A Lagrangian relaxation technique for the demand-based benefit maximization problem

Meritxell Pacheco Paneque
Bernard Gendron  Virginie Lurkin
Michel Bierlaire  Shadi Sharif Azadeh

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

06/09/2018
Outline

1. Introduction

2. Demand-based benefit maximization problem

3. Lagrangian relaxation

4. Preliminary results

5. Conclusions and future work
Introduction

Demand-based benefit maximization problem

Lagrangian relaxation

Preliminary results

Conclusions and future work
Discrete choice models and optimization

- Disaggregate demand modeling
- Behavioral realism
- Complex formulations

- Supply decisions
- Linearity and/or convexity
- MILP models
Bridging the gap

- Linear characterization of a discrete choice model
- Simulation to address stochasticity
- Demand-based benefit maximization problem (MILP example)
- General framework that can be applied with an existing choice model
1 Introduction

2 Demand-based benefit maximization problem

3 Lagrangian relaxation

4 Preliminary results

5 Conclusions and future work
Demand-based benefit maximization problem

Linearization of the discrete choice model

Choice set $\mathcal{C}(i)$

Population $N(n)$

$U_{in} = V_{in} + \varepsilon_{in}$  

draw distribution ($R$)  

$U_{inr} = V_{in} + \xi_{inr}$
Demand-based benefit maximization problem

Linearization of the discrete choice model

Choice set $\mathcal{C}(i)$

Population $N(n)$

$$U_{in} = V_{in} + \varepsilon_{in}$$

draw distribution $(R)$

$$U_{inr} = V_{in} + \xi_{inr}$$
Linearization of the discrete choice model

Choice set $C(i)$

Population $N(n)$

$$U_{in} = V_{in} + \epsilon_{in}$$

draw distribution $(R)$

$$U_{inr} = V_{in} + \xi_{inr}$$
Demand representation

Choice $w_{inr}$

$$w_{inr} = \begin{cases} 
1 & \text{if } U_{inr} \geq U_{j,n}, \forall j \in \mathcal{C}_n, j \neq i \\
0 & \text{otherwise}
\end{cases}$$

$$D_i = \frac{1}{R} \sum_r \sum_n w_{inr}$$
Demand-based benefit maximization problem

Demand representation

Choice $w_{inr}$

$$w_{inr} = \begin{cases} 
1 & \text{if } U_{inr} \geq U_{jnr}, \forall j \in C_n, j \neq i \\
0 & \text{otherwise}
\end{cases}$$

$$D_i = \frac{1}{R} \sum_r \sum_n w_{inr}$$
Demand-based benefit maximization problem

Demand representation

Choice $w_{inr}$

$w_{inr} = \begin{cases} 
1 & \text{if } U_{inr} \geq U_{jnr}, \ \forall j \in C_n, j \neq i \\
0 & \text{otherwise} 
\end{cases}$

$D_i = \frac{1}{R} \sum_r \sum_n w_{inr}$
Set of alternatives $C \ (i > 0)$  
Opt-out option $i = 0$  
Population $N \ (n \geq 1)$  
Price $a_{in} \leq p_{in} \leq b_{in}$  
Capacity levels $c_{iq} \ (Q \text{ levels, each with a certain cost})$
### Benefit maximization problem (2)

| obj. fun. | $\sum_{i>0} \text{Revenue}_i - \text{Cost}_i$ |
| availability | operator level and scenario level |
| disc. utility | variable capturing availability and utility |
| choice | linearization of the highest utility |
| price | linearization of the product $w_{inr} p_{in}$ (revenue) |
| capacity | relation with the availability at scenario level |
Computational results

- Parking choices: mixtures of logit model
- Distributed parameters (and correlated)
- $R = 50$ draws and $N = 50$ customers
- $|C| = 3$: PSP, PUP and FSP (opt-out)

- Several experiments
  - Price calibration (discrete and continuous prices)
  - Price differentiation by population segmentation

- Computational times up to 34 hours!
1 Introduction

2 Demand-based benefit maximization problem

3 Lagrangian relaxation

4 Preliminary results

5 Conclusions and future work
Motivation

Customer \( (n) \)
- Maximization of own utility
- Objective function and capacity constraints

Draw \( (r) \)
- Behavioral scenario
- Objective function
Why Lagrangian relaxation?

Lagrangian relaxation

obj. fun. + $\alpha$ hard constraints

easy constraints

hard constraints

Lagrangian decomposition

obj. fun. $x = y$

obj. fun. (x)

constraints (x)

obj. fun. (y)

constraints (y)
Revenue maximization + unlimited capacity

**obj. fun.**

\[ \sum_{i>0} \text{Revenue}_i \]

**availability**

no need for discounted utility (no availability)

**utility**

linearization of the highest utility

**choice**

**price**

linearization of the product \( w_{inr}p_{in} \) (revenue)

**capacity**
Lagrangian decomposition

⚠️ Price $p_{in}$ is the same across draws ⇒ **no** decomposition by $n$ and $r$

\[ p_{in1} = p_{in2} = \cdots = p_{inR} = p_{in1} \]

$p_{inr} - p_{in(r+1)} = 0$ ⇒ Lagrangian multipliers $\alpha_{inr} \Rightarrow$ decomposition by $n$ and $r$
Objective function:
- Price of the chosen alternative
- Lagrangian term: \((\alpha_{inr} - \alpha_{in(r-1)})p_{inr}\)

- One alternative is chosen (based on the highest utility)
- Price of the chosen alternative specific for the draw
Subgradient method (1)

1. **Initial values**
   - Lag. mult.

2. **Solve subproblems**

3. **Calculate step and direction**

4. **Update Lag. multipliers**

The process iterates between solving subproblems and updating Lagrange multipliers.
Subgradient method (2)

**Input:** UB: \(Z(\bar{\alpha})\) with \(\bar{\alpha}\) starting values, LB: \(Z^*\) (from a feasible solution)

1. while \(k < K\) or \(Z(\alpha(k))\) has not improved after some iterations do
2.   for \(r = 1 \ldots R\) do
3.     for \(n = 1 \ldots N\) do
4.       Lagrangian subproblem \(Z_{nr}(\alpha(k))\) (MILP);
5.       Obtain \(p_{inr}\) and \(Z_{nr} (\alpha(k))\);
6.     end
7.   end
8.   Compute \(Z(\alpha(k)) = \sum_r \sum_n Z_{nr}(\alpha(k))\);
9.   \(k \leftarrow k + 1\);
10.  Obtain \(\omega(k)\) (step) and \(d_{inr}(k)\) (direction);
11.  Update the Lagrangian multipliers: \(\alpha_{inr}(k) = \alpha_{inr}(k - 1) - \omega(k)d_{inr}(k)\)
12. end
1 Introduction

2 Demand-based benefit maximization problem

3 Lagrangian relaxation

4 Preliminary results

5 Conclusions and future work
Case study

- $N = 20$ and $R = 100$
- Price bounds PSP: $[0.5, 1.0]$
- Price bounds PUP: $[0.7, 1.2]$
- Number of iterations: $K = 250$
Evolution bounds

Computational time:
- Exact method: 32 min
- Subgradient method: 5.9 min (1.4 s/it)

Objective function:
- MILP: 11.0773
- LP relaxation: 21.4114
Valid inequalities

- Same optimal solution for the MILP
- Tighter formulation for the LP relaxation

\[ \sum_i U_{irn} w_{ir} \geq U_{jnr} \quad \forall j, n, r \]
Preliminary results

Evolution bounds (with valid inequality)

Computational time:

- Exact method: **11 min**
- Subgradient method: **34 min (7 s/it)**

Objective function:

- MILP: **11.0773**
- LP relaxation: **14.1934**
Observations

- Cheap iterations of the subgradient method
- LB provides a feasible solution
- Valid inequality
  - It helps to strength the LP bound (21.41 vs 14.19)
  - More expensive iterations but a smaller amount gives better bounds
1 Introduction

2 Demand-based benefit maximization problem

3 Lagrangian relaxation

4 Preliminary results

5 Conclusions and future work
Conclusions

- Efficient method to obtain lower and upper bounds + feasible solution
- Different configurations of parameters might help
- Valid inequalities
Future work

- Define other techniques to generate feasible solutions (LB)
- Try other valid inequalities: \((U_{nr} - U_{inr})w_{inr}\) in the objective function
  - \(w_{inr} = 0 \Rightarrow \text{term vanishes}\)
  - \(w_{inr} = 1 \Rightarrow U_{nr} = U_{inr} \Rightarrow \text{term vanishes}\)
- Evaluate other strategies (e.g., regularization term)
- Gradually include the complexity back (capacity, availability...)

MPP, BG, VL, MB, SSA
Questions?

meritxell.pacheco@epfl.ch
Subgradient method: step size and direction

\[ \alpha_{inr}(k) = \alpha_{inr}(k - 1) - \omega(k)d_{inr}(k) \]

**Step:**
- \( \omega(k) = \lambda(k) \frac{Z(\alpha(k-1)) - Z^*}{\|\gamma(k)\|^2} \)
- \( \lambda(0) \in [0, 2) \)
- \( \gamma_{inr}(k) = p_{inr}(k) - p_{in(r-1)}(k) \) (subgradients)
- \( \lambda(k) \) divided by \( \omega_1 \) if \( Z(\alpha(k)) \) has not improved in \( \omega_2 \) iterations

**Direction:**
- \( d(k) = \gamma(k) + \theta d(k - 1) \)
- \( \theta \in [0, 1) \)