

夏ゼミ@今治

行動モデルの推定計算とか

東京大学大学院工学系研究科

特任助教 柳沼秀樹

yaginuma@civil.t.u-tokyo.ac.jp

「実装なきものは無力」

Prato(2009)による体系的なレビュー

Garlo Giacomo Prato: Route choice modeling; past, present and future research directions, *Journal of Choice Modelling*, Vol.2(1), pp.65-100, 2009.

❖ モデリング (相関構造)

- ロジット型: ロジットモデルをベースに経路の相関構造を変数で表現
⇒ C-Logit、Path Size Logit、Path Size Correction Logit
- GEV型: 誤差項に一般化極地分布を適用して経路の相関構造を表現
⇒ Paired Combinatorial Logit、Cross Nested Logit、Generalized Nested Logit
- 非GEV型: GEVモデル体系とは異なるアプローチ
⇒ Probit model、Mixed Logit
(Random CoefficientsおよびFactor Analytic Approach)

Closed-formモデルの推定であってもサンプリング系の研究が進展しており計算量は増大. Open-formでは高速化が必須

1. 行動モデルと計算(経路選択を例に)

❖ 選択肢集合

- 確定的最短経路に基づく手法: 最短経路に基づいて選択集合を生成
⇒ K番目経路探索、Labeling法、Link elimination法、Link penalty法
(SPサーチの高速化: データ構造, アルゴリズム, エリア分割)
- 確率的 shortest 経路探索に基づく手法: 確率を加味したヒューリスティック
⇒ モンテカルロ・シミュレーション法、確率的生成関数を利用した方法
(高次元乱数の高速生成: GPGPUによる高密度コアでの並列)
- 制約条件つき列挙法: 選択規範や最適化手法に基づいて列挙
⇒ branch and bound、Path-based algorithms
(組合せ爆発: データ構造 ex: ZDD, 大規模並列)
- 確率的手法: 確率論に基づいて選択肢集合を生成する手法
⇒ Manskiの2段階選択モデル、Random walkに基づくサンプリング

- ✓ GPS観測データを用いて経路を確率的に生成する手法 (network-free、DDR、sub-network等) がより一層重要
- ❖ 経路選択肢集合が課題となっており, データに基づく確率的手法を中心に研究が進められている

⇒ 手法の高度化, 対象の大規模化にはアルゴリズムと並列計算の両面から攻める

➤ 経路選択と計算

- 誤差: 構造化プロビットモデル(屋井, 岩倉, 森地 1997)
- 選択肢: Sub-Network (Frejinger and Bierlaire 2007)
- 選択肢: Hyperpath (Spiess and Florian 1989)
- MACML推定法 (Bhat 2011)
- MCMCアプローチ (Flötteröd and Bierlaire 2013)

➤ 並列計算超入門

- 並列計算概要
- 実装における話

3-1. 構造化プロビットモデル(1)

5

多項ロジットモデル(MNL)

$$P(i) = \frac{\exp(\mu V_i)}{\sum_{j \in C} \exp(\mu V_j)}$$

- Luce(1959), McFadden(1974)
- Closed-form
- 選択肢間の相関を無視(IIA)

通常 of 最適化アルゴリズムで
パラメータ推定が可能であり、
ソースコードの開発が容易

⇒爆発的に各分野に普及

多項プロビットモデル(MNP)

$$P(i) = \int_{\varepsilon_1=-\infty}^{\varepsilon_i+V_i-\varepsilon_1} \cdots \int_{\varepsilon_i=-\infty}^{\infty} \cdots \int_{\varepsilon_J=-\infty}^{\varepsilon_i+V_i-\varepsilon_J} \phi(\varepsilon) d\varepsilon_J \cdots d\varepsilon_1$$
$$\phi(\varepsilon) = \frac{1}{(\sqrt{2\pi})^{J-1} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} \varepsilon \Sigma^{-1} \varepsilon'\right)$$

- Thurstone(1927)
- Open-form
- 選択肢間の相関を表現

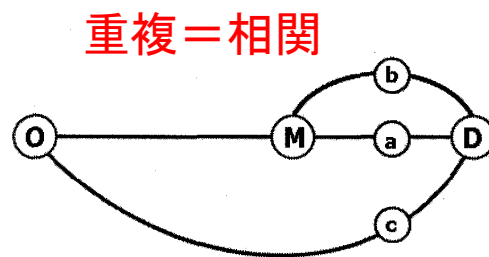
選択肢数-1の多重積分が必要であり、
パラメータ推定は
非常に煩雑

⇒計算が困難であるため
近年まで伸び悩む…

3-1. 構造化プロビットモデル(2)

6

首都圏鉄道では経路重複が存在するため、IIAを仮定したロジットでは便益計測時に誤差が発生
⇒プロビットが有効



課題

- OD毎に経路選択集合が異なるため、共分散が設定できない
⇒経路長に依存する誤差と経路固有の誤差に分解
- 計算コストが非常に高い（選択枝数-1の多重積分）
⇒シミュレーション法(GHK)による高速化

$$U_i = V_i + \varepsilon_i$$

経路長に依存する誤差

$$\varepsilon_i = \varepsilon_i^{Length} + \varepsilon_i^{Route}$$

経路固有の誤差

考え方はMXLと同じ！

$$U_{in} = V_{in} + [\eta_{in} + \varepsilon_{in}]$$

3-1. 構造化プロビットモデル(3)

7

誤差項とその共分散

経路長に依存する誤差

$$\varepsilon_r = \varepsilon_r^1 + \varepsilon_r^0$$

経路固有の誤差

$$\Sigma = \Sigma^1 + \Sigma^0$$

$$\Sigma = \sigma^2 \begin{pmatrix} L_1 & L_{12} & \cdots & L_{1R} \\ L_{12} & L_2 & \cdots & L_{2R} \\ \vdots & \vdots & \ddots & \vdots \\ L_{1R} & L_{2R} & \cdots & L_R \end{pmatrix} + \sigma_0^2 I$$

経路長に依存する誤差

- 経路長が長いほど効用の分散は**大**

$$\text{Var}(\varepsilon_r^1) = L_r \sigma^2$$

- 重複区間が長いほど2経路間の共分散は**大**
⇒ 共分散は重複距離の分散に等しい

$$\text{Cov}(\varepsilon_r^1, \varepsilon_q^1) = L_{rq} \sigma^2$$

経路固有の誤差

- 経路毎に独立に発生 (共分散=0)

$$\begin{aligned} \text{Cov}(\varepsilon_r^0, \varepsilon_q^0) &= \sigma_0^2, \quad q = r \\ &= 0, \quad q \neq r \end{aligned}$$

分散比で表現

$$\Sigma = \sigma_0^2 \begin{pmatrix} \eta L_1 + 1 & \eta L_{12} & \cdots & \eta L_{1R} \\ \eta L_{12} & \eta L_2 + 1 & \cdots & \eta L_{2R} \\ \vdots & \vdots & \ddots & \vdots \\ \eta L_{1R} & \eta L_{2R} & \cdots & \eta L_R + 1 \end{pmatrix}$$

$$\eta = \frac{\sigma^2}{\sigma_0^2}$$

分散比のみを推定するだけで良い！

L_r : 経路 r の経路長

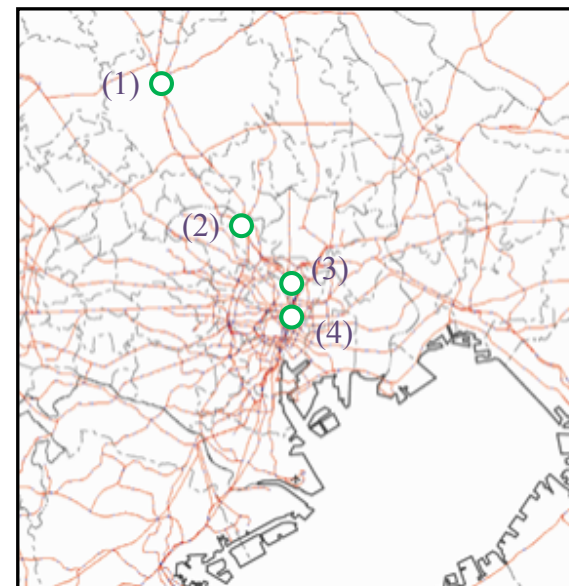
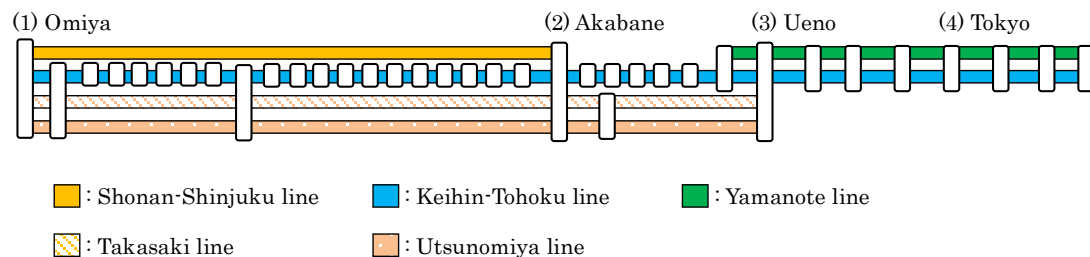
L_{rq} : 経路 r と q の重複経路長

σ^2 : 単位距離あたりの分散

3-1. 構造化プロビットモデル(4)

運輸政策18号答申に構造化プロビットを導入(岩倉2010)

例：大宮-神田間での再現



パラメータ推定結果

説明変数	通勤目的	
	推定値	t値
乗車時間(分)	-0.0943	-8.09
アクセス・イグレス時間(分)	-0.127	-11.7
乗換時間(分)	-0.112	-10.7
運賃(円)	-0.002	-3.98
混雑指標	-0.00869	-3.34
分散パラメータ	0.436	2.71
尤度比	0.39	
サンプル数	1218	

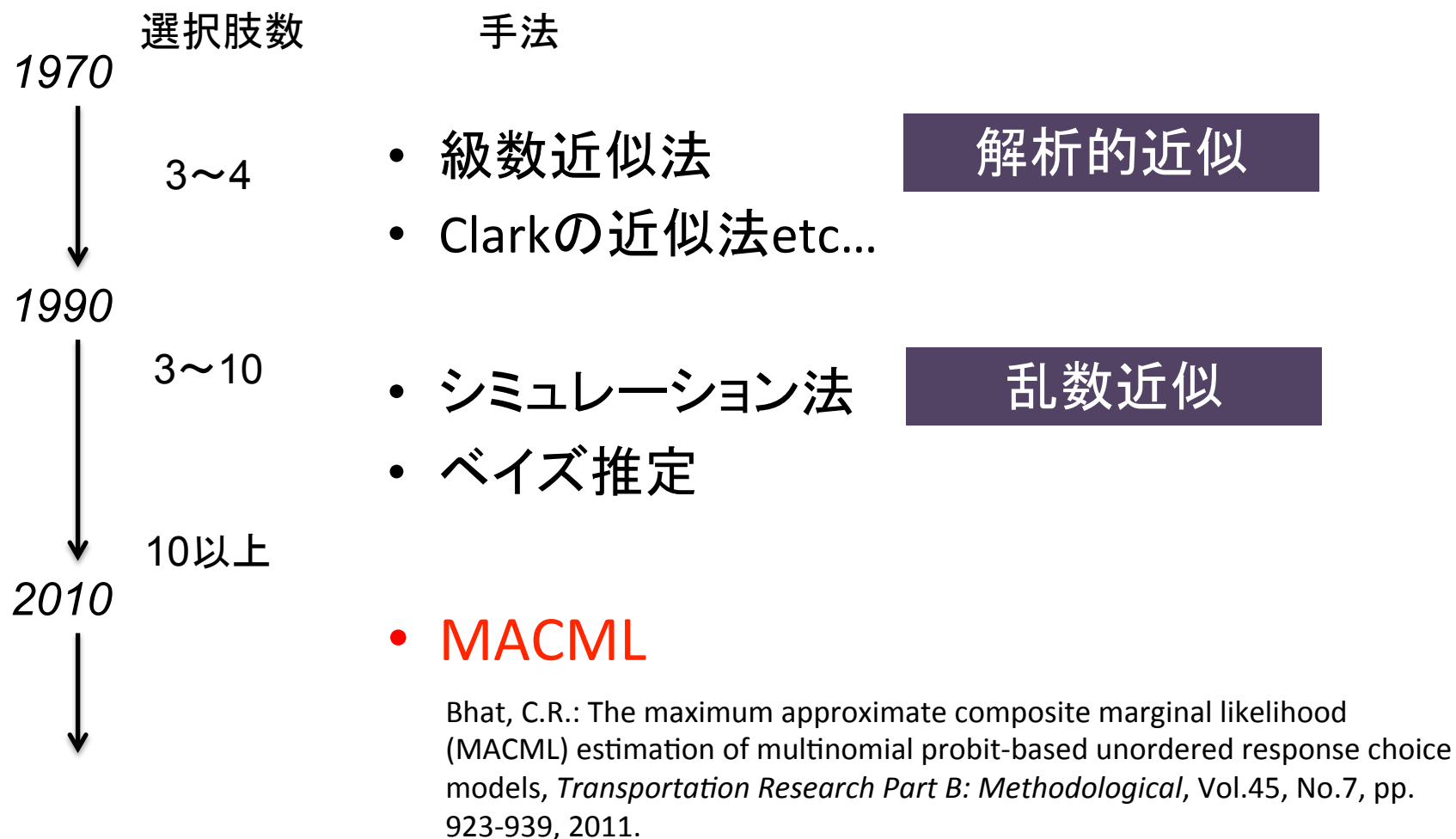
現況再現結果

		MNL		MNPSC	
発着駅	各経路の選択確率	実績(H7センサス)	独立性非考慮	独立性考慮	
大宮駅 → 神田駅	i 東北→山手	33%	28%	27%	48% } 52% } 47%
	j 東北→京浜東北	15%	24%	20%	
神田駅	k 京浜東北	53%	47%	52%	
選択確率計		100%	100%	100%	

経路重複の緩和により高い予測精度を実現
(全駅間断面交通量は実績の $\pm 10\%$)

- 首都圏の鉄道需要予測では、高い表現力を有する構造化プロビットが用いられている。また、当該モデルは目的選択や出発時刻選択にも適用可能。
- MXLよりも優れてる？（清水1999）
⇒計算時間が4倍速く計算時間の変動は1/10かつ推定パラメータの変動係数もMXLより小さい
- 無論、Open-formなので計算コストは高いが、シミュレーション法により高速化が可能。
⇒近年ではGHKやMCMC等のシミュレーションに加えて、MACML等の解析近似手法もある
- 構造化による重複は考慮できるが、今後は選択肢集合（経路集合）の設定方法が課題。

開発当初から多重積分の計算がネックとなり、各種の数値計算アルゴリズムが提案



(1) Maximum Approximate Composite Marginal Likelihood (MACML) estimationの提案

- ✓ Open-fromな離散選択モデル(c.f. MNP, MXL)のパラメータを
簡単かつ高速で推定する手法を構築
- ✓ MACML推定は2つのテクニックにより構成
 - ①多変量累積標準正規分布(MVNCD※1)の解析的近似手法
 - ②合成周辺尤度(CML※2)を用いたパラメータ推定

(2) 各種(Mixed)Probitモデルへの適用方法の提示

- ✓ Cross-section, Panel, Spatial Correlation etc...

(3) 数値実験による有効性の確認

- ✓ 通常の推定手法(MSL)と比較して, 計算時間は約38倍速く
(66.09→1.96), 推定値のバイアスは7.3ポイント低い(9.8%→2.5%).

※1 MVNCD: Multi-Variate standard Normal Cumulative Distribution

※2 CML: Composite Marginal Likelihood

2.1 MVNCD(多変量累積標準正規分布)の解析的近似

⇒多変量正規分布を単変量分布の積で近似

【設定1: 分布の分解】

$$\Pr(\mathbf{W} < \mathbf{w}) = \Pr(W_1 < w_1, W_2 < w_2, W_3 < w_3, \dots, W_I < w_I). \quad \mathbf{W}: \text{多変量標準正規分布}$$

同時確率を下記のように分布の積に分解

$$\Pr(\mathbf{W} < \mathbf{w}) = \underbrace{\Pr(W_1 < w_1, W_2 < w_2)}_{\text{二変量周辺分布}} \times \prod_{i=3}^I \underbrace{\Pr(W_i < w_i | W_1 < w_1, W_2 < w_2, W_3 < w_3, \dots, W_{i-1} < w_{i-1})}_{\text{単変量条件つき分布}(I>3)}.$$

【設定2: インディケータIでの分散共分散表現】

$$\tilde{\mathbf{I}}_i = \begin{cases} 1 & W_i < w_i \\ 0 & \text{otherwise} \end{cases} \quad E(\tilde{I}_i) = \Phi(w_i)$$

Iの期待値を単変量累積標準正規分布Φで評価

$$\text{Cov}(\tilde{I}_i, \tilde{I}_i) = \text{Var}(\tilde{I}_i) = \Phi(w_i) - \Phi^2(w_i) = \Phi(w_i)[1 - \Phi(w_i)],$$

$$\text{Cov}(\tilde{I}_i, \tilde{I}_j) = E(\tilde{I}_i \tilde{I}_j) - E(\tilde{I}_i)E(\tilde{I}_j) = \Phi_2(w_i, w_j, \rho_{ij}) - \Phi(w_i)\Phi(w_j), i \neq j$$

上記を統合

$$\Pr(W_i < w_i | W_1 < w_1, W_2 < w_2, W_3 < w_3, \dots, W_{i-1} < w_{i-1}) = E(\tilde{I}_i | \tilde{I}_1 = 1, \tilde{I}_2 = 1, \tilde{I}_3 = 1, \dots, \tilde{I}_{i-1} = 1).$$

【線形回帰モデルでの展開】

$$\Pr(W_i < w_i | W_1 < w_1, W_2 < w_2, W_3 < w_3, \dots, W_{i-1} < w_{i-1}) = E(\tilde{I}_i | \tilde{I}_1 = 1, \tilde{I}_2 = 1, \tilde{I}_3 = 1, \dots, \tilde{I}_{i-1} = 1).$$

$$\tilde{I}_i - E(\tilde{I}_i) = \alpha' [\tilde{\mathbf{I}}_{<i} - E(\tilde{\mathbf{I}}_{<i})] + \tilde{\eta}, \quad \text{誤差項}$$

$$\times \tilde{\mathbf{I}}_{<i} = (\tilde{I}_1, \tilde{I}_2, \dots, \tilde{I}_{i-1})$$

$$\hat{\alpha} = \Omega_{<i}^{-1} \cdot \Omega_{i,<i}, \quad \text{最小二乗係数}$$

$$\Omega_{<i} = \text{Cov}(\mathbf{I}_{<i}, \mathbf{I}_{<i}) = \begin{bmatrix} \text{Cov}(\tilde{I}_1, \tilde{I}_1) & \text{Cov}(\tilde{I}_1, \tilde{I}_2) & \text{Cov}(\tilde{I}_1, \tilde{I}_3) & \dots & \text{Cov}(\tilde{I}_1, \tilde{I}_{i-1}) \\ \text{Cov}(\tilde{I}_2, \tilde{I}_1) & \text{Cov}(\tilde{I}_2, \tilde{I}_2) & \text{Cov}(\tilde{I}_2, \tilde{I}_3) & \dots & \text{Cov}(\tilde{I}_2, \tilde{I}_{i-1}) \\ \text{Cov}(\tilde{I}_3, \tilde{I}_1) & \text{Cov}(\tilde{I}_3, \tilde{I}_2) & \text{Cov}(\tilde{I}_3, \tilde{I}_3) & \dots & \text{Cov}(\tilde{I}_3, \tilde{I}_{i-1}) \\ \vdots & & & & \\ \text{Cov}(\tilde{I}_{i-1}, \tilde{I}_1) & \text{Cov}(\tilde{I}_{i-1}, \tilde{I}_2) & \text{Cov}(\tilde{I}_{i-1}, \tilde{I}_3) & \dots & \text{Cov}(\tilde{I}_{i-1}, \tilde{I}_{i-1}) \end{bmatrix}, \quad \Omega_{i,<i} = \text{Cov}(\mathbf{I}_{<i}, \mathbf{I}_i) = \begin{bmatrix} \text{Cov}(\tilde{I}_1, \tilde{I}_1) \\ \text{Cov}(\tilde{I}_1, \tilde{I}_2) \\ \text{Cov}(\tilde{I}_1, \tilde{I}_3) \\ \vdots \\ \text{Cov}(\tilde{I}_i, \tilde{I}_{i-1}) \end{bmatrix}.$$

【単変量正規分布での近似】

$$\Pr(W_i < w_i | W_1 < w_1, W_2 < w_2, \dots, W_{i-1} < w_{i-1}) \approx \Phi(w_i) + (\Omega_{<i}^{-1} \cdot \Omega_{i,<i})' (1 - \Phi(w_1), 1 - \Phi(w_2), \dots, 1 - \Phi(w_{i-1}))'$$

多変量正規分布を選択肢数-1の単変量正規分布で表現
⇒計算量は大幅に削減される！

対象モデル

Cross-section random coefficients model (Mixed MNP)

$$U_{qi} = \beta_q' \mathbf{x}_{qi} + \varepsilon_{qi} \quad \beta_q \sim MVN(\mathbf{b}, \mathbf{\Omega}),$$

$$L_q = \int_{\beta=-\infty}^{\infty} \left\{ \int_{\lambda=-\infty}^{\infty} \left(\prod_{i \neq m} \left[\Phi \left\{ \left[-\sqrt{2}(\beta' \mathbf{z}_{qim}) \right] + \lambda \right\} \right] \right) \phi(\lambda) d\lambda \right\} f(\beta | \mathbf{b}, \mathbf{\Omega}) d\beta,$$

where $\mathbf{z}_{qim} = \mathbf{x}_{qi} - \mathbf{x}_{qm}$,

q : 個人
 i : 選択肢
 ε : 誤差項 IIDガンベル

真値の設定

$$\mathbf{b} = (1.5, -1, 2, 1, -2) \quad \mathbf{\Omega} = \begin{bmatrix} 1 & -0.50 & 0.25 & 0.75 & 0 \\ -0.50 & 1 & 0.25 & -0.50 & 0 \\ 0.25 & 0.25 & 1 & 0.33 & 0 \\ 0.75 & -0.50 & 0.33 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

5000人分の仮想データを乱数を用いて20セットの実験データを作成

Cross-sectional random coefficients model

【Diagonal case】

- 分散共分散行列の対角成分のみを推定
- 計算時間：平均で約34倍速く、かつ速度にバラツキがない(安定的)
- バイアス：平均で7.3ポイント低く、分散成分についても良好

Table 1a

Evaluation of the ability to recover true parameters for the cross-sectional diagonal case.

Parameter	True value	MSL method					MACML method				
		Parameter estimates		Standard error estimates			Parameter estimates		Standard error estimates		
		Mean estimate	Absolute percentage bias (%)	Asymptotic standard error	Simulation standard error	Simulation adjusted asymptotic standard error	Mean estimate	Absolute percentage bias (%)	Asymptotic standard error	Approximation standard error	Approximation adjusted asymptotic standard error
<i>Mean values of the β vector</i>											
b_1	1.500	1.366	9.0	0.129	0.050	0.139	1.472	1.9	0.167	0.022	0.169
b_2	−1.000	−0.906	9.4	0.089	0.033	0.095	−0.976	2.4	0.113	0.014	0.114
b_3	2.000	1.801	10.0	0.167	0.066	0.180	1.940	3.0	0.218	0.028	0.219
b_4	1.000	0.906	9.4	0.089	0.034	0.095	0.977	2.3	0.114	0.014	0.114
b_5	−2.000	−1.820	9.0	0.170	0.067	0.182	−1.960	2.0	0.220	0.028	0.222
<i>Standard deviations of the β vector</i>											
σ_1	1.000	0.885	11.5	0.111	0.038	0.117	0.958	4.2	0.135	0.017	0.137
σ_2	1.000	0.906	9.4	0.111	0.040	0.118	0.984	1.6	0.136	0.016	0.137
σ_3	1.000	0.867	13.3	0.112	0.041	0.119	0.941	5.9	0.135	0.017	0.136
σ_4	1.000	0.904	9.6	0.111	0.040	0.118	0.982	1.8	0.136	0.017	0.137
σ_5	1.000	0.927	7.3	0.117	0.041	0.124	1.002	0.2	0.140	0.016	0.141
Overall mean value across parameters	–		9.8%	0.121	0.045	0.129	–	2.5	0.151	0.019	0.153
Mean time		66.09					1.96				
Std. dev. of time		10.87					0.42				
% of Runs converged		100%					100%				

Cross-sectional random coefficients model

【Non Diagonal case】

- 分散共分散行列の下三角を推定
- Diagonal caseと同様に約33倍早い(バイアスは2.1ポイント)

$$\tilde{\Omega} = \begin{bmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ l_{31} & l_{32} & l_{33} & & \\ l_{41} & l_{42} & l_{43} & l_{44} & \\ l_{51} & l_{52} & l_{53} & l_{54} & l_{55} \end{bmatrix}$$

Table 1b

Evaluation of the ability to recover true parameters for the cross-sectional non-diagonal case.

Parameter	True value	MSL method					MACML method				
		Parameter estimates		Standard error estimates			Parameter estimates		Standard error estimates		
		Mean estimate	Absolute percentage bias (%)	Asymptotic standard error	Simulation standard error	Simulation adjusted asymptotic standard error	Mean estimate	Absolute percentage bias (%)	Asymptotic standard error	Approximation standard error	Approximation adjusted asymptotic standard error
Mean values of the β vector											
b1	1.500	1.374	8.4	0.133	0.049	0.142	1.443	3.8	0.147	0.022	0.148
b2	−1.000	−0.912	8.8	0.093	0.037	0.100	−0.959	4.1	0.102	0.014	0.103
b3	2.000	1.830	8.5	0.174	0.068	0.187	1.923	3.8	0.191	0.029	0.193
b4	1.000	0.914	8.6	0.092	0.032	0.097	0.958	4.2	0.101	0.014	0.102
b5	−2.000	−1.849	7.6	0.176	0.068	0.189	−1.941	3.0	0.194	0.028	0.196
Cholesky parameters characterizing the covariance matrix of the β vector											
l11	1.000	0.909	9.1	0.112	0.040	0.119	0.959	4.1	0.119	0.017	0.120
l12	−0.500	−0.463	7.3	0.085	0.029	0.090	−0.472	5.6	0.085	0.010	0.085
l13	0.250	0.231	7.5	0.089	0.036	0.096	0.233	6.7	0.087	0.009	0.088
l14	0.750	0.689	8.2	0.092	0.028	0.097	0.707	5.7	0.095	0.013	0.096
l15	0.000	0.006	0.6	0.086	0.040	0.095	0.015	1.5	0.088	0.008	0.089
l22	0.866	0.756	12.7	0.109	0.043	0.117	0.809	6.5	0.116	0.017	0.117
l23	0.433	0.431	0.5	0.105	0.050	0.117	0.436	0.6	0.100	0.012	0.101
l24	−0.144	−0.149	3.6	0.101	0.041	0.109	−0.170	17.8	0.093	0.010	0.094
l25	0.000	−0.021	2.1	0.101	0.055	0.115	−0.019	1.9	0.098	0.010	0.099
l33	0.866	0.750	13.4	0.130	0.073	0.149	0.812	6.3	0.131	0.019	0.132
l34	0.237	0.242	2.0	0.112	0.055	0.125	0.259	9.3	0.106	0.011	0.106
l35	0.000	−0.031	3.1	0.120	0.081	0.145	−0.029	2.9	0.116	0.011	0.117
l44	0.601	0.464	22.9	0.126	0.085	0.152	0.531	11.6	0.125	0.015	0.126
l45	0.000	−0.053	5.3	0.168	0.134	0.214	−0.053	5.3	0.171	0.017	0.172
l55	1.000	0.885	11.5	0.125	0.089	0.153	0.956	4.4	0.136	0.018	0.137
Overall mean value across parameters	–	–	7.6	0.116	0.057	0.130	–	5.5	0.120	0.015	0.121
Mean time		174.32					5.19				
Std. dev. of time		28.13					0.84				
% of Runs converged		100					100				

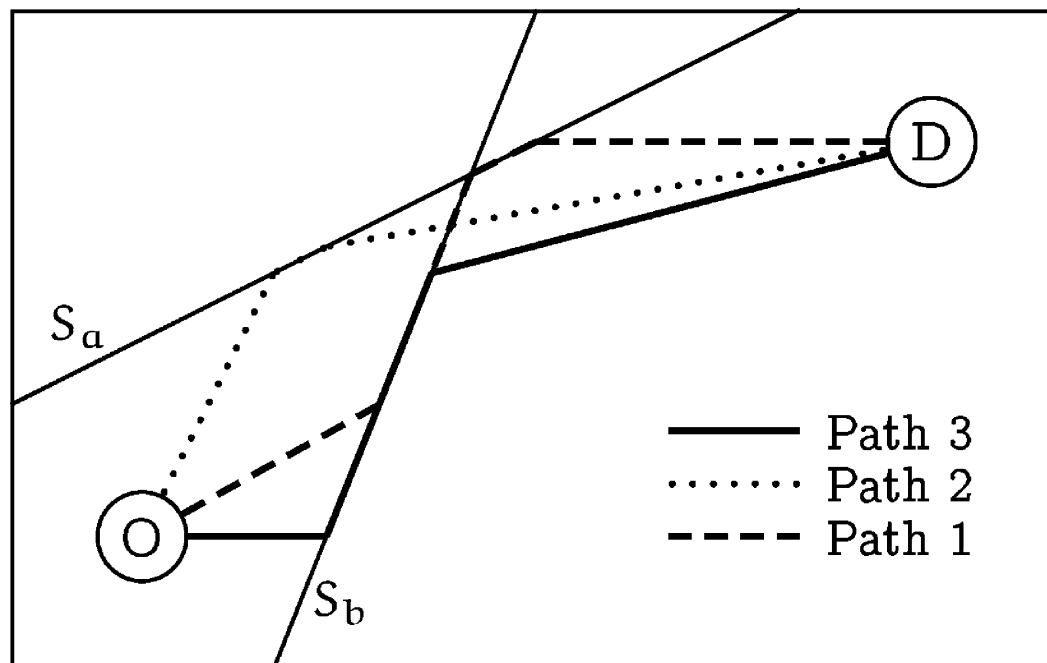
- 正規分布を持つOpen-formなモデル(Mixed-Probit)を対象とした新たな推定手法(MACML推定)を提案
- パネルや空間相関を考慮した改良モデルに対してもMACML推定が適用可能であることを提示
- 数値実験より高速かつ低バイアスなパラメータ推定値が得られる事を確認

【著者談】

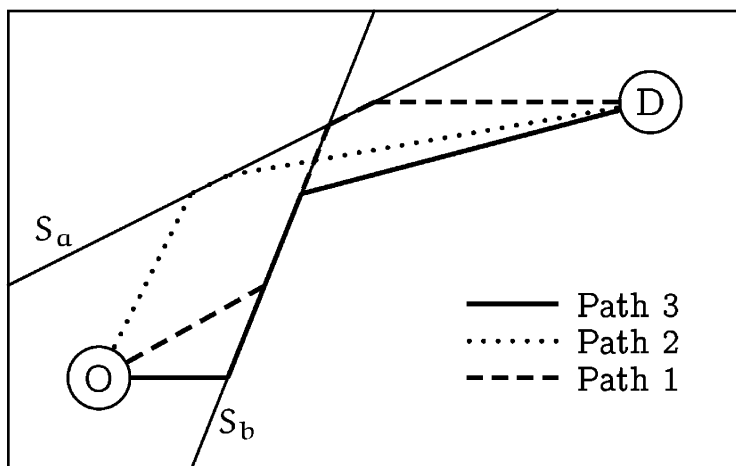
*In closing, the MACML inference approach has the **potential to dramatically influence** the use of the mixed multinomial probit model in practice, and should facilitate the practical application of rich model structures for unordered-response discrete choice modeling.*

G, Flötteröd., M, Bierlaire. : Capturing correlation with subnetworks in route choice models , *Transportation Research Part B*, Vol.41, pp.363-378, 2007.

- モデルを複雑にすることなく，経路重複構造（＝経路の相関）をシンプルに表現
- Sub-networksでは，ネットワークの骨格となるリンク以外を削除して部分グラフを構築し，物理的な重複の有無に関わらず経路がリンクを共有した場合には相関していると仮定



- mixed logit の一種であるError Component Logitを用いて記述
- 複数のモデル (MNL、PSL、通常のEC) との比較した結果、既往モデルと同等の尤度比が得られ、十分な適合度を持つ事が確認



$$U_1 = \beta^T X_1 + \sqrt{l_{1a}} \sigma_a \xi_a + \sqrt{l_{1b}} \sigma_b \xi_b + \varepsilon_1$$

$$U_2 = \beta^T X_2 + \sqrt{l_{2a}} \sigma_a \xi_a + \varepsilon_2$$

$$U_3 = \beta^T X_3 + \sqrt{l_{3a}} \sigma_a \xi_a + \varepsilon_3$$

β^T : 未知パラメータ \mathbf{X}_n : 説明変数ベクトル ε_n : 誤差項
 \mathbf{F}_n : Error Componentの要素行列 ($J \times Q$)
 \mathbf{T} : 分散共分散行列 ($Q \times Q$) l : subnetworkにおける経路重複長
 J : 経路数 Q : Subnetwork数 ξ : 標準正規乱数 $N(0,1)$

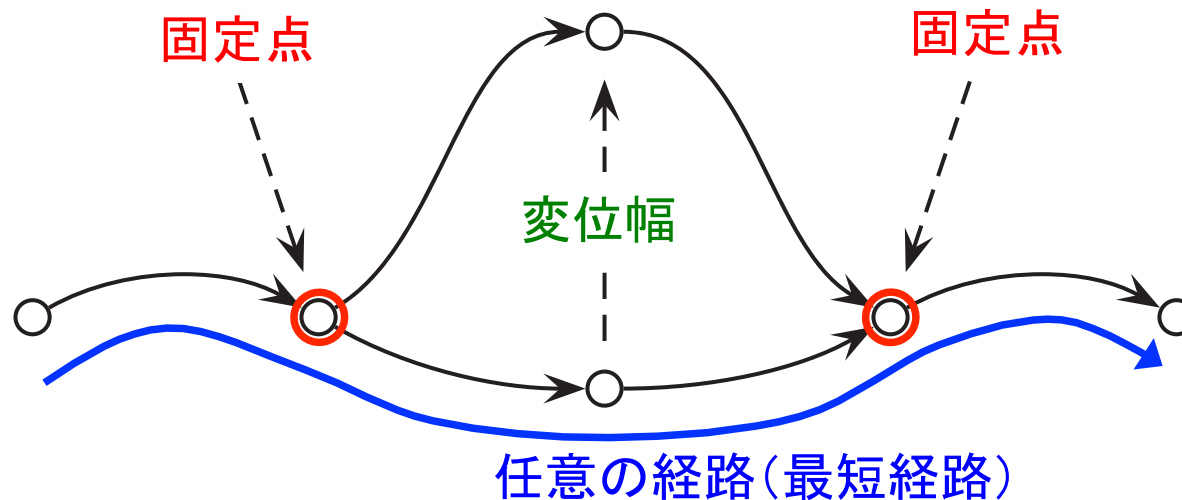
$$\mathbf{FTT}^T \mathbf{F}^T = \begin{pmatrix} l_{1a} \sigma_a^2 + l_{1b} \sigma_b^2 & \sqrt{l_{1a}} \sqrt{l_{2a}} \sigma_b^2 & \sqrt{l_{1b}} \sqrt{l_{3b}} \sigma_b^2 \\ \sqrt{l_{1a}} \sqrt{l_{2a}} \sigma_a^2 & l_{2a} \sigma_a^2 & 0 \\ \sqrt{l_{1b}} \sqrt{l_{3b}} \sigma_b^2 & 0 & l_{3b} \sigma_b^2 \end{pmatrix}$$

E, Frejinger., M, Bierlaire.: Metropolis-Hastings sampling of paths ,
Transportation Research Part B, Vol.48, pp53-66, 2013.

- 経路選択肢集合は選択肢の組合せ数だけ存在
⇒ 列挙することは困難であり、効率化を図るためには確率的なサンプリング方法が必要
- 確率的な経路選択肢集合の特定方法として、**ベイズ推定**に基づく新たなアプローチを提案
- 経路の抽出を「任意の確率分布」から実行するもので、**マルコフチェーン・モンテカルロ法 (MCMC)** の一種であるMetropolis-Hasting (MH) アルゴリズムを応用した抽出確率分布の更新により、最も現況再現性の高い経路をサンプリング
- **MCMCの例**

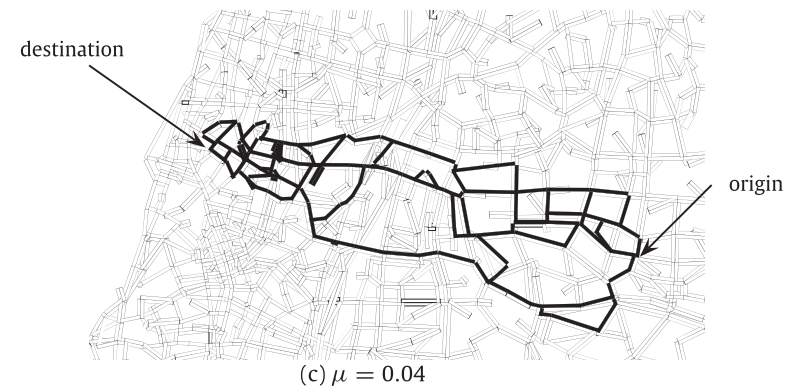
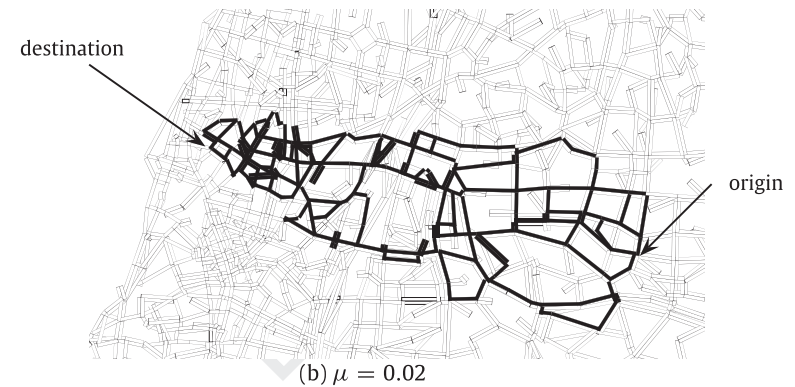
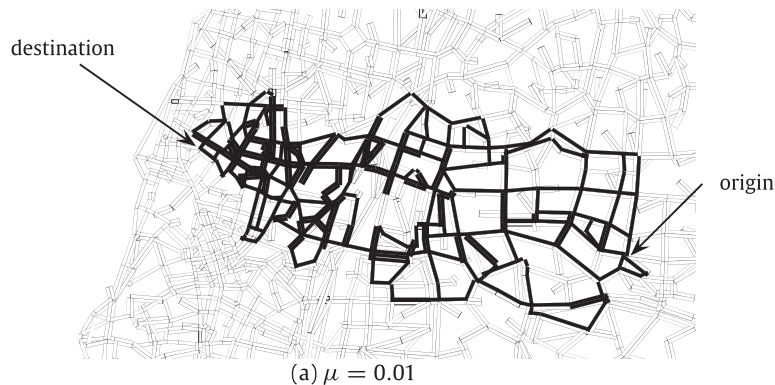
計算手順の概要

- 任意の経路（例えば対象ODペア間の最短経路）を基準として、固定点を定めて経路を変位させる区間を設定
- 区間内から1ヶ所の移動点を設定して、輪ゴムのようにな別の位置に変位させて経路を生成. 変位幅はネットワーク上に制限
- MHによるランダムな抽出を収束するまで繰り返す



4-2. MHを用いた経路サンプリング (3)

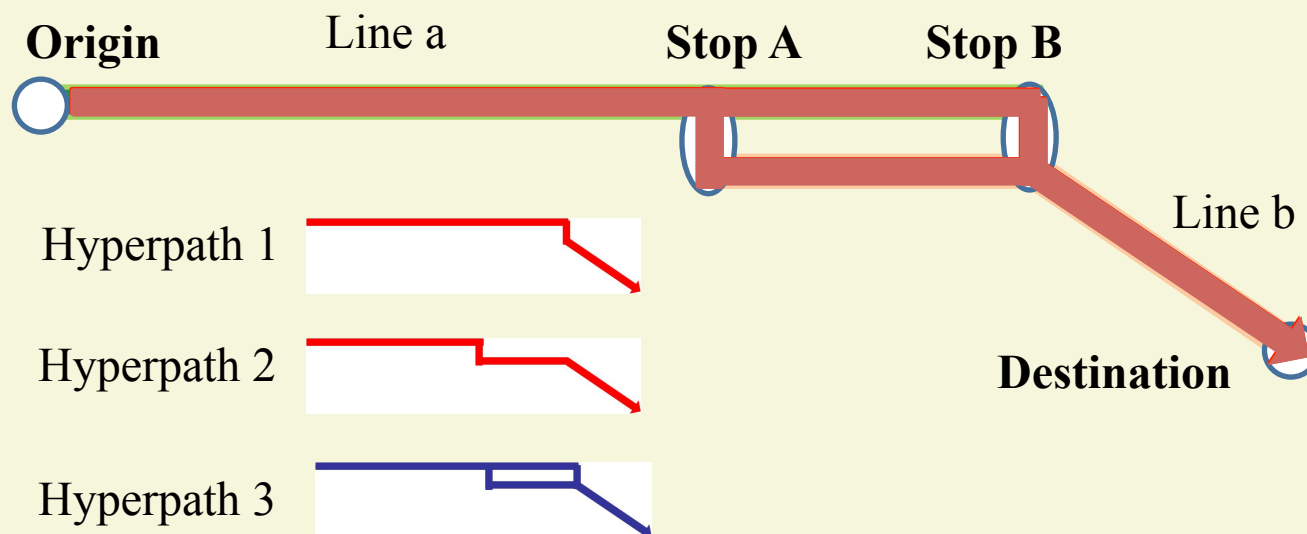
- イスラエルのTel-Avivの道路ネットワークに適用した結果、高速で経路選択枝集合を列挙可能であることが確認
- パラメータを変化させることで集合に含まれる経路数が変化



❖ 公共交通機関配分モデル (Spiess and Florian 1989)

頻度ベースで運行される公共交通機関を対象に, Common line problem に基づく **Hyperpath** 概念によって待ち時間を明示的に考慮した配分モデル

Hyperpath: 乗車方法の組み合わせ (**乗車経路群**)



旅客は一般化費用が最小となるHyperpathを選択することが「最適な戦略＝乗車経路群」となり, それに基づいて交通量が配分される

❖ Hyperpath p の期待一般化費用

$$g_p = \phi \sum_{a \in A_p} \alpha_{ap} t_a + \varphi \sum_{i \in I_p} \frac{\beta_{ip}}{F_{ip}} + \xi \sum_{a \in A_p} \alpha_{ap} CDU_a + \omega \sum_{a \in A_p} t t_a,$$

1. リンク a の期待乗車時間

混雑に対して不変であることを仮定

✧ 合成頻度:

$$F_{ip} = \sum_{a \in OUT_{p(i)}} f_{l(a)},$$

2. ノード i の期待待ち時間

有効頻度を用いてリンク交通量に依存した遅延による待ち時間の変化を記述

✧ 有効頻度:

$$f'_{l(a)} = \frac{1}{w_{l(a)}} = \left[\frac{1}{F_{ip}} + \rho \left(\frac{x_a}{Cap_{l(a)}} \right)^\kappa \right]^{-1},$$

遅延(線路混雑)に相当

$\phi, \varphi, \xi, \omega$: パラメータ, α, β : 通過確率, x : リンク交通量

❖ Hyperpath p の期待一般化費用

$$g_p = \phi \sum_{a \in A_p} \alpha_{ap} t_a + \varphi \sum_{i \in I_p} \frac{\beta_{ip}}{F_{ip}} + \xi \sum_{a \in A_p} \alpha_{ap} CDU_a + \omega \sum_{a \in A_p} t t_a,$$

3. ノード i の混雑不効用

旅客が混雑により感じる不効用を
“混雑指標”を用いて表現

列車内混雑に相当

4. リンク a の乗り換え時間(駅間)

混雑に対して不変であることを仮定

混雑の影響を考慮する場合には、出力される
リンク交通量を用いて逐次更新を行う

✧ 合成頻度:

$$F_{ip} = \sum_{a \in OUT_{p(i)}} f_{l(a)},$$

✧ 有効頻度:

$$f'_{l(a)} = \frac{1}{w_{l(a)}} = \left[\frac{1}{F_{ip}} + \rho \left(\frac{x_a}{Cap_{l(a)}} \right)^\kappa \right]^{-1},$$

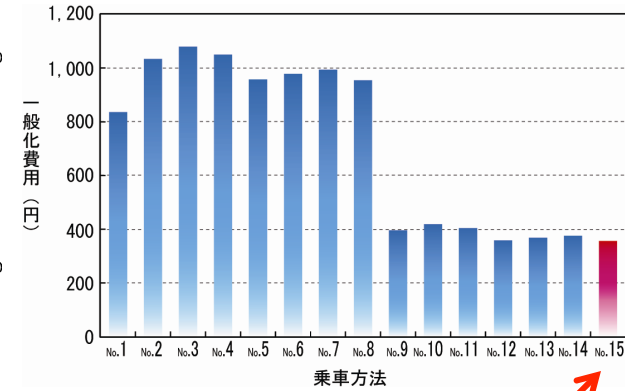
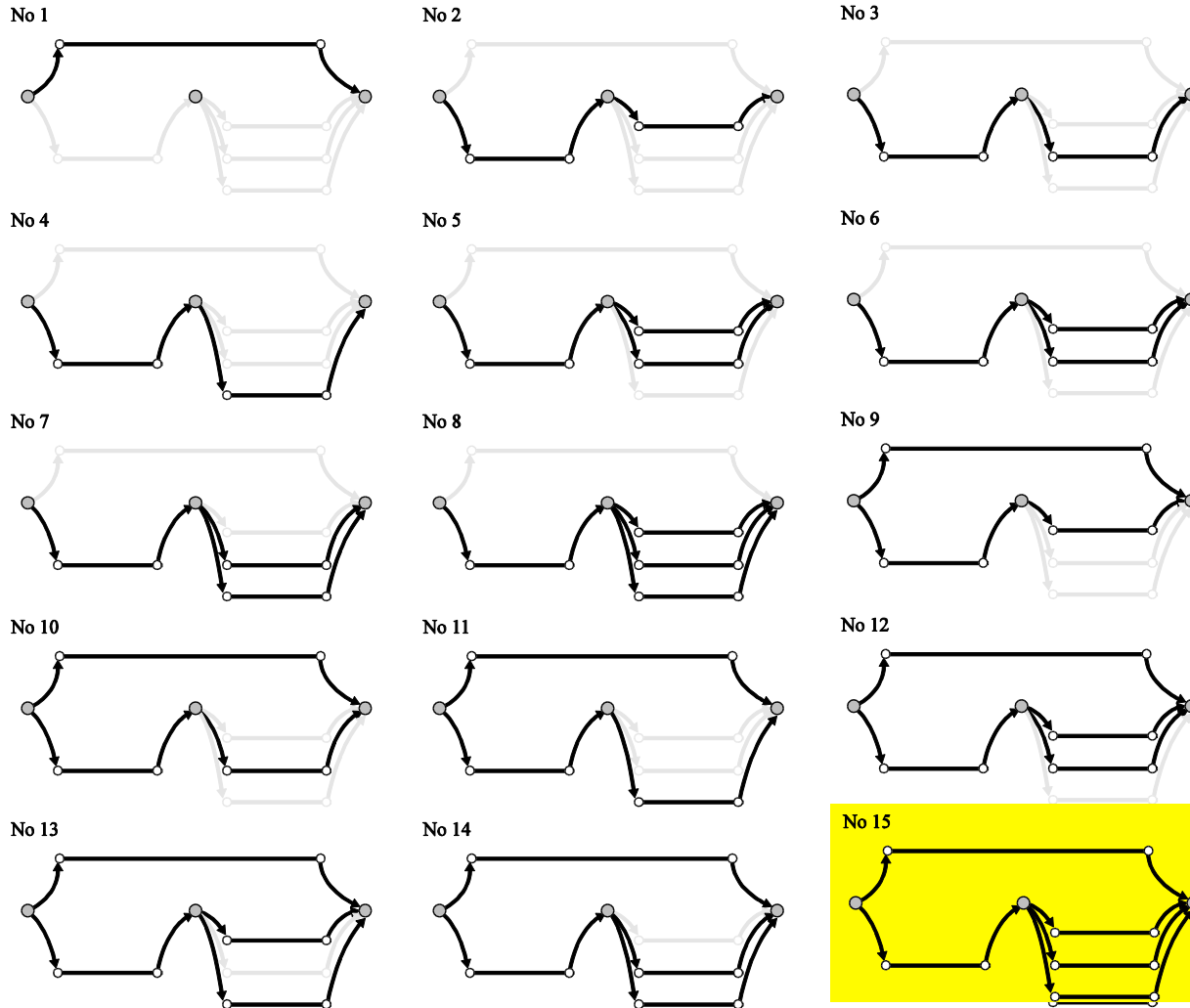
✧ 混雑不効用:

$$CDU_a = t_a \left(\frac{x_a}{Cap_{l(a)}} \right)^\psi.$$

$\phi, \varphi, \xi, \omega$: パラメータ, α, β : 通過確率, x : リンク交通量

4-3. Hyperpath (4)

26



No.15 is Optimal Hyperpath

➤ 分散メモリ型

複数のPCを接続したクラスター計算機（MPIなど）

【長所】 接続PC数を増やすほど大規模計算が可能

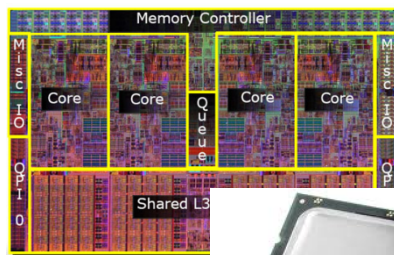
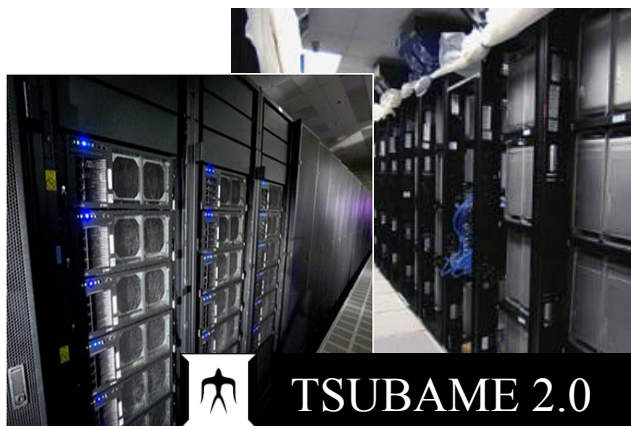
【短所】 マシン間での通信を考慮した実装，計算機の維持管理

➤ 共有メモリ型

マルチコアCPUやGPGPU（OpenMP, TBB, CUDAなど）

【長所】 家庭用PCで計算可能，実装が比較的簡単

【短所】 データ量（メモリ量）や並列数（コア数）に制限がある。

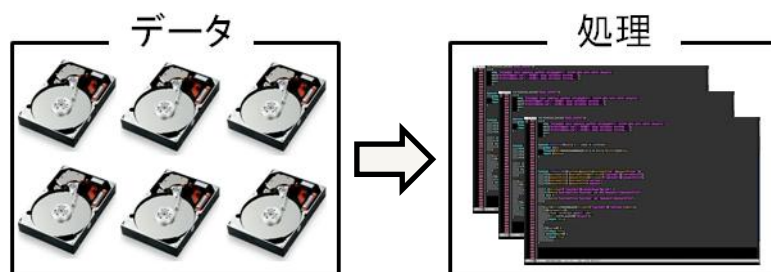


➤ データ並列

入力データを**複数のデータ**に分割して処理を行う

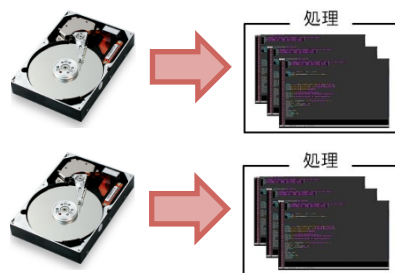
➤ タスク並列

実行する処理を**複数の処理**に分割して実行する.

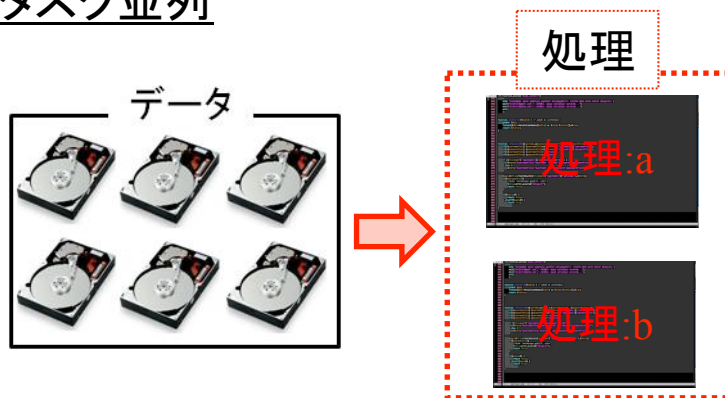


実行する**処理の性質**により
並列化手法を選択する.
また、両者を組み合わせる

データ並列



タスク並列



➤ 交通分野における並列化

- マイクロシミュレーション: エージェントや対象空間を分割した並列処理
- 交通量配分: 最短経路探索 (Hribar *et al.*, 2001), 最適化問題 (Chen and Meyer, 1988) を対象した並列処理

多くの研究はデータ並列に基づく処理を導入

- エージェントはシンプルな処理で並列化の余地が少ない
 - 最短経路探索は再帰的な構造で並列化が困難
- ⇒ 逐次的な処理が多く、タスク並列が難しい

- 総計算時間の 90%以上がhyperpath探索 に費やされる (Yaginuma *et al.*)

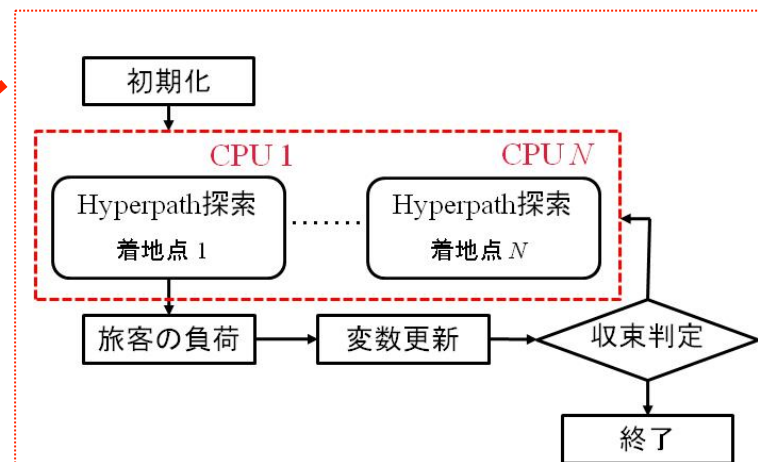
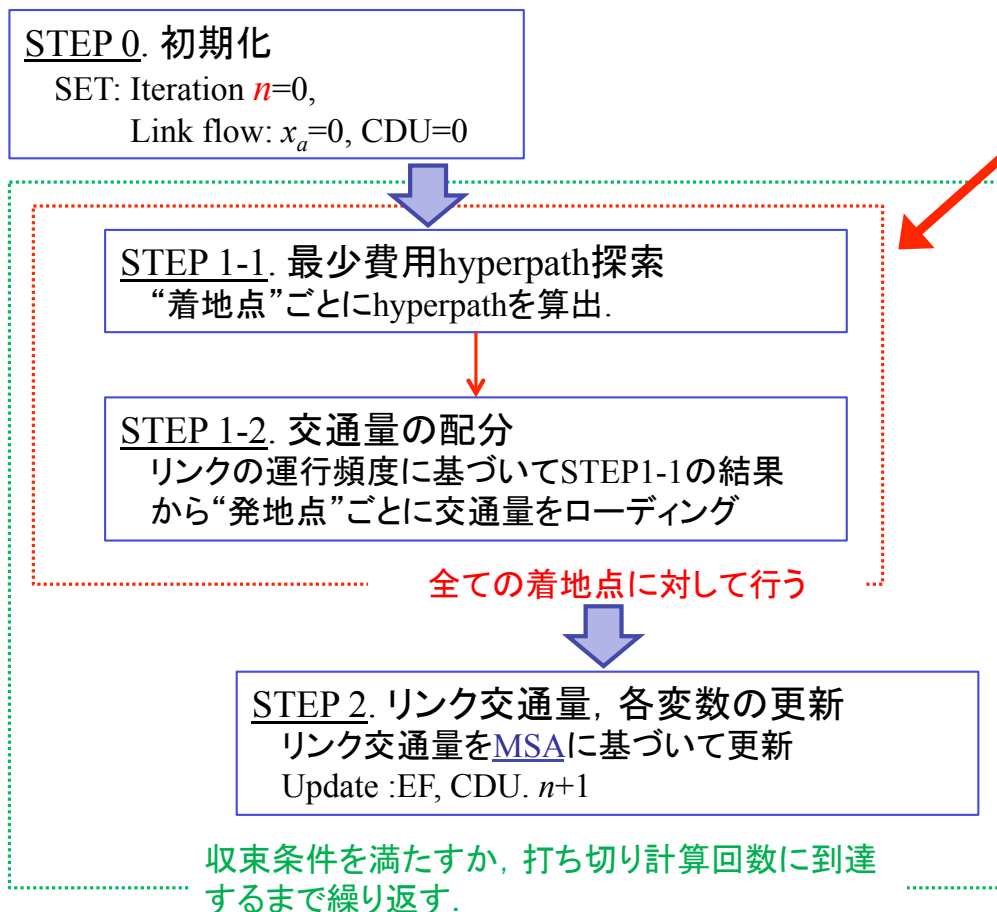
➡ 経路探索部分の並列化が必須となる...

(e.g. 最短経路探索: $O(n^2)$, hyperpath探索: $O(m^2)$ の計算量オーダー) ※ n : ノード数, m : リンク数

共有メモリ型の計算環境でデータ並列を導入:

“マルチコアCPUを用いて各CPUにタスクを割り当てる”手法を検討

❖ 並列化によるループ計算の高速化



高コストなhyperpath探索に対して、**着地点毎にCPUコア**を割り当てるデータ並列型処理を採用

➤ 実行環境の比較

コンパイラおよび最適化オプションがもたらす影響を比較

• コンパイラ

✓ gcc: フリーで利用できるコンパイラ.

✓ Intel: intel社の商用コンパイラ

• 最適化オプション

✓ なし(gccのみ): 何も最適化を施していない ※intel はデフォルトがO2

✓ O2 (gcc , intel): ベクトル化を行い実行速度を上げる

✓ O3 (gcc, intel) : O2 + メモリやデータアクセスを最適化

✓ fast (intelのみ) : O3 + 内部で利用される算術関数等を最適化

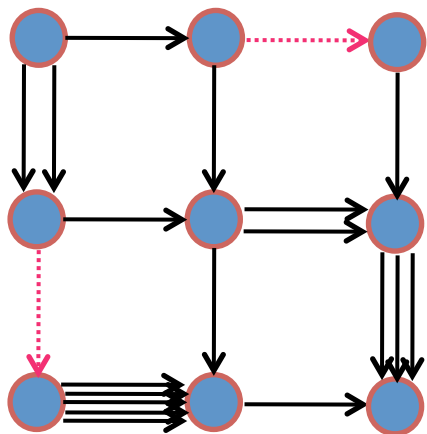
➤ 利用計算機

Apple iMac, CPU: intel core i7 (HT利用で仮想8コア) Memory: 8GB

実装はC言語とOpenMPライブラリーを用いた

• グリット型の仮想ネットワークをランダムに生成

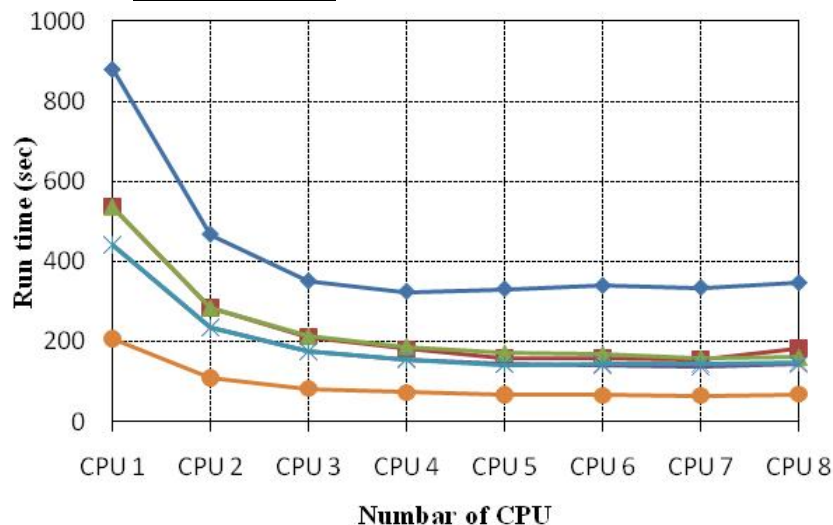
- ✓ 10×10のネットワークを以下の条件で生成
⇒hypergraph変換後は平均ノード数:897, リンク数:2687
- ✓ 並列路線を表現(最大10路線)
- ✓ 低い確率で駅間の**徒歩移動**を表現
- ✓ 全ODペアからランダムに50ペアを抽出



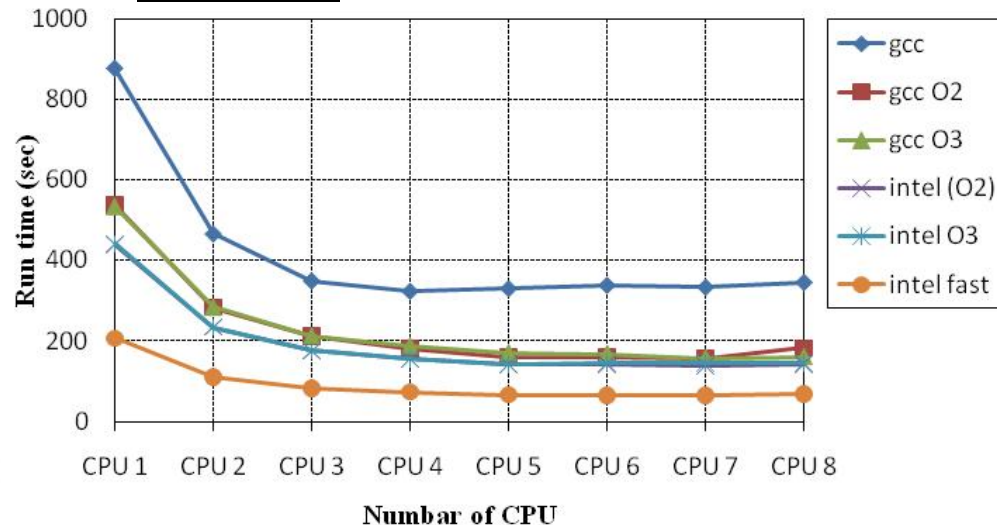
変数	発生方法	設定値
最大並列路線数	一様乱数	1～10
リンク乗車時間	一様乱数	8～12
リンク運行頻度	一様乱数	5～20
ODペア	一様乱数	50ペア
徒歩リンク	並列数が1の場合に 10%の割合でランダムに生成	

これらを10セット用意し、コンパイラと最適化オプションの組み合わせの下で計算時間を比較

並列化 1



並列化 2



- 2つの並列化手法でどの場合でも実行時間に**差が見られない**.
⇒ “旅客の負荷”が比較的高速($O(m)$).
ODペアが多い場合には高速化が期待される.
- コンパイラの違いで**1.99倍**, 最適化により**2.13倍**の速度差が生じる
⇒ Intel はCPUが持つコアやレジスタの制御レベルで最適化している.
- 並列数**4以上**で速度改善が**緩慢**になり, **6・7並列**で**最高速**となる.
⇒ **アムダール則に基づく並列化の低減効果**
CPUの特性(ハイパースレッティングの有無)により変化.

- 個別に相談にのります！