while the competitor responds accordingly. Both firms compete for market share within budget limits.

Besides competition, congestion occurs naturally in an environment with limited resources. It can arise either at facilities, or along the road. Although basic models are content to incorporate congestion in the form of maximum capacity, more elaborate models capture congestion through functional forms derived or not from queueing theory. Within this framework we note the work of [**Desrochers et al., 1995**] who consider an extension of a deterministic facility location problem, where individual delays (travel time) increase with traffic. The model is centralized, namely, users are assigned as to minimize the sum of opening cost, waiting delays, and travel times experienced by the users. Although the authors mention a user-choice version of their model that fits the bilevel programming paradigm, they do not suggest solution algorithms for its solution. A related formulation, where service rates are endogenous, is considered by [**Castillo et al., 2009**]. Users are assigned to facilities as to minimize the sum of the number of waiting customers and the total opening and service costs. Within the framework of centralized systems, [**Marianov, 2003**] formulates a model for locating facilities subject to congestion , and where demand is elastic with respect to travel time and queue length. In this framework, customers are assigned to centers in order to maximize total demand. Location of congested facilities when demand is elastic has also been investigated by [**Berman and Drezner, 2006**]. Similar to [**Marianov, 2003**], the objective of the model is to maximize total demand, subject to constraints on the waiting time at facilities. Heuristic procedures are proposed for its solution.

Another work worth mentioning is that of [**Zhang et al., 2010a**] who propose a methodology for addressing a congested facility network design problem, with the aim of improving healthcare accessibility, i.e., maximize the participation rate. The environment is user-choice, with users patronizing the facility that minimizes the sum of waiting and travel times, while demand is elastic with respect to total expected time experienced by clients. The authors illustrate the performance of a metaheuristic procedure on data issued from a network of mammography centers in Montreal, Canada. Congestion has also been considered by [**Abouee-Mehrizi et al., 2011**] in the context of simultaneous decisions of locations,

service rates and prices of facilities located at vertices of a network. They assumed that demand is elastic with respect to price, and clients spread among facilities based on proximity only, according to a multinomial logit random utility model. Congestion, which arises at facilities, is characterized by queueing equations. For a more elaborate review of congestion models in the context of facility location, the reader is referred to [**Boffey et al., 2007**].

Although congestion and competition have been previously combined, few papers have tackled both within a user-choice environment. Actually, most papers that incorporate congestion do not account for competition. On the other hand, when competition is present, users select facilities based on congestion-free traits such as distance or attractiveness. To the best of our knowledge, the only paper to address congestion in a competitive user-choice environment is that of [**Marianov et al., 2008**]. A taxonomy of the models most relevant to our research is provided in the e-companion to this article.

2.2. The model

2.2.1. Preliminaries

Let us consider the problem faced by a firm (a service center, for instance) that makes location and service level decisions, with the aim of maximizing the number of customers to attract with respect to its competitors, under a budget constraint. A salient feature of the model is that user behavior is explicitly taken into account. Precisely, users patronize the facility that maximizes their individual utility, i.e., minimizes their disutility. The latter is estimated as the sum of travel time to the facility, queueing at the facility, plus the actual probability of balking (facilities are modelled as finite-length queues). In this bilevel setting we assume a weak form of competition where competitors do not react to the leader's decisions, i.e., their locations and service levels are fixed.

Since our model is closely related to that of [**Marianov et al., 2008**], we provide a detailed description of the latter. In that work, the authors consider an oligopoly scenario in which firm A locates p new facilities in a market where competitors already operate. The 'game' takes place over a bipartite graph $V = I \times J$, where a vertex v may correspond to either a location ($v \in J$) or a demand node ($v \in I$), the latter endowed with demand d_v. We denote by $J_1 \subseteq J$ the set of candidate locations for firm A, and by J_c the set of locations of its competitors. A customer leaving vertex $i \in I$ for facility $j \in J$ incurs a fixed travel time t_{ij}.

At facility j, this customer enters an $M/M/s/K$ queue that involves s servers with identical mean service time μ, and an associated waiting time w_j. Whenever the queue reaches length $K - s$ (which corresponds to K customers in the system), any arriving customer is denied access and leaves the system as a lost customer. The disutility u_{ij} of a customer is defined as a convex combination of travel time t_{ij} and queueing delay w_j, and ignores the actual constant service time, i.e.,

$$u_{ij} = \alpha t_{ij} + (1 - \alpha) w_j, \tag{2.2.1}$$

for some scalar α between 0 and 1.

The arrival and service processes are governed by Poisson (memoryless) processes. If the arrival rate at facility j is λ_j, the probability that n customers are in the queue (or are served) is

$$p_{nj} = \begin{cases} (\rho_j^n / n!) p_{0j} & \text{if } n \leq s, \\ (\rho_j^n / (s! s^n - s)) p_{0j} & \text{if } s < n \leq K, \\ 0 & \text{if } n > K, \end{cases} \tag{2.2.2}$$

where $\rho_j = \lambda_j / \mu$ is the intensity of the queueing process and

$$p_{0j} = \left[1 + \sum_{n=1}^{s} \frac{\rho_j^n}{n!} + \frac{\rho_j^s}{s!} \sum_{n=s+1}^{K} \left(\frac{\rho_j}{s} \right)^n - s \right]^{-1}. \tag{2.2.3}$$

The demand side is cast within the framework of a random utility model, where flows between vertices i and j are determined according to the logit formula

$$x_{ij} = \frac{y_j e^{-\theta u_{ij}}}{\sum_{k \in J_1} y_k e^{-\theta u_{ik}} + \sum_{k \in J_c} e^{-\theta u_{ik}}}, \tag{2.2.4}$$

where y_j is a binary variable set to 1 if a facility is open at vertex $j \in J_1$, and to 0 otherwise. Competitors' facilities are already open, hence the absence of the factor y_k in the second summation in the denominator of Eq. (2.2.4). Parameter θ is set to $\pi / (\sigma \sqrt{6})$, where σ is the standard deviation of the Gumbel random variable yielding the probabilities (or proportions) x_{ij}. If one denotes by λ_j the arrival rate at node j, and by $\overline{\lambda}_j$ the throughput rate, the model of [**Marianov et al., 2008**] takes the form of the mathematical program

$$\max_{x,y,\lambda,w,L,\bar{\lambda},p,\rho} \quad \sum_{j \in J_1} \lambda_j$$

$$
\begin{aligned}
\text{s.t.} \quad & \lambda_j = \sum_{i \in I} d_i x_{ij}, && \forall j \in J_1 \cup J_c \\
& x_{ij} = \frac{y_j e^{-\theta u_{ij}}}{\sum_{k \in J_1} y_k e^{-\theta u_{ik}} + \sum_{k \in J_c} e^{-\theta u_{ik}}}, && \forall i \in I, \forall j \in J_1 \cup J_c \qquad (2.2.5) \\
& u_{ij} = \alpha t_{ij} + (1-\alpha) w_j, && \forall i \in I, \forall j \in J \\
& w_j = L_j / \overline{\lambda}_j, && \forall j \in J \\
& L_j = \sum_{n=s}^{K} (n-s) p_{nj}, && \forall j \in J \\
& \overline{\lambda}_j = \lambda_j (1 - p_{Kj}), && \forall j \in J \\
& x_{ij} \leq y_j, && \forall i \in I, \forall j \in J_1 \\
& \sum_{j \in J} x_{ij} = 1, && \forall i \in I \\
& \sum_{j \in J_1} y_j = P, \\
& 0 \leq x_{ij} \leq 1, && \forall i \in I, \forall j \in J \\
& \lambda_j \geq 0, && \forall j \in J \\
& \rho_j = \lambda_j / \mu_j, && \forall j \in J \\
& y_j \in \{0, 1\}, && \forall j \in J_1 \\
& \text{constraints (2.2.2) and (2.2.3).}
\end{aligned}
$$

Once the binary location variables y_j are set, the remaining quantities are determined through the solution of a nonlinear fixed point problem, where the probabilities x_{ij} of choosing a facility j depend on waiting times, which are themselves functions of the demand rate vector λ, while demand rates depend on the probabilities x_{ij}. This yields a mathematical program with an embedded fixed point problem described in Eq. (2.2.5). The authors show that this equation admits a unique solution, and propose a variant of Newton-Raphson algorithm for its determination. The model is then addressed by a two-phase meta-heuristic procedure that combines GRASP (Greedy Randomized Adaptive Search Procedure) and Tabu Search. In the initial phase, facility locations are selected and a nonlinear assignment problem is solved. In the second phase, Tabu Search is used to improve upon the initial location decisions.

A key feature of the model is the possible occurrence of balking, due to a fixed buffer of size $K - s$. Besides its practical importance, balking allows the arrival rate at a facility to actually exceed the service rate, without the queues growing unbounded. However, this has two important consequences. First, note that the objective is to maximize the number of clients $\sum_{j \in J_1} \lambda_j$ *showing up* at the facilities and not the number of clients $\sum_{j \in J_1} \bar{\lambda}_j$ that *actually* access service. It follows that a solution with a low rate of served clients might be preferred to one with a high rate, if both its arrival and rejection rates are very high. This situation is illustrated in Figure 2.1. In this example, facilities can be set up at three sites (A, B and D) coinciding with two demand vertices. The competitor's facility is located at C. Demand d_1 is 200 at vertex 1 and $d_2 = 10$ at vertex 2, while distances between vertices are shown next to the edges of the network. On the supply side, the common service rate at all facilities is equal to 100. Facilities are modelled as $M/M/1/99$ queues. For simplicity, we assume $\theta = \infty$, the limiting case of the random utility model. Accordingly, at equilibrium, clients issued from a common origin will experience identical delays (travel time plus queueing delay), thus achieveing a Wardrop equilibrium.

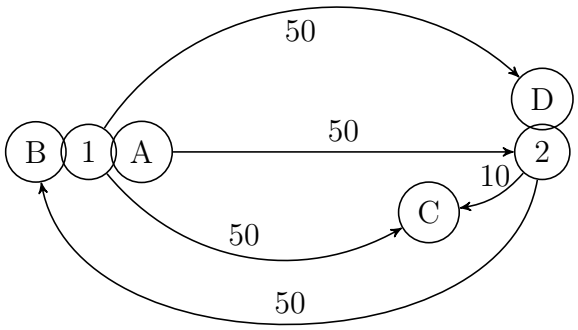


Figure 2.1. Paradox when maximizing λ instead of $\bar{\lambda}$.

Assuming that the leader's budget only allows two facilities to be opened, the options are to open sites A and B, or sites A and D (B–D is equivalent to A–D). In the first case, demand d_1 is assigned to sites A and B, while d_2 patronize the competitor's facility. Basic arithmetic shows that the total arrival rate at the leader's facilities is $\lambda = \lambda_1 + \lambda_2 = 200$, and that the number being serviced is $\bar{\lambda} = \bar{\lambda}_A + \bar{\lambda}_B = 198$. If facilities are opened at sites A and D, d_1 is assigned to site A, and d_2 to site D, with no client assigned to the competitor. The total arrival rate at the leader's facilities is $\lambda = \lambda_A + \lambda_B = 210$ and the amount of customers

receiving service is $\bar{\lambda} = \bar{\lambda}_A + \bar{\lambda}_B = 101$. In either case, the maximum λ corresponds to a much smaller value of $\bar{\lambda}$. Moreover, the solution that attracts more customers is less profitable, as roughly half of the clients will balk, due to no vacancies in the queue, and thus experience low delays at the facility.

A second issue is related to the definition of customer utility, which embeds travel and queueing delays, but ignores balking. Returning to the example of Figure 2.1, when sites A and D are open, demand d_1 originating in 1 patronizes site A, notwithstanding a probability of balking close to 50%. This situation is not realistic, given that facilities located at sites C and D are relatively close and have low waiting times and probability of rejection. Since the queueing delay is directly related to the buffer capacity $K - 1$, facilities with small buffers (or none at all!) will turn down most arriving customers, in contrast with facilities equipped with large buffer zones. This leads to the paradoxical situation where customers will favour facilities for which the probability of balking is high, since it will minimize the overall time spent in the system! This effect is exacerbated by the maximization of the arrival rate (rather than the throughput rate) and will only disappear if buffers have infinite capacities.

2.2.2. A new model

We now focus on a variant of the model of [**Marianov et al., 2008**] that differs in three significant ways: the objective is the throughput rate (rather than the arrival rate), service rates are decision variables, and users integrate within their utility function the probability of balking. Due to its generality, the model is flexible and can accommodate a number of situations that will be mentioned in Section 2.5. Its main elements are the following.

— The objective is to maximize the throughput rate. This is relevant to applications that arise in health care management (see [**Zhang et al., 2010b**]). In the context where revenue per customer served is constant, this is equivalent to maximizing the overall profit.

— Denial of service, i.e., balking, may be the result of either physical constraints or user behaviour.

— Customer utility is expressed as a linear combination of travel and waiting time, which is a standard assumption in the choice behaviour literature (see [**Berman and Krass, 2015**]).

— For the sake of computational tractability, facilities are modelled as single-server queue whose service rates are endogenous to the model. This is a standard approach in the location science literature, and can be preferable to the multi-server approach. In [**Berman and Krass, 2015**], the authors argue that this representation 'leads to cleaner analytical results' and can be more realistic in some practical situations. For instance, a medical clinic requires different types of personnel (doctors, nurses, machines, etc.) and it might be easier for the planner to reason in terms of people served per hour rather than to model each server separately.

— The leader has a limited budget that can be spent on building facilities or improving service rates. All the techniques developed in the paper would also apply to a model where setup costs enter the objective.

We now specify the notation specific to the model. The budget is set to B, the fixed cost of locating a new facility f to c_f, while the cost of improving the service rate of an $M/M/1/K$ queue ($K-1$ available places in the queue, and 1 place at the server) by one unit is c_μ. A customer observes the queue upon arrival, and leaves if there are more than $K-1$ customers already waiting.

In this context, the probability p_{nj} of having n customers in the queue (or being served) at facility j is given by

$$p_{nj} = \begin{cases} \rho_j^n \dfrac{1-\rho_j}{1-\rho_j^{K+1}}, & n \leq K,\ \rho_j \neq 1 \\ \dfrac{1}{K+1}, & n \leq K,\ \rho_j = 1 \\ 0, & n > K, \end{cases} \tag{2.2.6}$$

where $\rho_j = \lambda_j/\mu_j$ is the intensity of the process. Eq. (2.2.6) is another way of rewriting Eq. (2.2.2) and for a fixed n, p_{nj} is continuous in variable ρ_j. At facility j, the expected number L_j of customers in the system is

$$L_j = \sum_{n=0}^{K} n p_{nj}. \tag{2.2.7}$$

The effective arrival rate, i.e., the number of customers that access the service, is denoted by $\overline{\lambda}$, i.e.,

$$\overline{\lambda}_j = \lambda_j(1 - p_{Kj}), \quad \forall j \in J. \tag{2.2.8}$$

The average waiting time w_j in the system (including service time) is a function of the service and arrival rates. According to Little's formula, we have that

$$w_j = \frac{L_j}{\overline{\lambda}_j}, \quad \forall j \in J. \tag{2.2.9}$$

Basic algebra yields the expression of the waiting time at open facilities:

$$w_j(\lambda_j, \mu_j) = \begin{cases} \dfrac{1}{\mu_j}\left(K + \dfrac{K}{\rho_j^K - 1} - \dfrac{1}{\rho_j - 1}\right), & \rho_j \neq 1 \\ \dfrac{K+1}{2\mu_j}, & \rho_j = 1. \end{cases} \tag{2.2.10}$$

Note that w_j, according to this definition, is a continuous function with respect to ρ, λ and μ, even as $\rho_j = \lambda_j/\mu_j = 1$.

Stochastic assignment

In a random utility model, clients patronize the facility that minimizes their individual disutility, expressed as a linear combination of travel time, queueing, and probability of accessing service. In this framework, the utility of facility j for a customer issued from demand node i is given by

$$\begin{aligned} \tilde{u}_{ij} &= -u_{ij} + \varepsilon_{ij} \\ &= -(t_{ij} + \alpha w_j + \beta p_{Kj}) + \varepsilon_{ij}, \end{aligned}$$

where ε_{ij} are independent Gumbel variates with common scale parameter θ and variance $\pi^2/(6 \cdot \theta^2)$. It is fair to assume that as long as travel time is not too large, a customer will patronize a facility in which she is more likely to access service, as opposed to a highly congested one. In other words, in the customers' eyes, a facility located at a distance less than β and with a very low probability of balking is preferable to a near-by facility. In this context, the parameter β can be interpreted as the price of service accessibility. In practice, parameters α, β and θ could be estimated using customer surveys. The task of determining real-values for these parameters is outside the scope of this paper.

In this multinomial logit framework (see [**McFadden, 1974**]), the demand generated at node i that patronize an open facility j is given by the expression

$$x_{ij} = d_i \frac{e^{-\theta\,(t_{ij} + \alpha w_j + \beta p_{Kj})}}{\sum_{l \in J^*} e^{-\theta\,(t_{il} + \alpha w_l + \beta p_{Kl})}}, \tag{2.2.11}$$

where J^* represents the set of open facilities. For small values of θ, users are spread more or less evenly between facilities while, when θ is large, the assignment approaches that of a Wardrop equilibrium (see [**Fisk, 1980**]). According to our assumptions, the problem can be formulated as the equilibrium-constrained nonlinear mixed integer program involving a leader and a follower (users):

(P) LEADER: $\max_{\substack{y,\mu,\lambda,\bar{\lambda},\\ x,w,p,\rho}}$

$$\sum_{j \in J_1} \overline{\lambda}_j \tag{2.2.12}$$

$$\sum_{j \in J_1} c_f y_j + \sum_{j \in J_1} c_\mu \mu_j \leq B, \tag{2.2.13}$$

$$\mu_j \leq M y_j, \qquad \forall j \in J_1 \tag{2.2.14}$$

$$\overline{\lambda}_j = \lambda_j (1 - p_{Kj}), \qquad \forall j \in J \tag{2.2.15}$$

$$y_j \in \{0,1\}, \mu_j \geq 0, \qquad \forall j \in J_1 \tag{2.2.16}$$

USERS:

$$x_{ij} = d_i \frac{y_j \cdot e^{-\theta\,(t_{ij} + \alpha w_j + \beta p_{Kj})}}{\sum_{l \in J^*} e^{-\theta\,(t_{il} + \alpha w_l + \beta p_{Kl})}}, \quad \forall i \in I;\ \forall j \in J \tag{2.2.17}$$

$$\lambda_j = \sum_{i \in I} x_{ij}, \qquad \forall j \in J \tag{2.2.18}$$

$$\rho_j \mu_j = \lambda_j, \qquad \forall j \in J \tag{2.2.19}$$

constraints (2.2.6) and (2.2.10). (2.2.20)

The decision variables are the vectors μ and y, while the user assignment x is the solution of a fixed point problem. In Eq. (2.2.14), M is a sufficiently large constant that can be set to $M = (B - c_f)/c_\mu$.

Problem (P) has a stochastic basis and the limiting case $\theta = \infty$ yields a deterministic version of (P) where customers are assigned to facilities according to Wardrop's equilibrium principle. If $c_i(\mu)$ denotes the minimum disutility (travel + waiting time and probability of balking) for users originating from node i, the optimal solution x^* is then characterized by

the complementarity system

$$t_{ij} + \alpha w_j(x^*, \mu) + \beta p_{Kj}(x^*, \mu) \begin{cases} = c_i(\mu), & \text{if } x^*_{ij} > 0 \\ \geq c_i(\mu), & \text{if } x^*_{ij} = 0, \end{cases} \tag{2.2.21}$$

and the deterministic version of (P) takes the form

(P*)

$$\text{LEADER:} \quad \max_{\substack{y,\mu,\lambda\bar{\lambda},\\ x,w,p,\rho}} \sum_{j\in J_1} \bar{\lambda}_j \tag{2.2.22}$$

$$\text{s.t. constraints (2.2.13) –(2.2.16)} \tag{2.2.23}$$

$$\text{USERS:} \quad t_{ij} + \alpha w_j(x^*, \mu) + \beta p_{Kj}(x^*, \mu) - c_i(\mu) \geq 0, \qquad \forall i \in I;\ \forall j \in J \tag{2.2.24}$$

$$x_{ij}\left(t_{ij} + \alpha w_j(x^*, \mu) + \beta p_{Kj}(x^*, \mu) - c_i(\mu)\right) = 0, \qquad \forall i \in I;\ \forall j \in J \tag{2.2.25}$$

$$x_{ij} \geq 0, \qquad \forall i \in I;\ \forall j \in J \tag{2.2.26}$$

$$\text{constraints (2.2.18), (2.2.19) (2.2.6) and (2.2.10).} \tag{2.2.27}$$

In (P), the solution of the lower level equilibrium problem can be obtained by solving a convex optimization problem akin to [**Fisk, 1980**]. In our framework, this program takes the form

(P2)

$$\min_{x,\lambda,\rho,w,p} \quad \sum_{i\in I}\sum_{j\in J^*}\left[\frac{1}{\theta}x_{ij}\ln x_{ij} + x_{ij}t_{ij}\right] + \alpha\sum_{j\in J^*}\int_0^{\lambda_j} w_j(q,\mu_j)\,\mathrm{d}q + \beta\sum_{j\in J^*}\int_0^{\lambda_j} p_{Kj}(q,\mu_j)\,\mathrm{d}q \tag{2.2.28}$$

$$\text{s.t.} \quad \sum_{j\in J^*} x_{ij} = d_i, \qquad \forall i \in I \tag{2.2.29}$$

$$x_{ij} \geq 0, \qquad \forall i \in I; \forall j \in J^* \tag{2.2.30}$$

$$\lambda_j = \sum_{i\in I} x_{ij}, \qquad \forall j \in J^* \tag{2.2.31}$$

$$\rho_j\mu_j = \lambda_j, \qquad \forall j \in J \tag{2.2.32}$$

$$\text{constraints (2.2.6) and (2.2.10).} \tag{2.2.33}$$

Indeed, it is easy to check that, if θ is finite, x_{ij} cannot be zero at the solution, which implies that inequality (2.2.30) can be left out. If we let a_i, and c_j be the Lagrange multipliers

associated with Equations (2.2.29) and (2.2.31), respectively, the first-order necessary and sufficient optimality conditions are given by

$$\frac{\partial L}{\partial x_{ij}} = 0 \;\Rightarrow \qquad \frac{1}{\theta}\left(\ln x_{ij} + 1\ \right) + t_{ij} - a_i + c_j = 0 \tag{2.2.34}$$

$$\frac{\partial L}{\partial \lambda_j} = 0 \;\Rightarrow \qquad \alpha w_j(\lambda_j, \mu_j) + \beta p_{Kj}(\lambda_j, \mu_j) - c_j = 0. \tag{2.2.35}$$

It follows that $c_j = \alpha w_j(\lambda_j, \mu_j) + \beta p_{Kj}(\lambda_j, \mu_j)$, and Equation (2.2.34) yields

$$x_{ij} = \frac{e^{-\theta u_{ij}}}{e^{-\theta a_i} + 1}.$$

By substituting x_{ij} into (2.2.29) we obtain

$$x_{ij} = d_i \frac{e^{-\theta u_{ij}}}{\sum_{l \in J^*} e^{-\theta u_{il}}}.$$

Now, replacing the fixed point problem by its optimization counterpart, the original model can be formulated as a bilevel program. At the upper level, the firm maximizes total market capture, subject to a budget constraint while, at the lower level, the follower solves Problem (P2). The main advantage of this reformulation is that we can adapt for its solution methods and algorithms from convex bilevel programming.

2.2.3. Properties of the model

This subsection is devoted to the properties and features of our model. First, let us consider the integrals of the waiting time and probability of balking, $W_j(\lambda_j, \mu_j)$ and $P_{Kj}(\lambda_j, \mu_j)$ respectively, that enter the lower level's objective function. Note that w_j and p_{Kj} are continuous functions (as previously defined). We have

$$W_j(\lambda_j, \mu_j) = \int_0^{\lambda_j} w_j(q, \mu_j)\, \mathrm{d}q = \begin{cases} \displaystyle\int_0^{\lambda_j} \frac{1}{\mu_j}\left(K + \frac{K}{\left(\frac{q}{\mu_j}\right)^K - 1} - \frac{1}{\frac{q}{\mu_j} - 1}\right) \mathrm{d}q, & \text{if } q \neq \mu_j \\ \displaystyle\int_0^{\lambda_j} \frac{K+1}{2\mu_j}\, \mathrm{d}q, & \text{if } q = \mu_j. \end{cases}$$

$$P_{Kj}(\lambda_j,\mu_j)=\int_0^{\lambda_j} p_{Kj}(q,\mu_j)\,\mathrm{d}q=\begin{cases}\displaystyle\int_0^{\lambda_j}\frac{\left(\frac{q}{\mu_j}\right)^K-\left(\frac{q}{\mu_j}\right)^{K+1}}{1-\left(\frac{q}{\mu_j}\right)^{K+1}}\,\mathrm{d}q, & \text{if } q\neq\mu_j\\ \displaystyle\int_0^{\lambda_j}\frac{1}{K+1}\,\mathrm{d}q, & \text{if } q=\mu_j.\end{cases}$$

Let $l_{wj}=\dfrac{1}{\mu_j}\displaystyle\int_0^{\lambda_j}\frac{-1}{\frac{q}{\mu_j}-1}\,\mathrm{d}q$, and $l_{pj}=\displaystyle\int_0^{\lambda_j}\frac{\left(\frac{q}{\mu_j}\right)^K}{1-\left(\frac{q}{\mu_j}\right)^{K+1}}\,\mathrm{d}q$. Then

$$l_{wj}=\begin{cases}-\ln\left(\dfrac{q}{\mu_j}-1\right), & \text{if } q>\mu_j\\ -\ln\left(1-\left(\dfrac{q}{\mu_j}\right)\right), & \text{if } q<\mu_j\end{cases}\quad\text{and}\quad l_{pj}=\begin{cases}-\ln\left(\left(\dfrac{q}{\mu_j}\right)^{K+1}-1\right)\dfrac{\mu_j}{K+1}, & \text{if } q>\mu_j\\ -\ln\left(1-\left(\dfrac{q}{\mu_j}\right)^{K+1}\right)\dfrac{\mu_j}{K+1}, & \text{if } q<\mu_j,\end{cases}\tag{2.2.36}$$

which yields the following expression for the integral of the waiting time:

$$W_j(\lambda_j,\mu_j)=\begin{cases}K\dfrac{\lambda_j}{\mu_j}+l_{wj}+\dfrac{K}{\mu_j}\displaystyle\int_0^{\lambda_j}\frac{1}{\left(\frac{q}{\mu_j}\right)^K-1}\,\mathrm{d}q, & \text{if } q\neq\mu_j\\ \dfrac{(K+1)\lambda_j}{2\mu_j}, & \text{if } \lambda_j=\mu_j\end{cases}\tag{2.2.37}$$

and for the integral of the balking probability:

$$P_{Kj}(\lambda_j,\mu_j)=\begin{cases}\lambda_j+l_{pj}+\displaystyle\int_0^{\lambda_j}\frac{1}{\left(\frac{q}{\mu_j}\right)^{K+1}-1}dq, & \text{if } q\neq\mu_j\\ \dfrac{\lambda_j}{(K+1)}, & \text{if } \lambda_j=\mu_j.\end{cases}\tag{2.2.38}$$

Note that the general integral $\displaystyle\int\frac{1}{q^K-1}\,\mathrm{d}q=-qF_1^2(1,1/K;1+1/K;q^K)$, where F_1^2 stands for the hypergeometric function, and does not have a closed-form expression for general K, although it can be evaluated for any fixed value of K. We have that

$$\int_0^{\lambda_j} w_j(q,\mu_j)\,\mathrm{d}q=W_j(\lambda_j,\mu_j)-W_j(0,\mu_j).$$

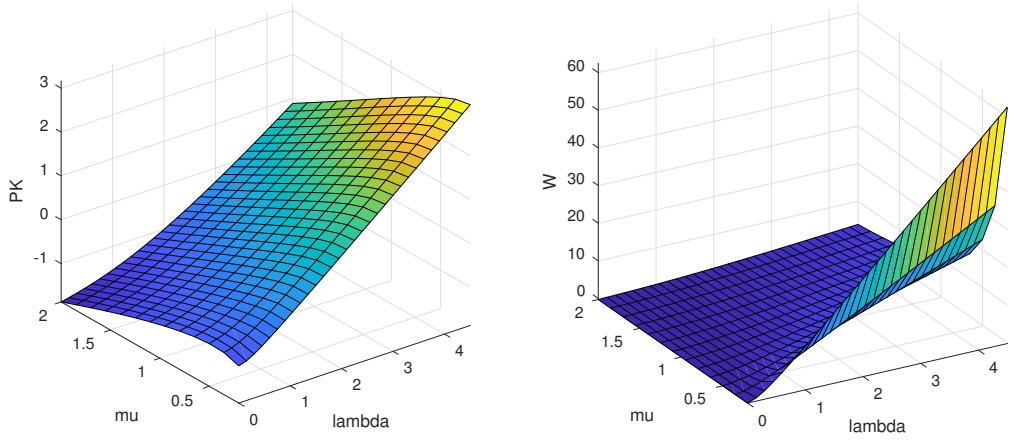


Figure 2.2

Since $W_j(0, \mu_j)$ is constant at the lower level, it can be removed from the objective function. Applying a similar operation to P_{Kj}, the lower level objective takes the form

$$\sum_{i\in I}\sum_{j\in J^*}\left[\frac{1}{\theta}x_{ij}\ln x_{ij} + x_{ij}t_{ij}\right] + \alpha\sum_{j\in J^*} W_j(\lambda_j,\mu_j) + \beta\sum_{j\in J^*} P_{Kj}(\lambda_j,\mu_j). \tag{2.2.39}$$

Proposition 1. *The waiting time w_j is increasing in λ_j.*

Proposition 2. *The probability of balking p_{Kj} is increasing in λ_j.*

From the convexity of the function $x_{ij}\ln x_{ij}$, and Propositions 1 and 2 it follows that:

Proposition 3. *The lower level objective function (2.2.39) is convex in x, hence Problem (P2) is convex.*

Proposition 4. *When $K = \infty$, i.e., balking does not occur (in this case, the model admits a solution only if the total service rate exceeds the total demand rate), W is convex jointly in λ and $\mu \Rightarrow$ the lower level objective function is convex jointly in x, λ and μ.*

Although the integral of the waiting time and probability of balking are convex in x_{ij} and λ_j, they are not *jointly* convex in λ_j and μ_j (see Figure 2.2).

Proposition 5. *$W_j(\lambda_j, \mu_j)$, the integral of the waiting time, is pseudoconvex.*

The proofs of Propositions 1, 2, 4 and 5 are provided in the e-companion to this paper.

2.3. Algorithms

This section is concerned with the design of two algorithms for addressing the bilevel location problem. The first approach is based on a piecewise-linear approximation of nonlinear (and nonconvex) functions, in order to obtain a linear bilevel problem that can be further reduced to a MILP. It is related to the MILP approximations proposed by **[Gilbert et al., 2015]** for a bilevel toll problem involving logit user assignment. Whenever the approximation is fine-grained, we expect its solution to be close to optimal. In the absence of balking, this algorithm is asymptotically exact, as proven by Theorem 9. When balking occurs, the algorithm is a matheuristic for which no formal bound on the error is guaranteed.

In contrast, the second 'matheuristic' algorithm solves a surrogate single-level problem that automatically yields user-optimized flows. It is akin to the approach of **[Marcotte, 1986]** for addressing a bilevel network design problem.

2.3.1. A linear approximation method

By linearizing the upper level nonlinear terms $\bar{\lambda}_j$ and the lower-level objective of the bilevel program, it is possible to reformulate (P) as a mixed integer linear bilevel program, which can be further reduced to a MILP. This is achieved through the following five operations:

(1) Approximate the lower-level objective function by a piecewise linear approximation.

(2) Write the KKT optimality conditions of the lower-level linear program to obtain a single-level mathematical program involving complementarity constraints (MPEC).

(3) Formulate the MPEC as an MILP, through the introduction of binary variables and 'big-M' constants.

(4) Solve the resulting MILP for optimum values of μ and y.

(5) Solve the original nonlinear lower-level problem to recover the true values of the assignment vector x associated with μ and y.

We now provide a detailed description of the linear approximation used at the first step of the algorithm. We let

$$\tilde{d} = \max_{i \in I}\{d_i\}, \quad \bar{\mu} = (B - c_f)/c_\mu, \quad \text{and} \quad \tilde{\mu} = \max\left\{\bar{\mu}, \max_{j \in J_c}\{\mu_j\}\right\},$$

and sample the interval $(0, \tilde{d}]$ using N points $x^n, n = 1, .., N$ such that $x^i < x^j$ for all $i < j$, and consider the linearization

$$f^n(x) = x(\ln x^n + 1) - x^n = a_f^n x + b_f^n. \tag{2.3.1}$$

Similarly, let $\tilde{\lambda} = \sum_{i \in I} d_i$ be the maximum arrival rate. We sample the interval $(0, \tilde{\lambda}]$ using R points $\lambda^r, r = 1, \ldots, R$ such that $\lambda^i < \lambda^j$ for $i < j$. We also generate P samples of μ over $(0, \tilde{\mu}]$ over the same interval. Let λ^r and μ^p be the samples hence obtained. We linearize $W_j(\lambda_j, \mu_j)$ and $P_{Kj}(\lambda_j, \mu_j)$ using tangent planes at points (λ^r, μ^p) for $r = 1, \ldots, R$, $p = 1, \ldots, P$ such that $\lambda^r \neq \mu^p$. Based on the gradients

$$\nabla W_j(\lambda_j, \mu_j) = (w_j(\lambda_j, \mu_j), -w_j(\lambda_j, \mu_j)\rho_j) \tag{2.3.2}$$

$$\nabla P_{Kj}(\lambda_j, \mu_j) = \left(p_{kj}(\lambda_j, \mu_j), \frac{1}{\mu_j} P_{Kj}(\lambda_j, \mu_j) - p_{kj}(\lambda_j, \mu_j)\rho_j\right), \tag{2.3.3}$$

we write the first-order Taylor approximations of $W_j(\lambda_j, \mu_j)$ and $P_{Kj}(\lambda_j, \mu_j)$, respectively:

$$g^{rp}(\lambda, \mu) = W_j(\lambda^r, \mu^p) + \nabla W_j(\lambda^r, \mu^p) \begin{pmatrix} \lambda - \lambda^r \\ \mu - \mu^p \end{pmatrix} = a_g^{rp}\lambda + b_g^{rp}\mu + c_g^{rp},$$

$$h^{rp}(\lambda, \mu) = P_{Kj}(\lambda^r, \mu^p) + \nabla P_{Kj}(\lambda^r, \mu^p) \begin{pmatrix} \lambda - \lambda^r \\ \mu - \mu^p \end{pmatrix} = a_h^{rp}\lambda + b_h^{rp}\mu + c_h^{rp}.$$

Next, we convexify $W_j and P_{Kj}$. More precisely, we construct convex piecewise linear approximations of these functions by setting them to the maximum of their linear approximations

$$W_j(\lambda_j, \mu_j) \approx \max_{r \in R, p \in P}\{g^{rp}(\lambda_j, \mu_j)\} \tag{2.3.4}$$

$$P_{Kj}(\lambda_j, \mu_j) \approx \max_{r \in R, p \in P}\{h^{rp}(\lambda_j, \mu_j)\} \tag{2.3.5}$$

We approximate $x \ln x$ along with W_j and P_{Kj}, using $f^n(x_{ij})$ as defined in Eq. (2.3.1).

$$x_{ij} \ln x_{ij} \approx \max_{n \in N}\{f^n(x_{ij})\} \tag{2.3.6}$$

Upon the introduction of auxiliary variables v, u and z, the linear approximation of (P2) takes the form

(P2-lin)

$$\min_{x,\lambda,v,u,z} \quad \sum_{i\in I}\sum_{j\in J^*}\left[\frac{1}{\theta}v_{ij}+t_{ij}x_{ij}\right]+\alpha\sum_{j\in J^*}u_j+\beta\sum_{j\in J^*}z_j \tag{2.3.7}$$

$$\text{s.t.} \quad \sum_{j\in J^*}x_{ij}=d_i, \qquad \forall i\in I \tag{2.3.8}$$

$$\lambda_j=\sum_{i\in I}x_{ij}, \qquad \forall j\in J^* \tag{2.3.9}$$

$$v_{ij}-a_f^n x_{ij}\geq b_f^n, \qquad \forall i\in I;\forall j\in J^*;\forall n\in N \tag{2.3.10}$$

$$u_j-a_g^{rp}\lambda_j-b_g^{rp}\mu_j\geq c_g^{rp}, \qquad \forall j\in J^*;\forall r\in R;\forall p\in P \tag{2.3.11}$$

$$z_j-a_h^{rp}\lambda_j-b_h^{rp}\mu_j\geq c_h^{rp}, \qquad \forall j\in J^*;\forall r\in R;\forall p\in P \tag{2.3.12}$$

$$x_{ij}\geq 0, \qquad \forall i\in I;\forall j\in J^*. \tag{2.3.13}$$

Note that (P2-lin) is an entirely linear formulation, and thus the variables x_{ij} could assume the value 0, although this cannot occur in the initial formulation (P2), due to the presence of the logarithmic barrier term $\ln x_{ij}$. Obviously, this causes no problem.

To achieve a MILP formulation, we first perform a linear approximation of the nonlinear constraint (2.2.15) using the triangle technique described in **[D'Ambrosio et al., 2010]**. This yields the equalities

$$\sum_{r=1}^{R-1}\sum_{p=1}^{P-1}\left(\bar{l}_{jrp}+\underline{l}_{jrp}\right)=1, \qquad \forall j\in J_1 \tag{2.3.14}$$

$$s_{jrp}\leq \bar{l}_{jrp}+\underline{l}_{jrp}+\bar{l}_{jrp-1}+\underline{l}_{jr-1p-1}+\bar{l}_{jr-1p-1}+\underline{l}_{jr-1p}, \quad \forall j\in J_1;\forall r\in R;\forall p\in P \tag{2.3.15}$$

$$\sum_{r=1}^{R}\sum_{p=1}^{P}s_{jrp}=1, \qquad \forall j\in J_1 \tag{2.3.16}$$

$$\lambda_j=\sum_{r=1}^{R}\sum_{p=1}^{P}s_{jrp}\lambda^r, \qquad \forall j\in J_1 \tag{2.3.17}$$

$$\mu_j=\sum_{r=1}^{R}\sum_{p=1}^{P}s_{jrp}\mu^p, \qquad \forall j\in J_1 \tag{2.3.18}$$

$$e_j = \sum_{r=1}^{R} \sum_{p=1}^{P} s_{jrp} \left(\lambda^r (1 - p_{Kj}(\lambda^r, \mu^p))\right), \qquad \forall j \in J_1. \tag{2.3.19}$$

Next, we write the optimality conditions of (P2-lin). Let γ_i, δ_j, ν_{ij}^n, π_j^{rp} and η_j^{rp} denote the dual variables associated with constraints (2.3.8), (2.3.9), (2.3.10), (2.3.11) and (2.3.12), respectively. We replace constraints (2.2.17), (2.2.18), (2.2.6) and (2.2.10) in (P) with the optimality conditions of (P2-lin), which yields a nonlinear program involving complementarity constraints. The standard method of dealing with this nonlinearity is to linearize these constraints through the introduction of binary variables and 'big-M' constants. Alternatively, one can substitute to the complementarity constraints the equality of the lower level primal and dual objectives. The latter involves bilinear terms that can be further linearized. Technical details, together with the corresponding MILP formulation, can be found in the e-companion. The MILP can be solved by an off-the-shelf software such as CPLEX. For given location variables y and service rates μ, a feasible assignment matrix x is then recovered by solving a convex assignment program that involves a simple structure. The corresponding running time (less than one second) is negligible for networks having up to 25 nodes. Note that, due to approximation errors in the MILP, the recovered solution may differ significantly from the MILP solution.

Bound on the linearization error for the $M/M/1/\infty$ case

If facilities are modelled as $M/M/1/\infty$ (infinite capacity) queues, the waiting time at a facility j is $w_j(\lambda_j, \mu_j) = 1/(\mu_j - \lambda_j)$, and its indefinite integral $W_j(\lambda_j, \mu_j) = -\log(\mu_j - \lambda_j)$, which is convex. We make the following assumptions:

i. The total service rate in the network can satisfy the entire demand.

ii. At all open facilities, $\mu_j \geq \psi + \lambda_j$ for some positive number ψ.

The latter condition ensures that waiting time at facilities is finite. In practice, ψ can be as as small as desired, and we have that $w_j \leq 1/\psi = w_{MAX}$. Let t_{MIN} and t_{MAX} represent the minimum and maximum travel time in the network, respectively. Furthermore, $w_{\text{MIN}} = 1/\mu_{\text{MAX}}$, diam$(t)$= $t_{\text{MAX}} - t_{\text{MIN}}$ and diam(w)= $w_{\text{MAX}} - w_{\text{MIN}}$. Let μ_{MAX} be the maximum service rate possible in the network, either allowed by the budget at leader's

facilities, or at competitor's facilities. Under our assumptions, we have that

$$x_{ij} = \frac{e^{-\theta\,(t_{ij}+\alpha w_j)}}{\sum\limits_{k\in J^*} e^{-\theta\,(t_{i,k}+\alpha w_k)}} \geq \frac{e^{-\theta\,(t_{\mathrm{MAX}}+\alpha w_{\mathrm{MAX}})}}{\sum\limits_{k\in J^*} e^{-\theta\,(t_{\mathrm{MIN}}+\alpha w_{\mathrm{MIN}})}} = \frac{e^{-\theta\,(\mathrm{diam}(t)+\alpha\ \mathrm{diam}(w))}}{|J^*|} \stackrel{\mathrm{def}}{=} r_{\min}, \tag{2.3.20}$$

Now, let $g(\mu, x)$ be the lower-level objective function, i.e.,

$$g(\mu,x) = \underbrace{\sum_{i\in I}\sum_{j\in J^*}\left[\frac{1}{\theta}x_{ij}\log x_{ij} + t_{ij}x_{ij}\right]}_{g_1(\mu,x)} + \alpha \underbrace{\sum_{j\in J^*} W_j(\mu_j, x)}_{g_2(\mu,x)}. \tag{2.3.21}$$

The lower-level problem can be written as:

$$(\mathrm{P}^\infty) \quad \min_x \ g(\mu,x) = g_1(\mu,x) + \alpha g_2(\mu,x) \tag{2.3.22}$$

$$\text{s.t.} \ \sum_{j\in J^*} x_{ij} = d_i \qquad \forall i \in I \tag{2.3.23}$$

$$x_{ij} \geq 0 \qquad \forall i \in I, \forall j \in J^*, \tag{2.3.24}$$

whose compact constraint set is denoted D. Next, we let

$$G(\mu,x) = \nabla_x g(\mu,x) = \nabla_x g_1(\mu,x) + \alpha\nabla_x g_2(\mu,x) = G_1(\mu,x) + \alpha G_2(\mu_x).$$

The proofs of the following results are provided in the e-companion to this paper.

Proposition 6. *G_1 is strongly monotone of modulus $\theta \cdot d_{MAX}$ with respect to x.*

Proposition 7. *G_2 is monotone in x.*

It follows directly that

Proposition 8. *G is strongly monotone in x, with modulus $\theta \cdot d_{MAX}$.*

Théorème 9. *The approximation error of the upper-level objective function is $O(1/N_1 + 1/N_2)$, where N_1 and N_2 are the number of samples for the linearization of g_1 and g_2, respectively.*

We now illustrate Theorem 9 for the instance based on the network illustrated in Figure 2.3. It involves two demand nodes, which are potential locations as well. Demand rates in nodes 1 and 2 are set to 5.5 and 15.0, respectively. The fixed cost of opening a facility is set to 5 and the unit service cost to 1, for a total budget of 25. The competition owns a

facility with service rate 25.1. On the demand side, parameters α and θ are set to 10 and 0.2, respectively.

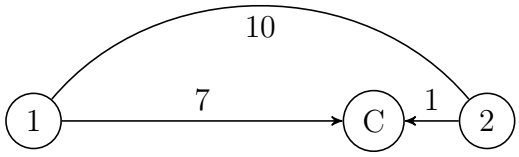


Figure 2.3. A three-node network.

For each set of open locations, the problem can be approximately solved by sampling a very large number of values of the parameter μ. This yields a quasi-optimal solution with objective 10.197, where both facilities are open, with respective service rates 5.325 and 9.675. The linear approximation algorithm was then run for different sample sizes.

The optimum of the approximation MILP, as well as the true objective values corresponding to the MILP solutions, are displayed in Figure 2.4, where we observe that

— The approximated objective mostly overestimates the true objective.

— The true objective obtained by solving for the actual equilibrium with respect to the service levels quickly reaches a near-optimal solution, and actually does so for a sample size as small as 4.

— The true (recovered) objective does not increase in a monotone fashion, but stabilizes fairly quickly close to the optimum.

2.3.2. A surrogate-based heuristic

In this section we present a parameterized heuristic based on replacing the original model by a single-level model involving a surrogate objective, whose optimal solution automatically satisfies the fixed point constraint. This strategy is akin to that proposed by [**Marcotte, 1986**] for addressing a bilevel network design problem involving user-optimized flow patterns.

The rationale behind this strategy is that both the leader and the users have a shared interest in minimizing delays. We therefore expect that, if the lower level is given full control, the resulting design should favor access to the leader's facilities, and therefore yield a high

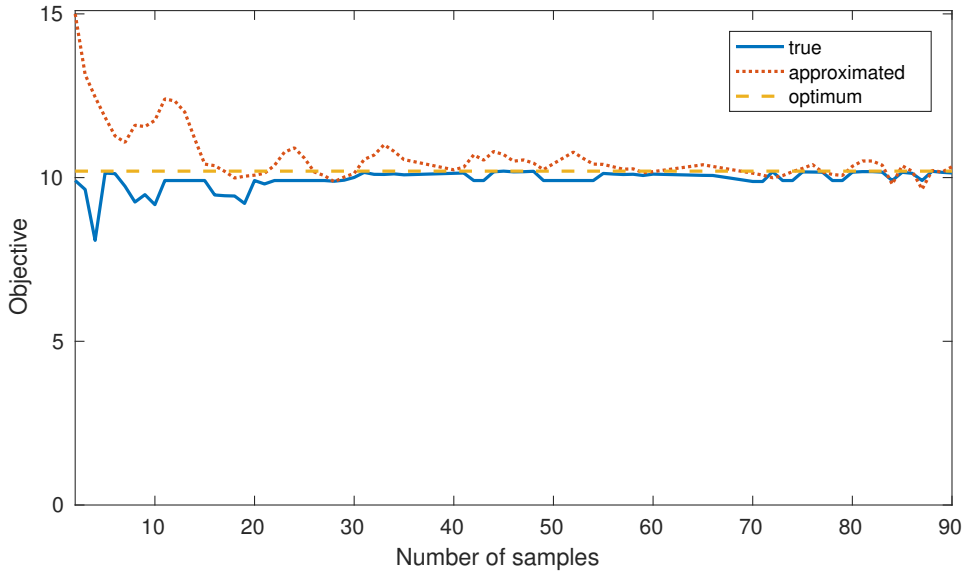


Figure 2.4. Evolution of the linear approximation MILP objective value with respect to sample size. We use the same number of samples on x, and λ. The 'approximated' line corresponds to the optimal objective of the approximate MILP. The 'true' line is the true(recovered) objective value corresponding to the MILP solution.

throughput. Incorporating the budget constraint to ensure feasibility, we obtain the single-level mixed nonlinear program

$$\text{(PH)} \quad \min_{\substack{x,y,\mu,\lambda,\\ w,p,\rho}} \quad \sum_{i\in I}\sum_{j\in J^*}\left[\frac{1}{\theta}x_{ij}\ln x_{ij} + x_{ij}t_{ij}\right] + \alpha\sum_{j\in J^*}\int_0^{\lambda_j} w_j(q,\mu_j)\,\mathrm{d}q + \beta\sum_{j\in J^*}\int_0^{\lambda_j} p_{Kj}(q,\mu_j)\,\mathrm{d}q \tag{2.3.25}$$

$$\text{s.t.} \sum_{j\in J} x_{ij} = d_i, \qquad \forall i \in I \tag{2.3.26}$$

$$\sum_{j\in J_1} y_j c_f + \sum_{j\in J_1} c_\mu \mu_j \le B, \tag{2.3.27}$$

$$\lambda_j = \sum_{i\in I} x_{ij}, \qquad \forall j \in J \tag{2.3.28}$$

$$y_j \in \{0,1\}, \qquad \forall j \in J \tag{2.3.29}$$

$$x_{ij} \ge 0, \qquad \forall i \in I; \forall j \in J \tag{2.3.30}$$

$$\rho_j = \lambda_j/\mu_j, \qquad \forall j \in J \tag{2.3.31}$$

$$\text{constraints (2.2.10) and (2.2.6)}, \tag{2.3.32}$$

whose x-solution is a logit flow assignment with respect to the design variables y and μ. For $\theta = \infty$, the limiting case (PH*) is a mathematical program involving user-equilibrium (Wardropian with respect to queueing delays) flows, and is expressed as

$$\text{(PH*)} \qquad \min_{\substack{x,y,\mu,\lambda,\\ w,p,\rho}} \quad \sum_{i\in I}\sum_{j\in J^*} x_{ij}t_{ij} + \alpha \sum_{j\in J^*}\int_0^{\lambda_j} w_j(q,\mu_j)\,\mathrm{d}q + \beta \sum_{j\in J^*}\int_0^{\lambda_j} p_{Kj}(q,\mu_j)\,\mathrm{d}q$$

$$\text{s.t.} \qquad \text{constraints (2.3.26) –(2.3.32).}$$

Properties of the surrogate model

The surrogate model always yields feasible solutions for the original model. Although its objective is nonconvex, some of its properties make it computationally tractable, as will be confirmed in Section 2.4. The proofs of the following results are provided in the e-companion.

Proposition 10. *If $K = \infty$ and there are no fixed costs, the surrogate model is convex.*

Proposition 11. *At the optimum of (PH*), if $K = \infty$, queue waiting times are equal for all leader's facilities.*

We close this section with an example that shows that, in the worst case, the difference between the heuristic optimum and the true optimum can be arbitrarily large. Let us consider the network shown in Figure 2.5, with site C belonging to the competitor, and sites A and B being potential opening nodes for the leader, with null fixed cost. We consider an infinite queue and $\alpha = 1$. Let $n > 1$, and $D_1 > 1$ and nD_1 be the demand at nodes 1 and 2, respectively. The total service rate available to the leader is $\bar{\mu} = (2n + 4)D_1$. The service rate at the competitor's facility is set to $\mu_C = 2nD_1$.

The heuristic solves the convex program (PHY*) (see the electronic companion, proof of Proposition 11), and at optimality, waiting times at facilities A and B must be equal. The KKT optimality conditions are sufficient, and any solution of the following system of equations is optimal.

$$(t_{1,A} + w_A - \gamma_1)\cdot x_{1,A} = 0 \qquad (t_{1,A} + w_A - \gamma_1) \geq 0 \qquad w_A = 1/(\mu_A - x_{1,A} - x_{1,A})$$

$$(t_{1,B} + w_B - \gamma_1)\cdot x_{1,B} = 0 \qquad (t_{1,B} + w_B - \gamma_1) \geq 0 \qquad w_B = 1/(\mu_B - x_{1,B} - x_{1,B})$$

$$(t_{1,C} + w_C - \gamma_1)\cdot x_{1,C} = 0 \qquad (t_{1,C} + w_C - \gamma_1) \geq 0 \qquad w_C = 1/(\mu_C - x_{1,C} - x_{1,C})$$

$$(t_{2,A} + w_A - \gamma_2)\cdot x_{2,A} = 0 \qquad (t_{2,A} + w_A - \gamma_2) \geq 0 \qquad (w_A - w_B)(\bar{\mu} - \mu_A - \mu_B) = 0$$

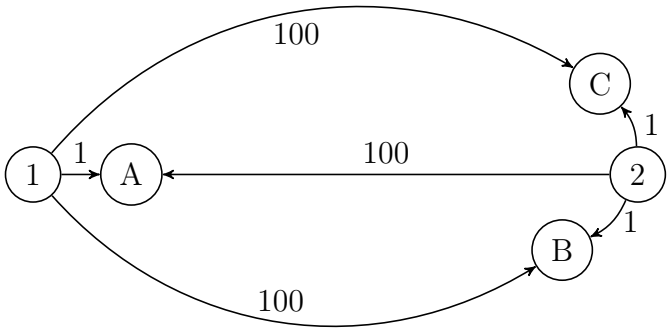


Figure 2.5. An instance where the gap between the heuristic and optimal value of the objective function can be arbitrarily large.

$$(t_{2,B} + w_B - \gamma_2) \cdot x_{2,B} = 0 \qquad (t_{2,B} + w_B - \gamma_2) \geq 0$$

$$(t_{2,C} + w_C - \gamma_2) \cdot x_{2,C} = 0 \qquad (t_{2,C} + w_C - \gamma_2) \geq 0$$

One solution of the above system is $\mu_A = \mu_B = (n+2)D_1$; $x_{1,A} = D_1$; $x_{1,B} = x_{1,C} = 0$; $x_{2,A} = 0$; $x_{2,B} = D_1$; $x_{2,C} = (n-1)D_1$; $w_A = w_B = w_C = \frac{1}{(n+1)D_1}$. In other words, users originating from 1 patronize solely facility A, while users issued from 2 choose facilities B and C. Then the objective value associated to this optimal solution is $2D_1$.

On the other hand, in the original program w_A need not be equal to w_B. Then one feasible solution is $\mu_A = 2D_1$ and $\mu_B = (2n+2)D_1$, and x, γ solve the following set of equations, issued from the KKT optimality conditions of the second level.

$$(t_{1,A} + 1/(2D_1 - x_{1,A} - x_{1,A}) - \gamma_1) \cdot x_{1,A} = 0; \qquad (t_{1,A} + 1/(2D_1 - x_{1,A} - x_{1,A}) - \gamma_1) \geq 0;$$

$$(t_{1,B} + 1/((2n+2)D_1 - x_{1,B} - x_{1,B}) - \gamma_1) \cdot x_{1,B} = 0; \qquad (t_{1,B} + 1/((2n+2)D_1 - x_{1,B} - x_{1,B}) - \gamma_1) \geq 0;$$

$$(t_{1,C} + 1/(2nD_1 - x_{1,C} - x_{1,C}) - \gamma_1) \cdot x_{1,C} = 0; \qquad (t_{1,C} + 1/(2nD_1 - x_{1,C} - x_{1,C}) - \gamma_1) \geq 0;$$

$$(t_{2,A} + 1/(2D_1 - x_{1,A} - x_{1,A}) - \gamma_2) \cdot x_{2,A} = 0; \qquad (t_{2,A} + 1/(2D_1 - x_{1,A} - x_{1,A}) - \gamma_2) \cdot x_{2,A} = 0;$$

$$(t_{2,B} + 1/((2n+2)D_1 - x_{1,B} - x_{1,B}) - \gamma_2) \cdot x_{2,B} = 0; \qquad (t_{2,B} + 1/((2n+2)D_1 - x_{1,B} - x_{1,B}) - \gamma_2) \geq 0;$$

$$(t_{2,C} + 1/(2nD_1 - x_{1,C} - x_{1,C}) - \gamma_2) \cdot x_{2,C} = 0; \qquad (t_{2,C} + 1/(2nD_1 - x_{1,C} - x_{1,C}) - \gamma_2) \geq 0.$$

We can easily check that the following solution solves the above system of equations: $\mu_A = 2D_1$; $\mu_B = (2n+2)D_1$; $x_{1,A} = D_1$; $x_{1,B} = x_{1,C} = 0$; $x_{2,A} = 0$; $x_{2,B} = D_1$; $x_{2,C} = (n-1)D_1$; $w_A = w_B = w_C = \frac{1}{(n+1)D_1}$. The objective value associated to this feasible solution

is $D_1(n+4)/2$. The ratio between the better option and the one found by the heuristic is $(n+4)/4$, which can be arbitrarily large.

A parameterized surrogate heuristic

One drawback of the heuristic solution presented in the previous section is that, for $K = \infty$ and $\theta = \infty$, queueing delays are equal, a property that might not hold at the true optimum. Actually, in order to maximize efficiency, one expects the leader to adapt its service rates to arrival rates. This can be achieved by incorporating a service-dependent linear term into the objective. This term depends on a set of positive parameters ξ_j, to be tuned, one for each facility. The resulting mathematical program is

$$(\mathrm{PH}(\xi)) \quad \min_{\substack{x,y,\mu,\lambda,\\ w,p,\rho}} \quad \sum_{i\in I}\sum_{j\in J^*}\left[\frac{1}{\theta}x_{ij}\ln x_{ij} + x_{ij}t_{ij}\right] + \alpha\sum_{j\in J^*} W_j(x,\mu_j) + \beta\sum_{j\in J^*} P_{Kj}(x,\mu_j) + \sum_{j\in J_1}\xi_j\mu_j$$

$$\text{s.t.} \quad \text{constraints (2.3.26), (2.3.27), (2.3.29)–(2.3.32).}$$

This program is transformed and solved as a MILP where the linearization is based on the techniques presented in Section 2.3.1. As before, a feasible flow assignment x compatible with the location vector y and the service rate vector μ is retrieved by solving a convex program. We now focus on the case $K = \infty$ and $\theta = \infty$, when there are no fixed costs:

$$(\mathrm{PHY}^*(\xi)) \quad \min_{\substack{y,\mu,\lambda,\\ w,p,\rho}} \quad \sum_{i\in I}\sum_{j\in J^*} x_{ij}t_{ij} - \alpha\sum_{j\in J^*}\ln(\mu_j - (\sum_{i\in I} x_{ij})) + \sum_{j\in J_1}\xi_j\mu_j$$

$$\text{s.t.} \quad \text{constraints (2.3.26), (2.3.29)–(2.3.30), (2.C.3),}$$

for which we provide a theoretical result, whose proof is provided in the e-companion.

Proposition 12. *There exists a value of ξ^* for which (PHY*(ξ^*)) yields an optimal solution for (P*).*

While the complexity of determining an optimal ξ vector is equivalent to that of solving the initial problem, educated guesses may yield good values, as will be observed later.

2.4. Experimental setup and results

The MILP formulation was solved by IBM ILOG CPLEX Optimizer version 12.5. All tests, either using the linear approximation method or heuristics, were performed on a 16 core Xeon(R) Intel(R) processor running at 2.4GHz frequency. For the linear approximation

method, we opted for the MILP formulation based on the equality between the primal and dual lower level objectives. Surprisingly, while approximate, this formulation outperformed that based on complementarity constraints.

An initial set of experiments was intended to compare the linear approximation-based method and the parameterized heuristic described in Sections 2.3.1 and 2.3.2, respectively, that involve the parameterized model (PH(ξ)). The latter is solved for different values of the parameter ξ. For each facility j, ξ_j is set to the negative of a scalar that increases with demand and decreases with distance:

$$\xi_j = -c \sum_{i \in I} \frac{d_i}{t_{ij} + 1}, \tag{2.4.1}$$

for some nonnegative parameter c. This is motivated by the fact that it makes sense, from the leader's perspective, to assign high service rates to facilities located close to high demand nodes: the lower ξ_j, the larger μ_j in the optimal solution. The term 1 in the denominator was added to t_{ij} to avoid dividing by a small number. The linear approximations involve 7, 5 and 5 uniformly distributed samples for x, λ and μ, respectively. The parameter α was set to 10, while the algorithms were run for different combinations of parameters θ and β. Travel times were varied between 0 and 100 for nodes belonging to a common cluster. Two sensible choices for the parameter β are 50 or 100, as previously explained in Section 2.2.2.

In CPLEX branching rules, priority was given to the strategic location variables over the binary variables required in the linearization process. The algorithm was stopped as soon as the optimality gap dropped below 1%, CPLEX ran out of memory (4GB), or that CPU exceeded 2,000 seconds.

Tables 2.1 and 2.2 report mean CPU times (in seconds), the optimality gap when the stopping criteria is met, and the average ratio between the objective value found by the heuristics and by the linear approximation method (as described in Section 2.3.1), for two values of the available budget. Heuristics are run for different values of parameter c, as in Eq. (2.4.1). We also report the best solutions found across these runs in the *best* column. Additionally, we let CPLEX run to optimality (gap<0.1%), regardless of the execution time, comparing the objective value obtained within 2,000 seconds and the one obtained with no time limit; we report the percentage increase (the PI column).

		heuristic over lin. approx. ratio			linear approximation relative gap(%)			linear approximation CPU(seconds)			linear approximation gap ≤0.1%	
θ	β	$c=0$	$c=1$	best	min	average	max	lin. approx.	$c=0$	$c=1$	PI(%)	CPU(s)
0.2	50	0.99	0.93	1.01	0.99	11.3	25.4	1,778	110	11	−0.66	31,239
0.5	50	1.00	0.95	1.01	0.98	12.1	25.6	1,834	17	8	0.08	14,375
2.0	50	0.99	0.93	1.00	0.88	11.5	25.6	1,833	9	7	1.20	44,832
0.2	100	0.99	0.98	1.00	0.98	11.8	26.0	1,930	101	10	1.14	13,852
0.5	100	0.98	0.97	0.99	0.98	11.1	26.1	1,836	18	9	0.00	13,888
2.0	100	1.03	1.01	1.04	0.99	11.9	26.0	1,929	9	8	3.46	13,874

Table 2.1. Comparison between the linear approximation method and two heuristics. Budget set to 500. Averages taken over 10 instances.

		heuristic over linear approximation ratio				linear approximation relative gap(%)			linear approximation CPU(seconds)				lin. approx. gap ≤0.1%	
θ	β	$c=0$	$c=1$	$c=10$	best	min	average	max	lin. approx.	$c=0$	$c=1$	$c=10$	PI(%)	CPU(s)
0.2	50	0.86	0.86	0.62	0.94	0.93	12.0	24.2	1,862	20	12	5	0.14	56,616
0.5	50	0.83	0.86	0.62	0.93	2.22	13.7	21.6	2,011	10	9	5	1.40	22,871
2.0	50	0.83	0.86	0.63	0.94	2.25	12.8	21.3	2,010	9	8	5	0.10	39,029
0.2	100	0.84	0.86	0.58	0.88	0.99	11.3	20.7	1,826	15	10	6	−0.60	23,990
0.5	100	0.83	0.84	0.62	0.90	1.92	12.4	21.7	2,009	9	9	6	0.30	22,850
2.0	100	0.82	0.84	0.59	0.87	0.99	10.9	19.3	1,903	8	8	6	0.25	11,089

Table 2.2. Comparison between the linear approximation method and three heuristics. Budget set to 250. Averages taken over 10 instances.

In most cases, CPLEX could not reach a gap less than 1% in the allotted CPU. As shown in Tables 2.1 and 2.2, the average optimality gap lies in the [11,14] interval, when time is limited. More precisely, when the budget is set to 500, roughly 40% of the tests finish with a gap between 15% and 25%, while 40% of the tests end with a gap between $1\% - 10\%$, and 20% of tests reach the optimality gap <0.1%. This was observed on all tests, regardless of the combination of parameters. On the other, when the budget is set to 250, the optimality gaps are slightly higher, after the allotted execution time. Namely, roughly 43% of gaps fall in the $[15\%, 25\%]$ interval, 25% in the $[10\%, 15\%]$ interval, 15% in the $[5\%, 10\%]$ interval, 15% in the $[1\%, 5\%]$ interval, and only 3% of tests reach the 1% optimality gap.

However, as illustrated in Figure 2.6, the optimal solutions are frequently found in the early stages of the Branch-and-Bound process, while the remaining iterations are merely used to prove optimality. The above observation is supported by the numbers in the PI column.

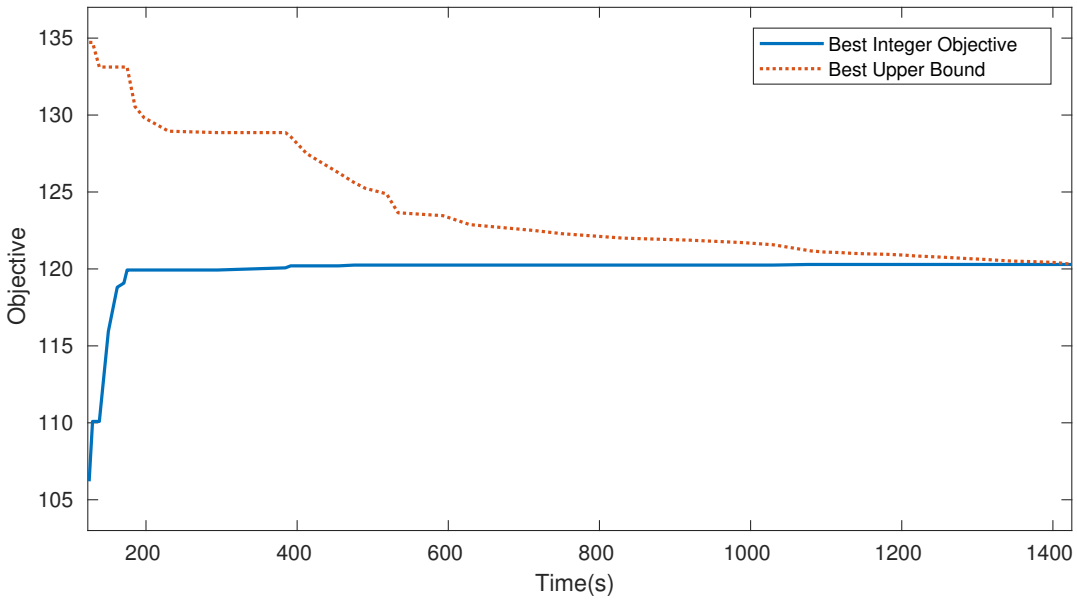


Figure 2.6. Lower and upper bounds throughout the branch-and-bound process for an instance of P-lin.

The percentage increase in objective value when running to optimality is not significant (less than 1.5%, in most cases, and 3.5% when the budget is 500, $\theta = 2.0$ and $\beta = 100$), despite a large increase in CPU. In some cases we observe a small decrease in the objective value, which is explained as follows: when running to optimality, there can be a small increase in the approximate objective value (the one found by solving the MILP), however the optimal solution corresponds to a slightly small true objective. We remind that the MILP is only an approximated version of a highly nonlinear program.

Table 2.1 shows that, for a high budget, heuristics perform well, managing to attract and serve, on average, the same number of customers as the linear approximation method, and in some cases, outperforms it. This counterintuitive result is due to approximation errors in the various linearizations performed at both the lower level and in the objective function of the linear approximation method.

Table 2.2 tells a similar story. In this case (budget = 250), taken individually, heuristics for $c = 0$, $c = 1$ and $c = 10$ do not perform very well, capturing as little as 58% of the linear approximated market in one case. However, when retaining the best out of the three, the objective value lies around $87 - 94\%$ of the linear approximated objective, and is achieved at a much lower computational cost. For instance, for a budget of 250, the CPU required by the linear approximation method exceeds by a factor of 50 ($\theta = 0.2, \beta = 50$) and up to 91 ($\theta = 2.0, \beta = 50$) the combined CPU of the three heuristics. This illustrates the limitations of the linear approximation method, which, although superior in terms of solution quality, does not scale well. We also observe that, in the heuristic case, and for identical values of the parameter β, CPU is a decreasing function of θ, a parameter that is inversely proportional to the standard deviation of the Gumbel random variable embedded into the logit process. When θ is small, users are spread over the facilities more or less regardless of their disutility, making for highly nonlinear instances that are difficult to linearize. In contrast, when θ is large, variance is small, and users focus on a limited number of destination facilities.

Within the same experimental setup, it is interesting to compare the facilities opened by the various algorithms. In Table 2.3 and Table 2.4, we display the ratio between two numbers: the number of facilities opened by both methods, and the total number of facilities openeds by the linear approximation algorithm.

The linear approximation method opens between 4 and 6 facilities, and on average $5.6-5.8$ for a budget of 500. When the budget is set to 250, the number of facilities opened by the linear approximation method decreases by at least one, on average. For both values of the budget, the leader opens less facilities for $\beta = 100$ than for $\beta = 50$. Indeed, as β increases, users require a higher service rate to make for the higher probability of balking. The budget is then more focused on increasing service rates.

For the high budget and low values of c (0 or 1) the heuristics open on average $6.8 - 8.4$ facilities. Only half of the facilities opened by the linear approximation method are among them. Nevertheless, the heuristic facilities yield large values of the objective function. For low budget, a similar situation occurs although all methods open, on average, less facilities. As a trend, the average number of open facilities decreases with c. The larger values of c yield smaller values of ξ, therefore, larger values of μ, and the heuristics put more emphasis on providing high service rates, versus opening several facilities. These results highlight the

		number of open facilities				ratio of common facilities	
θ	β	lin. approx.	$c=0$	$c=1$		$c=0$	$c=1$
0.2	50	5.9	8.3	6.8		0.52	0.54
0.5	50	6.0	8.4	6.8		0.56	0.45
2.0	50	5.9	8.4	6.8		0.66	0.55
0.2	100	5.7	8.2	7.4		0.62	0.55
0.5	100	5.7	8.2	7.6		0.54	0.48
2.0	100	5.6	8.2	7.3		0.48	0.51

Table 2.3. Number of open facilities. Budget set to 500. Averages over 10 runs.

		number of open facilities				ratio of common facilities		
θ	β	lin. approx.	$c=0$	$c=1$	$c=10$	$c=0$	$c=1$	$c=10$
0.2	50	3.8	5.7	5.7	2.7	0.54	0.54	0.28
0.5	50	4.1	5.9	5.7	2.8	0.45	0.40	0.21
2.0	50	3.8	5.9	5.6	2.7	0.51	0.50	0.26
0.2	100	3.5	5.7	5.5	3.0	0.50	0.50	0.40
0.5	100	3.7	5.7	5.6	2.8	0.54	0.55	0.35
2.0	100	3.7	5.7	5.5	3.0	0.54	0.54	0.33

Table 2.4. Number of open facilities. Budget set to 250. Averages over 10 runs.

fact that determining the optimal facility locations is hard, and that solutions of similar values can vastly differ in their topologies.

Although Table 2.2 suggests that heuristics do not perform very well when the budget is small, a closer inspection reveals that for some values of c, they yield results close to those of the linear approximation method, as shown in Table 2.5, where the best results among those run for values of c ranging from 0 to 10 are displayed. The best results were usually related to low values of c. In this setting, heuristics manage to capture between 90% and 95% of the number of customers obtained by the linear approximation method, at a much lower computational cost.

In Table 2.6, we report the impact of c on the number of facilities opened, as well as on the number of served customers, for 3 randomly chosen tests in our dataset. We vary the c from 0 to 10, and report the best solution found for each test. We then compute the average ratio between the latter and the optimum found by the linear approximation method. As c

		heuristic over	total
θ	β	lin. approx. ratio	CPU (sec.)
0.2	50	0.95	133
0.5	50	0.96	86
2.0	50	0.95	69
0.2	100	0.90	132
0.5	100	0.92	95
2.0	100	0.88	75

Table 2.5. Parameter c runs from 0 to 10. Budget set to 250.

increases, more importance is given to μ, and less budget is available for opening facilities. A second trend is the concave-like behaviour (increasing, levelling, decreasing) of the number of served customers with respect to c.

	# of open facilities			served customers		
c	test 1	test 2	test 3	test 1	test 2	test 3
0	6	6	6	113.92	124.70	122.57
1	6	6	6	113.92	124.88	122.64
2	6	6	6	114.82	124.95	123.29
3	6	6	6	115.94	122.55	123.33
4	4	5	7	99.39	119.54	123.75
5	3	4	6	84.42	98.82	124.07
6	3	1	6	84.42	45.70	123.99
7	2	1	5	74.21	45.70	116.87
8	1	1	4	53.64	45.70	106.71
9	1	1	3	53.64	45.70	90.48
10	1	1	1	53.64	45.70	62.30

Table 2.6. Sensitivity of analysis with respect to c in formula (2.4.1).

Finally, we assess the performance of the heuristics, given an optimal set of open facilities provided by the linear approximation method. Restricted to the determination of service levels, the problem remains a hard nonlinear bilevel program. All tests have been performed on the same aforementioned dataset, using 10 samples for x and 9 for λ and μ. The results are displayed in Table 2.7, where we observe a sharp improvement with respect to the case where

facility locations are decision variables (Table 2.2). More precisely, due to approximation errors, the linear approximation method was outperformed by the theoretically suboptimal heuristics on some instances.

		heuristic over lin. approx. ratio			
θ	β	$c = 0$	$c = 1$	$c = 10$	best
0.2	50	1.02 (5)	1.02 (4)	0.84 (1)	1.02
0.5	50	1.02 (6)	1.01 (2)	0.86 (2)	1.02
2.0	50	1.02 (6)	1.00 (4)	0.83 (1)	1.02
0.2	100	1.01 (7)	1.00 (3)	0.88 (1)	1.02
0.5	100	1.02 (5)	1.00 (5)	0.89 (0)	1.02
2.0	100	1.02 (8)	1.00 (3)	0.89 (3)	1.02

Table 2.7. Heuristics run from facility locations provided by the linear approximation method. Budget set to 250. Within parentheses: number of instances for which the corresponding value of c yielded the best result. The sum of values exceed in some cases the total number of tests, as sometimes, different heuristics yield the same optimum.

Accuracy of linearization

In order to measure the impact of the number of sample points involved in the approximation of the nonlinear functions $\tilde{W}$ and $\tilde{P}_K$, we varied λ and μ for values ranging from 1 to 10, for a step of 0.1. We then computed the difference between W and P_K, and their linearized counterparts across this fine-grained domain. Note that, due to nonconvexity in the vicinity of the origin (see Figure 2.2), the tangents in this area can be very steep and thus wildly overestimate the true value of the function. For this reason, linearization sample points were not selected close to 0. As observed in Table 2.8, increasing the number of sample points can actually worsen the approximation, due to non-convexity of the original functions. The way around this issue would be to make *nonconvex* piecewise linear approximations, the drawback being the addition of a significant number of binary variables that would yield a sharp increase in the running time of the algorithm. When selecting a number of samples, one has indeed to achieve a trade-off between the error on W, on P_K, the running time and the quality of the solution.

# of samples		Error (average)		# of samples		Error (average)	
R (on λ)	P (on μ)	W	P_K	R (on λ)	P (on μ)	W	P_K
3	3	1.34	0.29	7	3	2.10	0.20
3	5	1.33	0.36	7	5	2.00	0.42
3	7	1.77	0.38	7	7	2.17	0.42
3	10	2.94	0.41	7	10	5.51	0.42
5	3	1.13	0.38	10	3	2.05	0.26
5	5	1.24	0.41	10	5	2.00	0.43
5	7	2.67	0.41	10	7	3.36	0.43
5	10	4.37	0.42	10	10	3.18	0.43

Table 2.8. Linearization error for the waiting time and probability of balking. $K = 10$.

# of samples			CPLEX			recovered no of	estimated no of
N (on x)	R (on λ)	P (on μ)	CPU limit(s)	CPU(s)	gap(%)	served customers	served customers
2	2	2	1,000	562	9.71	88.15	80.35
5	3	3	2,000	829	0.92	97.89	100.65
7	3	3	5,000	1,057	0.97	98.33	100.65
7	5	5	7,000	5,752	0.94	102.24	103.42
10	5	5	10,000	8,856	7.78	100.66	103.81
10	7	7	15,000	12,478	1.14	104.91	106.80
12	7	7	20,000	16,921	16.56	94.13	93.69

Table 2.9. Number of attracted and served customers. $K = 10$, $\theta = 0.2$, $\beta = 50$.

At last, we investigate the impact of sample size on the quality of the optimal solution of the generated MILP. Surprisingly (see Table 2.9), this impact is almost negligible, and the objective can actually decrease when the sample size increases. This counterintuive phenomenon was also observed in the paper of **[Marcotte et al., 2013]**, and also in **[Marcotte, 1986]** for a bilevel pricing model where a probability density function was approximated by a coarse-grained histogram. Although the non-convexity of W_j and P_{Kj} certainly plays a role, we could not theoretically devise a rule for selecting 'optimal' samples. This behaviour can also be explained by factors such as travel time. For instance, if a facility is located far from a demand point, a small error in the waiting time will not significantly impact the number of arriving customers. Another reason is the non-convexity

of the approximated functions W_j and P_{Kj}. In this context, a larger number of ill-positioned samples might not necessarily imply a tighter, more precise approximation. As observed in Table 2.9, the value of the objective function estimated by the approximate model does not correlate well with the actual optimal value obtained by performing an assignment of users with respect to the service rate vector μ. Note that when the (N, R, P) triplets were set to (10, 7, 7) and (12, 7, 7) CPLEX was unable to find a feasible solution in the alloted time, in 3 out of 10 tests. Since the true number of attracted and served customers is quite insensitive to the number of samples, it is clearly advantageous to set those number to values as small as possible, but yet not too small.

An illustrative case

In the province of Québec (Canada), walk-in clinics provide professional assessment and treatment for minor illnesses or injuries, for the quarter of the population that lacks a family doctor, as reported in Statistics [**hea, 2017**]. These clinics often function on a first-come first-served basis and it is frequent that clients balk to avoid long waiting times. In this section, we focus on the issue of optimizing the location and service rates of clinics in the Mont-Tremblant area, with the aim of maximizing the number of patients served by the clinics.

	Number of open facilities								
	budget=15			budget=20			budget=25		
θ	$\beta = 10$	$\beta = 50$	$\beta = 100$	$\beta = 10$	$\beta = 50$	$\beta = 100$	$\beta = 10$	$\beta = 50$	$\beta = 100$
0.01	2	2	2	2	3	3	3	3	3
0.1	2	2	2	2	2	2	3	3	2
0.2	2	2	2	2	2	2	2	2	2
0.5	2	2	2	2	2	2	2	2	2

Table 2.10. Parametric analysis on θ, β and the budget.

Mont-Tremblant has 17 population zones, to which we assign demand nodes assumed to be spatially located in the center of each zone. The population count per demand node is generated as follows. The initial population data is taken from Statistics [**Census, 2016**], out of which only 25.2% would be interested to visit a walk-in clinic. Considering 250 days

a year, 8 hours a day, and an average of 4 doctor visits per year, per person, the hourly demand count represents 0.05% of the initial population.

There are already 4 medical clinics (the competition) in Mont-Tremblant that we consider serving on average between 1 and 3 clients per hour. Assuming the balking threshold at 10 (people balk if there are 10 or more people waiting in line), and a fixed cost/variable cost ratio of 5:1, we perform a parametric analysis on β, θ and the budget see Table 2.10.

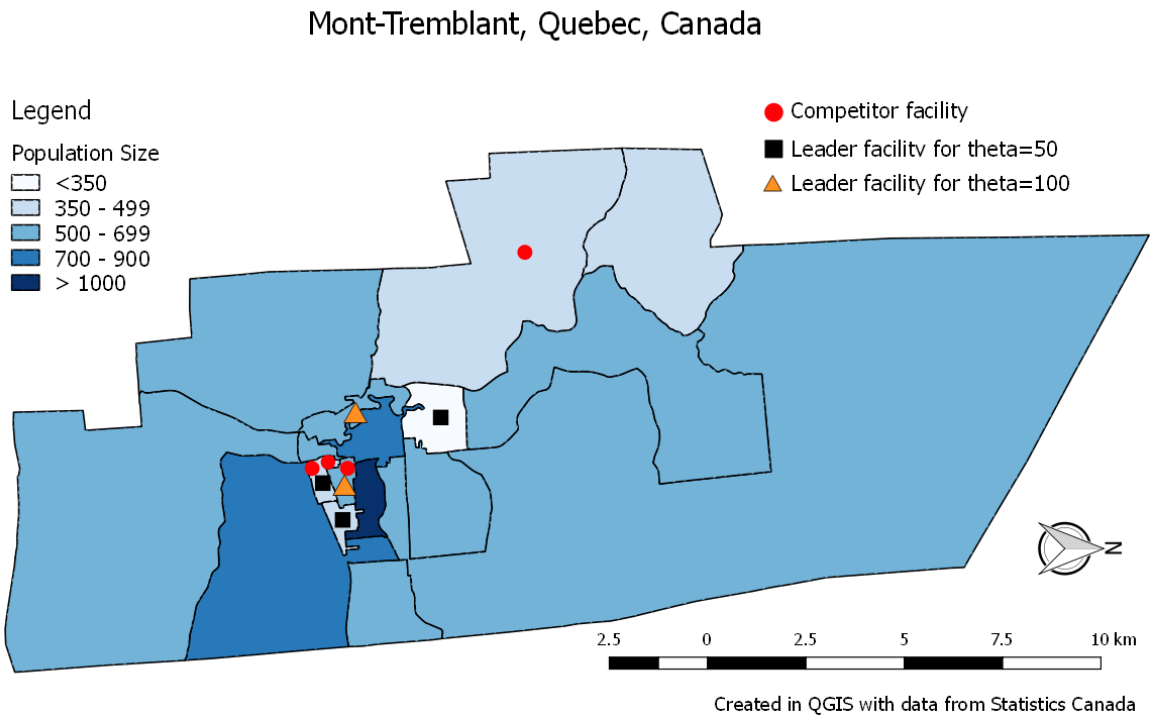


Figure 2.7. Population Map of Mont-Tremblant, Qc, Canada

Note that for small values of θ, the number of open facilities increases with the budget, which is expected. For higher values of θ, only two facilities are open, regardless of the increase in the budget. When θ is close to 0, clients choose facilities with almost no regard to their disutility. When θ is higher, the clinic must ensure low waiting time and probability of balking. For instance, when $\theta = 0.1$, it opens 3 facilities for $\beta = 10$ and only 2 when $\beta = 100$, for a budget of 25. When clients are not much impacted by the probability of balking (e.g. $\beta = 10$), more money can be spent in opening new clinics. On the other hand, when $\theta = 100$, only two facilities are opened, while the bulk of the budget is spent on service.

Figure 2.7 displays the spatial distribution of the facilities. The main observation is that facilities are opened adjacent to the highly populated areas, but not within them.

This illustrates the difficulty of determining the righ locations. We also note that emerging facilities lie close to the competitor's facilities.

2.5. Conclusion and extensions

In this paper, we addressed a bilevel location model involving both combinatorial and nonlinear elements, and proposed for its solution approximation schemes, as well as a heuristic that exploits the problem's structure. Our model is flexible and can accommodate numerous situations, while the proposed algorithms remain applicable. For instance, balking is an additional feature that can be removed if it is not suited for a given application. The budget constraint can be incorporated in the objective, as a total setup cost. For the sake of simplicity, only one server is available in our model, however, any $M/M/s/K$, and $M/M/s$ queues can be considered, provided that the number of server s is fixed, and the decision variable is the service rate μ. In this case the waiting time and probability of balking formulas would change, but they do not hinder the algorithm.

While the results are more than encouraging, our findings raise a number of issues, from either the modelling, theoretical or algorithmic viewpoints. For instance, the surprising result that the standard linearization of the lower level complementarity constraints proved less efficient, numerically, than an approach based on a triangular approximation involving a larger number of binary variables, is certainly worth investigating.

On the modelling side, future work will integrate features such as variable demand and the possibility of either increasing or decreasing the service rates of existing facilities. This will involve a piecewise affine investment function whose two slopes reflect the fact that economies resulting from lowering service are less than those of increasing it. More realistic models where the price of service depends on location should also be considered.

On the algorithmic side, three avenues can be pursued: (i) the design of improved approximations for the nonlinear terms involved in the linear approximation method, and (ii) the design of fast heuristics for determining good sets of facility locations, from which efficient methods for determining optimal service rates can be initiated and, finally (iii) the investigation of approximations based on the exact mixed integer formulation of the logit-based location models proposed by [**Haase, 2009**], [**Benati and Hansen, 2002**], [**Zhang et al., 2012**], and numerically analyzed by [**Haase and Müller, 2014**].

Acknowledgement: This research was supported by NSERC grant 5789-2011 RGPIN.

Appendix

2.A. Notation and proofs

In this e-companion we present the notation used throughout this paper, and we complete the proofs of some propositions.

2.B. Notation

	Sets
I:	set of demand nodes;
J:	set of candidate facility locations (leader and competition);
J_c:	set of competition's facilities;
J_1:	set of leader's candidate sites;
$J_1^* \subseteq J_1$:	set of leader's open facilities
$J^* \subseteq J$:	set of open facilities (leader and competitor).
	Parameters
d_i:	demand originating from node $i \in I$;
t_{ij}:	travel time between nodes $i \in I$ and $j \in J$;
α:	coefficient of the waiting time in the disutility formula;
β:	coefficient of the balking probability in the disutility formula;
B:	available budget (for opening new facilities and associated service rates);
c_f:	fixed cost associated with opening a new facility;
c_μ :	cost per unit of service;
$\bar{\mu}$:	maximum service rate allowed by the budget;
p:	number of facilities to open.

Basic decision variables

y_j:	binary variable set to 1 if a facility is open at site j, and to 0 otherwise;
μ_j:	service rate at open facilities.

Additional variables

x_{ij}:	arrival rate at at facility $j \in J$ originating from demand node $i \in I$;
λ_j:	arrival rate at node $j \in J$;
ρ_j:	utilization rate of facility $j \in J$;
$\bar{\lambda}_j$:	throughput rate (customers accessing service) at node $j \in J$;
w_j:	mean queueing time at facility j.

2.C. Proofs of Propositions 1, 2, 4, 5, 6, 7, 10, 11, 12 and Theorem 9

Proposition 1. The waiting time w_j is increasing in λ_j.

PROOF. Proof. The derivative of w_j with respect to λ_j (see Equation (2.2.10)) is

$$\frac{\partial w_j}{\partial \lambda_j} = \frac{\partial w_j}{\partial \rho_j}\frac{\partial \rho_j}{\partial \lambda_j} = \frac{\partial w_j}{\partial \rho_j}\frac{1}{\mu_j}.$$

To show that $\partial w_j/\partial \rho_j$ is nonnegative for all $\rho_j \neq 1$, let us consider

$$\frac{\partial w_j}{\partial \rho_j} = \frac{1}{\mu_j}\left(-\frac{K^2\rho_j^K - 1}{\left(\rho_j^K - 1\right)^2} + \frac{1}{\left(\rho_j - 1\right)^2}\right).$$

Basic algebraic manipulation yields

$$\frac{1}{\left(\rho_j - 1\right)^2} \geq \frac{K^2\rho_j^K - 1}{\left(\rho_j^K - 1\right)^2} \Longleftrightarrow \sum_{i=0}^{K-1} \rho_j^i \geq K\rho_j^{(K-1)/2}. \tag{2.C.1}$$

To prove that the right-hand inequality holds true, we consider two cases.

If K is odd:

$$\begin{aligned}\sum_{i=0}^{K-1} \rho_j^i &= \sum_{i=0}^{(K-1)/2-1} \left(\rho_j^i + \rho_j^K - 1 - i\right) + \rho_j^{(K-1)/2} \\ &\geq 2\sum_{i=0}^{(K-1)/2-1} \rho_j^{(K-1)/2} + \rho_j^{(K-1)/2} = K\rho_j^{(K-1)/2}.\end{aligned}$$

If K is even:

$$\sum_{i=0}^{K-1} \rho_j^i = \sum_{i=0}^{(K-2)/2} \left(\rho_j^i + \rho_j^K - 1 - i \right) \geq 2 \sum_{i=0}^{(K-2)/2} \rho_j^{(K-1)/2} = K \rho_j^{(K-1)/2}.$$

It follows that w_j is an increasing function of λ_j. □

Proposition 2. The probability of balking p_{Kj} is increasing in λ_j.

PROOF. Proof. The derivative of p_{Kj} with respect to λ_j is

$$\begin{aligned} p'_{Kj} &= \frac{\lambda_j^K - 1 \mu_j}{\left(\lambda_j^K + 1 - \mu_j^K + 1 \right)^2} \left[\lambda_j^K + 1 - (K+1) \lambda_j \mu_j^K + K \mu_j^K + 1 \right] \\ &= \sigma [x^K + 1 - (K+1)\, x + K], \end{aligned}$$

where σ is a positive number and $x = \lambda_j / \mu_j$. By differentiating with respect to x, we find that the right-hand-side achieves its minimum value 0 at $x = 1$, which concludes the proof. □

Proposition 4. When $K = \infty$, i.e., balking does not occur (in this case, the model admits a solution only if the total service rate exceeds the total demand rate), the lower level objective function is convex jointly in λ and μ.

PROOF. Proof. If $K = \infty$, the probability of balking can be removed from the objective, since it is equal to 0. Moreover, $w_j = 1/(\mu_j - \lambda_j)$, and the lower level objective takes the form

$$\sum_{i \in I} \sum_{j \in J^*} \left[\frac{1}{\theta} x_{ij} \ln x_{ij} + x_{ij} t_{ij} \right] - \alpha \sum_{j \in J^*} \ln(\mu_j - \lambda_j).$$

Basic algebra shows that its Hessian is positive semidefinite, hence the function is convex. □

Proposition 5. The integral of the waiting time, $W_j(\lambda_j, \mu_j)$ is pseudoconvex.

PROOF. Proof. Let $x = (\lambda_x, \mu_x)$ and $y = (\lambda_y, \mu_y)$. Assume that $\nabla W(x)(y - x) \geq 0$. Then we have:

$$\begin{aligned} & \left(w_j(x), -\frac{\lambda_x}{\mu_x} w_j(x) \right) (\lambda_y - \lambda_x, \mu_y - \mu_x) \geq 0 \\ \Rightarrow \quad & (\rho_y - \rho_x) w_j(x) \geq 0 \\ \Rightarrow \quad & \rho_y \geq \rho_x, \end{aligned} \tag{2.C.2}$$

since w_j is nonnegative. On the other hand, $\partial W_j/\partial\rho = \mu_j w_j$ is nonnegative, we have that W_j is increasing in ρ, so $\rho_y \geq \rho_x \Rightarrow W_j(y) \geq W_j(x)$. From Eq (2.C.2) it follows that if $\nabla W(x)(y-x) \geq 0$ then $W_j(y) \geq W_j(x)$, hence W_j is pseudoconvex. □

Proposition 6. G_1 is strongly monotone in x of modulus $\theta \cdot d_{\text{MAX}}$.

PROOF. Proof. **[Gilbert et al., 2015]** have already argued that G_1 is strongly monotone. Indeed, the associated Jacobian is a positive definite diagonal matrix over D, with the smallest possible eigenvalue $1/(\theta \cdot d_{\text{MAX}})$. It follows that G_1 is strongly monotone with modulus $\theta \cdot d_{\text{MAX}}$. □

Proposition 7. G_2 is monotone in x.

PROOF. Proof.

$$\begin{aligned}
\langle G_2(\mu,x) - G_2(\mu,y), x-y\rangle &= \sum_{i\in I}\sum_{j\in J^*}\left(\frac{1}{\mu_j - \sum_{l\in I} x_{l,j}} - \frac{1}{\mu_j - \sum_{l\in I} y_{l,j}}\right)\cdot(x_{ij}-y_{ij}) \\
&= \sum_{j\in J^*}\left[\frac{\mu_j - \sum_{l\in I} y_{l,j} - \mu_j + \sum_{l\in I} x_{l,j}}{\left(\mu_j - \sum_{l\in I} x_{l,j}\right)\cdot\left(\mu_j - \sum_{l\in I} y_{l,j}\right)}\cdot\sum_{i\in I}(x_{ij}-y_{ij})\right] \\
&= \sum_{j\in J^*}\left[\frac{\sum_{l\in I}(x_{l,j}-y_{l,j})\sum_{l\in I}(x_{l,j}-y_{l,j})}{\left(\mu_j - \sum_{l\in I} x_{l,j}\right)\cdot\left(\mu_j - \sum_{l\in I} y_{l,j}\right)}\right] \\
&\geq 0
\end{aligned}$$

□

Proposition 10. If $K = \infty$ and there are no fixed costs, the surrogate model is convex.

PROOF. Proof. According to Proposition 4, the objective is jointly convex in μ and λ. Moreover one can, without loss of generality, open all facilities and hence dispense with the binary vector y. Notwithstanding, a facility can be closed by setting its service level to zero. □

Proposition 11. At the optimum of (PH*), if $K = \infty$, queueing delays are equal for all leader's

PROOF. Proof. For fixed y variables, Equation (2.3.27) can be rewritten as

$$\sum_{j \in J^*} \mu_j \leq \bar{\mu}, \tag{2.C.3}$$

where $\bar{\mu}$ is the maximum possible total service rate allowed by the budget. But $K = \infty$, so $w_j(\lambda_j, \mu_j) = 1/(\mu_j - \lambda_j)$ and $p_{Kj}(\lambda_j, \mu_j) = 0$, which yields the mathematical program

$$\begin{aligned} \text{(PHY*)} \qquad \min_{\mu,\, x} \quad & \sum_{i \in I} \sum_{j \in J^*} x_{ij} t_{ij} - \alpha \sum_{j \in J^*} \ln(\mu_j - (\sum_{i \in I} x_{ij})) \\ \text{s.t.} \quad & \text{constraints (2.3.26), (2.3.29), (2.3.30), (2.C.3)} \end{aligned}$$

Let δ_i, π_{ij} and γ be the Lagrange multipliers associated with Equations (2.3.26), (2.3.30) and (2.C.3), respectively. Variables δ_i are free, while γ and π_{ij} are restricted to be nonnegative. The stationarity conditions of the above program are:

$$\frac{\partial L}{\partial x_{ij}} = 0 \;\Rightarrow \qquad t_{ij} + \alpha w_j(\lambda_j, \mu_j) - \delta_i - \pi_{ij} = 0, \;\forall i \in I, \forall j \in J^* \tag{2.C.4}$$

$$\frac{\partial L}{\partial \mu_j} = 0 \;\Rightarrow \qquad -\alpha w_j(\lambda_j, \mu_j) + \gamma = 0, \;\forall j \in J^* \cap J_1, \tag{2.C.5}$$

and the conclusion follows from Equation (2.C.5). □

We observe, after plugging $\alpha w_j(\lambda_j, \mu_j)$ from Equation (2.C.5) into Equation (2.C.4) for a given demand node i, that only one flow x_{ij} is nonzero, provided that transportation times to the leader's facilities are distinct.

Proposition 12. There exists a value of ξ^* for which (PHY*(ξ^*)) yields an optimal solution for (P*).

PROOF. Proof. Let y^* and μ^* be optimal for (P*). Without loss of generality (there are no fixed costs) we assume that all facilities are open. At equilibrium, let c_i^* be the cost associated with demand node i and optimal service rate μ^*. Let x^*, $w_j(x^*, \mu_j^*)$ and c_i^* satisfy Equation (2.2.24) and (2.2.25). If x_{ij} is positive, we have:

$$t_{ij} + \alpha w_j(x^*, \mu^*) = c_i^*, \quad \forall j \in J, \forall i \in I. \tag{2.C.6}$$

Let $C = \max_{i \in I} \{c_i^*\}$ in the initial formulation. For $j \in J$, we let $\xi_j = c_i^* - t_{ij} - C$ and select and index i corresponding to a positive flow x_{ij}^*. If no such i exists, then $\mu_j^* = 0$, otherwise the leader would waste monetary resources. We then set $\xi_j = -C$.

Now, let δ_i, π_{ij} and γ be the Lagrange multipliers associated with Equations (2.3.26), (2.3.30) and (2.C.3), respectively. Variables δ_i and γ are free, while π_{ij} are restricted to be nonnegative. The stationarity conditions of the program above take the form

$$\frac{\partial \mathcal{L}}{\partial x_{ij}} = 0 \;\Rightarrow \qquad t_{ij} + \alpha w_j(x, \mu_j) - \delta_i = 0, \;\; \text{if } x_{ij} > 0, \; \forall i \in I, \forall j \in J \tag{2.C.7}$$

$$\frac{\partial \mathcal{L}}{\partial \mu_j} = 0 \;\Rightarrow \qquad -\alpha w_j(x, \mu_j) + \gamma + \xi_j = 0, \; \forall j \in J_1. \tag{2.C.8}$$

Note that the derivative of the Lagrangian with respect to x_{ij} is left unchanged, i.e., Equation (2.C.7) is equivalent to Equation (2.C.4). If $\gamma = C$, we derive from Equation (2.C.8) that $\alpha w_j(x, \mu_j) = c_i^* - t_{ij}$, which is equivalent to Equation (2.C.7). This completes the proof, since for the given values of ξ, variables x and μ match the optimal solution of (P*). □

Theorem 9. The error of the upper-level objective function is $\mathrm{O}(1/N_1 + 1/N_2)$, where N_1 and N_2 are the number of samples for the linearization of g_1 and g_2, respectively.

Proof. Proof. Let $\bar{G}$ be an approximation of G. We denote by $\bar{x}$ the solution of $\mathrm{IV}(\bar{G}(\mu, \cdot), D)$, and by x the solution of $\mathrm{IV}(G(\mu, \cdot), D)$. Then the following inequalities hold:

$$\begin{aligned} \langle G(\mu, x), \bar{x} - x \rangle &\geq 0 \\ \left\langle \bar{G}(\mu, \bar{x}), x - \bar{x} \right\rangle &\geq 0 \\ \Rightarrow \left\langle \bar{G}(\mu, \bar{x}) - G(\mu, x), x - \bar{x} \right\rangle &\geq 0 \end{aligned} \tag{2.C.9}$$

From the strong monotonicity of G and Eq. (2.C.9) it follows that

$$\left\langle \bar{G}(\mu, \bar{x}) - G(\mu, \bar{x}), x - \bar{x} \right\rangle \geq \frac{1}{\theta \cdot d_{\mathrm{MAX}}} ||x - \bar{x}||^2. \tag{2.C.10}$$

Applying the Cauchy-Schwarz inequality, we obtain

$$\theta \cdot d_{\mathrm{MAX}} \cdot ||\bar{G}(\mu, \bar{x}) - G(\mu, \bar{x})|| \geq ||x - \bar{x}||. \tag{2.C.11}$$

It follows that

$$|f(x) - f(\bar{x})| = |\sum_{i \in I} \sum_{j \in J_1^*} (x_{ij} - \bar{x}_{ij})| \leq \sqrt{|I| \cdot |J|}||x - \bar{x}|| \quad \text{(Cauchy-Schwarz inequality)}$$

$$\leq \sqrt{|I| \cdot |J|}\ \theta \cdot d_{\text{MAX}} \cdot ||\bar{G}(\mu, \bar{x}) - G(\mu, \bar{x})||. \tag{2.C.12}$$

We perform two separate linear approximations on g_1 and g_2, respectively. Then the mappings $\bar{G}_1(x)$ and $\bar{G}_2(x)$ are piecewise constant approximations, that we detail separately.

A. $\bar{G}_1$: Each component (i, j) of this vector is a piecewise constant approximation of $\log(x_{ij})$, satisfying:

i) there are N_1 total samples on x_{ij}, starting from $r_{\min}$ to d_{MAX};

ii) the sampling points are chosen so that the segments are vertically equidistant;

iii) the vertical positions of the segments are the slopes of the tangents to $x \log(x)$, evaluated at the sampling points.

Let Δ_1 be the difference between two consecutive slope values:

$$\Delta_1 = \frac{\log(d_{\text{MAX}}) - \log(r_{\min})}{N_1 - 1}.$$

Then $|\bar{G}_{1(i,j)} - G_{1(i,j)}| \leq \Delta_1$, which yields:

$$||\bar{G}_1(x) - G_1(x)|| = \sqrt{\sum_{i \in I} \sum_{j \in J^*} |\bar{G}_{1(i,j)} - G_{1(i,j)}|^2} \leq \frac{(\log(d_{\text{MAX}}) - \log(r_{\min}))\ \sqrt{|I| \cdot |J|}}{N_1 - 1} \tag{2.C.13}$$

B. $\bar{G}_2$: is a mapping whose (i, j)-component is a constant piecewise approximation of $1/q_j$, where $q_j = \mu_j - \sum_{i \in I} x_{ij}$. Similar to $\bar{G}_1$, this linearization satisfies the following:

i) there are N_2 total samples, starting from ψ to μ_{MAX};

ii) the sampling points are chosen so that the segments are vertically equidistant;

iii) the vertical positions of the segments are the slopes of the tangents $-\log(q)$ evaluated at the sampling points.

We note by Δ_2 the difference between two consecutive slope values:

$$\Delta_2 = \frac{\dfrac{1}{\psi} - \dfrac{1}{\mu_{\text{MAX}}}}{N_2 - 1}$$

Then $|\bar{G}_{2(i,j)} - G_{2(i,j)}| \leq \Delta_2$, which yields

$$||\bar{G}_2(x) - G_2(x)|| = \sqrt{\sum_{i\in I}\sum_{j\in J^*} |\bar{G}_2(x_{ij}) - G_2(x_{ij})|^2} \leq \frac{\left(\frac{1}{\psi} - \frac{1}{\mu_{\text{MAX}}}\right)\sqrt{|I|\cdot|J|}}{N_2 - 1} \tag{2.C.14}$$

From Eq. (2.C.12) it follows that, given y and μ:

$$|f(x) - f(\bar{x})| \leq \theta \cdot d_{\text{MAX}}|I|\cdot|J| \left[\frac{(\log(d_{\text{MAX}}) - \log(r_{\min}))}{N_1 - 1} + \alpha\frac{\frac{1}{\psi} - \frac{1}{\mu_{\text{MAX}}}}{N_2 - 1}\right] \in O(\frac{1}{N_1} + \frac{1}{N_2}). \tag{2.C.15}$$

□

Theorem 9 has several implications.

— For a given set of open facilities, the absolute difference between the optimal and the approximated objective value is bounded by the right-hand-side of inequality (2.C.15). For large values of N_1 and N_2, the two values are very close.

— If the optimal solution is unique in terms of the location vector y, and the absolute difference between the objective and other solutions objectives are lower than the right hand side of inequality (2.C.15), the approximation algorithm will find the optimum locations.

2.D. Linearization of optimality conditions

2.D.1. Complementarity constraints for Program (P2-lin)

Let γ_i, δ_j, ν^n_{ij}, π^{rp}_j, η^{rp}_j, and ϕ_{ij} be the dual variables associated with constraints (2.3.8), (2.3.9), (2.3.10), (2.3.11), (2.3.12) and (2.3.13), respectively. Then the complementarity constraints for program (P2-lin) can be written as:

$$\gamma_i\left(\sum_{j\in J^*} x_{ij} - d_i\right) = 0 \qquad \forall i \in I \tag{2.D.1}$$

$$\delta_j\left(\lambda_j - \sum_{i\in I} x_{ij}\right) = 0 \qquad \forall j \in J^* \tag{2.D.2}$$

$$\nu^n_{ij}\left(v_{ij} - a^n_f x_{ij} - b^n_f\right) = 0 \qquad \forall i \in I;\ \ \forall j \in J^*;\ \ \forall n \in N \tag{2.D.3}$$

$$\pi^{rp}_j\left(u_j - a^{rp}_g\lambda_j - b^{rp}_g\mu_j - c^{rp}_g\right) = 0 \qquad \forall j \in J^*;\ \ \forall r \in R;\ \ \forall p \in P \tag{2.D.4}$$

$$\eta_j^{rp}\left(z_j - a_h^{rp}\lambda_j - b_h^{rp}\mu_j - c_h^{rp}\right) = 0 \qquad \forall j \in J^*;\ \ \forall r \in R;\ \ \forall p \in P \qquad (2.D.5)$$

$$\phi_{ij}x_{ij} = 0 \qquad \forall i \in I;\ \ \forall j \in J^*, \qquad (2.D.6)$$

and can be linearized in the standard fashion, through the introduction of binary variables and big-M constants. For instance, the last constraint is replaced by the inequalities

$$\begin{aligned} \phi_{ij} &\leq Mu_{ij} \\ x_{ij} &\leq M(1-u_{ij}), \end{aligned}$$

where $u_{ij} \in \{0,1\}$. Although is possible to find a valid upper bound for the variable ϕ_{ij}, a large value of M is required, which leads to a poor relaxation and consequently an ill-behaved branch-and-bound algorithm.

2.D.2. Equality between primal and dual objectives

Alternatively, constraints (2.D.1) – (2.D.6) can be replaced with constraint (2.D.7), which represents the equality between between the primal and dual objective of (P2-lin). Then the optimality constraints of (P2-lin) are

$$\begin{aligned} &\sum_{i\in I}\gamma_i d_i + \sum_{n\in N}\sum_{i\in I}\sum_{j\in J}\nu_{ij}^n b_f^n \ + \sum_{r\in R}\sum_{p\in P}\sum_{j\in J}\left(b_g^{rp}\mu_j\pi_j^{rp} + b_h^{rp}\mu_j\eta_j^{rp} + c_g^{rp}\pi_j^{rp} + c_h^{rp}\eta_j^{rp}\right) \\ &= \sum_{i\in I}\sum_{j\in J}\left[\frac{1}{\theta}v_{ij} + x_{ij}t_{ij}\right] + \alpha\sum_{j\in J}u_j + \beta\sum_{j\in J}z_j, \qquad (2.D.7) \\ &\sum_{j\in J}x_{ij} = d_i, && \forall i \in I \\ &\lambda_j = \sum_{i\in I}x_{ij}, && \forall j \in J \\ &v_{ij} - a_f^n x_{ij} \geq b_f^n, && \forall i \in I; \forall j \in J; \forall n \in N \\ &u_j - a_g^{rp}\lambda_j - b_g^{rp}\mu_j \geq c_g^{rp}, && \forall j \in J; \forall r \in R; \forall p \in P \\ &z_j - a_h^{rp}\lambda_j - b_h^{rp}\mu_j \geq c_h^{rp}, && \forall j \in J; \forall r \in R; \forall p \in P \\ &\gamma_i + \delta_j - \sum_{n\in N}a_f^n\nu_{ij}^n \leq t_{ij}, && \forall i \in I;\ \forall j \in J \\ &-\delta_j - \sum_{r\in R}\sum_{p\in P}\left(a_g^{rp}\pi_j^{rp} + a_h^{rp}\eta_j^{rp}\right) = 0, && \forall j \in J \end{aligned}$$

$$\sum_{n\in N} \nu_{ij}^n = \frac{1}{\theta}, \qquad \forall i \in I;\ \forall j \in J$$

$$\sum_{r\in R} \sum_{p \in P} \pi_j^{rp} = \alpha, \qquad \forall j \in J$$

$$\sum_{r\in R} \sum_{p\in P} \eta_j^{rp} = \beta, \qquad \forall j \in J$$

$$\pi_j^{rp}, \eta_j^{rp} \geq 0, \qquad \forall j \in J; \forall r \in R; \forall p \in P$$

$$x_{ij} \geq 0,\ \nu_{ij}^n \geq 0, \qquad \forall i \in I; \forall j \in J; \forall n \in N.$$

To obtain a MILP formulation, we linearize the nonlinear terms $\mu_j \pi_j^{rp}$ and $\mu_j \eta_j^{rp}$ via the triangle method described in [**D'Ambrosio et al., 2010**]. For each term $\mu_j \pi_j^{kq}$ we introduce $2(R-1)(P-1)$ binary variables $\overline{l}_{jrpkq}^{\pi}$ and $\underline{l}_{jrpkq}^{\pi}$ associated with the upper and lower triangles, respectively, of the rectangle defined by the intervals $[\pi^r, \pi^{r+1})$ and $[\mu^p, \mu^{p+1})$. Note that the values of π and η are upper bounded by α and β, respectively. Additionally, μ is bounded by the maximum value allowed by the leader's budget, $\bar{\mu}$. Next, we introduce J_1RP continuous variables $s_{jrpkq} \in [0,1]$ which will be used to express the couple (π_j^{kq}, μ_j) as a convex combination of triangle vertices. We introduce a similar linearization for the term $\mu_j \eta_j^{kh}$. The approximation of $\mu_j \pi_j^{kq}$ and $\mu_j \eta_j^{kq}$ is then

$$\sum_{r=1}^{R-1} \sum_{p=1}^{P-1} \left(\overline{l}_{jrpkq}^{\pi} + \underline{l}_{jrpkq}^{\pi} \right) = 1, \qquad \forall j \in J_1; \forall k \in R; \forall q \in P \tag{2.D.8}$$

$$s_{jrpkq}^{\pi} \leq \overline{l}_{jrpkq}^{\pi} + \underline{l}_{jrpkq}^{\pi} + \overline{l}_{jrp-1kq}^{\pi} + \underline{l}_{jr-1p-1kq}^{\pi} + \overline{l}_{jr-1p-1kq}^{\pi} + \underline{l}_{jr-1pkq}^{\pi}, \qquad \forall j \in J_1; \forall r \in R; \forall p \in P; \forall k \in R; \forall q \in P \tag{2.D.9}$$

$$\sum_{r=1}^{R} \sum_{p=1}^{P} s_{jrpkq}^{\pi} = 1, \qquad \forall j \in J_1; \forall k \in R; \forall q \in P \tag{2.D.10}$$

$$\pi_j^{kq} = \sum_{r=1}^{R} \sum_{p=1}^{P} s_{jrpkq}^{\pi} \pi^r, \qquad \forall j \in J_1; \forall k \in R; \forall q \in P \tag{2.D.11}$$

$$\mu_j = \sum_{r=1}^{R} \sum_{p=1}^{P} s_{jrpkq}^{\pi} \mu^p, \qquad \forall j \in J_1; \forall k \in R; \forall q \in P \tag{2.D.12}$$

$$e_{jkq}^{\pi} = \sum_{r=1}^{R} \sum_{p=1}^{P} s_{jrpkq}^{\pi} \pi^r \mu^p, \qquad \forall j \in J_1; \forall k \in R; \forall q \in P \tag{2.D.13}$$

$$\sum_{r=1}^{R-1}\sum_{p=1}^{P-1}\left(\overline{l}^{\eta}_{jrpkq}+\underline{l}^{\eta}_{jrpkq}\right)=1, \qquad \forall j\in J_1;\forall k\in R;\forall q\in P \tag{2.D.14}$$

$$s^{\eta}_{jrpkq}\le \overline{l}^{\eta}_{jrpkq}+\underline{l}^{\eta}_{jrpkq}+\overline{l}^{\eta}_{jrp-1kq}+\underline{l}^{\eta}_{jr-1p-1kq}\ +\overline{l}^{\eta}_{jr-1p-1kq}+\underline{l}^{\eta}_{jr-1pkq}, \qquad \forall j\in J_1;\forall r\in R;\forall p\in P;\forall k\in R;\forall q\in P \tag{2.D.15}$$

$$\sum_{r=1}^{R}\sum_{p=1}^{P}s^{\eta}_{jrpkq}=1, \qquad \forall j\in J_1;\forall k\in R;\forall q\in P \tag{2.D.16}$$

$$\eta^{kq}_{j}=\sum_{r=1}^{R}\sum_{p=1}^{P}s^{\eta}_{jrpkq}\eta^{r}, \qquad \forall j\in J_1;\forall k\in R;\forall q\in P \tag{2.D.17}$$

$$\mu_j=\sum_{r=1}^{R}\sum_{p=1}^{P}s^{\eta}_{jrpkq}\mu^{p}, \qquad \forall j\in J_1;\forall k\in R;\forall q\in P \tag{2.D.18}$$

$$e^{\eta}_{jkq}=\sum_{r=1}^{R}\sum_{p=1}^{P}s^{\eta}_{jrpkq}\eta^{r}\mu^{p}, \qquad \forall j\in J_1;\forall k\in R;\forall q\in P \tag{2.D.19}$$

The complete MILP formulation is presented below. It involves variables associated with the original fixed point (or bilevel) formulation (y, μ, x), together with variables issued from the linearizations and primal-dual optimality conditions.

(P-lin)

$$\begin{aligned}
\max_{\substack{x,y,\mu,\lambda,u,v,z,\\ e,\pi,\eta,\nu,\gamma,\delta,\overline{l}^{\pi},\underline{l}^{\pi},\\ s^{\pi},e^{\pi},s^{\eta},\overline{l}^{\eta},\underline{l}^{\eta}\\ s,\overline{l},\underline{l}}} \quad & \sum_{j\in J_1}e_j \\
& \sum_{j\in J_1}y_jc_f+\sum_{j\in J_1}c_\mu\mu_j\le B, \\
& \mu_j\le\bar{\mu}y_j, \qquad \forall j\in J_1 \\
& \sum_{r\in R}\sum_{p\in P}\sum_{j\in J_c}\left(b^{rp}_g\pi^{rp}_j\mu_j+b^{rp}_h\eta^{rp}_j\mu_j+c^{rp}_g\pi^{rp}_j+c^{rp}_h\eta^{rp}_j\right)\ +\sum_{n\in N}\sum_{i\in I}\sum_{j\in J}\nu^{n}_{ij}b^{n}_f \\
& +\sum_{r\in R}\sum_{p\in P}\sum_{j\in J_1}\left(b^{rp}_ge^{\pi}_{jrp}+b^{rp}_he^{\eta}_{jrp}+c^{mp}_g\pi^{rp}_j+c^{rp}_h\eta^{rp}_j\right)+\sum_{i\ \in\ I}\gamma_id_i \\
& \quad =\sum_{i\in I}\sum_{j\in J}\left[\frac{1}{\theta}v_{ij}+x_{ij}t_{ij}\right]+\alpha\sum_{j\in J}u_j+\beta\sum_{j\in J}z_j,
\end{aligned}$$

$$\sum_{j\in J} x_{ij} = d_i, \qquad \forall i \in I$$

$$\lambda_j = \sum_{i\in I} x_{ij}, \qquad \forall j \in J$$

$$v_{ij} - a_f^n x_{ij} \geq b_f^n, \qquad \forall i \in I; \forall j \in J; \forall n \in N$$

$$u_j - a_g^{rp}\lambda_j - b_g^{rp}\mu_j \geq c_g^{rp}, \qquad \forall j \in J; \forall r \in R; \forall p \in P$$

$$z_j - a_h^{rp}\lambda_j - b_h^{rp}\mu_j \geq c_h^{rp}, \qquad \forall j \in J; \forall r \in R; \forall p \in P$$

$$\gamma_i + \delta_j - \sum_{n\in N} a_f^n \nu_{ij}^n \leq t_{ij}, \qquad \forall i \in I;\ \forall j \in J$$

$$-\delta_j - \sum_{r\in R}\sum_{p\in P}\left(a_g^{rp}\pi_j^{rp} + a_h^{rp}\eta_j^{rp}\right) = 0, \qquad \forall j \in J$$

$$\sum_{n\in N} \nu_{ij}^n = \frac{1}{\theta}, \qquad \forall i \in I;\ \forall j \in J$$

$$\sum_{r\in R}\sum_{p\in P} \pi_j^{rp} = \alpha, \qquad \forall j \in J$$

$$\sum_{r\in R}\sum_{p\in P} \eta_j^{rp} = \beta, \qquad \forall j \in J$$

constraints (2.D.8)–(2.D.19) and (2.3.14)–(2.3.19),

$$y_j \in \{0,1\},\ \mu_j, \pi_j^{rp}, \eta_j^{rp} \geq 0, \qquad \forall j \in J; \forall r \in R; \forall p \in P$$

$$x_{ij} \geq 0,\ \nu_{ij}^n \geq 0, \qquad \forall i \in I; \forall j \in J; \forall n \in N.$$

2.D.3. Example of lower level linearization when $K = \infty$

Recall that, according to Proposition 4, the function is convex if the buffer zone is infinite (no balking). In that situation, the maximum of the linear approximations is consistent with the original function, give or take the approximation error. Proceeding as before, we obtain

$$g^{rp}(\lambda,\mu) = a_g^{rp}\lambda + b_g^{rp}\mu + c_g^{rp} = \frac{\alpha}{\mu^p - \lambda^r}\lambda - \frac{\alpha}{\mu^p - \lambda^r}\mu - \alpha(\ln(\mu^p - \lambda^r) - 1). \qquad (2.D.20)$$

This yields the linearized lower level program

$$(\mathrm{P2}^\infty) \quad \min_{x,v,u,\lambda} \quad \sum_{i\in I}\sum_{j\in J^*}\left[\frac{1}{\theta}v_{ij} + x_{ij}t_{ij}\right] + \alpha\sum_{j\in J^*} u_j \qquad (2.D.21)$$

$$\text{s.t.} \quad \sum_{j\in J^*} x_{ij} = d_i, \qquad \forall i \in I \qquad (2.D.22)$$

$$\lambda_j = \sum_{i \in I} x_{ij}, \qquad \forall j \in J^* \tag{2.D.23}$$

$$v_{ij} - a_f^n x_{ij} \geq b_f^n, \qquad \forall i \in I; \forall j \in J^*; \forall n \in N \tag{2.D.24}$$

$$u_j - a_g^{rp} \lambda_j - b_g^{rp} \mu_j \geq c_g^{rp}, \qquad \forall j \in J^*; \forall r \in R; \forall p \in P \tag{2.D.25}$$

$$x_{ij} \geq 0, \qquad \forall i \in I; \forall j \in J^*. \tag{2.D.26}$$

2.D.4. Taxonomy

This section provides a taxonomy of the models most relevant to our research, with respect to four features: (i) user choice environment (yes or no), (ii) stochastic (or not), (iii) inclusion of congestion (or not) at facilities, (iv) inclusion (or not) of competition. The relevant information is displayed in Table 2.D.1.

Authors	user choice	stochastic	congestion	competition
[**Abouee-Mehrizi et al., 2011**]	×	×	×	
[**Averbakh et al., 2007**]	×			
[**Berman and Drezner, 2006**]	×		×	
[**Camacho-Vallejo et al., 2014**]	×			
[**Castillo et al., 2009**]		×	×	
[**Desrochers et al., 1995**]			×	
[**Kim, 2013**]			×	
[**Küçükaydin et al., 2011**]	×	×		×
[**Labbé and Hakimi, 1991**]				×
[**Marianov and Serra, 2001**]			×	
[**Marianov, 2003**]		×	×	
[**Marianov et al., 2008**]	×	×	×	×
[**Marić et al., 2012**]	×			
[**Rahmati et al., 2014**]		×	×	
[**Vidyarthi and Jayaswal, 2014**]		×	×	
[**Zhang et al., 2010a**]	×		×	

Table 2.D.1. Taxonomy of congested facility location models

Chapter 3

An exact algorithm for a class of mixed-integer programs with equilibrium constraints

The second article is dedicated to an exact algorithm for solving a subclass of mathematical programs with equilibrium constraints (MPEC) involving both integer and continuous variables. We demonstrate that our algorithm can be successfully applied to a location problem, which embeds a variant of a queueing model where the number of servers at facilities are integer decision variables. Although the model considered in the second article is somewhat similar to the one in the first article, it calls for an entirely different approach due to its highly combinatorial aspects.

Let us consider the following global MPEC model

$$\begin{aligned} \min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} \quad & f(x,y) \\ \text{s.t.} \quad & g_k(x,y) \leq 0 \qquad k = 1,\ldots,r \\ & x_i \text{ integer} \qquad i = 1,\ldots,n\ , \end{aligned} \tag{3.0.1}$$

where the vector $y \in Y(x)$ satisfies the lower level variational inequality

$$\langle F(x,y), y - y' \rangle \leq 0 \quad \forall y' \in Y(x)\ , \tag{3.0.2}$$

with $Y(x) = \{y \geq 0 : h_j(x,y) = 0, j = 1,\ldots,p\}$.

We assume all functions continuously differentiable, f and $g_k : \mathbb{R}^n \to \mathbb{R}$ convex, $h_j : \mathbb{R}^{n+m} \to \mathbb{R}$ affine, and the mapping $F : \mathbb{R}^{n+m} \to \mathbb{R}^m$ monotone in its second argument y. If F is a gradient mapping, the global model can be written as a mathematical program with

complementarity constraints (MPCC)

$$
\begin{aligned}
\text{P:} \quad & \min_{\sigma,x,y} \quad f(x,y) \\
& \text{s.t.} \quad g_k(x,y) \geq 0 && k = 1, ..., r \\
& \qquad\quad x_i \text{ integer} && i = 1, \ldots, n \\
& \qquad\quad h_j(x,y) = 0 && j = 1, ..., p \qquad (3.0.3) \\
& \qquad\quad 0 \leq y \perp F(x,y) + \sum_{j=1}^{p} \sigma_j \nabla_y h_j(x,y) \geq 0 \,, && (3.0.4)
\end{aligned}
$$

where the vector of multipliers $\sigma \in \mathbb{R}^p$ is associated with the set of linear constraints.

One of the main challenges associated with MPCCs arises from the violation of the linear independence constraint qualification (LICQ) and of the Mangasarian-Fromovitz constraint qualification (MFCQ), at all feasible points. Failure of these regularity conditions results in unboundedness and nonuniqueness of the multipliers, which yields a poor performance of most nonlinear programming (NLP) algorithms [**Andreani and Martínez, 2001, Baumrucker et al., 2008**].

Designing exact algorithms for this class of problems is extremely difficult due to their nonconvexity. Our novel algorithmic framework is based on a sophisticated combination of several algorithmic ingredients like linearization, relaxation, reformulation, Mixed-Integer Linear Programming (MILP), and the iterative solution of convex (lower-level) subproblems. The main idea is to perform a standard Branch and Bound (B&B) on a MILP relaxation of the original problem, while virtually treating each node of the B&B tree as a separate optimization problem. This flexible tree management allows for full flexibility and one can potentially adapt the formulation, or the solution technique at some strategic nodes.

We illustrate our algorithm on an extension of the classical discrete FLP, that is different from the one in the first article, in the following aspects. The facilities are modelled as infinite capacity $M/M/s$ queues equipped with a fixed service rate μ. The decision variable is the integer vector s, i.e., the number of servers at each open facility. This adds to the combinatorial complexity of the problem, but it makes the upper-level entirely integer. In our algorithm we exploit this property, along with the other specific attributes of the model.

Author contributions

— The general research ideas were developed jointly with my supervisors, Patrice Marcotte and Andrea Lodi.

— The research (including proofs, code, experiments, etc.) was carried out by me.

— The article was written by me, and it was revised and corrected by Patrice Marcotte and Andrea Lodi.

Submitted to SIAM Journal on Optimization (SIOPT) August 2018.

An exact algorithm for a class of mixed-integer programs with equilibrium constraints

Teodora Dan, Andrea Lodi, Patrice Marcotte

ABSTRACT

In this study, we consider a rich class of mathematical programs with equilibrium constraints (MPECs) involving both integer and continuous variables. Such a class, which subsumes mathematical programs with complementarity constraints, as well as bilevel programs involving lower level convex programs is, in general, extremely hard to solve due to complementarity constraints and integrality requirements. For its solution, we design an (exact) branch-and-bound (B&B) algorithm that treats each node of the B&B tree as a separate optimization problem and potentially changes its formulation and solution approach by designing, for example, a separate B&B tree. The algorithm is implemented and computationally evaluated on a specific instance of MPEC, namely a competitive facility location problem that takes into account the queueing process that determines the equilibrium assignment of users to open facilities, and for which, to date, no exact method has been proposed.

Keywords: bilevel location, mixed integer programming, global optimization

3.1. Introduction

Mathematical programs with equilibrium constraints (MPECs) are NP-hard optimization problems that arise in engineering design, transportation, economics and multilevel games,

to name a few areas of application. They embed constraints that are typically expressed as variational inequalities, which makes them highly nonconvex, even in the simplest cases. The aim of this work if twofold. First, we design an exact algorithm for an important class of mixed-integer MPECs, i.e., MPECs that involve both continuous and discrete variables. Next, we computationally evaluate the algorithm on a complex location-queueing model.

Formally, the model that we consider takes the mathematical form

$$\begin{aligned} \min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} \quad & f(x,y) \\ \text{s.t.} \quad & g_k(x,y) \le 0 \qquad k = 1,\dots,r \\ & x_i \text{ integer} \qquad i = 1,\dots,n\ , \end{aligned} \tag{3.1.1}$$

where the vector $y \in Y(x)$ satisfies the lower level variational inequality

$$\langle F(x,y), y - y'\rangle \le 0 \quad \forall y' \in Y(x)\ , \tag{3.1.2}$$

with $Y(x) = \{y \ge 0 : h_j(x,y) = 0, j = 1,\dots,p\}$.

Throughout the paper, we make the assumptions that all functions involved are continuously differentiable. Furthermore, f and $g_k : \mathbb{R}^n \to \mathbb{R}$ are convex, $h_j : \mathbb{R}^{n+m} \to \mathbb{R}$ are affine, and the mapping $F : \mathbb{R}^{n+m} \to \mathbb{R}^m$ is monotone in its second argument y. Whenever the Jacobian of F with respect to y is symmetric, F is a gradient mapping, i.e., $F = \nabla\phi$ for some convex function ϕ. In that case, a vector y satisfies the variational inequality (3.1.2) if and only if it is a solution of the lower level convex program

$$\begin{aligned} \text{LL:} \qquad \min_{y} \quad & \phi(y) \\ \text{s.t.} \quad & h_j(x,y) = 0 \quad j = 1,\dots,p \\ & y \ge 0\ . \end{aligned} \tag{3.1.3}$$

Under our monotonicity and differentiability assumptions, the variational inequality (3.1.2) can be replaced by a Karush-Kuhn-Tucker (KKT) system. This allows to reformulate the global model as the mathematical program with complementarity constraints (MPCC)

$$\begin{aligned} \text{P:} \qquad \min_{\sigma,x,y} \quad & f(x,y) \\ \text{s.t.} \quad & g_k(x,y) \ge 0 && k = 1,...,r \\ & x_i \text{ integer} && i = 1,\dots,n \end{aligned}$$

$$h_j(x,y) = 0 \qquad\qquad j = 1,...,p \tag{3.1.4}$$

$$0 \leq y \perp F(x,y) + \sum_{j=1}^{p} \sigma_j \nabla_y h_j(x,y) \geq 0 \ , \tag{3.1.5}$$

where the vector of multipliers $\sigma \in \mathbb{R}^p$ is associated with the set of linear constraints.

The main difficulty associated with MPCCs is that the complementarity constraints involve both upper level (x), and lower level (y, σ) variables. Even in its simplest form $0 \leq a \perp b \geq 0$ over scalars a and b, the feasible set is the union of two convex polyhedra, namely $\{(a,0) : a \geq 0\} \cup \{(b,0) : b \geq 0\}$. The linear independence constraint qualification (LICQ), requiring the gradients of the active constraints to be independent, is violated at all feasible points, which results in nonuniqueness of the constraint multipliers. The weaker Mangasarian-Fromovitz constraint qualification (MFCQ) is not satisfied either. We recall that MFCQ requires linear independence of the gradients of the equality constraints, and the existence of a direction d pointing into the interior of the region defined by the gradients of the active inequality constraints, such that $\langle \nabla h_j, d \rangle = 0$. Failure of LICQ and MFCQ results in unboundedness and nonuniqueness of the multipliers, therefore, many nonlinear programming (NLP) algorithms (and codes) could perform poorly because their performance strongly rests on these regularity conditions [**Andreani and Martínez, 2001, Baumrucker et al., 2008**].

Designing exact algorithms for this class of problems is a challenging task. Our novel algorithmic approach is based on a sophisticated combination of Mixed-Integer Linear Programming (MILP), linearization techniques and the iterative solution of convex subproblems. Roughly speaking, we start by embedding the solution space of P into an MILP relaxation – which is already nontrivial due to its nonconvexity – and, while performing a standard Branch and Bound (B&B), we iteratively solve either problem LL, to recover the true value of the objective function of a leaf (corresponding to a mixed-integer node), or some subproblems with strengthened relaxation (associated with a fractional node) to perform additional pruning in the tree and speed up convergence. In other words, we virtually treat each node of the B&B tree as a separate optimization problem to which we can adapt the formulation and, consequently, the solution technique, by building, for example, a separate B&B tree.

The algorithm is implemented and computationally tested on a specific instance of P, namely the *competitive congested user-choice discrete facility location problem* CC–FLP.

CC–FLP is an extension of the classical (discrete) facility location problem, a fundamental structure in discrete optimization that is well suited to a variety of real-life applications. While it has been extensively considered in the literature, few studies have incorporated the specific features of a user-choice environment, which frequently involves congestion, either along the paths leading to a facility, or at the facility itself. The congested user-choice model belongs to the MPEC class and can be reformulated as an NP-hard bilevel program, thus falling into the category of mathematical programs with complementarity constraints.

Paper Contribution. The paper provides two strong contributions.

— On the one side, it proposes a novel exact algorithm for solving a fairly general class of mathematical programs with equilibrium constraints. To some extent, such an algorithm virtually treats every node as a separate, i.e., different, optimization problem, adapting either the relaxation or the solution method (or both) depending on some triggering conditions. This is done to achieve full flexibility and to exploit in the most effective way both the strength of the MILP solvers, e.g., their strong preprocessing at the root node, and the pieces of information acquired while exploring the enumeration tree, which could lead to alternative algorithmic decisions and/or problem formulations. Although this is not the only way to exploit this idea (see next section), we believe that it is a fundamental step in the direction of designing complex adaptive algorithms that have the capability, within the same B&B tree, to change solution strategies and formulations, whenever required.

— On the other side, and to the best of our knowledge, our algorithm is the first exact method for CC–FLP, which is a practically relevant and remarkably difficult generalization of the classical facility location problem. So far, in the literature, CC–FLP or variants thereof were only addressed by means of heuristic methods.

Paper Organization. The paper is structured as follows. In Section 3.2, we detail the novel solution method for P and discuss its connection with existing literature, together with its underlying assumptions and limitations. In Section 3.3, we describe CC–FLP, including a literature review, and we provide all necessary details for adapting the proposed algorithm to this application. In Section 3.5, we report on the extensive computational experiments for exactly solving CC–FLP. Finally, in Section 3.6, we draw some conclusions and open avenues for further research on this topic.

3.2. Algorithmic framework

Based on B&B, our solution approach consists of two main phases.

Phase I. In the first phase, we perform a linearization of nonlinear terms and constraints, in order to reduce the original program to an MILP. In our generic model P, there are two distinct sources of nonlinearity, namely F and the complementarity constraints. For linear F, the only linearization needed would involve the complementarity constraints, for which we could write an exact linear reformulation via binary variables [1]. For the sake of generality, we henceforth consider F to be nonlinear. Then, there are several options for linearization.

- Single level linearization:
  - Perform a piecewise linear approximation of F.
  - Introduce big-M constraints to linearize the complementarities.
- Bilevel linearization:
  - Replace the equilibrium constraints by a lower-level program, linearize it, then write the KKT optimality conditions.
  - Linearize the resulting program by replacing the bilinear terms with McCormick envelopes.

The main difficulty in both cases is to perform linear approximations that guarantee that the objective function of the obtained MILP is a bounding function for the objective of the original program. Let $\bar{F}$ be a piecewise linear approximation of F and let $(\tilde{\sigma}, \tilde{x}, \tilde{y})$ be a feasible solution of P. Similarly, given $\tilde{x}$, let $(\bar{\sigma}, \bar{y})$ satisfy (3.1.4)–(3.1.5) when F is replaced by $\bar{F}$. Due to the approximation of F, $\bar{y}$ may not correspond to the true solution of the lower level, therefore $\bar{y}$ may differ from $\tilde{y}$. It follows that $f(\bar{x}, \bar{y})$ may also differ from $f(\tilde{x}, \bar{y})$. This implies that, after linearization and relaxation of integrality constraints at integer nodes, the approximated objective value may not correspond to the true objective value. Our approach is to perform a linear approximation such that the objective function of the obtained MILP be a valid upper bound on the objective of the original program. This will ensure that

$$\textit{optimum of relaxed MILP} \leq \textit{optimum of MILP} \leq \textit{optimum of original problem.} \quad (3.2.1)$$

1. The reader is referred to [**Hijazi and Liberti, 2014, Jeroslow and Lowe, 1984**] for a discussion on the theoretical representability of unbounded disjunctions.

Inequality (3.2.1) must hold in order for B&B not to prune nodes that might contain the optimal solution. In other words, the constructed MILP must be a valid relaxation of P.
Phase II. In the second phase, we implement a B&B algorithm to find the optimal solution of the original problem. At the leaves of a standard B&B tree, the bound equals the value of the corresponding solution. In our case, this is not necessarily true, due to the approximations performed in Phase I. We tackle this issue by interacting with the B&B throughout its execution, i.e., by computing the correct value of the objective function at any associated leaf. To achieve this, we fix the upper level (integer) variables to their corresponding values at the current node, we compute the optimal solution of the (convex) lower-level problem, and we recover the associated objective value of the original problem. In order to reduce the size of the B&B tree and to make the algorithm more efficient, in some of the fractional nodes we compute an on-the-fly lower-bound for the subtree rooted at the corresponding node, by taking into account the bounds of some of the variables. This algorithmic framework is summarized in Algorithm 1 (see pseudocode below).

Algorithm 1

```
1: perform linear approximations of nonlinear terms and constraints so as to satisfy (3.2.1)
2: solve the resulting MILP using B&B:
3: for all nodes in the tree do
4:     if integer node then
5:         compute the true corresponding objective function value
6:     else
7:         if some condition holds then
8:             compute a tighter lower bound (by improving the relaxation)
9: return best found solution
```

Algorithm 1 performs Phase I at Step 1. This can be achieved either under the single level or bilevel linearization. The quality of those linearizations will be discussed later on in the paper and especially evaluated in the computational investigation for the special case of the CC–FLP. The rest of Algorithm 1 concerns Phase II and implements the idea, outlined in the introduction, that every node of the B&B tree can potentially be treated differently. What is described in the algorithm above is the distinction between fractional and integer nodes,

where we either (potentially) improve the relaxation or compute the real objective function value by solving an auxiliary continuous optimization problem, respectively. However, a third special case for the CC–FLP involves B&B nodes in which a certain set of variables, namely the facility location ones, is settled, i.e., all decisions on where locating the facilities are taken but some other integer decisions are not. Each of those nodes is reformulated and solved by the MILP solver IBM CPLEX as a separate MILP within the same B&B algorithm by taking advantage of `callback` functions, which allows the solver to exploit the full power of its preprocessing phase by heavily simplifying the formulation. Of course, the overall branching scheme is tailored so as the location variables are settled first, thus encountering those nodes as early as possible in the search tree.

We close this discussion by outlining the relationship between our solution approach and the relevant literature with respect to the management of the B&B tree.

— First, our algorithm clearly borrows from the vast literature on Global Optimization (GO) the idea (and the need) of iteratively improving the relaxation within the enumeration scheme. In GO, this is achieved by Spatial Branching (i.e., by iteratively improving the convex relaxation) and by applying expensive bound tightening in virtually any node. Of course, the relaxation can be (and is actually) improved in MILP as well by cutting plane generation. However, our approach is a hybrid because, within an MILP scheme, what we do in the majority of cases in (selected) fractional nodes is to tighten the formulation not by exploiting integrality (like for cut generation in MILP) but by improving the linearization of the nonlinear component(s) of the original problem. This is in the spirit, for example, of the work of [**Belotti et al., 2016**], where big-M constraints are iteratively strengthened within the MILP B&B as a GO solver would do. With respect to [**Belotti et al., 2016**], Step 8 of Algorithm 1 goes slightly further by providing a different formulation of the node (by improving the piecewise linear approximation) instead of simply tightening some of its constraints individually. This is related to the work of [**Furini and Traversi, 2013**] where, at some nodes of the B&B tree of a mixed-integer nonlinear programming (MINLP) approach to binary quadratic programming, an alternative semidefinite programming (SDP) relaxation is defined and solved with the aim of improving the bound, thus possibly fathoming the node. The difference is that in Algorithm 1 the improved reformulation

of a node is kept in the entire subtree rooted at the node itself, while the SDP node in [**Furini and Traversi, 2013**] is discarded if its improved bound does not allow the subtree to be fathomed. Ideally, of course, one can think of solving the associated mixed-integer SDP instead, which is not done in [**Furini and Traversi, 2013**].

— Second, the idea of reformulating some of the nodes of a B&B scheme so as to exploit some special structure has been used with different flavors, for example, in [**Raghunathan, 2013**]. More precisely, [**Raghunathan, 2013**] solves by B&B a convex MINLP relaxation of a water network design problem. However, at the nodes in which all binary variables are decided (namely, the diameters of the pipes in the designed network), instead of solving the associated continuous convex relaxation, it is observed that the original nonconvex counterpart has a unique solution and such a problem is efficiently solved directly. In other words, the convex MINLP relaxation is used to embed the solution approach within a rather efficient (although nonlinear) B&B framework, while the special structure of the continuous optimization problems obtained after fixing the diameters is exploited to more accurately solve those nodes. This is precisely what we achieve at Step 5 of Algorithm 1.

Overall, Algorithm 1 yields an effective framework that uses in a flexible way several algorithmic ingredients like preprocessing, reformulation, linearization, etc. through an extremely sophisticated tree management. We believe that managing the enumeration tree flexibly is key to solving extremely difficult nonconvex MINLPs like the one discussed in the next section.

3.3. CC–FLP

To illustrate our algorithmic framework, we consider the problem faced by a company making location and service level decisions in a market where competitors already operate. The aim of the company is to maximize the number of attracted customers, subject to a budget constraint. In the model, queueing at facilities, together with user's selfish behavior, are explicitly taken into account. Under these assumptions clients select and patronize the facility minimizing their disutility, expressed as the weighted sum of travel and waiting time. Assuming constant travel times, the underlying network reduces to a bipartite graph. Let $V = I \times J$ be a complete bipartite graph, where I denotes the set of demand nodes, and J

the set of locations. Every node $i \in I$ represents a population zone of a city, and is endowed with an inelastic demand d_i. Let J_1 denote the set of candidate locations of the emerging company, while J_c is the set of competitors' locations. A client originating from node $i \in I$ and patronizing facility $j \in J$ incurs a travel time t_{ij}. Arriving at facility j, she enters an $M/M/c_j/\infty$ queue, equipped with c_j servers having identical mean service rate μ, and mean (queueing plus service) delay w_j . The disutility u_{ij} is defined as a linear combination of travel time queueing, and service time at the facility, namely

$$u_{ij} = t_{ij} + \alpha w_j, \tag{3.3.1}$$

where α is a positive weight coefficient.

3.3.1. Literature review

Although location problems have been widely studied, few models incorporate user behavior, congestion and competition. In our model, customers select the facilities to patronize, based on the travel time and waiting time at facilities, the latter being a congestion trait. User behavior can be modeled as a Stackelberg game, in which the company locating facilities is the leader, and users represent the follower, fitting the bilevel paradigm. When describing the patronizing behavior, we identify two possibilities: deterministic and probabilistic.

In the deterministic case, users select the facility that minimizes their disutility. In simpler models, all users originating from a demand point patronize the same facility, and disutility does not account for congestion. Under these assumptions [**Marić et al., 2012**] implement three metaheuristics to solve a bilevel formulation of the uncapacitated FLP with user preferences. Similarly, in [**Camacho-Vallejo et al., 2014**], the bilevel FLP with user preferences is solved by using a Stackelberg Evolutionary Algorithm. In [**Vidyarthi and Jayaswal, 2014**], a model where congestion is minimized in the objective function, and users patronize the closest facility is considered. The authors solve their model by constraint generation.

When demand can be split between several facilities, a tie resolution rule must be implemented as well. Within this framework we note the work of [**Desrochers et al., 1995**] who consider a centralized model of a deterministic facility location problem, where individual delays increase with distance. The authors provide a user-choice version of their model that fits the bilevel programming paradigm, although they do not propose a solution algorithm.

In [**Berman and Drezner, 2006**], the location of congested facilities when demand is elastic with respect to distance is investigated. The objective is to maximize total demand, subject to constraints on the waiting time at facilities. Heuristic procedures are proposed.

The presence of non-linearities in the user utility makes the problem even more difficult to solve, thus, only few papers tackle this aspect. We cite here [**Sun et al., 2008**], who consider a generic bilevel facility location model, where the upper level is making locational decisions that minimize the sum of total cost and a congestion function, and the lower level (users) minimizes a non-linear function. The authors propose a heuristic algorithm as solution method. Another work worth mentioning is that of [**Zhang et al., 2010a**], who propose a methodology for addressing a congested facility network design, with the aim of maximizing the participation rate, in a preventive healthcare setting. Demand is elastic with respect to total expected time (travel + waiting time at facilities) experienced by clients. Users patronize the facility minimizing the sum of waiting and travel time. The proposed solution method is a Tabu Search procedure.

In the probabilistic case, users behave according to a discrete choice model based on the random utility paradigm. In the case of Gumbel distributed random terms, this yields Logit closed form expressions for the origin-destination flows. Similar to the deterministic case, most papers consider the utility to be solely based on proximity.

A more elaborate model is proposed by [**Abouee-Mehrizi et al., 2011**], who consider simultaneous decision-making over the location, service rate and price, for facilities located at vertices of a network. Demand is elastic with respect to price, and congestion arising at facilities is characterized by queueing equations. Clients spread among facilities based on proximity only, according to a Multinomial Logit random utility model that takes balking into account. As solution method, the authors propose a hybrid between Tabu Search and a tailored heuristic algorithm.

When it comes to utility, most papers make simplifying assumptions. For one, they assume that users patronize facilities based solely on proximity. To the best of our knowledge, the first paper to address congestion in a competitive user-choice environment is that of [**Marianov et al., 2008**]. The authors consider a scenario in which a company locates p facilities on the vertices of a network where competitors are already operating. The demand is inelastic and users patronize the facility minimizing their disutility, given by the

sum of travel and waiting time. The model is solved by using a two-phase metaheuristic procedure combining GRASP and Tabu Search. In the initial phase, facility locations are selected and a nonlinear assignment problem is solved using the Newton-Raphson algorithm. In [**Dan and Marcotte, 2017**], it is considered a bilevel network design problem, where a firm makes decisions on both location and service levels, and users patronize the facility minimizing the travel and waiting time at facilities, yielding a non-linear bilevel program. The authors propose an approximation algorithm that is asymptotically optimal, as well as a heuristic that exploits the structure of the problem.

3.3.2. Modeling CC–FLP

Throughout this paper we will be using the following notation.

Sets

I: set of demand nodes;

J: set of candidate facility locations (leader and competitor);

J_c: set of competition's facilities;

J_1: set of leader's candidate sites;

$J_1^* \subseteq J_1$: set of leader's open facilities;

$J^* \subseteq J$: set of open facilities (leader and competitor).

Parameters

d_i: demand originating from node $i \in I$;

μ: service rate at open facility $j \in J$;

t_{ij}: travel time between nodes $i \in I$ and $j \in J$;

α: coefficient of the waiting time in the disutility formula;

B: available budget (for opening new facilities and associated service rates);

f_c: fixed cost associated with opening a new facility;

v_c : cost per server.

Basic decision variables

y_j: binary variable set to 1 if a facility is open at site j, and to 0 otherwise;

c_j: number of servers at open facility $j \in J$.

Additional variables

x_{ij}: arrival rate at facility $j \in J$ originating from demand node $i \in I$;

λ_j: arrival rate at node $j \in J$;

ρ_j: traffic intensity at facility $j \in J$;

w_j: mean queueing time at facility j;

γ_i: disutility of users originating from node i.

Let x_{ij} be the number of clients from demand node i patronizing facility j. We define the arrival rate $\lambda_j = \sum_{i \in I} x_{ij}$ at facility j, as well as the number c_j of servers operating at j. Then, the mean waiting time (queueing plus service) at facilities w is a bivariate function depending on both the arrival rate and number of servers **[Kleinrock, 1975]**

$$w(\lambda_j, c_j) = \frac{1}{(\mu c_j - \lambda_j)\left(1 + \dfrac{(1-\rho_j)\, c_j!}{(c_j\rho_j)^{c_j}} \displaystyle\sum_{k=0}^{c_j-1} \frac{(c_j\rho_j)^k}{k!}\right)} + \frac{1}{\mu}, \tag{3.3.2}$$

where $\rho_j = \lambda_j/(\mu c_j) < 1$ is the traffic intensity.

In a user-choice environment, users patronize facilities minimizing their disutility. Given the facility locations and the assigned number of servers, the lower level problem is a user equilibrium problem (Wardrop). The equilibrium is defined by the complementarity system

$$0 \le x_{ij} \perp t_{ij} + \alpha w_j - \gamma_i \ge 0, \qquad i \in I;\ j \in J. \tag{3.3.3}$$

The complete model is as follows.

CC–FLP:

LEADER (COMPANY)

$$\max_{y,c,x,\gamma} \quad z = \sum_{i \in I} \sum_{j \in J_1} x_{ij} \tag{3.3.4}$$

$$\text{s.t.} \quad \sum_{j \in J_1} (f_c \cdot y_j + v_c \cdot c_j) \le B \tag{3.3.5}$$

$$c_j \le M \cdot y_j \qquad j \in J_1 \tag{3.3.6}$$

$$c_j \ge y_j \qquad j \in J_1 \tag{3.3.7}$$

$$y_j \in \{0, 1\} \qquad j \in J_1 \tag{3.3.8}$$

$$c_j \ge 0,\ c_j \text{ integer} \qquad j \in J_1 \tag{3.3.9}$$

FOLLOWER (USERS)

$$0 \leq x_{ij} \perp t_{ij} + \alpha w_j - \gamma_i \geq 0 \qquad i \in I;\ j \in J \qquad (3.3.10)$$

$$w_j = \frac{1}{(\mu c_j - \lambda_j)\left(1 + \dfrac{(1-\rho_j)\, c_j!}{(c_j\rho_j)^{c_j}} \displaystyle\sum_{k=0}^{c_j-1} \frac{(c_j\rho_j)^k}{k!}\right)} + \frac{1}{\mu} \qquad j \in J \qquad (3.3.11)$$

$$\lambda_j = \sum_{i \in I} x_{ij} \qquad j \in J \qquad (3.3.12)$$

$$\rho_j = \frac{\lambda_j}{\mu c_j} \qquad j \in J \qquad (3.3.13)$$

$$\sum_{j \in J} x_{ij} = d_i \qquad i \in I \qquad (3.3.14)$$

$$\lambda_j \leq \mu_j c_j \qquad j \in J \qquad (3.3.15)$$

$$x_{ij} \geq 0 \qquad i \in I; j \in J, \qquad (3.3.16)$$

where M is a suitably large constant, that we set to $(B - f_c)/v_c$, as a direct consequence of (3.3.6).

The decision variables are the vectors y (locations) and c (number of servers), while the user assignment x is the solution of an equilibrium problem that can be equivalently obtained by solving a convex optimization problem. The objective in Eq. (3.3.4) is to maximize the total number of users that patronize the leader's facilities, while constraint (3.3.5) ensures that the total cost does not exceed the budget B. Constraints (3.3.6) set the upper bound for the number of servers per facility. In order to avoid irrelevant solutions, constraints (3.3.7) specify that at least one server must be assigned to any open facility. Logical constraints (3.3.10)–(3.3.13) enforce the user equilibrium conditions. Finally, constraints (3.3.14) ensure that demand is satisfied, and constraints (3.3.15) guarantee that the arrival rate does not exceed the total service rate.

It is clear that CC–FLP fits the generic model P. We now discuss some simple but important properties of our model.

Proposition 3.3.1. *The waiting time function w is strictly increasing in λ and strictly decreasing in c.*

PROOF. We note that $A = \dfrac{c!}{(c\rho)^c}\sum_{k=0}^{c-1}\dfrac{(c\rho)^k}{k!}$. Then, $w(\lambda,c) = \dfrac{C}{\mu c-\lambda}$, where $C = \dfrac{1}{1+(1-\rho)\,A}$. We have that

$$\frac{\partial C}{\partial \rho} = -C^2\left[-A+(1-\rho)\frac{\partial A}{\partial \rho}\right] = C^2\left[\frac{c!}{c^c}\sum_{k=0}^{c-1}\frac{c^k\rho^{k-c-1}}{k!}\left((c-k)\,(1-\rho)+\rho\right)\right] > 0.$$

C is thus, strictly increasing in ρ. But $\dfrac{1}{\mu c-\lambda}$ is strictly increasing in λ and strictly decreasing in c, so the conclusion follows. □

Proposition 3.3.2. [Lee and Cohen, 1983] *The waiting time function w is convex in λ.*

Now, for given couples (λ, c), we consider under and over estimators of w, respectively $\tilde{w}$ and $\hat{w}$, such that

$$\tilde{w}(\lambda,c) \le w(\lambda,c) \text{ , and } \hat{w}(\lambda,c) \ge w(\lambda,c) \text{ .} \tag{3.3.17}$$

We will now show that, if we replace w_j with $\tilde{w}_j$ at the leader's facilities, and with $\hat{w}_j$ at competitors facilities, the optimal solution of the resulting program yields a valid upper bound on the optimum of the initial problem.

The objective function defined by Eq. (3.3.4) can be expressed as $z = \sum_{j\in J_1}\lambda_j$, where λ_j is the total number of users patronizing facility j. Let CC–FLP' be a modified version of CC–FLP, where w_j in constraints (3.3.10)–(3.3.11) has been replaced by $\tilde{w}_j$ at leader's facilities, and with $\hat{w}_j$ at competitors' facilities. Let x', λ' and γ' be an optimal solution of CC–FLP', with objective $z' = \sum_{j\in J_1}\lambda'_j$.

Proposition 3.3.3. *For any feasible pair (y,c), the optimal objective function of CC–FLP' is greater or equal than the optimal objective of CC–FLP.*

PROOF (BY CONTRADICTION). Assume that the proposition is false. Then, $\sum_{j\in J_1}\lambda_j > \sum_{j\in J_1}\lambda'_j$ implies the existence of a facility $j \in J_1$ such that $\lambda'_j < \lambda_j$, and there exists $k \in J_c$ such that $\lambda'_k > \lambda_k$. From Proposition 3.3.1, it follows that

$$w(\lambda_k,c) < w(\lambda'_k,c) \le \hat{w}(\lambda'_k,c) \tag{3.3.18}$$

$$w(\lambda_j,c) > w(\lambda'_j,c) \ge \tilde{w}(\lambda'_j,c) \text{ .} \tag{3.3.19}$$

In other words, a fraction of the population patronizing the leader's facilities in CC–FLP switches to competitors' facilities in CC–FLP'. Therefore, there exists a demand node i, together with a population ϵ that originates from i, patronizes both facility $j \in J_1$ in CC–FLP and facility $k \in J_c$ in CC–FLP'. Then,

$$0 < x'_{ij} + \epsilon = x_{ij} \text{ , and } \quad x_{ik} + \epsilon = x'_{ik} \text{ .}$$

From Eq.(3.3.10), we have that $t_{ij}+\alpha w(\lambda_j, c) = \gamma_i$ (since $x_{ij} > 0$). If follows from Eq.(3.3.19) that $t_{ij}+\alpha w(\lambda_j, c) > t_{ij}+\alpha \tilde{w}(\lambda'_j, c) \geq \gamma'_i$, and we have $\gamma_i > \gamma'_i$. Similarly, from Eq.(3.3.10), we have that $t_{ik}+\alpha w(\lambda_k, c) \geq \gamma_i$, and from Eq.(3.3.18) we have $t_{ik}+\alpha w(\lambda_k, c) < t_{ik}+\alpha \hat{w}(\lambda'_k, c) = \gamma'_i$, which yields $\gamma_i < \gamma'_i$, a contradiction. □

3.3.3. Linearization

The first phase of our algorithmic framework is the linearization of the non-linear terms and constraints, in order to reformulate CC–FLP as an MILP. The key task is the linearization of the highly nonlinear two-variable waiting time function. As previously mentioned for CC–FLP, we will consider two approximation schemes, respectively single-level and bilevel.

3.3.3.1. *Single-level linearization*

The underlying idea of this method is to directly linearize the complementarity condition (3.1.5), i.e., Wardrop conditions (3.3.3) and, independently, the waiting time functions, with the aim to obtain an MILP. This is outlined below.

Linearization of equilibrium constraints

For a given set of open facilities y and their assigned number of servers c, the lower level flows x_{ij} are solution of the non-linear complementarity problem

$$0 \leq x_{ij} \perp t_{ij} + \alpha w_j - \gamma_i \geq 0 \qquad i \in I;\ j \in J \tag{3.3.20}$$

which is linearized through the introduction of binary variables and big-M constants:

$$t_{ij} + \alpha w_j - \gamma_i \leq M_\gamma s_{ij} \qquad i \in I;\ j \in J \tag{3.3.21}$$

$$x_{ij} \leq d_i(1 - s_{ij}) \qquad i \in I;\ j \in J \tag{3.3.22}$$

$$s_{ij} \in \{0, 1\} \qquad i \in I;\ j \in J \text{ .} \tag{3.3.23}$$

Tight values for the constant M_γ can be derived from information concerning the maximal queueing time $w_{\max}$ and maximal travel time $t_{\max} = \max_{i\in I, j\in J}\{t_{ij}\}$. Since the total demand must be strictly less than the total service rate, there must exist a minimum number of servers C so that $\sum_{i\in I} d_i < C\mu$. Let $\ell = \lfloor B/f_c \rfloor$ be the maximum number of facilities allowed by the budget, and let $\varepsilon = (C - \sum_{i\in I} d_i)/(\ell + |J_c|)$. Then, an upper bound $\overline{C}$ on the number of servers at leader's or competitor's facilities is

$$\overline{C} = \max\left\{\max_{j\in J_c}\{c_j\}, (B - f_c)/v_c\right\}.$$

At optimality, there exists at least one facility j_1, belonging either to the leader or to the competition, such that $c_{j_1}\mu \geq \lambda_{j_1} + \varepsilon$. A priori, facility j_1 is unknown, but its number of servers can vary from 1 to $\overline{C}$. There follows the upper bound

$$w_{\max} \leq \max_{c\in\{1,\ldots,\bar{C}\}, c\mu>\varepsilon} \{w(c\mu - \varepsilon, c)\}. \tag{3.3.24}$$

Let $\gamma_{\max}$ and $\gamma_{\min}$ be the maximum and minimum values that γ can assume at optimality, respectively. Since $\gamma_{\min} \geq 0$ and $\gamma_{\max} \leq t_{\max} + \alpha w_{\max}$, one can set $M_\gamma = t_{\max} + \alpha w_{\max}$. Note that for a given number N_o of leader's open facilities the value of M_γ can be further reduced, yielding

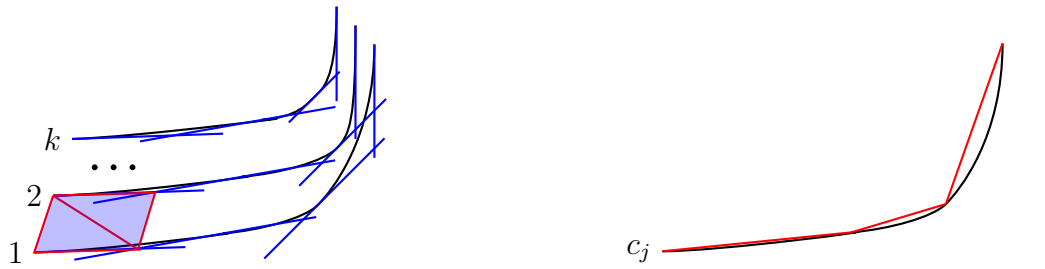
$$\overline{C} = \max\left\{\max_{j\in J_c}\{c_j\}, (B - N_o f_c)/v_c\right\}$$

and $\varepsilon = (C\mu - \sum_{i\in I} d_i)/(N_o - |J_c|)$. In some cases, this yields a smaller value of M_γ.

Linearization of waiting time

In order to derive an upper bound on the leader's profit, we perform an under approximation of queueing delays at its facilities, and an over approximation of the delays at the competitor's facilities, respectively, as illustrated in Figure 3.1.

The waiting time at the leader's facilities is a nonlinear bivariate function of variables λ_j and c_j. Given $w_{\max}$, and for each number of servers k, we select a number $\lambda^k_{\max} < \mu k$ such that $w(\lambda^k_{\max}, k) \geq w_{\max}$. Considering the maximum number of servers $c_{\max}$ allowed by the budget B, one samples each interval $[0, \lambda^k_{\max}]$ using N_l points $\lambda^{kn}, k = 0, \ldots, c_{\max}, n = 1, \ldots, N_l$ such that $\lambda^{ki} < \lambda^{kj}$ for all $1 \leq k \leq c_{\max}$ and $1 \leq i < j \leq N_l$. Next, we compute the tangent lines at λ^{kn}, as well as the intersections between each two consecutive lines, yielding $N_l + 1$ points, including the endpoints 0 and $\lambda^k_{\max}$. We denote these points as $(\tilde{\lambda}^{kn}, \underline{w}^{kn})$, where $\tilde{\lambda}^{kn}$ corresponds to the sample on λ, and $\underline{w}^{kn}$ is an under approximation of w at $(\tilde{\lambda}^{kn}, k)$. Since



(a) Leader: under approximation $\tilde{w}$ **(b)** Follower: over approximation $\hat{w}$

Figure 3.1. Piecewise linear approximation of queueing delay w.

w_j is convex in λ_j, the piecewise linear approximation provides a valid lower bound. Next, we base our linear approximation of the w constraint on the triangle technique described in [**D'Ambrosio et al., 2010**]. This yields

$$\sum_{k=0}^{c_{\max}} \sum_{n=1}^{N_l+1} \left(\underline{l}_{jkn} + \underline{l}_{jkn}\right) = 1, \qquad j \in J_1 \tag{3.3.25}$$

$$\tilde{s}_{jkn} \leq \bar{l}_{jkn} + \underline{l}_{jkn} + \bar{l}_{jkn-1} + \underline{l}_{jk-1n-1} + \bar{l}_{jk-1n-1} + \underline{l}_{jk-1n}$$
$$j \in J_1; k = 0, \ldots, c_{\max}; n = 1, \ldots, N_l + 1 \tag{3.3.26}$$

$$\sum_{k=0}^{c_{\max}} \sum_{n=1}^{N_l+1} \tilde{s}_{jkn} = 1, \qquad j \in J_1 \tag{3.3.27}$$

$$\lambda_j = \sum_{k=0}^{c_{\max}} \sum_{n=1}^{N_l+1} \tilde{s}_{jkn} \tilde{\lambda}^{kn}, \qquad j \in J_1 \tag{3.3.28}$$

$$c_j = \sum_{k=0}^{c_{\max}} \sum_{n=1}^{N_l+1} \tilde{s}_{jkn} k, \qquad j \in J_1 \tag{3.3.29}$$

$$\tilde{w}_j \geq \sum_{k=0}^{c_{\max}} \sum_{n=1}^{N_l+1} \tilde{s}_{jkn} \underline{w}^{kn}, \qquad j \in J_1 \tag{3.3.30}$$

$$\bar{l}_{jkn}, \underline{l}_{jkn} \in \{0, 1\} \qquad j \in J_1; k = 0, \ldots, c_{\max}; n = 1, \ldots, N_l + 1 \tag{3.3.31}$$

$$0 \leq \tilde{s}_{jkn} \leq 1 \qquad j \in J_1; k = 0, \ldots, c_{\max}; n = 1, \ldots, N_l + 1 \tag{3.3.32}$$

$$\bar{l}_{j,-1,n} = 0, \quad \underline{l}_{j,-1,n} = 0 \qquad j \in J_1; n = 1, \ldots, N_l + 1 \tag{3.3.33}$$

$$\bar{l}_{j,c_{\max},n} = 0, \quad \underline{l}_{j,c_{\max},n} = 0 \qquad j \in J_1; n = 1, \ldots, N_l + 1 \tag{3.3.34}$$

$$\bar{l}_{j,k,0} = 0, \quad \underline{l}_{j,k,0} = 0 \qquad j \in J_1; k = 0, \ldots, c_{\max} \tag{3.3.35}$$

$$\bar{l}_{j,k,N_l+1} = 0, \quad \underline{l}_{j,k,N_l+1} = 0 \qquad j \in J_1; k = 0, \ldots, c_{\max}. \tag{3.3.36}$$

We perform a similar linearization for the waiting time at competing facilities, where the number of servers c_j is constant. Next, we construct a piecewise linear function $\hat{w}_j$ so that $\hat{w}_j \leq w_j, \ \forall j \in J_c$. Given $w_{\max}$, we compute $\hat{\lambda}^j_{\max} < \mu c_j$ such that $w_j(\hat{\lambda}^j_{\max}, c_j) \geq w_{\max}$ and sample the interval $[0, \hat{\lambda}^j_{\max}]$ using N_c points $\hat{\lambda}^{jn}, n = 1, \ldots, N_c$ such that $\hat{\lambda}^{jn} < \hat{\lambda}^{jm}$ for all $1 \leq n < m \leq N_c$. This yields the linearization

$$\sum_{n=1}^{N_c} \hat{s}_{jn} = 1 \qquad j \in J_c \tag{3.3.37}$$

$$\lambda_j = \sum_{n=1}^{N_c} \hat{s}_{jn} \hat{\lambda}^{jn} \qquad j \in J_c \tag{3.3.38}$$

$$\hat{w}_j \leq \sum_{n=1}^{N_c} \hat{s}_{jn} w(\hat{\lambda}^{jn}, c_j) \qquad j \in J_c \tag{3.3.39}$$

$$\sum_{n=1}^{N_c} \hat{l}_{jn} = 1 \qquad j \in J_c \tag{3.3.40}$$

$$\hat{s}_{jn} \leq \hat{l}_{jn} + \hat{l}_{jn-1} \qquad j \in J_c; \ n = 1, \ldots, N_c \tag{3.3.41}$$

$$\hat{l}_{jn} \in \{0, 1\} \qquad j \in J_c; \ n = 1, \ldots, N_c \tag{3.3.42}$$

$$0 \leq \hat{s}_{jn} \leq 1 \qquad j \in J_c; \ n = 1, \ldots, N_c \tag{3.3.43}$$

$$\hat{l}_{j,0} = \hat{l}_{j,N_c} = 0 \qquad j \in J_c. \tag{3.3.44}$$

The inequality form of constraints (3.3.30) and (3.3.39) ensure that the original w is feasible for the approximated problem. The MILP relaxation of CC–FLP then becomes

$$\text{CC–FLP1:} \quad \max_{y,c,x,\gamma} \quad z = \sum_{i \in I} \sum_{j \in J_1} x_{ij}$$

$$\text{s.t.} \quad \text{constraints (3.3.5)–(3.3.9), (3.3.12)–(3.3.16), (3.3.20)–(3.3.23), (3.3.25)–(3.3.44).}$$

3.3.3.2. *Bilevel linearization*

An alternative to the linearization of the user equilibrium conditions (3.3.10)–(3.3.14) is to replace the latter by the convex optimization problem [**Beckmann et al., 1956**]

$$\text{PL:} \qquad \min_{x,\lambda} \qquad \sum_{i \in I} \sum_{j \in J^*} x_{ij} t_{ij} + \alpha \sum_{j \in J^*} \int_0^{\lambda_j} w_j(\tau, c_j) d\tau \tag{3.3.45}$$

$$
\begin{aligned}
\text{s.t.} \qquad & \sum_{j \in J^*} x_{ij} = d_i && i \in I \\
& x_{ij} \geq 0 && i \in I; j \in J^* \\
& \lambda_j = \sum_{i \in I} x_{ij} && j \in J^*,
\end{aligned}
$$

whose KKT conditions match the Wardrop conditions (3.3.20). Based on this formulation, an MILP approximation can be obtained by (i) performing a piecewise linearization of the sole nonlinear term $\int_0^{\lambda_j} w_j(q, c_j)$, (ii) writing an equivalent linear program, (iii) linearizing the complementarity term of the optimality conditions.

We now provide the technical details. Let us construct piecewise linear functions $\tilde{W}(\lambda, c)$ (leader) and $\hat{W}(\lambda, c)$ (competition), whose respective derivatives $\tilde{w}(\lambda, c)$ and $\hat{w}(\lambda, c)$ satisfy (3.3.17). This is achieved as follows. Given $w_{\max}$ as defined by Eq. (3.3.24), and for each number of servers c, let $\tilde{\lambda}^c_{\max} < \mu c$. We sample N_l points $\tilde{\lambda}^{cn}$ ($c = 1, \ldots, c_{\max}, n = 1, \ldots, N_l$), sorted in increasing order, in each interval $[0, \lambda^c_{\max}]$. Then, for each value of c, the integral of the waiting time at leader's facilities is approximated by the piecewise linear function

$$
\begin{aligned}
\tilde{W}_j(\lambda, c) &= (\lambda - \tilde{\lambda}^{cl}) w(\tilde{\lambda}^{cl}, c) + \sum_{k=2}^{l} (\tilde{\lambda}^{ck} - \tilde{\lambda}^{c,k-1}) w(\tilde{\lambda}^{c,k-1}, c) \qquad \text{if } \lambda \in [\tilde{\lambda}^{cl}, \tilde{\lambda}^{c,l+1}] \\
&= \lambda \underbrace{w(\tilde{\lambda}^{cl}, c)}_{\tilde{f}^{cl}_j} + \underbrace{\sum_{k=2}^{l} (\tilde{\lambda}^{ck} - \tilde{\lambda}^{c,k-1}) w(\tilde{\lambda}^{c,k-1}, c) - \tilde{\lambda}^{cl} w(\tilde{\lambda}^{cl}, c)}_{\tilde{g}^{cl}_j} \; . \qquad (3.3.46)
\end{aligned}
$$

Similarly, we linearize the integral of the waiting time at competitors' facilities. Given $w_{\max}$, we set $\hat{\lambda}^j_{\max} < \mu c_j$ such that $w_j(\hat{\lambda}^j_{\max}, c_j) \geq w_{\max}$. We sample the interval $[0, \hat{\lambda}^j_{\max}]$ using N_c points $\hat{\lambda}^{jn}, n = 1, \ldots, N_c$, sorted in increasing order and consider the piecewise linear approximation

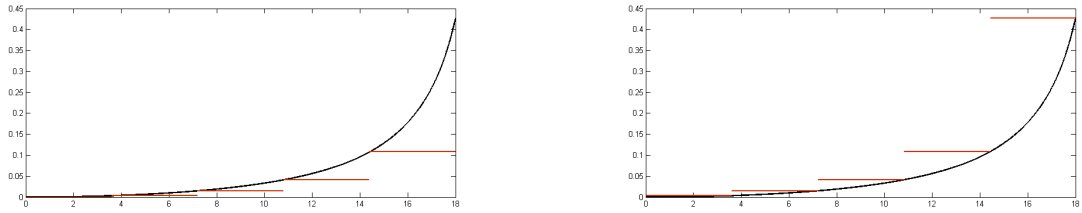
$$
\begin{aligned}
\hat{W}(\lambda, c_j) &= (\lambda - \hat{\lambda}^{jn}) w(\hat{\lambda}^{j,n+1}, c_j) + \sum_{k=2}^{n} (\hat{\lambda}^{jk} - \hat{\lambda}^{j,k-1}) w(\hat{\lambda}^{jk}, c_j) \qquad \text{if } \lambda \in [\hat{\lambda}^{jn}, \hat{\lambda}^{j,n+1}] \\
&= \lambda \underbrace{w(\hat{\lambda}^{j,n+1}, c_j)}_{\hat{f}^{jn}} + \underbrace{\sum_{k=2}^{n} (\hat{\lambda}^{jk} - \hat{\lambda}^{j,k-1}) w(\hat{\lambda}^{jk}, c_j) - \hat{\lambda}^{jn} w(\hat{\lambda}^{j,n+1}, c_j)}_{\hat{g}^{jn}} \; . \qquad (3.3.47)
\end{aligned}
$$

Let $\tilde{w}$ and $\hat{w}$ denote the derivatives of $\tilde{W}$ and $\hat{W}$, respectively. It is easy to check that

$$
\tilde{w}(\lambda, c) = w(\tilde{\lambda}^{c,l-1}, c) \qquad \text{if } \lambda \in [\tilde{\lambda}^{c,l-1}, \tilde{\lambda}^{c,l}) \text{ and } l \in \{2, 3, \ldots, N_l\}
$$

$$\hat{w}(\lambda, c_j) = w(\hat{\lambda}^{jn}, c_j) \qquad \text{if } \lambda \in [\hat{\lambda}^{j,n-1}\hat{\lambda}^{jn}) \text{ and } n \in \{1, 2, \ldots, N_c - 1\} \ .$$

Then, $\tilde{w}$ and $\hat{w}$ are piecewise constant approximations of w in λ, satisfying (3.3.17), as illustrated in Figure 3.2.



(a) $\tilde{w}$ **(b)** $\hat{w}$

Figure 3.2. Piecewise constant approximations of w.

Since w is increasing in λ according to Proposition 3.3.1, for each couple(λ_j, c_j) we have that $\tilde{w}_j(\lambda_j, c_j) \leq w_j(\lambda_j, c_j)$ and $\hat{w}_j(\lambda_j, c_j) \geq w_j(\lambda_j, c_j)$.

We now replace $\int_0^{\lambda_j} w(\tau, c_j)d\tau$ in (3.3.45) with $\tilde{W}_j$ for the leader and $\hat{W}_j$ for the competitor, as defined by (3.3.46) and (3.3.47). This yields the approximated lower level

$$\begin{aligned} \text{PL2:} \quad \min_{x} \quad & \sum_{i\in I}\sum_{j\in J^*} x_{ij}t_{ij} + \alpha\left[\sum_{j\in J_1^*}\tilde{W}(\lambda, c_j) + \sum_{j\in J_c}\hat{W}(\lambda, c_j)\right] \\ \text{s.t.} \quad & \text{constraints (3.3.12), (3.3.14), (3.3.16), (3.3.46)–(3.3.47).} \end{aligned}$$

Next, we substitute variables $\tilde{v}$ and $\hat{v}$ with $\tilde{W}$ and $\hat{W}$, respectively. This allows to rewrite PL2 as a linear program, whenever the upper level vector c is fixed:

$$\begin{aligned} \text{PLL:} \quad \min_{x} \quad & \sum_{i\in I}\sum_{j\in J^*} x_{ij}t_{ij} + \alpha\sum_{j\in J_1^*}\tilde{v}_j + \alpha\sum_{j\in J_c}\hat{v}_j \\ \text{s.t.} \quad & \tilde{v}_j - \tilde{f}_j^{c_j,l}\cdot\lambda_j - \tilde{g}_j^{c_j,l} \geq 0 && j \in J_1^*; l = 1, \ldots, N_l && (3.3.48) \\ & \hat{v}_j - \hat{f}^{jn}\cdot\lambda_j - \hat{g}^{jn} \geq 0 && j \in J_c; n = 1, \ldots, N_c && (3.3.49) \\ & \sum_{j\in J^*} x_{ij} = d_i && i \in I && (3.3.50) \\ & \sum_{i\in I} x_{ij} - \lambda_j = 0 && j \in J^* && (3.3.51) \\ & x_{ij} \geq 0 && \forall i \in I; j \in J^*. && (3.3.52) \end{aligned}$$

Upon introduction of the dual variables $\tilde{\pi}_j^l \geq 0$, $\hat{\pi}_j^n \geq 0$, η_i, δ_j and $\phi_{ij} \geq 0$ associated with constraints (3.3.48)–(3.3.52), we derive the primal-dual optimality conditions

$$
\begin{aligned}
t_{ij} + \delta_j + \eta_i - \phi_{ij} &= 0 && i \in I; j \in J^* \\
\sum_{l=1}^{N_l - 1} \tilde{f}_j^{cl} \tilde{\pi}_j^l - \delta_j &= 0 && j \in J_1^* \\
\sum_{n=1}^{N_c - 1} \hat{f}^{jn} \hat{\pi}_j^n - \delta_j &= 0 && j \in J_c \\
\sum_{l=1}^{N_l - 1} \tilde{\pi}_j^l &= \alpha && j \in J_1^* \\
\sum_{n=1}^{N_c - 1} \hat{\pi}_j^n &= \alpha && j \in J_c \\
\left(\tilde{v}_j - \tilde{f}_j^{c_j,l} \cdot \lambda_j - \tilde{g}_j^{c_j,l} \right) \tilde{\pi}_j^l &= 0 && j \in J_1^*; l = 1, \ldots, N_l - 1 \qquad (3.3.53) \\
\left(\hat{v}_j - \hat{f}^{jn} \cdot \lambda_j - \hat{g}^{jn} \right) \hat{\pi}_j^n &= 0 && j \in J_c; n = 1, \ldots, N_c - 1 \qquad (3.3.54) \\
x_{ij} \phi_{ij} &= 0 && i \in I; j \in J^* \qquad (3.3.55)
\end{aligned}
$$

constraints (3.3.48) – (3.3.52) .

We now replace constraints (3.3.10)–(3.3.16) in the original formulation CC–FLP with the above optimality conditions. The optimum of the latter program yields an upper bound on the objective function of CC–FLP .

Linear relaxation of bilinear terms

To complete the linearization process, we approximate the bilinear terms $\tilde{f}_j^{c_j,l} \tilde{\pi}_j^l$ and $\tilde{f}_j^{c_j,l} \lambda_j$ by their McCormick envelopes, and linearize the complementarity constraints in Eqs. (3.3.53), (3.3.54) and (3.3.55) using big-M constants. This yields the final MILP formulation

CC–FLP2: (3.3.56)

$$
\begin{aligned}
\max_{y,c} \quad & \sum_{i \in I} \sum_{j \in J_1} x_{ij} \\
\text{s.t.} \quad & \sum_{j \in J_1} (f_c \cdot y_j + v_c \cdot c_j) \leq B
\end{aligned}
$$

$$\lambda_j \leq \mu \cdot c_j \qquad j \in J_1$$

$$\lambda_j \leq \lambda_{\max} \qquad j \in J_c$$

$$c_j \leq M \cdot y_j \qquad j \in J_1$$

$$c_j \geq y_j \qquad j \in J_1$$

$$c_j = \sum_{k=0}^{c_{\max}} k \cdot l_{j,k}^w, \qquad j \in J_1$$

$$\sum_{k=0}^{c_{\max}} l_{j,k}^w = 1 \qquad j \in J_1$$

$$y_j \in \{0,1\},\ l_{j,k}^w \in \{0,1\} \qquad j \in J_1; k = 0, \ldots c_{\max}$$

$$c_j \geq 0,\ c_j \leq c_{\max},\ c_j \text{ integer} \qquad j \in J_1$$

$$\sum_{j \in J} x_{ij} = d_i \qquad i \in I$$

$$\sum_{i \in I} x_{ij} = \lambda_j \qquad j \in J$$

$$\tilde{v}_j - z_{j,l}^{w\lambda} - \tilde{G}_j^l \geq 0 \qquad j \in J_1; l = 1, \ldots, N_l$$

$$\hat{v}_j - \hat{f}^{jn} \cdot \lambda_j - \hat{g}^{jn} \geq 0 \qquad j \in J_c; n = 1, \ldots, N_c$$

$$0 \leq \tilde{c}_{jk} \leq 1 \qquad j \in J_1; k = 0, \ldots, c_{\max} \qquad (3.3.57)$$

$$\sum_{k=0}^{c_{\max}} \tilde{c}_{jk} = 1 \qquad j \in J_1 \qquad (3.3.58)$$

$$c_j = \sum_{k=0}^{c_{\max}} k \cdot \tilde{c}_{jk} \qquad j \in J_1 \qquad (3.3.59)$$

$$\tilde{f}_j^l = \sum_{k=0}^{c_{\max}} \tilde{c}_{jk} \cdot w(\tilde{\lambda}^{kl}, k) \qquad j \in J; l = 1, \ldots, N_l$$

$$\tilde{G}_j^l = \sum_{k=0}^{c_{\max}} \tilde{c}_{jk} \cdot \tilde{g}^{kl} \qquad j \in J; l = 1, \ldots, N_l$$

$$z_{j,l}^{w\lambda} \geq w_{\max} \cdot \lambda_j + \tilde{\lambda}_{\text{UB}}^l \cdot \tilde{f}_j^l - w_{\max} \cdot \tilde{\lambda}_{\text{UB}}^l \qquad j \in J_1; l = 1, \ldots, N_l$$

$$z_{j,l}^{w\lambda} \leq \tilde{\lambda}_{\text{UB}}^l \cdot \tilde{f}_j^l \qquad j \in J_1; l = 1, \ldots, N_l$$

$$z_{j,l}^{w\lambda} \leq w_{\max} \cdot \lambda_j \qquad j \in J_1; l = 1, \ldots, N_l$$

$$z_{j,l}^{w\lambda} \geq 0 \qquad j \in J_1; l = 1, \ldots, N_l$$

$$
\begin{aligned}
& x_{i,j} \geq 0 && i \in I; j \in J \\
& t_{ij} + \delta_j + \gamma_i \geq 0 && i \in I; j \in J \\
& \sum_{l=1}^{N_l - 1} \tilde{\pi}_j^l = \alpha && j \in J_1 \qquad (3.3.60) \\
& \sum_{n=1}^{N_c - 1} \hat{\pi}_j^n = \alpha && j \in J_c \qquad (3.3.61) \\
& \sum_{l=1}^{N_l - 1} z_{j,l}^{\pi,f} - \delta_j = 0 && j \in J_1 \\
& \sum_{n=1}^{N_c - 1} \hat{\pi}_j^n \hat{f}^{jn} - \delta_j = 0 && j \in J_c \\
& z_{j,l}^{\pi,f} \geq \alpha \cdot \tilde{f}_j^l + w_{\max} \pi_{j,l} - w_{\max} \cdot \alpha && j \in J_1; l = 1, \ldots, N_l \\
& z_{j,l}^{\pi,f} \leq w_{\max} \pi_{j,l} && j \in J_1; l = 1, \ldots, N_l \\
& z_{j,l}^{\pi,f} \leq \alpha \cdot \tilde{f}_j^l && j \in J_1; l = 1, \ldots, N_l \\
& z_{j,l}^{\pi,f} \geq 0 && j \in J_1; l = 1, \ldots, N_l \\
& \tilde{\pi}_j^l \leq \alpha \cdot \tilde{s}_{j,l}^{\pi} && j \in J_1; l = 1, \ldots, N_l \\
& \tilde{v}_j - z_{j,l}^{w\lambda} - \tilde{g}_j^l \leq \left(1 - \tilde{s}_{j,l}^{\pi}\right) \cdot M_\pi && j \in J_1; l = 1, \ldots, N_l \\
& \hat{\pi}_j^n \leq \alpha \cdot \hat{s}_{j,n}^{\pi} && j \in J_c; n = 1, \ldots, N_c \\
& \hat{v}_j - \hat{f}^{jn} \lambda_j - \hat{g}^{jn} \leq \left(1 - \hat{s}_{j,k}^{\pi}\right) \cdot M_\pi && j \in J_c; n = 1, \ldots, N_c \\
& t_{ij} + \delta_j + \gamma_i \leq s_{i,j}^{\phi} \cdot M_\phi && i \in I; j \in J \\
& x_{ij} \leq \left(1 - s_{i,j}^{\phi}\right) D && i \in I; j \in J \\
& s_{i,j}^{\phi} \in \{0,1\}, \quad s_{i,j}^{\pi} \in \{0,1\} && i \in I; j \in J
\end{aligned}
$$

where $\tilde{g}^{cl}$, $\hat{f}^{jn}$, and $\hat{g}^{jn}$ are computed according to (3.3.46) and (3.3.47), $\lambda_{UB}^l = \max_{c=0,\ldots,c_{\max}} (\tilde{\lambda}^{cl})$, and $D = \sum_{i \in I} d_i$. Valid expressions for the big-M constants M_ϕ and M_π are obtained as follows. Eqs. (3.3.60) and (3.3.61) imply that $\pi_j^l \leq \alpha$ and $\pi_j^n \leq \alpha$. We set

$$
\delta_{\max}^L = \sum_{l=1}^{N_l} \max_{c=1,\ldots,c_{\max}} \left(\alpha w(\tilde{\lambda}^{cl}, c)\right) \ , \ \ \delta_{\max}^C = \sum_{n=1}^{N_c} \max_{j \in J_c} \left(\alpha w(\hat{\lambda}^{j,l}, c_j)\right),
$$

and $\delta_{\max} = \max(\delta^L_{\max}, \delta^C_{\max})$. It follows that $\gamma_i \in [-t_{\max} - \delta_{\max}, 0]$, and $M_\phi = t_{\max} + \delta_{\max}$. Similarly, we set $M_\pi = \max(v^L_{\max}, v^C_{\max})$, where

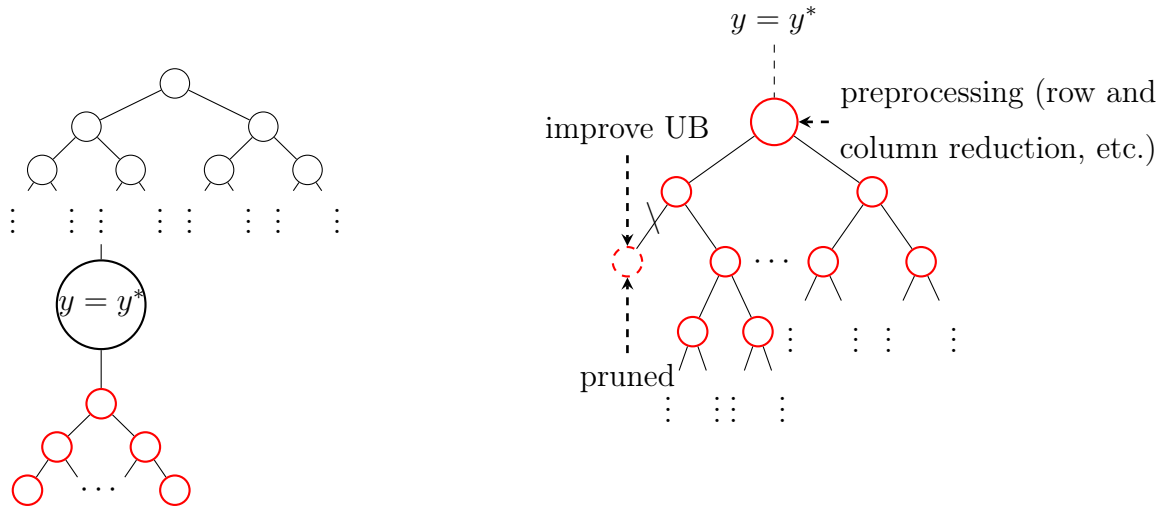
$$v^L_{\max} = \max_{c=1,\ldots,c_{\max}} \left\{ \max_{l=1,\ldots,N_l} \left(D \cdot w(\tilde{\lambda}^{cl}, c) + \tilde{g}^{cl} \right) \right\}$$
$$v^C_{\max} = \max_{j \in J_c} \left\{ \max_{n=1,\ldots,N_c} \left(D \cdot w(\hat{\lambda}^{jl}, c_j) + \hat{g}^{jn} \right) \right\} .$$

3.4. Branch-and-Bound Algorithm

This section is devoted to an exact algorithm for CC–FLP that exploits the upper bound on the objective provided by the approximate programs CC–FLP1 and CC–FLP2 while, for a given leader solution (y, c), a lower bound is obtained solving the corresponding lower level program for an equilibrium assignment of users to facilities. Our main issue is that, in sharp contrast with 'standard' B&B, there is a gap between the objective of the true formulation CC–FLP and that of the approximation CC–FLP1. Our aim is to overcome this difficulty through the efficient interaction with the Branch-and-Bound software, through callbacks. While our implementation is based on the IBM CPLEX suite, any software that allows for callbacks could have been used.

Our algorithm is based on a nested B&B tree structure. Nodes of the *main tree* relate to the location variables y_j and are labeled by the y vector. Whenever, at a given node of the main tree, the y-solution of the relaxed problem is integer-valued (such a node is called a 'leaf'), we grow an *inner subtree* (Figure 3.1) that focuses on the c_j variables (number of servers) and other intermediate variables by defining and solving a new and separate MILP. Due to the gap between the true objective and that of the relaxed problem, 'manual' interaction with the software is required in order not to wrongly fathom nodes of the main tree. In particular, the true value of a leaf's objective must be retrieved before it can be used for fathoming or pruning purposes, both in the main tree and the inner subtrees.

We now detail the implementation and functionality of the nested structure. As discussed in Section 3.1, the reason for treating each of those nodes as a separate MILP is to leverage at the same time the CPLEX preprocessing capability and the pieces of information collected until that point in the tree to tighten the formulation of the subproblem originated at this node.



(a) Main B&B tree

(b) Inner B&B subtree

Figure 3.1. The nested B&B trees.

3.4.1. Main B&B tree

Let CC–FLP1(y^*) and CC–FLP2(y^*) denote the restriction of CC–FLP1 and CC–FLP2 obtained by fixing the location vector y at y^*. The main B&B solves a relaxation of the MILP approximation (CC–FLP1 or CC–FLP2), where the integrality requirement is relaxed on all variables, with the exception of the location vector y. The initial lower and upper bounds for the MILP are computed by the solver. At every integer node, we use the current solution y^*, call the subroutine solving for the optimal number of servers, and append the no-good global cut ([**D'Ambrosio et al., 2010**])

$$\sum_{j \in J_1, y_j^* = 0} y_j + \sum_{j \in J_1, y_j^* = 1} (1 - y_j) \geq 1 \tag{3.4.1}$$

to the model. The latter ensures that the current solution does not appear elsewhere in the tree. The rejection of all feasible solutions is required to guarantee the validity of the branching and pruning rules. Indeed, if a solution were not rejected, CPLEX would accept it along with its approximate objective, and its update of bounds might be inconsistent with the values of the true model. Whenever an improved solution is obtained while solving a restricted problem, the corresponding cut is global and allows to update the lower bound, while the incumbent is updated externally.

We have implemented the sequence of operations described above in a *Lazy Constraint Callback*, called at every feasible (integer) node. A feature of this operation is that it is called only and in all feasible integer nodes, and that it allows for appending user cuts that might cut off integer parts of the domain. Its pseudocode is given in Algorithm 2.

Algorithm 2 Lazy Constraint Callback in the main B&B

1: $y^* \leftarrow currentSolution$
2: $restrictedBestObj \leftarrow$ Solve CC–FLP1(y^*) or CC–FLP2(y^*)
3: **if** $restrictedBestObj > bestFound$ **then**
4: $\quad objectiveCut \leftarrow (z > restrictedBestObj)$
5: $\quad$ addGlobalCut($objectiveCut$)
6: $\quad bestFound \leftarrow restrictedBestObj$
7: $yCut \leftarrow Eq.$ (3.4.1)
8: addGlobalCut($yCut$)

3.4.2. Inner subtrees

When the approximation CC–FLP1 is used, a subtree is associated with y^* of the main tree. One then solves subproblem CC–FLP1(y^*), to which one appends a set of constraints whose role is to discard feasible nodes, as achieved in the main B&B tree by means of constraints (3.4.1). Precisely, for all $j \in J_1,\ k = 0, \ldots, c_{\max}$, we introduce binary variables $\tilde{c}_{jk}$ and consider the following unary representation of c_j.

$$c_j = \sum_{k=0}^{c_{\max}} k \cdot \tilde{c}_{jk} \qquad j \in J_1 \tag{3.4.2}$$

$$\sum_{k=0}^{c_{\max}} \tilde{c}_{jk} = 1 \qquad j \in J_1 \tag{3.4.3}$$

$$\tilde{c}_{jk} \in \{0, 1\} \qquad j \in J_1; k = 0, \ldots, c_{\max}. \tag{3.4.4}$$

On the other hand, if CC–FLP2 is solved, no additional constraints are required since Eq. (3.3.57)–(3.3.59) are already part of the formulation.

At each integer node, the true objective is computed by solving the lower level assignment problem, for instance using the Frank-Wolfe algorithm. If an improved solution is uncovered,

the incumbent is saved externally, and the lower bound on the objective is updated through the addition of a global cut. Next, we reject the current integer solution by adding the global cut

$$\sum_{k=0}^{c_{\max}} \sum_{\substack{j \in J_1, \\ \tilde{c}^*_{jk}=0}} \tilde{c}_{jk} + \sum_{k=0}^{c_{\max}} \sum_{\substack{j \in J_1, \\ \tilde{c}^*_{jk}=1}} (1 - \tilde{c}_{jk}) \geq 1, \tag{3.4.5}$$

thus guaranteeing the consistency of the branching and pruning rules. Eqs. (3.4.2)–(3.4.4), together with Eq. (3.4.5), ensure that the current vector c will not reappear later in the tree. These operations have been implemented in the *Lazy Constraint Callback* described by Algorithm 3.

Algorithm 3 Lazy Constraint Callback in the inner B&B

1: $\tilde{c}^* \leftarrow$ current $\tilde{c}$; $c^* \leftarrow$ current c
2: $currentObj \leftarrow$ Frank-Wolfe(c^*)
3: **if** $currentObj > bestFound$ **then**
4: $\quad objectiveCut \leftarrow (z > currentObj)$
5: $\quad bestFound \leftarrow currentObj$
6: $\quad$ addGlobalCut($objectiveCut$)
7: $cCut \leftarrow$ *Eq.* (3.4.5)
8: addGlobalCut($cCut$)

In other words, the unary representation of c is necessary to impose the no-good constraints (3.4.5), which would be otherwise hard to write for general integer variables.

3.4.3. Improving the upper bound

At the fractional nodes of an inner B&B subtree, the location variables are fixed, while some c_j variables are fractional. Depending on the node, the lower and upper bounds on the integer variables might have been improved by the branching decisions taken so far. Let c_j^{UB} denote the upper bound on variable c_j, at a given fractional node in the subtree. Then we have the following result.

Proposition 3.4.1. *At a fractional node, let us set $c_j \leftarrow c_j^{UB}$, for all $j \in J_1$. Then, the objective value associated with this solution is a valid upper bound on the true objective, in the subtree rooted at that node.*

PROOF. Since c_j^{UB} exceeds any value that c_j may achieve in the subtree rooted at the current node, it follows from Proposition 3.3.1 that

$$w(\lambda, c_j^{UB}) \leq w(\lambda, c_j) \quad \forall j \in J_1. \tag{3.4.6}$$

Since the competitors' service rate remains unchanged, the conclusion is a direct consequence of Proposition 3.3.3. □

In most cases, fixing the number of server variables may be infeasible, due to the budget constraint. Let $c_{\text{TOTAL}} = B - |J_1^*| \cdot f_c$ be the leader's available budget at the current node, to be distributed among facilities. At a a fractional node, it is very likely that $\sum_{j \in J_1} c_j^{UB} > c_{\text{TOTAL}}$, due to a number of factors. One empirically expects that, deep into the enumeration tree, bounds are tight and yield feasible solutions.

Improved upper bounds have been implemented in a *User Cut Callback* that is invoked at every fractional node. Within the callback, we retrieve the upper bounds on variables c_j and set the current solution to those bounds. If condition (3.4.7) below holds, we compute the objective value associated with this solution, by evaluating the associated equilibrium flows, i.e., by solving PL. If this operation improves the bound provided by CPLEX, we append it to the model in the form of a local user cut as described in Algorithm 4.

When computed close to the root of the tree, bounds tend to be loose, and the probability of improving over the CPLEX bound is small. On the other hand, when computed deep into the tree, the tightness of the bounds improve, but only a small portion of the tree is pruned. In our implementation, the improvement procedure has been activated whenever the inequality

$$\sum_{j \in J_1^*} c_j^{UB} \leq c_{\text{TOTAL}} + q \cdot |J_1^*| \tag{3.4.7}$$

held. In (3.4.7), q plays the role of a flexibility or frequency parameter, to be tuned offline. When $q = \infty$ the upper bound is computed at all fractional nodes, while if $q = 0$, it is only computed at the leaves of the tree.

3.4.4. Computing a lower bound

The performance of the exact method can be improved by computing a good lower bound at the root of the B&B tree. The underlying idea is to linearize the queueing terms at the

Algorithm 4 User Cut Constraint Callback in the restricted B&B

```
c^UB ← vector of upper bounds with respect to c
if Eq. (3.4.7) then
    currentObj ← Frank-Wolfe(c^UB)
    CPLEXUB ← upper bound provided by CPLEX
    if CPLEXUB> currentObj then
        ubCut ← (z ≤ currentObj)
        addLocalCut(ubCut)
```

leader's facilities, without the introduction of binary variables. More specifically, in the sampling scheme described in Section 3.3.3, let us introduce triangles based on consecutive samples, as detailed in [**D'Ambrosio et al., 2010**]. The upper triangles are defined by the points $(\tilde{\lambda}^{k,n}, k)$, $(\tilde{\lambda}^{k+1,n}, k+1)$ and $(\tilde{\lambda}^{k+1,n+1}, k+1)$, while the lower triangles are defined by $(\tilde{\lambda}^{k,n}, k)$, $(\tilde{\lambda}^{k,n+1}, k)$ and $(\tilde{\lambda}^{k+1,n+1}, k+1), n = 1, \ldots, N_l, k = 1, \ldots, c_{\max} - 1$. The coefficient of the plane equations associated with the lower and upper triangles are, for every $1 \leq n \leq N_l$ and $1 \leq k \leq c_{\max} - 1$

$$
\begin{array}{ll}
\underline{u}_1^{kn} = \tilde{\lambda}^{k+1,n} - \tilde{\lambda}^{k,n} & \overline{u}_1^{kn} = \tilde{\lambda}^{k+1,n} - \tilde{\lambda}^{k+1,n+1} \\
\underline{u}_2^{kn} = 1 & \overline{u}_2^{kn} = 0 \\
\underline{u}_3^{kn} = \underline{w}^{k+1,n} - \underline{w}^{k,n} & \overline{u}_3^{kn} = \underline{w}^{k+1,n} - \underline{w}^{k+1,n+1} \\
\underline{v}_1^{kn} = \tilde{\lambda}^{k,n+1} - \tilde{\lambda}^{k,n} & \overline{v}_1^{kn} = \tilde{\lambda}^{k,n+1} - \tilde{\lambda}^{k+1,n+1} \\
\underline{v}_2^{kn} = 0 & \overline{v}_2^{kn} = -1 \\
\underline{v}_3^{kn} = \underline{w}^{k,n+1} - \underline{w}^{k,n} & \overline{v}_3^{kn} = \underline{w}^{k,n+1} - \underline{w}^{k+1,n+1} \\
\underline{a}^{kn} = \underline{u}_2^{kn} \cdot \underline{v}_3^{kn} - \underline{u}_3 kn \cdot \underline{v}_2^{kn} & \overline{a}^{kn} = \overline{u}_2^{kn} \cdot \overline{v}_3^{kn} - \overline{u}_3 kn \cdot \overline{v}_2^{kn} \\
\underline{b}^{kn} = \underline{u}_3^{kn} \cdot \underline{v}_1^{kn} - \underline{u}_1 kn \cdot \underline{v}_3^{kn} & \overline{b}^{kn} = \overline{u}_3^{kn} \cdot \overline{v}_1^{kn} - \overline{u}_1 kn \cdot \overline{v}_3^{kn} \\
\underline{c}^{kn} = \underline{u}_1^{kn} \cdot \underline{v}_2^{kn} - \underline{u}_2 kn \cdot \underline{v}_1^{kn} & \overline{c}^{kn} = \overline{u}_1^{kn} \cdot \overline{v}_2^{kn} - \overline{u}_2 kn \cdot \overline{v}_1^{kn} \\
\underline{d}^{kn} = -\left(\underline{a}^{kn}\tilde{\lambda}^{k,n} + \underline{b}^{kn}k + \underline{c}^{kn}\underline{w}^{kn}\right) & \overline{d}^{kn} = -\left(\overline{a}^{kn}\tilde{\lambda}^{k+1,n+1} + \overline{b}^{kn}(k+1) + \overline{c}^{kn}\underline{w}^{k+1,n+1}\right).
\end{array}
\tag{3.4.8}
$$

The plane equations defined by the lower and upper triangles are

$$
\underline{a}^{kn}\lambda_j + \underline{b}^{kn}c_j + \underline{c}^{kn}w_j + \underline{d}^{kn} = 0 \ , \ \text{ and } \quad \overline{a}^{kn}\lambda_j + \overline{b}^{kn}c_j + \overline{cb}^{kn}w_j + \overline{d}^{kn} = 0 \ .
$$

Next, we convexify w_j's by setting them to the maximum of their respective linear approximations, namely

$$w_j \approx \max_{\substack{k=1,\dots,c_{\max}-1 \\ n=1,\dots N-l}} \left\{ \max \left(-\frac{\underline{a}^{kn}}{\underline{c}^{kn}}\lambda_j - \frac{\underline{b}^{kn}}{\underline{c}^{kn}}c_j - \frac{\underline{d}^{kn}}{\underline{c}^{kn}},\ -\frac{\overline{a}^{kn}}{\overline{c}^{kn}}\lambda_j - \frac{\overline{b}^{kn}}{\overline{c}^{kn}}c_j - \frac{\overline{d}^{kn}}{\overline{c}^{kn}} \right) \right\} . \tag{3.4.9}$$

From the leader's point of view, the lower the waiting time, the more customers will be attracted to her facilities. This allows to replace Eq. (3.4.9) by the inequalities

$$\begin{aligned} w_j &\geq -\frac{\underline{a}^{kn}}{\underline{c}^{kn}}\lambda_j - \frac{\underline{b}^{kn}}{\underline{c}^{kn}}c_j - \frac{\underline{d}^{kn}}{\underline{c}^{kn}} \quad k = 1,\dots,c_{\max}-1, n = 1,\dots,N-l \\ w_j &\geq -\frac{\overline{a}^{kn}}{\overline{c}^{kn}}\lambda_j - \frac{\overline{b}^{kn}}{\overline{c}^{kn}}c_j - \frac{\overline{d}^{kn}}{\overline{c}^{kn}} \quad k = 1,\dots,c_{\max}-1, n = 1,\dots,N-l\ . \end{aligned} \tag{3.4.10}$$

We can now write CC–FLP as the following MILP

$$\begin{aligned} \text{CC–FLPH:} \quad \max_{y,c,x,\gamma} \quad & z = \sum_{i\in I}\sum_{j\in J_1} x_{ij} \\ \text{s.t.} \quad & \text{constraints (3.3.5)–(3.3.9), (3.3.12), (3.3.15), (3.3.16), (3.3.20)–(3.3.23),} \\ & \text{(3.3.37)–(3.3.44), (3.4.2)–(3.4.4), (3.4.8), (3.4.10).} \end{aligned}$$

Note that, for the competitor it would be ill-advised to write a linearization similar to (3.4.9) – (3.4.10), since increasing w at competitor's facilities would actually increase the objective value (of the leader). At optimality, the waiting time at competitor facilities would be set to very high values, in an attempt to maximize the objective, and no competitor inequality would be active, yielding a very poor approximation. Additionally, we maintain the presence of binary variables for the competitor, which are limited in number and hence do not increase significantly the difficulty of the model. Of course, the heuristic scheme can be used both for computing a standalone approximate solution for CC–FLP and as warm start for its exact solution. The approximate model is then solved by the nested B&B strategy.

We close this section by mentioning that the improved 'on-the-fly' upper bounds are not implemented in the heuristic, as they significantly slow down the exploration process.

3.5. Experimental setup and results

The MILP formulations were solved by IBM CPLEX Optimizer version 12.6. All tests were performed on a computer equipped with 96 GB of RAM, and two 6-core Intel(R) Xeon(R) X5675 processors running at 3.07GHz. To assess the performance of our algorithm, we could not compare with alternative methods from literature, which are nonexistent, as discussed in Section 3.3.1. The model of [**Marianov et al., 2008**] involves facility location within a user-choice environment, but the decision variables are limited to the locations, while a heuristic is designed for its solution. This lack of alternatives prompted us to compute an optimal solution through exhaustive search, iterating through all possible combinations of number of servers that satisfy the budget constraint. A feasible flow is then computed to yield a feasible solution to the bilevel program. Some solutions are discarded without computing the flow, for the following reason. At any open facility, the arrival rate needs to be lower that the service level. Thus, if the total number of servers times the service rate is lower than the incumbent, the current solution cannot yield a better objective value, and we have no incentive to compute the lower-level optimum.

Our experiments involve networks of varying sizes, and travel times ranging from 0 to 5000. In order to generate challenging instances, the combination of budget, fixed and variable costs was selected such as to ensure the existence of feasible solutions involving a number of open facilities roughly equal to half the total number, and where the number of servers could reach 20 on small tests, and 40 or more on larger tests. Note that, at optimality less than half facilities are typically open, and service levels do not reach their upper limits. In all our tests we used 5 λ-samples for both the leader and the competitor.

An initial set of experiments was intended to probe the efficiency of linearizations CC–FLP1 and CC–FLP2, respectively, for various values of parameter q. We show only the most successful results, namely $q = 15$ for the single level linearization, and $q = 25$ for the bilevel linearization. At the end of this section we provide a deeper analysis of the impact of this parameter on the overall performance of our algorithm.

Tables 3.1 and 3.2 display the results for 15 and 20-node networks, respectively, which are compared for reference with the full enumeration scheme. The exact methods were tested with and without a warm start (columns 'w/o warm start' and 'warm start', respectively). The warm start is provided by the best solution found by heuristic model CC–FLPH within

a time limit. In the case of 15-node networks, the warm start improves significantly the running time. It is not needed, however for the 20-node networks tested, as the methods perform well without it.

	single level linearization ($q = 15$)				bilevel linearization ($q = 25$)				
	w/o warm start		warm start		w/o warm start		warm start		
test #	total	opt.	total	opt.	total	opt.	total	opt.	enumeration
1	3199	577	6002	3615	87791	58457	81183	50708	170825
2	51242	44696	17237	11485	209778	168134	86522	23866	190268
3	110719	100898	38231	31913	619715	494483	1295419	1208952	1873586
4	47678	40360	11536	2423	132524	866	132761	32420	194206
5	26830	21353	14422	5552	670075	482895	227470	64780	624795
6	42051	39583	5500	2975	148006	112378	161920	126093	1419919
7	11985	11778	1268	600	2106842	2106820	73981	71867	11460
8	6840	5674	3738	3300	21935	15422	18854	15872	7098
9	10694	8798	6324	3118	147756	98541	78980	21630	300081
10	12424	10320	8474	6215	217631	202242	49474	28254	767436
Average	29951	28404	11273	7120	436205	374024	220656	164444	555967

Table 3.1. CPU times (seconds) on 15-node networks; 'opt.' refers to the CPU required to find an optimal solution.

Those initial results confirm the stability and performance of the algorithm when adopting the single level linearization scheme. This might be due to the use of McCormick envelopes. Under both schemes, the algorithm outperforms by two orders of magnitude the enumeration-based method. The single level linearization outperforms clearly the bilevel (McCormick) linearization on most 15 and 20-node instances.

We ran the same set of experiments on 25-node networks, and report the results in Table 3.3. All tests were warm started with the heuristic since, without it, the instances were intractable, with CPLEX running out of memory quickly. The enumeration scheme failed on all tests after running more than 15 weeks. The single level linearization performs faster than the bilevel counterpart on the fraction of tests that can be solved. Actually, the algorithm has to stop for lack of memory on more than half of the instances, which shows the limitation of our exact method. In that respect, the bilevel scheme looks slightly more robust, being able to solve 8 instances, versus 6 in the single level case. We note, however,

	single level linearization $q = 15$		bilevel linearization $q = 25$		
test #	total	opt.	total	opt.	enumeration
1	1358	1327	1705	1687	27660
2	36	8	234	209	5892
3	550	90	1504	821	8233
4	667	211	2880	2816	149672
5	307	66	3833	3553	26183
6	338	34	1644	1495	27212
7	1301	416	2265	1673	165337
8	495	406	2438	2401	42301
9	383	265	9823	9766	9571
10	2149	1614	2555	2102	280145
11	1024	958	4789	4718	19249
12	3404	2513	4573	2913	191448
13	305	184	1654	1579	10321
14	332	105	1372	1256	20674
15	383	222	1343	1189	242524
16	1450	313	5401	4090	168118
17	2203	307	4042	3429	309729
18	2149	1132	4018	3609	727332
19	427	137	1633	1002	9731
20	1462	392	2592	1947	191918
Average	1036	535	3015	2608	131663

Table 3.2. CPU times (seconds) on 20-node networks; 'opt.' refers to the CPU required to find an optimal solution. The methods were not warm started.

that the budget allows for up to 12 facilities to be open, and that the number of servers can vary between 1 and 55, which makes for a very challenging class of problems.

The aim of the second set of experiments is to assess the accuracy of the linearization of CC–FLP1, as shown in Table 3.5. In this process, the MILP is solved using B&B, without using callbacks. Once CPLEX has reached and proved optimality, Frank-Wolfe's algorithm is used to retrieve its corresponding true objective value (column 'objective') and compare it with the approximated objective value ('approximated') and with the actual optimum

	single level linearization		bilevel linearization	
	$q = 15$		$q = 25$	
test #	total	opt.	total	opt.
1	37.4	6.0	53.8	35.7
2	–	–	–	–
3	–	–	–	–
4	–	–	–	–
5	–	–	–	–
6	–	–	–	–
7	–	–	613.8	17.9
8	–	–	–	–
9	–	–	–	–
10	51.3	6.5	550.5	12.5
11	–	–	–	–
12	47.2	14.1	–	–
13	–	–	563.8	31.1
14	60.1	6.4	–	–
15	–	–	78.0	29.1
16	–	–	176.0	102.0
17	21.7	0.6	125.7	35.2
18	16.2	3.3	535.7	453.2
19	–	–	–	–
20	–	–	–	–
Average	39.9	6.15	337.2	89.6

Table 3.3. CPU times (**hours**) on 25-node networks; 'opt.' refers to the CPU required to find an optimal solution.

(column 'optimal') computed by our exact algorithm. We observe that, on average, the objective value of the solution found falls within 3.6% of the optimum. Additionally solving only the MILP yields 89.17% of the optimal locations, which suggests that the approximation can be used as a good heuristic on its own, as well. The behaviour of the approximation on the 15-node networks (see Table 3.4) is similar, capturing on average 94.4% of the optimum, but finding only 68.3% of the optimal facilities.

	MILP				
test #	MILP obj.	true obj.	CPU (s)	optimum	deviation (%)
1	213.57	192.29	1707	204.88	6.2
2	200.24	187.19	66915	193.02	3.1
3	151.82	138.89	158976	147.87	6.1
4	223.58	201.84	2668	212.97	5.2
5	191.29	174.53	91936	182.48	4.4
6	184.76	174.48	11288	180.61	3.4
7	230.41	210.43	27107	218.27	3.6
8	240.06	210.79	5481	220.54	4.4
9	191.69	158.74	4855	185.01	14.2
10	194.07	176.40	5911	186.02	5.2
Average	202.15	182.56	37684	193.17	5.5

Table 3.4. Comparison of the best solution found when solving only the approximation, versus the optimal solution, for 15-node networks.

In the third set of experiments, we demonstrate why it is advantageous to use the nested B&B tree structure described in Section 3.4. First we used a single B&B solving for all variables, with only the no-good cuts within the associated callback, and without computing the on-the-fly upper bound. In this case the problem became quickly intractable, even for small instances. For example, on the 9-node networks that we have tested, a single tree takes more than 3 hours and still does not prove optimality. In comparison, the nested tree structure takes less than 7 minutes. One reason for this is the preprocessing at the root nodes of the subtrees. In order to investigate this further, we have measured the objective value of every leaf of the *main tree* ('Original node' in Figure 3.1), which is an upper bound for the subtree rooted at this node, since the main tree is solving a relaxed problem. We then generated the respective *inner subtree* and we retrieved the bounds both after presolve and after solving the root node. If the subtree was found infeasible at the root node, the bound is shown as 0. Our measurements were taken on ten 9-node instances, for a total of 2129 explored subtrees, out of which 328 were cut off or found infeasible at the root node. As shown in Figure 3.1, we observe a significant improvement in the upper bound, after presolve and solving the root node.

	MILP				
test #	MILP obj.	true obj.	CPU (s)	optimum	deviation (%)
1	287.80	270.30	150	273.55	1.2
2	330.46	312.57	38	323.56	3.4
3	287.86	256.04	102	269.00	4.9
4	296.43	264.94	333	291.95	9.2
5	293.42	277.26	89	287.57	3.6
6	274.25	252.97	146	264.37	4.3
7	268.33	257.52	58	259.14	0.6
8	273.35	258.01	48	270.19	4.6
9	293.83	262.12	68	290.98	9.9
10	248.21	237.36	106	248.21	4.3
11	269.47	249.53	40	261.39	4.6
12	252.33	242.09	71	244.63	1.1
13	291.50	280.73	47	284.58	1.3
14	282.79	264.19	77	271.94	2.9
15	262.29	259.89	237	261.64	0.7
16	242.04	225.69	130	242.04	6.7
17	253.63	252.74	103	253.63	0.3
18	221.47	205.36	583	216.55	5.1
19	289.47	273.95	376	283.10	3.2
20	245.59	245.59	125	245.59	0.0
Average	273.23	257.44	146	267.18	3.6

Table 3.5. Comparison of the best solution found when solving only the approximation, versus the optimal solution, for 20-node networks.

The following measures were also computed:

— average improvement in UB after presolve: 7.26%

— average improvement in UB after root node: 10.28%

— average LP columns reduction: 52.56

— average LP rows reduction: 47.88

— average MIP columns reduction: 156.32

— average MIP rows reduction: 16.99.

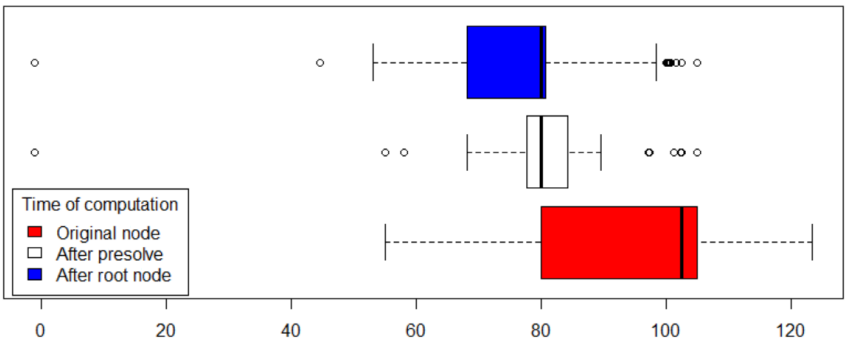


Figure 3.1. Distribution of the upper bounds throughout the execution of the algorithm. 'Original node' is as computed at integer nodes in the main tree. 'After presolve' and 'After Root Node' show the upper bound computed after the presolve and after solving the root node of the inner subtrees, respectively.

Our measurements demonstrate that by creating nested trees, rather than using a single tree, we make full use of the heuristics, cuts, rows and columns reductions, and other computations that CPLEX performs during presolve and at the root nodes. The problem becomes thus more tractable, even without the computation of the on-the-fly upper bounds.

It is important to note that the order in which the nodes are explored differ between the nested tree and single tree approach, which could impact the efficiency of the algorithm. When using nested trees, no new nodes in the main tree will be processed until the current subtree is solved. This particular behaviour cannot be easily achieved in a single tree, even if we prioritize branching on the location variables.

The fourth experiment was designed to assess the impact of the on-the-fly upper bounds on the overall performance of the algorithm. Since, without the strengthening of the UB, even small instances are intractable, we have limited our analysis to 9-node instances, using the single-level linearization. Figure 3.2 illustrates the typical evolution of the upper bounds and the best objective function, throughout the execution of our methods. Notice that the upper bound in the main tree does not improve fast, while the bounds on the subtrees vary significantly from one subtree to the other.

We have measured the average execution time ('CPU'), the number of integer solution explored ('# of sols'), and the number of integer nodes pruned ('# of cuts'), for ten 9-node instances, as shown in Table 3.6. We notice that, as q increases, the average CPU has a

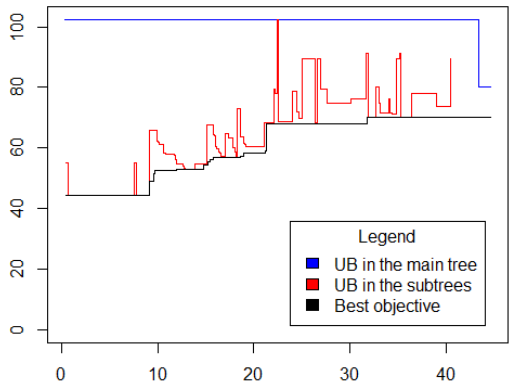


Figure 3.2. Typical evolution of the upper bounds and best objective during the execution of our algorithm.

	w/o q	q=3	q=12	q=20	q=30	q=50
CPU (s)	306.7	87.4	55.4	69.7	72.2	90.3
# of sols	11049	773	512	512	512	512
# of cuts	89741	100017	100278	100278	100278	100278

Table 3.6. Performance of the single-level linearization for different values of q, for 9-node networks.

convex-like behaviour (decreasing, levelling, increasing), which is to be expected. As q increases, the on-the-fly upper bound is computed at more nodes, and higher in the subtrees. Therefore, at first, the number of integer solutions visited decreases, as more cuts are computed and more nodes are pruned. However, for large values of q, the bound is computed often (it is a costly operation), and does not improve on the bound provided by CPLEX. Thus, the CPU increases, while the number of nodes pruned stalls. It is therefore ill-advised to set q to a large value. In our tests, values between 10 and 15 were most successful for the single level linearization, and around 25 for the bilevel linearization.

3.6. Conclusion

The MPEC framework allows the modelling of situations that are highly relevant in practice. However, the resulting mathematical program highly challenging combinatorial and nonlinear features, which explains the frequent recourse to heuristic (meta-heuristics, math-heuristics) for its solution, and the paucity of exact methods. Nevertheless, we expect

that generic algorithms that exploit MILP approximations of single-level reformulations, as well as a clever management of the B&B tree, deserve some consideration. We hope that the present work, which may be viewed as a step in that direction, will trigger further research in global approaches to bilevel programs involving a lower level variational inequality.

Chapter 4

Joint location and pricing within a user-optimized environment

In the third article we analyze a model that captures the key features of facility location and pricing, in a congestion-sensitive user choice environment. More precisely, we consider a problem faced by a service firm that is making revenue-maximizing location and pricing decisions in a competitive market.

In real-life situations, customers are sensitive not only to locations, but to service level as well as to prices. While low prices may attract more customers, they may also induce large waiting times at facilities, which may, ultimately deter customers. Alternatively, a smaller number of clients buying higher priced items might yield a high profit. In such an environment, the firm must take into account the user-optimized behaviour.

Three types of pricing strategies are usually considered literature [**Hanjoul et al., 1990**]:

— mill pricing: prices can vary between facilities;

— uniform pricing: all facilities charge the same price;

— discriminative pricing: customers can be charged different prices at the same facility.

From a modelling standpoint, we extend the models considered in the first two articles, by incorporating mill pricing, which is one of the most challenging form. Akin to the first article, facilities are modelled as infinite M/M/1 queues, and the decision variable is μ the service rate, however, balking is disregarded.

At the upper level, the firm maximizes its revenue, whilst at the lower level, we have users equilibrium problem. Clients minimize their personal utility, which incorporates pricing and

queueing

$$\tilde{u}_{ij} = t_{i,j} + \alpha w_j + \beta p_j$$

where $t_{i,j}$ stands for the travel time, w_j represents the waiting time at facilities, and p_j is the price charged per service. This situation fits the context of a Stackelberg game, and is best described using the bilevel programming framework or, more generally, a mathematical program with equilibrium constraints (MPEC).

Concepts from network pricing and CC–FLP are merged into a single model, which makes the problem much more challenging by the presence of facility location and service level decision variables, bivariate queueing delays, as well as highly nonlinear nonconvex equilibrium constraints. Our algorithm borrows ideas from both the bilevel pricing and location literature. We adapt a reformulation technique introduced in [**Julsain, 1999**] for coping with pricing of the arcs of a packet-switched communication network. The non-linear objective function of the resulting program is linearized via a technique applied in our first paper [**Dan and Marcotte, 2017**], which yields a tractable mixed integer linear program (MILP).

Author contributions

— The general research ideas were developed jointly with my supervisors, Patrice Marcotte and Andrea Lodi.

— The research (including proofs, code, experiments, etc.) was carried out by me.

— The article was written by me, and it was revised and corrected by Patrice Marcotte and Andrea Lodi.

Submitted to EURO Journal on Computational Optimization

Joint location and pricing within a user-optimized environment

Teodora Dan, Andrea Lodi, Patrice Marcotte

ABSTRACT

In a facility service setting, whenever the disutility of customers accessing a facility is impacted by queueing or congestion effects, the resulting equilibrium cannot be ignored by a firm that strives to maximize revenue within a competitive environment. We model this situation as a nonlinear bilevel program that involves both discrete and continuous variables, and for which we propose an efficient algorithm based on an approximation that can be solved for its global optimum.

Keywords: pricing, location pricing, bilevel programming, mixed integer programming, equilibrium, queueing, nonconvex

4.1. Introduction

In a competitive market, service levels and pricing, along with facility locations, are critical decisions that a service provider faces, in order to capture demand and maximize profit. In this context, an important trait of a user-choice market is congestion, which has been often overlooked in the pricing literature, where one routinely assumes that users patronize the closest facility, disregarding the congestion that may arise at facilities in the form of queues. However, in real-life situations, customers are sensitive to service level

as well as to prices. Actually, low prices that attract customers to a facility may in turn induce large waiting times that will deter customers and shift them to the competition. Alternatively, the smaller number of clients buying high-priced items might be offset by the better experience associated with lower waiting times. In such an environment, the firm that makes location and pricing decisions must take into account not only the price and location attributes of its competitors, but also the user-optimized behaviour of its potential customers, who patronize the facility that maximizes their individual utility. This situation fits the framework of a Stackelberg game, and is best formulated as a bilevel program or, more generally, a mathematical program with equilibrium constraints (MPEC). At the upper level, the firm makes revenue-maximizing location and pricing decisions, taking into account the user equilibrium resulting from those decisions.

The resulting MPEC, which involves highly nonlinear queueing terms, as well as continuous (user flows) and discrete (location decision) variables, looks formidable. The aim of this paper is to show that it is yet amenable to a strategy that involves approximation by a tractable mixed integer linear program. The paper's contributions are four-fold:

— The integration of location, service rates and prices as decision variables within a user-choice process based on service level, queueing and pricing considerations.
— The integration of congestion and competition in the context of mill pricing, i.e., prices that can vary between facilities.
— The explicit modelling of the queueing process that takes place at the facilities.
— The design of an efficient heuristic algorithm based on mixed discrete-continuous linear approximations and reformulations.

The remainder of this paper is organized as follows. In Section 4.1.1, we provide an overview of the existing facility location and pricing literature. Section 4.2 is devoted to the model, while, in Section 4.3, we describe the algorithmic framework. Numerical experiments and a discussion of our results are reported in Section 4.4. Finally, in Section 4.5, we draw conclusions and mention possible extensions of the current work.

4.1.1. Literature Review

In this section, we outline works that are relevant to ours, either from the modelling (facility location, pricing, user equilibrium) or computational (bilevel programming, MPECs)

points of view. For a more complete overview on facility location and pricing, one may refer to [**Eiselt et al., 1993**].

Although the facility location problem (FLP) has a rich history, most works disregard user behaviour related to congestion and competition, i.e., similar users are assigned to a single path leading to the facility they patronize. While some models incorporate congestion in the form of capacity limits, more elaborate ones capture congestion through nonlinear functions that can be derived from queueing theory.

With respect to congestion, an early model can be found in [**Desrochers et al., 1995**], who studies a centralized facility location problem where travel time increases with traffic, and users are assigned in a way that minimizes the total delay and costs. Towards the end, the authors mention a bilevel user-choice version of their model, but do not provide a solution algorithm. Within the same centralized framework, [**Fischetti et al., 2016**] propose a Benders decomposition method for a capacitated FLP. Similarly, [**Marianov, 2003**] formulates a model for locating facilities in a centralized system subject to congestion, and where demand is elastic with respect to travel time and queue length. Users are assigned to centers that maximize total demand. In [**Castillo et al., 2009**], users are assigned to facilities so as to minimize the sum of the number of waiting customers and the total opening and service costs. Similar to [**Marianov, 2003**], [**Berman and Drezner, 2006**], [**Aboolian et al., 2008**], and [**Aboolian et al., 2012**] consider models characterized by elastic demand, subject to constraints on the waiting time at facilities. Moreover, in [**Zhang et al., 2010a**] a model maximizing the participation rate is considered, in a preventive healthcare setting, when demand is elastic and users choose the facilities to patronize based on the waiting and travel time. Note that neither of the above papers consider competition or pricing.

With respect to competitive congested facility location problems (CC–FLP), we mention the work of [**Marianov et al., 2008**], who were the first to address congestion within a competitive user-choice environment. Similarly, [**Sun et al., 2008**] consider a generic bilevel facility location model, in which the upper level selects locations with the aim of minimizing the sum of total cost and a congestion function, while the lower level (users) minimizes a nonlinear cost. Both papers employ heuristics for solving their model. A more recent development is that of [**Dan and Marcotte, 2017**], who solve the competitive congested FLP using matheuristics and approximation algorithms. The present work can be considered a

pricing extension of [**Dan and Marcotte, 2017**]. Moreover, [**Ljubić and Moreno, 2018**] address a market share-maximization competitive FLP, where captured customer demand is represented by a multinomial logit model. The authors solve this problem using two branch-and-cut techniques, namely outer approximation cuts and submodular cuts.

The pricing literature is vast. Actually, many authors have addressed joint location and pricing problems, the common practice being to operate in a hierarchical manner: locations are specified first, and then price competition is defined according to the Bertrand model [**Pérez et al., 2004, Panin et al., 2014**]. This approach can be justified by the fact that location decisions are frequently made for the long term, while prices may fluctuate in the short term. However, determining the pricing strategy after the locations have been set limits the price choices and can yield sub-optimal locations, as argued in [**Hwang and Mai, 1990, Cheung and Wang, 1995, Aboolian et al., 2008**]. A joint decision is more suited in some practical applications, and can provide valuable insights into whether or not entry into a market is profitable.

To the best of our knowledge, the first paper to consider simultaneous decisions on location, price and capacity is [**Dobson and Stavrulaki, 2007**], who investigate a monopolistic market where a firm sells a product to customers located on the Hotelling line [**Hotelling, 1929**]. In his PhD thesis, [**Tong, 2011**] considers two profit maximizing models in a network, single-facility and multi-facility, respectively. Competition is not present, and demand is elastic with respect to travel distance, waiting time and price. The author analyzes both a centralized and a user-choice system. Within the same framework, [**Abouee-Mehrizi et al., 2011**] consider a model in which demand is elastic with respect to price only, and clients spread among facilities based on proximity, according to a multinomial Logit random utility model. Congestion, which arises at facilities, is characterized by queueing equations, and customers might balk upon arrival. Furthermore, [**Pahlavani and Saidi-Mehrabad, 2011**] address a pricing problem within a user-choice competitive network. Locations are fixed, and users select the facility to patronize based on price and proximity. Also, they might balk and veer, upon observation of the queue length. The authors propose two metaheuristics for solving their model. More recent contributions are given by [**Hajipour et al., 2016**] and [**Tavakkoli-Moghaddam et al., 2017**], who investigate multi-objective models for the centralized facility location problem with congestion

and pricing policies. Demand is elastic with respect to price and distance, while profit and congestion (waiting time, and idle probability) are decision variables. An extensive review of the literature concerning competition in queueing systems is provided in [**Hassin, 2016**].

From the algorithmic point of view, our approach borrows ideas from the bilevel pricing literature, which was initiated by [**Labbé et al., 1998**] and extended along several directions to include population heterogeneity, congestion or design, as exemplified in the papers by [**Meng et al., 2012**] or [**Brotcorne et al., 2008**], to name only two representative publications. We will in particular adapt a linearization technique introduced in [**Julsain, 1999**] for coping with pricing of the arcs of a packet-switched communication network.

4.2. Model formulation

The problem under consideration involves a firm that enters a market that is already served by competitors that can accommodate the total demand. At the upper level of the hierarchical model, a firm must make decisions pertaining to location, prices and quality of service, anticipating that users will reach an equilibrium where individual utilities are maximized. Note that, when it comes to pricing, three strategies are typically considered [**Hanjoul et al., 1990**]:

— mill pricing: prices can vary between facilities;

— uniform pricing: all facilities charge the same price;

— discriminative pricing: customers patronizing the same facility can be charged different prices.

In the present work, we consider mill pricing, which is actually the most challenging from the computational point of view.

At the lower level, customers purchase an item (this could be a service as well) at the facility where their disutility, expressed as the weighted sum of (constant) travel time, queueing delay, and price, is minimized. For the sake of simplicity, facilities are modelled as $M/M/1$ queues, endowed with only one server. Nevertheless, any $M/M/s$ queues can be considered, provided that the number of server s is fixed, and the decision variable is the service rate μ. The assignment of users to facilities thus follows Wardrop's user equilibrium principle, i.e., disutility is minimized with respect to current flows.

We now introduce the parameters and variables of the model.

sets

I: set of demand nodes;

J: set of candidate facility locations (leader and competitor);

J_1: set of leader's candidate sites;

$J_1^* \subseteq J_1$: set of leader's open facilities;

J_c: set of competition's facilities;

$J^* \subseteq J$: set of open facilities (leader and competitor).

parameters

d_i: demand originating from node $i \in I$;

t_{ij}: travel time between nodes $i \in I$ and $j \in J$;

α: coefficient of the waiting time in the disutility formula;

β: coefficient of the price in the disutility formula;

f_c: fixed cost associated with opening a new facility;

v_c : cost per unit of service.

decision variables

y_j: binary variable set to 1 if a facility is open at site j, and to 0 otherwise;

μ_j: service rate at a facility $j \in J$; typically 0 if the facility is closed;

p_j: price at an open facility $j \in J$.

additional variables

x_{ij}: arrival rate at facility $j \in J$ originating from demand node $i \in I$;

$\lambda_j = \sum_{i \in I} x_{ij}$: arrival rate at node $j \in J$;

w_j: mean queueing time at facility j.

At an open facility j, the mean waiting time in the system, w_j, is a bivariate function depending on both the arrival rate and the service rate, namely

$$w_j(\lambda_j, \mu_j) = \frac{1}{\mu_j - \lambda_j} \quad . \tag{4.2.1}$$

In the above, the waiting time w_j is only defined for open facilities, i.e., those for which μ_j is positive. However, one can generalize Eq. (4.2.1) to all facilities, open or not, through multiplication by $\mu_j - \lambda_j$:

$$w_j \mu_j - w_j \lambda_j = y_j \ . \tag{4.2.2}$$

Indeed, when facility j is closed, $y_j = \mu_j = \lambda_j = 0$, and w_j can assume any value. On the other hand, Eqs. (4.2.1) and (4.2.2) are equivalent when $y_j = 1$. Nevertheless, for simplicity and without loss of generality, we keep the original form (4.2.1) in the model, and will specify in Section 4.3.3 how we deal with null service rates.

At the lower level, let γ_i denote the minimum disutility for users originating from node i. The Wardrop conditions are expressed as the set of logical constraints

$$t_{ij} + \alpha w(\lambda_j, \mu_j) + \beta p_j \begin{cases} = \gamma_i, & \text{if } x_{ij} > 0 \\ \geq \gamma_i, & \text{if } x_{ij} = 0 \end{cases} \quad i \in I;\ j \in J \ .$$

In other words, the utility of the paths having positive flow must be lower or equal than the utility of paths carrying no flow. These conditions can alternatively be formulated as the complementarity system

$$\begin{aligned} & t_{ij} + \alpha w(\lambda_j, \mu_j) + \beta p_j - \gamma_i \geq 0 && i \in I;\ j \in J \\ & x_{ij} \cdot (t_{ij} + \alpha w(\lambda_j, \mu_j) + \beta p_j - \gamma_i) = 0 && i \in I;\ j \in J \\ & x_{ij} \geq 0 && i \in I;\ j \in J \ . \end{aligned}$$

Our model is as follows:

P: (Leader:)

$$\max_{y,\mu,x,p,\gamma} \quad z = \sum_{i \in I} \sum_{j \in J_1} x_{ij} p_j - \sum_{j \in J_1} (f_c \cdot y_j + v_c \cdot \mu_j) \tag{4.2.3}$$

$$\text{s.t.} \quad \mu_j \leq M_1 \cdot y_j \qquad j \in J_1 \tag{4.2.4}$$

$$y_j \in \{0, 1\} \qquad j \in J_1 \tag{4.2.5}$$

$$\mu_j \geq 0 \qquad j \in J_1 \tag{4.2.6}$$

(Users:)

$$t_{ij} + \alpha w(\lambda_j, \mu_j) + \beta p_j - \gamma_i \geq 0 \qquad i \in I;\ j \in J \tag{4.2.7}$$

$$x_{ij} \cdot (t_{ij} + \alpha w(\lambda_j, \mu_j) + \beta p_j - \gamma_i) = 0 \qquad i \in I;\ j \in J \tag{4.2.8}$$

$$w_j = \frac{1}{\mu_j - \lambda_j} \qquad j \in J \tag{4.2.9}$$

$$\lambda_j = \sum_{i \in I} x_{ij} \qquad j \in J \qquad (4.2.10)$$

$$\sum_{j \in J} x_{ij} = d_i \qquad i \in I \qquad (4.2.11)$$

$$\lambda_j \leq \mu_j \qquad j \in J \qquad (4.2.12)$$

$$x_{ij} \geq 0 \qquad i \in I; j \in J\ . \qquad (4.2.13)$$

The decision variables are the vectors y (locations) and μ (number of servers), while the user assignment x is the solution of an equilibrium problem that can be reduced to a convex optimization problem. The leader's objective in Eq. (4.2.3) is to maximize the difference between the total profit and the opening and service costs. Constraints (4.2.4) ensure that the service rate is strictly positive only at open facilities. When $y = 1$, it also helps strengthening the formulation by computing a tight value for M_1 such that μ values yielding solely negative profit are eliminated.

Constraints (4.2.7), (4.2.8) and (4.2.13) characterize the user equilibrium problem, where γ_i is the optimal disutility that users originating from node i are willing to incur. Typically, the equilibrium equations should only be enforced for open facilities. However, in our case, they are automatically satisfied for closed facilities, for the following reason: if a facility j closed, the service rate μ_j, and implicitly λ_j and x_{ij}, will be null, which implies that Eq. (4.2.8) is satisfied. Additionally, in our model, p_j can take any value for a closed facility (although this is sub-optimal from an economic standpoint), as its contribution to the objective value is canceled by the null terms x_{ij}. It follows that Eq. (4.2.7) is also satisfied. Finally, constraints (4.2.11) ensures that the total number of users originating from a demand point amounts to the demand associated to this node, and Eq. (4.2.12) guarantees that the arrival rate does not exceed the service rate at facility j.

For the sake of illustration, let us consider the example corresponding to the graph and data of Figure 4.1, where nodes 1 and 2 are endowed with a demand of 35 and 30, respectively. The competitor's facility situated at node C operates at a service rate of 70.5, and charges a price of 12. The fixed and variable costs are set at 50 and 1, respectively, and parameters $\alpha = 20$, $\beta = 10$. In this example, the leader opens facilities at both available sites. The profit is shown as a function of the prices charged at the two facilities, so the service rates have been fixed at 37.3 for A and 32.5 for B.

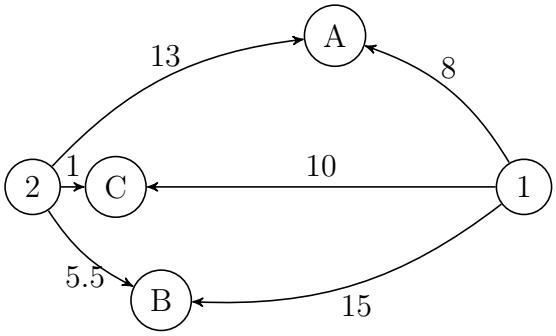


Figure 4.1. Example of a 2-demand node network, 2 location candidate sites.

The associated profit curve is illustrated in Figure 4.2. While it lacks the discontinuities associated with the basic network pricing problem (see [**Labbé et al., 1998**]), due to the smoothing effect of the nonlinear queueing terms, it is still highly nonlinear and nonconvex.

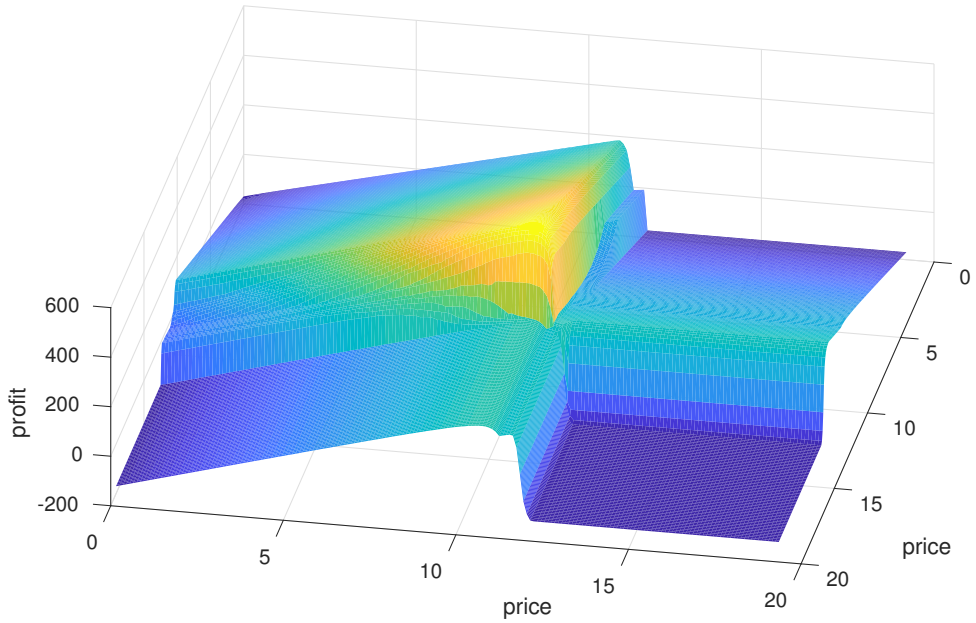


Figure 4.2. Profit associated with open facilities A and B, for the network displayed in Figure 4.1

Observation 1. *It is trivial to show that the waiting time w_j is jointly convex in μ_j and λ_j, for all $\mu_j > 0, \lambda_j < \mu_j$.*

4.3. A mixed-integer linear approximation

The general idea that underlies the algorithmic approach is to replace the original problem by a more manageable mixed-integer linear program (MILP), that we can further solve using an off-the-shelf software. This idea is not entirely novel, as it has been exploited before with different variants. For instance, in [**Dan and Marcotte, 2017**], the lower-level problem is linearized using tangent planes, before the optimality conditions are written. This yields a model containing bilinear and other nonlinear terms, which are further linearized, for instance, by using the triangle method of [**D'Ambrosio et al., 2010**]. Our approach is related to that of [**Julsain, 1999**], where univariate congestion functions are linearized in the context of a network pricing problem. In our case, concepts from network pricing and CC–FLP are merged into a single model, which makes the problem much more challenging by the presence of facility location and service level decision variables, as well as bivariate queueing delays.

The main steps of our resolution method are:

— Replace the bilinear terms in the objective with functions derived from the equilibrium constraints.

— Perform linear approximations of the complementarity constraints and the remaining nonlinear terms through the introduction of binary variables and 'big-M' constants.

— Use off-the-shelf MILP technology to solve the resulting MILP, or a carefully-designed sequence of MILPs.

4.3.1. Reformulation of the objective function

The key issue is to eliminate the bilinear terms $x_{ij}p_j,\ j \in J_1$, in the objective, through substitution and other algebraic manipulations of the model's constraints. From Eq. (4.2.8) we have

$$x_{ij}p_j = -\frac{1}{\beta}\left(t_{ij}x_{ij} + \alpha x_{ij}w_j - x_{ij}\gamma_i\right), \quad j \in J_1, \tag{4.3.1}$$

whose summation over $i \in I$ and $j \in J_1$ leads to

$$\sum_{i\in I}\sum_{j\in J_1} x_{ij}p_j = -\frac{1}{\beta}\left(\sum_{i\in I}\sum_{j\in J_1} t_{ij}x_{ij} + \alpha\sum_{i\in I}\sum_{j\in J_1} x_{ij}w_j - \sum_{i\in I}\sum_{j\in J_1} x_{ij}\gamma_i\right), \quad j \in J_1. \tag{4.3.2}$$

The RHS of Eq. (4.3.2) now contains linear and nonlinear terms. We can simplify some of the most 'complicating' ones, namely the bilinear $x_{ij}\gamma_i$, as follows.

$$\sum_{i\in I}\sum_{j\in J_1} x_{ij}\gamma_i = \sum_{i\in I}\left(\sum_{j\in J} x_{ij}\gamma_i - \sum_{j\in J_c} x_{ij}\gamma_i\right), \tag{4.3.3}$$

and since $J = J_1 \cup J_c$ and $J_1 \cap J_c = \emptyset$,

$$\sum_{i\in I}\sum_{j\in J_1} x_{ij}\gamma_i = \sum_{i\in I} d_i\gamma_i - \sum_{i\in I}\sum_{j\in J_c} x_{ij}\gamma_i. \tag{4.3.4}$$

For the bilinear terms $x_{ij}\gamma_i$ in the RHS of Eq. (4.3.4), we write the following equations, derived from Eq. (4.2.8).

$$x_{ij}\gamma_i = t_{ij}x_{ij} + \alpha x_{ij}w_j + \beta x_{ij}p_j, \quad i \in I, j \in J_c \tag{4.3.5}$$

or, equivalently,

$$\sum_{i\in I}\sum_{j\in J_c} x_{ij}\gamma_i = \sum_{i\in I}\sum_{j\in J_c}\left(t_{ij}x_{ij} + \alpha x_{ij}w_j + \beta x_{ij}p_j\right). \tag{4.3.6}$$

Recall that the price is fixed at competitors' facilities (i.e., $j \in J_c$), so $x_{ij}p_j$ is not a bilinear term when $j \in J_c$. Then, the only nonlinear terms in the RHS of Eq. (4.3.6) are $x_{ij}w_j$. Putting together Eqs. (4.3.2), (4.3.4) and (4.3.6) yields:

$$\sum_{i\in I}\sum_{j\in J_1} x_{ij}p_j = -\frac{1}{\beta}\left(\sum_{i\in I}\sum_{j\in J} t_{ij}x_{ij} + \alpha\sum_{j\in J}\frac{\sum_{i\in I} x_{ij}}{\mu_j - \sum_{i\in I} x_{ij}} - \sum_{i\in I} d_i\gamma_i + \beta\sum_{i\in I}\sum_{j\in J_c} p_j x_{ij}\right)$$

and, since $\lambda_j = \sum_{i\in I} x_{ij}$, the objective function can be written as

$$z = -\frac{1}{\beta}\sum_{i\in I}\sum_{j\in J} t_{ij}x_{ij} - \frac{\alpha}{\beta}\sum_{j\in J}\frac{\lambda_j}{\mu_j - \lambda_j} + \sum_{i\in I}\frac{d_i}{\beta}\gamma_i - \sum_{i\in I}\sum_{j\in J_c} p_j x_{ij} - \sum_{j\in J_1}\left(f_c \cdot y_j + v_c \cdot \mu_j\right). \tag{4.3.7}$$

All terms in Eq. (4.3.7) are linear, with the exception of $\lambda_j/(\mu_j - \lambda_j)$. Additionally, these terms are undefined for $\mu_j = 0$. We overcome these issues during the linearization process, as mentioned in Section 4.3.3. We now discuss some of their properties.

Proposition 4.3.1. *Each term* $\frac{\alpha}{\beta}\frac{\lambda_j}{\mu_j - \lambda_j}$ *is*

— *convex in* λ_j*, and convex in* μ_j

— *neither convex, nor concave jointly in* λ_j *and* μ_j *(see Figure 4.1).*

— *pseudolinear jointly in* λ_j *and* μ_j*.*

PROOF. The first statement is obvious. The proof of the second rests on the fact that the Hessian of the function $f(x,y) = x/(y-x)$ is indefinite. As for the pseudolinearity claim, let us consider pseudoconcavity first. The gradient of f is

$$\nabla f(x,y) = \left(\frac{y}{(y-x)^2}, \frac{-x}{(y-x)^2} \right)$$

Let $a = (x_a, y_a)$ and $b = (x_b, y_b)$, such that $\nabla f(a) \cdot (b-a) \geq 0$. We have that

$$\nabla f(a) \cdot (b-a) = \left(\frac{y_a}{(y_a - x_a)^2}, \frac{-x_a}{(y_a - x_a)^2} \right) \cdot (x_b - x_a, y_b - y_a) = \frac{y_a x_b - x_a y_b}{(y_a - x_a)^2} \tag{4.3.8}$$

and

$$\frac{y_a x_b - x_a y_b}{(y_a - x_a)^2} \geq 0 \Rightarrow y_a x_b - x_a y_b \geq 0 \quad . \tag{4.3.9}$$

We now proceed by contradiction. Let us assume that $f(b) < f(a)$. Then, $x_b/(y_b - x_b) < x_a/(y_a - x_a)$. This means that $x_b y_a - x_a y_b < 0$ and $x_b y_a - x_a y_b \geq 0$ by Eq. (4.3.9), a contradiction. This implies that

$$\nabla f(a) \cdot (b-a) \geq 0 \Rightarrow f(a) \leq f(b), \tag{4.3.10}$$

as required.

Using the same arguments, we can prove the pseudoconvexity of $-f$, and the pseudolinearity of $(\alpha/\beta)(\lambda_j)/(\mu_j - \lambda_j)$ follows.

□

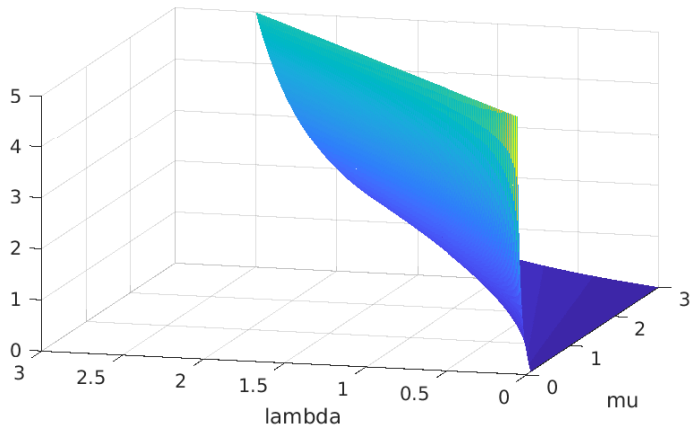


Figure 4.1. Function $x/(y-x)$. Although neither convex nor concave, it is pseudolinear (pseudoconvex, and pseudoconcave). The non-convexity is more accentuated in the vicinity of the origin.

4.3.2. Bounds on w, p and μ

Special attention is paid to tight bounds on the variables, since these will improve the numerical efficiency of the resolution algorithm. Based on the parameters of the network, we can derive upper and lower bounds for the waiting time at facilities, the prices set by the emerging firm, and the service rate profitable for the leader. It is obvious that, in order to make nonnegative profit, the minimum price that the leader can set must exceed the variable cost v_c associated with the service rate

$$p_{\min} = v_c.$$

Let (x', λ', w') be the solution of the lower level problem under a competing monopoly. Then, the maximum utility that users are willing to incur in order to access an item or a service is

$$u_{\max} = \max_{i \in I, j \in J_c} \left\{ t_{i,j} + \alpha w'_j + \beta p_j \right\} \ .$$

The equilibrium constraints guarantee that the above equation is satisfied even when the new firm enters the market. Then, for all couples (i, j) that have positive flows, the associated utility cannot exceed $u_{\max}$

$$t_{i,j} + \alpha w_j + \beta p_j \leq u_{\max},$$

and the bounds on p and w follow directly

$$\begin{aligned} &w_j \leq (u_{\max} - \beta p_{\min})/\alpha \ , && p_j \leq u_{\max}/\beta \\ &w_{\max} = (u_{\max} - \beta p_{\min})/\alpha \ , \text{ and} && p_{\max} = u_{\max}/\beta. \end{aligned} \tag{4.3.11}$$

The service rate at any given facility is limited by the service cost, the maximum price, fixed cost, and total demand. The maximum possible profit of the firm is obtained when all the demand is attracted, the maximum price is charged, and only one facility is open (fixed cost is minimal). Since the profit (objective function) must be nonnegative, we must have that

$$p_{\max} \sum_{i \in I} d_i - f_c - \mu_{\max} v_c \geq 0,$$

and the upper bound on μ follows directly:

$$\mu_{\max} = \frac{p_{\max}}{v_c} \sum_{i \in I} d_i - \frac{f_c}{v_c}.$$

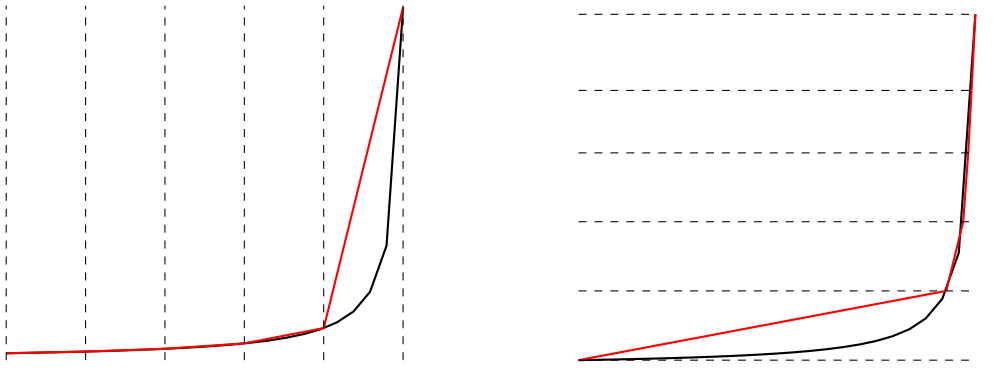
4.3.3. Linear approximation

This section is devoted to a detailed description of the techniques that allow to transform the original problem into a mixed integer linear program.

Sampling. We performed piecewise linear interpolations of the nonlinear functions involved in our model, namely $\lambda_j/(\mu_j - \lambda_j)$ and $1/(\mu_j - \lambda_j)$. These functions are bivariate for the leader (μ is a decision variable), and univariate for the competitor.

For the leader, we consider $N_\mu + 1$ equidistant sampling points on the x axis, within the interval $[0, \mu_{\max}]$: $\{\tilde{\mu}^n\}, n \in \{1, \ldots, N_\mu\}$ such that $\tilde{\mu}^i < \tilde{\mu}^j$ for all $1 \leq i < j \leq N_\mu$. Next, for each sample $\tilde{\mu}^n$ we define $\lambda^n_{\max} = \tilde{\mu}^n - 1/w_{\max}$, and we sample each interval $[0, \lambda^n_{\max}]$ using N_λ points $\{\tilde{\lambda}^{nk}\}, k \in \{1, \ldots, N_\lambda\}$, such that $\tilde{\lambda}^{ni} < \tilde{\lambda}^{nj}$ for all $1 \leq i < j \leq N_\lambda$. A similar sampling is performed for every facility of the competitor, where the sampling interval for λ is $[0, \mu_j],\ \forall j \in J_c$.

Special attention is paid to the type of sampling we use for λ. The sampling can be equidistant either 'horizontally' or 'vertically'. In the 'horizontal' case, for a given $\tilde{\mu}^n$ the difference between two consecutive values, $\tilde{\lambda}^{ni} - \tilde{\lambda}^{ni+1}$, remains constant. In contrast, in the vertical case, the samples are computed such that, for a given $\tilde{\mu}^n$, and for any two consecutive λ samples, $\tilde{\lambda}^{ni}$ and $\tilde{\lambda}^{ni+1}$, the difference between their respective waiting time, $1/(\tilde{\mu}^n - \tilde{\lambda}^{ni}) - 1/(\tilde{\mu}^n - \tilde{\lambda}^{ni+1})$, is constant. Both cases are illustrated in Figure 4.2 below.



(a) sampling equidistant on x axis **(b)** sampling equidistant on y axis

Figure 4.2. Illustration of the impact of sampling type on the approximation.

When using samples that are equidistant on the x axis, the approximation of waiting times is best on the region where the slope is small. It is important that this function be

well approximated in this area, as a small change in the waiting time value would cause a significant change in the x-variable, thus approximate badly the resulting objective function. On the other hand, a rougher approximation of the congested part would not yield a large error in the x-values, which justifies performing the sampling equidistant on y axis.

Piecewise linearization. We now detail the linear approximation of the terms $\frac{\lambda_j}{\mu_j - \lambda_j}$ in the reformulated objective function, and $\frac{1}{\mu_j - \lambda_j}$ in constraints (4.2.9). To this end, we use the sampling described above in a triangle piecewise linearization technique from [**D'Ambrosio et al., 2010**]. At a given point $(\tilde{\lambda}, \tilde{\mu})$ the function of interest is approximated by a convex combination of the function values at the vertices of the triangle containing the point $(\tilde{\lambda}, \tilde{\mu})$.

First, we approximate $\frac{\lambda_j}{\mu_j - \lambda_j}$ and $\frac{1}{\mu_j - \lambda_j}$ for the leader, using the following sets of variables:

— $\underline{l}_{j,n,k}$ and $\bar{l}_{j,n,k}$ are binary variables denoting the lower and upper triangles, respectively, used for evaluating the convex combinations for $n \in \{1, \ldots, N_\mu\}, k \in \{1, \ldots, N_\lambda\}, j \in J$. In a feasible solution these variables equal 1 if the point of interest falls inside their associated triangle, and 0 otherwise.

— $\overline{s}_{j,n,k}$ represents the weight of the convex combination associated with the vertices of the triangle containing the point of interest.

— $\overline{u}$ and $\overline{w}$ hold the approximated values of $\frac{\lambda_j}{\mu_j - \lambda_j}$ and $\frac{1}{\mu_j - \lambda_j}$, respectively.

The following constraints allow to linearize $\frac{\lambda_j}{\mu_j - \lambda_j}$ and $\frac{1}{\mu_j - \lambda_j}$ in the original model. Since they are not defined for $\mu_j = 0$, by convention, we set them to 0, whenever $\mu_j = 0$. The motivation is that users cannot patronize a facility offering no service, yielding a null waiting time at facilities. To accommodate this, the summation starts at $n = 2$ in constraints (4.3.17) and (4.3.18), below.

$$\sum_{n=1}^{N_\mu} \sum_{k=1}^{N_\lambda} \left(\bar{l}_{j,n,k} + \underline{l}_{j,n,k}\right) = 1 \qquad j \in J_1 \tag{4.3.12}$$

$$\overline{s}_{j,n,k} \leq \bar{l}_{j,n-1,k} + \underline{l}_{j,n-1,k-1} + \bar{l}_{j,n,k} + \underline{l}_{j,n,k} + \bar{l}_{j,n-1,k-1} + \underline{l}_{j,n,k-1}$$
$$j \in J_1; n \in \{1, \ldots, N_\mu\}; k \in \{1, \ldots, N_\lambda\} \tag{4.3.13}$$

$$\sum_{n=1}^{N_\mu}\sum_{k=1}^{N_\lambda}\overline{s}_{j,n,k}=1 \qquad j\in J_1 \tag{4.3.14}$$

$$\lambda_j=\sum_{n=1}^{N_\mu}\sum_{k=1}^{N_\lambda}\overline{s}_{j,n,k}\tilde{\lambda}^{nk} \qquad j\in J_1 \tag{4.3.15}$$

$$\mu_j=\sum_{n=1}^{N_\mu}\sum_{k=1}^{N_\lambda}\overline{s}_{j,n,k}\tilde{\mu}^{n} \qquad j\in J_1 \tag{4.3.16}$$

$$\overline{w}_j=\sum_{n=2}^{N_\mu}\sum_{k=1}^{N_\lambda}\frac{1}{\tilde{\mu}^n-\tilde{\lambda}^{nk}}\cdot\overline{s}_{j,n,k} \qquad j\in J_1 \tag{4.3.17}$$

$$\overline{u}_j=\sum_{n=2}^{N_\mu}\sum_{k=1}^{N_\lambda}\frac{\tilde{\lambda}^{nk}}{\tilde{\mu}^n-\tilde{\lambda}^{nk}}\cdot\overline{s}_{j,n,k} \qquad j\in J_1 \tag{4.3.18}$$

$$\overline{l}_{j,n,k},\underline{l}_{j,n,k}\in\{0,1\} \qquad j\in J_1; n\in\{1,\dots,N_\mu\}; k\in\{1,\dots,N_\lambda\} \tag{4.3.19}$$

$$0\le\overline{s}_{j,n,k}\le 1 \qquad j\in J_1; n\in\{1,\dots,N_\mu\}; k\in\{1,\dots,N_\lambda\} \tag{4.3.20}$$

$$\overline{l}_{j,n,0}=0,\quad \underline{l}_{j,n,0}=0 \qquad j\in J_1; n\in\{0,\dots,N_\mu\} \tag{4.3.21}$$

$$\overline{l}_{j,n,N_\lambda}=0,\quad \underline{l}_{j,n,N_\lambda}=0 \qquad j\in J_1; n\in\{0,\dots,N_\mu\} \tag{4.3.22}$$

$$\overline{l}_{j,0,k}=0,\quad \underline{l}_{j,0k}=0 \qquad j\in J_1; k\in\{0,\dots,N_\lambda\} \tag{4.3.23}$$

$$\overline{l}_{j,N_\mu,k}=0,\quad \underline{l}_{j,N_\mu,k}=0 \qquad j\in J_1; k\in\{0,\dots,N_\lambda\}. \tag{4.3.24}$$

We perform a similar linearization for the competitor. Recall that, in this case, the service rate, μ_j, is constant. We introduce variables, $\hat{l}$, $\hat{s}$ $\hat{w}$ and $\hat{u}$, having similar meaning to their leader counterparts. Given $w_{\max}$, we compute $\hat{\lambda}^j_{\max}=\mu_j-1/w_{\max}$, and we sample the interval $[0,\hat{\lambda}^j_{\max}]$ using N_c points $\hat{\lambda}^{jn}, n\in\{1,\dots,N_c\}$ such that $\hat{\lambda}^{jn}<\hat{\lambda}^{jm}$ for all $1\le n<m\le N_c$, and obtain the linearization

$$\sum_{n=1}^{N_c}\hat{s}_{j,n}=1 \qquad j\in J_c \tag{4.3.25}$$

$$\lambda_j=\sum_{n=1}^{N_c}\hat{s}_{j,n}\hat{\lambda}^{jn} \qquad j\in J_c \tag{4.3.26}$$

$$\hat{w}_j=\sum_{n=1}^{N_c}\frac{1}{\mu_j-\hat{\lambda}^{jn}}\cdot\hat{s}_{j,n} \qquad j\in J_c \tag{4.3.27}$$

$$\hat{u}_j = \sum_{n=1}^{N_c} \frac{\hat{\lambda}^{j,n}}{\mu_j - \hat{\lambda}^{jn}} \cdot \hat{s}_{j,n} \qquad j \in J_c \tag{4.3.28}$$

$$\sum_{n=1}^{N_c} \hat{l}_{j,n} = 1 \qquad j \in J_c \tag{4.3.29}$$

$$\hat{s}_{j,n} \leq \hat{l}_{j,n} + \hat{l}_{j,n-1} \qquad j \in J_c;\ n \in \{1, \ldots, N_c\} \tag{4.3.30}$$

$$\hat{l}_{j,n} \in \{0, 1\} \qquad j \in J_c;\ n \in \{1, \ldots, N_c\} \tag{4.3.31}$$

$$0 \leq \hat{s}_{j,n} \leq 1 \qquad j \in J_c;\ n \in \{1, \ldots, N_c\} \tag{4.3.32}$$

$$\hat{l}_{j,0} = 0, \hat{l}_{j,N_c} = 0 \qquad j \in J_c\ . \tag{4.3.33}$$

At last, the complementarity constraints Eq. (4.2.8) are linearized through the introduction of binary variables and big-M constants, as follows:

$$t_{ij} + \alpha \overline{w}_j + \beta p_j - \gamma_i \leq M_2 s_{ij} \qquad i \in I;\ j \in J_1 \tag{4.3.34}$$

$$t_{ij} + \alpha \hat{w}_j + \beta p_j - \gamma_i \leq M_2 s_{ij} \qquad i \in I;\ j \in J_c \tag{4.3.35}$$

$$x_{ij} \leq M_3 (1 - s_{ij}) \qquad i \in I;\ j \in J \tag{4.3.36}$$

$$s_{ij} \in \{0, 1\} \qquad i \in I;\ j \in J\ . \tag{4.3.37}$$

The values of M_2 and M_3 must be sufficiently large not to forbid feasible solutions, but not too large that they slow down the enumeration algorithm, due to a weak continuous relaxation. Based on the network's parameters, the following 'tight' values for M_2 and M_3 hold:

$$M_2 = \max_{i \in I, j \in J} \{t_{ij}\} + \alpha w_{\max} + \beta p_{\max}$$

$$M_3 = \max_{i \in I} \{d_i\}\ .$$

Putting together all linear terms yields the following MILP approximation of P:

PL:

$$
\begin{aligned}
\max_{y,c,x,\gamma} \quad & z = -\frac{1}{\beta}\sum_{i\in I}\sum_{j\in J} t_{ij}x_{ij} - \frac{\alpha}{\beta}\sum_{j\in J_1}\overline{u}_j - \frac{\alpha}{\beta}\sum_{j\in J_c}\hat{u}_j + \sum_{i\in I}\frac{d_i}{\beta}\gamma_i - \sum_{i\in I}\sum_{j\in J_c} p_j x_{ij} - \sum_{j\in J_1}(f_c\cdot y_j + v_c\cdot \mu_j) \\
\text{s.t.} \quad & t_{ij} + \alpha\overline{w} + \beta p_j - \gamma_i \geq 0 && i\in I;\ j\in J_1 \\
& t_{ij} + \alpha\hat{w} + \beta p_j - \gamma_i \geq 0 && i\in I;\ j\in J_c \\
& x_{ij}\cdot(t_{ij} + \alpha w(\lambda_j,\mu_j) + \beta p_j - \gamma_i) = 0 && i\in I;\ j\in J \\
& \text{constraints (4.2.4)–(4.2.6), (4.2.10)–(4.2.13), (4.3.12)–(4.3.37) .}
\end{aligned}
\tag{4.3.38}
$$

An interesting feature of this reformulation-linearization is that, since we use the same set of variables and constraints to approximate two different functions simultaneously, the number of variables is greatly reduced. This would not be the case if we were to linearize separately the waiting time and the bilinear terms $x_{ij}p_j$ present in the original formulation.

Another interesting feature of this reformulation is the pseudo-linearity of the functions replacing the bilinear terms in the objective. Although we do not exploit this property directly, we expect the linearization to be well-behaved.

Finally, an alternative algorithmic approach based on the power-based transformation originally proposed in [**Teles et al., 2011**] was initially implemented but did not perform satisfactorily. The main idea is to transform nonlinear polynomial problems into an MILP, by applying a term-wise disaggregation scheme, notwithstanding, with additional discrete and continuous variables. Kolodziej et. al incorporate this technique into a global optimization algorithm for bilinear programs [**Kolodziej et al., 2013**]. The authors argue that this technique scales better than the piecewise McCormick envelopes, and is comparable with global optimization solvers.

For the sake of completeness, and to warn other people tempted by that path, we thought it is useful to mention it. The interested reader can find it in the appendix of this Ph.D. thesis [**?**].

4.4. Experimental Setup and Results

The algorithm has been tested on randomly generated data. We focused on challenging instances, in which, at optimality, the number of open facilities represent more than one fifth of the nodes. Our experiments were conducted for different problem sizes, namely for 10, 15, 20, and 25 nodes, which were generated as follows. The travel time between nodes varied uniformly between 0 and 600. In order to ensure that problems were difficult enough, the combinations of fixed and variable costs were chosen such that there exist feasible solutions yielding nonnegative profit, where more than half of the facilities were open.

All procedures were implemented in Java, and the MILP formulations were solved by IBM CPLEX Optimizer version 12.6. The tests were performed on a computer equipped with 96 GB of RAM, and two 6-core Intel(R) Xeon(R) X5675 processors running at 3.07GHz. The default values of the parameters α and β were set to 20 and 10, respectively, unless specified otherwise. In all tests, the maximum tree size was set to 30GB. Throughout this section, the *estimated objective* refers to the MILP objective value as returned by CPLEX, whereas the *recovered objective* is computed as follows. The decision variables are set to the optimal values found by CPLEX, then the associated objective value is recovered by solving the (convex) lower level problem for its exact solution, by Frank-Wolfe algorithm.

4.4.1. Solving the MILP with different number of samples

An initial set of experiments was intended to assess the performance of the linear approximation method. The algorithm was stopped as soon as the optimality gap dropped below CPLEX's default value (10^{-4}), the 86,400 seconds (24 hours) limit was reached, or the tree size exceeded 30GB. Tables 4.1, 4.2, and 4.3 report the CPU needed, for various number of approximating samples. The relative MILP gap is shown in percentage, next to the CPU. The gap is omitted if the algorithm terminated at optimality (i.e., gap $< 10^{-4}$).

For 5 and 10 samples, the algorithm needs less than 100 seconds, and on average less than 35 seconds, to reach optimality. The CPU increases abruptly with the number of samples, which is to be expected. For 15-node networks, all tests finished at optimality when the number of samples is lower than 60. However, 6 over 10 instances exceeded the alloted time or memory when using 60 samples. For larger, 20-node networks, the algorithm terminated at optimality on very few instances, when using more than 30 samples, and ran out of time

	# of samples						
test #	5	10	20	30	40	60	(gap%)
1	4	9	25	9,473	1,363	14,205	
2	14	20	110	398	3,883	86,409	(10.95)
3	9	26	30	361	19,837	86,404	(9.97)
4	4	32	172	13,814	21,694	34,066	
5	6	18	149	11,025	52,951	73,124	
6	2	5	54	5,982	18,408	86,402	(1.03)
7	5	15	92	18,006	8,831	86,402	(3.94)
8	3	11	51	3,535	9,160	86,402	(1.86)
9	2	10	88	30,486	24,153	86,402	(8.22)
10	1	9	52	8,010	9,830	1,406	
Average	5	14	82	10,109	17,011	64,122	(3.60)

Table 4.1. CPU time (seconds) on 15-node networks, for different number of samples.

on all 25-node network instances. As suggested by Figure 4.1, the good solutions are found in the early stages of the algorithm, while the remaining steps are used to close the gap and prove optimality.

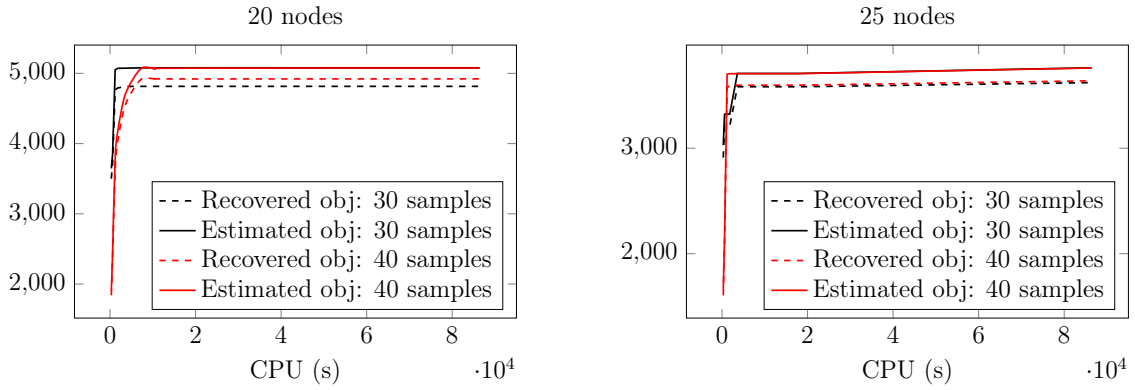


Figure 4.1. Variation of estimated (MILP) and recovered objective value with respect to time.

The increase in the running time is compensated by an improvement in the approximation quality, as illustrated in Figure 4.2. The difference between the estimated (MILP) optimal

	# of samples				
test #	5	10	20 (gap%)	30 (gap%)	40 (gap%)
1	22	94	1,459	64,348 (0.30)	86,402 (5.17)
2	6	15	1,297	59,626	77,542
3	12	52	86,401 (3.60)	86,403 (0.95)	86,402 (2.04)
4	7	24	1,035	1,853	86,401 (0.24)
5	13	20	86,402 (0.27)	86,402 (6.12)	86,401 (4.76)
6	7	13	782	86,402 (0.13)	52,097 (0.75)
7	6	27	228	30,892	86,401 (1.73)
8	7	20	305	2,462	28,330
9	20	78	86,401 (0.07)	86,401 (0.04)	86,402 (6.71)
10	3	9	146	86,401 (0.56)	18,096
Average	10	35	26,446 (0.39)	59,119 (0.81)	69,447 (2.14)

Table 4.2. CPU time (seconds) on 20-node networks, for different number of samples.

objective value, and the recovered one is decreasing with the increase of the number of samples, suggesting a solid improvement in the quality of the approximation.

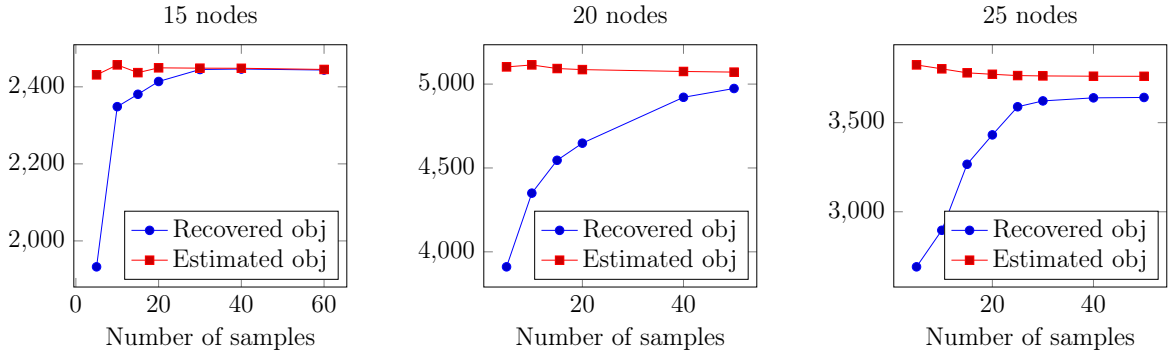


Figure 4.2. Evolution of the MILP objective value ('Estimated') and the true objective value ('Recovered'), as the number of samples increases.

	# of samples				
test #	5	10	20 (gap%)	30 (gap%)	40 (gap%)
1	3	5	143	86,402 (0.59)	22,702 (0.48)
2	9	23	259	5,891 (2.25)	86,403 (3.97)
3	2	11	233	78,143 (0.50)	37,895 (1.15)
4	8	32	86,401 (0.73)	25,010 (0.84)	16,177 (2.51)
5	8	24	86,413 (0.76)	86,401 (4.18)	86,403 (5.14)
6	4	12	58,331	68,406 (2.43)	86,403 (2.27)
7	3	24	86,402 (2.40)	15,545 (3.08)	7,864 (3.88)
8	5	30	9,650	86,405 (3.12)	71,371 (2.50)
9	2	16	170	6,633 (0.69)	68,635 (0.54)
10	3	17	9,127 (0.36)	86,402 (1.57)	8,789 (4.10)
Average	4	19	33,713 (0.43)	54,524 (1.93)	49,264 (2.65)

Table 4.3. CPU time (seconds) on 25-node networks, for different number of samples.

4.4.2. A math-heuristic approach

After careful inspection of the solutions, we have noticed that the number of facilities opened at optimality does not vary significantly with the number of samples, nor with the alloted execution time (on average around $5-7$ are opened for the 20 and 25-node instances). This suggests that quasi-optimal locations are found on the early stages of the algorithm, or for coarse approximations.

Next, we assessed the quality of these opened facilities, restricting the problem to the determination of price and service levels, which remains a difficult nonlinear bilevel problem. We now solve the linearized problem PL using the following algorithm whose main steps are:

I. Solve PL for a small number of samples, and a limited time.

II. Retrieve the locations associated with the incumbent.

III. Solve PL, where locations are fixed from step II, using a more fine-grained sampling, for a limited time.

IV. Retrieve the associated solution (μ and p) and compute the lower-level equilibrium required to obtain the true objective. This last operation can be achieved by solving a convex program. To this purpose, we implemented the classical Franc-Wolfe algorithm.

This matheuristc version of our algorithmic has been tested on instances involving 5, 10 and 30 samples, and a time limit of 30 minutes, at step I, and 40 samples and a time limit of 1 hour in total, for all three steps. Tables 4.4 and 4.5 show the comparison between the values obtained in this way, and the objective values yielded by the original algorithm for 40 and 50-samples approximations, with running time limited to 1 hours, and a 50-sample approximation running for 24 hours, for 20 and 25-node networks, respectively.

	40 samples, 1 hour in total					
test #	from 5 samples	from 10 samples	from 30 samples, 30 min	40 samples, 1 hour	50 samples, 1 hour	50 samples, 24 hours
1	3,454.01	3,454.01	3,454.01	345.14	–	3,455.85
2	4,931.14	4,931.14	4,931.14	4,931.14	4,933.98	4,933.98
3	10,083.30	10,083.30	10,091.46	10,091.46	–	10,145.76
4	4,892.30	4,892.30	4,892.30	4,892.30	4,887.66	4,887.66
5	5,106.06	5,862.84	5,788.60	5,757.25	6,219.17	6,201.88
6	4,200.60	4,200.60	4,200.60	4,200.60	4,227.83	4,227.83
7	4,398.22	4,398.22	4,201.22	4,345.16	4,401.96	4,401.96
8	3,141.79	3,141.79	3,141.79	3,141.79	3,154.11	3,154.11
9	3,318.63	3,318.63	3,318.63	3,291.84	3,325.85	3,354.89
10	–	–	–	–	–	–

Table 4.4. Objective value comparison on 20-node networks, when 40 samples are used for linearization, locations are fixed, and the CPU is limited to 1 hour in total (including the warm start).

For the 20-node networks the best performance corresponds the 10-sample starting point. On 1 instance it outperformed the 50-sample approximation, and on 8 instances it falls, on average, within 2.4% of the optimum found by the latter, at a much smaller computational

cost (1 hour for the 10-sample starting point as opposed to 24 hours for the 50 samples). On most tests, the deviation is less than 1%, but the average is increased by an outlier (instance # 5) that has an error of 11%. The 5 and 30-sample starting point yield similar results. In almost all cases in which the 40 and 50-sample algorithm finds an initial solution in 1 hour, such a solution is as good, or even better than the 40-sample boosted by the 10-sample locations. However, the boosted version looks more robust.

	40 samples, 1 hour in total					
test #	from 5 samples	from 10 samples	from 40 samples, 30 min	40 samples, 1 hour	50 samples, 1 hour	50 samples, 24 hours
1	2,783.93	2,783.93	2,820.28	2,820.28	2,840.15	2,840.15
2	3,653.74	3,751.08	3,775.86	3,775.86	–	3,775.44
3	3,531.34	3,531.34	3,549.39	3,549.39	3,550.34	3,550.34
4	3,477.32	3,477.32	3,482.76	3,482.76	–	3,482.60
5	3,793.96	3,841.20	3,849.36	3,849.36	3,793.38	3,849.02
6	3,211.12	3,211.12	3,223.18	3,223.18	–	–
7	3,401.53	3,441.98	3,450.50	3,450.50	3,427.26	3,452.59
8	2,881.09	2,881.09	2,881.09	2,881.07	–	2,883.04
9	4,590.49	4,590.49	4,590.49	4,590.49	4,590.80	4,592.41
10	4,277.79	4,277.79	4,304.77	4,353.92	4,347.62	4,347.62

Table 4.5. Objective value comparison on 25-node networks, when 40 samples are used for linearization, locations are fixed, and the CPU is limited to 1 hour.

Table 4.5 tells a similar story about the 25-node networks. On almost half of the instances, the 30-sample starting point outperforms the 50-sample approximation, and on the other half of instances it falls, on average, within 0.3% of the optimum, and at a much smaller computational cost (1 hour for the 40-sample starting point as opposed to 24 hours for the 50 samples). When the 40 and 50-samples algorithm finds an initial solution in 1 hour, such a solution is equally good, or even better than the 40 samples boosted by the 30 samples locations.

These results demonstrate that, 'good' locations can are found in the initial stages of the algorithm. From an execution time point of view, it is advantageous to stop the algorithm early on, retrieve the locations, then solve for optimal service levels and prices, using a limited number of samples, for a small running time.

4.4.3. Comparison with general-purpose solvers

Finally, we compare our linear approximation algorithm with a general purpose solvers for mixed-integer nonlinear optimization problems, such as BARON. We have measured the objective values yielded by BARON, and we compare them with the results of our reformulation technique run for 1000s.

Next, we attempted to improve the solutions found by our algorithm, using IPOPT, an open-source software for large-scale nonlinear optimization based on a primal-dual interior point algorithm [**?**]. For this experiment, we fixed the locations given by a 30-sample approximation within 1 hour, yielding a fully-continuous restricted problem. We have warm-started IPOPT with the respective 30-sample price, service levels, and user flows. The results are shown in Table 4.6.

All BARON and IPOPT tests were run for 1,000 seconds on the NEOS server, on computers equipped with 64 GB of RAM, and processors running at a frequency between 2.2 and 2.8 GHz [1].

Our reformulation technique clearly outperforms BARON on all instances. IPOPT is capable of improving the initial solution only in three instances while, on the others, the solution worsens significantly. On one instance, marked with * in the table, the objective value is negative, despite being warm started with a good (positive) solution, likely indication of numerical difficulties.

4.5. Conclusions

In this paper, we addressed a highly nonlinear bilevel pricing location model involving both combinatorial and continuous elements, and proposed for its solution an algorithm based on reformulation and piecewise linear approximations.

1. A detailed description of the NEOS server computers' specifications can be found here https://neos-guide.org/content/FAQ

	30 samples (1000s)	BARON (1000s)	y from 30 samples, 1h, IPOPT (1000s)
1	3,454.28	3,330.10	3,139.12
2	4,932.44	4,444.79	3,625.29
3	9,926.58	9,385.23	6,147.16
4	4,891.93	4,323.95	3,053.51
5	5,336.68	4,446.12	5,195.17
6	4,105.17	3,901.16	3,965.02
7	4,426.14	3,789.63	* -6,419.41
8	3,093.31	2,550.13	2,852.44
9	3,215.63	2,666.95	2,374.10
10	4,053.45	2,689.02	667.08

Table 4.6. Objective value comparison with BARON and IPOPT, on 20-node networks.

Our results are encouraging, but our algorithms have some limitations. For instance, one of the remaining challenges is to design algorithms that scale well, and can be applied successfully on large networks.

Future work could integrate other realistic features, such as variable demand. On the algorithmic side, an interesting development could be a method that exploits the pseudolinearity property of the nonlinear terms present in the reformulated objective function.

Chapter 5

Conclusions

In this thesis, we addressed a number of bilevel location models involving both combinatorial and nonlinear, nonconvex elements. The resulting mathematical programs are extremely challenging. Our models are characterized by non-linear lower level, non-convex upper level, and the KKT optimality conditions of the lower-level can not be reformulated into an MILP. This explains the frequent recourse to heuristic (meta-heuristics, math-heuristics) in the literature, since the exact and quasi-exact bilevel algorithms typically rest on these conditions.

Metaheuristics could be applied in our case (Tabu search, genetic algorithms, etc.), but they are not desirable, as the solution space would increase tremendously when modelling the non-binary variables and they only guarantee local optimality.

We believe that algorithms that exploit MILP approximations deserve some consideration, and we have explored this idea in our papers. Our models are flexible and can accommodate numerous real-life applications, while the proposed algorithms remain applicable.

While our results are encouraging, our methods have their limitations. The main issue is that our algorithms are intractable for large networks, due to the highly combinatorial nature of the problems, so further research is needed in this sense.

On the modelling side, future work could integrate a number of features such as elastic demand or the possibility of either increasing or decreasing the service rates of existing facilities. When balking is present, users might decide veer (patronize a second facility) instead of leaving the system. More realistic models where the competition reacts could also be considered. Of course, the presence of these features would add more difficulty to an already challenging problem.

Bibliography

[hea, 2017] (2017). Health Fact Sheets. Statistics Canada Online Catalogue no. 82-625-X.

[Aboolian et al., 2008] Aboolian, R., Berman, O., and Krass, D. (2008). Optimizing pricing and location decisions for competitive service facilities charging uniform price. *Journal of the Operational Research Society*, 59(11):1506–1519.

[Aboolian et al., 2012] Aboolian, R., Berman, O., and Krass, D. (2012). Profit maximizing distributed service system design with congestion and elastic demand. *Transportation Science*, 46(2):247–261.

[Abouee-Mehrizi et al., 2011] Abouee-Mehrizi, H., Babri, S., Berman, O., and Shavandi, H. (2011). Optimizing capacity, pricing and location decisions on a congested network with balking. *Mathematical Methods of Operations Research*, 74(2):233–255.

[Andreani and Martínez, 2001] Andreani, R. and Martínez, J. M. (2001). On the solution of mathematical programming problems with equilibrium constraints. *Mathematical Methods of Operations Research*, 54(3):345–358.

[Averbakh et al., 2007] Averbakh, I., Berman, O., Drezner, Z., and Wesolowsky, G. O. (2007). The uncapacitated facility location problem with demand-dependent setup and service costs and customer-choice allocation. *European Journal of Operational Research*, 179(3):956–967.

[Azarmand and Neishabouri, 2009] Azarmand, Z. and Neishabouri, E. (2009). Location allocation problem. In Zanjirani Farahani, R. and Hekmatfar, M., editors, *Facility Location*, Contributions to Management Science, pages 93–109. Physica-Verlag HD.

[Baumrucker et al., 2008] Baumrucker, B., Renfro, J., and Biegler, L. (2008). Mpec problem formulations and solution strategies with chemical engineering applications. *Computers & Chemical Engineering*, 32(12):2903–2913.

[Beckmann et al., 1956] Beckmann, M., McGuire, C. B., and Winsten, C. B. (1956). *Studies in the Economics of Transportation*. Yale University Press, New Haven.

[Belotti et al., 2016] Belotti, P., Bonami, P., Fischetti, M., Lodi, A., Monaci, M., Nogales-Gómez, A., and Salvagnin, D. (2016). On handling indicator constraints in mixed integer programming. *Computational Optimization and Applications*, 65:545–566.

[Benati and Hansen, 2002] Benati, S. and Hansen, P. (2002). The maximum capture problem with random utilities: Problem formulation and algorithms. *European Journal of Operational Research*, 143(3):518–530.

[Beresnev, 2013] Beresnev, V. (2013). Branch-and-bound algorithm for a competitive facility location problem. *Computers & Operations Research*, 40(8):2062–2070.

[Berman and Drezner, 2006] Berman, O. and Drezner, Z. (2006). Location of congested capacitated facilities with distance-sensitive demand. *IIE Transactions*, 38(3):213–221.

[Berman and Krass, 2015] Berman, O. and Krass, D. (2015). *Location Science*, chapter 17: Stochastic Location Models with Congestion, pages 443–486.

[Boffey et al., 2007] Boffey, B., Galvão, R., and Espejo, L. (2007). A review of congestion models in the location of facilities with immobile servers. *European Journal of Operational Research*, 178(3):643–662.

[Brotcorne et al., 2008] Brotcorne, L., Labbé, M., Marcotte, P., and Savard, G. (2008). Joint design and pricing on a network. *Operations Research*, 56:1104–1115. Language of publication: en.

[Camacho-Vallejo et al., 2014] Camacho-Vallejo, J.-F., Cordero-Franco, A. E., and González-Ramírez, R. G. (2014). Solving the bilevel facility location problem under preferences by a stackelberg-evolutionary algorithm. *Mathematical Problems in Engineering*, 2014:14.

[Castillo et al., 2009] Castillo, I., Ingolfsson, A., and Sim, T. (2009). Socially optimal location of facilities with fixed servers, stochastic demand and congestion. *Production & Operations Management*, 18(6):721–736.

[Census, 2016] Census (2016). Dissemination Area Boundary File, 2016 Census. Statistics Canada Catalogue no. 92-169-X.

[Chen and Wan, 2003] Chen, H. and Wan, Y.-W. (2003). Price competition of make-to-order firms. *IIE Transactions*, 35(9):817–832.

[Cheung and Wang, 1995] Cheung, F. K. and Wang, X. (1995). Spatial price discrimination and location choice with non-uniform demands. *Regional Science and Urban Economics*, 25(1):59–73.

[D'Ambrosio et al., 2010] D'Ambrosio, C., Lodi, A., and Martello, S. (2010). Piecewise linear approximation of functions of two variables in MILP models. *Operations Research Letters*, 38(1):39–46.

[Dan and Marcotte, 2017] Dan, T. and Marcotte, P. (2017). Competitive facility location with selfish users and queues. Technical Report 46, CIRRELT. Accepted for publication in Operations Research.

[Dantrakul et al., 2014] Dantrakul, S., Likasiri, C., and Pongvuthithum, R. (2014). Applied p-median and p-center algorithms for facility location problems. *Expert Systems with Applications*, 41(8):3596–3604.

[Desrochers et al., 1995] Desrochers, M., Marcotte, P., and Stan, M. (1995). The congested facility location problem. *Location Science*, 3(1):9–23.

[Dobson and Stavrulaki, 2007] Dobson, G. and Stavrulaki, E. (2007). Simultaneous price, location, and capacity decisions on a line of time-sensitive customers. *Naval Research Logistics (NRL)*, 54(1):1–10.

[Drezner et al., 2015] Drezner, T., Drezner, Z., and Kalczynski, P. (2015). A leader-follower model for discrete competitive facility location. *Computers & Operations Research*, 64:51–59.

[D'Ambrosio et al., 2010] D'Ambrosio, C., Frangioni, A., Liberti, L., and Lodi, A. (2010). On interval-subgradient and no-good cuts. *Operations Research Letters*, 38(5):341 – 345.

[Eiselt and Laporte, 1997] Eiselt, H. and Laporte, G. (1997). Sequential location problems. *European Journal of Operational Research*, 96(2):217–231.

[Eiselt et al., 1993] Eiselt, H. A., Laporte, G., and Thisse, J.-F. (1993). Competitive location models: A framework and bibliography. *Transportation Science*, 27(1):44–54.

[Eiselt and Marianov, 2011] Eiselt, H. A. and Marianov, V. (2011). Pioneering developments in location analysis. In Eiselt, H. A. and Marianov, V., editors, *Foundations Of Location Analysis*, volume 155 of *International Series in Operations Research & Management Science*, chapter 1. Springer US.

[Fischetti et al., 2016] Fischetti, M., Ljubić, I., and Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557 – 569.

[Fisk, 1980] Fisk, C. (1980). Some developments in equilibrium traffic assignment methodology. *Transportation Research B*, 14(3):243–256.

[Furini and Traversi, 2013] Furini, F. and Traversi, E. (2013). Hybrid SDP Bounding Procedure. In Bonifaci, V., Demetrescu, C., and Marchetti-Spaccamela, A., editors, *Experimental Algorithms*, pages 248–259, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Gilbert et al., 2015] Gilbert, F., Marcotte, P., and Savard, G. (2015). A numerical study of the logit network pricing problem. *Transportation Science*, 49(3):706–719.

[Haase, 2009] Haase, K. (2009). Discrete location planning. Technical report, Institute of Transport and Logistics Studies, University of Sydney.

[Haase and Müller, 2014] Haase, K. and Müller, S. (2014). A comparison of linear reformulations for multinomial logit choice probabilities in facility location models. *European Journal of Operational Research*, 232(3):689–691.

[Hajipour et al., 2016] Hajipour, V., Farahani, R. Z., and Fattahi, P. (2016). Bi-objective vibration damping optimization for congested location-pricing problem. *Comput. Oper. Res.*, 70(C):87–100.

[Hakimi, 1983] Hakimi, S. (1983). On locating new facilities in a competitive environment. *European Journal of Operational Research*, 12(1):29–35.

[Hanjoul et al., 1990] Hanjoul, P., Hansen, P., Peeters, D., and Thisse, J.-F. (1990). Uncapacitated plant location under alternative spatial price policies. *Management Science*, 36(1):41–57.

[Hassin, 2016] Hassin, R. (2016). *Rational Queueing*. CRC Press.

[Hijazi and Liberti, 2014] Hijazi, H. and Liberti, L. (2014). Constraint qualification failure in second-order cone formulations of unbounded disjunctions. Technical report, NICTA, Canberra ACT Australia.

[Hotelling, 1929] Hotelling, H. (1929). Stability in competition. *The Economic Journal*, 39(153).

[Hwang and Mai, 1990] Hwang, H. and Mai, C.-C. (1990). Effects of spatial price discrimination on output, welfare, and location. *The American Economic Review*, 80(3):567–575.

[Jeroslow and Lowe, 1984] Jeroslow, R. and Lowe, J. (1984). Modelling with integer variables. In Korte, B. and Ritter, K., editors, *Mathematical Programming at Oberwolfach II, Volume 22 of Mathematical Programming Studies*, pages 167–184. Springer, Berlin.

[Julsain, 1999] Julsain, H. (1999). *Tarification dans les réseaux de télécommunications [microforme] : une approche par programmation mathématique à deux niveaux*. Canadian theses. Thèse (M.Sc.A.)–École polytechnique de Montréal.

[Kim, 2013] Kim, S. (2013). Heuristics for congested facility location problem with clearing functions. *Journal of the Operational Research Society*, 64(12):1780–1789.

[Kleinrock, 1975] Kleinrock, L. (1975). *Theory, Volume 1, Queueing Systems*. Wiley-Interscience.

[Kolodziej et al., 2013] Kolodziej, S., Castro, P. M., and Grossmann, I. E. (2013). Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization*, 57(4):1039–1063.

[Küçükaydin et al., 2011] Küçükaydin, H., Aras, N., and Altınel, I. K. (2011). Competitive facility location problem with attractiveness adjustment of the follower: A bilevel programming model and its solution. *European Journal of Operational Research*, 208(3):206–220.

[Labbé and Hakimi, 1991] Labbé, M. and Hakimi, S. L. (1991). Market and locational equilibrium for two competitors. *Operations Research*, 39(5):749–756.

[Labbé et al., 1998] Labbé, M., Marcotte, P., and Savard, G. (1998). A bilevel model of taxation and its application to optimal highway pricing. *Manage. Sci.*, 44(12):1608–1622.

[Lee and Cohen, 1983] Lee, H. L. and Cohen, M. A. (1983). A note on the convexity of performance measures of m/m/c queueing systems. *Journal of Applied Probability*, 20(4):920–923.

[Ljubić and Moreno, 2018] Ljubić, I. and Moreno, E. (2018). Outer approximation and submodular cuts for maximum capture facility location problems with random utilities. *European Journal of Operational Research*, 266(1):46–56.

[Marcotte, 1986] Marcotte, P. (1986). Network design problem with congestion effects: A case of bilevel programming. *Mathematical Programming*, 34(2):142–162.

[Marcotte et al., 2013] Marcotte, P., Savard, G., and Schoeb, A. (2013). A hybrid approach to the solution of a pricing model with continuous demand segmentation. *EURO Journal on Computational Optimization*, 1(1):117–142.

[Marianov, 2003] Marianov, V. (2003). Location of multiple-server congestible facilities for maximizing expected demand, when services are non-essential. *Annals of Operations Research*, 123(1-4):125–141.

[Marianov et al., 2008] Marianov, V., Ríos, M., and Icaza, M. J. (2008). Facility location for market capture when users rank facilities by shorter travel and waiting times. *European Journal of Operational Research*, 191(1):32–44.

[Marianov and Serra, 2001] Marianov, V. and Serra, D. (2001). Hierarchical location-allocation models for congested systems. *European Journal of Operational Research*, 135(1):195–208.

[Marianov and Serra, 2011] Marianov, V. and Serra, D. (2011). Location of multiple-server common service centers or facilities, for minimizing general congestion and travel cost functions. *International Regional Science Review*, 34(3):323–338.

[Marić et al., 2012] Marić, M., Stanimirović, Z., and Milenković, N. (2012). Metaheuristic methods for solving the bilevel uncapacitated facility location problem with clients' preferences. *Electronic Notes in Discrete Mathematics*, 39(0):43 – 50. EURO Mini Conference.

[McFadden, 1974] McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. *FRONTIERS IN ECONOMETRICS*, pages pp. 105–142.

[Meng et al., 2012] Meng, Q., Liu, Z., and Wang, S. (2012). Optimal distance tolls under congestion pricing and continuously distributed value of time. *Transportation Research Part E: Logistics and Transportation Review*, 48(5):937–957. Selected papers from the 14th ATRS and the 12th WCTR Conferences, 2010.

[Pahlavani and Saidi-Mehrabad, 2011] Pahlavani, A. and Saidi-Mehrabad, M. (2011). Optimal pricing for competitive service facilities with balking and veering customers. 7:3171–3191.

[Panin et al., 2014] Panin, A. A., Pashchenko, M., and Plyasunov, A. V. (2014). Bilevel competitive facility location and pricing problems. *Automation and Remote Control*, 75(4):715–727.

[Pérez et al., 2004] Pérez, M. D. G., Hernández, P. F., and Pelegrín, B. P. (2004). On price competition in location-price models with spatially separated markets. *Top*, 12(2):351–374.

[Plastria, 2001] Plastria, F. (2001). Static competitive facility location: An overview of optimisation approaches. *European Journal of Operational Research*, 129(3):461–470.

[Raghunathan, 2013] Raghunathan, A. (2013). Global optimization of nonlinear network design. *SIAM Journal on Optimization*, 23:268–295.

[Rahmati et al., 2014] Rahmati, S. H. A., Ahmadi, A., Sharifi, M., and Chambari, A. (2014). A multi-objective model for facility location-allocation problem with immobile servers within queuing framework. *Computers & Industrial Engineering*, 74(0):1–10.

[Saidani et al., 2012] Saidani, N., Chu, F., and Chen, H. (2012). Competitive facility location and design with reactions of competitors already in the market. *European Journal of Operational Research*, 219(1):9 – 17.

[Sattinger, 2002] Sattinger, M. (2002). A queuing model of the market for access to trading partners. *International Economic Review*, 43(2):533–547.

[Sun et al., 2008] Sun, H., Gao, Z., and Wu, J. (2008). A bi-level programming model and solution algorithm for the location of logistics distribution centers. *Applied Mathematical Modelling*, 32(4):610 – 616.

[Tavakkoli-Moghaddam et al., 2017] Tavakkoli-Moghaddam, R., Vazifeh-Noshafagh, S., Taleizadeh, A. A., Hajipour, V., and Mahmoudi, A. (2017). Pricing and location decisions in multi-objective facility location problem with m/m/m/k queuing systems. *Engineering Optimization*, 49(1):136–160.

[Teles et al., 2011] Teles, J. P., Castro, P. M., and Matos, H. A. (2011). Multi-parametric disaggregation technique for global optimization of polynomial programming problems. *Journal of Global Optimization*, 55(2):227–251.

[Tong, 2011] Tong, D. (2011). *Optimal Pricing and Capacity Planning in Operations Management*. PhD thesis.

[Vidyarthi and Jayaswal, 2014] Vidyarthi, N. and Jayaswal, S. (2014). Efficient solution of a class of location-allocation problems with stochastic demand and congestion. *Computers & Operations Research*, 48(0):20–30.

[Zhang et al., 2010a] Zhang, Y., Berman, O., Marcotte, P., and Verter, V. (2010a). A bilevel model for preventive healthcare facility network design with congestion. *IIE Transactions*, 42(12):865–880.

[Zhang et al., 2010b] Zhang, Y., Berman, O., Marcotte, P., and Verter, V. (2010b). A bilevel model for preventive healthcare facility network design with congestion. *IIE Transaction*, 42(12):865–880.

[Zhang et al., 2012] Zhang, Y., Berman, O., and Verter, V. (2012). The impact of client choice on preventive healthcare facility network design. *OR Spectrum*, 34(2):349–370.

Chapter A

A power-based linearization technique

In this section we detail a linearization technique for program (P) in Chapter 4, based on the power-based transformation originally proposed in [**Teles et al., 2011**]. The main idea is to transform nonlinear polynomial problems into an MILP , by applying a term-wise disaggregation scheme, notwithstanding, with additional discrete and continuous variables. Kolodziej et. al incorporate this technique into a global optimization algorithm for bilinear programs ([**Kolodziej et al., 2013**]). The authors argue that this technique scales better than the piecewise McCormick envelopes, and is comparable with global optimization solvers.

On our problem, this mixed-integer linearization technique became quickly intractable even for smaller, 10-node networks, perhaps as a consequence of the large number of additional binary variables. However, we describe it in detail for scientific purposes.

Linearization of w_j for the leader.

We sample the interval $(0, \mu_{\max}]$ using P points $\mu^p, p \in \{1, \ldots, P\}$. Then, for each sample μ^p we generate m samples of λ over $(0, \mu^p)$. Let $\lambda^{p,m}$ and μ^p be the samples hence obtained. We linearize $w_j(\lambda, \mu)$ using tangent plane at points $(\lambda^{p,m}, \mu^p)$ for $p \in \{1, \ldots, P\}$, $m \in \{1, \ldots, M\}$ such that $\lambda^{p,m} \leq \mu^p$. Based on the gradient

$$\nabla w_j(\lambda, \mu) = \left(1/(\mu - \lambda)^2, -1/(\mu - \lambda)^2\right), \tag{A.0.1}$$

we write the first-order Taylor approximations of $w_j(\lambda, \mu)$:

$$f^{p,m}(\lambda, \mu) = w(\lambda^{p,m}, \mu^p) + \nabla w_j(\lambda^{p,m}, \mu^p) \begin{pmatrix} \lambda - \lambda^{p,m} \\ \mu - \mu^p \end{pmatrix}$$

$$= \underbrace{\frac{1}{(\mu^p - \lambda^{p,m})^2}}_{a^{p,m}} \cdot \lambda + \underbrace{\frac{-1}{(\mu^p - \lambda^{p,m})^2}}_{b^{p,m}} \cdot \mu + \underbrace{\frac{2}{(\mu^p - \lambda^{p,m})^2}}_{c^{p,m}} .$$

Next, we write a piecewise linear approximation of w_j by setting it to the maximum of its linear approximations:

$$w_j(\lambda_j, \mu_j) \approx \max_{\substack{p\in\{1,\dots,P\} \\ m\in\{1,\dots,M\}}} \{f^{p,m}(\lambda_j, \mu_j)\} \tag{A.0.2}$$

Eq. (A.0.2) can be replaced with

$$w_j(\lambda_j, \mu_j) \geq \max_{\substack{p=1,\dots,P, \\ m=1,\dots,M}} \{f^{p,m}(\lambda_j, \mu_j)\} \tag{A.0.3}$$

Intuitively, since the leader is in control of variables λ_j (provided they satisfy the equilibrium constraints), for a given value of μ_j and λ_j, it is in her best interest to set w_j as small as possible. From another angle, for a given value of w_j, the leader will fix λ as high as possible, in order to maximize her profit. This is not the case when the leader can attract the entire demand, even for values of w_j higher than the under approximation.

Linearization of $\mathbf{w_j}$ for the competitor. For the competitor we perform a piecewise linear approximation via binary variables. For each $j \in J_c$ we sample the interval $(0, \mu_j)$ using R points. Let $\lambda^{j,r}$ be the aforementioned samples. We linearize $w_j(\lambda, \mu_j)$ using line equations between consecutive pair of samples, and we have:

$$\sum_{r=1}^{R} \hat{s}_{jr} = 1 \qquad j \in J_c \tag{A.0.4}$$

$$\lambda_j = \sum_{r=1}^{R} \hat{s}_{jr} \hat{\lambda}^{jr} \qquad j \in J_c \tag{A.0.5}$$

$$\hat{w}_j \leq \sum_{r=1}^{R} \hat{s}_{jr} w(\hat{\lambda}^{jr}, \mu_j) \qquad j \in J_c \tag{A.0.6}$$

$$\sum_{r=1}^{R} \hat{l}_{jr} = 1 \qquad j \in J_c \tag{A.0.7}$$

$$\hat{s}_{jr} \leq \hat{l}_{jr} + \hat{l}_{jr-1} \qquad j \in J_c;\ r \in \{1, \dots, R\} \tag{A.0.8}$$

$$\hat{l}_{jr} \in \{0, 1\} \qquad j \in R;\ r \in \{1, \dots, R\} \tag{A.0.9}$$

$$0 \leq \hat{s}_{jr} \leq 1 \qquad j \in J_c;\ r \in \{1, \dots, R\} \tag{A.0.10}$$

$$\hat{l}_{j,0} = 0, \hat{l}_{j,R} = 0 \qquad\qquad j \in J_c \ . \tag{A.0.11}$$

Linearization of bilinear terms $\lambda_j \cdot p_j$. We linearize $\lambda_j p_j$ using the parametric disaggregated method where we represent p_j to a certain level of precision π (e.g. $\pi = -3$ then p_j is precise to three decimal places). Let p_{MAX} be the maximum value the price can have (it can be easily computed), and let $\lambda_{MAX} = \sum_{i \in I} d_i$, i.e. the total demand rate in the network. We define $\Pi = \log_{10}(p_{MAX})$ and the set of equations characterizing terms $u_j = \lambda_j p_j$ is:

$$u_j = \sum_{l=\pi}^{\Pi} \sum_{k=0}^{9} 10^l \cdot k \cdot \hat{\lambda}_{j,k,l} \qquad j \in J_1 \tag{A.0.12}$$

$$p_j = \sum_{l=\pi}^{\Pi} \sum_{k=0}^{9} 10^l \cdot k \cdot z_{j,k,l} \qquad j \in J_1 \tag{A.0.13}$$

$$\hat{\lambda}_{j,k,l} \leq \lambda_{MAX} \cdot z_{j,k,l} \qquad j \in J_1; k \in \{0, \ldots, 9\}; l \in \{\pi, \ldots \Pi\} \tag{A.0.14}$$

$$\hat{\lambda}_{j,k,l} \geq 0 \qquad j \in J_1; k \in \{0, \ldots, 9\}; l \in \{\pi, \ldots \Pi\} \tag{A.0.15}$$

$$\lambda_j = \sum_{k=0}^{9} \hat{\lambda}_{j,k,l} \qquad j \in J_1; l \in \{\pi, \ldots \Pi\} \tag{A.0.16}$$

$$\sum_{k=0}^{9} z_{j,k,l} = 1 \qquad j \in J_1; l \in \{\pi, \ldots \Pi\} \tag{A.0.17}$$

$$z_{j,k,l} \in \{0, 1\} \qquad j \in J_1; k \in \{0, \ldots, 9\}; l \in \{\pi, \ldots \Pi\} \ . \tag{A.0.18}$$

Linearization of complementarity constraints. We linearize Eq. (**??**) through the introduction of binary variables and big-M constants, as follows:

$$t_{ij} + \alpha w_j + \beta p_j - \gamma_i \leq M s_{ij} \qquad i \in I; \ j \in J \tag{A.0.19}$$

$$x_{ij} \leq M(1 - s_{ij}) \qquad i \in I; \ j \in J \tag{A.0.20}$$

$$s_{ij} \in \{0, 1\} \qquad i \in I; \ j \in J \ . \tag{A.0.21}$$

The value of M must be sufficiently large, but not too large that it slows down the algorithm. We can compute a tight value for the M constant, based on the maximum waiting and travel time in the network.

Finally, we write (P) as the following MILP:

PL: (Leader:)

$$\max_{y,\mu,x,p,\gamma} \quad z = \sum_{j \in J_1} u_j - \sum_{j \in J_1} (f_c \cdot y_j + v_c \cdot \mu_j)$$

s.t.

$$\mu_j \leq M_1 \cdot y_j \qquad j \in J_1$$

$$y_j \in \{0,1\} \qquad j \in J_1$$

$$\mu_j \geq 0 \qquad j \in J_1$$

$$t_{ij} + \alpha w_j + \beta p_j - \gamma_i \geq 0 \qquad i \in I;\ j \in J$$

$$\lambda_j = \sum_{i \in I} x_{ij} \qquad j \in J$$

$$\sum_{j \in J} x_{ij} = d_i \qquad i \in I$$

$$\lambda_j \leq \mu_j \qquad j \in J$$

$$x_{ij} \geq 0 \qquad i \in I; j \in J$$

constraints (A.0.3) – (A.0.21) .