

夏合宿ゼミ # keynote 1

# 動学問題と最適化

pedestrian

cycle

bus & tram

cycle parking

cyclist

pedestrian

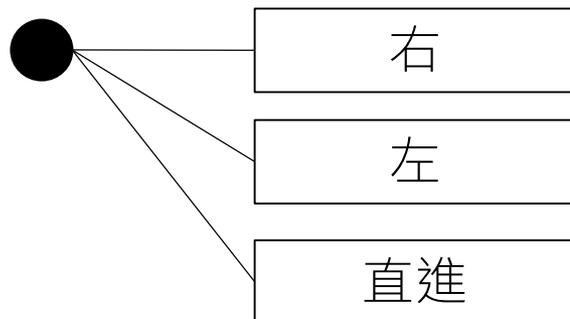
2015/9/19(土)

浦田 淳司

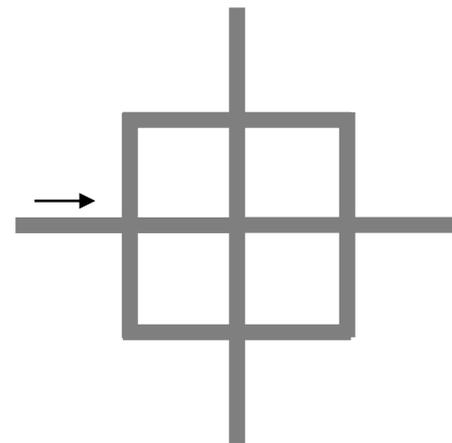
# なぜ、動学問題が難しいか.

## 静学問題

時刻 $t_0$

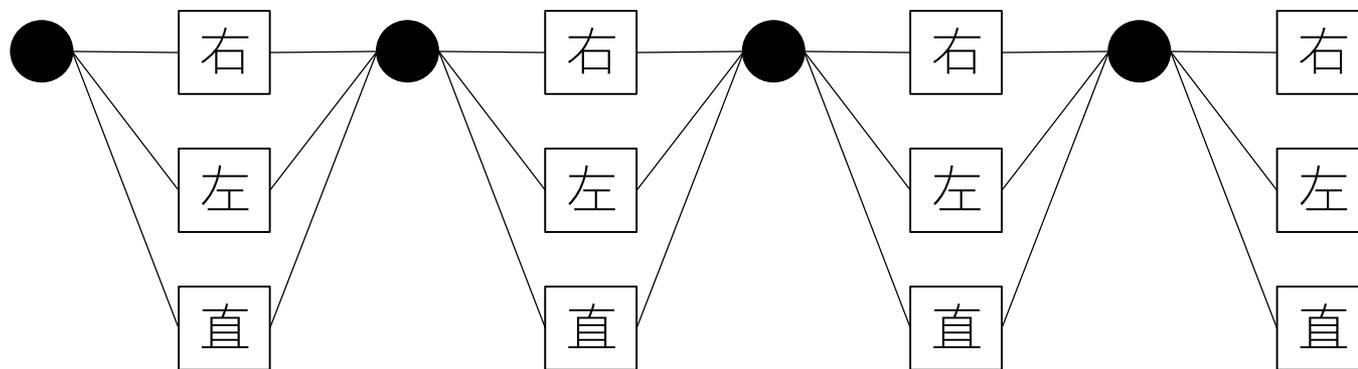


3選択肢



## 動学問題

時刻 $t$



$3^4$ 選択肢

# なぜ、最適化か.

## 最尤法

モデルパラメータによる現象分析→現象の再現

$$\max_{\theta} \prod_i^N P_i(d_i|x_i; \theta)$$

$d$  選択肢  
 $x$  属性  
 $\theta$  パラメータ  
 $i$  個人  
 $N$  集団  
 $P$  選択確率

## 計画＝目的関数の最適化

数理的に計画を捉える

$$\max_{\mu} V(x, \mu)$$

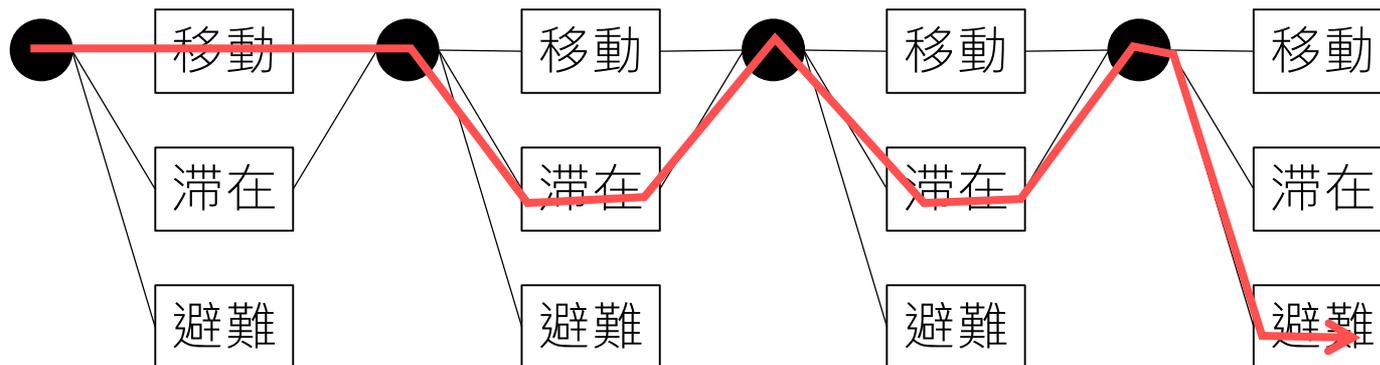
$V$  目的関数  
 $x$  状態・属性  
 $\mu$  制御変数

# 本日の構成

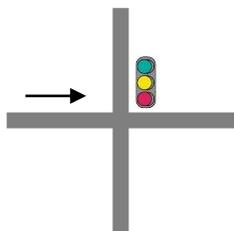
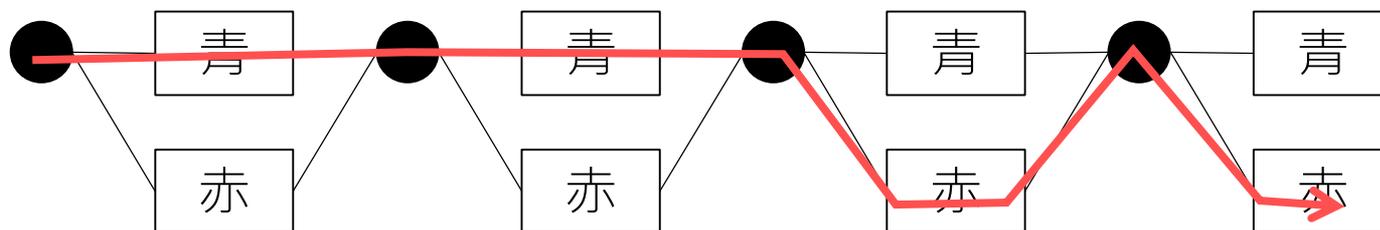
1. はじめに
2. 動学的最適化 ( 動学問題の最適化の定式化と解法 )
  - 2.1 変分法
  - 2.2 最大値原理
  - 2.3 動的計画法 (動的離散選択モデル)
3. 動的離散選択モデルにおけるパラメータ推定
  - 3.1 NFXP
  - 3.2 NPL (擬似最尤推定)
  - 3.3 MPEC型
4. おまけ

# 動学的最適化

時刻  $t$



時刻  $t$



※制御変数が連続量の場合もある

# 動学的最適化へのアプローチ

どう解くか・

1. 変分法 ( Newton, I(1687) )

$$\max_y V(y) = \int_0^T F(t, y(t), y'(t))dt \quad \text{subject to } y(0) = A, \quad y(T) = Z$$

2. 最適制御理論 ( Pontryagin · L.S. et al.(1962) )

$$\max_{\mu} V(y) = \int_0^T F(t, y(t), \mu(t))dt \quad \text{subject to } y(0) = A, \quad y'(t) = f(t, y(t), \mu(t))$$

3. 動的計画法 ( Bellman, R.E.(1953) )

$$\max_{\mu_t} V(y(t)) = u_{\mu_t}(y(t)) + \beta \int V(y(t+1))p(dy(t+1) | y(t), \mu_t)$$

$V, F$ : 目的関数,  $y$ : 状態(属性),  $\mu$ : 制御変数,  
 $f$ : 運動方程式,  $p$ : 状態の推移確率,  $A, Z$ : 端点

# 変分法の概要

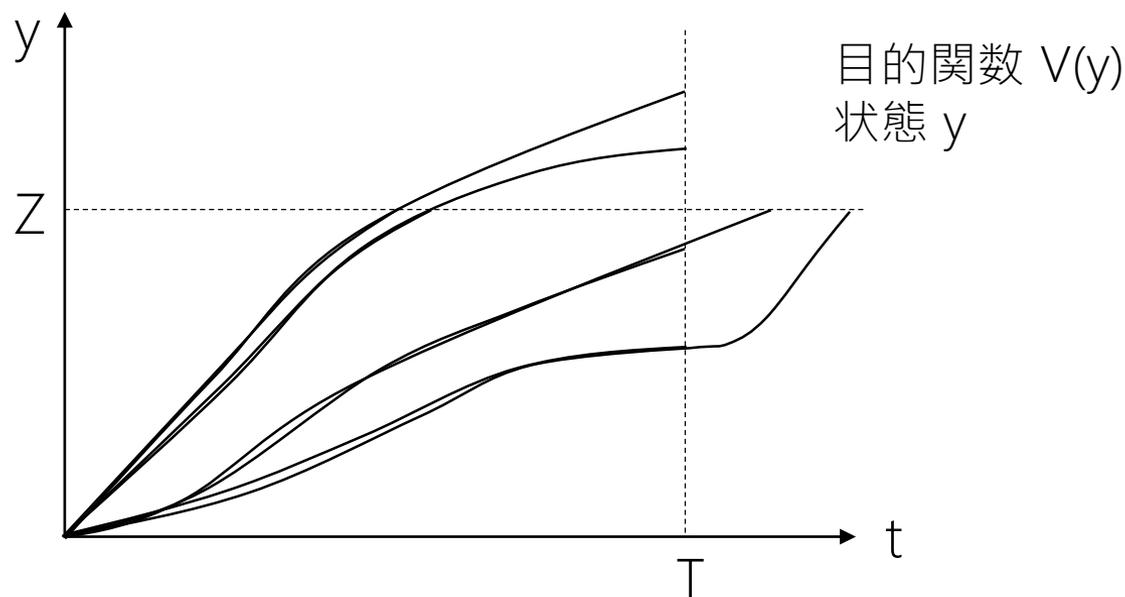
## 1. 変分法 ( Newton, I(1687) )

- 最適経路の決定 (最速降下曲線など cf.ある時間軸上でどの状態に推移するのが最適か)
- 終端Tを自由とすることも可能
- 全ての関数が連続であることが必要

・ 曲線 $y(t)$ を求める問題

→制御のスイッチング(非連続制御)を表せない

$$\max_y V(y) = \int_0^T F(t, y(t), y'(t)) dt \quad \text{subject to } y(0) = A, \quad y(T) = Z$$



# 解の必要条件(オイラー方程式)

$$\max_y V(y) = \int_0^T F(t, y(t), y'(t)) dt \quad \text{subject to } y(0) = A, \quad y(T) = Z$$

<オイラー方程式の導出>

$y(t) = y^*(t) + \varepsilon p(t)$  ( $y^*(t)$ : 最適経路,  $p(t)$ は任意, ただし $p(0)=p(T)=0$ ) とおく

$y^*(t)$ が最適経路であるためには次が必要

$$\frac{dV}{d\varepsilon} (\varepsilon = 0) = 0 \quad (1)$$

---

$$V(\varepsilon) = \int_0^T F[t, y^* + \varepsilon p(t), y'^* + \varepsilon p'(t)] dt \quad (2)$$

$$\frac{dV}{d\varepsilon} (\varepsilon) = \int_0^T \frac{dF}{d\varepsilon} dt = \int_0^T \left( \frac{dF}{dy} \frac{dy}{d\varepsilon} + \frac{dF}{dy'} \frac{dy'}{d\varepsilon} \right) dt = \int_0^T (F_y p(t) + F_{y'} p'(t)) dt \quad (3)$$

$$\int_0^T F_y p(t) dt + \int_0^T F_{y'} p'(t) dt = 0 \quad (4)$$

# (続き)オイラー方程式の導出

$$\int_0^T F_{y'} p'(t) dt = [F_{y'} p(t)]_0^T - \int_0^T p(t) \frac{d}{dt} F_{y'} dt = - \int_0^T p(t) \frac{d}{dt} F_{y'} dt \quad (5)$$

(4)より

$$\int_0^T p(t) \left( F_y - \frac{d}{dt} F_{y'} \right) dt = 0 \quad (4)'$$

$p(t)$ は任意より, 全ての $t$ で次が成り立つ

$$\left( F_y - \frac{d}{dt} F_{y'} \right) = 0$$

また,

$$\frac{d}{dt} F_{y'} = \frac{\partial F_{y'}}{\partial t} + \frac{\partial F_{y'}}{\partial y} \frac{dy}{dt} + \frac{\partial F_{y'}}{\partial y'} \frac{dy'}{dt}$$

より, オイラー方程式は次となる

$$F_{y'y'} y''(t) + F_{yy'} y'(t) + F_{ty'} - F_y = 0, \forall t \in [0, T]$$

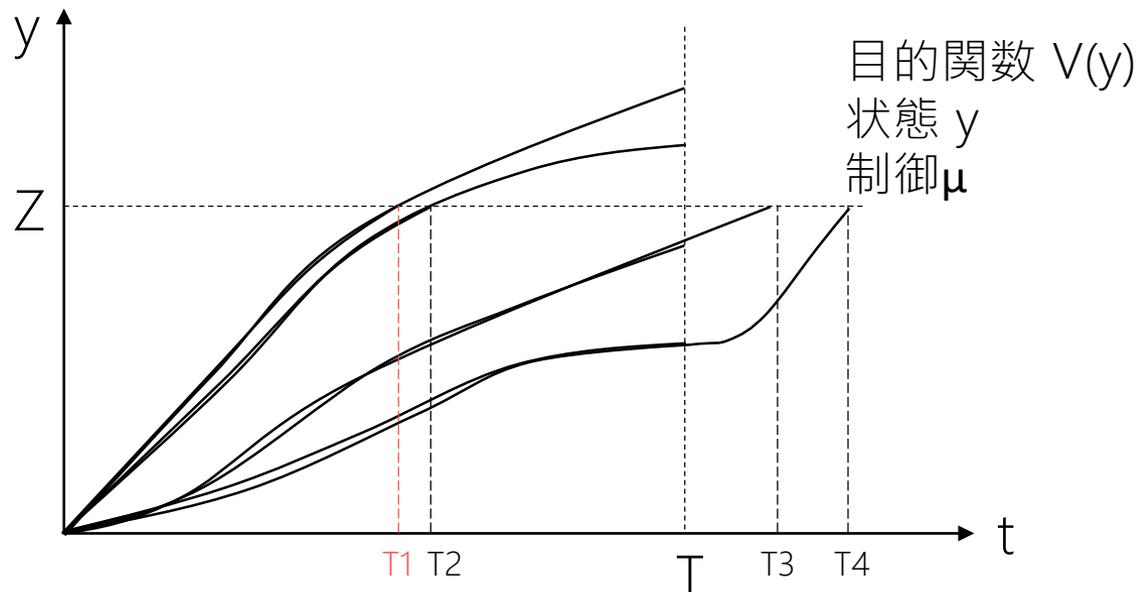
# 動的最適制御の解法(最適制御理論)

どう解くか・

2. 最適制御理論 ( Pontryagin, L.S. et al.(1962) )

- 変分法制御変数 $\mu(t)$ の問題 → 非連続な制御(スイッチング)に対応
- 各時間帯の最適軌跡を解析 → 終端時刻 $T$  は自由でも可 → 時間最適化問題

$$\max_{\mu} V(y) = \int_0^T F(t, y(t), \mu(t)) dt \quad \text{subject to } y(0) = A, \quad y'(t) = f(t, y(t), \mu(t))$$



# 最適制御理論 (Dynamic Control Problem)

状態変数  $y(t) = \{y_1, \dots, y_i, \dots, y_n\}(t)$  連続 (区分的に微分可能)  
制御変数  $\mu(t) = \{\mu_1, \dots, \mu_i, \dots, \mu_n\}(t)$  非連続可 (区分的に連続)

目的関数  $\max V = \max_{\mu} \int_0^T F(t, y, \mu) dt$

制約条件

運動方程式  $\frac{dy_i}{dt} = f^i(t, y(t), \mu(t))$  (yは初期条件と運動方程式で決定)

初期条件  $y(0)$

終端条件  $y(T)$  自由

制御集合  $\mu(t) \in U$

等価  
最大値原理 (ハミルトニアンHの最大化)

$H \equiv F(t, y, \mu) + \sum_j s_j f^j(t, y, \mu)$

$\max_{\mu} H(t, y, \mu, s) \quad \forall t$

←時間積分を各時間帯の最適化計算に分解

yについての運動方程式  $\frac{dy_i}{dt} = \frac{dH}{ds}$

$\lambda$ についての運動方程式  $\frac{ds_i}{dt} = \frac{dH}{dy}$

横断性条件  $s(t) = 0, H(T) = 0$

# 最適制御理論の使い方の例

Friesz, T.L., Luque, J., Tobin, R.L., Wie, B-W.:  
Dynamic network traffic assignment  
considered as a continuous time optimal control problem,  
Operations Research, Vol. 37(6), pp. 893–901, 1989.



Nonlinear Pricing and Revenue Optimization  
Freight Systems and Logistics  
Dynamic Network Games and Dynamic Traffic Assignment  
Network Design and MPECs

2014/6/27(金)  
集中理論談話会 #4  
D2 浦田 淳司

# 研究概要

研究概要：

- ・ 動的交通配分の定式化(各リンクでの流入量の最適制御)
- ・ Pontryaginの最大値原理の導入
- ・ システム最適解を求める

目的関数：全時間帯全リンクでの旅行時間最小化

Notation:

$G(N, A)$ : ノード $N$ , リンク $A$ で形成されるネットワーク

ノード $N$ のうち $n$ を終着点ノード,  $M=\{1,2,\dots,n-1\}$ は発ノード.

$t$ : 時刻  $t \in [0, T]$

$x_a(t)$ : 時刻 $t$ でのリンク $a$ 上の交通量 = 状態変数

$C_a[x_a(t)]$ : 交通量 $x_a(t)$ のときの旅行時間コスト

$u_a(t)$ : リンク $a$ への流入交通量 = 制御変数

$g_a[x_a(t)]$ : リンク $a$ からの流出交通量

# (制約条件)交通量保存則, 基本条件

$S_k(t)$ : 時刻 $t$ でのノード $k$ からの発生交通量

$A(k)$ : ノード $k$ が着点となるリンク集合

$B(k)$ : ノード $k$ が発点となるリンク集合

$$\begin{array}{l} \text{交通量保存則} \\ \text{(制約条件)} \end{array} \quad S_k(t) = \sum_{a \in A(k)} u_a(t) - \sum_{a \in B(k)} g_a[x_a(t)] \quad \forall k \in M, \forall t \quad (2)$$

$$\begin{array}{l} \text{基本条件} \\ \text{(制約条件)} \end{array} \quad x_a(0) = x_a^0 \quad \forall a \quad (3)$$

$$u_a(t) \geq 0 \quad \forall a, \forall t \quad (4)$$

$$x_a(t) \geq 0 \quad \forall a, \forall t \quad (5)$$

$$\text{定義} \quad \Omega \equiv \{(x, u) : (1), (2), (3), (4) \text{ are satisfied}\} \quad (6)$$

# 目的関数とハミルトニアン

システム最適の目的関数 
$$\text{minimize } J_1 = \sum_{a \in A} \int_0^T C_a(x_a) dt \quad (7)$$
  
$$\text{subject to } (x, u) \in \Omega$$

(全期間全リンクの旅行時間の合計の最小化)

ハミルトニアンの定義 
$$H(x, u, \tau, t) = \sum_{a \in A} C_a(x_a) + \sum_{a \in A} \tau_a [u_a - g_a(x_a)] \quad (8)$$

$\tau_a$ : 随伴変数(状態方程式に対するラグランジュ乗数)

- (8)式が解を持つには, Arrow-Kurzの十分定理より, Hが凸であることが必要
- Hが凸であるためには,  $\tau_a \geq 0$  ( $\forall a$ )であることが求められる.

# ラグランジアン・KKT条件

制御変数の制約条件(2)(4)を考慮して, Hの最小化問題に対し, ラグランジアンを設定

$$L = H + \sum_{k \in M} \mu_k \left[ S_k - \sum_{a \in A(k)} u_a + \sum_{a \in B(k)} g_a(x_a) \right] + \sum_{a \in A} \beta_a [-u_a] \quad (11)$$

Lの最小化問題

↕ 必要十分条件

KKT条件

$\mu_k, \beta_a$ : ラグランジュ乗数

$$\tau_a - \mu_k - \beta_a = 0 \quad \forall a \in A(k), \forall k \in M, \forall t \in [0, T] \quad (12)$$

$$\beta_a \geq 0, u_a \geq 0, \beta_a u_a = 0 \quad \forall a \in A, \forall t \in [0, T] \quad (13)$$

$$\mu_k \geq 0, \mu_k \left[ S_k - \sum_{a \in A(k)} u_a + \sum_{a \in B(k)} g_a(x_a) \right] = 0 \quad \forall k \in M, \forall t \in [0, T] \quad (14)$$

Eular-Lagrange方程式より

$$-\dot{\tau}_a = \frac{\partial L}{\partial x_a} \quad \forall a \in A, \forall t \in [0, T] \quad (15)$$

$$-\dot{\tau}_a = C'_a - \tau_a g'_a + \mu_k g'_a \quad \forall a \in B(k), \forall k \in M, \forall t \in [0, T] \quad (16)$$

横断性条件

$$\tau_a(T) = 0 \quad \forall a \in A \quad (17)$$

※  $\dot{\cdot}$  はt微分, 'はx微分を示す

# 最適制御問題の必要条件の再定式化

$$(12) \quad (13) \text{ より} \quad \tau_a \geq \mu_k \quad \forall a \in A(k), \forall k \in M, \forall t \in [0, T] \quad (18)$$

$$(17) \text{ より} \quad \tau_a(T) = 0 \quad \forall a \in A(k) \quad (19)$$

$$(16) \text{ より} \quad -\dot{\tau}_a = C'_a - \tau_a g'_a + \mu_k g'_a \quad \forall a \in B(k), \forall k \in M, \forall t \in [0, T] \quad (20)$$

$$(12) \quad (13) \text{ より} \quad u_a(\tau_a - \mu_k) = 0 \quad \forall k, a, t \quad \forall a \in A(k), \forall k \in M, \forall t \in [0, T] \quad (21)$$

(14)と(18)より,  $\tau_a \geq 0$  が成立するため,  $H$ は凸であり, 解を持つ.

(18)(21)より, 制御変数 $u_a$ は $(\tau_a - \mu_k) = 0$ のときに0でない値をとる.  
これを求めたい.

# 制御変数の算出

$(\tau_a - \mu_k) = 0$ が $\forall a, k$ で成立することが必要

$$\dot{\tau}_a(t) = \dot{\mu}_k(t) \quad \ddot{\tau}_a(t) = \ddot{\mu}_k(t) \quad \text{for } t \in [t_1, t_2] \subseteq [0, T] \quad (22)$$

(22)を(20)に代入

$$(\tau_a - \mu_l)g'_a - C'_a - \dot{\mu}_k = 0 \quad (23)$$

$$(\dot{\tau}_a - \dot{\mu}_l)g'_a + (\tau_a - \mu_l)g''_a \dot{x}_a - C''_a \dot{x}_a - \ddot{\mu}_k = 0 \quad (24) \quad ((23)をxで微分)$$

(22)(24)(1)より

$$u_a = \frac{[(\mu_k - \mu_l)g''_a - C''_a]g_a - (\dot{\mu}_k - \dot{\mu}_l)g'_a + \ddot{\mu}_k}{(\mu_k - \mu_l)g''_a - C''_a} \quad (26)$$

(26)式または0で, 制御変数は与えられる.

# 動的計画法(動的離散選択モデル)

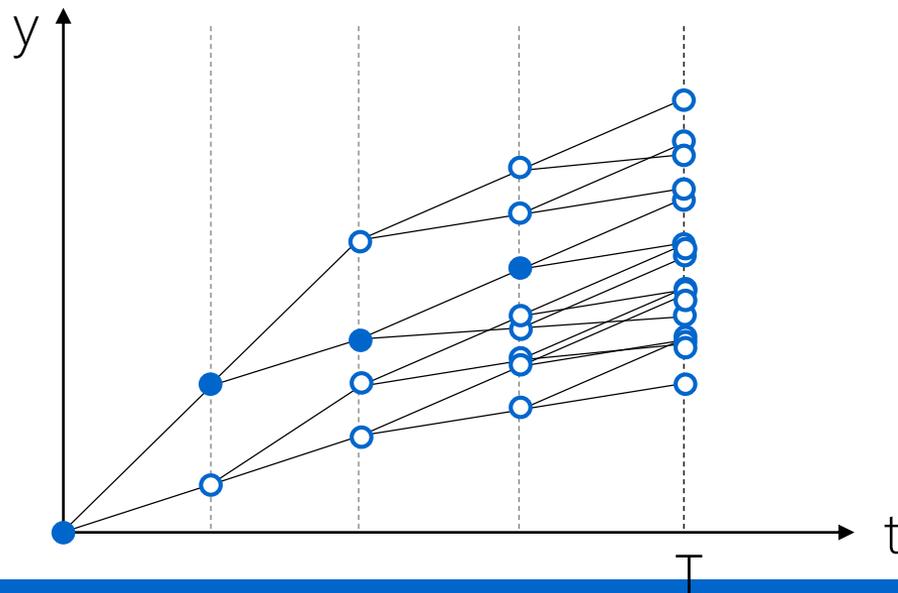
どう解くか .

## 3. 動的計画法 ( Bellman, R.E.(1953) )

- 多段階意思決定モデル
- 再帰的計算過程が必要
  - ・ 将来状態の計算
  - ・ 状態推移を確率的に与えることも可能
  - ・ 終端時刻  $T$  が決まっている必要

$$\max_{\mu_t} V(y(t)) = u_{\mu_t}(y(t)) + \beta \int V(y(t+1)) p(dy(t+1) | y(t), \mu_t)$$

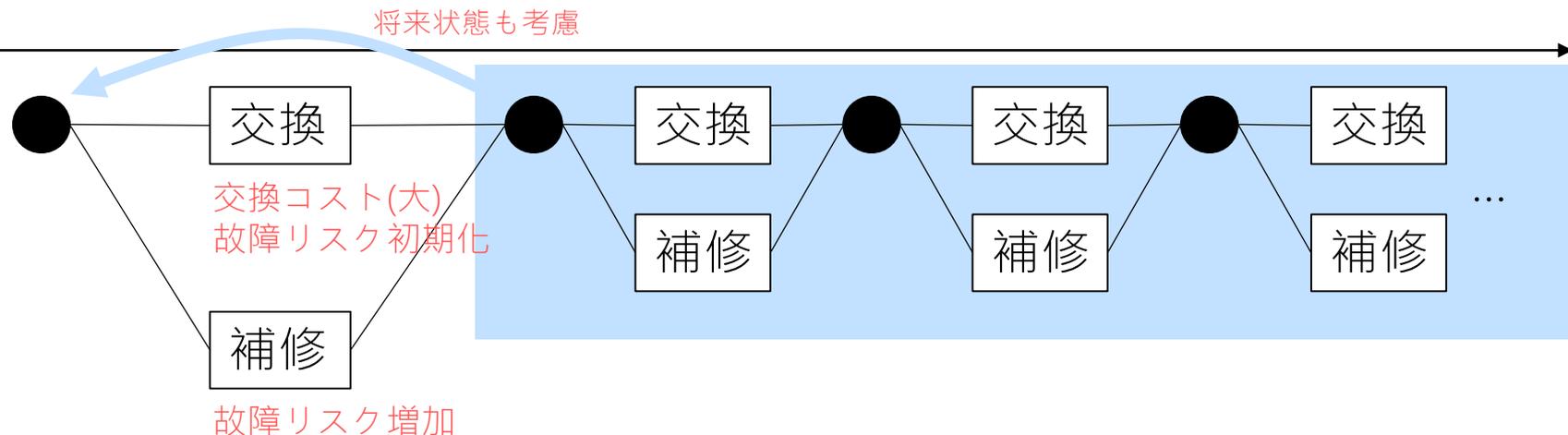
目的関数  $V(y)$   
状態  $y$   
制御  $\mu$



# 動的離散選択モデル(概要)

Rust, J.(1987) のバスエンジン補修/交換の問題を例に説明

時刻t



ある期の効用関数u

$$u(x, d, \varepsilon; \theta_1, RC) = v(x, d; \theta_1, RC) + \varepsilon(d)$$

$$v(x, d; \theta_1, RC) = \begin{cases} -c(x; \theta_1), & \text{if } d = 0 \\ -RC - c(0; \theta_1), & \text{if } d = 1 \end{cases}$$

d=1: 交換, d=0: 補修のみ

x: バスエンジンの状態

$\varepsilon$ : 非観測項

c: 補修コスト

$\theta_1$ : 補修パラメータ

RC: 交換コスト

$\theta_2, \theta_3$ : 状態推移パラメータ

状態xの推移確率(一次マルコフ性を仮定)

$$p(x_{t+1}, \varepsilon_{t+1} | x_t, \varepsilon_t, d_t; \theta_2, \theta_3)$$

# 動的離散選択モデル(定式化)

将来の価値関数は、時間割引を考慮した効用の最大化となる

$$V(x_t, \varepsilon_t) = \max_{\{d_t, d_{t+1}, d_{t+2}, \dots\}} \mathbb{E} \left[ \sum_{\tau=t}^{\infty} \beta^{\tau-t} u(x_\tau, d_\tau, \varepsilon_\tau; \theta_1, \text{RC}) \right] \quad (6)$$

時間割引率  $\beta \in (0,1)$

最適意思決定は無限期間先までを考慮しており、期によらず一定である

$$V(x, \varepsilon) = \max_d \left\{ \begin{array}{l} \text{今期の効用} \\ \nu(x, d; \theta_1, \text{RC}) + \varepsilon(d) \\ \text{次期の効用} \quad \text{今期の選択後の次期状態の推移確率} \\ + \beta \int_{x'} \int_{\varepsilon'} V(x', \varepsilon') p(x', \varepsilon' | x, \varepsilon, d; \theta_2, \theta_3) dx' d\varepsilon' \end{array} \right\} \quad (7)$$

次期の期待価値関数 次期状態  $(x', \varepsilon')$

- 今期と次期の状態により、価値関数を定義
- 次期の期待価値関数の中に、次々期以降の効用は入っている

# 期待値関数・選択確率

仮定: 推移確率の条件つき独立性(CI)

$$p(x', \varepsilon' | x, \varepsilon, d; \theta_2, \theta_3) = p_2(\varepsilon' | x'; \theta_2) p_3(x' | x, d; \theta_3)$$

CIの仮定を用いて, 期待値関数を次で定義する.

$$EV(x) = \int_{\varepsilon} V(x, \varepsilon) p_2(\varepsilon | x; \theta_2) d\varepsilon$$

選択肢の期待値関数((7)式から $\varepsilon$ を除いて)

$$EV(x, d) = v(x, d; \theta_1, RC) + \varepsilon(d) + \beta \int_{x'} EV(x') p_3(x' | x, d; \theta_3) dx'$$

これを期待値関数の定義式に戻す

$$EV(x) = \int_{\varepsilon'} \max_d \{EV(x, d)\} p_2(\varepsilon' | x; \theta_2) d\varepsilon' \quad (8)$$

$\varepsilon$ を極値分布とし, ロジット型の条件付き選択確率を導出( $\theta = (RC, \theta_1, \theta_3)$ がパラメータ)

$$P(d | x; \theta) = \frac{\exp[v(x, d; \theta_1, RC) + \beta EV(x, d)]}{\sum_{d' \in \{0,1\}} \exp[v(x, d'; \theta_1, RC) + \beta EV(x, d')]} \quad (9)$$

# 最尤法によるパラメータ推定

バス会社*i*の尤度  $\ell_i(X^i; \theta) = \prod_{t=2}^T P(d_t^i | x_t^i; \theta) p_3(x_t^i | x_{t-1}^i, d_{t-1}^i; \theta_3)$

M社の全体尤度  $\ell(\theta) = \prod_{i=1}^M \ell_i(X^i; \theta) = \prod_{i=1}^M \prod_{t=2}^T P(d_t^i | x_t^i; \theta) p_3(x_t^i | x_{t-1}^i, d_{t-1}^i; \theta_3)$  (14)

対数尤度  $L(\theta) = \log \ell(\theta)$

$$= \sum_{i=1}^M \sum_{t=2}^T \log[P(d_t^i | x_t^i; \theta)] + \sum_{i=1}^M \sum_{t=2}^T \log[p_3(x_t^i | x_{t-1}^i, d_{t-1}^i; \theta_3)] \quad (10)$$

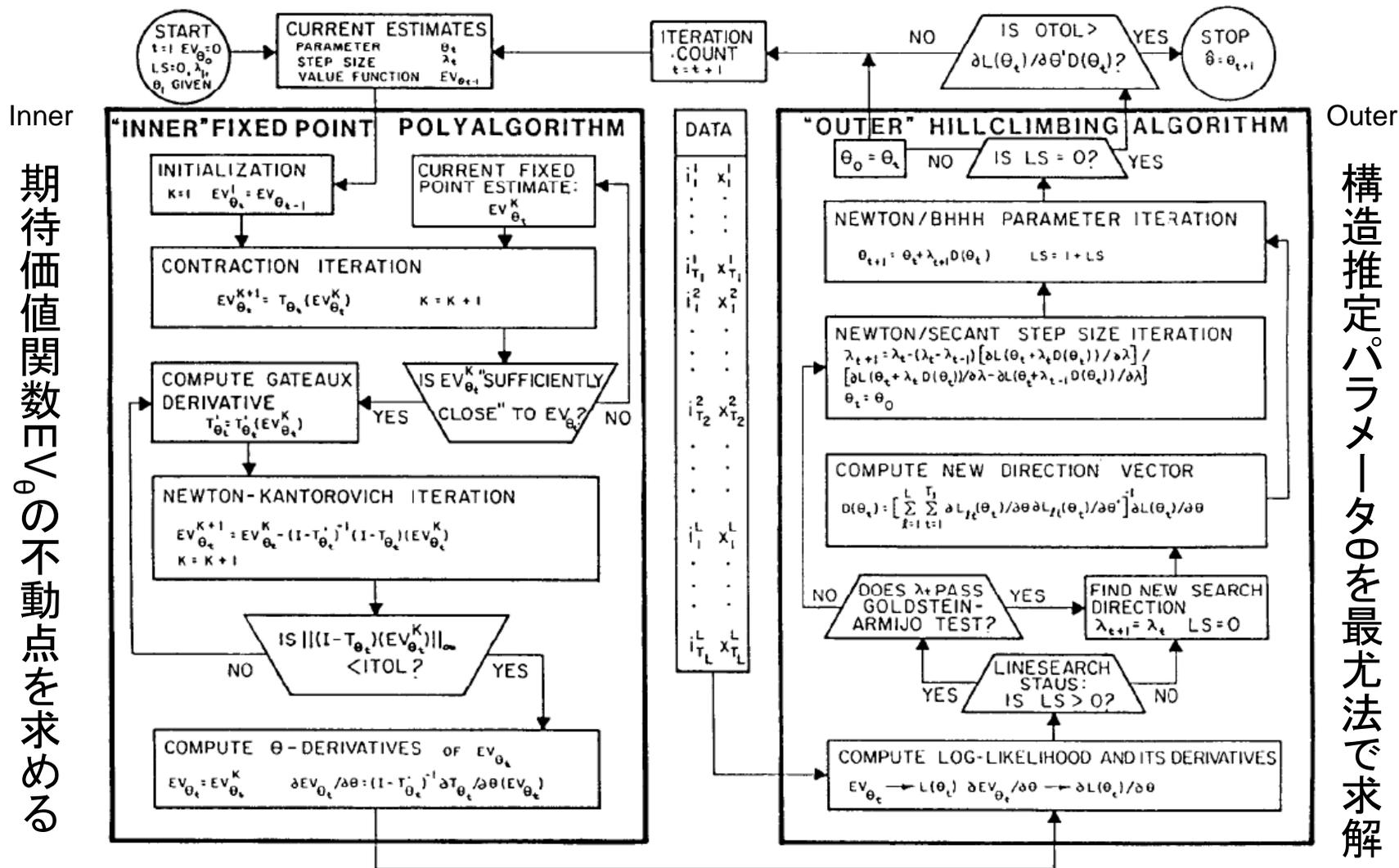
対数尤度最大化により、 $\theta = (RC, \theta_1, \theta_3)$ を求める

$$\max_{\theta} \frac{1}{M} L(\theta) \quad (11)$$

# 推定方法1: NFXP法

Rust, J.(1988) Statistical Models of Discrete Choice Processes, Transportation Research Part B, Vol. 22(2), pp. 125-158.

## NESTED FIXED POINT MAXIMUM LIKELIHOOD ALGORITHM



Inner  
期待値関数  $E V_{\theta}$  の不動点を求める

Outer  
構造推定パラメータ  $\theta$  を最尤法で求解

# NFXP法：Fixed-Point Solution

(7)式と極値分布により, Fixed-Point equationは次となる

$$\begin{aligned} EV(x, d) = & \int_{x'=0}^{\infty} \log \left\{ \sum_{d' \in \{0,1\}} \exp[\nu(x', d'; \theta_1, RC) + \beta EV(x', d')] \right\} \\ & \times p_3(dx'|x, d, \theta_3). \end{aligned} \quad (12)$$

今期の状態 $x$ を $K$ 分割, 次期の状態 $x'$ を $J$ 分割して, 表記.

$$p_3(x'|x_k, d, \theta_3) = \begin{cases} \Pr\{x' = \hat{x}_{k+j} | \theta_3\}, & \text{if } d = 0 \\ \Pr\{x' = \hat{x}_{1+j} | \theta_3\}, & \text{if } d = 1 \end{cases} \quad (13)$$

(11)式に代入

$$\begin{aligned} EV(\hat{x}_k, d) = & \sum_{j=0}^J \log \left\{ \sum_{d' \in \{0,1\}} \exp[\nu(x', d'; \theta_1, RC) + \beta EV(x', d')] \right\} \\ & \times p_3(x' | \hat{x}_k, d, \theta_3). \end{aligned} \quad (14)$$

※ この式が定義されている  $[EV = [EV(\hat{x}_1, 0), \dots, EV(\hat{x}_K, 0), EV(\hat{x}_1, 1), \dots, EV(\hat{x}_K, 1)]]$

式を簡略化

$$EV = T(EV, \theta) \quad (15)$$

# NFXP法: inner algorithm

$\theta$ 固定の下で, 不動点  $EV_\theta = T_\theta(EV_\theta)$ (15)' を求める  
(15)は収縮写像であり, 次が成立する.

$$\|T_\theta(W) - T_\theta(V)\| \leq \beta \|W - V\| \quad (16)$$

この性質から,  $EV_k$ を次で更新する.

$$EV_{k+1} = T_\theta(EV_k) \quad (17)$$

つまり, 下記となる.

$$\begin{aligned} |T_\theta(EV_{k+1}) - T_\theta(EV_k)| &\leq \beta |EV_{k+1} - EV_k| \\ \Leftrightarrow |EV_{k+2} - EV_{k+1}| &\leq \beta |EV_{k+1} - EV_k| \end{aligned} \quad (18)$$

$\beta$ の速度で $EV_\theta$ に近づく.

ただし,  $EV_\theta$ を得るためには,  $k=\infty$ が必要であり, 不動点近傍では収縮が遅くなる.  
不動点近傍では, 次式を用いる(Newton-Kantorovich法).

$$0 = (I - T_\theta)EV_{k+1} \sim (I - T_\theta)EV_k + (I - T'_\theta)(EV_{k+1} - EV_k) \quad (19)$$

$$EV_{k+1} = EV_k - (I - T'_\theta)^{-1}(I - T_\theta)EV_k \quad (20)$$

# 推定法2:NPL(擬似最尤推定法)

Step1:  $EV_0$ をランダムに与える

Step2:  $EV_k$ を(9)式の右辺に代入し, 尤度最大化するパラメータ $\theta_k$ を求める

$$P(d|x; \theta) = \frac{\exp[\nu(x, d; \theta_1, RC) + \beta EV(x, d)]}{\sum_{d' \in \{0,1\}} \exp[\nu(x, d'; \theta_1, RC) + \beta EV(x, d')]} \quad (9)$$

Step3:  $EV_k$ と $\theta_k$ を用いて, (14)式より $EV_{k+1}$ を求める

$$EV(\hat{x}_k, d) = \sum_{j=0}^J \log \left\{ \sum_{d' \in \{0,1\}} \exp[\nu(x', d'; \theta_1, RC) + \beta EV(x', d')] \right\} \quad (14)$$
$$\times p_3(x'|\hat{x}_k, d, \theta_3).$$

収束判定:  $|EV_{k+1} - EV_k|$ と $|\theta_{k+1} - \theta_k|$ が十分小さければ収束  
収束していなければ, step2に戻る

無限回の繰り返し計算によって不動点を得ることができ, NFXPと漸近等価

# NPL(擬似最尤推定法)とNFXPの比較

Aguirregabiria and Mira(2002)

TABLE I  
MONTE CARLO EXPERIMENT

Experiment design	
Model:	Bus engine replacement model (Rust)
Parameters:	$\theta_0 = 10.47$ ; $\theta_1 = 0.58$ ; $\beta = 0.9999$
State space:	201 cells
Number observations:	1000
Number replications:	1000
Initial probabilities:	Kernel estimates

Monte Carlo distribution of MLE  
(In parenthesis, percentages over the true value of the parameter)

	$\theta_0$	$\theta_1$
Mean Absolute Error:	2.08 (19.9%)	0.17 (29.0%)
Median Absolute Error:	1.56 (14.9%)	0.13 (22.7%)
Std. dev. estimator:	2.24 (21.4%)	0.16 (26.9%)
Policy iterations (avg.):	6.2	

Monte Carlo distribution of PI estimators (relative to MLE)  
(All entries are 100\* (K-PI statistic-MLE statistic)/MLE statistic)

Parameter	Statistics	Estimators		
		1-PI	2-PI	3-PI
$\theta_0$	Mean AE	4.7%	1.6%	0.3%
	Median AE	14.2%	0.2%	-0.3%
	Std. dev.	6.8%	1.2%	0.3%
$\theta_1$	Mean AE	18.7%	1.5%	0.2%
	Median AE	25.1%	0.7%	0.6%
	Std. dev.	11.0%	1.3%	0.2%

・計算速度はパラメータが4つの場合は、NPLは9倍速い

*Nested Pseudo Likelihood Algorithm (NPL):*

Let  $\hat{\theta}_f$  be an estimate of  $\theta_f$ . Start with an initial guess for the conditional choice probabilities,  $P^0 \in [0, 1]^{MJ}$ . At iteration  $K \geq 1$ , apply the following steps:

*Step 1:* Obtain a new pseudo-likelihood estimate of  $\alpha$ ,  $\alpha^K$ , as

$$(11) \quad \alpha^K = \arg \max_{\alpha \in \Theta} \sum_{i=1}^n \ln \Psi_{(\alpha, \hat{\theta}_f)}(P^{K-1})(a_i | x_i)$$

where  $\Psi_{\theta}(P)(a|x)$  is the element  $(a, x)$  of  $\Psi_{\theta}(P)$ .

*Step 2:* Update  $P$  using the 'arg max' from step 2, i.e.

$$(12) \quad P^K = \Psi_{(\alpha^K, \hat{\theta}_f)}(P^{K-1}).$$

Iterate in  $K$  until convergence in  $P$  (and  $\alpha$ ) is reached.

MPEC (Mathematical Programming with Equilibrium Constraints)を用いた求解

(11)式, (15)式を等価な均衡制約条件付き最適化問題と定式化.

(等価であることの証明は定理1より)

$$\max_{(\theta, EV)} \frac{1}{M} \mathcal{L}(\theta, EV) \quad (11)$$

$$\text{subject to } EV = T(EV, \theta) \quad (15)$$

NFXPが不動点の算出プロセスを毎回解いているのに比べて,  
Bellmanの等式(15)を一度評価すればよいので, 計算負荷は小さい.

観測可能な状態  $x$

選択結果  $d$

意思決定者  $i$  ( $1 \leq i \leq M$ )

構造パラメータ  $\theta$ : コスト, 利益などを示す

内生変数  $\sigma$ : 意思決定者の政策関数. 構造パラメータ  $\theta$  の関数となる.

$\sigma$  は均衡条件を通じて,  $\theta$  に拠る.

$$h(\theta, \sigma) = 0 \quad (21)$$

$\theta$  を所与としたときの,  $\Sigma(\theta)$  を (1) 式を満たす  $\sigma$  の集合と定義する

$$\Sigma(\theta) := \{\sigma: h(\theta, \sigma) = 0\}$$

$\hat{\sigma}(\theta)$  は  $\Sigma(\theta)$  の要素であり, 期待価値関数とし, 一意に定まる.

また,  $\Sigma(\theta)$  は  $\theta$  と 1 対 1 対応する.

$L(\theta, \hat{\sigma}(\theta); X)$  を観測データ  $X$  の対数尤度とし, 尤度最大化パラメータは次と定義

$$\hat{\theta} = \operatorname{argmax}_{\theta} \frac{1}{M} \left\{ \max_{\hat{\sigma}(\theta) \in \Sigma(\theta)} L(\theta, \hat{\sigma}(\theta); X) \right\} \quad (22)$$

# MPECによる定式化

MPEC側

構造パラメータ $\theta$ を推定する際, 内生変数 $\sigma$ が $\theta$ から算出されたものであればよい  
⇒均衡制約条件(21)を用いた制約付き最適化問題として定義する

$$\max_{(\theta, \sigma)} \frac{1}{M} L(\theta, \sigma; X) \quad \text{subject to } h(\theta, \sigma) = 0 \quad (23)$$

定理1: (22)式と(23)式の等価性を示す

$\hat{\theta}$ は尤度最大化の(22)式で定義され, (23)式の解は $(\bar{\theta}, \bar{\sigma})$ とする.

$\hat{\sigma}^*(\theta) = \arg \max_{\hat{\sigma}(\theta)} L(\theta, \hat{\sigma}(\theta))$ と定義する.

また,  $L(\hat{\theta}, \hat{\sigma}^*(\hat{\theta})) = L(\bar{\theta}, \bar{\sigma})$ となる. このとき,  $\hat{\theta} = \bar{\theta}$ となる.

証明:

$(\bar{\theta}, \bar{\sigma})$ は(21)式を満足し,  $\bar{\sigma} \in \Sigma(\bar{\theta})$ であるため, (22)式より,  $L(\hat{\theta}, \hat{\sigma}^*(\hat{\theta})) \geq L(\bar{\theta}, \bar{\sigma})$ .

$\hat{\sigma}^*(\hat{\theta}) \in \Sigma(\hat{\theta})$ であるため, 制約条件(21)を満足し, (23)式より  $L(\hat{\theta}, \hat{\sigma}^*(\hat{\theta})) \leq L(\bar{\theta}, \bar{\sigma})$ .

2式を同時に満たす場合は等式となり, そ解は1つである.そのため,  $\hat{\theta} = \bar{\theta}$ となる.

# 計算精度の比較

$\beta$	Implementation	True values:	Parameters						MSE
			RC	$\theta_{11}$	$\theta_{30}$	$\theta_{31}$	$\theta_{32}$	$\theta_{33}$	
0.975	MPEC/AMPL	Mean	12.212	2.607	0.0943	0.4473	0.4454	0.0127	3.111
		Std. dev.	(1.613)	(0.500)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-
	MPEC/MATLAB	Mean	12.212	2.607	0.0943	0.4473	0.4454	0.0127	3.111
		Std. dev.	(1.613)	(0.500)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-
	NFXP/MATLAB	Mean	12.213	2.606	0.0943	0.4473	0.4445	0.0127	3.123
		Std. dev.	(1.617)	(0.500)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-
0.980	MPEC/AMPL	Mean	12.134	2.578	0.0943	0.4473	0.4455	0.0127	2.857
		Std. dev.	(1.570)	(0.458)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	-
	MPEC/MATLAB	Mean	12.134	2.578	0.0943	0.4473	0.4455	0.0127	2.857
		Std. dev.	(1.570)	(0.458)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	-
	NFXP/MATLAB	Mean	12.139	2.579	0.0943	0.4473	0.4455	0.0127	2.866
		Std. dev.	(1.571)	(0.459)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	-
0.985	MPEC/AMPL	Mean	12.013	2.541	0.0943	0.4473	0.4455	0.0127	2.140
		Std. dev.	(1.371)	(0.413)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	-
	MPEC/MATLAB	Mean	12.013	2.541	0.0943	0.4473	0.4455	0.0127	2.140
		Std. dev.	(1.371)	(0.413)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	-
	NFXP/MATLAB	Mean	12.021	2.544	0.0943	0.4473	0.4455	0.0127	2.136
		Std. dev.	(1.368)	(0.411)	(0.0037)	(0.0057)	(0.0060)	(0.0015)	-
0.990	MPEC/AMPL	Mean	11.830	2.486	0.0943	0.4473	0.4455	0.0127	1.880
		Std. dev.	(1.305)	(0.407)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-
	MPEC/MATLAB	Mean	11.830	2.486	0.0943	0.4473	0.4455	0.0127	1.880
		Std. dev.	(1.305)	(0.407)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-
	NFXP/MATLAB	Mean	11.830	2.486	0.0943	0.4473	0.4455	0.0127	1.880
		Std. dev.	(1.305)	(0.407)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-
0.995	MPEC/AMPL	Mean	11.819	2.492	0.0942	0.4473	0.4455	0.0127	1.892
		Std. dev.	(1.308)	(0.414)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-
	MPEC/MATLAB	Mean	11.819	2.492	0.0942	0.4473	0.4455	0.0127	1.892
		Std. dev.	(1.308)	(0.414)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-
	NFXP/MATLAB	Mean	11.819	2.492	0.0942	0.4473	0.4455	0.0127	1.892
		Std. dev.	(1.308)	(0.414)	(0.0036)	(0.0057)	(0.0060)	(0.0015)	-

- 250回計算の平均をとり、パラメータの平均、分散を比較
- $\beta$ が大きい場合は、NFXPと同じ結果であるなど、両者の精度の差はほぼない

<sup>a</sup>For each  $\beta$ , there are 250 replications. Standard deviations are reported in parentheses. MSE is calculated by summing over all structural parameters.

# 計算時間の比較

・MPECのほうがNFXPよりも計算時間は短い(AMPLで180倍以上, MATLABで3倍以上)

$\beta$	Implementation	Runs Converged (out of 1250 runs)	CPU Time (in sec.)	# of Major Iter.	# of Func. Eval.	# of Contraction Mapping Iter.
0.975	MPEC/AMPL	1240	0.13	12.8	17.6	–
	MPEC/MATLAB	1247	7.90	53.0	62.0	–
	NFXP	998	24.60	55.9	189.4	134,748
0.980	MPEC/AMPL	1236	0.15	14.5	21.8	–
	MPEC/MATLAB	1241	8.10	57.4	70.6	–
	NFXP	1000	27.90	55.0	183.8	162,505
0.985	MPEC/AMPL	1235	0.13	13.2	19.7	–
	MPEC/MATLAB	1250	7.50	55.0	62.3	–
	NFXP	952	43.20	61.7	227.3	265,827
0.990	MPEC/AMPL	1161	0.19	18.3	42.2	–
	MPEC/MATLAB	1248	7.50	56.5	65.8	–
	NFXP	935	70.10	66.9	253.8	452,347
0.995	MPEC/AMPL	965	0.14	13.4	21.3	–
	MPEC/MATLAB	1246	7.90	59.6	70.7	–
	NFXP	950	111.60	58.8	214.7	748,487

<sup>a</sup>For each  $\beta$ , we use five starting points for each of the 250 replications. CPU time, number of major iterations, number of function evaluations and number of contraction mapping iterations are the averages for each run.

# 3章

## 完全合理性を緩和した動的避難開始選択



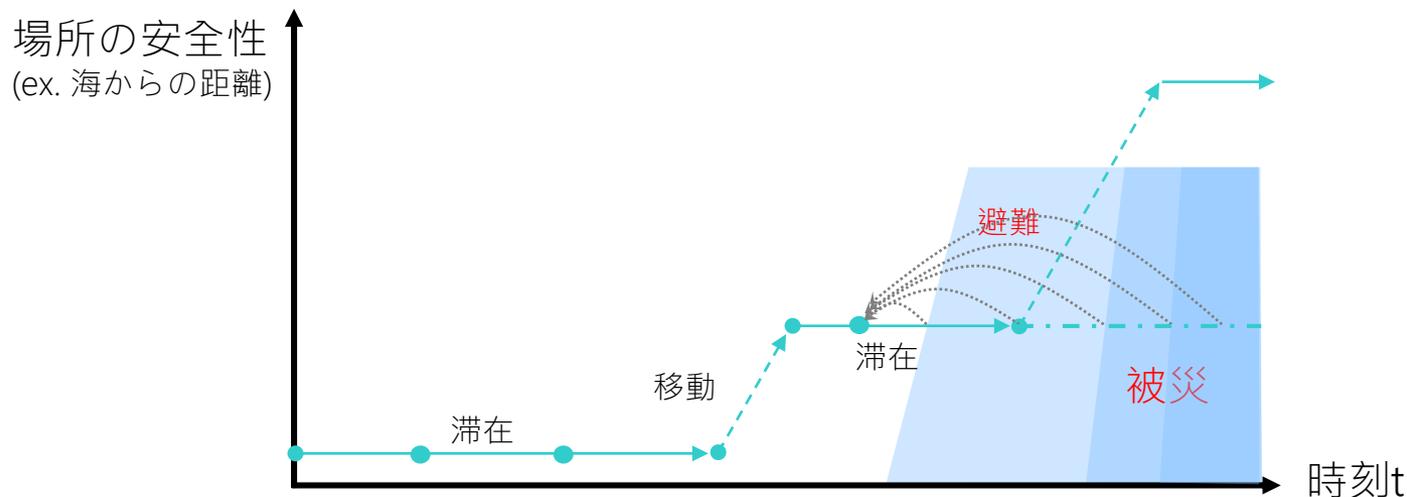
# 災害時不確実下の避難開始選択

## ■ 災害時行動の前提

↑ 経験と情報の不足のため、将来状態がわからない ↓

## ■ 避難 = 将来リスクの回避

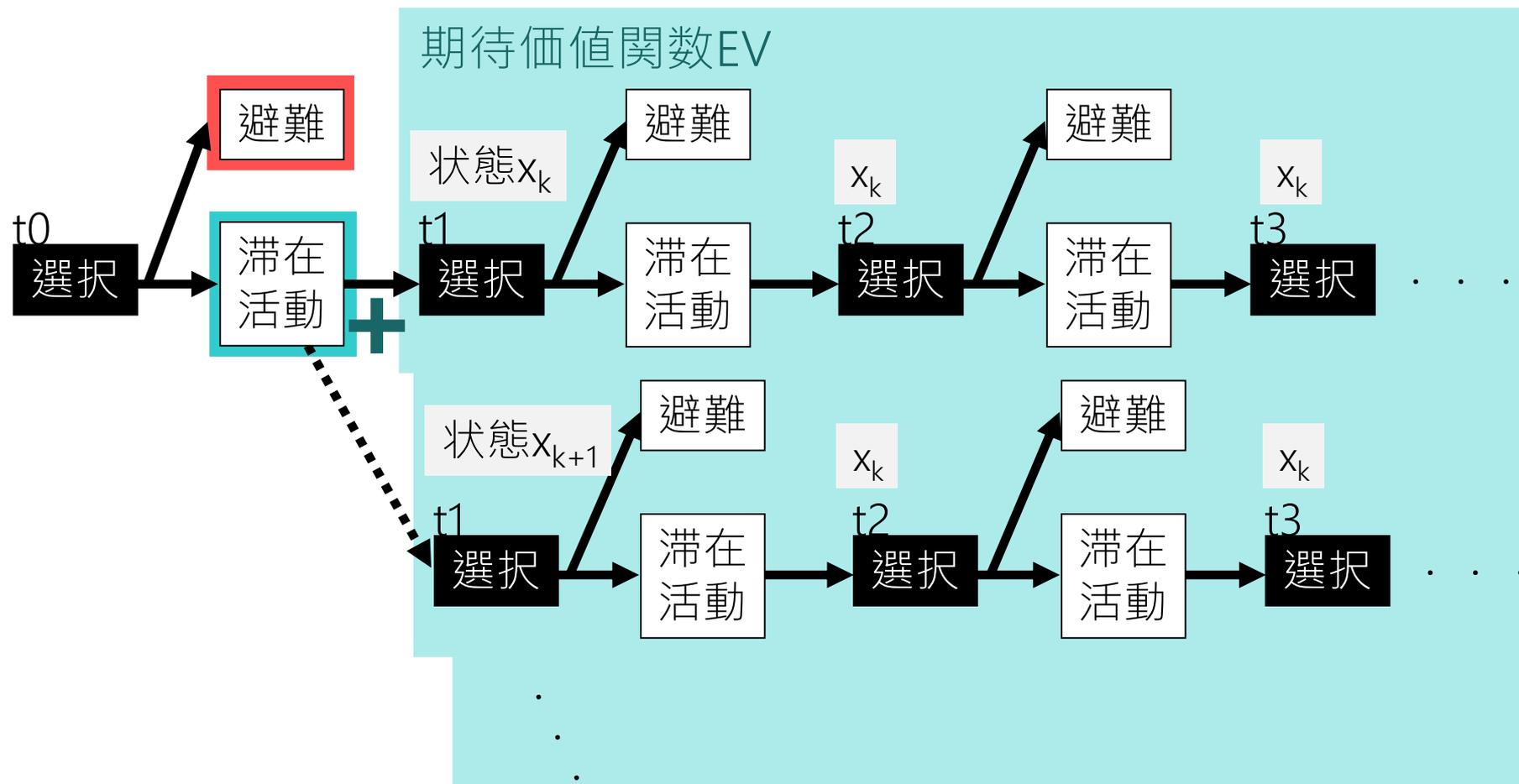
将来状況を想定した避難開始の選択



目的：不確実な将来状況を考慮した行動選択の記述

# アプローチ: 動的離散選択の導入

動的離散選択モデル: Bellman(1957)  
将来の選択から得られる効用を想定

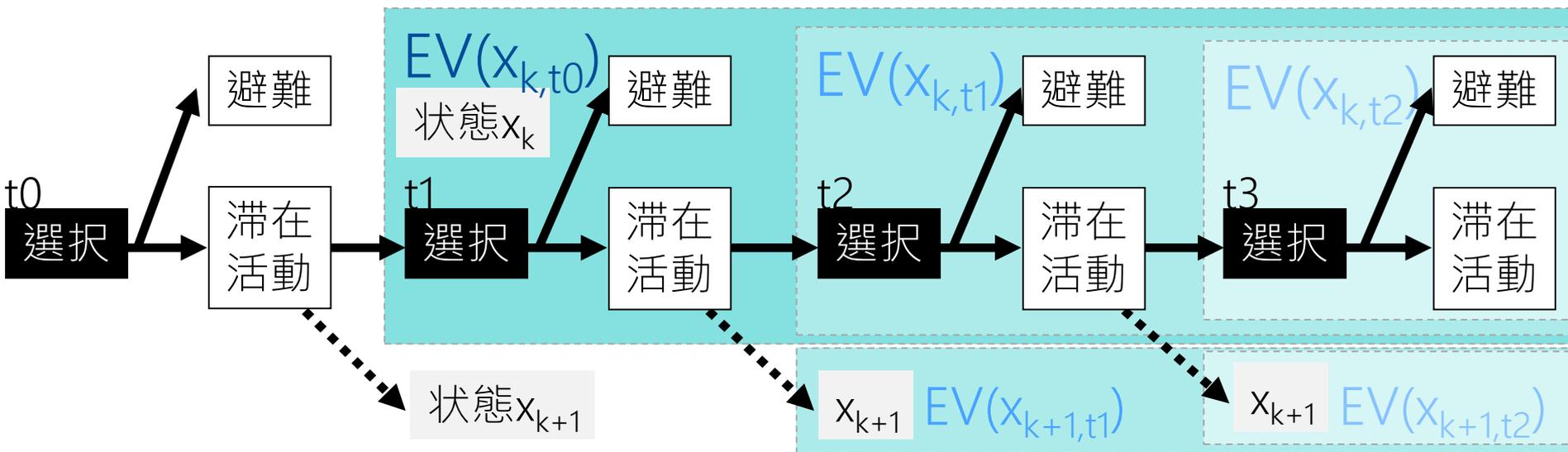


# 従来の動的離散選択モデル

期待価値関数EVの計算コストは非常に大きい

—Nested Fixed Point algorithm(Rust(1987))では、非線形のJ次の連立方程式を求解

期待価値関数EVは真値を用いる



制約条件

期待価値関数EV

$$EV(x_t, d_t) = \sum_{j=0}^J \left\{ \left( \log \left( \sum_{\forall d'} \exp(v(x'_j, d') + \beta EV(x'_j, d')) \right) \right) \times p_3(x'_j | x_t, d_t) \right\}$$

今期の期待価値関数

次期以降の推移より算出した今期の期待価値関数

# 将来効用に異質性を導入

異質性を導入した定式化

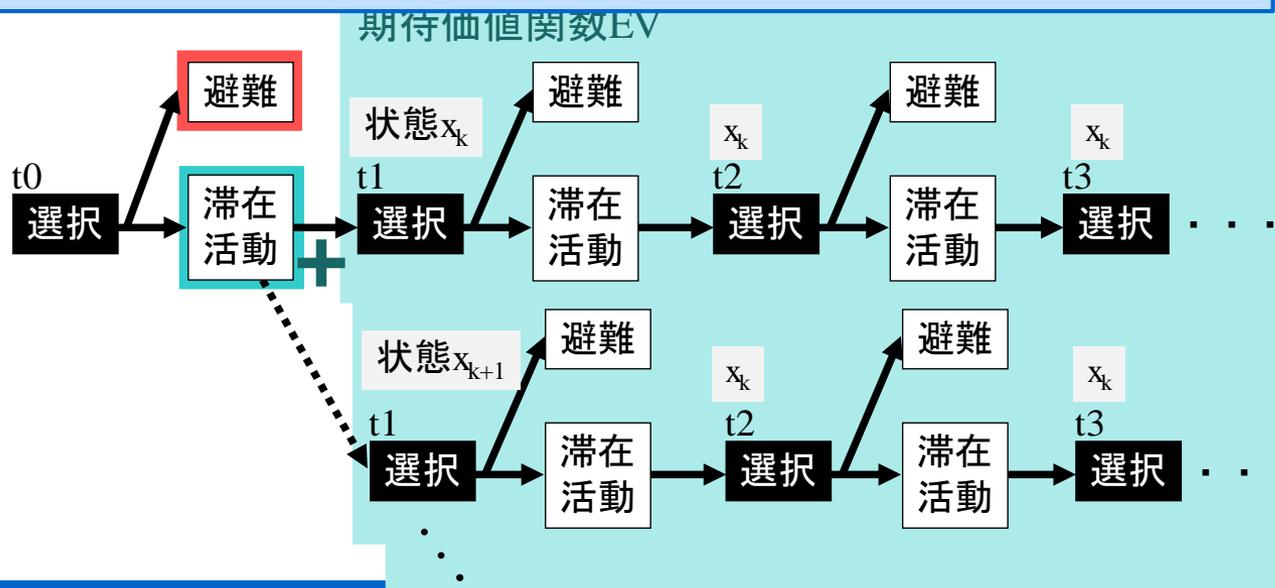
$$\min_{\theta} -L(\theta) \quad \text{subject to } c(\theta, EV') \in \Omega$$

$$c(EV, \theta, x_t, d_t) = \underbrace{EV(x_t, d_t)}_{\text{今期の期待価値関数}} - \sum_{j=0}^J \left\{ \underbrace{\left( \log \left( \sum_{v,d'} \exp(v(x'_j, d') + \beta EV(x'_j, d')) \right) \right)}_{\text{次期以降の推移より算出した今期の期待価値関数}} \times p_3(x'_j | x_t, d_t) \right\} \quad (3)'$$

今期の期待価値関数

次期以降の推移より算出した今期の期待価値関数

災害時に真の期待価値関数EVを持つことは困難



# MPEC型の解法の適用

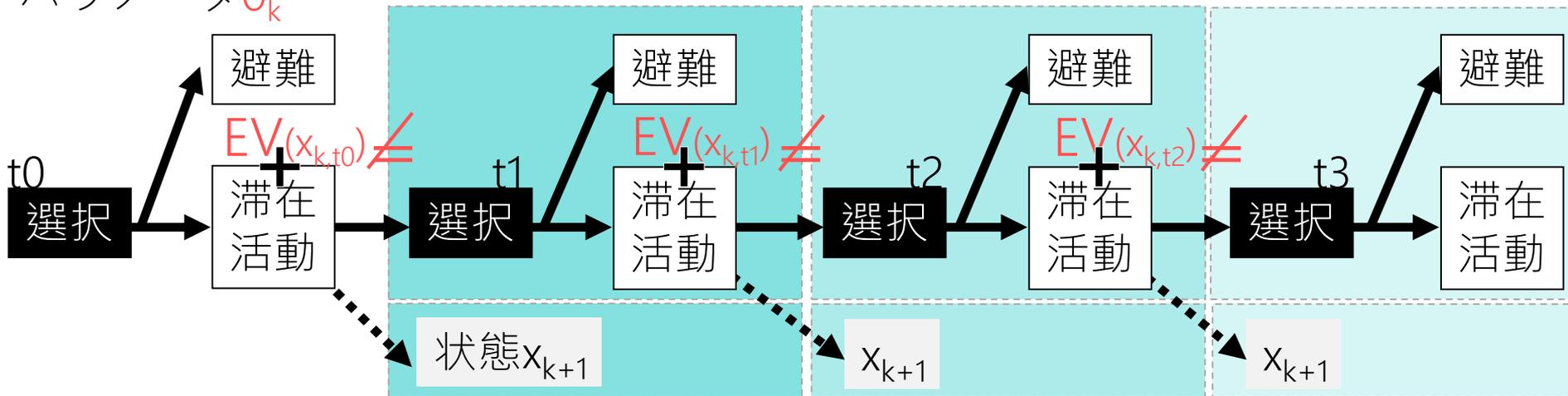
非線形等式制約付き最適化問題

$$\min_{\theta, EV} -L(\theta, EV)$$

$$\text{subject to } c(\theta, EV) = 0$$

Su & Judd(2012)

パラメータ  $\theta_k$



パラメータEVを等式制約で評価 (連立方程式を解かない)

異質性評価が可能となる解法&計算性の問題も改善

# 求解アルゴリズム

非線形制約付き最適化問題：非線形制約数が離散状態 $(x_j, d_j)$ の数  
：パラメータは離散状態数 + 効用関数 $\theta$

- $\Omega$ の設定
1.  $c(\theta, EV') \in R^M$
  2.  $|c_i(\theta, EV')| < \phi, \forall i$
  3.  $\sum_{\forall i} |c_i| < \Phi$



ペナルティ関数を用いた求解

k: 解の更新回数   r: ペナルティパラメータ  
 $\Phi$ : 完全合理性の緩和領域

$$\min_{\theta, EV} \left( -L(\theta, EV) + \sum_{\forall i} r |c_i(\theta, EV)| - r\Phi \right) \Leftrightarrow \min_{\theta, EV} -L(\theta, EV) \quad \text{subject to } c_i(\theta, EV) = 0, \forall i$$

( 緩和領域の制約を満たさない場合 )

更新:  $(\theta, EV)_{k+1} = (\theta, EV)_k + \alpha_k S_k$



逐次二次計画法(SQP)

# 求解アルゴリズム(逐次二次計画法(SQP))

$$\min_{\gamma=(EV,\theta)} A(\gamma)(= -L(EV, \theta)), \text{ subject to } c_i(\gamma) = 0, i \in \mathcal{V}(x_k, d)$$

A: 目的関数式(4) の置き換え  
 $c_i$ : 制約式(2)' の置き換え

最急降下方向  $s_k$  の決定(部分問題)

k: 繰り返し計算の回数  
 B: Aのラグランジアンへのヘッセ行列  
 $\lambda, b_i$ : ペナルティパラメータ

$$\min_{\gamma=(EV,\theta)} \nabla A(\gamma_k)^T s_k + s_k^T B^k s_k / 2, \text{ subject to } c_i(\gamma_k) + \nabla c_i(\gamma_k)^T s_k = 0 \quad (5)$$

2乗のメリット関数を設定し、最急降下法で求解

$s_0$ をどう置くか

$$Q(s_k) = \nabla A(\gamma_k)^T s_k + \frac{s_k^T B^k s_k}{2} + \frac{b_1}{2} \sum_i (c_i(\gamma_k) + \nabla c_i(\gamma_k)^T s_k)^2 \quad (6)$$

$$\text{収束判定: } \|c_i(\gamma_k) + \nabla c_i(\gamma_k)^T s_k\| < \epsilon, \forall i \quad (7)$$

$$\text{値が動かない場合に増加: } b_1 \quad (8)$$

増加のバランスとkを何回繰り返すか  
 ( $\epsilon$ が大きくても探索は可能)

最急降下方向  $s_k$

ペナルティ法による解  $\gamma_k$  の更新

$r_k$ が大きすぎるとAを無視  
 小さすぎるとcが小さくならない  
 $v$ : Aのラグランジュ乗数  
 $b_i$ : ペナルティパラメータ

$$\text{ペナルティ関数: } F_r(\gamma_k) = A(\gamma_k) + r_k \sum_i |c_i(\gamma_k)| \quad (9)$$

$$\text{黄金分割法等の線形探索により } \alpha_k \text{ 決定: } \gamma_{k+1} = \gamma_k + \alpha_k s_k$$

$$\text{収束判定: } |c_i(\gamma_k)| < \delta, \forall i \quad (11)$$

$\alpha_k$ が微小だと局所収束

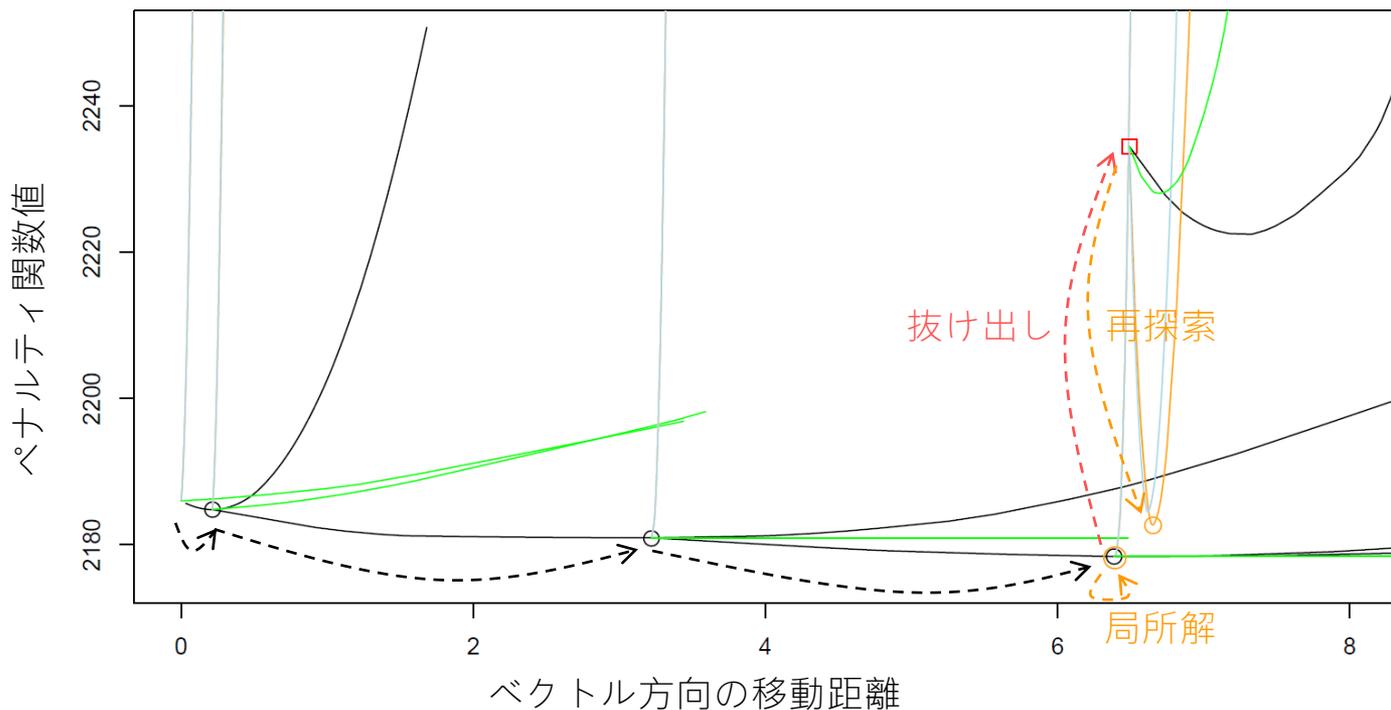
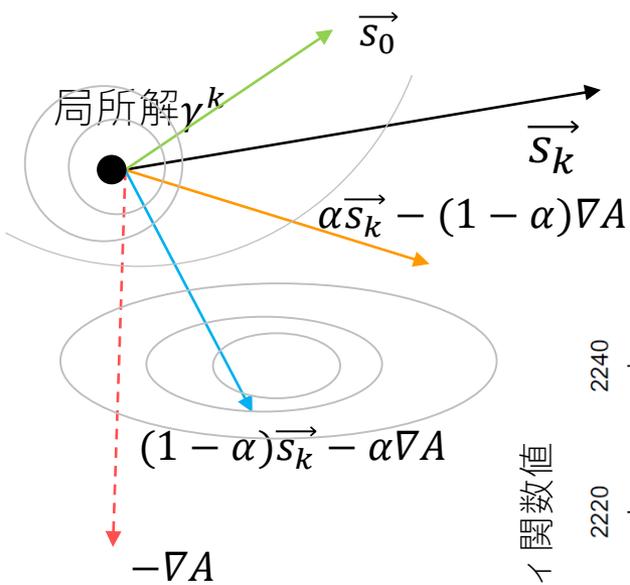
$$\text{収束しない場合の更新: } r_{k+1} = \max_i (r_k, |v_i^k|), \nabla A(\gamma_k) + \sum_i v_i^k \nabla c_i(\gamma_k) = 0 \quad (12)$$

再計算  
 k+1回目

# 局所収束回避の工夫

パラメータ数が多く、非線形制約を持つため、局所解が多い

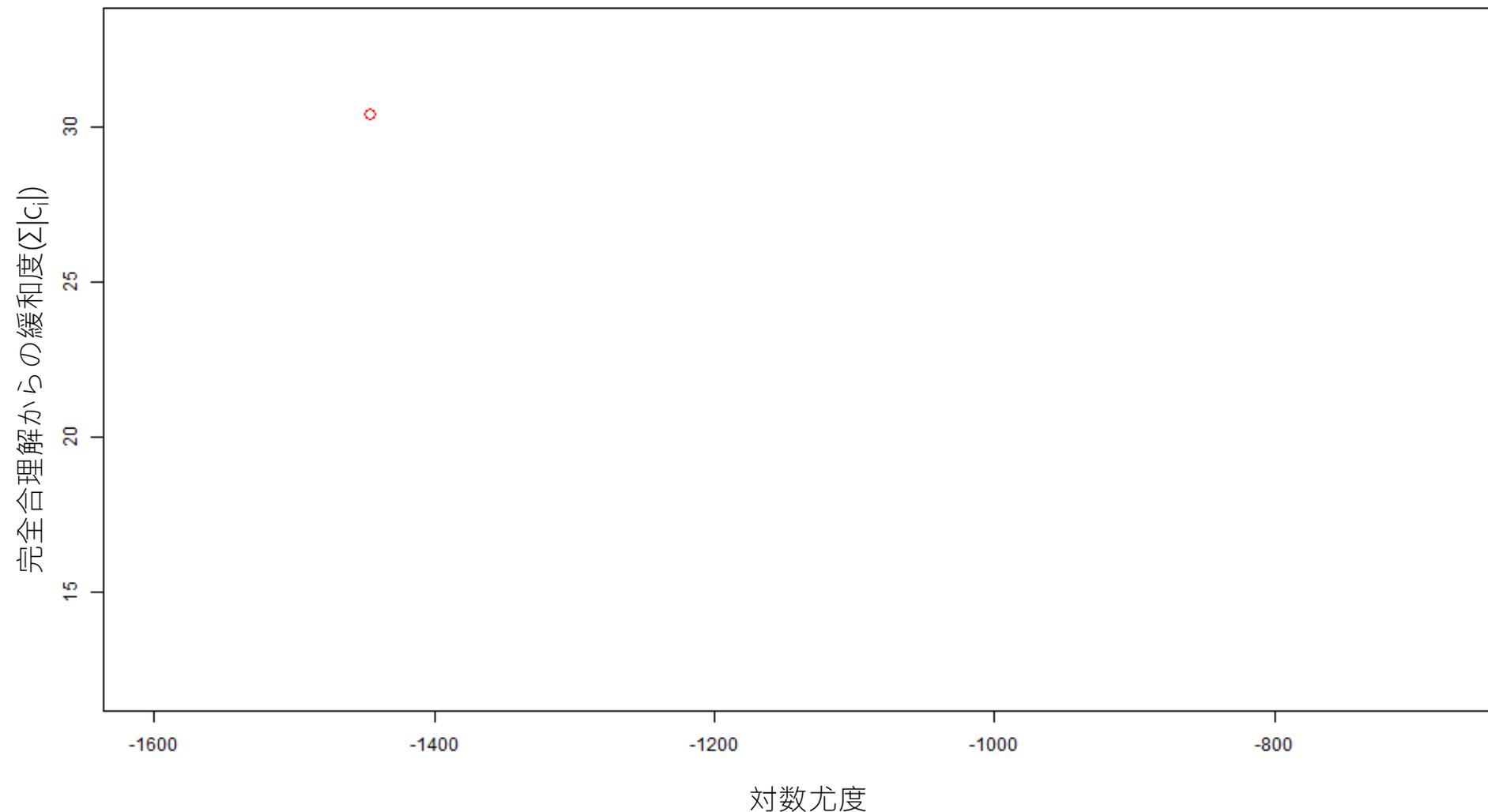
降下方向ベクトル  $\vec{s}_k$  を工夫  $\Rightarrow$  複数のベクトルによる局所解からの離脱



# 局所収束回避の工夫(計算過程)

Computer : Intel Core(TM) CPU i5-4200M @ 2.50GHz & RAM 8.00GB

Code: C言語 1回の計算時間: 5~60秒



# おまけ1： 二段階計画問題

というか、ここからが長い？

# 二段階計画問題とは

ある問題の最適化が別の問題の最適化の制約条件になっている問題

## 上位問題

$$\begin{aligned} \max_{\mu} V(x, \mu) \\ \text{subject to } G(x, \mu) \leq 0, H(x, \mu) = 0 \end{aligned}$$

$\mu$ : 上位側の制御変数  
(計画者の政策変数)

## 下位問題

$$\begin{aligned} \max_x f(x, \mu) \\ \text{subject to } g(x, \mu) \leq 0, h(x, \mu) = 0 \end{aligned}$$

$x$ : 下位側の制御変数  
(利用者の行動結果(効用最大化))

### 本研究

#### 上位問題

道路管理事業者の  
道路維持管理費用最小化

#### 下位問題

物流事業者の  
配送費用最小化

制約条件

物流事業者の最適配送行動がなされる前提で  
道路管理事業者が道路維持管理における最適施策を導入する

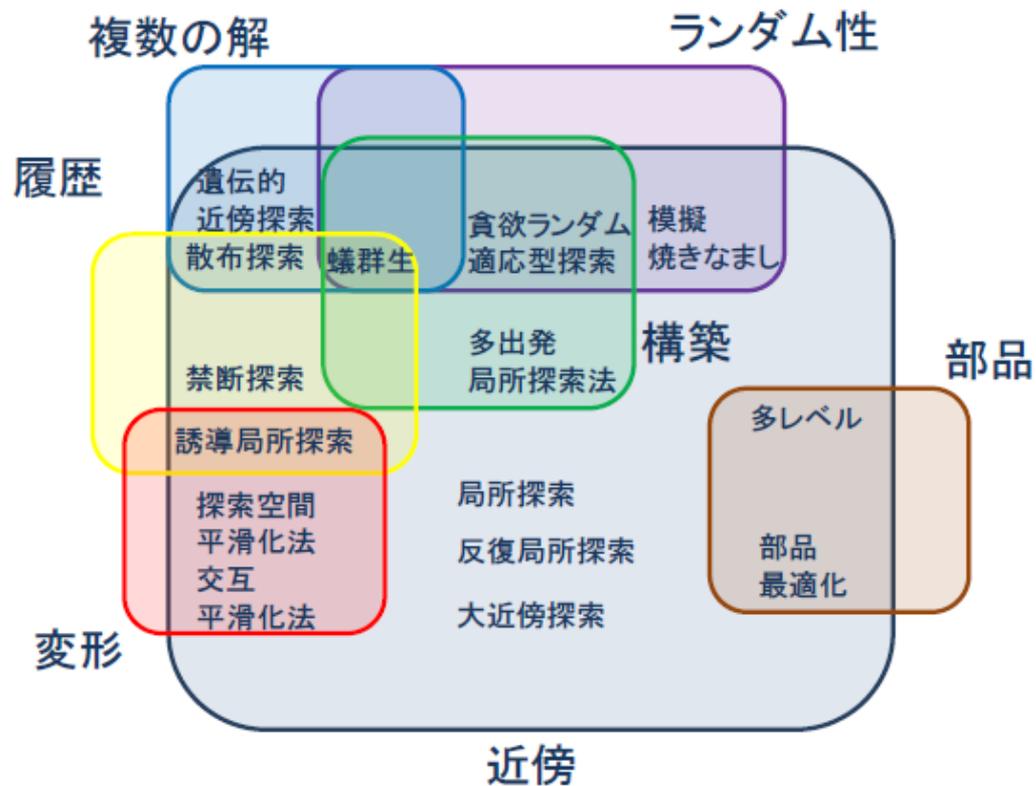
今泉, 羽藤(2013)

# メタヒューリスティクス(制御変数が離散量)

: 求められた解に精度の保証はないが、  
経験的に近似解が求められるとわかっている手法

## メタヒューリスティクスの基本戦略

- 近傍の利用
- 構築法の利用
- 部品の利用
- 複数解の保持
- ランダム性の利用
- 問題の変形
- 探索履歴の利用



### メタヒューリスティクスの数理

～第2章 代表的なメタヒューリスティクス～

- 局所探索法
- 多出発局所探索法
- 反復局所探索法
- 模擬焼きなまし法
- 禁断探索法
- 誘導局所探索法



メタヒューリスティクス ～夏の祭典～  
2013/07/10(水) M1 今泉孝章

夏合宿ゼミ # keynote 1

# 動学問題と最適化

*pedestrian* →

*cycle* →

*bus & tram*

*cycle parking*

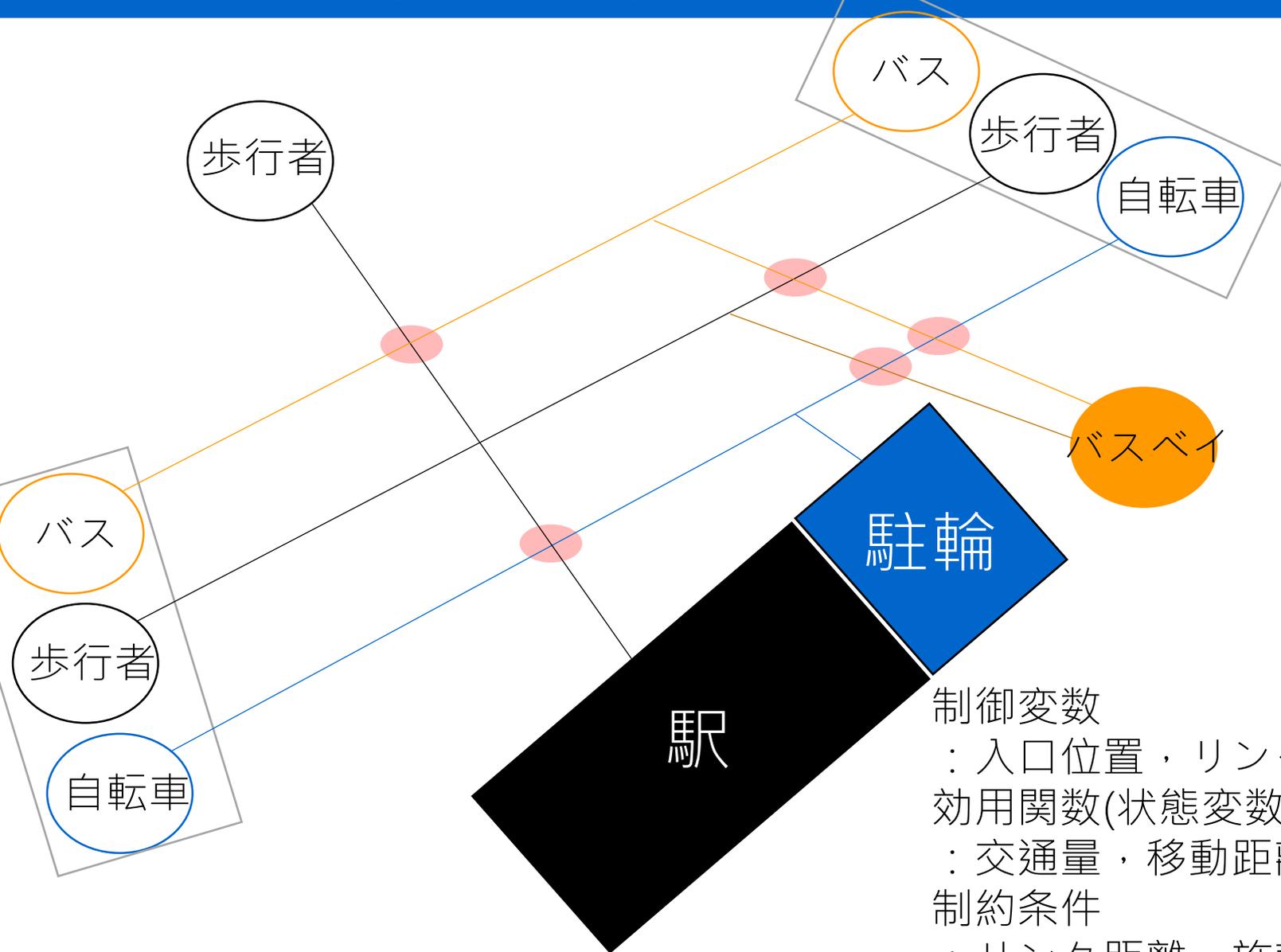
*cyclist* →

*pedestrian* ←

2015/9/19(土)

浦田 淳司

# イメージ:最適施設計画



制御変数

: 入口位置, リンク, 施設位置  
効用関数(状態変数)

: 交通量, 移動距離, 交差点

制約条件

: リンク距離, 施設数

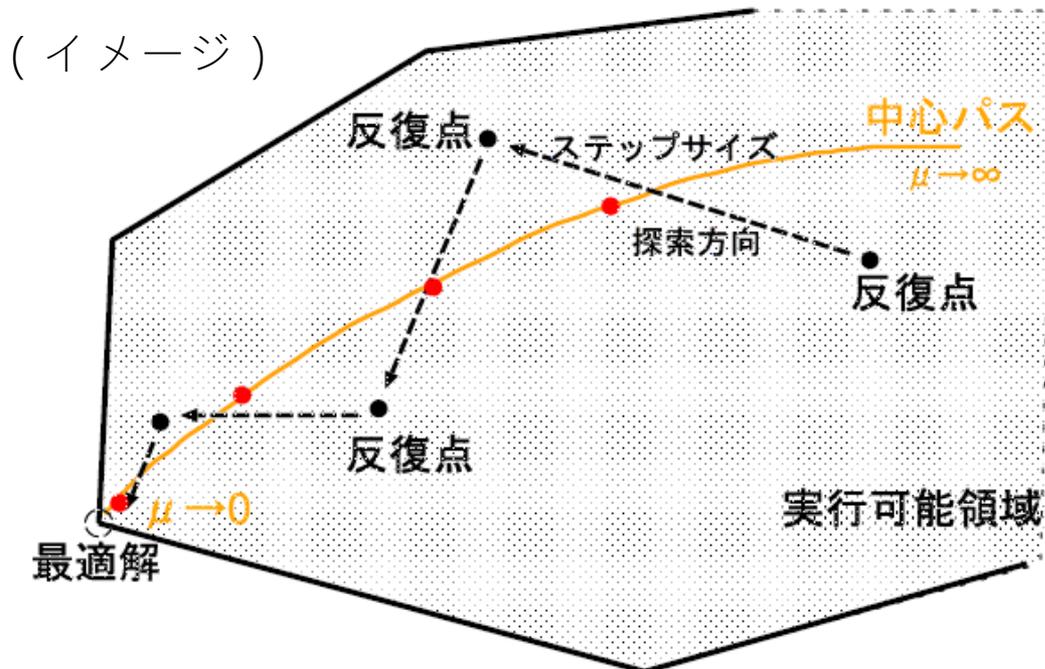
# おまけ2: 内点法関連

参考: 東工大水野研究室テキスト

# 内点法

- ・最適化問題において、制約条件に含まれる全ての不等式を等号ではなく厳密に不等号で満たす点を(解析的な)内点と呼ぶ
- ・一つの内点の近傍で、不等式制約を考慮せず、等式制約条件のみの中で目的関数を減少させる方向を求め、あるステップサイズだけ進み、内点を更新する
- ・内点法は、初期内点からこれを繰り返すことにより、内点列を生成

※線形計画問題では、最適解は、内点ではなく、いくつかの不等式を等号でみたす境界点である。そこで、内点法は最適解以外の境界に近づかないようにしながら、最適解に近づく点列を生成する



- ・様々な最適化問題に適用可能
- ・理論的に収束が保証・計算量が前もって評価可能
- ・大規模な問題を高速に求解できる

# 主双対内点法

線形計画問題 ( 主問題 )

$$\begin{array}{ll} \text{最小化} & c^T x \\ \text{制約条件} & Ax = b \\ & x \geq 0 \end{array}$$

( 双対問題 )

$$\begin{array}{ll} \text{最大化} & b^T y \\ \text{制約条件} & A^T y + z = c \\ & z \geq 0 \end{array}$$

( 主双対問題 )

$$\begin{array}{l} Ax = b \\ A^T y + z = c \\ Xz = 0 \\ x \geq 0, z \geq 0 \end{array} \quad (1)$$

$X$  :  $x$  の対角成分

主双対問題の条件を全て満たす集合が実行可能解(実行可能内点)

$$F_{PD} = \{(x, y, z) | Ax = b, A^T y + z = c, x \geq 0, z \geq 0\}$$

次の二つの仮定が成り立つとき、主双対問題は最適解を持つ

仮定1: 行列 $A(m \times n)$ のランクが $m$ である

仮定2: 主双対問題に実行可能内点が存在し、そのうち一つが既知である

# 主双対内点法

k番目の実行可能内点 $(x_k, y_k, z_k)$ が求められているとき，  
探索方向 $(\Delta x, \Delta y, \Delta z)$ とステップサイズ $\alpha$ を用いて，次の点を次で定める

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \\ z^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \\ z^k \end{pmatrix} + \alpha \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}$$

$(x_{k+1}, y_{k+1}, z_{k+1})$ が実行可能内点となる必要十分条件は，

$A\Delta x = 0, A^T \Delta y + \Delta z = 0, x_{k+1} > 0, z_{k+1} > 0$ を満たすことである

# 主双対内点法(アフィンスケーリング法)

$$h(x, y, z) = \begin{pmatrix} Ax - b \\ A^T y + z - c \\ Xz \end{pmatrix}$$

としたとき、(1)の最適解は  $h(x, y, z) = 0$  の解。

この近似解を求めるためにニュートン法を用い、計算方向を探索方向とする

点  $u'$  におけるニュートン方向  $\Delta u$  は、線型方程式系  $\nabla f(u') \Delta u = -f(u')$  の解。

これを用いて、 $(x^k, y^k, z^k)$  のニュートン方向  $(\Delta x, \Delta y, \Delta z)$  は、次の解となる。

$$\nabla h(x^k, y^k, z^k) \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = -h(x^k, y^k, z^k)$$

$$\Leftrightarrow \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & E \\ Z_k & 0 & X_k \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X_k z^k \end{pmatrix}$$

$$\Delta y = (AZ_k^{-1}X_k A^T)^{-1} b$$

$$\Leftrightarrow \Delta z = -A^T \Delta y$$

$$\Delta x = -Z_k^{-1} X_k \Delta z - x^k$$

# 二次錐計画問題

## 特徴 (対称錐計画問題)

- ▷ 目的関数は線形
- ▷ 条件は線形制約と対称錐制約  
(非負制約, 2次錐制約, 半正定値制約など)
- ▷ 線形計画・2次錐計画・半正定値計画を含む
- ▷ 主双対内点法により, 多項式時間で最適解が求まる

中田和秀: 対称錐計画問題に対する主双対内点法

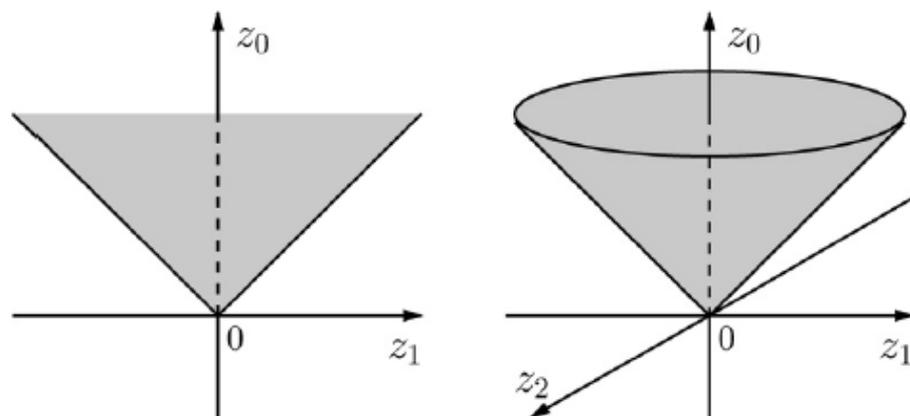


図1 2次元(左)と3次元(右)の2次錐

福田・福島(2014)

# 二次錐計画問題

## 特徴 (対称錐計画問題)

- ▷ 目的関数は線形
- ▷ 条件は線形制約と対称錐制約  
(非負制約, 2次錐制約, 半正定値制約など)
- ▷ 線形計画・2次錐計画・半正定値計画を含む
- ▷ 主双対内点法により, 多項式時間で最適解が求まる

$$\min c^T x$$

$$\text{subject to } Ax = b \\ x \in K^n$$

$$\text{二次錐 } K^n = \left\{ (x_0; z) \in R \times R^{n-1} : \sqrt{\sum_{i=1}^{n-1} z_i^2} \leq x_0 \right\}$$

# 応用例: ロバスト最適化

楕円形の不確実性をもつロバスト最適化問題を二次錐計画問題として定式化可能

$\min t$

subject to  $c^T x \leq t$

$Ax - b \leq 0$

$(c, A, b) \in U \leftarrow c, A, b$ に楕円形の不確実性がある

$$c \in \{c^* + Cu : C \succcurlyeq 0, \|u\| \leq 1\}$$

$$\begin{pmatrix} a_j \\ b_j \end{pmatrix} \in \left\{ \begin{pmatrix} a_j^* \\ b_j^* \end{pmatrix} + A_j \begin{pmatrix} u \\ v \end{pmatrix} : A_j \succcurlyeq 0, \left\| \begin{pmatrix} u \\ v \end{pmatrix} \right\| \leq 1 \right\}$$

楕円形の不確実性を展開

$$c^T x - t \leq 0, c \in \{c^* + Cu : C \succcurlyeq 0, \|u\| \leq 1\}$$

$$\Rightarrow (c^*)^T x + \max_{\|u\| \leq 1} u^T Cx - t \leq 0$$

$$\Rightarrow (c^*)^T x + \|Cx\| - t \leq 0$$

$$\|Cx\| \leq t - (c^*)^T x \leftarrow \text{二次錐制約}$$

A, bも次となる

$$\left\| A_j^T \begin{pmatrix} x \\ -1 \end{pmatrix} \right\| \leq b_j^* - (a_j^*)^T x$$

# まとめ

2.1 変分法・2.2 最大値原理・4.1 二段階計画問題

⇒ ( 解法を理解した上で ) いかに解ける問題を設定するかが重要

2.3 動的計画法・3 動的離散選択のパラメータ推定

⇒ ( 定式化後も ) 制約のある最適化の求解アルゴリズムの問題

⇒ ( 式は理解しても ) 計算プログラムを行列で書くと問題発生

1. はじめに
2. 動学的最適化 ( 動学問題の最適化の定式化と解法 )
  - 2.1 変分法
  - 2.2 最大値原理
  - 2.3 動的計画法 (動的離散選択モデル)
3. 動的離散選択モデルにおけるパラメータ推定
  - 3.1 NFXP
  - 3.2 NPL (擬似最尤推定)
  - 3.3 MPEC型
4. おまけ
  - 4.1 二段階計画問題
  - 4.2 内点法と二次錐計画問題

# 参考図書・推薦図書

## 2. 動学的最適化 ( 動学問題の最適化の定式化と解法 )

### 2.1 変分法, 2.2 最大値原理

A. C. チャン(訳小田正雄, 仙波憲一, 高森寛, 平澤典男): 動学的最適化の基礎, シーエーピー出版, 2006.

### 2.3 動的計画法 (動的離散選択モデル)

## 3. 動的離散選択モデルにおけるパラメータ推定

### 3.1 NFXP

Rust, J., Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher, *Econometrica*, Vol.55, pp.999-1033, 1987.

### 3.2 NPL (擬似最尤推定)

Aguirregabiria, V., Mira, P.: Swapping the nested xed point algorithm: A class of estimators for discrete Markov decision models, *Econometrica*, Vol. 70(4), pp. 1519-1543, 2002.

### 3.3 MPEC型

Su, C.J., Judd, K.L.: Constrained optimization approaches to estimation of structural models *Econometrica* Vol 80 pp 2213-2230 2012

寒野善博, 土谷隆: 基礎系数学最適化と変分法(東京大学工学教程), 丸善出版, 2014.

久保幹雄・J.P. ペドロソ: メタヒューリスティクスの数理, 共立出版, 2009.

Nocedal, J., Wright, S.J.: Numerical Optimization, Springer, 2006.

杉原正顕,, 室田一雄: 数値計算法の数理, 岩波書店, 2003.

土谷政和, 土谷隆, 寒野善博, 仙波憲一, 高森寛, 平澤典男: 動的計画法, 丸善出版, 2004.

ご清聴ありがとうございました・