Fast Algorithms for (Capacitated) Continuous Pricing with Discrete Choice Demand Models

Tom Haering   Robin Legault   Michel Bierlaire

Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne

33rd European Conference on Operational Research, Copenhagen
June 30 - July 3, 2024

Outline

- Introduction
- Methodology
- Experimental Results
- Conclusions

The Choice-based Pricing Problem (CPP)

CPP

- Supplier offers J products for sale. Goal: determine optimal **price** for each product to maximize total **revenue**.
- There always exists an **opt-out** option (competition, etc).
- Demand for each product is modeled using a **discrete choice model** (DCM).

DCM

- For every **costumer** n and **product** i a stochastic **utility** U_{in} is defined, which depends on **socio-economic** characteristics and **attributes** of the products (e.g. the price).

The Choice-based Pricing Problem (CPP)

Pre-estimated DCM

- **Utility** of alternative i for customer n:

$$U_{in} = V_{in} + \beta_{in}^{p} p_i + \varepsilon_{in}$$

- V_{in} : deterministic utility (exogenous)
- β_{in}^{p} : price sensitivity parameter (exogenous)
- p_i : price of alternative i (**endogenous**)
- ε_{in} : stochastic error term

Objective function

- maximize expected revenue $= \sum_n \sum_i P_n(i) p_i$

The Choice-based Pricing Problem (CPP)

- **Probability** that customer n chooses alternative i:

$$P_n(i) = \mathbb{P}(U_{in} \geqslant U_{jn} \; \forall j \in J)$$

- **Logit** ($\varepsilon_{in} \sim$ i.i.d. Gumbel(0, 1)):

$$P_n(i) = \frac{e^{V_{in}}}{\sum_{j \in C_n} e^{V_{jn}}}$$

- **Mixed Logit** (Logit + $\beta_k \sim F(\beta_k|\theta)$):

$$P_n(i) = \int \frac{e^{V_{in}(\beta_{kn})}}{\sum_{j \in C_n} e^{V_{jn}(\beta_{kn})}} f(\beta_k|\theta) d\beta_k$$

Literature

Integrating **Logit** into...

- Revenue Management (Shen and Su, 2007; Korfmann, 2018)

Integrating **Nested Logit** into...

- Toll setting (Wu et al., 2012)
- Pricing (Gallego and Wang, 2014; Müller et al., 2021)

Integrating **Mixed Logit** into...

- Toll setting (Gilbert et al., 2014)
- Pricing (Marandi and Lurkin, 2023; van de Geer and den Boer, 2022)

Literature

Integrating general DCM into optimization problems

- Formulation as a mixed-integer-linear program (**MILP**) using **Monte-Carlo simulation** (Paneque et al., 2021)
- **Heuristic** based on **Lagrangian decomposition** and grouping of scenarios (Paneque et al., 2022)
- **Exact** method based on **spatial Branch-and-Benders decomposition** (B&BD) + low-dimensional polynomial algorithm (**BEA**) (without capacity constraints) (Haering et al., 2023)

New contribution:

- Extend BEA to deal with **capacity constraints**, develop **heuristic** (with and without capacities) to handle higher dimensions, use it to **speed up** B&BD.

Outline

- Introduction
- Methodology
- Experimental Results
- Conclusions

Base layer: Monte Carlo Simulation

- **Simulate** R scenarios (draws), each with **deterministic** utilities U_{inr}:

$$
\begin{aligned}
U_{inr} &= V_{in} + \beta^{p}_{inr} p_i + \varepsilon_{inr} && \forall n \in \mathcal{N}, i \in C_n, r \in \mathcal{R} \\
&= c_{inr} + \beta^{p}_{inr} p_i && \forall n \in \mathcal{N}, i \in C_n, r \in \mathcal{R}
\end{aligned}
$$

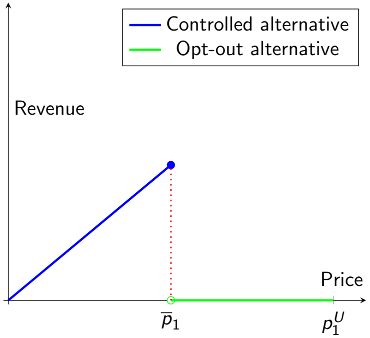
TRANSP-OR

EPFL

Breakpoints: Illustration

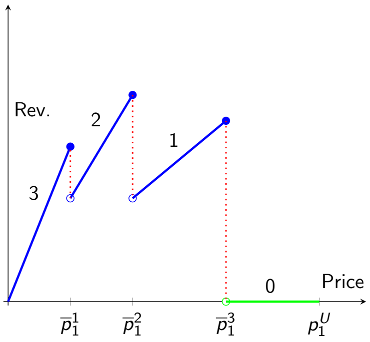
- **1** customer, **1** controlled price + opt-out
- **Breakpoint** $\overline{p}_1$:

$$U_0 = U_1 \implies U_0 = c_1 + \beta_1^p \overline{p}_1 \implies \overline{p}_1 = \frac{U_0 - c_1}{\beta_1^p}.$$

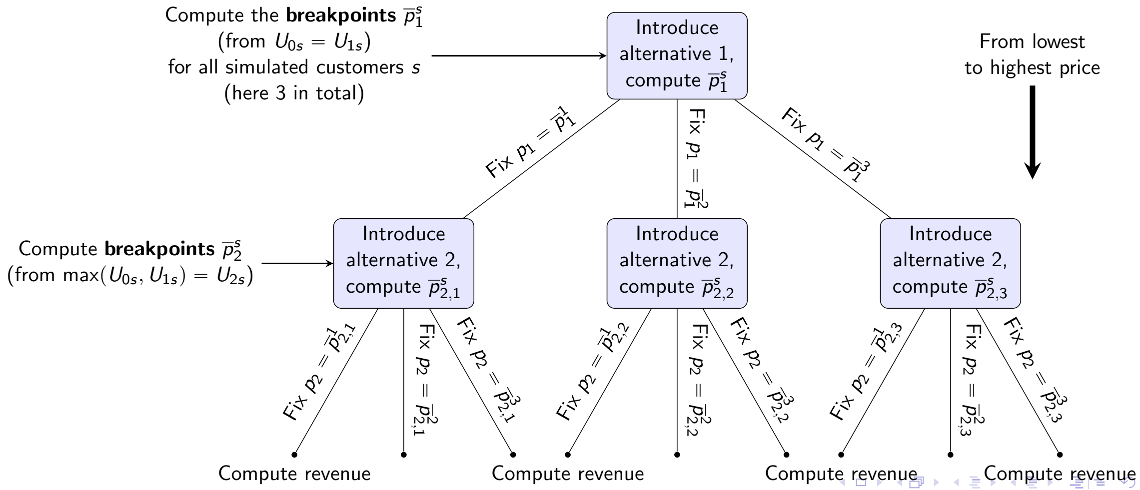


Breakpoints: Illustration

- **3** customers, **1** controlled price + opt-out
- Numbers: how many customers are **captured**



Breakpoint Exact Algorithm (BEA) (Haering et al., 2023)

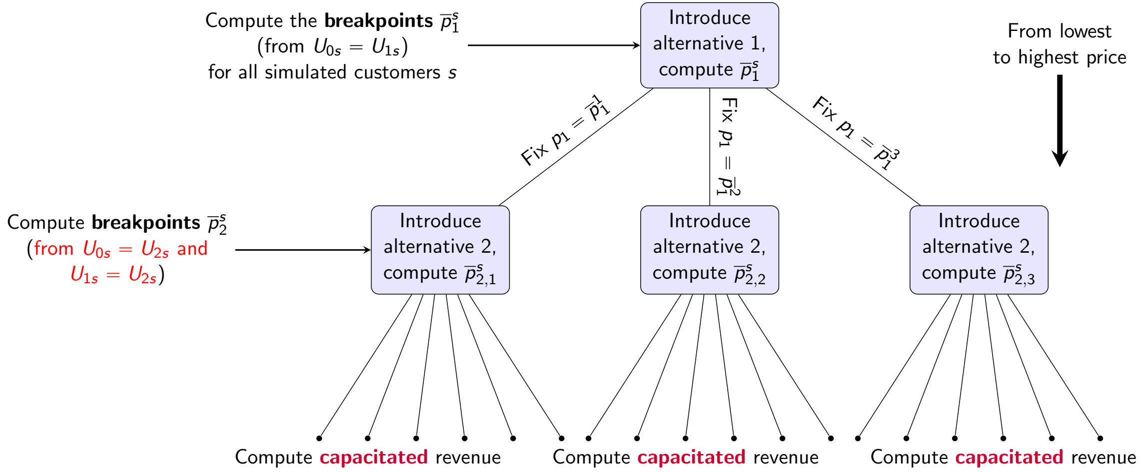


Adding capacity constraints

- Evaluating the objective function is not more difficult (assume exogenous priority queue).
- Need to compute breakpoints from not only the utility of the best alternative so far but from **all** alternative's utilities, due to **people no longer always choosing highest utility** alternative.

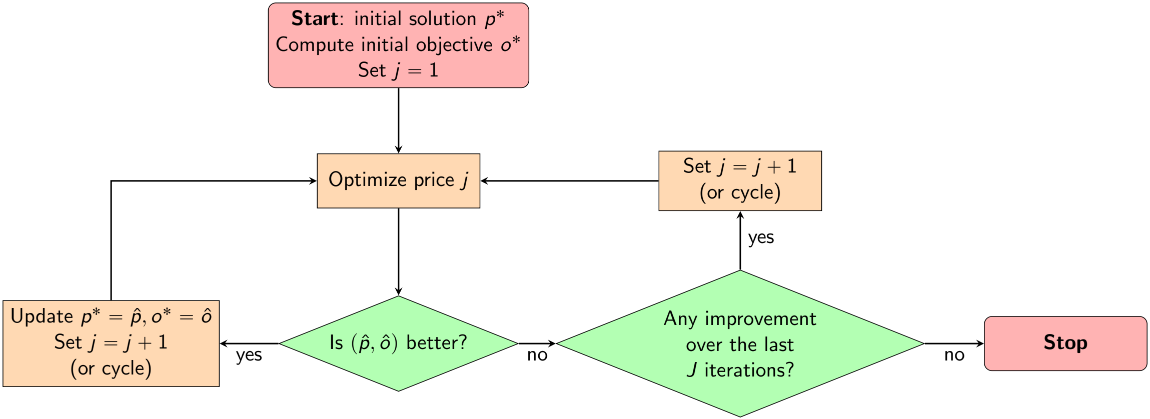
  $\implies$ Customers may switch from **any** of the previously introduced alternatives.

Breakpoint Exact Algorithm with Capacities (BEAC)



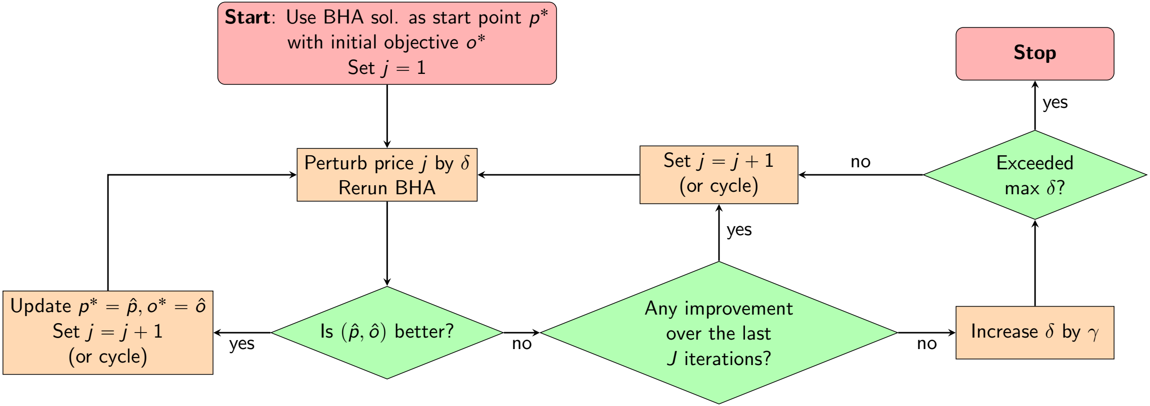
Breakpoint Heuristic Algorithm (BHA)

Coordinate descent



BHA extension via Iterated Local Search (ILS)

Escape local optima



Guiding an exact method using the heuristic solution

- Goal is to improve exact **spatial Branch & Benders** algorithm.
- Main way to **speed up** a Branch and Bound algorithm is to **improve the bounds**.
- Heuristic solution provides **strong upper bound** (initial feasible solution)
  $\rightarrow$ Reduces the number of nodes in the tree.
- Improve **lower bounds**: Valid inequalities.

Valid inequalities

Breakpoints only work if everything but one price is fixed. But...

For each simulated customer (n, r):

- **minimal breakpoint** $\check{p}_i^{nr}$ (assuming strongest competition)
- **maximal breakpoint** $\hat{p}_i^{nr}$ (assuming weakest competition)

$$p_i \leqslant \check{p}_i^{nr} \implies (n, r) \text{ is guaranteed to select } i \implies \omega_{inr} \geqslant 1$$

$$p_i \geqslant \hat{p}_i^{nr} \implies (n, r) \text{ is guaranteed to } \textit{not} \text{ select } i \implies \omega_{inr} \leqslant 0, \eta_{inr} \leqslant 0$$

Improving bounds on prices

We can consider:

$$\check{p}_i \coloneqq \min_{n,r} \check{p}_i^{nr}$$

$$\hat{p}_i \coloneqq \max_{n,r} \hat{p}_i^{nr}$$

knowing that:

$$p_i > \hat{p}_i \implies \textbf{no one} \text{ chooses alternative } i$$

$$p_i < \check{p}_i \implies \textbf{everyone} \text{ chooses alternative } i$$

(if it is in their choice set)

Improving bounds on prices

We can also say:

$$p_i > m\text{-th highest } \hat{p}_i^{nr} \implies \textbf{at most } m \text{ simulated customers choose alternative } i$$

$$p_i < m\text{-th lowest } \check{p}_i^{nr} \implies \textbf{at least } m \text{ simulated customers choose alternative } i$$

- Allows to adapt bounds to **aim at specific outcomes**.
- **We** will **assume** that for each product there should be **at least one** customer/scenario in which a product is chosen, as **else it could be removed** from the set of offered products.
- $\implies$ Replace p_i^U by $\hat{p}_i$ whenever $\hat{p}_i < p_i^U$.

Outline

- Introduction
- Methodology
- Experimental Results
- Conclusions

Case Study

Parking space operator (Ibeas et al., 2014)

- **Alternatives**: Paid-Street-Parking (PSP), Paid-Underground-Parking (PUP) and Free-Street-Parking (FSP).
- Optimize prices for PSP and PUP, FSP is the **opt-out** alternative.
- **Socio-economic characteristics**: trip origin, vehicle age, driver income, residence area.
- **Product attributes**: access time to parking, access time to destination, and parking fee (price).
- Add more alternatives by increasing access time to destination.
- Choice model is a **Mixed Logit**, $\beta_{\text{fee}}, \beta_{\text{time\_parking}} \sim \mathcal{N}(\mu, \sigma)$.

Results

Table 1: MILP vs. BEAC in the capacitated case

			MILP		BEAC	
N	R	J	Time (s)	Revenue	Time (s)	Revenue
50	2	2	4.17	27.61	0.43	27.61
50	5	2	46.95	26.51	1.72	26.51
50	10	2	180.85	27.06	11.42	27.06
50	25	2	3119.66	27.08	169.08	27.08
50	50	2	$>$5 hours	$\geqslant$25.15	1272.68	26.85
50	100	2	$>$25 hours	$\geqslant$25.11	9928.57	26.85
50	250	2	$>$45 hours	$\geqslant$23.45	$>$45 hours	$\geqslant$25.00

Results

Table 2: BHA and ILS vs. MILP and BEAC in the capacitated case

			MILP		BEAC		BHA		ILS	
N	R	J	Time (s)	Revenue	Time (s)	Revenue	Time (s)	Revenue	Time (s)	Revenue
50	2	2	4.17	27.61	0.43	27.61	0.22	27.61	1.03	27.61
50	5	2	46.95	26.51	1.72	26.51	0.32	26.46	5.91	26.51
50	10	2	180.85	27.06	11.42	27.06	0.58	27.05	20.34	27.06
50	25	2	3119.66	27.08	169.08	27.08	3.40	27.05	129.66	27.08
50	50	2	>5 hours	⩾25.15	1272.68	26.85	8.31	26.53	559.04	26.85
50	100	2	>25 hours	⩾25.11	9928.57	26.85	51.77	26.72	2791.28	26.85
50	250	2	>45 hours	⩾23.45	>45 hours	⩾25.00	455.37	26.66	15867.67	26.71
50	10	4	>10 hours	⩾22.21	>10 hours	⩾25.41	7.08	26.78	527.34	26.83
50	50	4	>20 hours	⩾22.19	>20 hours	⩾27.00	166.21	27.00	7234.88	27.00
50	100	4	>45 hours	⩾20.50	>45 hours	⩾24.86	866.97	26.67	34050.57	26.67
50	200	4	>72 hours	⩾20.32	>72 hours	⩾24.79	2762.39	26.70	106286.13	26.70

Results

N	R	J	BHA (s)
50	1000	2	15093
50	1000	3	25326
50	1000	4	69134
50	1000	5	112042
50	1000	6	178923
50	2000	2	51637
50	2000	3	84231
50	2000	4	150132
50	2000	5	193233
50	3000	2	164922
50	3000	3	184293
50	3000	4	$>$259200

Results

Table 3: BHA and ILS vs. B&BD and BEA in the uncapacitated case

			B&BD		BEA		BHA		ILS	
N	R	J	Time (s)	Revenue	Time (s)	Revenue	Time (s)	Revenue	Time (s)	Revenue
50	200	1	19	23.96	0	23.96	0.00	23.96	0.02	23.96
50	200	2	1413	26.99	12	26.99	0.00	26.99	0.03	26.99
50	200	3	34340	26.54	39,636	26.54	0.01	26.54	0.05	26.54
20	100	4	12478	10.40	>24 hours	⩾9.81	0.00	10.40	0.14	10.40
20	200	4	29213	10.40	>24 hours	⩾10.40	0.01	10.40	0.41	10.40
20	300	4	>24 hours	⩾10.38	>24 hours	⩾10.13	0.02	10.24	0.64	10.24
20	400	4	>24 hours	⩾9.81	>24 hours	⩾9.42	0.05	10.26	0.78	10.26
20	500	4	>24 hours	⩾10.01	>24 hours	⩾9.67	0.13	10.24	1.37	10.24

Results

Table 4: BHA vs. B&BD solution quality

N	R	J	BHA	B&BD	Gap (%)
20	20	3	10.281	10.281	0
20	20	4	10.271	10.28	0.09
20	20	5	10.283	10.294	0.11
20	20	6	10.290	10.302	0.12
20	20	7	10.292	10.306	0.14
20	20	8	10.330	10.336	0.06
20	20	9	10.329	10.335	0.06
20	20	10	10.293	10.300	0.07

Results

N	R	J	BHA (s)
50	500000	2	56.19
50	500000	3	77.46
50	500000	4	187.41
50	500000	5	163.23
50	500000	6	194.24
50	1000000	2	68.24
50	1000000	3	132.98
50	1000000	4	312.43
50	1000000	5	300.40
50	1000000	6	412.53

Results

Table 5: B&BD with Guidance - 10% gap

N	R	J	normal w/out VIs (s)	normal w VIs (s)	Guided w/out VIs (s)	Guided w VIs (s)	Speedup from just VIs (%)	Add. Speedup from Sol. (%)	Total speedup (%)
50	1000	3	987	1132	731	816	-14.69	27.92	17.33
50	2000	3	2878	3490	2513	2693	-21.26	22.84	6.43
50	3500	3	10325	12919	6390	7454	-25.12	42.3	27.81
50	1000	4	4662	3311	3705	2472	28.98	25.34	46.98
50	2000	4	17599	12068	10868	8288	31.43	31.32	52.91
50	3500	4	48445	31210	40061	29504	35.58	5.47	39.1
50	1000	5	8242	5428	5664	3914	34.14	27.89	52.51
50	2000	5	25842	16641	17420	12268	35.6	26.28	52.53
50	3500	5	114216	81826	85083	58754	28.36	28.2	48.56

Results

Table 6: B&BD with Guidance - 5% gap

N	R	J	normal w/out VIs (s)	normal w VIs (s)	Guided w/out VIs (s)	Guided w VIs (s)	Speedup from just VIs (%)	Add. Speedup from Sol. (%)	Total speedup (%)
50	1000	3	2372	2454	1933	2245	-3.46	8.52	5.35
50	2000	3	7883	8359	7106	7342	-6.04	12.17	6.86
50	3500	3	51964	57229	42991	47282	-10.13	17.38	9.01
50	1000	4	12062	10668	10490	8934	11.56	16.25	25.93
50	2000	4	43829	36524	36222	32929	16.67	9.84	24.87
50	3500	4	259200	240767	238777	198981	7.11	17.36	23.23
50	1000	5	24371	20590	19519	16930	15.51	17.78	30.53
50	2000	5	84104	60814	70676	48541	27.69	20.18	42.28
50	3500	5	259200	259200	259200	247944	-	-	-

Results

Table 7: B&BD with Guidance - 1% gap

N	R	J	normal w/out VIs (s)	normal w VIs (s)	Guided w/out VIs (s)	Guided w VIs (s)	Speedup from just VIs (%)	Add. Speedup from Sol. (%)	Total speedup (%)
50	1000	3	15840	16933	13239	14594	-6.9	13.81	7.87
50	2000	3	42261	45223	35882	37137	-7.01	17.88	12.12
50	3500	3	183696	195743	152833	162594	-6.56	16.93	11.49
50	500	4	47101	46719	47963	43190	0.81	7.55	8.3
50	1000	4	131122	135564	107288	105596	-3.39	22.11	19.47
50	1500	4	229620	230187	203348	202560	-0.25	12	11.78
50	2000	4	259200	259200	259200	259200	-	-	-
50	500	5	139618	125755	115783	109084	9.93	13.26	21.87
50	1000	5	259200	259200	259200	259200	-	-	-

Unfair comparison to a Mixed-Logit-specific algorithm

Table 8: Runtime (in seconds) against CoBiT (Marandi and Lurkin, 2023) for the uncapacitated CPP.

N	Approx. Points / Draws	J	CoBiT	B&B	B&BD	BEA	BHA
10	1	2	514	0	0	1	0
10	9	2	62	0	3	1	0
10	16	2	184	1	8	1	0
10	25	2	225	2	11	1	0
10	49	2	323	9	45	1	0
10	64	2	362	8	36	1	0
10	100	2	278	15	69	1	0
10	121	2	53	20	100	1	0
10	144	2	407	25	105	1	0
10	169	2	84	35	160	1	0
10	400	2	989	101	408	1	0
10	900	2	2912	286	966	4	0

Outline

- Introduction
- Methodology
- Experimental Results
- **Conclusions**

Conclusions

With capacity constraints

- **Exact:** BEAC $\approx$ **20** times faster than MILP (for **two prices** or less).
- **Heuristic:** BHA up to **5000x** times faster than BEAC (especially in high dim).

Without capacity constraints

- **Heuristic:** BHA outspeeds other approaches by factors $\geqslant \mathbf{10^6}$ but can get stuck locally.
- **Exact:** Using the solution of the BHA together with valid inequalities, we can speed up the exact spatial B&BD algorithm by $\approx$ **20%** (more in the beginning).

Future work

Pricing

- **Assortment optimization** on top of pricing.
- Could add **any constraints** for BEA / BHA since they only evaluate objective function.
- Improve escaping local optima.

Extension to other optimization problems

- Facility location, Airline scheduling and fleet assignment.
- **Maximum likelihood estimation** (utility depending on multiple parameters)
  - B&BD ✔
  - BEA ✘
  - BHA ✔ $\rightarrow$ Tradeoff between large R and optimality gap. Does not require linearity in β.

Appendix - Utility parameters reported in (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

MILP formulation (Paneque et al., 2021)

$$\max_{p,\omega,U,h} \frac{1}{R} \sum_{r\in\mathcal{R}} \sum_{n\in\mathcal{N}} \sum_{i\in C_n} p_i \omega_{inr} \qquad (o)$$

s.t.

$$\sum_{i\in C_n\cup\{0\}} \omega_{inr} = 1 \qquad \forall n \in \mathcal{N}, r \in \mathcal{R} \quad (\mu_{nr})$$

$$h_{nr} = c_{0nr}\omega_{0nr} + \sum_{i\in C_n} U_{inr}\omega_{inr} \qquad \forall n \in \mathcal{N}, r \in \mathcal{R} \quad (\zeta_{nr})$$

$$h_{nr} \geqslant c_{0nr} \qquad \forall n \in \mathcal{N}, r \in \mathcal{R} \quad (\alpha_{0nr})$$

$$h_{nr} \geqslant U_{inr} \qquad \forall i \in C_n, n \in \mathcal{N}, r \in \mathcal{R} \quad (\alpha_{inr})$$

$$U_{inr} = c_{inr} + \beta_p^{in} p_i \qquad \forall i \in C_n, n \in \mathcal{N}, r \in \mathcal{R} \quad (\kappa_{inr})$$

$$\omega \in \{0,1\}^{(J+1)NR}$$

$$p \in [p_1^L, p_1^U] \times \ldots \times [p_J^L, p_J^U]$$

$$U, h \in \mathbb{R}^{JNR}, \mathbb{R}^{NR}$$

Results BEA

N	R	J	BEA (s)
50	500	3	117167
50	1000	3	259200
50	1500	3	259200
50	2000	3	259200
50	2500	3	259200
50	3000	3	259200
50	3500	3	259200

Breakpoint Exact Algorithm (BEA) (Haering et al., 2023)

Algorithm 1: Breakpoint Exact Algorithm (BEA) to solve the CPP

Result: optimal solution p^* and revenue o^* for CPP.

$p_j^* \leftarrow 0 \quad \forall j \in \{1, \dots, J\}$
$o^* \leftarrow 0$
for s **in** S **do**
    $p_{s_j} \leftarrow 0 \quad \forall j \in \{1, \dots, J\}$
    $h_{nr}^{s_1} \leftarrow c_{0nr} \quad \forall (n, r) \in \mathcal{N} \times \mathcal{R}$
    $\eta_{nr} \leftarrow 0 \quad \forall (n, r) \in \mathcal{N} \times \mathcal{R}$
    $(\hat{p}, \hat{o}) \leftarrow \texttt{enumerate}(s, p, h^{s_1}, \eta, 1)$
    if $\hat{o} > o^*$ **then**
        $p^* \leftarrow \hat{p}$;
        $o^* \leftarrow \hat{o}$;
    end
end

Capacity constraints

$$\omega_{inr} \leqslant y_{inr} \qquad \forall i \in C_n, \in \mathcal{N}, r \in \mathcal{R}$$

$$\sum_{m=1}^{n} \omega_{imr} \leqslant (c_i - 1) y_{inr} + (n-1)(1 - y_{inr}) \qquad \forall i \in C_n, n > c_i \in \mathcal{N}, r \in \mathcal{R}$$

$$\sum_{m=1}^{n} \omega_{imr} \geqslant c_i (1 - y_{inr}) \qquad \forall i \in C_n, n > 1 \in \mathcal{N}, r \in \mathcal{R}$$

TRANSP-OR

EPFL

Compute Objective Value with Priority Queue

```
Function compute_objective_value_with_priority_queue(p, c, prio_queue):
    ς ← (0)_{i∈C}
    for idx ∈ prio_queue do
        u ← [U^i_idx for i ∈ C]
        a ← sort(u, descending)
        φ ← false
        j ← 1
        while j ⩽ C − 1 and !φ do
            if ς_{a_j} ⩽ c_{a_j} − 1 then
                ς_{a_j} += 1
                φ ← true
            end
            else
                j += 1
            end
        end
    end
    o ← Σ_{i∈C} ς_i · p_i
    return o
end
```

Compute Objective Value with Capacities (revenue max/min)

```
Function compute_objective_value_with_capacities(p, c; max):
    s ← sortperm(p)
    ς ← (0)_{i∈C}
    A ← {}
    for idx ∈ 𝒩 × ℛ do
        u ← [U^i_{idx} for i ∈ C]
        a ← sort(u, descending)
        A ← A ∪ {a}
    if max then
        A ← sort(A, ascending)
    else
        A ← sort(A, descending)
    while |A| ⩾ 1 do
        π ← A_{11}
        A ← A \ {A_1}
        if π ⩾ 1 then
            ς_{s_{next_pref}} += 1
            if ς_{s_{next_pref}} = c_{s_{next_pref}} then
                Remove all entries π from A
                if max then
                    A ← sort(A, ascending)
                else
                    A ← sort(A, descending)
    ρ ← Σ_{i∈C} ς_i · p_i
```

Results

Table 9: Test 2: Priority queue vs. Max revenue vs. Robust Optimization

			BEAC		BEAC-M		BEAC-R	
N	R	J	Time (s)	Revenue	Time (s)	Revenue	Time (s)	Revenue
50	2	2	0.43	27.61	0.44	28.81	0.45	27.61
50	5	2	1.72	26.51	1.78	28.44	1.82	26.46
50	10	2	11.42	27.06	12.88	28.3	12.98	27.01
50	25	2	169.08	27.08	197.23	28.58	189.28	27.06
50	50	2	1272.68	26.85	1513.44	28.61	1523.89	26.85
50	100	2	9928.57	26.85	12093.8	28.57	12494.13	26.85
50	250	2	$>$45 hours	$\geqslant$25.00	$>$45 hours	$\geqslant$26.63	$>$45 hours	$\geqslant$24.34

Bibliography I

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

Gilbert, F., Marcotte, P., and Savard, G. (2014). Mixed-logit network pricing. *Computational Optimization and Applications*, 57:105–127.

Haering, T., Legault, R., Torres, F., Ljubic, I., and Bierlaire, M. (2023). Exact algorithms for continuous pricing with advanced discrete choice demand models. Technical Report TRANSP-OR 231211, Transport and Mobility Laboratory, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

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

TRANSP-OR

EPFL

Bibliography II

Korfmann, F. (2018). *Essays on Advanced Discrete Choice Applications*. PhD thesis, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky.

Marandi, A. and Lurkin, V. (2023). An exact algorithm for the static pricing problem under discrete mixed logit demand. *EURO Journal on Computational Optimization*, 11:100073.

Müller, D., Nesterov, Y., and Shikhman, V. (2021). Dynamic pricing under nested logit demand. *arXiv preprint arXiv:2101.04486*.

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

Paneque, M. P., Gendron, B., Azadeh, S. S., and Bierlaire, M. (2022). A lagrangian decomposition scheme for choice-based optimization. *Computers & Operations Research*, 148:105985.

Bibliography III

Shen, Z.-J. M. and Su, X. (2007). Customer behavior modeling in revenue management and auctions: A review and new research opportunities. *Production and operations management*, 16(6):713–728.

van de Geer, R. and den Boer, A. V. (2022). Price optimization under the finite-mixture logit model. *Management Science*, 68(10):7480–7496.

Wu, D., Yin, Y., Lawphongpanich, S., and Yang, H. (2012). Design of more equitable congestion pricing and tradable credit schemes for multimodal transportation networks. *Transportation Research Part B: Methodological*, 46(9):1273–1287.