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

14/09/2018

Outline

1. Introduction
2. Demand-based benefit maximization problem
3. Lagrangian relaxation
4. Preliminary results
5. Conclusions and future work

Demand vs. supply

- Discrete choice models
- Disaggregate demand modeling
- Behavioral realism
- Complex formulations

Demand vs. supply

- MILP models
- Supply decisions
- Tractability of the formulations
- Linearity and/or convexity

Demand vs. supply

- Discrete choice models
- Disaggregate demand modeling
- Behavioral realism
- Complex formulations

- MILP models
- Supply decisions
- Tractability of the formulations
- Linearity and/or convexity

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Bridging the gap

- Linear characterization of a discrete choice model
  - Simulation to address stochasticity
  - Direct usage of the utility variables (instead of the choice probabilities)
- Demand-based benefit maximization problem (MILP example)
- General framework that can be applied with an existing choice model
- Computationally expensive: need for decomposition techniques

Linearization of the discrete choice model

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

Population N (n)

$U_{in} = V_{in} + \varepsilon_{in}$ $\xrightarrow{\text{draw distribution } (R)}$ $U_{inr} = V_{in} + \xi_{inr}$

TRANSP-OR

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Linearization of the discrete choice model

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

Population N (n)

$$U_{in} = V_{in} + \varepsilon_{in} \xrightarrow{\text{draw distribution } (R)} U_{inr} = V_{in} + \xi_{inr}$$

Linearization of the discrete choice model

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

Population N (n)

$$U_{in} = V_{in} + \varepsilon_{in} \xrightarrow{\text{draw distribution } (R)} U_{inr} = V_{in} + \xi_{inr}$$

Demand representation

Demand is not obtained from the choice probabilities!

Choice w_{inr}

$$w_{inr} = \begin{cases} 1 & \text{if } U_{inr} \geq U_{jnr}, \forall j \in \mathscr{C}_n, j \neq i \\ 0 & \text{otherwise} \end{cases} \xrightarrow{\text{demand}} D_i = \frac{1}{R} \sum_r \sum_n w_{inr}$$

Demand representation

⚠ Demand is not obtained from the choice probabilities!

Choice w_{inr}

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

demand

$$D_i = \frac{1}{R} \sum_r \sum_n w_{inr}$$

TRANSP-OR

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Demand representation

⚠ Demand is not obtained from the choice probabilities!

Choice w_{inr}

$$w_{inr} = \begin{cases} 1 & \text{if } U_{inr} \geq U_{jnr}, \forall j \in \mathscr{C}_n, j \neq i \\ 0 & \text{otherwise} \end{cases} \xrightarrow{\text{demand}} D_i = \frac{1}{R} \sum_r \sum_n w_{inr}$$

Benefit maximization problem (1)

- Set of alternatives $\mathscr{C}$ $(i > 0)$
- Opt-out option $i = 0$
- Population N $(n \geq 1)$
- Price $a_{in} \leq p_{in} \leq b_{in}$
- Capacity levels c_{iq} (Q 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
- $|\mathscr{C}| = 3$: 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!

Motivation: why decomposition techniques?

Customer (n)

- Utility maximization
- Objective function and capacity constraints

Draw (r)

- Behavioral scenario
- Objective function

Lagrangian relaxation

obj. fun.	obj. fun.	$+\alpha$	hard constraints
easy constraints	easy constraints		
hard constraints			

Lagrangian decomposition

obj. fun.		obj. fun. (x)		obj. fun. (y)
	$x = y$		$+$	
constraints		constraints (x)		constraints (y)

Lagrangian relaxation

obj. fun.		obj. fun.	$+\alpha$	hard constraints
easy constraints		easy constraints		
hard constraints				

Lagrangian decomposition

obj. fun.		obj. fun. (x)		obj. fun. (y)
	$x = y$		$+$	
constraints		constraints (x)		constraints (y)

The problem now: revenue maximization & infinite capacity

obj. fun.	$\sum_{i>0} \text{Revenue}_i$
availability	
utility	no need for discounted utility (no availability)
choice	linearization of the highest utility
price	linearization of the product $w_{inr} p_{in}$ (revenue)
capacity	

Lagrangian decomposition

Idea: decompose the problem in subproblems per customer and draw

⚠ 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 α_{inr} ⇒ decomposition by n and r

Lagrangian decomposition

Idea: decompose the problem in subproblems per customer and draw

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

$$p_{in1} = p_{in2} = \cdots = p_{inR} = p_{in1}$$

$p_{inr} - p_{in(r+1)} = 0 \Rightarrow$ Lagrangian multipliers $\alpha_{inr} \Rightarrow$ decomposition by n and r

Lagrangian decomposition

Idea: decompose the problem in subproblems per customer and draw

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

$$p_{in1} = p_{in2} = \cdots = p_{inR} = p_{in1}$$

$p_{inr} - p_{in(r+1)} = 0 \Rightarrow$ Lagrangian multipliers $\alpha_{inr} \Rightarrow$ decomposition by n and r

Lagrangian decomposition

Idea: decompose the problem in subproblems per customer and draw

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

$$p_{in1} = p_{in2} = \cdots = p_{inR} = p_{in1}$$

$p_{inr} - p_{in(r+1)} = 0 \Rightarrow$ Lagrangian multipliers $\alpha_{inr} \Rightarrow$ decomposition by n and r

Lagrangian subproblems

- One subproblem per customer and draw
- Objective function:
  - Revenue term: $p_{inr} w_{inr}$ (linearized)
  - Lagrangian term: $(\alpha_{inr} - \alpha_{in(r-1)}) p_{inr}$
- Constraints:
  - Utility maximization
  - One alternative is chosen
  - Linearization revenue term

Subgradient method

What are the values of the Lagrangian multipliers (α_{inr})?

initial values Lag. mult. → solve subproblems → calculate step and direction → update Lag. multipliers

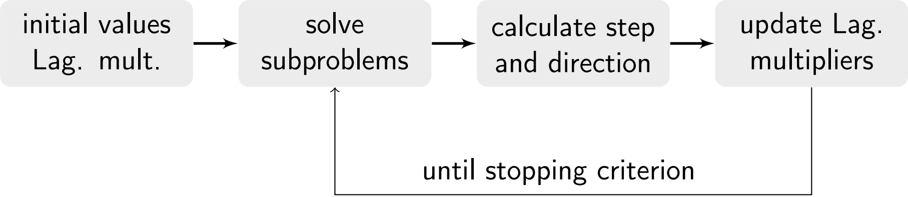
until stopping criterion

At every iteration:

- UB: from the subgradient method
- LB: from a feasible solution (based on the subgradient method)

Subgradient method

What are the values of the Lagrangian multipliers (α_{inr})?

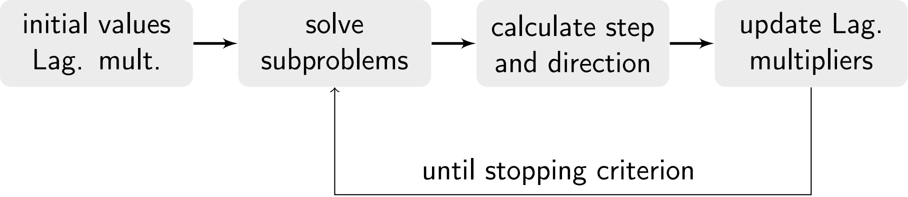


At every iteration:

- UB: from the subgradient method
- LB: from a feasible solution (based on the subgradient method)

Subgradient method

What are the values of the Lagrangian multipliers (α_{inr})?



At every iteration:

- UB: from the subgradient method
- LB: from a feasible solution (based on the subgradient method)

Case study

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

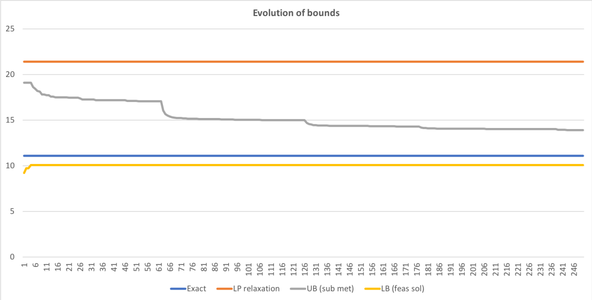
Evolution bounds

Computational time:

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

Objective function:

- MILP: 11.08
- LP relaxation: 21.41



Evolution bounds (with valid inequality)

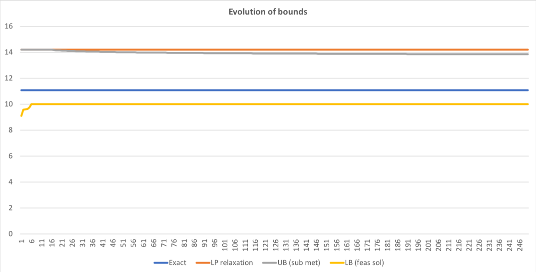
$$\sum_i U_{inr} w_{inr} \geq U_{jnr} \ \forall j, n, r$$

Computational time:

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

Objective function:

- MILP: 11.08
- LP relaxation: 14.19



Conclusions

- Efficient method to obtain LB and UB + feasible solution
- Valid inequalities help to tighten the solution space
- Need for convergence: LB and UB sufficiently close

Future work

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

Questions?

meritxell.pacheco@epfl.ch

Subgradient method: algorithm

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

while $k < K$ *or* $Z(\alpha(k))$ *has not improved after some iterations* **do**

    for $r = 1 \dots R$ **do**

        for $n = 1 \dots N$ **do**

            Lagrangian subproblem $Z_{nr}(\alpha(k))$ (MILP);

            Obtain p_{inr} and $Z_{nr}(\alpha(k))$;

        end

    end

    Compute $Z(\alpha(k)) = \sum_r \sum_n Z_{nr}(\alpha(k))$;

    $k \leftarrow k + 1$;

    Obtain $\omega(k)$ (**step**) and $d_{inr}(k)$ (**direction**);

    Update the **Lagrangian multipliers**: $\alpha_{inr}(k) = \alpha_{inr}(k-1) - \omega(k) d_{inr}(k)$

end

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 ω_1 if $Z(\alpha(k))$ has not improved in ω_2 iterations

Direction:

- $d(k) = \gamma(k) + \theta d(k-1)$
- $\theta \in [0,1)$