Static Pricing Problems under Mixed Multinomial Logit Demand

DOI:
10.48550/arXiv.2005.07482

Document status and date:
Published: 18/05/2020

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Static Pricing Problems under Mixed Multinomial Logit Demand

Ahmadreza Marandi
Eindhoven University of Technology, Department of Industrial Engineering & Innovation Sciences, The Netherlands, a.marandi@tue.nl.

Virginie Lurkin
Eindhoven University of Technology, Department of Industrial Engineering & Innovation Sciences, The Netherlands, v.j.c.lurkin@tue.nl.

Price differentiation is a common strategy for many transport operators. In this paper, we study a static multiproduct price optimization problem with demand given by a continuous mixed multinomial logit model. To solve this new problem, we design an efficient iterative optimization algorithm that asymptotically converges to the optimal solution. To this end, a *linear optimization* (LO) problem is formulated, based on the trust-region approach, to find a "good" feasible solution and approximate the problem from below. Another LO problem is designed using piecewise linear relaxations to approximate the optimization problem from above. Then, we develop a new branching method to tighten the optimality gap. Numerical experiments show the effectiveness of our method on a published, non-trivial, parking choice model.

Key words: static multi-product pricing, mixed logit model, nonlinear optimization

1. Introduction

Offering different products at different prices to different travelers is a common practice in many transportation markets. Classic examples include business-, first-, and economy-class flight tickets as well as first- and second-class railway tickets. With product and price differentiation, transport operators are able to get higher revenues by adapting their fares based on the price sensitivity of their travelers. Basically higher fares are offered to the ones who are willing to pay more.

Inferring travelers willingness to pay (WTP) is a long-standing practice in applied economics (Hensher et al. 2005). Discrete-choice modeling (DCM) has established itself as an important and widely-used methodology for extracting valuations such as willingness to pay (Hess et al. 2018). Transport researchers have used these disaggregate demand models for more than 40 years, from the pioneer work of McFadden and Zarembka (1974) to more recent studies on WTP for self driving vehicles (Daziano et al. 2017) or willingness to travel with green modes in the context of shared mobility (Li and Kamargianni 2019).

Formulating pricing policies based on such disaggregate demand representations allows to better account for the heterogeneity of the population of interest, where different customers have different tastes and preferences. Even more importantly, it better reflects the supply-demand interactions by capturing the tradeoff between the operator objective of maximizing the expected revenue and the customer objective of maximizing the expected utility (Sumida et al. 2019).

Despite a more comprehensive representation, including discrete choice models within pricing problems increases the computational complexity because the choice probabilities are nonlinear. As a result the expected revenue is highly nonlinear in the prices of the products and customary used nonlinear algorithms may get terminated at a local optimum.

Due to the importance of the problem, the Operations Research and Management Science communities put remarkable efforts on analyzing it. Hanson and Martin (1996) pioneer this research by showing that the expected revenue function is not concave in prices, even for the simple *multinomial logit* (MNL) model. Subsequent authors have demonstrated that, under uniform price sensitivities across all products, the expected revenue function is concave in the choice probability vector (Song and Xue 2007, Dong et al. 2009, Zhang and Lu 2013). Li and Huh (2011) show that this concavity result also holds under asymmetric price-sensitivities, not only for the MNL model, but also for the *nested logit* (NL) model that generalizes the MNL model by grouping product alternatives into different nests based on their degree of substitution (McFadden 1977).

Parallel to these work, several authors have shown that under restrictive conditions on the degree of asymmetry in the price sensitivity parameters, unique price solutions exist for some logit models. This has been shown for the MNL model (e.g., Aydin and Ryan (2000), Hopp and Xu (2005), Maddah and Bish (2007), Aydin and Porteus (2008), Akçay et al. (2010)), the NL model (e.g., Aydin and Ryan (2000), Hopp and Xu (2005), Maddah and Bish (2007), Aydin and Porteus (2008), Akçay et al. (2010), Gallego and Wang (2014), Huh and Li (2015)), the *paired combinatorial logit* (PCL) model (Li and Webster 2017) and lately generalized to any *generalized extreme value* (GEV) model (Zhang et al. 2018). In this stream of research, first-order condition is generally used to find optimal prices. It is worthy to note that in some of these studies and additional recent ones, pricing decisions are optimized jointly with other decisions such as assortment or scheduling decisions ((e.g., Du et al. (2016), Jalali et al. (2019), Bertsimas et al. (2020)).

To accommodate heterogeneity across individuals in their sensitivities to price, Li et al. (2019) consider a pricing problem under a *discrete mixed logit model.* As explained by the authors, the expected revenue function under the mixed logit model is not well-behaved and the concavity property with respect to the choice probabilities breaks down, even for entirely symmetric price sensitivities across products and segments. Accordingly, the theoretical results as well as the solution methods developed for other logit models do not apply to the pricing problem with demand characterized by a discrete mixed logit model. So, the authors propose two concave maximization problems that work as lower and upper bounds for the objective value of the revenue function, under some conditions. Then, they propose an algorithm that converges to a local optimum.

In this paper, we consider a more general problem, namely optimal pricing under a *continuous mixed logit* model. Our study therefore fits in the established literature on static price optimization under the family of logit choices. As showed by McFadden and Train (2000), under mild regularity conditions, the mixed logit model can approximate choice probabilities of any discrete choice model derived from random utility maximization (RUM) assumption.

We design an efficient iterative optimization algorithm that asymptotically converges to the optimal solution. To this end, a *linear optimization* (LO) problem is formulated, based on the trust-region approach, to find a "good" feasible solution and approximate the problem from below. Another LO problem is designed using piecewise linear approximations as well as the McCormick relaxation (McCormick 1976) to approximate the optimization problem from above. Then, we develop a new branching method to tighten the optimality gap and show that the algorithm converges to the optimal solution asymptotically. The effectiveness of this algorithm is demonstrated on a parking services pricing case for which the demand model comes from a published, non-trivial, parking choice model.

Therefore, our work extends the results of the literature in three ways. First, our pricing problem includes the continuous setting of the mixed logit model, which is more general than its discrete counterpart, and better reflects the many transport applications in which mixed logit models have been used (Train 2003). Second, our algorithm can deal with any linear dependencies in the prices of service, while the results in the literature can only deal with lower and upper bounds on the prices (Hanson and Martin 1996, Dong et al. 2009, Li et al. 2019). Third, we show that the algorithm converges to a global optimum without posing any assumptions, while in the literature either local optimality is mainly considered or restrictive conditions are posed to have global optimality (Hanson and Martin 1996, Dong et al. 2009, Li et al. 2019).

The remaining sections are organized as follows. Section 2 further defines the problem under consideration. Section 3 presents our global algorithm, while Section 4 shows the results of our numerical experiments. The final section concludes our paper.

2. Problem description

In this paper, we are interested in solving a static multi-product pricing problem under a continuous mixed logit model. Static pricing involves the simultaneous pricing of multiple products, where a fixed price is set for each product (Soon 2011). In our setting, we assume that a single seller must decide at what price to offer each product from a finite set of alternatives (also known as product assortment). On the demand side, we assume that customers choose among the products according to a consumer choice model. The demand for each product is thus the result of the individual purchase choice of N customers. The purchase choice is captured by a discrete choice model, that predicts the customer choice from a finite set of discrete alternatives (Ben-Akiva and Bierlaire 2003).

Let $\mathcal{N}$ represent the set of N customers and let $\mathcal{I}$ indicate the set of I products available for purchase, including the ones offered by the seller. Utility functions U_{in} are defined for each customer $n \in \mathcal{N}$ and product $i \in \mathcal{I}$. Each utility function takes into account the socio-economic characteristics and the tastes of the individual as well as the attributes of the alternative. According to Random Utility Maximization (RUM) theory (Manski 1977), U_{in} can be decomposed into a systematic component $V_{in}(\beta)$, which includes all observations of the decision maker (including the offered price p_i as well as the price parameter, also called willingness to pay, β^p_{in}), and a random term ε_{in}, which captures the uncertainties caused by unobserved attributes and unobserved taste variations:

$$U_{in} = V_{in}(\beta) + \varepsilon_{in} \tag{1}$$

$$= \beta^p_{in} p_i + q_{in}(\beta^q) + \varepsilon_{in}, \tag{2}$$

where p_i is the endogenous price variable and $q_{in}(\beta^q)$ is the exogenous part of the utility, obtained by adding all observed product attributes other than price, weighted based on customers' preferences.

The resulting discrete choice model is therefore naturally probabilistic. The probability that customer n chooses alternative i is defined as

$$P_{in} = \Pr\left[V_{in}(\beta) + \varepsilon_{in} = \max_{j\in\mathcal{I}}\{V_{jn}(\beta) + \varepsilon_{jn}\}\right].$$

The optimal expected revenues obtained from the sales of offering the products is then naturally given by:

$$\begin{aligned}
\max_{p\in\mathbb{R}^I} \quad & \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}} p_i P_{in}, \\
\text{s.t.} \quad & P_{in} = \Pr\left[V_{in}(\beta) + \varepsilon_{in} \geq V_{jn}(\beta) + \varepsilon_{jn}, \ \ \forall j \in \mathcal{I}\right] \quad , \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \\
& V_{in}(\beta) = \beta^p_{in} p_i + q_{in}(\beta^q), \qquad \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \\
& 0 \leq p_i \leq \bar{p}_i, \qquad \forall i \in \mathcal{I},
\end{aligned} \tag{3}$$

where $\bar{p} \in \mathbb{R}^I$ is a vector containing upper bounds on the prices of products. The most commonly used discrete choice models, the multinomial logit (MNL) model, is built upon the assumption of independent and identically extreme value distributed error terms (Manski 1977), that is $\varepsilon_{in} \overset{\text{i.i.d.}}{\sim} EV(0,1)$. Under this assumption, the probability for customer n to select choice alternative i is given by

$$P_{in} = \frac{e^{V_{in}(\beta)}}{\sum_{j\in\mathcal{I}} e^{V_{jn}(\beta)}}. \tag{4}$$

Mixed logit probabilities are the integrals of these standard logit probabilities over a density of parameters (Train 2003). The choice probabilities can then be expressed in the form:

$$P_{in} = \int \frac{e^{V_{in}(\beta)}}{\sum_{j\in\mathcal{I}} e^{V_{jn}(\beta)}} d\nu_\beta, \tag{5}$$

where ν_β is a multivariate probability measure.

The mixed logit model is often considered to be the most popular discrete choice model for incorporating random taste heterogeneity (Vij and Krueger 2017). As a result, it has been extensively used in transport studies (e.g., Ye et al. (2020), Han et al. (2020)). In this paper, we are interested in incorporating this popular choice model into static pricing problems. This involves solving the following nonlinear maximization problem:

$$\begin{aligned}\max_{p\in\mathbb{R}^I} \ & \sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}} p_i P_{in}, \\ \text{s.t. } & P_{in} = \int \frac{e^{V_{in}(\beta)}}{\sum_{j\in\mathcal{I}} e^{V_{jn}(\beta)}} d\nu_\beta, \quad \forall i\in\mathcal{I}, \forall n\in\mathcal{N}, \\ & V_{in}(\beta) = \beta^p_{in} p_i + q_{in}(\beta^q), \quad \forall i\in\mathcal{I}, \forall n\in\mathcal{N}, \\ & 0 \le p_i \le \bar{p}_i, \quad \forall i\in\mathcal{I}.\end{aligned} \tag{6}$$

As easily seen in Equation (5), the mixed logit probability is a weighted average of logit probabilities evaluated at different values of β. The standard MNL model is therefore simply a special case of the mixed logit where the mixing probability measure ν_β is degenerate at a fixed parameter $\bar{\beta}$ (Train 2003), i.e., $\nu_\beta\left(\{\bar{\beta}\}\right) = 1$.

In Li et al. (2019), which is the closest study to our work, the probability measure ν_β is assumed to be discrete. In other words, they assume that β^p can take only M distinct values b^p_m, b^p_2, ..., b^p_M, resulting in the following logit choice probability:

$$P_i = \sum_{m=1}^{M} w_m \frac{e^{V_i(b^p_m)}}{\sum_{j\in\mathcal{I}} e^{V_j(b^p_m)}}, \tag{7}$$

where w_m is the probability that $\beta^p = b^p_m$.

Since no individual specific variables are included in the utility specifications, the choice probability is the same for all individuals (i.e., $P_{in} = P_i$, $\forall n \in N$) and the objective function becomes $\max_{p\in\mathbb{R}^I} \sum_{i\in\mathcal{I}} p_i P_i$.

Their pricing problem is therefore also a specific case of the pricing problem under continuous mixed logit demand. In many transport applications in which mixed logit models have been used, ν_β is specified to be a continuous measure (Train 2003) and individual specific variables are included in the utility specifications. Therefore, we pose no assumption on the probability measure ν_β and provide a method to solve (6), which has never been considered before, to the best of our knowledge.

3. Methodology

In this section, we introduce a new efficient optimization algorithm for solving the static multiproduct pricing problem under continuous mixed logit model. The proposed algorithm is a global optimizer, meaning that it asymptotically converges to the optimal solution. This is done by designing a method to find a "good" feasible solution, which provide a lower bound, as well as a method to check the quality of the obtained solution, which provides an upper bound.

Let us reformulate the optimization problem (6) as

$$\begin{aligned} opt = \max_{p\in\mathbb{R}^I} \ & f(p) \\ \text{s.t.} \ & Ap \geq b, \ \ p \geq 0, \end{aligned} \tag{8}$$

where $f(p) = \sum_{i\in\mathcal{I}} \frac{p_i}{f_i(p)}$, $f_i : \mathbb{R}^I \to \mathbb{R}$ is a positive convex function with continuous second derivative, and the feasible region is a polytope. More specifically, (6) can be formulated as (8) by setting

$$f_i(p) := \frac{1}{\sum_{n\in\mathcal{N}} \int \frac{1}{\sum_{j\in\mathcal{I}} e^{\beta^p_{jn} p_j + q_{jn}(\beta^q) - \beta^p_{in} p_i - q_{in}(\beta^q)}} d\nu_\beta}.$$

As $\frac{1}{x}$ is a convex function on $\{x : \ x > 0\}$ and integral preserves convexity (see Page 79 of (Boyd and Vandenberghe 2004)), $f_i(p)$ is a positive convex function with continuous second derivative.

3.1. Designing a method to construct lower bounds

To construct the lower bounds, we use a trust-region method (Conn et al. 2000), where solutions are obtained iteratively in the neighborhood of the previous feasible solution. A typical way of finding a better solution is by approximating the objective function with a quadratic function and solving the following optimization problem in the k^{th} iteration:

$$\begin{aligned} \max_{p \in \mathbb{R}^I} \ & \frac{1}{2} p^T H_k p + g_k^T p \\ \text{s.t.} \ & \|p - p^k\|_2 \leq r_k \\ & Ap \geq b, \ \ p \geq 0, \end{aligned} \tag{9}$$

where $H_k \in \mathbb{R}^{I \times I}$ is the Hessian matrix and $g_k \in \mathbb{R}^I$ is the gradient vector of the objective function at the feasible solution p^k obtained in the $(k-1)^{\text{st}}$ iteration, r_k is the radius of the neighborhood, and where $\|.\|_2$ is the Euclidean norm. The issue is that the objective function of (8) is neither convex nor concave, and hence (9) might be a noncocave quadratic optimization problem, known to belong to the class of NP-hard problems (Pardalos and Vavasis 1991). To avoid this issue, we use the linear approximation of the objective function in each iteration and use the following optimization problem:

$$\begin{aligned} \max \ & g_k^T p \\ \text{s.t.} \ & \|p - p^k\|_1 \leq r_k \\ & Ap \geq b, \ \ p \geq 0, \end{aligned} \tag{10}$$

where $\|.\|_1$ is the ℓ_1-norm.

Algorithm 1 provides the steps taken to find a “good” feasible solution using (10). As one can see, (10) is a linear optimization problem and hence optimal solutions are in the boundary points of its feasible region. So, the algorithm starts with searching for a good solution in the boundary of the neighborhood of the initial solution with radius 1. It continues the search unless it does not reach to a point with improvement in the objective function. Then, the radius of the neighborhood gets halved with the hope of finding a better solution. The algorithm gets terminated when the improvement in the last two iterations are less than a given tolerance error θ, hence a local optimum.

Algorithm 1 Steps to obtain a "good" feasible solution using (10)

select a random feasible solution p^0

$f^1 := +\infty$, $r^0 := 1$, $k = 0$,

while $|f^1 - f(p^k)| > \theta$, for a given $error$, **do**

    find p^{k+1} by solving (10) with radius r^0

    $\bar{p}^0 \leftarrow p^k$, $\bar{p}^1 \leftarrow p^{k+1}$, $f^0 \leftarrow f(\bar{p}^0)$, $f^1 \leftarrow f(\bar{p}^1)$

    while $f^1 > f^0$ **do**

        $\bar{p}^0 \leftarrow \bar{p}^1$, $f^0 \leftarrow f^1$

        find $\bar{p}^1$ by solving (10) with initial point $\bar{p}^0$ and radius r^0

        $f^1 \leftarrow f(\bar{p}^1)$, $r^0 \leftarrow 1$

    $r^0 \leftarrow \frac{r^0}{2}$

    $p^{k+1} \leftarrow \bar{p}^1$, increase k by 1

return p^k

3.2. Designing a method to construct upper bounds

In this section, we explore the properties of the optimization problem (8) and use them to develop an overestimator to construct an upper bound on the objective value of the problem. To this end, we first reformulate (8) as a biconvex optimization problem:

$$\begin{aligned} \max_{p,\tau\in\mathbb{R}^I} \quad & \sum_{i\in\mathcal{I}} p_i\tau_i \\ \text{s.t.} \quad & f_i(p)\tau_i \le 1, \quad \forall i\in\mathcal{I}, \\ & Ap \ge b, \\ & \tau_i, p_i \ge 0, \qquad \forall i\in\mathcal{I}. \end{aligned} \tag{11}$$

It is clear that (8) and (11) are equivalent as $f_i(p)$ is a positive function. Problem (11) belongs to the class of biconvex optimization problems, as it contains functions that are convex in p and convex in τ. There are different methods to solve biconvex optimization problems (see the review paper by Gorski et al. (2007)). In this section, we use McCormick relaxation (McCormick 1976) as well as piece-wise linear underestimators of $f_i(p)$ to construct a linear optimization problem that approximate the objective function of (8) from above.

To do so, let us assume that we have a collection of K feasible points $\mathcal{P} = \{p^0, p^1, ..., p^K\}$. As $f_i(p)$ is a convex function for any $i\in\mathcal{I}$, we have (Bazaraa et al. 2013)

$$f_i(p) \ge f_i(p^k) + \nabla f_i(p^k)^T(p - p^k), \;\; \forall i\in\mathcal{I}, \;\; k = 1, ..., K.$$

Therefore, as τ_i, $i\in\mathcal{I}$, are nonnegative, the following bilinear optimization problem provides an upper bound on the objective value of (8):

$$\begin{aligned} \max_{p,\tau\in\mathbb{R}^I} \quad & \sum_{i\in\mathcal{I}} p_i\tau_i \\ \text{s.t.} \quad & \left(f_i(p^k) + \nabla f_i(p^k)^T(p - p^k)\right)\tau_i \le 1, \quad \forall i\in\mathcal{I}, \;\; k = 1, ..., K, \\ & Ap \ge b, \\ & \tau \ge 0. \end{aligned} \tag{12}$$

To have a tractable approximation, we further approximate (12) by the following linear optimization problem:

$$\max_{\substack{p,\tau\in\mathbb{R}^I\\ W\in\mathbb{R}^{I\times I}}} \quad \sum_{i\in\mathcal{I}} W_{ii} \tag{13a}$$

$$\text{s.t.} \quad Ap \geq b, \tag{13b}$$

$$f_i(p^k)\tau_i + \nabla f_i(p^k)^T (W_{i:} - p^k\tau_i) \leq 1, \qquad \forall i\in\mathcal{I}, k=1,\ldots,K, \tag{13c}$$

$$AW_{i:} \geq b\tau_i, \qquad \forall i\in\mathcal{I}, \tag{13d}$$

$$LB_{\tau_i}(Ap-b) \leq AW_{i:} - b\tau_i, \qquad \forall i\in\mathcal{I}, \tag{13e}$$

$$AW_{i:} - b\tau_i \leq UB_{\tau_i}(Ap-b), \qquad \forall i\in\mathcal{I}, \tag{13f}$$

$$W_{ij} \geq LB_{\tau_i}p_j + \tau_i LB_{p_j} - LB_{\tau_i}LB_{p_j}, \qquad \forall i,j\in\mathcal{I}, \tag{13g}$$

$$W_{ij} \geq UB_{\tau_i}p_j + \tau_i UB_{p_j} - UB_{\tau_i}UB_{p_j}, \qquad \forall i,j\in\mathcal{I}, \tag{13h}$$

$$W_{ij} \leq UB_{\tau_i}p_j + \tau_i LB_{p_j} - UB_{\tau_i}LB_{p_j}, \qquad \forall i,j\in\mathcal{I}, \tag{13i}$$

$$W_{ij} \geq LB_{\tau_i}p_j + \tau_i UB_{p_j} - LB_{\tau_i}UB_{p_j}, \qquad \forall i,j\in\mathcal{I}, \tag{13j}$$

$$LB_\tau \leq \tau \leq UB_\tau, \tag{13k}$$

$$LB_p \leq p \leq UB_p. \tag{13l}$$

where $LB_p, UB_p \in \mathbb{R}^I$ are the vectors containing component-wise lower and upper bounds of p, $LB_\tau, UB_\tau \in \mathbb{R}^I$ are vectors containing the component-wise lower and upper bounds of τ, respectively, and $W_{i:} = [W_{ij}]_{j\in\mathcal{I}}$. Problem (13) is constructed by linearization of the bilinear optimization problem equivalent to (12) including some redundant constraints. The variable W_{ij} is added to linearize the bilinear term $\tau_i p_j$, for $i,j\in\mathcal{I}$. Constraint (13c) is a linearization of the first constraint in (12). Constraint (13d) linearizes the redundant constraint $(Ap-b)\tau_i \geq 0$, for $i\in\mathcal{I}$. Constraints (13e) and (13f) are the constraints proposed by Zhen et al. (2018) to tighten the linear relaxation. Constraints (13g), (13h), (13i), and (13j) are obtained by using McCormick relaxation (McCormick 1976). Therefore, the objective value of (13) is an upper bound on the objective value of (12) and hence (11).

REMARK 1. To construct (13), we need to compute LB_τ and UB_τ. Since in the optimal solution (τ^*, p^*) of (11), we have $\tau_i^* = \frac{1}{f_i(p^*)}$, we can compute LB_{τ_i} by solving the convex optimization

$$\min_{p \in \mathbb{R}^I} \frac{1}{f_i(p)}$$
$$\text{s.t. } Ap \geq b, p \geq 0,$$

and can compute $\frac{1}{UB_{\tau_i}}$ by solving the convex optimization

$$\min_{p \in \mathbb{R}^I} f_i(p)$$
$$\text{s.t. } Ap \geq b, p \geq 0.$$

□

Hitherto, we have provided a method to obtain a "good" feasible solution (Section 3.1) and an optimization problem to provide an upper bound on the objective value of (8) (Section 3.2). In the next section, we provide a new branching method to tighten the gap between the lower and upper bounds.

3.3. New branching method for continuous variables

A typical branching method in continuous optimization is done by first choosing the branching variable and then splitting its feasible interval into two intervals (Misener and Floudas 2014, Floudas et al. 2005, Akrotirianakis and Floudas 2004). As one can notice, such branching methods result in binary trees, as in each iteration we only have two branches. In this section, we implement Voronoi diagram (Aurenhammer 1991) as a branching strategy, which provides us with many branches in each iteration with the hope of closing the optimality gap faster.

Voronoi diagram is designed to partition a set with respect to finite number of points in it. Let us assume that $\mathcal{P} = \{p^1, ..., p^K\}$ is the set of K points in the feasible set S. Then, Voronoi diagram partitions S into K subsets each of which contains only one point. The partitioning is done such that the points in the k^{th} subset are closer to p^k than any other points. Mathematically, the k^{th} subset corresponding to p^k is constructed as follows:

$$S_k = S \cap \left\{p \in \mathbb{R}^n : \;\; \|p^k - p\|_2 \leq \|p^k - p^j\|_2, \;\; \forall j : \;\; j \neq k \right\},$$

which can equivalently be formulated as

$$S_k = S \cap \left\{p \in \mathbb{R}^n : \;\; (p^j - p^k)^T p \leq \frac{1}{2}(p^j - p^k)^T (p^j + p^k), \;\; \forall j : \;\; j \neq k \right\}. \tag{14}$$

In our branching, we start with two feasible solutions $\mathcal{P}^1 = \{p^1, p^2\}$ and partition the feasible region into two subsets S_1^1 and S_2^1 using Voronoi diagram. Then, for each subset the method in Section 3.1 provides two feasible solutions p^3 and p^4 in S_1^1 and S_2^1, respectively. The second iteration uses the updated set of solutions $\mathcal{P}^2 = \{p^1, p^2, p^3, p^4\}$ to partition S. Let us denote by $\bar{S}_k^2$ the partition corresponding to the feasible solution p^k in $\mathcal{P}^2$, $k = 1, ..., 4$. Then, the branching is done by intersecting S_1^1 and S_2^1 with $\bar{S}_k^2$, $k = 1, ..., 4$. Figure 1 illustrates the first two levels of branching obtained by the algorithm.

Similar to the standard branch and bound algorithms, to avoid extra branching, we solve the relaxation problem (13) in each node to get an upper bound on the optimal value in that node. If the upper bound is lower than the objective value of the best obtained solution or the same as the lower bound on that node, then we terminate branching of that node, as we know the branching does not result in a better solution.

It is important to notice that in a node corresponding to a feasible solution p^k, the trust-region method in Section 3.1 can return p^k again, which results in a loop. To avoid loops, we have an extra step in which we check whether we obtain a new solution from the

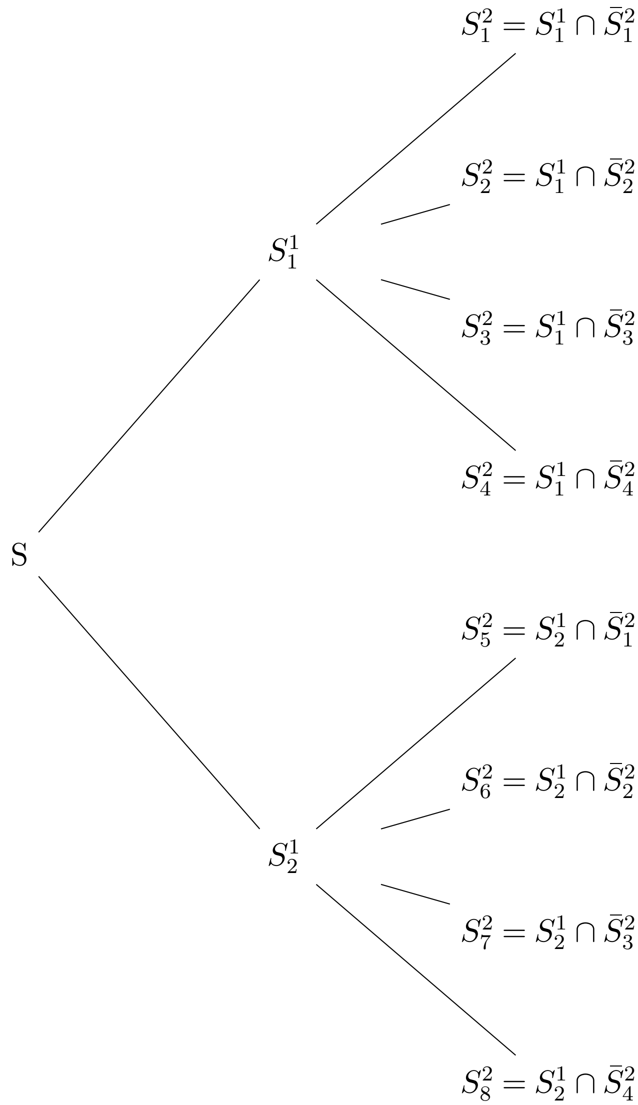


Figure 1 **Branching tree obtained by Voronoi diagram.**

trust-region method. In case the solution is the same as p^k, we find a new solution by the following procedure.

Let us denote by $\overline{S}$, $\overline{LB}_p$, and $\overline{UB}_p$, the feasible region of the current node in which we cannot find a new solution, lower bounds, and upper bounds on the solutions in $\overline{S}$, respectively. Also, let us denote by $\bar{i}$ the index that has the maximum $\overline{UB}_{p_i} - \overline{LB}_{p_i}$ among $i = 1, ..., I$. So, we have

$$p_{\bar{i}}^k \lesseqgtr \overline{LB}_{p_{\bar{i}}} + \frac{\overline{UB}_{p_{\bar{i}}} - \overline{LB}_{p_{\bar{i}}}}{2} + \epsilon,$$

where $\lessgtr$ is either $<$ or $>$ and

$$\epsilon = \begin{cases} 0 & \text{if } p_{\bar{i}}^k > \overline{LB}_{p_{\bar{i}}} + \dfrac{\overline{UB}_{p_{\bar{i}}} - \overline{LB}_{p_{\bar{i}}}}{2}, \\ 0 & \text{if } p_{\bar{i}}^k < \overline{LB}_{p_{\bar{i}}} + \dfrac{\overline{UB}_{p_{\bar{i}}} - \overline{LB}_{p_{\bar{i}}}}{2}, \\ \rho & \text{if } p_{\bar{i}}^k = \overline{LB}_{p_{\bar{i}}} + \dfrac{\overline{UB}_{p_{\bar{i}}} - \overline{LB}_{p_{\bar{i}}}}{2}, \end{cases}$$

where $\rho > 0$ is a small enough scalar. Let us assume without loss of generality that

$$p_{\bar{i}}^k < \overline{LB}_{p_{\bar{i}}} + \frac{\overline{UB}_{p_{\bar{i}}} - \overline{LB}_{p_{\bar{i}}}}{2} + \epsilon, \tag{15}$$

then, we solve the following convex quadratic optimization problem to find the new solution:

$$\begin{aligned} \min_{p \in \mathbb{R}^I} \ & (p - p^k)^T (p - p^k) \\ \text{s.t. } & p \in \overline{S} \\ & p_{\bar{i}} \geq \overline{LB}_{p_{\bar{i}}} + \frac{\overline{UB}_{p_{\bar{i}}} - \overline{LB}_{p_{\bar{i}}}}{2} + \epsilon. \end{aligned} \tag{16}$$

Using (16), we try to find the closest solution to p^k in

$$\left\{ p \in \mathbb{R}^n : p \in \overline{S}, p_{\bar{i}} \geq \overline{LB}_{p_{\bar{i}}} + \frac{\overline{UB}_{p_{\bar{i}}} - \overline{LB}_{p_{\bar{i}}}}{2} + \epsilon \right\}.$$

The reason to solve (16) is that if the trust-region algorithm converges to a local maximum, the new solution helps our method to jump to another part of the feasible region (but not far from the current solution) to explore more parts of the region. Another reason is that we want to decrease the volume of the feasible region in each branching iteration. Hence, by selecting $\bar{i}$ we are ensured that in the next branching iteration the feasible region has a shorter length along side the $\bar{i}^{\text{th}}$ axis, which implies asymptotically convergence of the algorithm, based on the following theorem.

THEOREM 1. *Let us denote by S the feasible region of (8), and its Voronoi diagram partitions S_k^m, $k = 1, ..., K^m$, in the m^{th} iteration, based on the obtained set of feasible* *solutions. Also, let us denote by $\mathcal{B}_r(p)$ a hyperball with the center p and radius r. Let opt^m be the upper bound obtained in the m^{th} iteration. Set*

$$r^m := \max_{k=1,...,K^m} \{\min\{r : S_k^m \subseteq \mathcal{B}_r(p), for\ some\ p \in S_k^m\}\}.$$

In other words, r^m is the maximum radius of the smallest hyperball among those covering the partitions S_k^m. If $r_m \to 0$ as m tends to $+\infty$, then $opt^m \searrow opt$, meaning the sequence of upper bounds asymptotically converges to the optimal value of (8)*.*

Proof Let us first reformulate the optimization problem we are dealing with in the m^{th} iteration. Given the set of obtained solutions, $\mathcal{P}^m = \{p^1, ..., p^{K^m}\}$, in the m^{th} iteration we want to solve

$$\begin{aligned} \overline{opt}^m = \max_{p \in \mathbb{R}^I} \ & \sum_{i \in \mathcal{I}} \frac{p_i}{f_i^m(p)} \\ \text{s.t.} \ & p \in S, \end{aligned} \tag{17}$$

where

$$f_i^m(p) := \max_{k=1,...,K^m} \left\{ f_i(p^k) + \nabla f_i(p^k)^T (p - p^k) \right\}.$$

Let p^* be an optimal solution of (8) and p^{*^m} be an optimal solution of (17), $m = 1, ..., +\infty$. As S is a polytope and $f_i(p)$ is convex with continuous second derivative, we know there exists $\gamma > 0$ such that for any $p \in S$, $i \in \mathcal{I}$,

$$\nabla^2 f_i(p) \preceq \gamma \mathbb{I},$$

where $\nabla^2 f_i(p) \in \mathbb{R}^{I \times I}$ is the Hessian matrix and $\mathbb{I} \in \mathbb{R}^{I \times I}$ is the identity matrix. The rest of the proof is split into proving the following four statements:

(I) If $r_m \to 0$ as m tends to $+\infty$, then for any $i \in \mathcal{I}$, we have $f_i^m(p) \xrightarrow{\text{Uniform}} f_i(p)$;

(II) For any $i \in \mathcal{I}$, if $f_i^m(p) \xrightarrow{\text{Uniform}} f_i(p)$, then

$$\sum_{i \in \mathcal{I}} \frac{p_i}{f_i^m(p)} \xrightarrow{\text{Uniform}} \sum_{i \in \mathcal{I}} \frac{p_i}{f_i(p)};$$

(III) If for any $p \in S$, $\lim_{m\to+\infty} \sum_{i\in\mathcal{I}} \frac{p_i}{f_i^m(p)} = \sum_{i\in\mathcal{I}} \frac{p_i}{f_i(p)}$, then $\max_{p\in S} \sum_{i\in\mathcal{I}} \frac{p_i}{f_i^m(p)}$ reaches $\max_{p\in S} \sum_{i\in\mathcal{I}} \frac{p_i}{f_i(p)}$,

(IV) If $\max_{p\in S} \sum_{i\in\mathcal{I}} \frac{p_i}{f_i^m(p)}$ reaches $\max_{p\in S} \sum_{i\in\mathcal{I}} \frac{p_i}{f_i(p)}$, then $opt^m \searrow opt$,

where $\xrightarrow{\text{Uniform}}$ denotes the uniform convergence.

Proof of (I): Let us fix $i \in \mathcal{I}$. To prove $f_i^m(p) \xrightarrow{\text{Uniform}} f_i(p)$, we should show that for any $\epsilon > 0$, there exists $M \in \mathbb{N}$ such that for any $m \geq M$ and any $p \in S$ we have $f_i(p) - f_i^m(p) < \epsilon$.

Given $\epsilon > 0$, let us set $\bar{\epsilon} := \sqrt{\frac{2\epsilon}{\gamma}}$. From the assumption, we know there exists $M \in \mathbb{N}$ such that for any $m \geq M$, we have $r_m < \bar{\epsilon}$. In other words, for any $m \geq M$ and $p \in S$ there exists $k \in \{1, ..., K^m\}$, such that $\|p - p^k\|_2 \leq \bar{\epsilon}$.

Furthermore, for any $p, y \in S$, we know that (see Theorem 1 in Chapter 10 of (Grossman 2014))

$$f_i(y) - f_i(p) - \nabla f_i(p)^T (y - p) \leq \frac{\gamma}{2} \|y - p\|_2^2.$$

Therefore, for any $m \geq M$ and $p \in S$, there exists $k = 1, ..., K^m$ such that

$$f_i(p) - f_i^m(p) \leq f_i(p) - f_i(p^k) - \nabla f_i(p^k)^T (p - p^k) \leq \frac{\gamma}{2} \|p^k - p\|_2^2 \leq \frac{\gamma}{2} \bar{\epsilon}^2 = \epsilon,$$

or equivalently, $f_i^m(p) \xrightarrow{\text{Uniform}} f_i(p)$.

Proof of (II): Let us fix $i \in \mathcal{I}$. Set $\bar{\gamma}_i := \max_{p\in S} p_i$, and $\bar{\gamma} := \max_{i=1,...,I} \bar{\gamma}_i$. Let us fix $\epsilon > 0$ and set $\bar{\epsilon}_i := \frac{\epsilon\Omega_i^2}{2\bar{\gamma}}$, where $\Omega_i = \min_{p\in S} f_i(p)$. As $f_i(p) > 0$ for $p \in S$ and S is bounded, $\Omega_i > 0$. By the assumption, there exists $M \in \mathbb{N}$ such that for any $m > M$ and any $p \in S$, we have $f_i(p) - f_i^m(p) < \min\left\{\frac{\Omega_i}{2}, \bar{\epsilon}\right\}$. Hence, $f_i^m(p) > \frac{\Omega_i}{2}$ for any $m \geq M$ and $p \in S$. So, we have

$$\left|\frac{p_i}{f_i^m(p)} - \frac{p_i}{f_i(p)}\right| = |p_i| \left|\frac{1}{f_i^m(p)} - \frac{1}{f_i(p)}\right| \leq \bar{\gamma} \frac{|f_i(p) - f_i^m(p)|}{|f_i(p) f_i^m(p)|} \leq \frac{2\bar{\gamma}\bar{\epsilon}}{\Omega_i^2} = \epsilon.$$

Thus, we have proved that for any $i \in \mathcal{I}$, $\frac{p_i}{f_i^m(p)} \xrightarrow{\text{Uniform}} \frac{p_i}{f^i(p)}$. As the finite summation preserves uniform convergence, we have

$$\sum_{i\in\mathcal{I}} \frac{p_i}{f_i^m(p)} \xrightarrow{\text{Uniform}} \sum_{i\in\mathcal{I}} \frac{p_i}{f_i(p)}.$$

Proof of (III): Let us set

$$g_m(p) := \sum_{i \in \mathcal{I}} \frac{p_i}{f_i^m(p)}, \quad g(p) := \sum_{i \in \mathcal{I}} \frac{p_i}{f_i(p)}.$$

We know that for any $p \in S$, $g_m(p) \geq g(p)$. Let us denote by p^{*^m} an optimal solution of $\max_{p \in S} g_m(p)$, $m = 1, ..., +\infty$, and by p^* an optimal solution of $\max_{p \in S} g(p)$. By assumption, we know for a given value of $\epsilon > 0$, there exists $M \in \mathbb{N}$ such that for any $m > M$ and any $p \in S$ we have $g_m(p) - g(p) < \epsilon$. So, for any $m > M$

$$g(p^*) + \epsilon \geq g(p^{*^m}) + \epsilon > g_m(p^{*^m}),$$

where the left inequality is due to optimality of p^*, and the right inequality is because of the uniform convergence. Therefore, for any $m > M$

$$g_m(p^{*^m}) - g(p^*) < \epsilon,$$

which concludes (III).

Proof of (IV) Based on the assumption, for a given $\epsilon > 0$, there exists $M \in \mathbb{N}$ such that for any $m \geq M$, we have $\overline{opt}^m - opt < \frac{\epsilon}{2}$. Let us denote by $\mathcal{P}^M$ the set of feasible solutions obtained until the M^{th} iteration. In the ℓ^{th} iteration of the algorithm, let us denote by $\widetilde{opt}^\ell$ the optimal value of the linearization of (17) constructed by including (13c) in (13) only for the points in $\mathcal{P}^M$. Therefore, if $\ell \geq M$, then $\widetilde{opt}^\ell \geq opt^\ell$, as $\mathcal{P}^M \subseteq \mathcal{P}^\ell$.

As it is shown by McCormick (1976), if r^ℓ reaches 0 as ℓ tends to $+\infty$ then $\widetilde{opt}^\ell \searrow \overline{opt}^M$. Therefore, there exists $L \in \mathbb{N}$ such that for any $\ell \geq L$, we have $\widetilde{opt}^\ell - \overline{opt}^M \leq \frac{\epsilon}{2}$. Hence, for any $\ell > \max\{L, M\}$ we have

$$opt^\ell - opt \leq \widetilde{opt}^\ell - opt = \widetilde{opt}^\ell - \overline{opt}^M + \overline{opt}^M - opt \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

Proof of the theorem: Combining (I), (II), (III), and (IV) implies that $opt^m \searrow opt$ as m tends to $+\infty$. □

Theorem 1 asserts that the objective value of (13) converges to the optimal value of (8). In the next section, we show how the algorithm efficiently works on a case study on a parking services pricing problem.

4. Case study

In this section, we illustrate the effectiveness of our method to solve static pricing problems formulated from continuous mixed logit model. We refer to our method as LiBiT, as it is based on **li**nearization of a **bi**convex optimization and **t**rust-region algorithm. The numerical results of this work were carried out on a Laptop featuring 4 processors 2.60 GHz and 8.00 GB RAM running Julia 1.0.3 (Bezanson et al. 2017) and MATLAB R2016a. We use JuMP 0.18.6 (Dunning et al. 2017) to pass Linear Optimization problems to IBM ILOG CPLEX 12.7.1. To compare the performance of our method, we also use two algorithms in the NLOPT package (Johnson 2014). To have a fair comparison with the other algorithms, in LiBiT we numerically find the gradient using FiniteDiff package in Julia.

Remark 2. We also used SNOPT 7.7 (Gill et al. 2005) to solve (6), however, for all instances it ran into numerical issues. Therefore, we have not reported the results obtained by this solver. □

4.1. Parking choice model description

The selection of this case study is motivated by the availability of a published, non-trivial, disaggregate parking choice model by Ibeas et al. (2014), that we can use to characterize the demand. Furthermore, this case study has been recently used by Paneque et al. (2018) to demonstrate how to integrate advanced discrete choice models in pricing problems using a a mixed integer linear programming (MILP) formulation.

The parking choice consists in three services:

- paid on-street parking (PSP),

- paid parking in an underground car park (PUP),
- free on-street parking (FSP).

The latter does not provide any revenue to the operator. Table 1 shows all explanatory variables used in the utility functions of the mixed logit model. These are features related to the age of the vehicle, the income of customers, the type of trip, the access time to the destination from the parking, and information whether the customer is a resident or not.

Given these features and following the mixed logit model proposed by Ibeas et al. (2014), we build the following three utility specifications:

$$\begin{aligned} V_{FSP,n} &= \beta^p_{FSP,n} \times p_{FSP} + q_{FSP,n} \\ &= q_{FSP,n}, \\ V_{PSP,n} &= \beta^p_{PSP,n} \times p_{PSP} + q_{PSP,n}, \\ V_{PUP,n} &= \beta^p_{PUP,n} \times p_{PUP} + q_{PUP,n}. \end{aligned}$$

The utility specification of the free on-street parking only contains the exogenous part $q_{FSP,n}$ since there is no fee to pay for that option ($p_{FSP} = 0$). The price sensitivities parameters $\beta^p_{PSP,n}$ and $\beta^p_{PUP,n}$ are then further expressed as:

$$\begin{aligned} \beta^p_{PSP,n} &= \beta_{FEE} + \beta_{FEE_{PSP(LowInc)}} \times LowInc_n + \beta_{FEE_{PSP(Resident)}} \times Residence_n \\ \beta^p_{PUP,n} &= \beta_{FEE} + \beta_{FEE_{PUP(LowInc)}} \times LowInc_n + \beta_{FEE_{PUP(Resident)}} \times Residence_n. \end{aligned}$$

The exogenous parts of utilities are modeled as:

$$\begin{aligned} q_{FSP,n} &= \quad && \beta_{AT} \times AT_{FSP} + \beta_{TD} \times TD_{FSP} && + \beta_{Origin} \times Origin_n, \\ q_{PSP,n} &= ASC_{PSP} + && \beta_{AT} \times AT_{PSP} + \beta_{TD} \times TD_{TSP}, && \\ q_{PUP,n} &= ASC_{PUP} + && \beta_{AT} \times AT_{PUP} + \beta_{TD} \times TD_{PUP} && + \beta_{AgeVeh_{\leq 3}} \times AgeVeh_{\leq 3_n}. \end{aligned}$$

Features	Definition
ASC_{PSP}	Alternative specific constant for the PSP alternative.
ASC_{PUP}	Alternative specific constant for the PUP alternative.
AT_{FSP}	The access time to the free on-street parking.
AT_{PSP}	The access time to the paid on-street parking.
AT_{PUP}	The access time to the paid underground parking.
TD_{FSP}	The access time to the destination from the free on-street parking.
TD_{PSP}	The access time to the destination from the paid on-street parking.
TD_{PUP}	The access time to the destination from the paid underground parking.
$Origin$	A dummy parameter that is 1 if the origin of the trip is internal to the town.
$\mathbf{p_{PSP}}$	**Fee for the paid on-street parking.**
$\mathbf{p_{PUP}}$	**Fee for the paid underground parking.**
$LowInc$	A dummy parameter that is 1 if the income of the customer is below 1200€/month.
$Residence$	A dummy parameter that is 1 if the customer is a resident.
$AgeVeh_{\leq 3}$	A dummy parameter that is 1 if the age of the vehicle is lower than 3 years.

Table 1 **Features used in the parking choice model.**

The values of coefficient parameters used in Ibeas et al. (2014) are depicted in Table 2. Parameters β_{AT} and β_{FEE} are assumed to be normally distributed and correlated, with $cov(\beta_{AT};\beta_{FEE}) = -\frac{3}{2}$.

	Mixed Logit
ASC_{PSP}	32
ASC_{PUP}	34
β_{AT}	$\sim Normal(-0.788, 1.06)$
β_{TD}	-0.612
β_{Origin}	-5.762
β_{FEE}	$\sim Normal(-32.3, 14.12)$
$\beta_{FEE_{PSP(LowInc)}}$	-10.995
$\beta_{FEE_{PSP(Resident)}}$	-11.44
$\beta_{FEE_{PUP(LowInc)}}$	-13.729
$\beta_{FEE_{PUP(Resident)}}$	-10.668
$\beta_{AgeVeh_{\leq 3}}$	4.037

Table 2 Values of coefficient parameters.

The pricing problem is to determine the optimal prices (or parking fees) of the two paid parking services, *i.e.*, p_{PSP} and p_{PUP}, so that the revenue of the operator is maximized. Since the purpose is to show the practicality of LiBiT, we consider an unlimited capacity for the parking services. In the pricing problem, p_{PSP} and p_{PUP} are the only endogenous variables, and all others are exogenous demand variables for which values are given.

4.2. Numerical results

We compare the performance of LiBiT with the other algorithms by applying them to instances generated with the above-mentioned features.

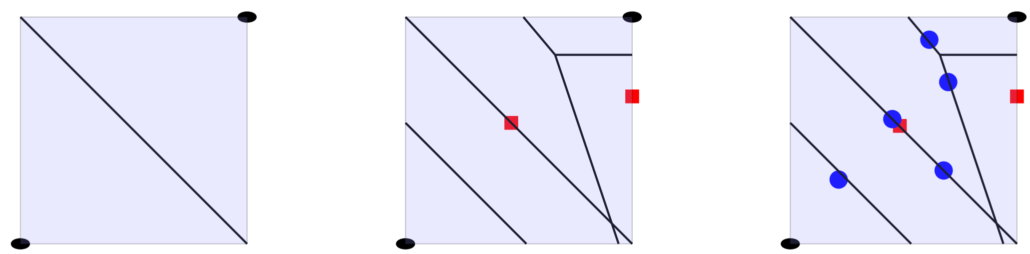
We first show how LiBiT works to solve the pricing problem containing the continuous mixed logit model. As there are limited solvers capable of dealing with optimization functions including integral, we compare our results with two algorithms provided in NLopt package (Johnson 2014) that are capable of solving (6) asymptotically. We use Direct−L (Gablonsky and Kelley 2001), which uses systematic splitting methods to divide the feasible region into smaller rectangles, and ESCH (da Silva Santos et al. 2010), which is a modified evolutionary algorithm for global optimization problems. The main drawback of the algorithms is that they do not provide any optimality guarantee.

Then, we apply LiBiT to solve the pricing problem with discrete mixed logit model and compare its performance with the existing methods in the literature.

4.2.1. Results on the continuous mixed logit model For the continuous mixed logit model (6), we set the time limit to 17 hours. This is because the objective function contains an integral, which needs a lot of computational efforts. To be able to compute the integral we limit the box to the 0.99 confidence set, i.e., $[-3.6, 1.94] \times [-68.52, 3.92]$, and use the Cuba package (Hahn 2005, 2016) in Julia.

To have a better understanding of the optimization problem (6), we plot the objective functions for $N = 10$ and $N = 50$ customers over the standard box in Appendix A. As one can see, the objective function of the instance with $N = 10$ is more flat than the one with $N = 50$. However, both objective functions contain many local optimums.

Let us discuss the performance of our method on the instance with $N = 10$. To solve the optimization problem, LiBiT starts from trivial feasible solutions $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and use Voronoi diagram to partition the feasible region (Figure2a). Then, for each partition of the feasible region, the trust-region algorithm, Algorithm 1, is employed to obtain new solutions (red squares in Figure 2b). Then, using Voronoi diagram LiBiT partitions the



(a) First iteration (b) Second iteration (c) Third iteration

Figure 2 **Illustration of the first three iterations of LiBiT applied to the parking choice model with $N = 10$ customers**

feasible region based on the set of four feasible solutions (Figure 2b) and for each partition finds new solutions (blue dots in Figure2c) using trust-region algorithm. One can notice that there is no solution obtained by LiBiT on the bottom-right partition in Figure 2c. The reason is that LiBiT recognizes that the upper bound obtained on this partion is lower than the objective value of the best found solution and hence there is no need to investigate this area.

LiBiT continues the procedure until either the time limit is reached or the objective value of the best obtained feasible solution does not deviate from the upper bound obtained by the linearization by at most 10^{-4}.

	Lower bound	Upper bound	Opt. gap	Time (Minutes)
LiBiT	6.21	6.21	0.00%	275.17
Direct−L	6.21	-	-	1020
ESCH	6.21	-	-	1020

Table 3 **Information obtained on solving** (6) **with** $N = 10$**.**

As one can see in Table 3, all the algorithms can find the optimal solution, however, LiBiT is the only one with optimality guarantee. LiBiT can solve the problem in 4.58

hours, while the other two methods are unable to guarantee optimality of the obtained solutions within the time limit of 17 hours.

To have a better understanding of the behaviour of LiBiT to solve (6) with $N = 10$ customers, we illustrates in Figure 3 how the lower and upper bounds are improved over time. For this instance, LiBiT finds the optimal solution after 4 iterations but needs 27 iterations to close the optimality gap.

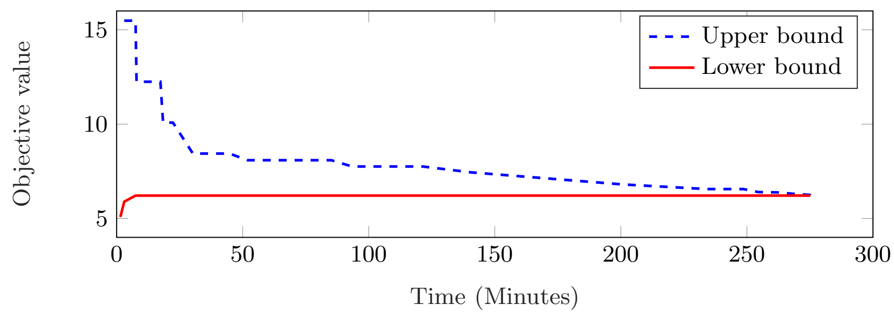


Figure 3 Illustration of the LiBiT convergence for solving (6) **with** $N = 10$ **after each branching.**

The performance of the algorithms are similar when we increase N from 10 to 50. The only difference is that LiBiT can guarantee optimality of the obtained solution after 16.92 hours. The main reason for the long computation time is that, the computational complexity of the objective function increases when N increases due to numerical derivations as well as integral. Because of this complexity, each iteration of LiBiT takes much longer when N increases from 10 to 50. It is worth emphasizing that the other two algorithms cannot guarantee optimality of the solution.

To further analyze the behaviour of LiBiT, we illustrate the improvements of lower and upper bounds over time in Figure 4. As one can see, after the fourth iteration, there is no improvement in the lower bound while the upper bound keeps getting improved until the 21st iteration.

	Lower bound	Upper bound	Opt. gap	Time (Minutes)
LiBiT	31.30	31.30	0.00%	1015.22
Direct−L	31.30	-	-	1020
ESCH	31.30	-	-	1020

Table 4 Information obtained on solving (6) with $N = 50$.

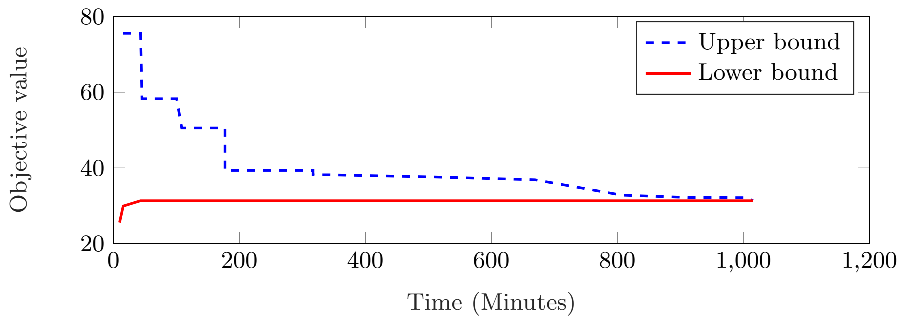


Figure 4 Illustration of the LiBiT convergence for solving (6) with $N = 50$ after each branching.

We can also compare the improvement obtained after each iteration in LiBiT for the instances with $N = 10$ and $N = 50$. Figure 5 shows that the improvements in the optimality gaps are rather close for the instances with $N = 10$ and $N = 50$. Such a similar improvement occurs as the computational complexity of the problems (10) and (13) used in LiBiT are not dependent on N. So, changes in N should not affect the performance of LiBiT after each iteration, while they affect the computational time. We should emphasize that the use of numerical derivation and integral is not necessary for the above instances, and one can analytically derive the gradient function as well as the integral function, which can boost the computation times of LiBiT. We do not use the analytical gradients and integrals to have a fair comparison with other methods. Moreover, the branching algorithm is compatible with parallel computations. After each branching iteration, the computations of the lower and upper bounds on different nodes can be done in different CPUs and the results can be analyzed in one specific CPU. Such computations can decrease the computation time dramatically.

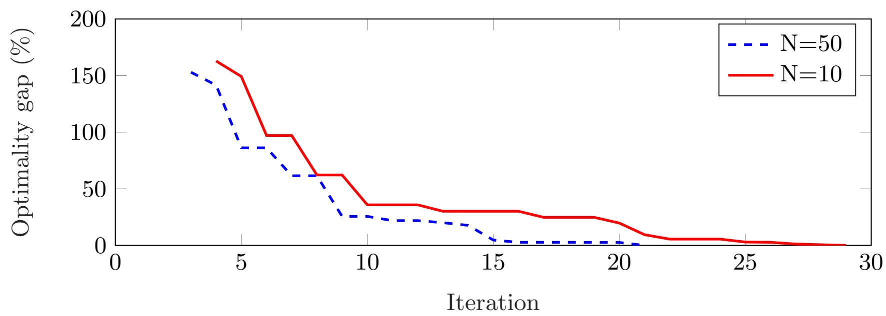


Figure 5 **Illustration of the improvement of optimality gap for solving** (6) **with** $N = 10$ **and** $N = 50$ **after each branching iteration.**

4.2.2. Results on the discrete mixed logit model In order to compare LiBiT with the algorithm proposed by Li et al. (2019), which is a local optimizer, we followed their approach by considering that the market contains $M = 100$ customer segments and that the probability for a single customer to belong to any of these segments is the same (*i.e.*, $w_m = \frac{1}{M}, m = 1, 2, ...M$). Since no individual specific variables are included in their utility specifications, we used the features of one single individual, and then randomly generated 100 points for β_{AT} and β_{FEE}. The choice probabilities are then given by (7).

Table 5 provides the results obtained by both algorithms. As one can see, both algorithms can obtain the optimal solution rather fast (in around half a Second). Since the algorithm proposed by Li et al. (2019) is a local optimizer it is expected to reach to a solution faster than LiBiT, but the time difference is negligible. For this instance, LiBiT finds the optimal solution without conducting any branching iteration. In other words, the optimal value of the solution found using the trust-region method is the same as the optimal value of (13).

Finally, we also tested our LiBiT optimizer on the simple MNL model by assuming fixed $\beta_{AT} = -0.788$ and $\beta_{FEE} = -32.3$ parameters for 10 and 50 customers. As this type of

	Lower bound	Upper bound	Opt. Gap	Time (Seconds)
LiBiT	0.2824	0.2824	0.00%	0.51
Local optimizer (Li et al. 2019)	0.2824	-	-	0.45

Table 5 **Comparison between the local optimizer algorithm proposed by Li et al. (2019) and LiBiT.**

problem can be solved using the off-the-shelf optimization solver, we use the open-source mixed integer nonlinear optimization solver SCIP 5.0.1 (Gleixner et al. 2017). To pass the Nonlinear Optimization problems to SCIP, we use OPTI Toolbox (Currie and Wilson 2012) developed in MATLAB.

Table 6 provides the comparison between LiBiT and the solver SCIP. As one can see, SCIP not only cannot reach to a feasible solution better than the origin but also it is unable to find a proper upper bound on the optimal value of the problem within the time limit of 600 Seconds. Despite SCIP's inabilities, LiBiT can find the optimal solution and guarantee optimality in less than 33 seconds.

	Lower bound	Upper bound	Opt. Gap	Time (Seconds)
LiBiT	6.36	6.36	0.00%	32.22
SCIP	0	4.2×10^{19}	10^{22}%	600

Table 6 **Information obtained on solving** (6) **with degenerate mixing probability measures with** $N = 10$ **customers.**

To have a better understanding on how LiBiT converges to the optimal solution, we illustrate how the optimality gap is reduced over time. As Figure 6 shows LiBiT finds the optimal solution in the first iteration and attempts to close the optimality gap by doing 23 branching iterations.

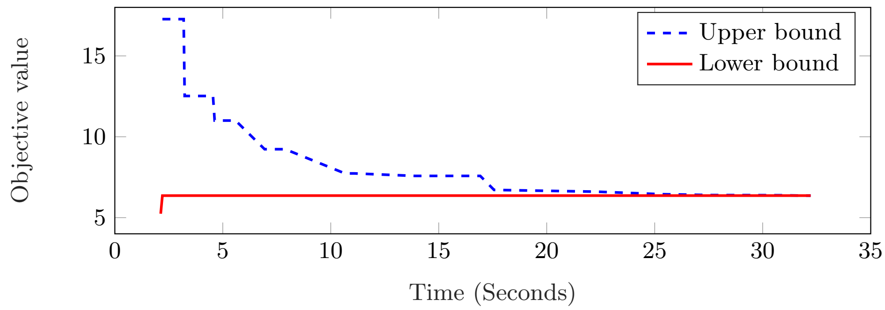


Figure 6 **Illustration of the LiBiT convergence for solving** (6) **with degenerate mixing probability measures with** $N = 10$ **after each branching.**

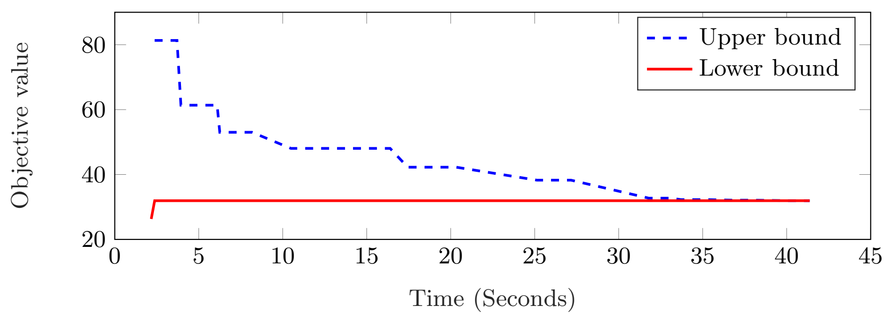


Figure 7 **Illustration of the LiBiT convergence for solving** (6) **with degenerate mixing probability measures with** $N = 50$ **after each branching.**

For the instance with $N = 50$, we see similar performances of SCIP and LiBiT. As Table 7 shows, SCIP struggles in finding a nontrivial solution and a proper upper bound for this instance in 600 Seconds. However, LiBiT finds the optimal solution and can guarantee its optimality in 41.37 seconds

	Lower bound	Upper bound	Opt. gap	Time (seconds)
LiBiT	31.93	31.93	0.00%	41.37
SCIP	0	10^{20}	10^{23}%	600

Table 7 **Information obtained on solving** (6) **with degenerate mixing probability measures with** $N = 50$.

Similar to the results of applying LiBiT to solve instances with continuous mixed logit model, the increase in N results in the increase in the computational time to solve instances with MNL model. However, the difference between the computation time of instances with the MNL model is much less than the one for the continuous model. This is because, as mentioned before, numerical computations of derivative and integral are time consuming procedure, which can be avoided by using analytical formulas.

5. Conclusions

Pricing problems under disaggregate demand assumptions is still an under explored area of research in transportation in despite of its numerous applications. In this paper, we explored a static multi-product pricing problem under a continuous mixed logit model. To the best of our knowledge, this highly general and highly used continuous mixed logit model had never been considered in pricing problems before.

We designed an efficient iterative optimization algorithm that asymptotically converges to the optimal solution. We used linear optimization problems designed based on a trust-region approach to approximate the problem from below and therefore find a "good" feasible solution. We then used piecewise linear approximations as well as McCormick relaxation to obtain an upper bound on the optimal value of the nonlinear optimization problem. Thanks to a new branching method, we then tightened the optimality gap and proved asymptotic convergence of our algorithm.

The effectiveness of this general algorithm was demonstrated on a parking services pricing case, and benchmark against solvers and existing contributions in the literature were performed. Our algorithm can accommodate a large variety of choice models available in the literature, including advanced choice models allowing complex and precise representations of individual behavior. We therefore hope that our work can motivate further research on pricing problems that better capture the interactions between supply and demand decisions.

References

Akçay Y, Natarajan HP, Xu SH (2010) Joint dynamic pricing of multiple perishable products under consumer choice. *Management Science* 56(8):1345–1361.

Akrotirianakis IG, Floudas CA (2004) Computational experience with a new class of convex underestimators: Box-constrained NLP problems. *Journal of Global Optimization* 29(3):249–264.

Aurenhammer F (1991) Voronoi diagramsa survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)* 23(3):345–405.

Aydin G, Porteus EL (2008) Joint inventory and pricing decisions for an assortment. *Operations Research* 56(5):1247–1255.

Aydin G, Ryan JK (2000) Product line selection and pricing under the multinomial logit choice model. *Proceedings of the 2000 MSOM conference* (Citeseer).

Bazaraa MS, Sherali HD, Shetty CM (2013) *Nonlinear programming: theory and algorithms* (John Wiley & Sons).

Ben-Akiva M, Bierlaire M (2003) Discrete choice models with applications to departure time and route choice. *Handbook of transportation science*, 7–37 (Springer).

Bertsimas D, Sian Ng Y, Yan J (2020) Joint frequency-setting and pricing optimization on multimodal transit networks at scale. *Transportation Science* (Articles in Advance).

Bezanson J, Edelman A, Karpinski S, Shah V (2017) Julia: A fresh approach to numerical computing. *SIAM Review* 59(1):65–98.

Boyd S, Vandenberghe L (2004) *Convex optimization* (Cambridge University Press).

Conn AR, Gould NI, Toint PL (2000) *Trust region methods*, volume 1 (SIAM).

Currie J, Wilson DI (2012) OPTI: Lowering the barrier between open source optimizers and the industrial MATLAB user. Sahinidis N, Pinto J, eds., *Foundations of Computer-Aided Process Operations* (Savannah, Georgia, USA).

da Silva Santos CH, Goncalves MS, Hernandez-Figueroa HE (2010) Designing novel photonic devices by bio-inspired computing. *IEEE Photonics Technology Letters* 22(15):1177–1179.

Daziano RA, Sarrias M, Leard B (2017) Are consumers willing to pay to let cars drive for them? Analyzing response to autonomous vehicles. *Transportation Research Part C: Emerging Technologies* 78:150–164.

Dong L, Kouvelis P, Tian Z (2009) Dynamic pricing and inventory control of substitute products. *Manufacturing & Service Operations Management* 11(2):317–339.

Du C, Cooper WL, Wang Z (2016) Optimal pricing for a multinomial logit choice model with network effects. *Operations Research* 64(2):441–455.

Dunning I, Huchette J, Lubin M (2017) JuMP: A modeling language for mathematical optimization. *SIAM Review* 59(2):295–320.

Floudas C, Akrotirianakis I, Caratzoulas S, Meyer C, Kallrath J (2005) Global optimization in the 21st century: Advances and challenges. *Computers & Chemical Engineering* 29(6):1185 – 1202.

Gablonsky JM, Kelley CT (2001) A locally-biased form of the direct algorithm. *Journal of Global Optimization* 21(1):27–37.

Gallego G, Wang R (2014) Multiproduct price optimization and competition under the nested logit model with product-differentiated price sensitivities. *Operations Research* 62(2):450–461.

Gill PE, Murray W, Saunders MA (2005) SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review* 47(1):99–131.

Gleixner A, Eifler L, Gally T, Gamrath G, Gemander P, Gottwald RL, Hendel G, Hojny C, Koch T, Miltenberger M, Müller B, Pfetsch ME, Puchert C, Rehfeldt D, Schlösser F, Serrano F, Shinano Y, Viernickel JM, Vigerske S, Weninger D, Witt JT, Witzig J (2017) The SCIP Optimization Suite 5.0. Technical report, Optimization Online.

Gorski J, Pfeuffer F, Klamroth K (2007) Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research* 66(3):373–407.

Grossman SI (2014) *Multivariable calculus, linear algebra, and differential equations* (Academic Press).

Hahn T (2005) Cubaa library for multidimensional numerical integration. *Computer Physics Communications* 168(2):78–95.

Hahn T (2016) Concurrent cuba. *Computer Physics Communications* 207:341–349.

Han B, Ren S, Bao J (2020) Mixed logit model based on improved nonlinear utility functions: A market shares solution method of different railway traffic modes. *Sustainability* 12(4):1406.

Hanson W, Martin K (1996) Optimizing multinomial logit profit functions. *Management Science* 42(7):992–1003.

Hensher DA, Rose JM, Greene WH (2005) *Applied choice analysis: a primer* (Cambridge University Press).

Hess S, Daly A, Batley R (2018) Revisiting consistency with random utility maximisation: theory and implications for practical work. *Theory and Decision* 84(2):181–204.

Hopp WJ, Xu X (2005) Product line selection and pricing with modularity in design. *Manufacturing & Service Operations Management* 7(3):172–187.

Huh WT, Li H (2015) Pricing under the nested attraction model with a multistage choice structure. *Operations Research* 63(4):840–850.

Ibeas A, DellOlio L, Bordagaray M, Ortúzar JdD (2014) Modelling parking choices considering user heterogeneity. *Transportation Research Part A: Policy and Practice* 70:41–49.

Jalali H, Carmen R, Van Nieuwenhuyse I, Boute R (2019) Quality and pricing decisions in production/inventory systems. *European Journal of Operational Research* 272(1):195–206.

Johnson SG (2014) The NLopt nonlinear-optimization package.

Li H, Huh WT (2011) Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications. *Manufacturing & Service Operations Management* 13(4):549–563.

Li H, Webster S (2017) Optimal pricing of correlated product options under the paired combinatorial logit model. *Operations Research* 65(5):1215–1230.

Li H, Webster S, Mason N, Kempf K (2019) Product-line pricing under discrete mixed multinomial logit demand: Winner2017 M&SOM practice-based research competition. *Manufacturing & Service Operations Management* 21(1):14–28.

Li W, Kamargianni M (2019) An integrated choice and latent variable model to explore the influence of attitudinal and perceptual factors on shared mobility choices and their value of time estimation. *Transportation Science* Vol. 54, No. 1:6283.

Maddah B, Bish EK (2007) Joint pricing, assortment, and inventory decisions for a retailer's product line. *Naval Research Logistics (NRL)* 54(3):315–330.

Manski CF (1977) The structure of random utility models. *Theory and Decision* 8(3):229–254.

McCormick GP (1976) Computability of global solutions to factorable nonconvex programs: Part iconvex underestimating problems. *Mathematical Programming* 10(1):147–175.

McFadden D (1977) Modelling the choice of residential location. Cowles Foundation Discussion Papers 477, Cowles Foundation for Research in Economics, Yale University.

McFadden D, Train K (2000) Mixed MNL models for discrete response. *Journal of Applied Econometrics* 15(5):447–470.

McFadden D, Zarembka P (1974) Frontiers in econometrics. *Conditional Logit Analysis of Qualitative Choice Behavior* 105–142.

Misener R, Floudas CA (2014) ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations. *Journal of Global Optimization* 59(2-3):503–526.

Paneque MP, Gendron B, Lurkin V, Azadeh SS, Bierlaire M (2018) A lagrangian relaxation technique for the demand-based benefit maximization problem. *Proceedings of the 18th Swiss Transport Research Conference* (Ascona, Switzerland).

Pardalos PM, Vavasis SA (1991) Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization* 1(1):15–22.

Song JS, Xue Z (2007) Demand management and inventory control for substitutable products. *Working paper.*

Soon W (2011) A review of multi-product pricing models. *Applied Mathematics and Computation* 217(21):8149–8165.

Sumida M, Gallego G, Rusmevichientong P, Topaloglu H, Davis J (2019) Revenue-utility tradeoff in assortment optimization under the multinomial logit model with totally unimodular constraints. Technical report, Cornell University, Ithaca, NY.

Train KE (2003) *Mixed Logit*, 138154 (Cambridge University Press).

Vij A, Krueger R (2017) Random taste heterogeneity in discrete choice models: Flexible nonparametric finite mixture distributions. *Transportation Research Part B: Methodological* 106:76–101.

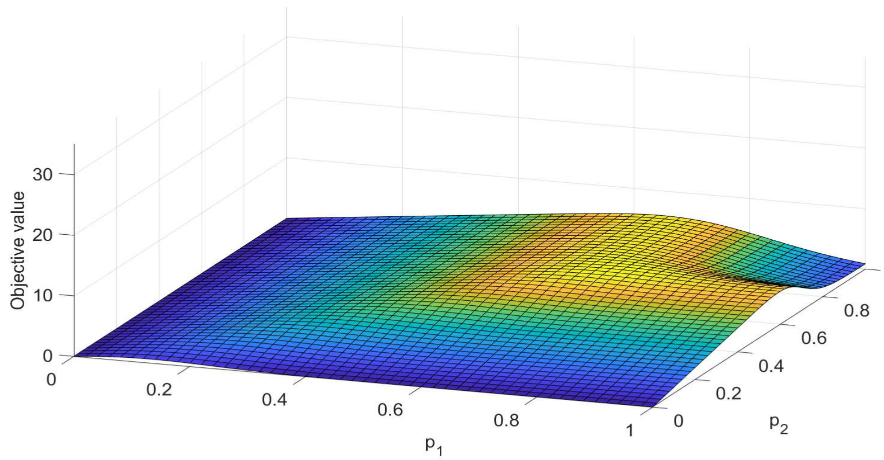
Ye M, Chen Y, Yang G, Wang B, Hu Q (2020) Mixed logit models for travelers mode shifting considering bike-sharing. *Sustainability* 12(5):2081.

Zhang D, Lu Z (2013) Assessing the value of dynamic pricing in network revenue management. *INFORMS Journal on Computing* 25(1):102–115.

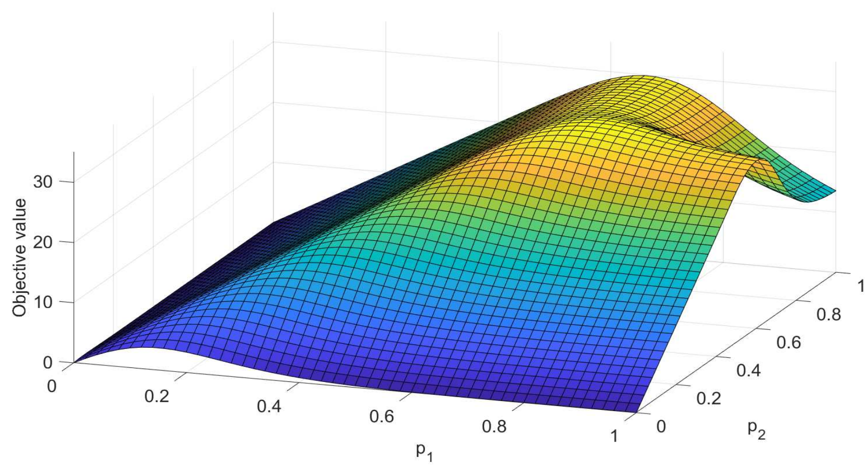
Zhang H, Rusmevichientong P, Topaloglu H (2018) Multiproduct pricing under the generalized extreme value models with homogeneous price sensitivity parameters. *Operations Research* 66(6):1559–1570.

Zhen J, Marandi A, den Hertog D, Vandenberghe L (2018) Disjoint bilinear programming: a two-stage robust optimization perspective. *Optimization Online*.

A. Illustration of the objective functions of the continuous mixed logit model for the case study



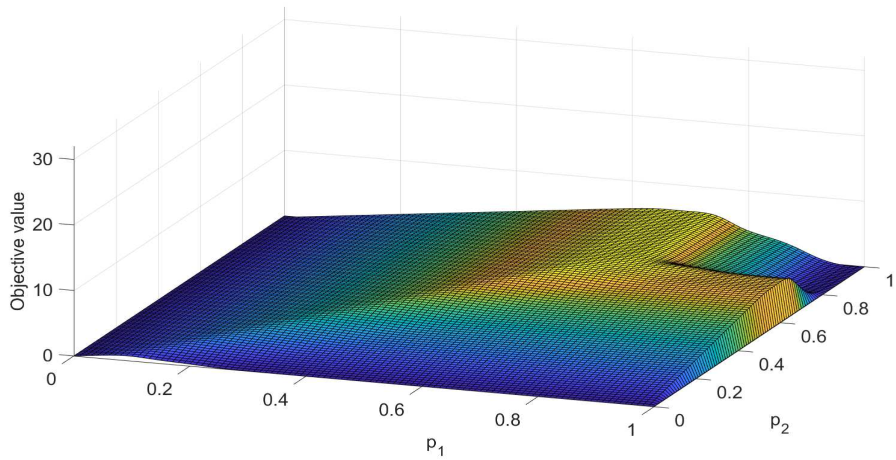
(a) $N = 10$



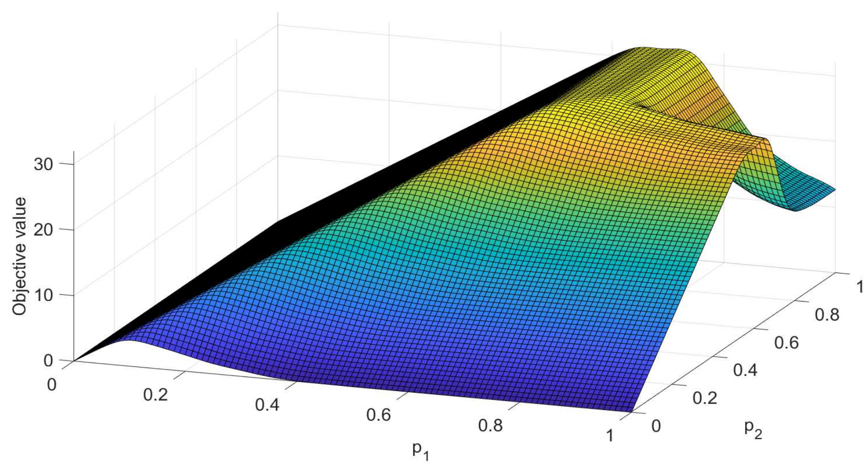
(b) $N = 50$

Figure 8 **Illustration of the objective function of** (6) **for the parking choice model with** $N = 10$ **and** $N = 50$ **customers.**

B. Illustration of the objective functions of the MNL model for the case study



(a) $N = 10$



(b) $N = 50$

Figure 9 Illustration of the objective function of (6) **with a degenerate mixing probability measure for the parking choice model with** $N = 10$ **and** $N = 50$ **customers.**