Discrete choice models and heuristics for global nonlinear optimization

Michel Bierlaire

Transport and Mobility Laboratory, Ecole Polytechnique Fédérale de Lausanne
Introduction

• Econometrics
  • Discrete choice models
  • Recent development in random utility models

• Operations Research
  • Nonlinear optimization
  • Global optimum for non convex functions
Random utility models

- Choice model:
  \[ P(i|C_n) \text{ where } C_n = \{1, \ldots, J\} \]

- Random utility:
  \[ U_{in} = V_{in} + \varepsilon_{in} \]

  and

  \[ P(i|C_n) = P(U_{in} \geq U_{jn}, j = 1, \ldots, J) \]

- Utility is a latent concept
Multinomial Logit Model

- Assumption: $\varepsilon_{in}$ are i.i.d. Extreme Value distributed.
- Independence is both across $i$ and $n$
- Choice model:

$$P(i|C_n) = \frac{e^{V_{in}}}{\sum_{j \in C_n} e^{V_{jn}}}$$
Relaxing the independence assumption

...across alternatives

\[
\begin{pmatrix}
U_{1n} \\
\vdots \\
U_{Jn}
\end{pmatrix} =
\begin{pmatrix}
V_{1n} \\
\vdots \\
V_{Jn}
\end{pmatrix} +
\begin{pmatrix}
\varepsilon_{1n} \\
\vdots \\
\varepsilon_{Jn}
\end{pmatrix}
\]

that is

\[U_n = V_n + \varepsilon_n\]

and \(\varepsilon_n\) is a vector of random variables.
Relaxing the independence assumption

- $\varepsilon_n \sim \mathcal{N}(0, \Sigma)$: multinomial probit model
  - No closed form for the multifold integral
  - Numerical integration is computationally infeasible

- Extensions of multinomial logit model
  - Nested logit model
  - Multivariate Extreme Value (MEV) models
MEV models

Family of models proposed by McFadden (1978)

Idea: a model is generated by a function

\[ G : \mathbb{R}^J \rightarrow \mathbb{R} \]

From \( G \), we can build

- The cumulative distribution function (CDF) of \( \varepsilon_n \)
- The probability model
- The expected maximum utility

Called Generalized EV models in DCM community
MEV models

1. $G$ is homogeneous of degree $\mu > 0$, that is
   \[ G(\alpha x) = \alpha^\mu G(x) \]

2. \[ \lim_{x_i \to +\infty} G(x_1, \ldots, x_i, \ldots, x_J) = +\infty, \forall i, \]

3. the $k$th partial derivative with respect to $k$ distinct $x_i$ is non negative if $k$ is odd and non positive if $k$ is even, i.e., for all (distinct) indices $i_1, \ldots, i_k \in \{1, \ldots, J\}$, we have
   \[ (-1)^k \frac{\partial^k G}{\partial x_{i_1} \ldots \partial x_{i_k}}(x) \leq 0, \forall x \in \mathbb{R}^J_+. \]
MEV models

- Cumulative distribution function:
  \[ F(\varepsilon_1, \ldots, \varepsilon_J) = e^{-G(e^{-\varepsilon_1}, \ldots, e^{-\varepsilon_J})} \]

- Probability:
  \[ P(i|C) = \frac{e^{V_i + \ln G_i(e^{V_1}, \ldots, e^{V_J})}}{\sum_{j\in C} e^{V_j + \ln G_j(e^{V_1}, \ldots, e^{V_J})}} \]
  with \( G_i = \frac{\partial G}{\partial x_i} \). This is a closed form

- Expected maximum utility:
  \[ V_C = \frac{\ln G(\cdot) + \gamma}{\mu} \]
  where \( \gamma \) is Euler’s constant.

- Note:
  \[ P(i|C') = \frac{\partial V_C}{\partial V_i}. \]
MEV models

Example: Multinomial logit:

\[ G(e^{V_1}, \ldots, e^{V_J}) = \sum_{i=1}^{J} e^{\mu V_i} \]
MEV models

Example: Nested logit

\[ G(y) = \sum_{m=1}^{M} \left( \sum_{i=1}^{J_m} y_{i}^{\mu m} \right)^{\frac{\mu}{\mu m}} \]

Example: Cross-Nested Logit

\[ G(y_1, \ldots, y_J) = \sum_{m=1}^{M} \left( \sum_{j \in C} (\alpha_{jm}^{1/\mu} y_j)^{\mu m} \right)^{\frac{\mu}{\mu m}} \]
Nested Logit Model

Discrete choice models and heuristics for global nonlinear optimization – p.12/52
Nested Logit Model

- Motorized
  - Bus
  - Train
- Unmotorized
  - Car
  - Ped.
  - Bike
Cross-Nested Logit Model

Nest 1

Bus
Train

Nest 2

Car
Ped.
Bike
MEV models

Issues:

- Formulation not in term of correlations
  Abbe, Bierlaire & Toledo (2005)
- Require heavy proofs
  Daly & Bierlaire (2006)
- Homoscedasticity
  McFadden & Train (2000)
- Sampling issues
  Bierlaire, Bolduc & McFadden (2006)
Sampling issue

- Sampling is never random in practice
- Choice-based samples are convenient in transportation analysis
- Estimation is an issue

Main references:
- Manski and Lerman (1977)
- Manski and McFadden (1981)
- Cosslett (1981)
- Ben-Akiva and Lerman (1985)
Sampling issues

Main result:

- Estimator for random samples is valid of exogenous samples
- It is both consistent and efficient
- If observations are weighted, it becomes inefficient

Exogenous Sample Maximum Likelihood (ESML)
Sampling issue: estimation

Conditional Maximum Likelihood (CML) Estimator

\[
\max_{\theta} \mathcal{L}(\theta) = \sum_{n=1}^{N} \ln \Pr(i_n|x_n, s, \theta)
\]

\[
= \sum_{n=1}^{N} \ln \frac{R(i_n, x_n, \theta) P(i_n|x_n, \theta)}{\sum_{j \in C_n} R(j, x_n, \theta) P(j|x_n, \theta)}
\]

where \( R(i, x, \theta) = \Pr(s|i, x, \theta) \) is the probability that a population member with configuration \((i, x)\) is sampled
Estimation of MEV models

The main term in the CML formulation is:

\[
\frac{R(i, x, \theta) P(i|x, \theta)}{\sum_{j \in C} R(j, x, \theta) P(j|x, \theta)} = e^{V_i + \ln G_i(\cdot) + \ln R(i, x, \theta)} \frac{e^{V_j + \ln G_j(\cdot) + \ln R(j, x, \theta)}}{\sum_{j \in C} e^{V_j + \ln G_j(\cdot) + \ln R(j, x, \theta)}}.
\]

where index \( n \) has been dropped.
Estimation of MEV models

- Case of MNL model: $G_i = 0$ when $\mu = 1$.

$$
\frac{R(i, x, \theta)P(i|x, \theta)}{\sum_{j \in C} R(j, x, \theta)P(j|x, \theta)} = \frac{e^{V_i + \ln R(i, x, \theta)}}{\sum_{j \in C} e^{V_j + \ln R(j, x, \theta)}}.
$$

- Well-known result: if ESML is used, only constants are biased

- Indeed, $V_i = \sum_k \beta_k x_k + c_i$

- Question: does this generalize to all MEV?

- Answer: NO
Estimation of MEV models

- The $V$’s are shifted in the main formula

$$e^{V_i} + \ln G_i(\cdot) + \ln R(i,x,\theta)$$

$$\sum_{j \in C} e^{V_j} + \ln G_j(\cdot) + \ln R(j,x,\theta).$$

- ... but not in the $G_i$

$$G_i(\cdot) = \frac{\partial G}{\partial e^{V_i}} (e^{V_1}, \ldots, e^{V_J}).$$

- ESML will not produce consistent estimates on non-MNL MEV models.
Estimation of MEV models

\[ e^{V_i + \ln G_i(\cdot)} + \ln R(i, x, \theta) \]
\[ \sum_{j \in C} e^{V_j + \ln G_j(\cdot)} + \ln R(j, x, \theta) \cdot \]

- New idea: estimate \( \ln R(i, x, \theta) \) from data
- Cannot be done with classical software
- But easy to implement due to the MNL-like form
- Available in BIOGEME, an open source freeware for the estimation of random utility models:
  
  biogeme.epfl.ch
Reference


[transp-or.epfl.ch](http://transp-or.epfl.ch)
Global optimization

Motivation:

- (Conditional) Maximum Likelihood estimation of MEV models
- More advanced models:
  - continuous and discrete mixtures of MEV models
  - estimation with panel data
  - latent classes
  - latent variables
  - discrete-continuous models
  - etc...
Global optimization

Objective: identify the global minimum of

$$\min_{x \in \mathbb{R}^n} f(x),$$

where

- $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable.
- No special structure is assumed on $f$. 

Discrete choice models and heuristics for global nonlinear optimization – p.25/52
Literature

Local nonlinear optimization:

- Main focus:
  - global convergence
  - towards a local minimum
  - with fast local convergence.

- Vast literature
- Efficient algorithms
- Softwares
Global nonlinear optimization: exact approaches

- Real algebraic geometry (representation of polynomials, semidefinite programming)
- Interval arithmetic
- Branch & Bound
- DC - difference of convex functions
Literature

Global nonlinear optimization: heuristics

- Usually hybrid between derivative-free methods and heuristics from discrete optimization. Examples:
  - Glover (1994) Tabu + scatter search
  - Franze and Speciale (2001) Tabu + pattern search
  - Hedar and Fukushima (2006) Tabu + direct search
  - Mladenovic et al. (2006) Variable Neighborhood search (VNS)
Our heuristic

Framework: VNS
Ingredients:

1. Local search

\[(\text{SUCCESS}, y^*) \leftarrow \text{LS}(y_1, \ell_{\text{max}}, \mathcal{L}),\]

where

- \(y_1\) is the starting point
- \(\ell_{\text{max}}\) is the maximum number of iterations
- \(\mathcal{L}\) is the set of already visited local optima
- Algorithm: trust region
Our heuristic

1. Local search

\[(\text{SUCCESS}, y^*) \leftarrow \text{LS}(y_1, \ell_{\text{max}}, \mathcal{L}),\]

- If \(\mathcal{L} \neq \emptyset\), LS may be interrupted prematurely
- If \(\mathcal{L} = \emptyset\), LS runs toward convergence
- If local minimum identified, SUCCESS=true
Our heuristic

2. Neighborhood structure
   - Neighborhoods: $\mathcal{N}_k(x)$, $k = 1, \ldots, n_{\text{max}}$
   - Nested structure: $\mathcal{N}_k(x) \subset \mathcal{N}_{k+1}(x) \subseteq \mathbb{R}^n$, for each $k$
   - Neighbors generation
     $$(z_1, z_2, \ldots, z_p) = \text{NEIGHBORS}(x, k).$$
   - Typically, $n_{\text{max}} = 5$ and $p = 5$. 
The VNS framework

**Initialization** \( x_1^* \) local minimum of \( f \)
- Cold start: run LS once
- Warm start: run LS from randomly generated starting points

**Stopping criteria** Interrupt if
1. \( k > n_{\text{max}} \): the last neighborhood has been unsuccessfully investigated
2. CPU time \( \geq t_{\text{max}} \), typ. 30 minutes (18K seconds).
3. Number of function evaluations \( \geq \text{eval}_{\text{max}} \), typ. \( 10^5 \).
The VNS framework

Main loop Steps:

1. Generate neighbors of $x_{\text{best}}^k$:

   $$(z_1, z_2, \ldots, z_p) = \text{NEIGHBORS}(x_{\text{best}}^k, k).$$

   (1)

2. Apply the $p$ local search procedures:

   $$(\text{SUCCESS}_j, y^*_j) \leftarrow \text{LS}(z_j, \ell_{\text{large}}, \mathcal{L}).$$

   (2)

3. If SUCCESS$_j$ = FALSE, for $j = 1, \ldots, p$, we set $k = k + 1$ and proceed to the next iteration.
The VNS framework

Main loop  Steps (ctd):

4. Otherwise,

\[ \mathcal{L} = \mathcal{L} \cup \{ y_j^* \}. \]  \hspace{1cm} (3)

for each \( j \) such that SUCCESS\( j \) = TRUE

5. Define \( x_{k+1}\)\text{best}

\[ f(x_{k+1}\text{best}) \leq f(x), \text{ for each } x \in \mathcal{L}. \]  \hspace{1cm} (4)

6. If \( x_{k+1}\)\text{best} = \( x_k\)\text{best}, no improvement. We set \( k = k + 1 \) and proceed to the next iteration.
The VNS framework

Main loop Steps (ctd):

7. Otherwise, we have found a new candidate for the global optimum. The neighborhood structure is reset, we set $k = 1$ and proceed to the next iteration.

Output The output is the best solution found during the algorithm, that is $x^k_{\text{best}}$. 
Local search

- Classical trust region method with quasi-newton update
- Key feature: premature interruption
- Three criteria: we check that
  1. the algorithm does not get too close to an already identified local minimum.
  2. the gradient norm is not too small when the value of the objective function is far from the best.
  3. a significant reduction in the objective function is achieved.
Neighborhoods

The key idea: analyze the curvature of $f$ at $x$

- Let $v_1, \ldots, v_n$ be the (normalized) eigenvectors of $H$
- Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues.
- Define direction $w_1, \ldots, w_{2n}$, where $w_i = v_i$ if $i \leq n$, and $w_i = -v_i$ otherwise.
- Size of the neighborhood: $d_1 = 1$, $d_k = 1.5d_{k-1}$, $k = 2, \ldots$
Neighborhoods

- Neighbors:

\[ z_j = x + \alpha d_k w_i, \quad j = 1, \ldots, p, \] (5)

where
- \( \alpha \) is randomly drawn \( U[0.75, 1] \)
- \( i \) is a selected index

- Selection of \( w_i \):
  - Prefer directions where the curvature is larger
  - Motivation: better potential to jump in the next valley
Neighborhoods: selection of $w_i$

$$P(w_i) = P(-w_i) = \frac{e^{\beta \frac{|\lambda_i|}{d_k}}}{2 \sum_{j=1}^{n} e^{\beta \frac{|\lambda_j|}{d_k}}}.$$ 

- In large neighborhoods ($d_k$ large), curvature is less relevant and probabilities are more balanced.
- We tried $\beta = 0.05$ and $\beta = 0$.
- The same $w_i$ can be selected more than once.
- The random step $\alpha$ is designed to generate different neighbors in this case.
Numerical results

- 25 problems from the literature
- Dimension from 2 to 100
- Most with several local minima
- Some with “crowded” local minima
- Measures of performance:
  1. Percentage of success (i.e. identification of the global optimum) on 100 runs
  2. Average number of function evaluations for successful runs
Shubert function

\[
\left( \sum_{j=1}^{5} j \cos((j + 1)x_1 + j) \right) \left( \sum_{j=1}^{5} j \cos((j + 1)x_2 + j) \right)
\]
Numerical results

Competition:


5. General variable neighborhood search (GVNS) Mladenovic et al. (2006)
Numerical results: success rate

<table>
<thead>
<tr>
<th>Problem</th>
<th>VNS</th>
<th>CHA</th>
<th>DSSA</th>
<th>DTS</th>
<th>SAHPS</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>ES</td>
<td>100</td>
<td>100</td>
<td>93</td>
<td>82</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>RT</td>
<td>84</td>
<td>100</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>78</td>
<td>100</td>
<td>94</td>
<td>92</td>
<td>86</td>
<td>100</td>
</tr>
<tr>
<td>$R_2$</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$Z_2$</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>DJ</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>$H_{3,4}$</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>95</td>
<td>100</td>
</tr>
<tr>
<td>$S_{4,5}$</td>
<td>100</td>
<td>85</td>
<td>81</td>
<td>75</td>
<td>48</td>
<td>100</td>
</tr>
<tr>
<td>$S_{4,7}$</td>
<td>100</td>
<td>85</td>
<td>84</td>
<td>65</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>$S_{4,10}$</td>
<td>100</td>
<td>85</td>
<td>77</td>
<td>52</td>
<td>48</td>
<td>100</td>
</tr>
</tbody>
</table>
### Numerical results: success rate

<table>
<thead>
<tr>
<th>Problem</th>
<th>VNS</th>
<th>CHA</th>
<th>DSSA</th>
<th>DTS</th>
<th>SAHPS</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_5$</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>85</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>$Z_5$</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>$H_{6,4}$</td>
<td>100</td>
<td>100</td>
<td>92</td>
<td>83</td>
<td>72</td>
<td>100</td>
</tr>
<tr>
<td>$R_{10}$</td>
<td>100</td>
<td>83</td>
<td>100</td>
<td>85</td>
<td>87</td>
<td>100</td>
</tr>
<tr>
<td>$Z_{10}$</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>HM</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$GR_6$</td>
<td>100</td>
<td></td>
<td>90</td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>$GR_{10}$</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CV</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DX</td>
<td>100</td>
<td></td>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MG</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Numerical results: success rate

<table>
<thead>
<tr>
<th>Problem</th>
<th>VNS</th>
<th>CHA</th>
<th>DSSA</th>
<th>DTS</th>
<th>SAHPS</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{50}$</td>
<td>100</td>
<td>79</td>
<td></td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Z_{50}$</td>
<td>100</td>
<td>100</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$R_{100}$</td>
<td>100</td>
<td>72</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- Excellent success rate on these problems
- Best competitor: GVNS (Mladenovic et al, 2006)
Performance Profile

Performance Profile proposed by Dolan and Moré (2002)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method A</td>
<td>20 10 ** 10 ** 20 10 15 25 **</td>
</tr>
<tr>
<td>Method B</td>
<td>10 30 70 60 70 80 60 75 ** **</td>
</tr>
</tbody>
</table>
## Performance Profile

→ Performance Profile proposed by Dolan and Moré (2002)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method A</td>
<td>2 1 ( r_{fail} ) 1 ( r_{fail} ) 1 1 1 1 ( r_{fail} )</td>
</tr>
<tr>
<td>Method B</td>
<td>1 3 1 6 1 4 6 5 ( r_{fail} ) ( r_{fail} )</td>
</tr>
</tbody>
</table>
Performance Profile

→ Performance Profile proposed by Dolan and Moré (2002)

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method A</td>
<td>2 1 $r_{fail}$ 1 $r_{fail}$ 1 1 1 1 1 $r_{fail}$</td>
</tr>
<tr>
<td>Method B</td>
<td>1 3 1 6 1 4 6 5 $r_{fail}$ $r_{fail}$</td>
</tr>
</tbody>
</table>

Discrete choice models and heuristics for global nonlinear optimization – p.46/52
Numerical results: efficiency

Number of function evaluations (4 competitors)
Numerical results: efficiency

Number of function evaluations (zoom)

![Graph showing the probability of different algorithms](image-url)
Numerical results: efficiency

Number of function evaluations (GVNS)

![Graph showing probability of reaching Pi within a certain number of function evaluations for VNS and GVNS algorithms.](image)

- **Probability (r ≤ Pi)**
- **Pi**
- **VNS**
- **GVNS**

Discrete choice models and heuristics for global nonlinear optimization – p.49/52
Numerical results: efficiency

Number of function evaluations (zoom)
Conclusions

- Use of state of the art methods from
  - nonlinear optimization: TR + Q-Newton
  - discrete optimization: VNS
- Two new ingredients:
  - Premature stop of LS to spare computational effort
  - Exploits curvature for smart coverage
- Numerical results consistent with the algorithm design
Global optimization

• Collaboration with Michaël Thémans (EPFL) and Nicolas Zufferey (U. Laval, Québec).
• Paper under preparation

Thank you!