Fast Algorithms for Capacitated Continuous Pricing with Discrete Choice Demand Models

Tom Haering Fabian Torres Michel Bierlaire

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

2nd EPFL Symposium on Transportation Research, Barcelona
5-7 February 2024

Outline

- Introduction
- Methodology
- Experimental Results
- Conclusions

The Continuous Pricing Problem (CPP)

CPP

- Supplier offers J products for sale. Goal: determine optimal **price** for each product to maximize total **profit**.
- 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 of the individual and **attributes** of the products (e.g. the price).

The Continuous Pricing Problem (CPP)

Utility

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

$$U_{in} = \sum_{k \neq p} \beta_k x_{ink} + \beta_p p_i + \varepsilon_{in}$$

- β_k : parameters (exogenous)
- x_{ink} : attributes (exogenous)
- p_i : price of alternative i
- ε_{in} : stochastic error term

The Continuous Pricing Problem (CPP)

Probability

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

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

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

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

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

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

Literature

Integrating **Logit** into...

- Facility location [Mai and Lodi, 2017, Ljubić and Moreno, 2018]
- Revenue Management [Shen and Su, 2007, Korfmann, 2018]
- Railway Timetabling [Cordone and Redaelli, 2011, Robenek et al., 2018]

Integrating **Nested Logit** into...

- Toll setting [Wu et al., 2012]
- Pricing [Gallego and Wang, 2014]

Integrating **Mixed Logit** into...

- School location [Haase and Müller, 2013]
- Toll setting [Gilbert et al., 2014]
- Pricing [Marandi and Lurkin, 2020, 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]
- **Heuristic** based on **Benders decomposition** (w/out capacity constraints) [Haering et al., 2022]
- **Exact** method based on **spatial Branch-and-Bound** and **Benders decomposition** (w/out capacity constraints) [Haering et al., 2023]

So are we done?

- No. We want to go **faster**. In general. And include **capacity constraints**.

Outline

- Introduction
- Methodology
- Experimental Results
- Conclusions

Base layer: Monte Carlo Simulation

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

$$\begin{aligned} U_{inr} &= \sum_{k \neq p} \beta_k x_{ink} + \beta_p p_i + \varepsilon_{inr} \qquad && \forall n \in \mathcal{N}, i \in C_n, r \in \mathcal{R} \\ &= c_{inr} + \beta_p p_i && \forall n \in \mathcal{N}, i \in C_n, r \in \mathcal{R} \end{aligned}$$

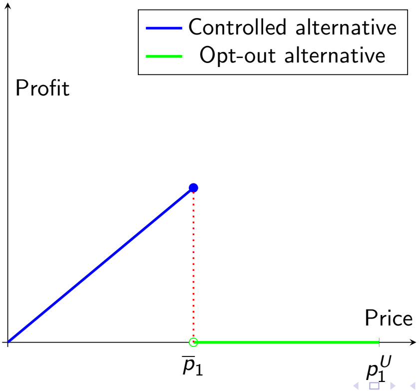
Breakpoints

- **Don't worry**. We're not doing *complicated* decomposition methods or actual *math*.
- Instead we will use the simple idea of decision making **breakpoints**.

Breakpoints: Illustration

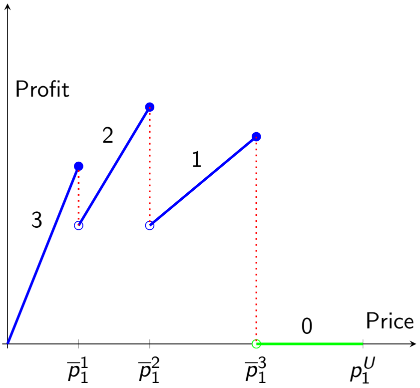
- **1** customer, **1** controlled price + opt-out
- **Breakpoint** $\overline{p}_1$:

$$U_0 = U_1 \implies U_0 = c_1 + \beta_p^1 \overline{p}_1 \implies \overline{p}_1 = \frac{U_0 - c_1}{\beta_p^1}.$$



Breakpoints: Illustration II

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



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

Simplified:

For all possible orderings of prices $p_1 \leq p_2 \leq \cdots \leq p_J$ do:

- Introduce the **cheapest** alternative 1.
- Compute the **breakpoints** $\overline{p}_1^s$ (from $U_{0s} = U_{1s}$) for all simulated customers s.
- **Iterate** over all breakpoints $\overline{p}_1^s$, highest to lowest. For each $\overline{p}_1^s$, fix $p_1 = \overline{p}_1^s$ and do:
  - Introduce alternative **2**. Now compute the **breakpoints** $\overline{p}_2^s$ (from $\max(U_{0s}, U_{1s}) = U_{2s}$).
  - **Iterate** over all breakpoints $\overline{p}_2^s$, highest to lowest. For each $\overline{p}_2^s$, fix $p_2 = \overline{p}_2^s$ and do:
    - Introduce alternative **3**. ...
      - Once **all prices** are **fixed**, **compute profit**

Keep the price combination with **highest profit**!

BEA

- Very fast for one or two alternatives, then it **rapidly** breaks down.
- Complexity $O(J!(NR)^J log(NR))$ **exponential** in number of alternatives J.
- Our new contributions:
  1. Extend BEA to **include capacity constraints**
  2. Introduce Breakpoint **Heuristic** Algorithm (BHA) to solve high-dimensional problems much faster (and with high accuracy!)

Capacity constraints

- Need to compute breakpoints from not only $\max(U_{0s}, \ldots, U_{(i-1)s},) = U_{is}$) but from all $U_{js} = U_{is}$ separately, due to **people no longer always choosing highest utility** alternative.

Breakpoint Exact Algorithm with Capacities (BEAC)

For all possible orderings of prices $\{p_1, p_2, \ldots, p_J\}$ do:

- Introduce the **cheapest** alternative 1.
- Compute the **breakpoints** $\overline{p}_1^s$ (from $U_{0s} = U_{1s}$) for all simulated customers s.
- **Iterate** over all breakpoints $\overline{p}_1^s$. For each $\overline{p}_1^s$, fix $p_1 = \overline{p}_1^s$ and do:
  - Introduce alternative **2**. Now compute the breakpoints $\overline{p}_{02}^s, \overline{p}_{12}^s$ (from $U_{0s} = U_{2s}$ and $U_{1s} = U_{2s}$).
  - **Iterate** over all breakpoints $\overline{p}_{x2}^s$. For each, fix $p_2 = \overline{p}_{x2}^s$ and do:
    - Introduce alternative 3. ...
      - Once **all prices** are **fixed**, **compute capacitated profit**

Keep the price combination with **highest profit**!

Computing the capacitated profit

- Having to compute the profit for each price combination is **computationally expensive, but:**
- It **adds flexibility** as to how we evaluate the profit
- We propose **three** different strategies:
  1. Based on an **exogenous priority queue**
  2. Supplier controls access and assigns people in such a way that **maximizes total profit**
  3. Same as 2. but now we want to minimize profit. Finding the price such that the worst-case profit is highest = **robust optimization**

Breakpoint Heuristic Algorithm (BHA)

1. Choose **starting point** for the heuristic, ex. $p^* = (\frac{p_i^L + p_i^U}{2})_{i \in C}$.
2. $o^* =$ `compute_objective_function`(p^*).
3. Set $j = 1$.
4. **Fix the bounds** to be $= p^*$ **except for alternative** j and solve this simplified problem using the BEA / BEAC. If the resulting $(\hat{p}, \hat{o})$ is better than (p^*, o^*), set $p^* = \hat{p}, o^* = \hat{o}$.
6. Set j = j + 1 and repeat from step 4. If $j = J$ then set $j = 1$.
7. **Terminate** after **no improvement** is found **over** J **iterations**.

... i.e., a **coordinate descent** (ascent).

BHA extended via dynamic line search (DLS)

Idea: escape local optima by adding small controlled perturbations in starting point.

- Base point = solution from BHA.
- **Change one coordinate** by δ and **start BHA from there**.
- If the BHA converges to a better solution than the best known, make it the **new base point**.
- Iterate through all coordinates.
- **Gradually increase the search distance** δ along each direction until a maximal step size is reached.

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).
- Choice model is a **Mixed Logit**, $\beta_{\text{fee}}, \beta_{\text{time\_parking}} \sim \mathcal{N}(\mu, \sigma)$.

Results

Table 1: Test 1: MILP vs. BEAC in the capacitated case

			MILP		BEAC	
N	R	J	Time (s)	Profit	Time (s)	Profit
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	$\geq$25.15	1272.68	26.85
50	100	2	$>$25 hours	$\geq$25.11	9928.57	26.85
50	250	2	$>$45 hours	$\geq$23.45	$>$45 hours	$\geq$25.00

Results

Table 2: Test 3: BHA and DLS vs. B&BD and BEA in the uncapacitated case

			B&BD		BEA		BHA		DLS	
N	R	J	Time (s)	Profit	Time (s)	Profit	Time (s)	Profit	Time (s)	Profit
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 3: Test 4: BHA and DLS vs. MILP and BEAC in the capacitated case

			MILP		BEAC		BHA		DLS	
N	R	J	Time (s)	Profit	Time (s)	Profit	Time (s)	Profit	Time (s)	Profit
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	$\geq$25.15	1272.68	26.85	8.31	26.53	559.04	26.85
50	100	2	$>$25 hours	$\geq$25.11	9928.57	26.85	51.77	26.72	2791.28	26.85
50	250	2	$>$45 hours	$\geq$23.45	$>$45 hours	$\geq$25.00	455.37	26.66	15867.67	26.71
50	10	4	$>$10 hours	$\geq$22.21	$>$10 hours	$\geq$25.41	7.08	26.78	527.34	26.83
50	50	4	$>$20 hours	$\geq$22.19	$>$20 hours	$\geq$27.00	166.21	27.00	7234.88	27.00
50	100	4	$>$45 hours	$\geq$20.50	$>$45 hours	$\geq$24.86	866.97	26.67	34050.57	26.67
50	200	4	$>$72 hours	$\geq$20.32	$>$72 hours	$\geq$24.79	2762.39	26.70	106286.13	26.70

Outline

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

Conclusions

- **Exact method:** BEAC $\approx$20 times faster than MILP
- **Heuristic:** BHA $\approx$100-5000 times faster than MILP with capacities
- **Heuristic:** BHA several orders of magnitudes times faster than MILP and B&BD without capacities
- BHA **optimality gaps** $< 0.2\%$.
- DLS finds **global optimum**, but **too inefficient**. Alternatives should be explored.

Thank you for your attention!

Appendix

Table 4: 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.

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

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

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

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

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

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

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

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

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

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

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

$p_j^* \leftarrow 0 \quad \forall j \in \{1, \ldots, J\}$

$o^* \leftarrow 0$

for s **in** S **do**

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

end

return (p^*, o^*)

Recursive enumeration function within BEA

Algorithm 2: Recursive enumeration function within BEA

Function enumerate(s, p, h^{s_j}, η, j):

$p_{s_j} \leftarrow p^U_{s_j}$

$\bar{p}^{nr}_{s_j} \leftarrow \frac{h^{s_j}_{nr} - c_{s_j nr}}{\beta^{s_j n}_p} \quad \forall (n, r) \in \mathcal{N} \times \mathcal{R}$

$\mathcal{N}_1 \leftarrow \{(n, r) | \bar{p}^{nr}_{s_j} \geq p^U_{s_j}\}$

$\mathcal{N}_2 \leftarrow \{(n, r) | \max\{p^L_{s_j}, p_{s_{j-1}}\} < \bar{p}^{nr}_{s_j} < p^U_{s_j}\}$

$h^{s_{j+1}}_{nr} \leftarrow h^{s_j}_{nr} \quad \forall (n, r) \in \mathcal{N} \times \mathcal{R} \setminus \mathcal{N}_1$

$h^{s_{j+1}}_{nr} \leftarrow c_{s_j nr} + \beta^{s_j n}_p p^U_{s_j} \quad \forall (n, r) \in \mathcal{N}_1$

$\eta_{nr} \leftarrow p^U_{s_j} \quad \forall (n, r) \in \mathcal{N}_1$

$o^* \leftarrow \sum_{n \in \mathcal{N}} \sum_{r \in \mathcal{R}} \eta_{nr}$

$p^* \leftarrow p$

Sort the elements of $\mathcal{N}_2$ so that $\bar{p}^{n_1 r_1}_{s_j} \geq \bar{p}^{n_2 r_2}_{s_j} \geq \cdots \geq \bar{p}^{n_{|\mathcal{N}_2|} r_{|\mathcal{N}_2|}}_{s_j}$

if $j \leq J - 1$ **then**

  for $i \in \{1, \ldots, |\mathcal{N}_2|\}$ **do**

    $p_{s_j} \leftarrow \bar{p}^{n_i r_i}_{s_j}$

    $h^{s_{j+1}}_{nr} \leftarrow c_{s_j nr} + \beta^{s_j n}_p p_{s_j} \quad \forall (n, r) \in \{(n_1, r_1), \ldots, (n_i, r_i)\} \cup \mathcal{N}_1$

    $\eta_{nr} \leftarrow p_{s_j} \quad \forall (n, r) \in \{(n_1, r_1), \ldots, (n_i, r_i)\} \cup \mathcal{N}_1$

    $(\hat{p}, \hat{o}) \leftarrow$ enumerate$(s, p, h^{s_{j+1}}, \eta, j + 1)$

    if $\hat{o} > o^*$ **then**

      $o^* \leftarrow \hat{o}$

      $p^* \leftarrow \hat{p}$

    end

  end

end

else

  $\tilde{o} \leftarrow o^* - \sum_{(n,r) \in \mathcal{N}_1} \eta_{nr}$

  for $i \in \{1, \ldots, |\mathcal{N}_2|\}$ **do**

    $\tilde{o} \leftarrow \tilde{o} - \eta_{n_i r_i}$

    $p_{s_j} \leftarrow \bar{p}^{n_i r_i}_{s_j}$

    $o \leftarrow \tilde{o} + (|\mathcal{N}_1| + i) p_{s_j}$

    if $o > o^*$ **then**

      $o^* \leftarrow o$

      $p^* \leftarrow p$

    end

  end

  return (p^*, o^*)

end

end

Capacity constraints

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

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

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

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 (profit max/min)

Function `compute_objective_value_with_capacities`($p, c; max$):
- $s \leftarrow \text{sortperm}(p)$
- $\varsigma \leftarrow (0)_{i \in C}$
- $A \leftarrow \{\}$
- **for** $idx \in \mathcal{N} \times \mathcal{R}$ **do**
  - $u \leftarrow [U^i_{idx} \text{ for } i \in C]$
  - $a \leftarrow \text{sort}(u, \text{descending})$
  - $A \leftarrow A \cup \{a\}$
- **if** max **then**
  - $A \leftarrow \text{sort}(A, \text{ascending})$
- **else**
  - $A \leftarrow \text{sort}(A, \text{descending})$
- **while** $|A| \geq 1$ **do**
  - $\pi \leftarrow A_{11}$
  - $A \leftarrow A \setminus \{A_1\}$
  - **if** $\pi \geq 1$ **then**
    - $\varsigma_{s_{next\_pref}} \mathrel{+}= 1$
    - **if** $\varsigma_{s_{next\_pref}} = c_{s_{next\_pref}}$ **then**
      - Remove all entries π from A
      - **if** max **then**
        - $A \leftarrow \text{sort}(A, \text{ascending})$
      - **else**
        - $A \leftarrow \text{sort}(A, \text{descending})$
- $o \leftarrow \sum_{i \in C} \varsigma_i \cdot p_i$
- **return** o

Results

Table 5: Test 2: Priority queue vs. Max profit vs. Robust Optimization

			BEAC		BEAC-M		BEAC-R	
N	R	J	Time (s)	Profit	Time (s)	Profit	Time (s)	Profit
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	$\geq$25.00	$>$45 hours	$\geq$26.63	$>$45 hours	$\geq$24.34

Bibliography I

Cordone, R. and Redaelli, F. (2011).
Optimizing the demand captured by a railway system with a regular timetable.
Transportation Research Part B: Methodological, 45(2):430–446.

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.

Bibliography II

Haase, K. and Müller, S. (2013).
Management of school locations allowing for free school choice.
Omega, 41(5):847–855.

Haering, T., Bongiovanni, C., and Bierlaire, M. (2022).
A benders decomposition for maximum simulated likelihood estimation of advanced discrete choice models.
In *Proceedings of the 22nd Swiss Transport Research Conference (STRC)*.

Haering, T., Legault, R., Torres, F., Ljubic, I., and Bierlaire, M. (2023).
Exact algorithms for continuous pricing with advanced discrete choice demand models.

TRANSP-OR

EPFL

Bibliography III

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.

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

Ljubić, I. and Moreno, E. (2018).
Outer approximation and submodular cuts for maximum capture facility location problems with random utilities.
European Journal of Operational Research, 266(1):46–56.

TRANSP-OR

EPFL

Bibliography IV

Mai, T. and Lodi, A. (2017).
Solving large-scale competitive facility location under random utility maximization models.
In *Technical Report*. CERC, Polytechnique Montréal, Canada.

Marandi, A. and Lurkin, V. (2020).
An exact algorithm for the static pricing problem under discrete mixed logit demand.
arXiv preprint arXiv:2005.07482.

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.

Bibliography V

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.

Robenek, T., Azadeh, S. S., Maknoon, Y., de Lapparent, M., and Bierlaire, M. (2018).
Train timetable design under elastic passenger demand.
Transportation research Part b: methodological, 111:19–38.

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.

TRANSP-OR

EPFL

Bibliography VI

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.