A Spatial Branch and Bound Algorithm for Continuous Pricing with Advanced Discrete Choice Demand Modeling

Tom Haering*[1] and Michel Bierlaire[2]

[1]Doctoral assistant, TRANSP-OR, EPFL, Switzerland
[2]Professor, TRANSP-OR, EPFL, Switzerland

Short summary

In this paper, we present a spatial branch and bound algorithm to tackle the continuous pricing problem, where demand is captured by an advanced discrete choice model (DCM). Advanced DCMs, like mixed logit or latent class models, are capable of modeling demand on the level of individuals very accurately due to a focus on behavioral realism. The downside of such realistic models is that it is highly nontrivial to include the resulting demand probabilities into an optimization problem, as they usually do not have a convex or even closed-form expression when decision variables are part of the choice model. To this end, a simulation procedure proposed by Paneque et al. (2021) is applied to get a formulation as a mixed integer linear program (MILP). However, due to the large number of variables stemming from the simulation, this MILP is very hard to solve. We first propose to solve the problem as a non-convex quadratically constrained quadratic program (QCQP) instead, where total unimodularity guarantees the integrality of the solution. Isolating all non-convexity into a set of bilinear constraints leads to a formulation as a non-convex quadratically constrained linear program (QCLP) that proves computationally beneficial for general-purpose solvers. Lastly, we present a spatial branch and bound algorithm that employs the McCormick envelope to obtain relaxations and makes use of total unimodularity to generate feasible solutions and thus lower bounds for the maximization fast. We compare the proposed method to the fastest commercially available solver GUROBI, on a parking choice case study from Ibeas et al. (2014). The results show that the custom spatial branch and bound approach outspeeds GUROBI by a factor of at least 35x for the MILP formulation and at least 2.5x for the QCLP in single-price optimization, and a factor of at least 4.5x for the QCQP and 1.3x for the QCLP when optimizing multiple prices simultaneously. The ratio of the speedup further increases with the size of the instance.
Keywords: branch and bound, discrete choice, mixed multinomial logit, optimization, pricing, simulation

1 Introduction

Pricing optimization is essential when pricing decisions need to be made for one or multiple products, particularly when there are cross-effects between their demands (Talluri & Van Ryzin, 2004). This problem can arise in various areas, including revenue management for airlines, railways, and hotels, assortment pricing in retail, or product line pricing in consumer goods industries.
While previous research has utilized the price-dependent multinomial logit (MNL) model to optimize prices for firms offering multiple products (Dong et al., 2009; Song et al., 2021), advanced discrete choice models such as mixed logit or latent class models have not been commonly used.
DCMs can capture the heterogeneity of customer preferences and the complex interactions between product attributes, which are often lost when the demand is aggregated. Furthermore, individual-level data can be used to identify profitable customer segments and to develop targeted pricing strategies. However, modeling demand at an individual level requires more data and computational resources compared to modeling the demand on an aggregate level.
In product assortment (PA) optimization, where a seller must make discrete decisions about the selection of products and their prices, the mixed multinomial logit (MMNL) has become increasingly popular (see e.g. Feldman et al., 2022). MMNL is regarded as a potent tool that captures the cross-effects in demand and can approximate any random utility choice model arbitrarily closely (Train, 2009). Since the PA problem under the MMNL choice model (or any other advanced choice

model that leads to non-convex probability formulas) is NP-hard (G. Li et al., 2015; Désir et al., 2015), much work has been focused on deriving upper bounds and efficient approximations, with the recent exception of Sen et al. (2017) who propose an exact conic MIP approach. Despite its theoretical and practical relevance, the MMNL model and its incorporation into revenue maximization have received little attention in the dynamic pricing literature (e.g. Keskin, 2014), with researchers often sacrificing behavioral realism for tractable (concave) formulations and therefore considering MNL (Dong et al., 2009; Keller et al., 2014) or nested logit H. Li & Huh (2011) instead. A general implementation approach for integrating any advanced choice model into an optimization problem has been proposed in Paneque et al. (2021), where Monte Carlo simulation is used to generate a deterministic problem at the cost of an increase in complexity since the resulting mixed integer linear problem (MILP) involves finding the best price over a large number of scenarios, generated by taking draws from the stochastic components of the formulation. With a sufficiently large number of draws, the MILP formulation guarantees convergence to globally optimal solutions. However, since the complexity of the MILP scales exponentially with the number of draws, the approach can currently only be applied to solving small-scale instances, i.e., with few individuals and alternatives.

In this work, we extend the MILP approach in Paneque et al. (2021) by first restating it as a non-convex quadratically constrained quadratic program (QCQP), and then as a non-convex quadratically constrained linear program (QCLP), for which we develop a spatial branch and bound algorithm that efficiently solves the problem for large numbers of draws. We compare the MILP, the QCQP, and QCLP formulations (all solved using the mathematical solver GUROBI) to our spatial branch and bound approach by application to a parking choice case study by Ibeas et al. (2014).

2 Methodology

We first present the original MILP formulation of the CPP that results when applying the approach of Paneque et al. (2021) directly:

MILP formulation

$$\max_{p,\omega,U,H} \frac{1}{R}\sum_r\sum_n\sum_i p_i\omega_{inr}$$

s.t.

$$\begin{aligned}
\sum_i \omega_{inr} &= 1 && (\mu_{nr})\\
H_{nr} &= \sum_i U_{inr}\omega_{inr} && (\zeta_{nr})\\
H_{nr} &\geq U_{inr} && (\alpha_{inr})\\
U_{inr} &= \sum_{k\neq p}\beta_k x_{ink} + \beta_p p_i + \varepsilon_{inr} && (\kappa_{inr})\\
\omega &\in \{0,1\}\\
p, U, H &\in \mathbb{R}
\end{aligned}$$

Formulation 1 – CPP as a MILP

Consider a set of $n = \{1, \ldots, N\}$ individuals choosing exactly one alternative among a set of $i = \{1, \ldots, I\}$ alternatives. Assume that in every scenario r, each individual n selects the alternative i corresponding to the maximal utility U_{inr}. The utility function depends on K parameters β_k which are exogenous and estimated using an advanced discrete choice model where all prices were fixed. These parameters are multiplied by individual or alternative specific attributes x_{ink}, which are also exogenous, just like the added error term draws ε_{inr}. For the CPP the price p_i of alternative i becomes a decision variable that is to be optimized in order to maximize profit. Note that the only assumption we make on the utilities is that they are linear in the price variables. Denote by ω_{inr} the binary decision variable that indicates whether individual n chooses alternative i in scenario r. The choice probabilities are then approximated by $P_n(i) \approx \frac{1}{R}\sum_r \omega_{inr}$ and are guaranteed to converge to the real probabilities with a sufficiently large number of scenarios R, see Paneque et al. (2021). The objective function is equal to the profit and is thus defined as the average number of times that individual n chooses alternative i over all scenarios r (i.e. its choice probability) multiplied by the alternative's price p_i. The constraints define the individual choices: Constraints (μ_{nr}) guarantee that only one alternative can be chosen per individual and scenario. Constraints (κ_{inr}) model the utility U_{inr} of each alternative i for individual n in scenario r. Constraints (ζ_{nr}) and constraints (α_{inr}) ensure that the choice being made corresponds to the one with the highest utility. Note that both the objective and the constraints ζ_{nr} contain the product $p_i\omega_{inr}$, which can be linearized using a big-M approach.

Continuous reformulation

$$\begin{aligned}
&\max_{p,\omega,U,H} \frac{1}{R}\sum_r\sum_n\sum_i p_i\omega_{inr} \\
&\text{s.t.} \\
&\sum_i \omega_{inr} = 1 && (\mu_{nr}) \\
&H_{nr} = \sum_i U_{inr}\omega_{inr} && (\zeta_{nr}) \\
&H_{nr} \geq U_{inr} && (\alpha_{inr}) \\
&U_{inr} = \sum_{k\neq p}\beta_k x_{ink} + \beta_p p_i + \varepsilon_{inr} && (\kappa_{inr}) \\
&\omega \in [0,1] \\
&p, U, H \in \mathbb{R}
\end{aligned}$$

Formulation 2 – CPP as a QCQP

$$\begin{aligned}
&\max_{p,\omega,\eta,U,H} \frac{1}{R}\sum_r\sum_n\sum_i \eta_{inr} \\
&\text{s.t.} \\
&\sum_i \omega_{inr} = 1 && (\mu_{nr}) \\
&H_{nr} = \sum_i \Big(\sum_{k\neq p}\beta_k x_{ink} + \varepsilon_{inr}\Big)\omega_{inr} + \beta_p\eta_{inr} \\
&H_{nr} \geq U_{inr} && (\alpha_{inr}) \\
&U_{inr} = \sum_{k\neq p}\beta_k x_{ink} + \beta_p p_i + \varepsilon_{inr} && (\kappa_{inr}) \\
&\eta_{inr} = p_i\omega_{inr} && (\lambda_{inr}) \\
&\omega \in [0,1] \\
&p, \eta, U, H \in \mathbb{R}
\end{aligned}$$

Formulation 3 – CPP as a QCLP

Our first reformulation (Formulation 2) defines the CPP as a non-convex quadratically constrained quadratic problem (QCQP), with a quadratic objective and quadratic equality constraints, making them non-convex. The formulation is equivalent to the MILP in Formulation 1, except that the variables ω_{inr} are no longer constrained to be binary and instead are relaxed to be in the interval $[0,1]$. Integrality still holds, since for any price p_i the problem of choosing the alternative with the highest utility is a knapsack problem, which is totally unimodular.

Our second reformulation (Formulation 3) isolates all non-convexity into a set of bilinear constraints λ_{inr} which define the product $p_i\omega_{inr}$, turning the problem into a non-convex quadratically constrained linear program (QCLP).

Spatial Branch and Bound algorithm

We start from the QCLP formulation of the CPP shown in Formulation 3. We then construct a linear relaxation (Formulation 4) by replacing the bilinear constraints λ_{inr} by a set of inequalities (λ^1_{inr} to λ^4_{inr}) that define the McCormick envelope, see McCormick (1976). This relaxation is a commonly used device to tackle problems with bilinear constraints. For the McCormick envelope, we need to provide bounds for both variables in the product $p_i\omega_{inr}$. For ω_{inr} this is straightforward, as we can simply set the lower and upper bound to 0 and 1 respectively, whereas for the prices p_i we have to assume that its possible to define a reasonable range for each price, $p_i \in [p_i^{\mathrm{L}}, p_i^{\mathrm{U}}]$, which is usually the case in practice. To go from solving a relaxation to an approximation of the optimal solution of the original problem, we employ a so-called spatial branch and bound algorithm: We start by solving the relaxation with the initial bounds $p \in [p^L, p^U]$. If it is infeasible, the original problem is also infeasible and we are done. If it is feasible and the optimal solution found is also feasible for the original problem, we are done as well. There are multiple ways to define feasibility for the original problem, one way would be to check if all ω are close enough to integer values, another is to check how strongly the relaxed bilinear constraints are violated. If we find an optimal solution for the relaxation that is infeasible for the original, we store the objective value of that solution as an upper bound for the objective value of all subpolyhedra and we start the branching: This means that instead of looking at the entire space $p \in [p^L, p^U]$, we choose an alternative i and split the domain $[p_i^L, p_i^U]$ of p_i into two smaller intervals $[p_i^L, \frac{p_i^L+p_i^U}{2}]$ and $[\frac{p_i^L+p_i^U}{2}, p_i^U]$. Using these two new sets of bounds, we create two new (sub)polyhedra where we solve the relaxation again and iterate the procedure. After a branching, we always proceed on the branch which has the highest upper bound on its objective value (this is also called a *best-first-search*). Furthermore, after solving each relaxation, we can use the optimal value of the p_i variable that we get from the relaxation to compute an integer solution to the original problem. This can be done very efficiently as for fixed prices, all individuals and scenarios become completely independent, and finding the optimal values of the ω variables reduces to assigning 1 to the alternative with the highest utility and 0 to all others.

$$
\begin{aligned}
\max_{p,\omega,\eta,U,H} & \frac{1}{R}\sum_r\sum_n\sum_i \eta_{inr} \\
\text{s.t.} & \\
\sum_i \omega_{inr} &= 1 & (\mu_{nr}) \\
H_{nr} &= \sum_i\Big(\sum_{k\neq p}\beta_k x_{ink} + \varepsilon_{inr}\Big)\omega_{inr} + \beta_p \eta_{inr} \\
H_{nr} &\geq U_{inr} & (\alpha_{inr}) \\
U_{inr} &= \sum_{k\neq p}\beta_k x_{ink} + \beta_p p_i + \varepsilon_{inr} & (\kappa_{inr}) \\
\eta_{inr} &\geq p_i^L \omega_{inr} & (\lambda^1_{inr}) \\
\eta_{inr} &\geq p_i^U \omega_{inr} + p_i - {p_i}^U & (\lambda^2_{inr}) \\
\eta_{inr} &\leq p_i^L \omega_{inr} + p_i - {p_i}^L & (\lambda^3_{inr}) \\
\eta_{inr} &\leq p_i^U \omega_{inr} & (\lambda^4_{inr}) \\
\omega &\in [0,1] \\
p,\eta,U,H &\in \mathbb{R}
\end{aligned}
$$

Formulation 4 – Linear relaxation of the CPP using the McCormick envelope

This integer solution is feasible for the original problem and thus provides us with a global lower bound for the branching tree, meaning we can delete all branches whose upper bound is less or equal to the best (highest) known lower bound. If we ever find a solution during the branching which is also feasible for the original, we can use it as a lower bound as well. This process continues until the highest upper bound u_{max} from all active branches is at most a certain tolerance percentage $\texttt{perc}_{\texttt{tol}}$ away from the best lower bound l_{max}, i.e.:

$$\frac{l_{max}-u_{max}}{l_{max}} \cdot 100 \leq \texttt{perc}_{\texttt{tol}}.$$

If we deal with more than one price that we want to optimize, an important decision we have to make at each branching is which alternative's price to branch on. The traditional approach of spatial branch and bound is to always select the alternative i where the interval $[p_i^L, p_i^U]$ is the largest *(longest-edge-branching)*. This is comparable to a strategic exhaustive search. In our case we utilize a custom branching rule where we branch along the asset which displays the largest maximum violation of the constraints $\eta_{inr} = p_i\omega_{inr}$. Algorithm 1 provides the pseudo-code for the described procedure. It is worth noting that this branch and bound algorithm will always terminate with a 0% gap in a finite number of steps since we do not actually need to find the exact optimal price, but rather the bounds for the price such that the optimal choices are generated. It then follows that, within those sufficiently optimal bounds, the obtained price will be optimal as well.

Algorithm 1: A spatial Branch & Bound algorithm to solve the CCP

Result: $\texttt{perc}_{\texttt{tol}}$-optimal solution (p^*, ω^*, η^*) for Formulation 3.

Initialization: Set $j := 0$, $\Delta^j := [p_1^L, p_1^U] \times \cdots \times [p_J^L, p_J^U]$, $o^* := -\infty$, $\hat{o}_j := \infty$, $\Omega := \{\{\Delta^j,\ \hat{o}_j\}\}$

while $\frac{o^* - \max_j\{\hat{o}_j\}}{o^*} \cdot 100 \leq \texttt{perc}_{\texttt{tol}}$ **and** $\Omega \neq \varnothing$ **do**

- let $j := \text{argmax}\{\hat{o}_j | \{\Delta^j,\ \hat{o}_j\} \in \Omega\}$. Remove $\{\Delta^j,\ \hat{o}_j\}$ from Ω and solve Formulation 4 with bounds Δ^j.
- **if** Formulation 4 is feasible **then**
  - denote its optimal solution by (p_j, ω_j, η_j) and its optimal objective value by o_j as well as its integer optimal value $\bar{o}_j$.
  - **if** $\bar{o}_j > o^*$ **then**
    - compute $\bar{\omega}_j, \bar{\eta}_j$ from p_j and set $o^* = o_j$, $(p^*, \omega^*, \eta^*) := (p_j, \bar{\omega}_j, \bar{\eta}_j)$, delete from Ω all instances $\{\Delta^j,\ \hat{o}_j\}$ where $\hat{o}_j \leq o^*$.
  - **end**
  - **if** $o_j > o^*$ **then**
    - **if** (p_j, ω_j, η_j) is feasible for Formulation 3 **then**
      - $o^* = o_j$, $(p^*, \omega^*, \eta^*) := (p_j, \omega_j, \eta_j)$, delete from Ω all instances $\{\Delta^j,\ \hat{o}_j\}$ where $\hat{o}_j \leq o^*$.
    - **else**
      - let $i = \text{argmax}\{\max_{nr} |\eta_{inr} - p_i\omega_{inr}| \mid i \in J\}$ and divide the interval $[p_i^L, p_i^U]$ into two new intervals $[p_i^L, \frac{p_i^L + p_i^U}{2}]$ and $[\frac{p_i^L + p_i^U}{2}, p_i^U]$. Construct the two new subpolyhedra $\Delta^{'}$ and $\Delta^{''}$. Define $\hat{o}^{'} = \hat{o}^{''} := o_j$ and augment $\Omega = \Omega \cup \{\Delta^{'}, \hat{o}^{'}\} \cup \{\Delta^{''}, \hat{o}^{''}\}$.
    - **end**
  - **end**
- **end**

end

3 Results and discussion

To test the presented methodology we rely on a case study of a parking services operator, which is motivated by a published disaggregate demand model for parking choice by Ibeas et al. (2014). The choice set consists of three services: paid on-street parking (PSP), paid parking in an underground car park (PUP), and free on-street parking (FSP). Since the latter does not provide any revenue to the operator, it represents the opt-out option. We assume that all customers must pay the same price for the same service. The explanatory variables considered in the discrete choice model estimated by Ibeas et al. (2014) include the following socioeconomic characteristics: trip origin (if outside town, it affects the utility of free street parking), age of the vehicle (if less than three years old, it affects the utility of paid underground parking), the income of the driver (if low, it affects the utility of paid alternatives), area of residency of the driver (if in town, it affects the utility of paid alternatives). Additionally, the following attributes of the alternatives are considered: access time to destination, access time to parking and parking fee. For the latter two continuous variables, the corresponding coefficients are normally distributed in the utility function, making the choice model a mixed multinomial logit (MMNL). Table 1 illustrates the parameters of the discrete choice model.

Table 1 – Utility parameters derived from Ibeas et al. (2014)

Parameter	Value
ASC_{FSP}	0.0
ASC_{PSP}	32.0
ASC_{PUP}	34.0
Fee (€)	$\sim \mathcal{N}(-32.328, 14.168)$
Fee PSP - low income (€)	-10.995
Fee PUP - low income (€)	-13.729
Fee PSP - resident (€)	-11.440
Fee PUP - resident (€)	-10.668
Access time to parking (min)	$\sim \mathcal{N}(-0.788, 1.06)$
Access time to destination (min)	-0.612
Age of vehicle (1/0)	4.037
Origin (1/0)	-5.762

We run two series of tests: in the first, we fix the price of PSP to be 0.6€ and only optimize the price of PUP. This reduces the complexity enough to make meaningful comparisons to the computationally heavy MILP model. We consider a random subset of 100 customers and stepwise increase the number of random draws from 100 to 1000. For the second series of tests, we optimize both the price of PSP and PUP, but we only consider a set of 50 customers, with the same range of random draws. All experiments are performed using GUROBI 10.0.0 (Gurobi Optimization, LLC, 2021) on a 2.6 GHz 6-Core Intel Core i7 processor with 16 GB of RAM, on a single thread and with a two-hour time limit per instance. Tables 2 and 3 show the solve time with achieved optimality gap and the objective values with computed prices respectively for optimizing only the price of the PUP alternative, whereas Tables 4 and 5 depict the same outputs when optimizing PSP and PUP prices together. It is evident that computing the prices of two competing alternatives that influence each other's demands simultaneously is much more computationally challenging than optimizing a single price only. For optimizing only the PUP price, our algorithm outspeeds the MILP by a factor of at least 35x, the QCQP by at least 4x, and the QCLP by at least 2.5x, with the ratio increasing with the number of draws. When optimizing both the PSP and PUP prices, the MILP solver never terminates but for the QCQP we note a speedup of at least 4.5x and for the QCLP of at least 1.3x. Again, increasing the number of draws also increases the ratio of the speedup. For more than 600 draws, the QCLP solver is not able to generate any feasible solutions in the two-hour time window, whereas our spatial branch and bound approach finds feasible solutions with an objective value up to 2.5x higher than the one found by the MILP or the QCQP.

Table 2 – Solve time (seconds) for optimizing PUP price only

		MILP		QCQP		QCLP		B&B	
N	R	Time	Gap (%)	Time	Gap (%)	Time	Gap (%)	Time	Gap (%)
100	100	3360	0	291	0	182	0	95	0
100	200	7200	22.89	1256	0	794	0	398	0
100	300	7200	134.28	3307	0	2584	0	976	0
100	400	7200	128.56	6522	0	4275	0	1593	0
100	500	7200	141.54	7200	0.74	7093	0	2661	0
100	600	7200	118.14	7200	26.19	7200	0.57	3620	0
100	700	7200	128.45	7200	36.83	7200	0.75	5283	0
100	800	7200	113.12	7200	-	7200	5.27	7200	0.37
100	900	7200	142.49	7200	-	7200	-	7200	1.94
100	1000	7200	149.03	7200	-	7200	-	7200	10.89

Table 3 – Objective value and optimal solution for optimizing PUP price only

		MILP		QCQP		QCLP		B&B	
N	R	Obj.	Price	Obj.	Price	Obj.	Price	Obj.	Price
100	100	54.13	[0.6, 0.66]	54.13	[0.6, 0.66]	54.13	[0.6, 0.66]	54.13	[0.6, 0.66]
100	200	54.54	[0.6, 0.65]	54.6	[0.6, 0.66]	54.6	[0.6, 0.66]	54.6	[0.6, 0.66]
100	300	54.38	[0.6, 0.64]	54.48	[0.6, 0.67]	54.48	[0.6, 0.67]	54.48	[0.6, 0.67]
100	400	54.15	[0.6, 0.63]	54.39	[0.6, 0.66]	54.39	[0.6, 0.66]	54.39	[0.6, 0.67]
100	500	54.27	[0.6, 0.66]	54.23	[0.6, 0.65]	54.29	[0.6, 0.67]	54.29	[0.6, 0.67]
100	600	54.15	[0.6, 0.64]	49.13	[0.6, 0.97]	54.25	[0.6, 0.65]	54.29	[0.6, 0.67]
100	700	54.14	[0.6, 0.63]	49.18	[0.6, 0.97]	54.37	[0.6, 0.65]	54.39	[0.6, 0.66]
100	800	54.32	[0.6, 0.66]	-	-	53.82	[0.6, 0.61]	54.32	[0.6, 0.65]
100	900	54.43	[0.6, 0.67]	-	-	-	-	54.44	[0.6, 0.67]
100	1000	54.42	[0.6, 0.66]	-	-	-	-	54.35	[0.6, 0.68]

Table 4 – Solve time (seconds) for optimizing PSP and PUP prices together

		MILP		QCQP		QCLP		B&B	
N	R	Time	Gap (%)	Time	Gap (%)	Time	Gap (%)	Time	Gap (%)
50	100	7200	33.76	4283	0	1231	0	918	0
50	200	7200	70.03	6719	0	5187	0	3495	0
50	300	7200	125.65	7200	1.03	7200	1.53	7200	0.28
50	400	7200	186.21	7200	9.55	7200	20.7	7200	2.88
50	500	7200	272.78	7200	33.07	7200	40.88	7200	7.7
50	600	7200	379.53	7200	42.8	7200	41.99	7200	10.59
50	700	7200	440.25	7200	260.91	7200	-	7200	16.82
50	800	7200	495.39	7200	260.28	7200	-	7200	22.91
50	900	7200	493.85	7200	260.97	7200	-	7200	25
50	1000	7200	-	7200	260.67	7200	-	7200	29.16

Table 5 – Objective value and optimal solution for optimizing PSP and PUP prices together

		MILP		QCQP		QCLP		B&B	
N	R	Obj.	Price	Obj.	Price	Obj.	Price	Obj.	Price
50	100	27.37	[0.58, 0.72]	27.55	[0.61, 0.7]	27.55	[0.61, 0.7]	27.55	[0.61, 0.7]
50	200	23.92	[0.66, 0.95]	27	[0.55, 0.68]	27	[0.55, 0.68]	26.99	[0.55, 0.68]
50	300	16.76	[1.02, 1.1]	27.12	[0.56, 0.67]	27.09	[0.56, 0.68]	27.12	[0.56, 0.67]
50	400	15.83	[1.01, 1.17]	27.07	[0.59, 0.69]	26.29	[1.09, 0.67]	27.15	[0.56, 0.66]
50	500	12.18	[1.3, 1.34]	26.4	[0.59, 0.8]	26.42	[0.59, 0.8]	27.13	[0.57, 0.68]
50	600	9.45	[1.32, 1.66]	26.36	[0.59, 0.8]	26.37	[0.59, 0.8]	27.23	[0.56, 0.69]
50	700	8.43	[1.33, 1.85]	11.73	[1.2, 1.39]	-	-	26.87	[0.62, 0.69]
50	800	7.65	[1.76, 1.76]	11.74	[1.2, 1.39]	-	-	26.37	[0.75, 0.62]
50	900	7.63	[1.74, 1.76]	11.7	[1.2, 1.39]	-	-	26.36	[0.75, 0.62]
50	1000	-	-	11.73	[1.2, 1.39]	-	-	26.35	[0.75, 0.62]

4 Conclusions

We propose a spatial branch and bound algorithm to tackle the continuous pricing problem, where demand is captured by an advanced discrete choice model (DCM). The stochasticity in the demand is dealt with using simulation, which leads to a large MILP formulation that is difficult to solve. We show that already reformulating the MILP as a non-convex QCQP improves computational speed significantly, even more so when formulated as a non-convex QCLP. The spatial branch and bound procedure solves the problem significantly faster GUROBI on the tested instances, outspeeding the MILP by a factor of at least 35x and the QCLP by at least 2.5x for single price optimization, and outspeeding the QCQP by a factor of at least 4.5x and the QCLP by at least 1.3x for optimizing two prices simultaneously. Increasing the size of the instance also increases the ratio of the speedup. The methodology could be substantially advanced in future research by making use of the separability of each relaxed subproblem, since the price is the only complicating variable preventing the problem from being solved for each individual and scenario separately. Thus a Benders decomposition would be a suitable candidate for further improvement of the method. The authors would also like to perform a series of comparisons using only open-source software like for example SCIP and COUENNE, as not every business might be capable of purchasing a license for GUROBI. Last but not least, the code has not yet been optimized in terms of language or data structure usage, which could both have a strong impact on the performance.

References

Désir, A., Goyal, V., Segev, D., & Ye, C. (2015). Capacity constrained assortment optimization under the markov chain based choice model. *Available at SSRN 2626484*.

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

Feldman, J., Wagner, L., Topaloglu, H., & Bai, Y. (2022). Assortment optimization under the multinomial logit model with utility-based rank cutoffs. *Available at SSRN*.

Gurobi Optimization, LLC. (2021). *Gurobi Optimizer Reference Manual.* Retrieved from `https://www.gurobi.com`

Ibeas, A., Dell'Olio, L., Bordagaray, M., & Ortúzar, J. d. D. (2014). Modelling parking choices considering user heterogeneity. *Transportation Research Part A: Policy and Practice*, *70*, 41–49.

Keller, P. W., Levi, R., & Perakis, G. (2014). Efficient formulations for pricing under attraction demand models. *Mathematical Programming*, *145*, 223–261.

Keskin, N. B. (2014). Optimal dynamic pricing with demand model uncertainty: A squared-coefficient-of-variation rule for learning and earning. *Available at SSRN 2487364*.

Li, G., Rusmevichientong, P., & Topaloglu, H. (2015). The d-level nested logit model: Assortment and price optimization problems. *Operations Research*, *63*(2), 325–342.

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

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, *10*(1), 147–175.

Paneque, M. P., Bierlaire, M., Gendron, B., & Azadeh, S. S. (2021). Integrating advanced discrete choice models in mixed integer linear optimization. *Transportation Research Part B: Methodological*, *146*, 26–49.

Sen, A., Atamturk, A., & Kaminsky, P. (2017). A conic integer programming approach to constrained assortment optimization under the mixed multinomial logit model. *arXiv preprint arXiv:1705.09040*.

Song, J.-S. J., Song, Z. X., & Shen, X. (2021). Demand management and inventory control for substitutable products. *Available at SSRN 3866775*.

Talluri, K., & Van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, *50*(1), 15–33.

Train, K. E. (2009). *Discrete choice methods with simulation.* Cambridge university press.