BinN Studies シリーズ

マルコフ連鎖から大規模言語モデルへ

現代行動理論の基礎と応用

羽藤英二　編著

小川大智　古橋郁一　松永隆宏　今村啓太　須賀理帆　著

饗場清日　石河万衣　長田晴人　桑本球太郎　小西将真　冨矢航大

名嘉山結月　永井鴻平　三木隆斗　李康民　鷲野萌花　コラム執筆


```
## パラメータ値の最適化
res <- optim(b0, fn, method="BFGS", hessian=TRUE,control=list(fnscale=-1))

## パラメータ推定値，ヘッセ行列
b <- res$par
hhh <- res$hessian

## t 値の計算
tval <- b/sqrt(-diag(solve(hhh)))

## 初期尤度
L0 <- fn(b0)
## 最終尤度
LL <- res$value

##### 結果の出力 #####
## 初期尤度
print(L0)
## 最終尤度
print(LL)
## ρ^2 値
print(1-LL/L0)
## 修正済み ρ^2 値
print(1-(LL-length(b))/L0)
## パラメータ推定値
print(b)
## t 値
print(tval)
```

目 次

第 1 章　はじめに *11*
参考文献 *13*

第 2 章　交通量配分 *17*
序章　交通ネットワーク配分とは *17*
2.1　交通ネットワークの記述 *21*
2.1.1　ネットワークの構成要素 *21*
2.1.2　基本変数と関係式 *22*
2.1.3　リンクパフォーマンス関数（BPR 関数） *23*
2.1.4　配分モデルの全体像 *24*
2.2　非混雑型配分（フロー非依存配分） *25*
2.2.1　All-or-Nothing 配分 *26*
2.2.2　吸収マルコフ過程配分 *29*
2.2.3　確率的配分（ロジット配分） *31*
2.2.4　Dial のアルゴリズム *32*
2.3　利用者均衡配分（UE） *35*
2.3.1　Wardrop の第 1 原則 *36*
2.3.2　等価最適化問題（Beckmann 変換） *37*
2.3.3　Frank-Wolfe 法 *38*
2.3.4　Braess のパラドックス *42*
2.4　確率的利用者均衡配分（SUE） *45*
2.4.1　完全情報仮定の緩和 *45*
2.4.2　ロジット型 SUE の等価最適化問題 *47*
2.4.3　SUE の解法 *48*
2.5　システム最適配分（SO）と混雑課金 *51*
2.5.1　Wardrop の第 2 原則と定式化 *51*
2.5.2　UE と SO の比較 *52*
2.5.3　混雑課金（限界費用課金） *54*
2.5.4　余剰分析 *55*
2.6　動的交通配分（DTA） *61*

2.6.1　静的 UE と DTA の対比 …… 61
2.6.2　動的利用者均衡（DUE）の定式化 …… 62
2.6.3　時間分解と効率的アルゴリズム …… 63
2.6.4　解法と静的 TA との比較 …… 66
2.7　進化ゲーム理論と交通配分 …… 67
2.7.1　ポテンシャルゲームとしての UE …… 68
2.7.2　進化ゲームと Day-to-day ダイナミクス …… 69
2.7.3　集団ゲームの枠組み …… 70
2.7.4　改訂プロトコルと進化動学 …… 70
2.7.5　代表的な進化動学 …… 71
2.7.6　安定性と収束定理 …… 73
2.7.7　Day-to-day ダイナミクスへの応用 …… 75
2.8　ピグー課金と混雑課金の一般理論 …… 76
2.8.1　外部性とピグー税の考え方 …… 77
2.8.2　最適課金の定式化（第一最善） …… 78
2.8.3　第二最善課金とネットワーク制約 …… 79
2.8.4　Vickrey ボトルネックモデルと動的課金 …… 80
2.8.5　収入の再分配と公平性 …… 82
2.9　シミュレーション …… 84
2.9.1　マクロシミュレーション（MFD） …… 84
2.9.2　ミクロシミュレーション …… 88
2.10　補足 A　黄金分割法（ステップサイズ探索） …… 96
2.10.1　黄金分割法のアルゴリズム …… 96
2.11　補足 B　All-or-Nothing 配分の効率的アルゴリズム …… 97
2.11.1　直感的アイデア …… 97
2.11.2　アルゴリズムの詳細 …… 97
2.12　補足 C　利用者均衡と Nash 均衡の等価性 …… 98
2.12.1　定義：Wardrop 均衡と Nash 均衡 …… 98
2.12.2　設定 …… 99
2.12.3　Nash 均衡との等価性（命題 C.1） …… 99
2.13　補足 D　相補性条件と KKT 条件 …… 100
2.13.1　KKT 条件の一般形 …… 100
2.13.2　UE/FD-Primal への適用（Wardrop 条件との等価性） …… 101
2.13.3　SUE/FD-path への適用（ロジット確率との等価性） …… 102
参考文献 …… 103

第 3 章　行動モデル *107*
序章 行動モデルとは *107*
3.1　ランダム効用最大化理論 *108*
3.1.1　効用の記述とランダム性の導入 *109*
3.1.2　誤差項の独立同一性の仮定 *109*
3.1.3　誤差項の分布形状 *110*
3.1.4　選択肢・個人間の相関・異質性の扱いと IIA 性の問題 *112*
3.2　MNL からの拡張: 相関・異質性 *112*
3.2.1　IIA の限界 *112*
3.2.2　Nested Logit モデル *113*
3.2.3　個人間の異質性と Mixed Logit モデルの導入 *115*
3.3　選択肢非列挙型モデル *118*
3.3.1　選択肢集合の爆発 *118*
3.3.2　動的離散選択モデルと Recursive Logit モデル *118*
3.3.3　時間構造化ネットワークと Recursive Logit *121*
3.3.4　摂動効用型経路選択モデル *124*
3.4　アクティビティモデル *129*
3.4.1　Activity-Based Models (ABMs) の基本的な考え方 *129*
3.4.2　経路選択モデルの拡張 *129*
3.4.3　日間変動を捉えるモデル：活動ニーズモデル *130*
3.4.4　離散連続モデル *131*
3.5　モデルの推定と検定 *145*
3.5.1　尤度と最尤推定 *145*
3.5.2　識別可能性 *146*
3.5.3　検定 *146*
3.5.4　観測の不確実性とモデル推定 *149*
参考文献 *154*

第 4 章　最適化 *157*
4.1　最適化とは *157*
4.1.1　最適化の基本的な考え方 *157*
4.1.2　最適化問題の数学的定式化 *157*
4.1.3　最大化問題と最小化問題 *158*
4.1.4　連続最適化と離散最適化 *159*
4.1.5　実行可能領域と最適解 *160*
4.1.6　線形最適化と非線形最適化 *163*
4.1.7　都市交通における最適化の例 *164*
4.2　線形計画法 *165*

4.2.1 線形計画法とは 165
4.2.2 線形計画問題の幾何学的理解 166
4.2.3 標準形と実行可能基底解 167
4.2.4 双対問題 169
4.2.5 KKT 条件と相補性条件 170
4.2.6 双対性と KKT 条件の関係 171
4.2.7 代表的な解法 173
4.2.8 線形計画法の例 175
4.3 組合せ最適化の基礎 189
4.3.1 組合せ最適化とは 189
4.3.2 整数計画と 0–1 計画 189
4.3.3 NP 困難性と計算の難しさ 191
4.3.4 最短経路問題 191
4.3.5 巡回セールスマン問題 192
4.3.6 配送計画問題 194
4.3.7 厳密解法と近似解法 198
4.4 厳密解法 200
4.4.1 厳密解法の考え方 200
4.4.2 分枝限定法 200
4.4.3 切除平面法 202
4.4.4 列生成法 203
4.4.5 ベンダーズ分解 205
4.4.6 分枝カット法 206
4.4.7 Branch-and-Price 207
4.4.8 厳密解法の意義と限界 209
4.5 ヒューリスティクスと近傍探索 211
4.5.1 ヒューリスティクス 211
4.5.2 初期解の構成 212
4.5.3 近傍探索の基本概念 212
4.5.4 局所探索と局所最適解 213
4.5.5 代表的な近傍構造 213
4.5.6 貪欲法と局所改善の違い 214
4.5.7 メタヒューリスティクスの考え方 215
4.5.8 ヒューリスティクスの評価 218
参考文献 233

第 5 章　機械学習の基礎 *235*
5.1　機械学習の基礎 *235*
5.1.1　機械学習の歴史 *235*
5.1.2　機械学習の基本設定 *236*
5.1.3　損失関数と最適化 *238*
5.1.4　確率的勾配降下法 (SGD) *239*
5.1.5　誤差逆伝播法 *241*
5.1.6　ニューラルネットワークの基本構造 *242*
5.1.7　正則化と汎化 *244*
5.1.8　CNN *245*
5.1.9　RNN *246*
5.1.10　Transformer *247*
5.2　強化学習 *255*
5.2.1　強化学習の歴史 *255*
5.2.2　マルコフ決定過程 *256*
5.2.3　多腕バンディット問題 *258*
5.2.4　動的計画法 *259*
5.2.5　価値ベース手法：Q 学習, Deep-Q 学習 *260*
5.2.6　方策勾配法：Actor-Critic, PPO *262*
5.2.7　発展的話題 *263*
5.2.8　まとめ *265*
5.3　拡散モデルと生成モデル *284*
5.3.1　生成モデルの考え方 *284*
5.3.2　EM アルゴリズム *285*
5.3.3　VAE *288*
5.3.4　拡散モデル *289*
5.3.5　連続時間拡散モデル *291*
5.3.6　自己回帰モデルと LLM *292*
5.3.7　世界モデル *293*
参考文献 *302*

コラム一覧

コラム：Braess のパラドックスの検討 〜松山市の PT 調査をもとに〜 – 冨矢航大 58
コラム：東京都心部の道路網における second-best pricing – 鷲野萌花 92
コラム：交通手段選択モデルの推定 – 石河万衣 117
コラム：ログサム変数の導出 – 小西将真 127
コラム：コロナ前後における活動の変化 – 饗場清日 136
コラム：バスのルート最適化の手法提案と比較 – 桑本球太郎 178
コラム：神奈川県央地域でのデマンド型交通の対象地区の選定 – 永井鴻平 220
コラム：高尾駅周辺における自動運転オンデマンド交通の拠点配置分析 – 名嘉山結月 226
コラム：深層学習による交通流予測モデルの検討 – 李康民 250
コラム：探索と活用のジレンマ（多腕バンディット問題の実験） – 長田晴人 266
コラム：Persona 駆動型 LLM による災害後居住地選択モデル – 三木隆斗 296

第 1 章
はじめに

羽藤英二

人口減少が進んでいた地方都市を襲った巨大災害直後の破壊の風景の中に立ったとき，何かできたことはなかったか，どうすればよかったのかと思わずにはいられなかった．今，都市と交通を学ぶことにどのような意味があるのだろうか．あるいは激変する社会の中で今私たちに何ができるだろうか．おそらくそれは，道路や鉄道，駅前広場や沿道空間の形を単に学ぶことではない．現実の空間とネットワークの上で，人びとが LLM が得意とする言語のやりとりを超えて「空間」をどのように認知し，選択し，互いに影響し合い，その集積として流れ = 交通が生まれ，外部不経済としての混雑や滞留，さらには回遊，分断，格差がどのように発生しているのかを捉え直すことである．そして，よりよい理解に基づく行為の反復の果てに，都市空間そのものの価値と構造が歴史的な過程を通じていかにその地理的関係が変容していくのか，徹底的に理解することなくして，いかなる政策も計画も設計もサービスも，威勢のいい掛け声をかけたところで何も変わらないのではないだろうか．

行動モデル，ネットワーク配分，最適化，機械学習という本書の四つの柱は，この過程を，個人の意思決定から社会的な空間変容まで貫いて理解するための基礎理論である．本書の目的は，それらを別々の計算技術として紹介することではない．研究の作業と思考の往復のなかで私たちが考えてきたように，人びとの選択の連鎖がネットワーク現象を生成し，その現象を変えようとしてなされる現実の都市設計や交通計画の行為が，再び人びとの行動そのものを変え，その反復が都市自体を更新していく．この連関を，私たち自身が経験してきた都市設計，交通計画，そしてそれらを下支えする基礎理論の構築と実証の蓄積を踏まえながら，ひとつの連続した知の体系として描き直すことに本書の狙いがある．

その出発点のひとつが，McFadden [4] によって基礎づけられた離散選択理論である．2000 年のノーベル経済学賞へと結実したこの理論は，人はなぜこの経路を選び，この交通手段を選び，この場所へ向かうのかという問いに対し，効用と確率的選択という考え方によって，個人の意思決定を政策や空間設計に応答する構造として厳密に記述する道を開いた．こうして離散選択モデルは世界中の交通計画に利用され，インフラ整備の基礎理論として定着するに至った．ここで重要なのは，ネットワーク上の行動が単なる心理学的記述にとどまらず，制度変更や空間条件の変化に対して地理的分布として応答する対象となったことである．この視点があったからこそ，交通行動は社会基盤計画の中心に置かれ，公共交通計画，都市計画，駅まち設計，自動車交通制御に共通する理論的基礎が築かれたといえよう．

しかし，個人の選択を理解するだけでは都市は見えない．選択はネットワーク上で重なり合い，混雑，所要時間，遅延，容量制約，乗換え負荷，滞在時間といった現象へと変換されるからである．マルコフ過程はこうした行動の連鎖を記述する上で最も基礎的な理論であり，最短経路探索，利用者均衡，確率的配分，動的配分と結びつくことでネットワーク上の行動記述という点で大きな力を発揮することとなった．Wardrop [9] と Beckmann [1] は，この問題をネットワーク上の均衡として定式化する道を拓いた．さらに，物流や配車，公共交通運用に目を向ければ，単なる経路選択にとどまらず，誰がどの需要をいつどの順序で担うかという組合せ設計が前景化し，Dantzig [3] 以来の VRP，さらには列生成，双対問題，Benders [2] に代表される分解法が，現実の大規模問題に対抗するための基礎となってきた．こうしたアルゴリズムの実装は，線形代数に基礎づけられた行列計算によって支えられている．そして近年では，ChatGPT の登場によって，その実装や試行錯誤が以前よりはるかに身近なものとなったといってよいだろう．

さらに，現実の交通と都市は，唯一の均衡点に収束する静的な系としてではなく，学習し適応する動的な系として再定義されるべきであろう．制度が変われば人は行動を変え，情報が変われば期待も変わる．Sandholm [6, 7] による人口ゲームと進化動学は，この適応の反復そのものを理論化した点で，従来の均衡配分を決定的に拡張する契機となった．さらにいえば，交通政策とは，ある時点の均衡をずらす一度きりの操作ではない．人びとの模倣，学習，反応，慣性の上に新たな安定状態を形成していく過程への介入なのである．近年の Matsunaga and Hato（2026）は，この系譜を受け継ぎつつ，部分可制御性の下で平均場的インセンティブと逆オークションを結びつけ，都市鉄道混雑制御を設計問題として捉え直している．そこでは，限られた予算，私的情報，日々の適応，平均場的結合が，ひとつの閉ループ系として統合されつつある．

同時に，日々の経路変更や需要変動を実証的に捉えるには，個票レベルの長期観測と，それを扱いうる学習理論が必要になる．Ogawa and Hato（2026）は，1 か月規模の個人 GPS 軌跡から日々の経路変更を捉え，Temporal Boltzmann Machine と Graph Attention Network，さらに correlated equilibrium の制約を組み合わせることで，共有情報に対する異質で相関した適応反応を表現しようとしている．ここで重要なのは，深層学習を単なる高精度予測器として用いるのではなく，信念更新や相関した適応の代理構造をもつ実証モデルとして位置づけている点である．学習は理論を置き換えるのではなく，理論が十分に捉えきれてこなかった観測の厚みに踏み込むための手段となる．

この文脈の上に，Transformer や大規模言語モデルを起点とする近年の AI の急速な発展が重なる．Vaswani [8] 以降，系列の長距離依存を捉える注意機構，表現学習，生成モデルは，活動スケジュール，経路系列，都市記述，需要変動の理解に新しい可能性を開いた．さらに，物理法則や保存則を学習に埋め込む PINN は，Raissi [5] 以降，データ駆動型学習と数理モデルの橋渡しとして注目されている．しかし，ここで立ち止まらなければならない．都市と交通の本質は，言語の中だけに存在するのではない．人は文を読んで移動するのではなく，身体をもち，制約を受け，他者と干渉し，空間の抵抗や誘因の中で行動する．そして，その行動は実際にネットワーク負荷を変え，土地利用を変え，拠点の価値を変え，都市そのものを作り替えてしまう．私たちは世界各地の現実の都市や駅を設計し，交通計画を評価し，都市計画に取り組んできた．

その経験から考えるとき，既存の理論はなお，本質的な都市の問題を記述するには十分ではないのではないか．たしかに LLM は強力である．しかしなお最終理論としての弱点がないわけではな

い．都市と交通に必要なのは，LLM の基盤となる「言語」を超えて，現実の「空間」に作用する物理的な選択，相互作用，制約，学習，設計を統合する理論である．人の行動がネットワーク上でどのように干渉し合い，その反復がどのように都市の形を変え，さらにその変わった都市が次の行動をどのように規定するのか．この循環を捉えるには，効用理論，均衡理論，進化動学，最適化，深層学習，さらには物理制約までを横断しなければならない．都市を「語れる」知能ではなく，都市を「説明し，予測し，設計できる」知能をめぐって世界中の研究者たちが訥々と自分たちの課題と向き合い続けている．

最前線のひとつは自動走行であろう．従来のルールベースを超える E2E 的発想では，周囲の行動を外生的に予測し，その上で自車だけを最適化すればよいと考えられがちであった．しかし実際の交通は，相手もまたこちらを見て応答する相互作用の場である．Furuhashi and Hato（2026）は，この限界を正面から捉え，予測依存型の制御から，相互作用そのものを均衡埋め込みで扱う I2I 型の自動運転へと理論の重心を移している．そこでは，shared constraints の下での variational generalized Nash equilibrium，ICNN による学習可能なコスト設計，そして equilibrium-embedded predictive control が統合され，E2E を越えて現実空間の内生性に向き合う知能の姿が模索されている．さらに彼らは，この枠組みを，これまでにない歩行者と自動走行の混在空間における最適制御と最適設計の理論へと拡張しようとしている．

本書が学部生・大学院生に伝えたいのは，まさにこうしたダイナミックな研究のプロセスとその基礎となる理論のバリエーションである．行動モデル，ネットワーク配分，最適化，機械学習は，互いに隣接する技法ではない．それらは，人間の行動がネットワークを通じて都市を変え，その都市の設計が再び人間の行動を変えるという現実の循環構造に迫ろうとする，ひとつの知の系譜である．McFadden [4] から Wardrop [9]，Beckmann [1]，Sandholm [7]，Vaswani [8]，Raissi [5] へと至る流れは，単なる理論の並列ではなく，観測，均衡，設計，学習を再接続し続けてきた歴史である．そして Matsunaga and Hato（2026），Ogawa and Hato（2026），Furuhashi and Hato（2026）に見られるように，その流れは今，公共交通計画，日々の経路選択，自動走行の制御という異なる対象を通じて，新たな段階に入りつつある．理論を下敷きに新たな駅まち空間を設計しなおすこと．格差を是正しうる公共交通を構想し導入すること．今までにない都市と地域を構想し，計画し，実装し，運営すること．そのための基礎理論を，本書を通じて学んでもらいたい．都市と交通を学ぶとは，変わりゆく現実を理解するだけでなく，失われつつある場所と壊れた風景の前で，それでもなお次の危機に向けた新たな空間を構想し直す力を身につけることなのである．

参考文献

[1] Beckmann, M., McGuire, C. B., Winsten, C. B.: Studies in the Economics of Transportation, Yale University Press, New Haven, CT, 1956.

[2] Benders, J. F.: Partitioning procedures for solving mixed-variables programming problems, Numerische Mathematik, Vol. 4, No. 1, pp. 238–252, 1962.

[3] Dantzig, G. B., Ramser, J. H.: The Truck Dispatching Problem, Management Science, Vol. 6, No. 1, pp. 80–91, 1959.

[4] McFadden, D.: Conditional logit analysis of qualitative choice behavior, In Zarembka, P., ed.: Frontiers in Econometrics, Academic Press, New York, pp. 105–142, 1974.

[5] Raissi, M., Perdikaris, P., Karniadakis, G. E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, Vol. 378, pp. 686–707, 2019.

[6] Sandholm, W. H.: Evolutionary Implementation and Congestion Pricing, Review of Economic Studies, Vol. 69, No. 3, pp. 667–689, 2002.

[7] Sandholm, W. H.: Population Games and Evolutionary Dynamics, MIT Press, Cambridge, MA, 2010.

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I.: Attention Is All You Need, Advances in Neural Information Processing Systems, Vol. 30, 2017.

[9] Wardrop, J. G.: Some theoretical aspects of road traffic research, Proceedings of the Institution of Civil Engineers, Part II, Vol. 1, No. 3, pp. 325–362, 1952.

第 2 章

交通量配分

小川大智

序章　交通ネットワーク配分とは

交通量配分問題の設定

私たちが日常的に経験する道路の渋滞は，どのような仕組みで生まれるのだろうか．朝のラッシュ時，多くのドライバーが同じ幹線道路に集中し，互いの走行を妨げ合う．この現象の背後には，個々の利用者が自己の所要時間を最短化しようとする合理的行動と，それが引き起こす相互依存関係がある．**交通量配分**（Traffic Assignment）とは，この複雑な相互作用を数理モデルとして記述し，ネットワーク上の各道路区間（リンク）に流れる交通量を推定する分析手法である [31, 25].

より厳密に言えば，交通量配分問題とは次のように定式化される．交通需要は，出発地（Origin, r）と目的地（Destination, s）の間を移動する交通量 q_{rs}（OD 交通量）として記述される．ネットワーク上には複数の経路が存在し，各利用者はその中から経路を選択する．交通量配分問題の目的は，すべての OD ペアに対する経路選択行動を集約することで，各リンク a の交通量 x_a を推定することである．

この問題が単純でないのは，リンクの混雑度（旅行時間）が交通量そのものに依存するためである．多くの利用者が特定のリンクに集中すれば，そのリンクの旅行時間は増大し，利用者の経路選択に影響を与える．交通量配分は，この**相互依存関係**を整合的に扱う分析の枠組みを提供する．

四段階推定法における位置づけ

交通量配分は，交通計画における需要予測の古典的な手順である**四段階推定法**（Four-Step Model）の第 4 段階を構成する（図 2.1）[31].

四段階推定法では，以下の順序で需要予測が行われる．

第 1 段階：発生・集中（Trip Generation）　ゾーンの人口，土地利用，経済活動量などを説明変数として，各ゾーンから発生・集中する交通量を推定する．

第 2 段階：分布（Trip Distribution）　各ゾーン間の OD 交通量を推定し，OD 行列を作成する．重力モデルやフレーター法などが用いられる．

第 3 段階：機関分担（Modal Split）　自動車，鉄道，バスなど交通手段ごとの利用割合を推定する．ロジットモデルなど離散選択モデルが広く用いられる．

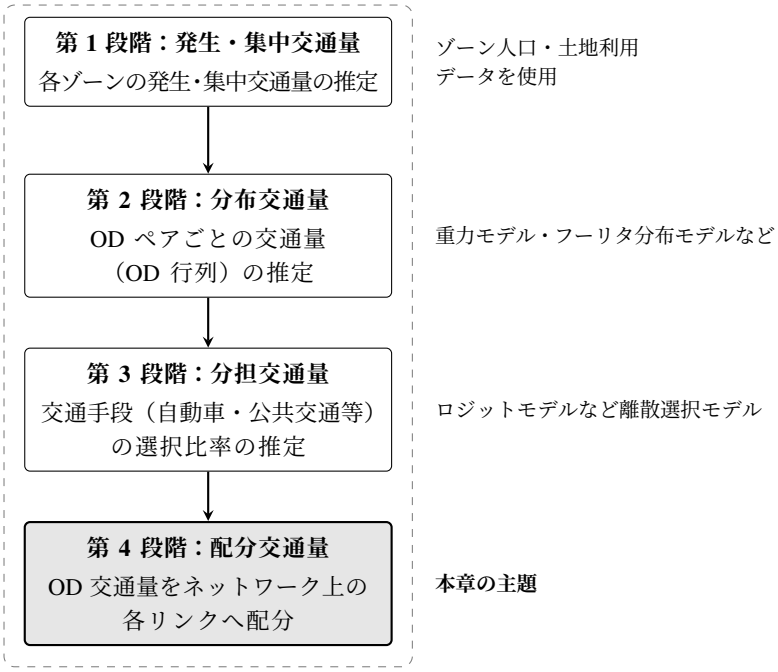


図 2.1: 交通需要予測の四段階推定法（網掛けが本節の対象）

第 4 段階：配分（Traffic Assignment） 前段階で得られた OD 交通量を，道路・鉄道などのネットワーク上の各リンクに割り当てる．**本節の主題**である．

なお，四段階推定法は各段階を逐次的に処理することで計算を簡便にしているが，現実には需要と供給が連立しているため，厳密には各段階の間にフィードバックが必要である．また，近年では多段階を統合したアクティビティベースモデルなども提案されている [25].

交通量配分モデルの応用分野

交通量配分モデルは，次のような幅広い実務・研究場面で活用されている．

道路計画・ネットワーク設計 新規道路の整備や既存道路の拡幅が交通流に与える影響を事前評価する．単純な容量増強が時に逆効果となることを示す Braess のパラドックス（第 2.3 節 第 2.3.4 小節）は，モデルによる事前検討の重要性を示す典型例である [5].

混雑課金・需要管理 外部不経済（混雑の外部性）を内部化するための最適な課金額を算定する．Pigouvian 税の概念を交通に適用した混雑課金（第 2.5 節・第 2.8 節）は，ロンドン，ストックホルム，シンガポールなど世界の主要都市で実施されている [20, 29].

交通制御・情報提供 リアルタイムの交通情報や経路誘導が利用者行動に与える影響を分析し，ネットワーク全体の効率改善策を設計する．

防災・危機管理 大規模災害時の交通需要と道路網の機能低下を同時に考慮し，混雑分布や避難経路の脆弱性を評価する．

	フロー非依存型 （非混雑型）	フロー依存型 （混雑型）
確定的 選択	**All-or-Nothing 配分** 全交通量を最短経路に集中配分 （第 2 章 §2.1）	**利用者均衡配分（UE）** Wardrop 第 1 原則に基づく均衡 （第 3 章）
確率的 選択	**確率的配分** ロジット確率に従う経路選択 Dial のアルゴリズム等 （第 2 章 §2.3, 2.4）	**確率的利用者均衡配分（SUE）** 知覚旅行時間に基づく均衡 （第 4 章）

図 2.2: 交通量配分モデルの 4 象限分類

シミュレーションとの連携 マクロ基本図（MFD）に基づくゾーンシミュレーションやミクロシミュレーション（第 2.9 節）の初期条件・境界条件として配分結果が利用される [6, 13].

交通量配分モデルの分類

交通量配分モデルは，(i) 旅行時間がリンク交通量に依存するか否か（フロー依存性），および (ii) 利用者の経路選択が確定的か確率的かの観点から，大きく 4 種類に分類できる（図 2.2）[31, 25].

All-or-Nothing 配分（AoN） 交通量によらず旅行時間が一定（フロー非依存）と仮定し，各利用者が最短経路にすべての交通量を集中させる手法（第 2.2 節 第 2.2.1 小節）．計算は単純だが，混雑の非線形性を無視する．

確率的配分 フロー非依存の下で，利用者が複数の経路からロジット等の確率モデルに従って選択する手法（第 2.2 節 第 2.2.3 小節・第 2.2.4 小節）．経路の分散利用を反映できる．

利用者均衡配分（User Equilibrium, UE） 旅行時間がフロー依存でも，個々の利用者が自己の旅行時間を最小化した結果，均衡状態が実現すると仮定する手法（第 2.3 節）．Wardrop の第 1 原則を満たす配分であり，実務で最も広く用いられる [30].

確率的利用者均衡配分（Stochastic User Equilibrium, SUE） フロー依存の下で，旅行時間の認知に確率的誤差が存在することを考慮した均衡配分（第 2.4 節）．UE より現実的な行動仮定を持つが，計算コストは増大する [7].

さらに，社会全体の総旅行時間を最小化するシステム最適配分（第 2.5 節），均衡へ至る動的調整過程を記述する進化ゲーム理論的アプローチ（第 2.7 節），混雑課金の経済理論（第 2.8 節），そして各種シミュレーション手法（第 2.9 節）へと議論は展開していく．

交通量配分モデルの特長

交通量配分モデルが交通計画や政策評価において広く用いられてきた理由は，以下の 4 つの特長による [31].

1. **論理の明快さ**：「利用者は自己の旅行時間を最小化する」という単純な行動仮定から出発し，直感的で説得力のある政策提言が得られる．
2. **理論解の一意性**：適切な条件（単調増加なリンクコスト関数）の下で UE 解は一意に存在し，誰が計算しても同じ結果が得られる（第 2.3 節 第 2.3.2 小節）．これは意思決定支援ツールとしての信頼性を担保する．
3. **ミクロ経済学との整合性**：消費者余剰・生産者余剰の概念を用いた余剰分析（第 2.5 節 第 2.5.4 小節）や，混雑課金の最適設計（第 2.8 節）は，ミクロ経済理論と厳密に整合しており，福祉評価・費用便益分析に直接応用できる．
4. **複雑モデルの比較基準**：エージェントベースシミュレーションや機械学習モデルなど，より複雑な新しいアプローチの結果を評価する際，均衡配分モデルの解は理論的な基準点（ベンチマーク）としての役割を果たす．

本章の構成

本章は以下の 9 節から構成される．

第 2.1 節　交通ネットワークの記述 ネットワークの構成要素（ノード，リンク，経路，OD ペア）を定義し，基本変数と関係式，および旅行時間をリンク交通量の関数とする BPR 関数を導入する．

第 2.2 節　非混雑型配分（フロー非依存配分） All-or-Nothing 配分，吸収マルコフ過程配分，確率的配分（ロジット型），および Dial のアルゴリズムを解説する [8].

第 2.3 節　利用者均衡配分（UE） Wardrop の第 1 原則，Beckmann 変換による等価最適化問題の導出，Frank-Wolfe 法，および Braess のパラドックスを扱う [3, 10, 30].

第 2.4 節　確率的利用者均衡配分（SUE） 完全情報仮定を緩和したランダム効用モデルに基づく配分と，ロジット型 SUE の等価最適化問題・解法を解説する [9, 16].

第 2.5 節　システム最適配分（SO）と混雑課金 Wardrop の第 2 原則，UE と SO の比較，限界費用課金の考え方，および余剰分析を扱う [20].

第 2.6 節　動的交通配分（DTA） 静的 UE との対比を通じて DTA の位置づけを整理し，変分不等式による動的利用者均衡（DUE）の定式化と解法の概要を解説する．

第 2.7 節　進化ゲーム理論と交通配分 ポテンシャルゲームとしての UE，集団ゲームの枠組み，代表的な進化動学（Smith ダイナミクス，レプリケーター動学等），および Day-to-day ダイナミクスへの応用を解説する [23, 24].

第 2.8 節　ピグー課金と混雑課金の一般理論 外部性の経済学，第一最善・第二最善課金，Vickrey ボトルネックモデルによる動的課金，および課金収入の再分配・公平性について論じる [29].

第 2.9 節　シミュレーション マクロ基本図（MFD）に基づくゾーンシミュレーションとミクロシミュレーション（車両追従モデル）の基礎を紹介する [6, 13, 12].

また，本章末には補足として，黄金分割法（第 2.10 節），All-or-Nothing 配分の効率的アルゴリズム（第 2.11 節），および利用者均衡と Nash 均衡の等価性の証明（第 2.12 節）を付す [18, 19].

2.1　交通ネットワークの記述

本節では，交通量配分問題を数学的に定式化するための基礎概念を整理する．まずネットワークの構成要素（ノード・リンク・セントロイド・OD ペア・経路）を定義し，次に配分計算で用いる基本変数と関係式を導入する．続いて，リンク旅行時間をリンク交通量の関数として表す**リンクパフォーマンス関数**（BPR 関数）を説明し，最後に本テキストで扱う配分モデルの全体像を概観する．

2.1.1　ネットワークの構成要素

交通ネットワークは，交差点やゾーン重心を表す**ノード**と，道路区間を表す**リンク**からなる**有向グラフ**として表現される [31]．この基本構造に，需要情報（OD ペア・OD 交通量）および経路の概念を加えることで，交通量配分問題が定式化される（図 2.3）．

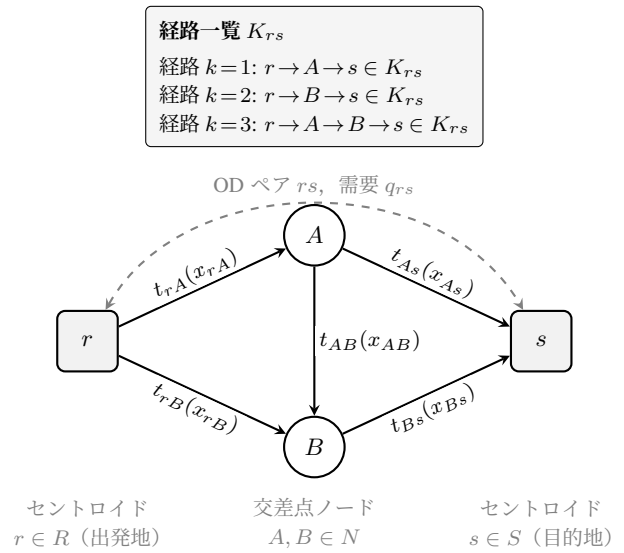


図 2.3: 交通ネットワークの構成要素（4 ノード例）．四角形ノードはセントロイド（$r \in R$, $s \in S$），円形ノードは通常の交差点ノードを表す．OD ペア rs の需要 q_{rs} は 3 本の経路 $K_{rs} = \{k_1, k_2, k_3\}$ に配分される．

各構成要素は以下のように定義される [31, 25].

ノード集合 N　ネットワーク上の節点（交差点・ゾーン重心など）の集合．個々のノードを $i, j \in N$ で表す．実際の道路網では交差点や信号機の位置がノードに対応する．

リンク集合 A　ノード間を結ぶ有向辺（道路区間）の集合．リンクは通常，方向を持つ（一方通行，または往復を別々の有向辺として定義する）．個々のリンクを $a \in A$ で表し，始点ノード i と

終点ノード j の組 (i, j) で特定することもある．

セントロイドおよびゾーン集合　交通需要の発生・集中地点となる特別なノードを**セントロイド**（centroid）と呼ぶ．出発地（Origin）の役割を持つセントロイドの集合を R, 到着地（Destination）の役割を持つセントロイドの集合を S とする．セントロイドは，ゾーン（地区・行政区など）内に分散する起終点を 1 点に集約した仮想的なノードであり，ゾーン重心とも呼ばれる．

OD ペア集合 Ω　出発地 $r \in R$ と目的地 $s \in S$ の組合せ (r, s) を **OD ペア**（OD pair）と呼ぶ．すべての有効な OD ペアの集合を Ω で表す．OD ペア rs 間の移動需要（OD 交通量）を $q_{rs} \geq 0$ とする．

経路集合 K_{rs}　OD ペア rs を結ぶリンク列，すなわち始点が r，終点が s となる有向パスの集合を K_{rs} と表す．各経路を $k \in K_{rs}$ で表し，個々の経路 k は通過するリンクの列として識別される．一般に同一 OD ペアに対して複数の経路が存在しうる．

2.1.2　基本変数と関係式

交通量配分で中心的に扱う変数を表 2.1 にまとめる．

表 2.1: 交通量配分の主要記号

記号	意味	単位（例）
x_a	リンク a の交通量	台/時
$t_a(x_a)$	リンク a の旅行時間（リンクパフォーマンス関数）	分
t_{a0}	リンク a の自由流旅行時間（$x_a = 0$ のとき）	分
C_a	リンク a の交通容量（キャパシティ）	台/時
f_k^{rs}	OD ペア rs の経路 k の交通量	台/時
q_{rs}	OD ペア rs の OD 交通量（需要）	台/時
c_k^{rs}	OD ペア rs の経路 k の旅行時間	分
u_{rs}	OD ペア rs の均衡最小旅行時間	分
$\delta_{a,k}^{rs}$	リンク a が経路 k^{rs} に含まれるなら 1，含まれないなら 0	—

これらの変数の間には以下の基本的な関係式が成り立つ [31].

リンク交通量と経路交通量の関係

リンク a の交通量 x_a は，そのリンクを通過するすべての経路の交通量の和として表される：

$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs}\, \delta_{a,k}^{rs} \tag{2.1}$$

ここで $\delta_{a,k}^{rs}$ は**リンク–経路付替行列**（link-path incidence matrix）の要素であり，経路 k^{rs} がリンク a を通るとき $\delta_{a,k}^{rs} = 1$，通らないとき $\delta_{a,k}^{rs} = 0$ の値をとる．

流量保存条件（非負条件・需要充足条件）

各 OD ペア $rs \in \Omega$ に対し，経路交通量は以下の条件を満たす必要がある：

$$\sum_{k \in K_{rs}} f_k^{rs} = q_{rs}, \qquad f_k^{rs} \geq 0 \quad \forall\, k \in K_{rs},\ \forall\, rs \in \Omega \tag{2.2}$$

第 1 式は，ある OD ペア間の全需要 q_{rs} が何らかの経路に必ず配分されることを要求する（需要充足条件）．第 2 式は交通量の非負性を保証する（負の交通量は物理的に無意味である）．

経路旅行時間

経路 k^{rs} の旅行時間 c_k^{rs} は，その経路上のリンク旅行時間の和として計算される：

$$c_k^{rs} = \sum_{a \in A} t_a(x_a)\, \delta_{a,k}^{rs} \tag{2.3}$$

式 (2.1)〜(2.3) は，未知変数として経路交通量 $\{f_k^{rs}\}$ またはリンク交通量 $\{x_a\}$ を持つ．両者は式 (2.1) によって結ばれており，一方が決まれば他方も定まる．実際の配分計算ではリンク交通量 $\{x_a\}$ を主要変数として扱うことが多い（第 2.3 節 第 2.3.2 小節参照）[3].

2.1.3 リンクパフォーマンス関数（BPR 関数）

リンク旅行時間 t_a がリンク交通量 x_a に依存する場合，その関係を表す関数を**リンクパフォーマンス関数**（Link Performance Function, LPF）または**旅行時間関数**と呼ぶ．交通量の増加に伴って旅行時間が長くなる混雑現象を定量的に表現するために用いる．

実務で最も広く使われるのが，米国道路局（Bureau of Public Roads, BPR）が提案した **BPR 関数**（BPR function）である [31]：

$$t_a(x_a) = t_{a0}\left\{1 + \alpha\left(\frac{x_a}{C_a}\right)^{\beta}\right\} \tag{2.4}$$

ここで t_{a0} は自由流旅行時間（交通量ゼロ時の所要時間），C_a は交通容量，α, β はパラメータである．実務的に広く用いられる標準値は $\alpha = 0.15$，$\beta = 4$ である．

式 (2.4) の主な性質は以下のとおりである（図 2.4）．

- $x_a = 0$ のとき $t_a(0) = t_{a0}$（自由流速度での走行）．
- $x_a = C_a$ のとき $t_a(C_a) = (1+\alpha)\, t_{a0} = 1.15\, t_{a0}$（標準値では，容量交通量における旅行時間は自由流の 15% 増し）．
- $x_a > C_a$ でも関数値は有限であり，過飽和状態を連続的に表現できる．
- $\beta > 1$ のとき，$t_a(x_a)$ は $x_a \geq 0$ 上で単調増加かつ狭義凸である．この凸性は，利用者均衡配分の等価最適化問題が一意な解を持つことの十分条件として重要な役割を果たす（第 2.3 節 第 2.3.2 小節参照）．

非混雑型（フロー非依存）と混雑型（フロー依存）の区別

旅行時間がリンク交通量に依存しない（$t_a = \text{const}$）配分モデルを**フロー非依存型**（flow-independent）または**非混雑型**と呼ぶ．一方，BPR 関数のように旅行時間が交通量に依存するモデルを**フロー依存型**または**混雑型**と呼ぶ．この区別は，第 2.2 節（非混雑型配分）と第 2.3 節・第 2.4 節（均衡配分）の節立てに対応する（次節参照）．

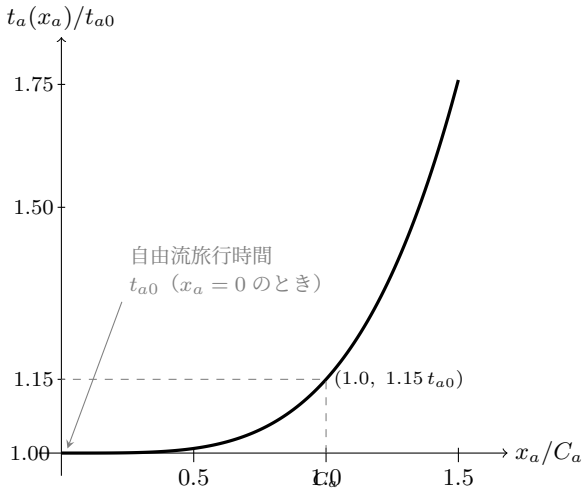


図 2.4: BPR 関数のグラフ（$\alpha = 0.15, \beta = 4$）．横軸は交通量 x_a を容量 C_a で正規化した値，縦軸は旅行時間 $t_a(x_a)$ を自由流旅行時間 t_{a0} で正規化した値．$x_a = C_a$（容量到達点）での旅行時間は $1.15\,t_{a0}$ となる（破線）．

クイズ 1：首都高 3 号線，朝ラッシュで何分遅れる？

首都高速 3 号渋谷線（用賀 → 大橋 JCT，自由流旅行時間 $t_{a0} = 10$ 分，容量 $C_a = 2{,}000$ 台/時）で BPR パラメータ $\alpha = 0.15$，$\beta = 4$ を仮定します．交通量が容量ちょうど（$x_a = C_a$）のとき，旅行時間はいくらになるでしょうか？

(ア) 10.0 分（変わらない）

(イ) 11.5 分（1.15 倍）

(ウ) 17.6 分（1.76 倍）

(エ) 25.0 分（2.5 倍）

→ 正解・解説は節末を参照

2.1.4　配分モデルの全体像

交通量配分モデルは，「フロー依存性」と「経路選択の確定性・確率性」という 2 軸で大きく 4 種類に分類できる [25]．この 4 象限分類は序章の図 2.2 に示したとおりであり，本テキストの構成の骨格をなす．

第 2.2 節　非混雑型配分（フロー非依存配分）　旅行時間が交通量に依存しない（$t_a = \text{const}$）と仮定する．全利用者が最短経路にすべての交通量を集中させる **All-or-Nothing 配分**（第 2.2.1 小節）と，利用者が経路をロジットモデルに従って確率的に選ぶ**確率的配分**（第 2.2.3 小節・

第 2.2.4 小節）を扱う．後者の効率的な計算法として **Dial のアルゴリズム**を紹介する [8].

第 2.3 節　利用者均衡配分（UE）　混雑を考慮する（フロー依存型）かつ利用者が完全情報のもとで確定的に最短経路を選ぶと仮定する．**Wardrop の第 1 原則**を満たす均衡状態を，Beckmann 変換による等価最適化問題として定式化し，**Frank-Wolfe 法**で求解する [3, 10]．また，新規リンクの追加が全利用者の旅行時間を悪化させうる **Braess のパラドックス**を紹介する [5].

第 2.4 節　確率的利用者均衡配分（SUE）　混雑を考慮しつつ，利用者が旅行時間の認知に確率的誤差を持つと仮定する．ロジット型 SUE の等価最適化問題と各種解法を扱う [7, 9].

第 2.5 節〜第 2.9 節　発展的トピック　上記を基礎として，社会全体の総旅行時間を最小化する**システム最適配分**と **Pigouvian 混雑課金**（第 2.5 節・第 2.8 節 [20]），UE をポテンシャルゲームとして捉え均衡収束過程を記述する**進化ゲーム理論と Day-to-day ダイナミクス**（第 2.7 節 [24]），および**マクロ・ミクロシミュレーション**（第 2.9 節 [6]）へと議論は展開していく．

本節のまとめ

本節で整理した主要な概念を振り返る：

1. 交通ネットワークはノード集合 N・リンク集合 A・セントロイド・OD ペア集合 Ω・経路集合 K_{rs} の 5 要素で表現される有向グラフであり，交通量配分問題の基本的な解空間を定義する（第 2.1.1 小節）.
2. リンク交通量 x_a・経路交通量 f_k^{rs}・リンク–経路付替行列 $\delta_{a,k}^{rs}$ の 3 変数に加え，リンク–経路関係式・流量保存条件・経路旅行時間の 3 式が基本関係式をなし，配分計算ではリンク変数 $\{x_a\}$ を主要変数として扱う（第 2.1.2 小節）.
3. BPR 関数 $t_a(x_a) = t_{a0}\{1 + \alpha(x_a/C_a)^\beta\}$ は自由流旅行時間・容量・混雑回帰係数を組み込んだ実務的な混雑関数であり，$\beta > 1$ における狭義単調増加・狭義凸の性質が UE 解の一意性の重要な十分条件となる（第 2.1.3 小節）.
4. 交通量配分モデルは「フロー依存性」と「経路選択の確定性・確率性」の 2 軸で 4 象限に分類され，本テキストの章構成（第 2.2 節〜第 2.9 節）の基本的な骨格をなす（第 2.1.4 小節）.

クイズ 1 正解・解説

正解：(イ)

$t_a = 10 \times [1 + 0.15 \times 1^4] = 11.5$ 分．容量ちょうどではわずかな増加にとどまりますが，$x_a/C_a = 1.5$ になると 17.6 分に急増します．4 乗の効果が効くのは容量を大きく超えてからです.

2.2　非混雑型配分（フロー非依存配分）

本節では，リンク旅行時間が交通量に依存しない**フロー非依存型**の配分モデルを扱う．第 2.1 節 第 2.1.3 小節 で述べたように，非混雑型ではリンク旅行時間 t_a を定数として扱うため，経路費用

c_k^{rs} が交通量とは無関係に決まる．このシンプルな仮定のもとで，経路選択の「確定性」と「確率性」という観点から 4 種類のモデルを順に紹介する（図 2.2，本節部分）．

第 2.2.1 小節 All-or-Nothing 配分　各 OD ペアの利用者が唯一の最短経路にすべての交通量を集中させる，最も単純な確定的配分モデルである．混雑を考慮しないため現実の交通分散は再現できないが，後続の均衡配分（第 2.3 節）のアルゴリズムに繰り返し現れる基本部品でもある．

第 2.2.2 小節 吸収マルコフ過程配分　ネットワーク上の移動を確率的なランダムウォークとして定式化するモデルである．経路を陽に列挙せずに流量を求められ，循環フローも原理上取り扱える [1]．

第 2.2.3 小節 確率的配分（ロジット配分）　利用者が経路費用に応じたロジット確率で経路を選択するモデルである [16]．パラメータ θ によって「最短経路への集中」と「均等分散」の間を連続的に制御できる．

第 2.2.4 小節 Dial のアルゴリズム　経路を陽に列挙せずにロジット配分を実現する効率的な計算手法である [8]．前進・後退の 2 パス処理によって計算量を大幅に削減する．

本節で新たに用いる主要記号を表 2.2 にまとめる（基本記号は表 2.1 参照）．

表 2.2: 非混雑型配分の主要記号

記号	意味	単位（例）
$c(i)$	ノード i への最短経路費用（起点からの最小旅行時間）	分
F_j	先行ポインタ（最短経路ツリー上のノード j の直前ノード）	—
y_{ij}	AoN 集計補助フロー（起点からリンク (i,j) に集計される交通量）	台/時
P_{ij}	ノード i からノード j への遷移確率（マルコフ配分）	—
$\mathbf{N}$	基本行列 $(I-Q)^{-1}$（一時的状態間の平均通過回数を表す行列）	—
v_j^*	目的地までの最短残余費用（目的地考慮型ロジットで事前計算）	分
θ	ロジットの感度パラメータ（$\theta \to \infty$ で AoN 配分に収束）	分 $^{-1}$
P_k^{rs}	OD ペア rs の利用者が経路 k を選択するロジット確率	—
$L[i \to j]$	Dial のアルゴリズムにおけるリンク尤度	—
$W[i \to j]$	Dial のアルゴリズムにおけるリンクウェイト	—

2.2.1　All-or-Nothing 配分

基本的考え方

All-or-Nothing（AoN）配分は，各利用者がリンク旅行時間を一定と見なしたうえで最短経路のみを選択するという行動仮定に基づく [31]．全利用者が同じ経路情報を持ち，各 OD ペアの最短経路は一意に定まると仮定すると，すべての交通量 q_{rs} は 1 本の最短経路に集中する．残りの経路には交通量がまったく流れないという「all or nothing（全か無か）」の割り当て方がその名前の由来である（図 2.5）．

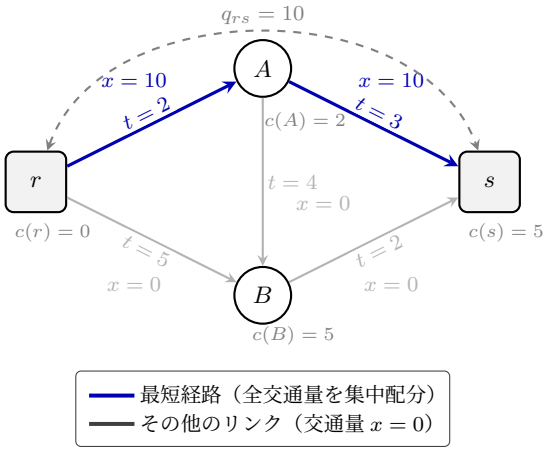


図 2.5: All-or-Nothing 配分の例（4 ノードネットワーク，OD 交通量 $q_{rs} = 10$）．最短経路 $r \to A \to s$（費用 5）に全量が配分され，他のリンクには交通量が流れない．ノード横の $c(i)$ は起点 r からの最短経路費用を表す．

表 2.3: 図 2.5 のネットワークにおける最短経路費用と先行ポインタ（起点 r，Dijkstra 法の出力）．

ノード j	最短経路費用 $c(j)$	先行ポインタ F_j
r	0	—
A	2	r
B	5	r
s	5	A

アルゴリズム

AoN 配分は以下の手順で実施する [31, 25].

Step 1（**初期化**） 全リンクの交通量をゼロに初期化：$x_{ij}^{(0)} = 0$，$n \leftarrow 1$.

Step 2（**最短経路計算**） 起点 $o \in \mathcal{R}$ を選択する．Dijkstra 法などで起点 o から全ノードへの最短経路費用 $\{c(j)\}$ および先行ポインタ $\{F_j\}$ を計算する．

Step 3（**ツリー順走査・集計**） 最短経路ツリーを $c(j)$ の降順（起点から遠い順）にたどりながら，各ノード $j \in \mathcal{N} \setminus \{o\}$ に対し $i \coloneqq F_j$（先行ポインタ）とおくと，リンク (i, j) の上流フロー y_{ij} を以下の式で集計する：

$$y_{ij} = q_{oj} + \sum_{m \in \mathcal{O}_j^{(o)}} y_{jm} \tag{2.5}$$

ここで q_{oj} は起点 o・目的地 j の OD 交通量，$\mathcal{O}_j^{(o)}$ は最短経路ツリー上でノード j から流出するリンクの終点集合である．式 (2.5) の直感的意味は「ノード j を通過する交通量は，j 自身を目的地とする OD 需要と，j 以降のノードへ向かう流量の合計」である．リンク交通量を $x_{ij}^{(n)} = x_{ij}^{(n-1)} + y_{ij}$ で更新する．

Step 4（**終了判定**） $n < |\mathcal{R}|$ ならば $n \leftarrow n + 1$ として Step 2 へ戻る．全起点を処理したら終了．

Step 3 では，最短経路ツリーが**木構造（ツリー）**をなすことを利用する．各起点 o の最短経路ツリーは無閉路で，宛先に向かう唯一の経路が定まっているため，葉ノード（目的地に相当するノード）から根ノード（起点 o）に向かって流量を逆算的に集計できる（式 (2.5)）．経路を陽に列挙・保存する必要がないため，計算量は $O(|\mathcal{N}|)$ と効率的である（詳細は第 2.11 節参照）．

問題点

AoN 配分は単純で計算コストが低い反面，以下の問題点を持つ．

- 混雑を考慮しないため，最短経路に交通量が集中し，現実の分散を再現できない．
- OD ペアごとに最短経路が 1 本しか選ばれないため，並行リンクへの配分が過小になりやすい．
- 旅行時間が更新されないので，配分後の旅行時間と配分の根拠（定数旅行時間）が矛盾する．

混雑を考慮した均衡を求める Frank-Wolfe 法（第 2.3 節 第 2.3.3 小節）では，各反復で現在の旅行時間に基づく AoN 配分を繰り返す「補助解問題」にこのアルゴリズムが用いられる．すなわち，AoN 配分は均衡配分の部品として重要な役割を果たす．

クイズ 2：カーナビが全員に同じ「最短ルート」を案内したら？

1000 人のドライバー全員がカーナビやルート案内アプリを使い，同じ「最短ルート」に誘導されました．その後，何が起きるでしょうか？

(ア) 全員が最短経路を走れるので，全体の旅行時間が最短になる

(イ) 最短経路に交通量が集中し，かえって渋滞が悪化する

(ウ) 混雑が嫌で他のルートに移る人が出るため，自然に分散する

→ 正解・解説は節末を参照

2.2.2　吸収マルコフ過程配分

基本的考え方

AoN 配分では，各利用者が決定論的に最短経路を選ぶと仮定した．これに対して**吸収マルコフ過程配分**は，ネットワーク上の移動を確率的なプロセスとして定式化するモデルである [1]．利用者は各ノードを訪問するたびに次のリンクを確率的に選択し，目的地ノードに到達した時点でその OD トリップが完了する（吸収される）．

この定式化は次の利点を持つ．

1. 経路を陽に列挙することなく，リンクあたりの期待通過回数として流量を解析的に求められる．
2. 一般的な遷移確率行列を用いるため，循環フロー（同一リンクを複数回通過するルート）を原理上取り扱える [1].
3. ロジット型確率を遷移確率として採用することで，確率的な経路選択を表現できる．

定式化

ノード集合を**吸収状態**（目的地の集合 $\mathcal{D} \subset \mathcal{N}$）と**一時的状態**（それ以外のノード $\mathcal{T} = \mathcal{N} \setminus \mathcal{D}$）に分割する．OD ペア rs に対して，ノード間の遷移確率行列 P を次のように整理する：

$$P = \begin{pmatrix} Q & R \\ O & I \end{pmatrix} \tag{2.6}$$

ここで，Q（$|\mathcal{T}| \times |\mathcal{T}|$ 行列）は一時的状態間の遷移確率，R（$|\mathcal{T}| \times |\mathcal{D}|$ 行列）は一時的状態から吸収状態への遷移確率，O はゼロ行列，I は単位行列である．遷移確率の行和は 1，すなわち $Q\mathbf{1} + R\mathbf{1} = \mathbf{1}$（ただし $\mathbf{1}$ は全成分 1 のベクトル）が成立する．

ランダムウォークが最終的に吸収状態へ到達することを保証するため，Q がすべての固有値が絶対値 1 未満であることを仮定する．このとき**基本行列**（fundamental matrix）

$$\mathbf{N} = (I - Q)^{-1} = I + Q + Q^2 + \cdots \tag{2.7}$$

が存在する．$\mathbf{N}_{ij}$（$\mathbf{N}$ の (i,j) 成分）は，起点 i から出発したウォーカーが一時的状態 j を平均何回通過するかを表す．

OD ペア rs（$r \in \mathcal{T}, s \in \mathcal{D}$）に対して，リンク $a = (i,j)$（$i \in \mathcal{T}$）の期待フローは以下のように求められる：

$$x_a^{rs} = q_{rs} \cdot \mathbf{N}_{ri} \cdot P_{ij} \tag{2.8}$$

$$x_a = \sum_{rs \in \Omega} x_a^{rs} \tag{2.9}$$

式 (2.8) の直感的意味は「OD ペア rs の q_{rs} 人のうち，ノード i を通過するのは平均 $\mathbf{N}_{ri}$ 人で，そのうちリンク (i,j) を選ぶ割合が P_{ij}」である．

遷移確率の設定

遷移確率 P_{ij} の設定方法はモデルの仮定に依存する．最も単純な**ローカルロジット型**では，リンク旅行時間のみで選択確率を定める：

$$P_{ij}^{\mathrm{local}} = \frac{\exp(-\theta t_{ij})}{\displaystyle\sum_{k \in \mathcal{A}(i)} \exp(-\theta t_{ik})} \tag{2.10}$$

ただしこの設定は目的地の方向を考慮しないため，ランダムウォークが循環フローを生じやすい（後述）．Akamatsu（1996）が提案する**目的地考慮型ロジット**では，リンク旅行時間に目的地 s までの最適残余費用 v_j^*（後退 Dijkstra 法により事前計算）を加えた有効費用 $t_{ij} + v_j^*$ に基づいて遷移確率を定める：

$$P_{ij} = \frac{\exp\bigl(-\theta(t_{ij} + v_j^*)\bigr)}{\displaystyle\sum_{k \in \mathcal{A}(i)} \exp\bigl(-\theta(t_{ik} + v_k^*)\bigr)} \tag{2.11}$$

ここで $\mathcal{A}(i)$ はノード i から出るリンクの端点集合である．v_j^* が小さい（目的地に近い）ノードほど高確率で選ばれるため，経路は自然に目的地方向へ誘導される．$\theta \to \infty$ のとき最短経路リンクのみ $P_{ij} \to 1$ となり，AoN 配分に収束する．

リスク回避型利用者均衡（RAUE）への展開

Bell & Cassir（2002）は，旅行時間に不確実性がある場合に各利用者が期待費用 μ_p だけでなく標準偏差 σ_p も考慮して経路選択を行うと仮定し，**リスク回避型利用者均衡（Risk-Averse User Equilibrium, RAUE）**を定式化した [4]．一般化知覚費用 $\mu_p + \lambda \sigma_p$（$\lambda \geq 0$：リスク回避度パラメータ）の最小化を各交差点でのゲーム理論的（ミニマックス）逐次意思決定として定式化し，マルコフ連鎖によって解を求めている．RAUE は通常のロジット型 SUE とは異なる経路分担を与えるとともに，利用者均衡（第 2.3 節参照）の枠組みを前提としている点で，本節が扱うフロー非依存・非均衡配分の単純な拡張とは性格が異なることに注意する．

注意点

マルコフ過程配分のうちローカルロジット型（式 (2.10)）を用いた場合，目的地方向が考慮されないため同一リンクを複数回通過する循環フローが生じやすく，交通量が過大推計される可能性がある．この問題は，目的地考慮型ロジット（式 (2.11)）を用いることで大幅に緩和でき，第 2.2.4 小節の Dial アルゴリズムのように前進方向のリンクのみを許容する効率的リンク制限（$c(i) < c(j)$ のリンクのみ使用）を課すことで完全に解消できる [1, 8].

2.2.3 確率的配分（ロジット配分）

基本的考え方

AoN 配分において利用者は最短経路のみを選択すると仮定したが，現実の利用者は旅行時間の認知に誤差を持つため，より長い経路が選ばれる場合も観察される．**確率的配分**は，経路選択をランダム効用理論に基づいて確率的に定式化するモデルである [16, 7].

ランダム効用モデルでは，利用者が経路 k^{rs} を選択したときの**知覚経路費用**を

$$\tilde{c}_k^{rs} = c_k^{rs} + \varepsilon_k \tag{2.12}$$

と表す（c_k^{rs}：実際の経路費用，ε_k：ランダム誤差項）．誤差項 ε_k がそれぞれ独立に**ガンベル分布** $\mathrm{Gumbel}(0, \theta^{-1})$ に従うと仮定すると，利用者が知覚費用を最小化する条件から，経路 k^{rs} の**選択確率**が閉形式で得られる [16]：

$$P_k^{rs} = \frac{\exp(-\theta c_k^{rs})}{\displaystyle\sum_{k' \in \mathcal{K}_{rs}} \exp(-\theta c_{k'}^{rs})} \tag{2.13}$$

この式を**多項ロジット（MNL）モデル**と呼ぶ．OD ペア rs の経路 k への期待配分交通量は

$$E[f_k^{rs}] = q_{rs} \cdot P_k^{rs} \tag{2.14}$$

で与えられる．

パラメータ θ の解釈

パラメータ $\theta > 0$ は経路選択の**感度**（逆分散）を制御する：

- $\theta \to \infty$：誤差が小さくなり，全利用者が最短経路を選択する（AoN 配分に収束）．
- $\theta \to 0$：誤差が大きくなり，経路費用によらず全経路が等確率で選択される（均等配分）．
- 中間の θ：費用が低い経路ほど高確率で選ばれるが，費用の高い経路にも一定割合が流れる．

式 (2.13) において，短経路と長経路の費用差が大きくなるほど，また θ が大きいほど，確率比 $P_k^{rs}/P_{k'}^{rs}$ が指数的に拡大し，短経路に交通量が集中する様子が確認できる．

ロジットモデルの性質：IIA 特性

式 (2.13) は，任意の 2 経路 k, k' の選択確率比が

$$\frac{P_k^{rs}}{P_{k'}^{rs}} = \exp(-\theta(c_k^{rs} - c_{k'}^{rs})) \tag{2.15}$$

と他の経路の費用に依存しないという性質を持つ．これを**無関係な選択肢からの独立性**（Independence of Irrelevant Alternatives, IIA）という [16]．IIA 特性はモデルを扱いやすくする一方，経路が重複部分を多く共有する場合（例：高速道路の複数のインターチェンジを経由する経路）に選択確率を過大評価するという問題（経路重複問題）を引き起こす．この問題への対応として，Nested Logit モデルやプロビットモデルが用いられることがある [7].

経路の列挙問題

式 (2.13) を直接適用するには，全経路集合 $\mathcal{K}_{rs}$ を陽に列挙する必要がある．しかし実際のネットワークでは経路数が指数的に多く，列挙が現実的でない場合が多い．次節では，経路を列挙せずにロジット配分を実現する Dial のアルゴリズムを紹介する．

クイズ 3：あなたの θ はいくつですか？

感度パラメータ θ が大きいほど，コスト差に対して鋭敏に反応し最短経路を厳格に選ぶドライバーを表します．次の 3 人のうち，θ が最も大きいのは誰でしょうか？

(ア) 東京 23 区内を 10 年以上走り続けているタクシードライバー

(イ) 地方から上京したばかりで土地勘のない大学生（ナビなし）

(ウ) リアルタイム渋滞データを見ながら走るカーナビ搭載のトラック

→ 正解・解説は節末を参照

2.2.4　Dial のアルゴリズム

基本的考え方

Dial（1971）が提案したアルゴリズムは，第 2.2.3 小節 のロジット確率を経路列挙なしに計算する効率的な方法である [8]．AoN 配分と同様に，起点からの最短経路ツリーを構築したうえで，**合理的リンク**（efficient link，最短経路方向に進むリンク）のみを対象にロジット確率の計算を行う．

各リンク (i,j) の**リンク尤度**（link likelihood）$L[i \to j]$ を

$$L[i \to j] = \begin{cases} \exp\big[\theta\{c(j) - c(i) - t_{ij}\}\big] & \text{if } c(i) < c(j) \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

と定義する（$c(i)$：起点からノード i への最短経路費用）．分子の指数部の直感的解釈：$c(j)-c(i)$ は「最短経路上でリンク (i,j) が貢献する費用増分」，t_{ij} は実際のリンク費用であり，最短経路上のリンクでは両者が一致して $L=1$，迂回するリンクでは $t_{ij} > c(j)-c(i)$ となり $L<1$ となる（後退リンク $c(i) \geq c(j)$ は $L=0$）[8]．

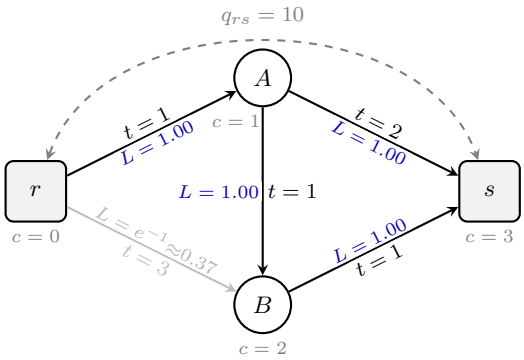
図 2.6 に数値例を示す（$\theta = 1$）．

アルゴリズムの詳細

Dial のアルゴリズムは以下の 4 ステップからなる [8]．以下では，$\mathcal{O}_i = \{j \in \mathcal{N} : (i,j) \in \mathcal{A}\}$（ノード i の後継ノード集合），$\mathcal{I}_i = \{j \in \mathcal{N} : (j,i) \in \mathcal{A}\}$（ノード i の先行ノード集合）と記す．

Step 1（**最短経路**）　起点 r から全ノードへの最短経路費用 $\{c(i)\}$ を Dijkstra 法で計算する．全リンクの配分交通量を初期化 $y_{ij} = 0$.

Step 2（**リンク尤度**）　式 (2.16) により全リンクのリンク尤度 $L[i \to j]$ を計算する．最短経路上のリンクは $L=1$，後退リンクは $L=0$，迂回リンクは $0 < L < 1$ となる．



前進処理（Step 3）：$c(i)$ の昇順（$r \to s$ 方向）にノードを処理
後退処理（Step 4）：$c(i)$ の降順（$s \to r$ 方向）にノードを処理
青字：$L > 0$（合理的リンク）　灰色：$L < 1$（非最短方向）

図 2.6: Dial のアルゴリズムにおけるリンク尤度の計算例．ネットワーク：リンク費用 $t_{rA} = 1$, $t_{rB} = 3, t_{As} = 2, t_{AB} = 1, t_{Bs} = 1$，最短経路費用 $c(r) = 0, c(A) = 1, c(B) = 2, c(s) = 3$，$\theta = 1$．合理的リンク（$c(i) < c(j)$）には $L \geq 0$ が付与される．$r \to B$ は最短経路方向ではあるが，最短ではない（$L = e^{-1} \approx 0.37$）．

Step 3（**前進処理**）　$c(i)$ の昇順（起点 r に近い順）にノードを処理し，各リンク $(i \to j)$ のリンクウェイト $W[i \to j]$ を以下で計算する：

$$W[i \to j] = \begin{cases} L[i \to j] & \text{if } i = r \\ L[i \to j] \displaystyle\sum_{m \in \mathcal{I}_i} W[m \to i] & \text{otherwise} \end{cases} \tag{2.17}$$

$W[i \to j]$ は「起点 r からリンク $(i \to j)$ を通る全経路の尤度の総和」であり，確率的な経路選択のウェイトとして機能する．

Step 4（**後退処理**）　$c(i)$ の降順（起点 r から遠い順）にノードを処理し，リンク交通量 y_{ij} および累積交通量 x_{ij} を以下で計算する：

$$y_{ij} = \left(q_{rj} + \sum_{m \in \mathcal{O}_j} y_{jm} \right) \frac{W[i \to j]}{\displaystyle\sum_{m \in \mathcal{I}_j} W[m \to j]} \tag{2.18}$$

$$x_{ij}^{(n)} = x_{ij}^{(n-1)} + y_{ij} \tag{2.19}$$

式 (2.18) の括弧内は「ノード j を通過すべき総流量」（目的地 j への OD 需要 q_{rj} とノード j 以降へ向かう流量 y_{jm} の合計），右辺の分数はリンクウェイト W の比率に基づく配分比率を表す．

全起点 $r \in \mathcal{R}$ について Step 1〜4 を繰り返し，すべてのリンク交通量 x_{ij} を累積することでロジット確率に基づく非混雑型配分が完成する．

AoN 配分との関係

Dial のアルゴリズムは $\theta \to \infty$ の特殊例として AoN 配分を包含する．$\theta \to \infty$ では最短経路上のリンクのみ $L = 1$（その他 $L \to 0$）となるため，ウェイト W は最短経路ツリー上のリンクにのみ集中し，後退処理での配分比率が AoN 配分の流量集計（式 (2.5)）と一致する．逆に $\theta \to 0$ では全リンクの $L \to 1$ となり，交通量が均等に分散する．

計算効率

経路を陽に列挙する場合の計算量は OD ペア数と経路数の積に比例するが，Dial のアルゴリズムは前進・後退の 2 パス処理で $O(|\mathcal{N}| + |\mathcal{A}|)$ の計算量で完了する [8]．ただし，式 (2.16) において「後退リンク（$c(i) \geq c(j)$）を無視する」合理的経路の仮定が必要である点に注意する．この仮定により循環フローは排除され，有向非巡回グラフ（DAG）上の分析と等価になる．循環フローを考慮すべき状況では，第 2.2.2 小節 のマルコフ過程配分が適切である [1].

本節のまとめ

本節で示した主要な結果を整理する：

1. **AoN 配分**は，すべての交通量を最短経路 1 本に集中させる最も単純な確定的配分モデルであり，混雑を考慮した均衡配分（Frank-Wolfe 法）の補助解問題において基本的な構成要素として繰り返し利用される（第 2.2.1 小節）．
2. **吸収マルコフ過程配分**は，ネットワーク上の移動を確率的ランダムウォークとして定式化し，基本行列 $(I - Q)^{-1}$ によって経路を陽に列挙せずにリンク流量を解析的に求める手法であり，目的地考慮型ロジットを遷移確率として採用することで循環フロー問題を緩和できる（第 2.2.2 小節）．
3. **ロジット配分**は，知覚費用の誤差項がガンベル分布に従う仮定から多項ロジット（MNL）モデルを導き，パラメータ θ によって AoN 配分（$\theta \to \infty$）と均等配分（$\theta \to 0$）の間を連続的に制御する；IIA 特性が計算を容易にする一方，経路重複問題に注意が必要である（第 2.2.3 小節）．
4. **Dial のアルゴリズム**は，リンク尤度に基づく前進・後退の 2 パス処理によって，経路を陽に列挙せず $O(|\mathcal{N}| + |\mathcal{A}|)$ の計算量でロジット配分を実現し，$\theta \to \infty$ の特殊例として AoN 配分を包含する（第 2.2.4 小節）．

クイズ 2 正解・解説

正解：（イ）

カーナビが全員に「同じ 1 つの最短経路」を返すとき，リンク旅行時間が交通量に依存しない仮定のもとでは全量が集中します——それが AoN 配分の挙動です．（ウ）の「反応して分散が起きる」プロセスこそが，均衡配分の核心です．

クイズ 3 正解・解説

正解：（ア）または（ウ）
経験・情報が豊富なほど θ は大きくなります．（イ）のような不確実性の高い利用者が，第 4 章の SUE で扱う「知覚誤差の大きな利用者」の典型です．現実のネットワークには（ア）〜（ウ）が混在しており，均一の θ を仮定する UE より SUE が現実に近い理由がここにあります．

2.3　利用者均衡配分（UE）

第 2.1 節で整備した記法と，第 2.2 節で扱ったフロー非依存型配分を踏まえ，本節では**利用者均衡配分**（User Equilibrium Assignment, UE）を体系的に扱う．UE は，リンク旅行時間がリンク交通量に依存する混雑ネットワークにおいて，利用者が自己の旅行時間を最小化しようとする合理的行動の均衡状態を記述するモデルであり，交通量配分理論の根幹をなす [30, 3].

本節の構成は次のとおりである．第 2.3.1 小節 で均衡条件を与える Wardrop の第 1 原則を導入し，第 2.3.2 小節 でこの条件を等価な凸最適化問題（Beckmann 変換）に変換する．第 2.3.3 小節では，この最適化問題の標準的な解法である **Frank-Wolfe 法**を他手法との比較も交えて説明する．第 2.3.4 小節 では，UE の本質的な性質を浮き彫りにする事例として **Braess のパラドックス**を紹介する．

クイズ 4：東京駅 → 羽田空港，なぜ 2 経路に分かれる？

平日朝，東京駅から羽田空港へ向かうタクシーには大きく 2 経路があります．首都高速（C1 環状 →1 号羽田線，高速・料金あり）と，国道 1 号・第一京浜経由（一般道，無料）です．どちらかが明らかに速ければ全車がそちらに集中するはずなのに，実際には両経路に車が分散しています．この分散を最もよく説明するのはどれでしょうか？（注：このモデルでは旅行時間のみを考慮します．）

(ア) 東京都が法律でルートごとの交通量上限を定めているから

(イ) ドライバーがそれぞれ旅行時間を最小化した結果，両経路の旅行時間が均等になっているから

(ウ) ランダムにルートを選んでいるので，自然と分散するから

(エ) 首都高の料金を嫌うドライバーが一定数いるから

→ 正解・解説は節末を参照

2.3.1　Wardrop の第 1 原則

均衡条件の直感的背景

混雑を考慮したネットワークでは，リンク旅行時間 $t_a(x_a)$ はリンク交通量 x_a の増加関数である（第 2.1 節 第 2.1.3 小節 参照）．このとき，多数の利用者が同時に自己の旅行時間を最小化しようとすれば，最初は最短経路に交通量が集中して旅行時間が上昇し，やがて各利用者が経路をそれ以上変更する動機を失う状態（均衡状態）に落ち着く．この均衡を定式化したのが英国の交通工学者 J. G. Wardrop の第 1 原則である [30].

UE モデルは以下の 2 つの行動仮定に立脚する：

1. **旅行時間最小化**：すべての利用者は，自己の旅行時間が最小となる経路を選択しようとする．
2. **完全情報**：すべての利用者は，利用可能な全経路の旅行時間に関する完全な情報を持つ．

これらの仮定のもとで，利用者が経路選択を最適化した結果として到達する均衡状態が，以下の原則として定式化される．

Wardrop の第 1 原則（利用者均衡条件）

「利用されているすべての経路の旅行時間は等しく，かつ利用されていない経路の旅行時間以下である．」

均衡状態では，どの利用者も経路を変更することによって自己の旅行時間をそれ以上短縮できない．言い換えると，UE は交通量配分における **Nash 均衡**（Nash equilibrium）に対応する [18, 19]（Nash 均衡との等価性の厳密な議論は第 2.12 節を参照）．

UE 条件の数学的表現

Wardrop の第 1 原則を数式として表すと，任意の OD ペア $rs \in \Omega$ について以下のとおりとなる [3, 25]：

$$f_k^{rs} > 0 \implies c_k^{rs} = u_{rs}, \qquad f_k^{rs} = 0 \implies c_k^{rs} \geq u_{rs} \tag{2.20}$$

ここで u_{rs} は OD ペア rs における均衡最小旅行時間（利用されている経路の共通旅行時間）である．$c_k^{rs} = \sum_a t_a(x_a)\,\delta_{a,k}^{rs}$ は経路 k^{rs} の旅行時間（式 (2.3) 参照）である．

条件 (2.20) を**相補性条件**（相補性条件および KKT 条件の一般形については第 2.13 節を参照）の形式に書き直すと，

$$f_k^{rs} \geq 0, \qquad c_k^{rs} - u_{rs} \geq 0, \qquad f_k^{rs}(c_k^{rs} - u_{rs}) = 0 \qquad \forall\, k \in K_{rs},\ \forall\, rs \in \Omega \tag{2.21}$$

となる．これは，経路 k^{rs} に交通量が流れている（$f_k^{rs} > 0$）ときかつそのときに限り，その経路の費用 c_k^{rs} が最小費用 u_{rs} に等しい（$c_k^{rs} = u_{rs}$）ことを意味する．

2.3.2 等価最適化問題（Beckmann 変換）

直接解法の困難

式 (2.21) の UE 条件を直接満たす解を求めることは，実際のネットワークでは計算上困難である：経路を列挙しながら相補性条件を解くには組み合わせ爆発の問題が伴う．Beckmann, McGuire, and Winsten [3] は，UE 条件と**等価な凸最適化問題**を導出した（Beckmann 変換）．この等価性により，確立された凸最適化アルゴリズムを直接 UE 解法に適用できる．

UE 等価最適化問題（UE/FD-Primal）

UE/FD-Primal（需要固定型利用者均衡の主問題）を次式で定義する：

$$\min_{\boldsymbol{f}} \quad Z(\boldsymbol{f}) = \sum_{a \in A} \int_0^{x_a(\boldsymbol{f})} t_a(w)\, dw \tag{2.22}$$

$$\text{subject to} \quad \sum_{k \in K_{rs}} f_k^{rs} = q_{rs} \qquad \forall\, rs \in \Omega \tag{2.23}$$

$$f_k^{rs} \geq 0 \qquad \forall\, k \in K_{rs},\ \forall\, rs \in \Omega \tag{2.24}$$

ここで $x_a(\boldsymbol{f}) = \sum_{rs} \sum_k f_k^{rs}\, \delta_{a,k}^{rs}$ は式 (2.1) のリンク交通量である．$Z(\boldsymbol{f})$ は **Beckmann ポテンシャル関数**と呼ばれる．

KKT 条件と Wardrop 条件の等価性

問題 (2.22)–(2.24) の KKT（Karush-Kuhn-Tucker）条件が Wardrop 条件 (2.21) に一致することを示す [3].

等式制約 (2.23) に対する Lagrange 乗数を u_{rs} とすると，Lagrangian は次式となる：

$$L(\boldsymbol{f},\, \boldsymbol{u}) = \sum_a \int_0^{x_a} t_a(w)\, dw - \sum_{rs} u_{rs} \left(\sum_k f_k^{rs} - q_{rs} \right) \tag{2.25}$$

連鎖律を用いると，経路フロー f_k^{rs} に関する偏微分は，

$$\frac{\partial Z}{\partial f_k^{rs}} = \sum_{a \in A} t_a(x_a)\, \delta_{a,k}^{rs} = c_k^{rs} \tag{2.26}$$

（リンク a の旅行時間 $t_a(x_a)$ は，経路 k^{rs} がリンク a を通るとき $\delta_{a,k}^{rs} = 1$ で寄与する．）KKT 停留条件 $\partial L / \partial f_k^{rs} = c_k^{rs} - u_{rs}$ に非負制約 $f_k^{rs} \geq 0$ の相補性条件を合わせると，

$$f_k^{rs} \geq 0, \qquad c_k^{rs} - u_{rs} \geq 0, \qquad f_k^{rs}(c_k^{rs} - u_{rs}) = 0 \tag{2.27}$$

これは Wardrop 条件 (2.21) と同一である．すなわち，**問題 (2.22) の最適解は Wardrop の第 1 原則を満たす**，つまり UE 解である．

解の存在と一意性

リンクパフォーマンス関数 $t_a(x_a)$ が $x_a \geq 0$ 上で**単調非減少**（$t'_a(x_a) \geq 0$）であるとき，$Z(\boldsymbol{f})$ は凸であり，最適解（UE 解）が存在することが保証される [3, 31]．さらに，$t_a(x_a)$ が**狭義単調増加**（$t'_a(x_a) > 0$）である場合（BPR 関数の $\beta \geq 1$ のとき：式 (2.4) 参照），$Z(\boldsymbol{f})$ は狭義凸となり，**リンク交通量** $\{x_a\}$ に関する UE 解が**一意**に定まる [25].

- 経路交通量 $\{f_k^{rs}\}$ の一意性は一般に保証されない（複数の経路で旅行時間が同じになれば，その間の配分比率は不定）.
- リンク交通量が一意に定まれば，交通計画実務上は十分である.

2.3.3　Frank-Wolfe 法

(a)　制約つき最適化問題へのアプローチ比較

問題 (2.22) は，フロー保存条件 (2.23) と非負条件 (2.24) という制約をもつ凸最適化問題である．以下では，制約つき最適化問題の代表的な解法を比較し，Frank-Wolfe 法が UE 解法として特に適している理由を述べる [21].

（1）射影法（Projection-based methods）

勾配降下法（$\boldsymbol{x}$ を $-\nabla Z$ 方向に移動）と，更新後の点が制約領域 P 外に出た場合に最近傍点へ引き戻す**射影**（projection）を組み合わせた方法である：

$$\boldsymbol{x}_{t+1} = \Pi_P(\boldsymbol{x}_t - \gamma_t \nabla Z(\boldsymbol{x}_t)), \qquad \Pi_P(\boldsymbol{x}) = \arg\min_{\boldsymbol{y} \in P} \|\boldsymbol{x} - \boldsymbol{y}\| \tag{2.28}$$

直交制約（球面）やボックス制約（各変数の上下界）では射影を解析的に計算できるが，UE のフロー保存制約のような多次元等号制約に対しては，各ステップごとに別の最適化問題を解く必要があり計算コストが高い.

（2）内点法（Interior-point methods）

実行可能領域の境界に近づくと発散する**バリア関数** $B(\boldsymbol{x})$ を目的関数に加え，制約なし最適化問題に変換する：

$$\min_{\boldsymbol{x}} \big\{ Z(\boldsymbol{x}) + \mu\, B(\boldsymbol{x}) \big\} \tag{2.29}$$

精度向上のためにヘッセ行列（2 階微分）の情報が必要であり，また制約を正確にバリア関数で表現することが難しい場合がある.

（3）Frank-Wolfe 法（条件付き勾配法）

Frank and Wolfe が提案した本手法は，制約つき問題をそのままの形で扱い，最適化の全過程を通じて実行可能性を自動的に保証する [10]．中心的なアイデアは，目的関数を現在の解まわりで**線形近似**し，その線形近似が制約領域上で最小となる点（補助解）を求めることである：

$$\boldsymbol{y}_t = \arg\min_{\boldsymbol{y} \in P} \nabla Z(\boldsymbol{x}_t)^{\top} \boldsymbol{y} \tag{2.30}$$

式 (2.30) において $\nabla Z(\boldsymbol{x}_t)$ は現在点で評価した定数ベクトルであり，式 (2.26) より各リンク成分は現在の交通量で評価したリンク旅行時間に等しい：

$$\left[\nabla Z(\boldsymbol{x}_t)\right]_a = t_a\left(x_a^{(t)}\right) \tag{2.31}$$

したがって，式 (2.30) は最適化変数 $\boldsymbol{y}$ に関して**線形な問題**である．凸多面体 P 上の最小解は必ずその端点（頂点）となる．その後，現在の解 $\boldsymbol{x}_t$ と補助解 $\boldsymbol{y}_t$ を線形補間して更新する：

$$\boldsymbol{x}_{t+1} = (1-\gamma_t)\boldsymbol{x}_t + \gamma_t \boldsymbol{y}_t, \qquad 0 \le \gamma_t \le 1 \tag{2.32}$$

実行可能領域 P が凸集合であれば，$\boldsymbol{x}_t,\, \boldsymbol{y}_t \in P$ の凸結合である $\boldsymbol{x}_{t+1}$ も常に P 内に留まる——射影（式 (2.28)）を一切必要としないため，「射影なし法」（projection-free method）とも呼ばれる [21].

UE への対応：$Z(\boldsymbol{f})$ の勾配は $\partial Z/\partial f_k^{rs} = c_k^{rs}$ （式 (2.26)）であるから，式 (2.30) の線形計画は，現在のリンク旅行時間 $\{t_a(x_a^{(n)})\}$ に基づいて各 OD ペアの全需要を最小費用経路に集中させる，まさに **All-or-Nothing 配分**の問題に対応する．All-or-Nothing 配分の結果は凸多面体 P の端点（頂点）であり，常に実行可能解である．

(b) Frank-Wolfe 法のアルゴリズムと収束条件

適用条件

Frank-Wolfe 法が有効に機能するための条件を示す [21]：

- 目的関数 Z が凸かつ微分可能
- 実行可能領域 P が凸集合
- 線形計画 $\min_{\boldsymbol{y}\in P} \nabla Z(\boldsymbol{x})^{\top}\boldsymbol{y}$ が簡単に解ける

UE/FD-Primal(2.22) はこれらをすべて満たす（BPR 関数の凸性より Z が凸，線形計画は All-or-Nothing 配分で効率的に解ける）.

アルゴリズムの手順

アルゴリズムの各ステップを表 2.4 に示す.

図 2.7 に Frank-Wolfe 法の 1 ステップの幾何学的イメージを示す．現在の解 $\boldsymbol{x}^{(n)}$ から，線形近似が最小となる実行可能領域の頂点（補助解 $\boldsymbol{y}^{(n)}$ = All-or-Nothing 解）を求め，線探索で決定したステップサイズ α^* に従って $\boldsymbol{x}^{(n+1)}$ に移動する.

収束判定：相対ギャップ

実用的な収束判定基準として，**相対ギャップ**（Relative Gap）を用いる：

$$\mathrm{Gap}^{(n)} = \frac{\displaystyle\sum_{a\in A} t_a\left(x_a^{(n)}\right)\left(x_a^{(n)} - y_a^{(n)}\right)}{\displaystyle\sum_{a\in A} t_a\left(x_a^{(n)}\right) x_a^{(n)}} \tag{2.33}$$

表 2.4: UE を解く Frank-Wolfe 法のアルゴリズム

Step	内容
1	**初期化**：すべてのリンク交通量を $x_a^{(0)} = 0$ とし，自由流旅行時間 $t_a(0) = t_{a0}$ に基づいて All-or-Nothing 配分を実行し，初期解 $\boldsymbol{x}^{(1)}$ を得る．$n \leftarrow 1$.
2	**旅行時間の更新**：現在の解 $\boldsymbol{x}^{(n)}$ の各リンクで旅行時間 $t_a\left(x_a^{(n)}\right)$ を計算する．
3	**補助解の計算**（Frank-Wolfe ステップ）： 更新された旅行時間のもとで All-or-Nothing 配分を実行し，補助解 $\boldsymbol{y}^{(n)}$ を得る（式 (2.30) の線形計画の解）．
4	**ステップサイズ探索**（線探索）： $\alpha^* = \arg\min_{0 \le \alpha \le 1} Z\big(\boldsymbol{x}^{(n)} + \alpha(\boldsymbol{y}^{(n)} - \boldsymbol{x}^{(n)})\big)$ 黄金分割法などの 1 次元最小化を用いる（第 2.10 節参照）．
5	**解の更新**：$\boldsymbol{x}^{(n+1)} = (1-\alpha^*)\boldsymbol{x}^{(n)} + \alpha^* \boldsymbol{y}^{(n)}$.
6	**収束判定**：相対ギャップ（式 (2.33)）が閾値 ε 以下なら終了．そうでなければ $n \leftarrow n+1$ として Step 2 へ．

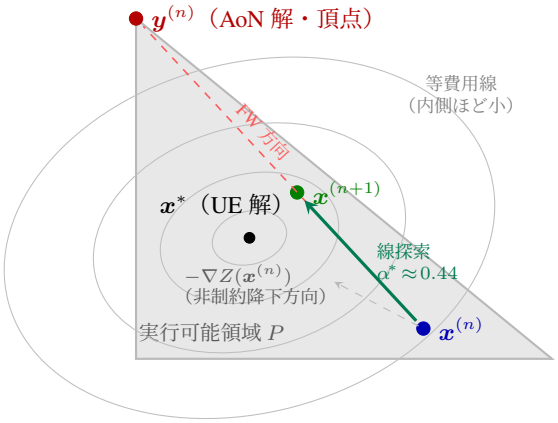


図 2.7: Frank-Wolfe 法の 1 ステップ（概念図）．シャドウ付き多角形が実行可能領域 P（フロー保存条件を満たすフロー集合の模式的表現，2 次元単体）．曲線は目的関数 $Z(\boldsymbol{x})$ の等費用線（内側ほど小）．$\boldsymbol{x}^*$ が UE 最適解，$\boldsymbol{y}^{(n)}$ は現在の旅行時間での All-or-Nothing 解（実行可能領域の頂点），$\boldsymbol{x}^{(n+1)}$ は線探索後の更新点．

分子 $\sum_a t_a(x_a^{(n)})(x_a^{(n)}-y_a^{(n)})$ は Frank-Wolfe ギャップ $\nabla Z(\boldsymbol{x}^{(n)})^\top(\boldsymbol{x}^{(n)}-\boldsymbol{y}^{(n)})$ に等しく（$\nabla_{x_a} Z = t_a(x_a)$），UE 解からの残差を表す．分母は現在の総旅行時間であり，正規化の役割を果たす．$\mathrm{Gap}^{(n)} \geq 0$ であり，$\mathrm{Gap}^{(n)} \to 0$ のとき $\boldsymbol{x}^{(n)}$ は UE 解に収束する．

収束条件の意味（大域最適性の証明）

$\mathrm{Gap}^{(n)} = 0$，すなわち $\nabla Z(\boldsymbol{x}^{(n)})^\top(\boldsymbol{y}^{(n)} - \boldsymbol{x}^{(n)}) \geq 0$ が成立するとき（$\boldsymbol{y}^{(n)}$ が線形近似の最小解であることからこの内積は常に ≤ 0 であり，≥ 0 との組み合わせで $= 0$ と同値である），$\boldsymbol{x}^{(n)}$ が大域最適解（UE 解）であることを示す [10, 21]．実行可能領域内の任意の点 $\boldsymbol{x}$ に対して，

$$
\begin{aligned}
Z(\boldsymbol{x}) &\geq Z(\boldsymbol{x}^{(n)}) + \nabla Z(\boldsymbol{x}^{(n)})^\top(\boldsymbol{x} - \boldsymbol{x}^{(n)}) && \text{（}Z\text{ の凸性）} && (2.34)\\
&\geq Z(\boldsymbol{x}^{(n)}) + \nabla Z(\boldsymbol{x}^{(n)})^\top(\boldsymbol{y}^{(n)} - \boldsymbol{x}^{(n)}) && \text{（}\boldsymbol{y}^{(n)}\text{ が線形近似の最小解）} && (2.35)\\
&\geq Z(\boldsymbol{x}^{(n)}) && \text{（収束条件より）} && (2.36)
\end{aligned}
$$

よって，すべての実行可能解 $\boldsymbol{x}$ について $Z(\boldsymbol{x}) \geq Z(\boldsymbol{x}^{(n)})$ が成立し，$\boldsymbol{x}^{(n)}$ が大域最適，すなわち UE 解であることが確認される．

(c) 特徴と改良手法

長所

- **実装が単純**：各イテレーションは旅行時間計算 → All-or-Nothing 配分 → 線探索 → 更新という明快な手順で構成される．
- **メモリ効率が良い**：経路を陽に列挙・記憶する必要がなく，リンク交通量のみを保持すればよい．
- **実行可能性の自動保証**：更新点は常に実行可能領域内に留まり，射影計算が不要である（前節で証明）．

短所

最大の欠点は**収束の遅さ**である [21]：

- ステップサイズ α^* が収束するにつれて小さくなり，最適解付近で解が「ジグザグ運動」を繰り返す（図 2.7 の $\boldsymbol{x}^{(n)}$ と $\boldsymbol{y}^{(n)}$ を結ぶ方向が交互に振れるイメージ）．
- 1 階の微分（勾配）情報のみを使い，目的関数の曲率（2 階微分）を活用しないため，収束速度は $O(1/t)$ のオーダーに留まる（t はイテレーション数）．

改良手法

この欠点を克服するための改良手法として主に以下が知られている [31, 25]：

打ち切り二次計画法（Truncated Quadratic Programming） 目的関数のテイラー展開を 2 次項まで用い，局所的な二次近似を最小化する．収束速度が改善されるが，ヘッセ行列の計算コストが生じる．

Simplicial Decomposition 法 均衡解を，これまでの反復で得られた All-or-Nothing 解（端点解）の凸結合として表し，その結合係数を最適化する．少数の端点解を記憶するコストはあるが，Frank-Wolfe 法より収束が大幅に加速される．

クイズ 5：ソウルの高架道路を撤去したら，渋滞が改善した——なぜ？

2003 年 7 月にソウル市は清渓川（チョンゲチョン）沿いの高架道路（1 日約 17 万台が走行）の撤去工事を始め，2005 年 9 月に開通しました．交通専門家の多くが「渋滞が悪化する」と予測しましたが，実際には周辺の渋滞がむしろ改善しました．この「道路を減らして渋滞が改善した」現象は，次のどの考え方から説明できるでしょうか？

(ア) 道路が減ると交通需要自体も減るから（誘発交通の逆）

(イ) 道路を追加すると逆に渋滞が悪化することがある Braess のパラドックスの逆——逆 Braess 現象が起きたから

(ウ) 撤去エリアのドライバーが公共交通機関に転換したから

→ 正解・解説は節末を参照

2.3.4　Braess のパラドックス

パラドックスの概要

UE の本質的な性質を示す重要な例を紹介する [5].

Braess のパラドックス（Braess's paradox）

「ネットワークに新しいリンクを追加すると，利用者均衡配分のもとで**すべての**利用者の旅行時間が増加することがある．」

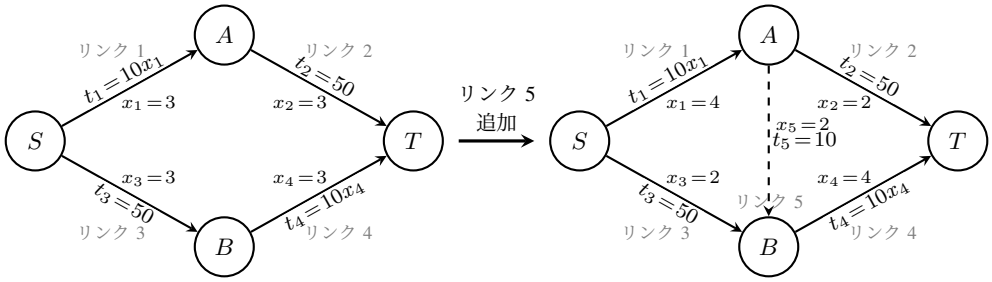
これは，UE では各利用者が自己の旅行時間だけを考慮し，自分がリンクに追加した際の他利用者への混雑増大（外部コスト）を考慮しないために生じる．Nash 均衡としての個人最適化が，社会全体の最適化と食い違うことの端的な例である．

数値例の設定

図 2.8 のネットワークを考える．OD 交通量は $q_{ST} = 6$（台/時），リンクコスト関数は表 2.5 のとおりである．

表 2.5: Braess ネットワークのリンクコスト関数（$q_{ST} = 6$）

リンク番号	向き	コスト関数	種別
1	$S \to A$	$t_1(x_1) = 10x_1$	混雑型
2	$A \to T$	$t_2 = 50$	定数型
3	$S \to B$	$t_3 = 50$	定数型
4	$B \to T$	$t_4(x_4) = 10x_4$	混雑型
5	$A \to B$	$t_5 = 10$	定数型（新設）



(a) リンク 5 追加**前**
UE 旅行時間：$10\times3+50=\mathbf{80}$ 分

(b) リンク 5 追加**後**
UE 旅行時間：$10\times4+50=\mathbf{90}$ 分（悪化）

図 2.8: Braess のパラドックス：リンク 5（$A\to B$，破線）の追加前後の UE 解と均衡旅行時間.

リンク 5 追加前

利用可能経路は，$S\to A\to T$（経路 1）と $S\to B\to T$（経路 2）の 2 本である．Wardrop 条件（利用される 2 経路の費用が等しい）と流量保存則を連立すると，

$$t_1(x_1)+t_2=t_3+t_4(x_4),$$
$$10x_1+50=50+10x_4 \implies x_1=x_4 \tag{2.37}$$
$$x_1+x_3=q_{ST}=6 \tag{2.38}$$

ネットワークの対称性$(t_1=t_4, t_2=t_3)$により $x_1=x_3=x_4=3$, 均衡旅行時間は $10\times3+50=\mathbf{80}$ 分である．

リンク 5 追加後

3 本目の経路 $S\to A\to B\to T$（経路 3）が利用可能となる．Wardrop 条件（3 経路すべての費用が等しい）を立てる：

$$10x_1+50=50+10x_4=10x_1+10+10x_4 \tag{2.39}$$

左辺と右辺の等式から $50=10+10x_4$ より $x_4=4$，中辺と右辺の等式から $50=10+10x_1$ より $x_1=4$.

次に，経路交通量 f_1, f_2, f_3（それぞれ経路 1・2・3）とリンク交通量の関係式

$$x_1=f_1+f_3,\quad x_2=f_1,\quad x_3=f_2,\quad x_4=f_2+f_3,\quad x_5=f_3 \tag{2.40}$$

に $x_1=x_4=4$ と流量保存則 $f_1+f_2+f_3=6$ を代入すると，$f_1=f_2=f_3=2$ が得られ，$(x_1, x_2, x_3, x_4, x_5)=(4, 2, 2, 4, 2)$ となる．

流量保存則の確認（各ノードで流入＝流出）：

- ノード S：$x_1+x_3=4+2=6=q_{ST}$ ✓
- ノード A：$x_1=x_2+x_5 \implies 4=2+2$ ✓

- ノード B：$x_3 + x_5 = x_4 \implies 2 + 2 = 4$ ✓

3 経路すべての旅行時間は，$10 \times 4 + 50 = 50 + 10 \times 4 = 10 \times 4 + 10 + 10 \times 4 = \mathbf{90}$ 分に増加した．結果を表 2.6 にまとめる．

表 2.6: Braess のパラドックス：数値例のまとめ（$q_{ST} = 6$）

状況	リンク交通量 $(x_1, x_2, x_3, x_4, x_5)$	UE 旅行時間
リンク 5 追加前	$(3,\ 3,\ 3,\ 3,\ -)$	80 分
リンク 5 追加後	$(4,\ 2,\ 2,\ 4,\ 2)$	90 分（悪化）

考察

追加後の均衡では，すべての利用者が経路 3（$A \to B$ 経由の迂回路）を部分的に利用し，リンク 1・4 に交通量が集中した結果，旅行時間が 80 分から 90 分へと悪化した．

- **インフラ整備の逆効果**：新設リンクが全利用者に開放されても，均衡における集合的行動の結果として旅行時間が悪化しうる（「新道を作ったのに渋滞が増えた」という実社会の観察と合致）．
- **SO との対比**：社会的に最適な配分で利用者を制御すれば，このパラドックスは生じない（第 2.5 節参照）．UE と SO のギャップが，混雑課金による政策介入（Pigouvian 課金，第 2.5 節・第 2.8 節 [20]）の根拠となる．
- **ネットワーク設計への教訓**：インフラ整備計画においては，個人の経路選択行動を明示的に考慮しなければ，投資が逆効果をもたらしうる．

本節のまとめ

本節で示した主要な結果を整理する：

1. **Wardrop の第 1 原則**（「使用経路はすべて最短で等時間，未使用経路はそれ以上」）が混雑ネットワークの Nash 均衡を定式化し，利用者均衡（UE）の均衡条件を与える；UE は完全情報・旅行時間最小化という 2 仮定のもとで定義される（第 2.3.1 小節）．
2. **Beckmann 変換**により UE 条件は凸最適化問題（UE/FD-Primal）と等価であることが示され，その KKT 条件が Wardrop 均衡条件に一致する；BPR 関数のような狭義単調増加リンク費用のもとでは目的関数が狭義凸となり，UE 解は一意に定まる（第 2.3.2 小節）．
3. **Frank-Wolfe 法**は，補助解問題（AoN 配分）とステップサイズ最適化の反復を通じて相対ギャップが収束するまで UE 解を逐次的に求める実用的アルゴリズムであり，大規模ネットワークへの適用が現実的である（第 2.3.3 小節）．
4. **Braess のパラドックス**は，新規リンク追加後の Nash 均衡（UE）において全利用者の旅行時間が悪化しうることを示す典型的な反例であり，UE と社会最適配分（SO）との乖離が混雑課金介入（第 2.5 節・第 2.8 節）の根拠となることを示唆する（第 2.3.4 小節）．

クイズ 4 正解・解説

正解：（イ）
誰も全体を調整していないのに，利己的行動の集積として均衡が「自然に」生まれます——これが Wardrop 均衡の核心です．（エ）の料金効果は第 7 章の一般化費用で扱われます．

クイズ 5 正解・解説

正解：（イ）（（ウ）も部分的に正しい）
Braess のパラドックス（この節）の数値例では，リンクを 1 本追加すると全員の旅行時間が 80 分から 90 分に悪化します．ソウルの清渓川事例はその逆——撤去によって利用者均衡が改善された「逆 Braess 現象」の実例として交通工学の文献で引用されています．
参考：Wikipedia – Cheonggyecheon

2.4　確率的利用者均衡配分（SUE）

第 2.3 節で扱った利用者均衡配分（UE）は，すべての利用者がネットワーク上の旅行時間情報を完全に把握したうえで，最短経路を確定的に選択すると仮定した．しかし現実には，利用者が得られる旅行時間情報は不完全であり，同じ経路でも個人の経験や好みの違いにより旅行時間の認知は異なる．本節では，この**完全情報仮定**を緩和し，利用者の旅行時間認知に確率的な誤差が存在することを明示的に取り込んだ**確率的利用者均衡配分**（Stochastic User Equilibrium, **SUE**）を扱う．

まず，ランダム効用理論の枠組みから SUE の動機づけを述べ（第 2.4.1 小節），次にロジット型 SUE の等価最適化問題を導出し（第 2.4.2 小節），最後に代表的な解法を比較する（第 2.4.3 小節）．

2.4.1　完全情報仮定の緩和

UE モデルの仮定とその限界

Wardrop の第 1 原則に基づく UE は，「すべての利用者がネットワーク上のすべての経路の旅行時間を正確に知り，最も短い経路を確定的に選択する」という**完全情報仮定**を前提とする（第 2.3 節 第 2.3.1 小節）．この仮定のもとで，均衡状態では使用されているすべての経路の旅行時間が等しくなることが保証される．

しかし，この仮定は現実の利用者行動を単純化しすぎている．実際の利用者が直接知ることのできる旅行時間は，過去に自ら経験した経路に限られており，標識・カーナビ・渋滞情報サービスなどから得られる情報の精度にも個人差がある．さらに，同じ経路での旅行時間は日によっても変動する．こうした情報の不完全性や認知の個人差を無視した場合，UE が予測する配分パターンと現実の交通量観測値が乖離することがある [7, 25]．

ランダム効用理論

SUE は**ランダム効用理論**（Random Utility Theory, RUT）の枠組みに基づく [16]．RUT では，利用者 n が経路 k に対して知覚する効用（負の旅行時間）を

$$U_k^n = -c_k + \varepsilon_k^n \tag{2.41}$$

と表す．ここで c_k は経路 k の（客観的な）旅行時間，ε_k^n は利用者 n の知覚誤差を表すランダム変数である．各利用者は U_k^n を最大化する，すなわち主観的な旅行時間を最小に見込む経路を選択する．

誤差項 ε_k^n に仮定する確率分布により，以下の 2 種類のモデルが導かれる [7, 25].

ロジットモデル（Multinomial Logit, MNL）　各経路の誤差項が独立かつ同一の**ガンベル（Gumbel）分布**に従うと仮定する：$\varepsilon_k \overset{\text{i.i.d.}}{\sim} \mathrm{Gumbel}(0,\, 1/\theta)$．このとき，経路 k^{rs} の選択確率は閉形式で

$$P(k \mid rs) = \frac{\exp(-\theta\, c_k^{rs})}{\displaystyle\sum_{k' \in K_{rs}} \exp(-\theta\, c_{k'}^{rs})} \tag{2.42}$$

と表される [16]．ここで $\theta > 0$ は**感度パラメータ**（精度パラメータ）であり，$\theta \to \infty$ のとき選択確率は最短経路に集中し（UE への収束），$\theta \to 0$ のとき全経路が均等に選ばれるランダム配分に近づく．閉形式で選択確率が得られるため数理的に扱いやすい．一方，**IIA 特性**（Independence of Irrelevant Alternatives：2 経路間の選択確率の比が第三の経路の有無に依存しない）を持つため，経路間に区間の重複がある場合に非現実的な挙動を示すことがある．

プロビットモデル（Probit）　各経路の誤差項が多変量**正規分布**に従うと仮定する：$\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0},\, \boldsymbol{\Sigma})$．分散共分散行列 $\boldsymbol{\Sigma}$ を通じて経路間の相関を表現できるため，経路重複に起因する IIA 問題を原理的に回避できる．ただし選択確率の閉形式は一般に存在せず，モンテカルロ法（シミュレーション）による数値積分が必要となる [7].

以降の定式化ではロジットモデルを採用する（プロビット型 SUE の解法については 第 2.4.3 小節 で簡単に触れる）.

SUE の不動点条件

ロジットモデルを用いた場合，OD ペア rs の経路 k への期待交通量は

$$E[f_k^{rs}] = q_{rs}\, P(k \mid rs) = q_{rs}\, \frac{\exp\bigl(-\theta\, c_k^{rs}(\boldsymbol{f})\bigr)}{\displaystyle\sum_{k' \in K_{rs}} \exp\bigl(-\theta\, c_{k'}^{rs}(\boldsymbol{f})\bigr)} \tag{2.43}$$

となる．ここで経路費用 $c_k^{rs}(\boldsymbol{f})$ が経路交通量 $\boldsymbol{f}$ に依存する点が，フロー非依存の確率的配分（第 2.2 節 第 2.2.3 小節）と根本的に異なる．

配分の**自己整合性**（均衡条件）を要求すると，経路交通量は次の**不動点条件**（fixed-point condition）を満たさなければならない [7]：

$$\boldsymbol{f}^* = \boldsymbol{\Lambda}(\boldsymbol{c}(\boldsymbol{f}^*)) \tag{2.44}$$

ここで，$\boldsymbol{\Lambda}(\cdot)$ は経路費用ベクトル $\boldsymbol{c}$ を与えたときのロジット選択確率に基づく配分写像である．この不動点として定まる均衡が**確率的利用者均衡**（SUE）である．Daganzo & Sheffi [7] は，Brouwer の不動点定理を用いて SUE 解の存在を保証している．

2.4.2　ロジット型 SUE の等価最適化問題

Fisk の等価最適化問題

不動点条件 (2.44) を直接解くことは一般に困難である．Fisk [9] は，ロジット型 SUE の解が以下の**等価最適化問題**（SUE/FD-path）の最適解と一致することを示した：

$$\min_{\boldsymbol{f}\in\mathcal{F}} Z(\boldsymbol{f}) = \underbrace{\sum_{a\in A}\int_0^{x_a} t_a(w)\,dw}_{\text{ポテンシャル項（UE と同一）}} + \underbrace{\frac{1}{\theta}\sum_{rs\in\Omega}\sum_{k\in K_{rs}} f_k^{rs}\ln f_k^{rs}}_{\text{エントロピー項}} \tag{2.45}$$

ここで $\mathcal{F}$ は式 (2.2) の流量保存条件と非負条件を満たす実行可能経路交通量の集合, $x_a = \sum_{rs}\sum_k f_k^{rs}\delta_{a,k}^{rs}$（式 (2.1)）である．

エントロピー項の解釈

Shannon エントロピー

$$H_{rs}(\boldsymbol{f}^{rs}) = -\sum_{k\in K_{rs}} \frac{f_k^{rs}}{q_{rs}}\ln\frac{f_k^{rs}}{q_{rs}} \geq 0 \tag{2.46}$$

は経路交通量の多様性（分散度）を表す量であり，すべての経路が均等に選ばれるときに最大となる．これを用いると，エントロピー項は定数差を除いて $-\frac{1}{\theta}\sum_{rs} q_{rs}H_{rs}(\boldsymbol{f}^{rs})$ と書き直せる（定数項は最適解を変えない）．

目的関数 (2.45) は，UE の Beckmann 変換（ポテンシャル項）に加えてエントロピー項を最小化する問題である．ポテンシャル項は混雑コストの低い配分を促し，エントロピー項は経路交通量の均等な分散を促す方向に働く．この 2 項のバランスが感度パラメータ θ によって調整される．

KKT 条件によるロジット確率の導出

ポテンシャル項とエントロピー項はともに凸であるため，目的関数 (2.45) は全体として**狭義凸**であり，流量保存制約は線形であることから，最適解は唯一存在する [9].

流量保存制約 $\sum_k f_k^{rs} = q_{rs}$ に対する Lagrange 乗数を μ_{rs} とすると，$f_k^{rs} > 0$ の場合の KKT 条件は

$$\frac{\partial Z}{\partial f_k^{rs}} = c_k^{rs}(\boldsymbol{f}) + \frac{1}{\theta}\bigl(1+\ln f_k^{rs}\bigr) - \mu_{rs} = 0 \tag{2.47}$$

となる．これを f_k^{rs} について解くと

$$\ln f_k^{rs} = \theta\mu_{rs} - 1 - \theta\, c_k^{rs} \quad\Longrightarrow\quad f_k^{rs} \propto \exp\bigl(-\theta\, c_k^{rs}\bigr)$$

であり，流量保存条件 $\sum_{k'} f_{k'}^{rs} = q_{rs}$ を用いて正規化すると，

$$f_k^{rs} = q_{rs}\,\frac{\exp\bigl(-\theta\, c_k^{rs}\bigr)}{\displaystyle\sum_{k'\in K_{rs}}\exp\bigl(-\theta\, c_{k'}^{rs}\bigr)} \tag{2.48}$$

が得られる．これは式 (2.43) のロジット選択確率に基づく期待交通量に完全に一致し，最適化問題 (2.45) の最適解が SUE 条件を満たすことが示される．

パラメータ θ の役割と極限ケース

感度パラメータ θ は，UE とランダム配分を連続的につなぐ橋渡しの役割を果たす．

$\theta \to \infty$（**UE への収束**） エントロピー項の係数 $1/\theta \to 0$ となり，目的関数 (2.45) は Beckmann 変換と等しくなる．KKT 条件 (2.47) は Wardrop の UE 条件に収束し，使用される経路は最短経路のみとなる（第 2.3 節 第 2.3.1 小節参照）．

$\theta \to 0$（**ランダム配分への収束**） エントロピー最大化が支配的になり，すべての経路が一定割合ずつ OD 交通量を担うランダム配分に近づく．

有限の θ（現実的な状況） 利用者は概ね短い経路を選ぶ傾向があるが，ランダムな誤差の影響で最短でない経路にも正の交通量が流れる．パラメータ θ の推定は，観測された経路選択データや SP 調査（Stated Preference）から行われる [25].

2.4.3　SUE の解法

SUE 解の計算アルゴリズムは，不動点条件 (2.44) を反復的に解く手法と，等価最適化問題 (2.45) を直接最小化する手法に大別される（表 2.7）．

表 2.7: ロジット型 SUE の代表的な解法の比較

解法	補助問題	ステップサイズ	収束速度
逐次平均法（MSA）	ロジット配分	$1/n$（固定漸減）	遅い
部分線形化法	ロジット配分	ラインサーチ	中程度
Simplicial Decomposition	ロジット配分	制限凸最適化	速い

逐次平均法（Method of Successive Averages, MSA）

逐次平均法は SUE の最も単純な解法であり [7]，UE の Frank-Wolfe 法（第 2.3 節 第 2.3.3 小節）からラインサーチを除いた構造に対応する．

表 2.8: 逐次平均法（MSA）のアルゴリズム

Step	内容
0	**初期化**：任意の実行可能解 $\boldsymbol{f}^{(0)} \in \mathcal{F}$ を設定する（例：$\theta = 0$ のランダム配分）．反復回数 $n = 1$.
1	**補助解の計算**：現在のリンク旅行時間 $t_a(x_a^{(n-1)})$ を固定し，ロジット配分（フロー非依存）を実施して補助解 $\hat{\boldsymbol{f}}$ を求める：$\hat{f}_k^{rs} = q_{rs}\,\dfrac{\exp\!\big(-\theta\, c_k^{rs}(\boldsymbol{f}^{(n-1)})\big)}{\sum_{k'} \exp\!\big(-\theta\, c_{k'}^{rs}(\boldsymbol{f}^{(n-1)})\big)}$
2	**更新**：ステップサイズ $1/n$ を用いた荷重平均で経路交通量を更新する：$\boldsymbol{f}^{(n)} = \boldsymbol{f}^{(n-1)} + \dfrac{1}{n}\left[\hat{\boldsymbol{f}} - \boldsymbol{f}^{(n-1)}\right]$
3	**収束判定**：$\lVert \boldsymbol{f}^{(n)} - \boldsymbol{f}^{(n-1)} \rVert \,/\, \lVert \boldsymbol{f}^{(n-1)} \rVert \le \epsilon$ であれば収束として終了；そうでなければ $n \leftarrow n+1$ として Step 1 へ戻る．

MSA はステップサイズの探索を必要とせず実装が容易である．一方，$1/n \to 0$ のペースで縮小するステップサイズは解の近傍で十分に小さくならないため収束が遅く，大規模ネットワークでは実用上の問題となる [7].

部分線形化法（Partial Linearization）

Fisk [9] が提案した部分線形化法では，目的関数 (2.45) の**ポテンシャル項のみ**を現在点 $\boldsymbol{f}^{(n-1)}$ のまわりで線形近似し，エントロピー項はそのまま保持した補助問題を解く．

反復 n における補助問題は

$$\min_{\boldsymbol{f}\in\mathcal{F}} \sum_{a\in A} t_a\big(x_a^{(n-1)}\big)\, x_a + \frac{1}{\theta}\sum_{rs}\sum_{k} f_k^{rs}\ln f_k^{rs}$$

の形をとる（$x_a^{(n-1)} = \sum_{rs}\sum_k f_k^{rs,(n-1)}\delta_{a,k}^{rs}$）．これは現在のリンク旅行時間 $t_a(x_a^{(n-1)})$ を固定コストとみなしたフロー非依存のロジット配分問題に帰着するため，Dial のアルゴリズム（第 2.2 節第 2.2.4 小節）などで効率的に解くことができる [9].

得られた補助解 $\hat{\boldsymbol{f}}$ を用いてラインサーチでステップサイズ α^* を決定し，

$$\boldsymbol{f}^{(n)} = (1-\alpha^*)\,\boldsymbol{f}^{(n-1)} + \alpha^*\,\hat{\boldsymbol{f}}$$

と更新する．最適ステップサイズの採用により，MSA より目的関数値を効率的に減少させることができる [9, 25].

Simplicial Decomposition 法

Simplicial Decomposition 法では，均衡解 $\boldsymbol{f}^*$ を過去のロジット補助解（**極点解**）の凸結合として表現し，その結合重みを反復的に最適化する [25].

各反復は次の 2 段階で構成される（表 2.9）.

表 2.9: Simplicial Decomposition 法の各反復における手順

Step	内容
1	**外側反復**（極点の追加）：現在のリンク旅行時間に基づくロジット補助解 $\hat{\boldsymbol{f}}^{(n)}$ を新たな極点として保持集合 $\mathcal{S}^{(n)}$ に追加する．
2	**内側反復**（制限最短化）：$\mathcal{S}^{(n)}$ に含まれる極点解の凸包上で目的関数 (2.45) を最小化する制限凸最適化問題を解き，結合重みを更新する．

内側反復は低次元の凸最適化問題であるため効率的に解ける．この方法の長所は，少数の極点解で均衡解をよく近似できる点と，各外側反復後に目的関数値が単調減少することが保証される点にある [25].

プロビット型 SUE の解法

確率論的な積分の閉形式を持たないプロビット型 SUE では，次の手順による**モンテカルロ法**が標準的に用いられる [7]：各反復において，(i) 誤差項 $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ を多数回サンプリングし，(ii) 各サンプルに対して AoN 配分を実施し，(iii) 得られた配分をサンプル平均してロジット配分の代わりに使用する．その後は MSA や部分線形化法と同様の更新ステップを適用する．計算コストは

ロジット型より大幅に増大するが，経路重複問題への対応が求められるネットワークで有効である [7, 25].

クイズ 6：ナビが全員に「今すぐ迂回せよ」と告げたら，翌日はどうなる？

2019 年 10 月，台風 19 号接近時に首都圏の高速道路が相次いで通行止めになり，複数のルート案内アプリが一斉に特定の一般道を代替ルートとして案内しました．その数分後，案内された道路は大渋滞になりました．翌日の同じ時間帯，今度はその道路はどうなっていたでしょうか？

(ア) 引き続き混雑している——一度定着したルートは変わらない

(イ) 今度は空いている——昨日の渋滞を見て全員が別のルートへ移動する．しかしそちらが混雑し，翌々日はまた戻ってくる……

(ウ) 完全に均等分散している——完全情報があれば瞬時に均衡に達する

→ 正解・解説は節末を参照

本節のまとめ

本節で示した主要な結果を整理する：

1. **SUE** はランダム効用理論（知覚旅行時間 = 客観的旅行時間 + 誤差項）の枠組みで完全情報仮定を緩和した配分モデルであり，誤差分布としてガンベル分布を仮定するとロジット型（MNL），正規分布を仮定するとプロビット型が導かれる（第 2.4.1 小節）．
2. ロジット型 SUE の均衡条件は **Fisk の等価最適化問題**（Beckmann 目的関数からエントロピー項 $-\theta^{-1}\sum_k f_k \log f_k$ を引いた凸問題）として定式化でき，その KKT 条件からロジット選択確率が直接導出される（第 2.4.2 小節）．
3. SUE の実用的な解法として，**逐次平均法**（MSA），**部分線形化法**，および **Simplicial Decomposition 法**の 3 手法があり，それぞれ実装容易性・収束速度・大規模化の観点でトレードオフを持つ；プロビット型 SUE にはモンテカルロ法が標準的に用いられる（第 2.4.3 小節）．

クイズ 6 正解・解説

正解：(イ)

「全員が同じ完全情報で一斉に動く」とき，日々振動が続く不安定なダイナミクスが生まれます．SUE が記述する「不完全情報による確率的分散」は，この振動を自然に緩和する機能を持ちます——完全情報化が必ずしも望ましくないというパラドックスです．このダイナミクスは第 6 章で理論化されます．

2.5　システム最適配分（SO）と混雑課金

第 2.3 節で扱った利用者均衡配分（UE）では，各利用者が自己の旅行時間を最小化するように経路選択を行う結果として均衡が成立することを示した（第 2.3.2 小節）．しかし，第 2.3 節の Braess のパラドックス（第 2.3.4 小節）が示すように，個人の合理的行動の集積が必ずしも社会全体にとって望ましい結果をもたらすとは限らない．個々の利用者は道路に参入するとき，自分の旅行時間しか考慮しないが，実際には他の利用者の旅行時間も増大させている．これが**混雑の外部性**である．

本節では，この外部性を明示的に扱い，ネットワーク全体の**総旅行時間を最小化する**配分として定義される**システム最適配分**（System Optimal, SO）を導入する（第 2.5.1 小節）．続いて UE と SO の相違を比較し（第 2.5.2 小節），混雑課金（Pigouvian Tax）によって UE を SO と一致させる**限界費用課金**の考え方を説明する（第 2.5.3 小節）．最後に，余剰分析の手法を用いて，UE における死荷重損失と混雑課金の効果を定量的に整理する（第 2.5.4 小節）[20, 3].

本節で新たに用いる主要記号を表 2.10 にまとめる（基本記号は表 2.1 参照）．

表 2.10: システム最適配分・混雑課金の主要記号

記号	意味	単位（例）
$Z_{SO}(\boldsymbol{x})$	SO の目的関数（ネットワーク全体の総旅行時間）	台・分
PMC_a	私的限界費用 $= t_a(x_a)$（ドライバー自身の旅行時間）	分
SMC_a	社会的限界費用 $= t_a(x_a) + x_a\, t_a'(x_a)$	分
MEC_a	限界外部費用 $= x_a\, t_a'(x_a) = \mathrm{SMC}_a - \mathrm{PMC}_a$	分
τ_a^*	最適混雑課金額（ピグー税；$\tau_a^* = x_a^{SO}\, t_a'(x_a^{SO})$）	分
$D(x_a)$	需要曲線（交通量 x_a 台目の移動への支払意思額）	分
$W(x_a)$	社会的余剰（総便益 − 総社会的費用）	台・分
ΔW	死荷重損失（UE 配分による社会的余剰の損失）	台・分

2.5.1　Wardrop の第 2 原則と定式化

利用者均衡配分の根拠となる Wardrop の第 1 原則に対応して，**Wardrop の第 2 原則**（Wardrop's Second Principle）は次のように述べる [30].

「ネットワーク全体の総旅行時間（旅行費用）が最小となるように交通量が配分される．」

この原則を実現する配分が**システム最適配分**（SO）である．SO は，中央調整者（交通管理機関など）がすべての利用者の経路を社会的観点から最適に割り当てることができる場合の理想的な配分に対応する．

SO-Primal 問題の定式化

SO は，制約条件を UE/FD-Primal（第 2.3.2 小節参照）と共有しつつ，**目的関数だけが異なる**最適化問題として定式化される [3].

SO-Primal

$$\min_{\boldsymbol{x},\,\boldsymbol{f}} \; Z_{SO}(\boldsymbol{x}) = \sum_{a\in A} x_a\, t_a(x_a) \tag{2.49}$$

$$\begin{aligned}
\text{s.t.}\quad & x_a = \sum_{rs\in\Omega}\sum_{k\in K_{rs}} f_k^{rs}\,\delta_{a,k}^{rs} \qquad \forall a \in A \\
& \sum_{k\in K_{rs}} f_k^{rs} = q_{rs} \qquad \forall rs \in \Omega \\
& f_k^{rs} \geq 0 \qquad \forall k \in K_{rs},\ \forall rs \in \Omega
\end{aligned}$$

目的関数 $Z_{SO}(\boldsymbol{x}) = \sum_a x_a t_a(x_a)$ はネットワーク全体の**総旅行時間**（各リンクの旅行時間 t_a と交通量 x_a の積の和）を表す．リンクパフォーマンス関数 $t_a(x_a)$ が単調非減少かつ凸であれば Z_{SO} も凸関数となり，大域最適解が一意に定まる．

Frank-Wolfe 法の適用

SO を解く方法は UE の場合（第 2.3.3 小節参照）と同様に Frank-Wolfe 法をそのまま適用できる [10]．ただし，線形化ステップ（All-or-Nothing 配分）で用いるリンクコストが異なる点に注意する．

UE-Frank-Wolfe では各リンクの旅行時間 $t_a(x_a^{(n)})$（私的限界費用）を最短経路探索のコストとして用いる [31]．一方，SO-Frank-Wolfe では目的関数の勾配，すなわち

$$\frac{\partial Z_{SO}}{\partial x_a} = \frac{d}{dx_a}\big[x_a\, t_a(x_a)\big] = t_a(x_a) + x_a\, t_a'(x_a) \tag{2.50}$$

を各リンクのコストとして用いる．式 (2.50) の右辺は後述の**社会的限界費用**（SMC）に一致しており，SO の KKT 条件は「全使用経路の社会的限界費用の和が等しい」という Wardrop の第 2 原則に対応する．

アルゴリズムの手順

アルゴリズムの各ステップを表 2.11 に示す．UE の場合（表 2.4）と Step 2 の費用計算のみが異なる．

2.5.2　UE と SO の比較

UE と SO の本質的な違いを一言で表せば，「個人の最適化」対「全体の最適化」である．

UE では各利用者が**私的限界費用**（PMC），すなわち自分が経験する旅行時間 $t_a(x_a)$，のみを考慮して経路を選択するため，自分が道路に加わることで他の利用者に与える追加的な旅行時間遅延（外部コスト）は無視される．この外部性の無視が，UE 配分を社会的に非効率にする根本原因である．

SO では，すべての利用者の旅行時間を考慮することで，外部コストを含む**社会的限界費用**（SMC）が等化される均衡が実現し，ネットワークが全体として最も効率的に利用される．

一般に $x_{SO} \leq x_{UE}$ および $t_{SO} \leq t_{UE}$ が成り立つ（需要が価格に反応する場合は厳密不等号）．UE では各利用者が外部コストを負担しないまま道路を利用するため，社会的に最適な水準 x_{SO} を超えた「過剰な交通量」が発生する．

表 2.11: SO を解く Frank-Wolfe 法のアルゴリズム

Step	内容
1	**初期化**：すべてのリンク交通量を $x_a^{(0)} = 0$ とし，自由流社会的限界費用 $\mathrm{SMC}_a(0) = t_{a0}$ に基づいて All-or-Nothing 配分を実行し，初期解 $\boldsymbol{x}^{(1)}$ を得る．$n \leftarrow 1$.
2	**社会的限界費用の更新**：現在の解 $\boldsymbol{x}^{(n)}$ の各リンクで社会的限界費用 $\mathrm{SMC}_a\left(x_a^{(n)}\right) = t_a\left(x_a^{(n)}\right) + x_a^{(n)}\, t_a'\left(x_a^{(n)}\right)$ を計算する（式 (2.50)）.
3	**補助解の計算**（Frank-Wolfe ステップ）： 更新された社会的限界費用のもとで All-or-Nothing 配分を実行し，補助解 $\boldsymbol{y}^{(n)}$ を得る.
4	**ステップサイズ探索**（線探索）： $\alpha^* = \arg\min_{0\le\alpha\le 1} Z_{SO}\big(\boldsymbol{x}^{(n)} + \alpha\big(\boldsymbol{y}^{(n)} - \boldsymbol{x}^{(n)}\big)\big)$ 黄金分割法などの 1 次元最小化を用いる.
5	**解の更新**：$\boldsymbol{x}^{(n+1)} = (1-\alpha^*)\boldsymbol{x}^{(n)} + \alpha^* \boldsymbol{y}^{(n)}$.
6	**収束判定**：相対ギャップ（式 (2.33)）が閾値 ε 以下なら終了．そうでなければ $n \leftarrow n+1$ として Step 2 へ.

表 2.12: 利用者均衡配分（UE）とシステム最適配分（SO）の比較

比較項目	UE	SO
原則	Wardrop の第 1 原則	Wardrop の第 2 原則
目的	個人の旅行時間最小化	社会全体の総旅行時間最小化
目的関数	$\displaystyle\sum_a \int_0^{x_a} t_a(w)\, dw$	$\displaystyle\sum_a x_a\, t_a(x_a)$
最短経路コスト	$t_a(x_a)$（PMC）	$t_a(x_a) + x_a t_a'(x_a)$（SMC）
均衡交通量	x_{UE}（大）	x_{SO}（小）
均衡旅行時間	$t_{UE} = t_a(x_{UE})$（長）	$t_{SO} = t_a(x_{SO})$（短）
外部コスト	無視	内部化

Braess のパラドックスとの関係

第 2.3 節（第 2.3.4 小節）で示した Braess のパラドックスは，UE の非効率性の顕著な例でもある．道路を追加したとき，UE 配分では総旅行時間がかえって悪化するにもかかわらず，SO 配分においてこのような逆説は生じない：最適化が社会全体の視点から行われるため，新設リンクが実際に有益な場合にのみそれが積極的に利用される [5].

クイズ 7：日本の渋滞損失――年間いくら？

国土交通省の推計（H27 年道路交通センサスベース）によると，日本全国の渋滞による時間損失は年間約 **38 億人時間**，経済損失換算で年間約 **12 兆円**に相当します．UE 配分の総旅行費用が SO 配分より大きい理由として最も正確なものはどれでしょうか？

(ア) UE では交通量が多すぎるので，単純に $x \times t$ が大きくなる

(イ) UE では各ドライバーが他者への影響（限界外部費用 MEC $= x_a t'_a(x_a)$）を無視して参入するため，社会的に過剰な交通量が発生する

(ウ) SO はコンピュータが最適化するので，当然 UE より良い

→ 正解・解説は節末を参照

2.5.3　混雑課金（限界費用課金）

UE を SO に一致させるための政策的手段が**混雑課金**（Congestion Toll），あるいは**限界費用課金**（Marginal Cost Pricing）である．この考えは Pigou（1920）に遡り，外部性を価格メカニズムを通じて内部化する**ピグー税**（Pigouvian Tax）の交通版に相当する [20].

社会的限界費用の分解

1 台の車両が混雑したリンク a に追加されるとき，ネットワーク全体（リンク a のみを考える単純な場合）に与えるコスト変化は次のように分解される：

$$\begin{aligned} \mathrm{SMC}_a(x_a) &= \frac{d}{dx_a}\left[x_a\, t_a(x_a)\right] = t_a(x_a) + x_a\,\frac{d\,t_a(x_a)}{dx_a} \\ &= \underbrace{t_a(x_a)}_{\text{私的限界費用（PMC）}} + \underbrace{x_a\, t'_a(x_a)}_{\text{外部コスト}} \end{aligned} \tag{2.51}$$

第 1 項の私的限界費用（PMC）は，利用者自身が経験する旅行時間である．第 2 項の外部コストは，1 台が追加されることで既存の x_a 台すべての旅行時間が $t'_a(x_a)$ だけ増加することによる社会的損失を表す．

UE では利用者が外部コスト $x_a t'_a(x_a)$ を無視して意思決定するため，個人均衡（UE）において過剰な交通量が発生する．

最適課金額の導出

各リンク a に課金額 τ_a を設定すると，利用者が直面するコストは $t_a(x_a) + \tau_a$ となる．このとき，利用者の私的最適化が社会的最適解 $\boldsymbol{x}^{SO}$ を実現するためには，SO 解における社会的限界費用

を課金後の私的限界費用と一致させればよい [3]：

$$\tau_a^* = x_a^{SO}\, t_a'\!\left(x_a^{SO}\right) \tag{2.52}$$

この最適料金 τ_a^* を各リンクに設定したとき，修正後の私的費用に基づく UE 解は SO 解と一致する．すなわち，$x_{UE}^{\text{toll}} = x_{SO}$ が成立する．

BPR 関数への適用

BPR 関数 $t_a(x_a) = t_{a0}\left\{1 + \alpha(x_a/C_a)^{\beta}\right\}$（第 2.1.3 小節）に対しては：

$$t_a'(x_a) = t_{a0}\,\alpha\,\beta\,\frac{x_a^{\beta-1}}{C_a^{\beta}} \tag{2.53}$$

したがって最適課金額は：

$$\tau_a^* = x_a^{SO}\cdot t_{a0}\,\alpha\,\beta\,\frac{(x_a^{SO})^{\beta-1}}{C_a^{\beta}} = t_{a0}\,\alpha\,\beta\,\frac{(x_a^{SO})^{\beta}}{C_a^{\beta}} \tag{2.54}$$

となる．混雑が激しく x_a^{SO}/C_a が大きいリンクほど高い料金が設定されることがわかる．

クイズ 8：首都高の「深夜割引」は混雑課金の考え方か？

首都高速道路は 2022 年 4 月から，深夜 0〜4 時に入口を通過する ETC 車を対象に通行料金を一律 **20% 割引**する深夜割引を導入しました．余剰分析の数値例（$\mathrm{PMC}(2) = 2$, $\mathrm{SMC}(2) = 3$）から最適課金額 τ^* はいくらで，深夜割引の方向性は理論と同じでしょうか？

(ア) $\tau^* = 0.5$，深夜割引は理論と逆方向（値下げなので）

(イ) $\tau^* = 1$，深夜割引は理論と同方向（需要の少ない時間帯を安くして，ピーク需要を平準化する）

(ウ) $\tau^* = 2$，深夜割引は理論とは無関係

→ 正解・解説は節末を参照

2.5.4　余剰分析

混雑課金の効果を定量的に評価するために，ミクロ経済学の**余剰分析**（Welfare Analysis）の枠組みを用いる [31]．ここでは，交通量の変化に伴う社会的厚生の変化を，需要曲線・PMC 曲線・SMC 曲線の関係として図示する（図 2.9）．

社会的余剰の定義と最大化

需要曲線 $D(x_a)$ は，交通量 x_a 台目の移動がもたらす社会的便益（**支払意思額**）を表す．社会的余剰は，総便益から総社会的費用を差し引いた量として定義される：

$$W(x_a) = \int_0^{x_a} D(v)\,dv - \int_0^{x_a} \mathrm{SMC}(v)\,dv = \int_0^{x_a} \left[D(v) - \mathrm{SMC}(v)\right] dv \tag{2.55}$$

この W を最大化する条件は：

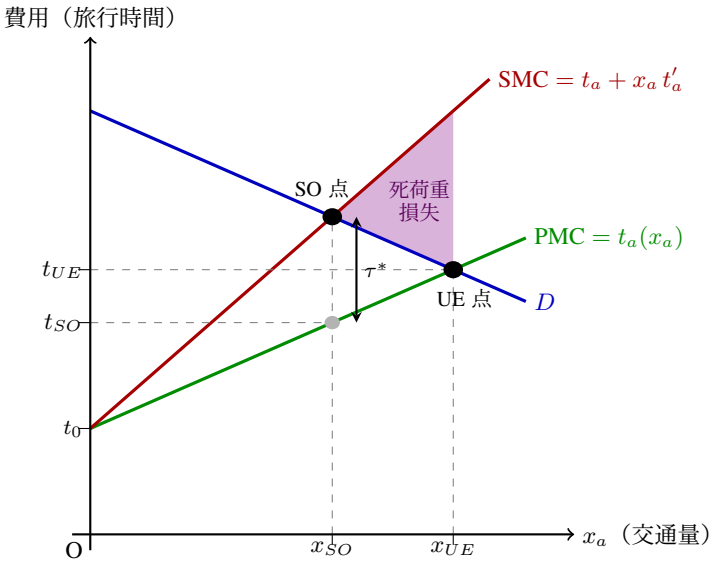


図 2.9: 余剰分析：UE と SO の比較．横軸は交通量 x_a，縦軸は費用（旅行時間・支払意思額）．D は需要曲線（支払意思額），PMC は私的限界費用（旅行時間 $t_a(x_a)$），SMC は社会的限界費用 $\frac{d(x_a t_a)}{dx_a}$．SO 点（x_{SO}）では $D = \text{SMC}$；UE 点（x_{UE}）では $D = \text{PMC}$．紫の三角形（死荷重損失）は UE 配分によって生じる社会的余剰の損失を表す．τ^* は最適混雑課金額（外部コスト）．

$$\frac{dW}{dx_a} = D(x_a) - \text{SMC}(x_a) = 0 \quad \Longleftrightarrow \quad D(x_a) = \text{SMC}(x_a) \tag{2.56}$$

すなわち，需要曲線と SMC 曲線が交わる点が社会的余剰を最大化する配分であり，これが SO 配分 x_{SO} にほかならない（図 2.9 の「SO 点」）．

UE 配分における死荷重損失

UE では各利用者が PMC のみを参照するため，$D(x_a) = \text{PMC}(x_a)$ となる点 x_{UE} で均衡する．$x_{SO} < x_{UE}$ であるため，交通量は社会的最適水準を超えて過剰に供給される．

この過剰供給によって生じる社会的余剰の損失（**死荷重損失**，Deadweight Loss）は：

$$\Delta W = \int_{x_{SO}}^{x_{UE}} \big[\text{SMC}(x) - D(x)\big]\, dx \quad > 0 \tag{2.57}$$

すなわち，x_{SO} から x_{UE} の範囲で SMC が D を上回る部分の面積（図 2.9 の紫の三角形）に等しい．

混雑課金による死荷重損失の解消

最適課金 $\tau_a^* = x_a^{SO}\, t_a'(x_a^{SO})$（式 (2.52)）を各リンクに設定すると，利用者が直面する一般化費用が PMC から SMC へシフトし，UE 点が x_{UE} から x_{SO} に移動する．このとき：

- 死荷重損失 ΔW（紫の三角形）が解消される．
- 課金収入 $\tau_a^* \times x_a^{SO}$ は政府が徴収する．この収入を利用者に一括再配分（lump-sum transfer）することで，パレート改善（全員の厚生が改善）が原理的に可能である．

- 実際の旅行時間 $t_{SO} = t_a(x_a^{SO})$ は，課金前の UE における旅行時間 $t_{UE} = t_a(x_a^{UE})$ より短くなる（$x_{SO} < x_{UE}$ かつ t_a 単調増加より）．

本節のまとめ

本節で示した主要な結果を整理する：

1. **システム最適配分**（SO）は Wardrop の第 2 原則に基づき，社会的限界費用 $\mathrm{SMC}_a = t_a + x_a t'_a$ を経路費用として扱った均衡として定式化され，ネットワーク全体の総旅行時間を最小化する中央集権的配分に対応する（第 2.5.1 小節）．
2. UE と SO を比較すると，UE では私的限界費用（PMC）のみを参照するため過剰交通量と死荷重損失が生じる一方，SO は社会的余剰を最大化する；また Braess のパラドックスは UE では生じるが SO では生じない（第 2.5.2 小節）．
3. **限界費用課金**（混雑課金）$\tau_a^* = x_a^{SO} t'_a(x_a^{SO})$ を各リンクに設定することで，利用者が直面する費用が PMC から SMC へシフトし，UE 解を SO 解に一致させて外部費用を内部化できる（第 2.5.3 小節）．
4. 余剰分析により，UE における死荷重損失（紫の三角形）が混雑課金によって回収される社会的便益に対応することが示され，課金収入の一括再配分によるパレート改善が原理的に可能である（第 2.5.4 小節）．

クイズ 7 正解・解説

正解：(イ)

（ア）は事実ですが「なぜ過剰になるのか」を答えていません．ドライバーは自分の旅行時間（PMC）だけを見て行動し，参入によって他全員に上乗せされる MEC を無視します——この「見えない費用の無視」が年間 38 億人時間・12 兆円の損失の根本原因です．

参考：国土交通省 道路関係データ（渋滞損失時間）；H27 全国道路交通情勢調査プレスリリース（PDF）

クイズ 8 正解・解説

正解：(イ)

$\tau^* = \mathrm{SMC} - \mathrm{PMC} = 3 - 2 = 1$．混雑課金の本義は「ピーク帯に上乗せする」ことですが，「混雑しない時間帯を安くしてピークを分散させる」も同じ効果をねらいます．深夜割引は政治的受容性を優先した「次善の一手」と解釈できます．

参考：首都高速 深夜割引；首都高プレスリリース（2021 年 12 月）

コラム：Braess のパラドックスの検討 ～松山市の PT 調査をもとに～ – 冨矢航大

Braess のパラドックス

都市の交通渋滞を解消するためには、新道の建設や車線拡幅が不可欠であると考えられがちです。しかし、利用者が自身の利益を最大化しようと経路を選択する「利用者均衡配分」の理論下では、特定の道が存在することで、かえって全体の効率が下がる「Braess のパラドックス」が生じることがあります。ミニ研究として、松山市中心部のネットワークデータと、2007 年の PT 調査データを用いて、この現象を検証しました。混雑時間帯である朝 8 時台のトリップを対象に、全 2,140 本のリンクを一本ずつ仮想的に「消去」し、総旅行時間の変化を網羅的に計算しました。その結果、全リンクの約 40 %に相当する 854 本において、その道を消去したほうが都市全体の総旅行時間を短縮できるという結果が得られました。その多くは微差に留まるものの、一本消去するだけで総旅行時間が 1 時間近く短縮されるリンクも確認されました。

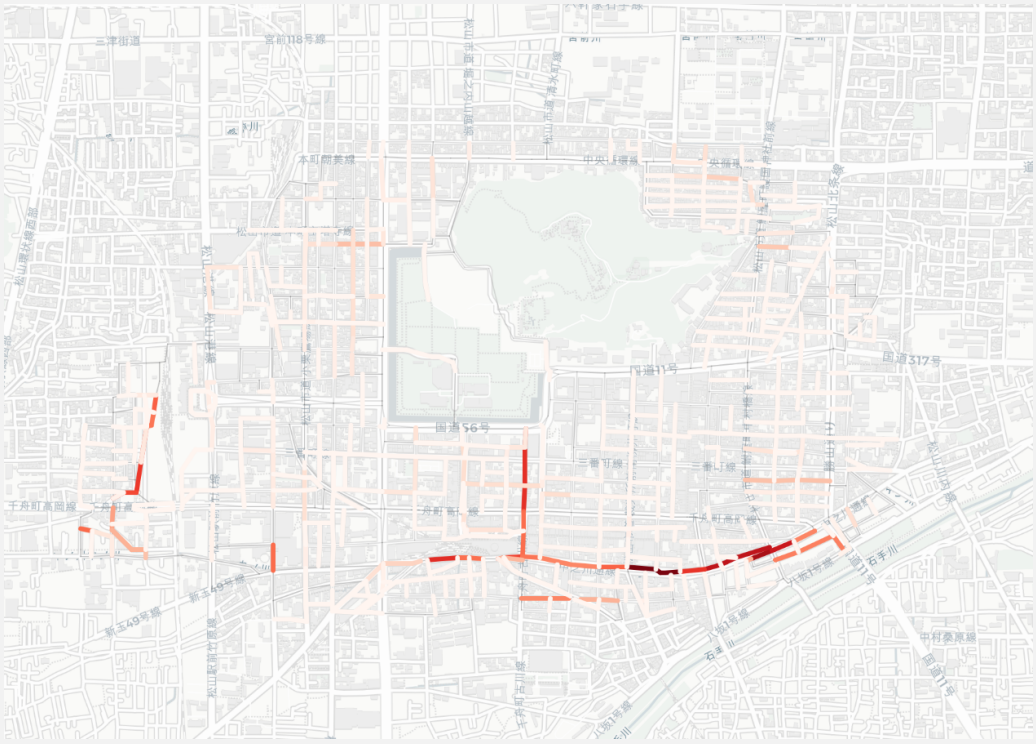


図 2.10: 計算結果を地図上に示したものです。図の色のついた道が消すことによって総旅行時間を減少させる道で、色が濃いほど減少効果が大きいことを示しています。

「中ノ川通り」閉鎖実験：なぜ主要幹線が渋滞を招くのか

具体例として、松山市の主要幹線道路である「中ノ川通り」の東行き車線（松山市駅から離れる方向）に注目します。この道は片側 3 車線で交通量も多い区間ですが、シミュレーション上で図の青色の矢印付近のリンクを閉鎖したところ、全体の総旅行時間が約 32 分も短縮

されるという結果が出ました。交通の変化としては、中ノ川通りを閉鎖することで、それまで特定区間に集中していた交通が南北の隣接道路へと分散されます。一見すると迂回によるロスが増えるように思えますが、閉鎖前、中ノ川通りの松山市駅付近（下図でむらさき色で囲った部分、閉鎖した区間よりも西側）において交通量が過剰であったために、この区間での時間ロスが大きくなっていました。中ノ川通りの閉鎖によりこのむらさき色で囲った混雑区間の交通量が減少し、その減少効果は迂回による増加分を上回ったためにこのような結果が得られたと解釈しました。

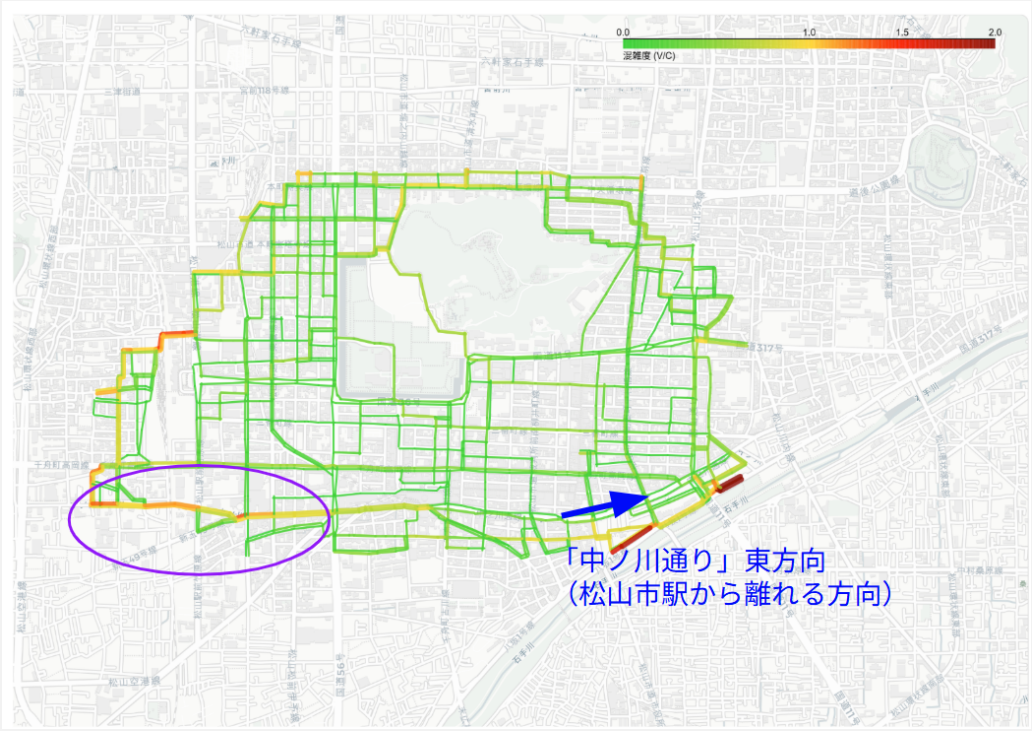


図 2.11: 各リンクの混雑度を図示したものです。オレンジ色や赤色のリンクが混雑していることを示しています。

JR 松山駅高架化がもたらす効果

現在進められている JR 松山駅の高架化に合わせ、JR 駅の下を通り抜ける道路が新設される計画となっています。本検証では、実際に計画されているのと同じ駅を東西に貫く仮想道路（片側 2 車線、延長 286m）を新設し、その影響を計測しました。その結果、都市全体の総旅行時間は 18.89 時間も減少するという結果が得られました。新設道路により、既存の周辺道路の交通量は大きく変化します。シミュレーション上では、東行きで 1249 台、西行きで 532 台もの交通量が新設区間へと転換されました。注目すべきは、これまで渋滞が激しかった特定区間の交通負荷を、新設リンクが「バイパス」として効率よく引き受けている（地図を拡大しているため分かりづらいですが、先ほどの図でむらさき色で囲ったリ

ンクの交通量が大幅に減っている）点です。

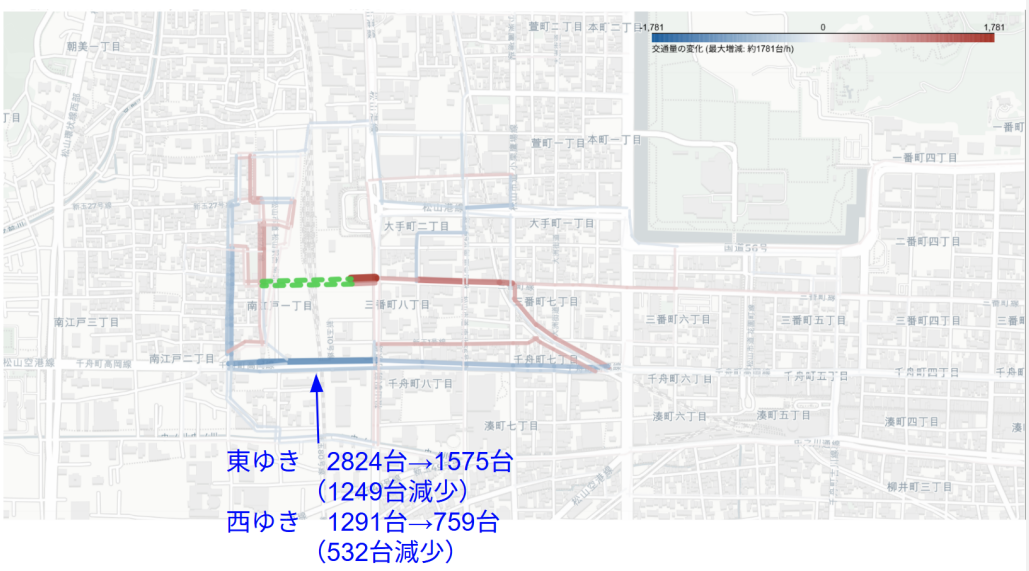


図 2.12: 黄緑色の点線が今回の検証で新設したリンク。この地図ではリンク新設の前後での交通量の増減を示しており、赤が増加・青が減少を示しています。

混雑緩和のメカニズム：下に凸の増加関数を攻略する

なぜ、特定の地点への交通集中を緩和することが、これほどまでに大きな効果を生むのでしょうか。その鍵は、交通工学で用いられるリンクパフォーマンス関数（BPR 関数）の性質にあります。今回の計算に用いたリンクパフォーマンス関数は、本テキストでも既出の以下の式です。

$$t_a(x_a) = t_{a0}\left\{1 + 0.15\left(\frac{x_a}{C_a}\right)^4\right\} \tag{2.58}$$

これは交通量が増加するにつれ、旅行時間が急激に立ち上がる「下に凸の増加関数」です。つまり、渋滞ポイントで交通量を 1 台減らすことで得られる時間短縮効果が、空いている道路での 1 台の削減効果よりも圧倒的に大きくなります。中ノ川通りの閉鎖による交通分散も、JR 松山駅の新設道路によるバイパス効果も、本質的には「極端に混雑しているリンクの負荷を減らす」という点で共通しています。都市全体の旅行時間を最小化するためには、ボトルネックへの流入をいかに制御するかがポイントとなります。

結びに

今回の検証は、松山市という実在の都市構造をモデル化することで、理論上のパラドックスが現実の都市計画においても十分起こり得ることを示唆しました。道路を「消す」ことでも「加える」ことでも交通の流れを改善できる（逆に言えば悪化させる可能性もある）ということは、データに基づいたシミュレーションによる都市計画を進めることの重要さを示しています。

2.6　動的交通配分（DTA）

第 2.3 節から第 2.5 節にかけて扱った交通配分モデルはいずれも，時刻に依存しない**静的**なモデルであった．静的モデルでは，OD 交通量 q_{rs} を時間的に一定な値として与え，ネットワーク全体が定常状態にあることを前提とする．しかし現実の道路交通は，朝のラッシュアワー，夕方のピーク，昼間のオフピークのように，時刻とともに著しく変動する．このような**時刻依存の需要**と，それに伴うキューの形成・伝播・解消を整合的に記述するのが**動的交通配分**（Dynamic Traffic Assignment, DTA）の枠組みである．

本節では，まず静的 UE との対比を通じて DTA の位置づけを整理し（第 2.6.1 小節），続いて動的利用者均衡（DUE）の変分不等式（VI）定式化を導入する（第 2.6.2 小節）．さらに，FIFO 条件に基づく時間分解アルゴリズムによって DTA 問題が効率的に解けることを説明し（第 2.6.3 小節），最後に主な解法を整理して静的 TA との相違点をまとめる（第 2.6.4 小節）[11, 14].

本節で新たに用いる主要記号を表 2.13 にまとめる（基本記号は表 2.1 参照）.

表 2.13: 動的交通配分の主要記号

記号	意味	単位（例）
τ	出発時刻（departure time）	分
t	ネットワーク上の観測時刻（clock time）	分
$q_{rs}(\tau)$	出発時刻 τ の OD 需要流率	台/分
$f_k^{rs}(\tau)$	出発時刻 τ に経路 k を利用する流率	台/分
$x_a(t)$	時刻 t のリンク a の交通量（流率）	台/分
$t_a(x_a(t), t)$	時刻 t のリンク a の旅行時間	分
$\theta_a(t)$	リンク a に時刻 t に進入した車両の退出時刻	分
$C_k^{rs}(\boldsymbol{f}, \tau)$	出発時刻 τ の経路 k^{rs} の所要時間	分
$u_{rs}(\tau)$	出発時刻 τ の OD ペア rs の均衡旅行時間	分
Λ	実行可能フロー集合（FIFO・非負・需要充足を満たす）	–

2.6.1　静的 UE と DTA の対比

静的 UE の限界

静的 UE モデルは，交通量が定常状態でリンクに均等に流れていると仮定する．その強みは，Beckmann 変換（第 2.3 節 第 2.3.2 小節）による凸最適化問題への帰着と，Frank-Wolfe 法による効率的な求解が成立することにある．しかし，以下の点で現実の交通を記述する能力に限界がある：

- OD 交通量 q_{rs} が時刻によって変化することを表現できない（朝ラッシュ・夕方ピーク等）.
- リンク上でのキュー（渋滞の列）の形成・成長・解消という時空間的進展を記述できない.
- 利用者の**出発時刻選択**（departure time choice）を内生的に扱えない.

DTA で導入される変数と概念

DTA はこれらの限界を克服するために，以下の時刻依存変数を陽に導入する：

時刻依存需要 $q_{rs}(\tau)$：出発時刻 τ に OD ペア rs を出発する交通流率．静的 UE の OD 交通量 q_{rs} は，$q_{rs} = \int q_{rs}(\tau)\, d\tau$ と解釈できる．

時刻依存フロー $f_k^{rs}(\tau)$：出発時刻 τ に経路 k^{rs} を選択する流率．需要充足条件は $\sum_k f_k^{rs}(\tau) = q_{rs}(\tau)$ となる．

時刻依存リンクコスト $t_a(x_a(t), t)$：時刻 t のリンク a の旅行時間．リンク内のキュー状態を通じてフロー $x_a(t)$ に依存する．

フロー伝播（Flow Propagation） 車両がリンクを通過する際の入出力関係．時刻 t にリンク a に進入した車両が退出する時刻 $\theta_a(t)$ は，リンク旅行時間によって $\theta_a(t) = t + t_a(x_a(t), t)$ と定まる（リンク遅延モデル）．

FIFO 条件

DTA の物理的整合性を保証する重要な制約が **FIFO 条件**（First-In First-Out）である：

> リンク a へ時刻 $t_1 < t_2$ に進入した 2 台の車両について，退出時刻 $\theta_a(t_1) \leq \theta_a(t_2)$ が成立する．

FIFO 条件は，後から進入した車両が先行車両を追い越せないという当然の物理的制約である．これが満たされないと，現実に観察されない「早出しほど遅くなる」という矛盾が生じる．DTA において FIFO 条件が特に重要な役割を果たすのは，**時間方向の一方向依存性**，すなわち「過去の状態が未来に影響するが，逆はない」ことを保証するためである．この性質が後述の時間分解アルゴリズム（第 2.6.3 小節）の成立根拠となる．

時刻依存 Wardrop 条件

静的 UE における Wardrop の第 1 原則（第 2.3 節 第 2.3.1 小節）は，DUE において次のように拡張される：

> 出発時刻 τ に正のフローを持つすべての使用経路の旅行時間 $C_k^{rs}(\boldsymbol{f}, \tau)$ は等しく，未使用経路の旅行時間以下である．

静的 UE との本質的な相違は，この均衡条件が「出発時刻 τ ごとに」成立することに加え，旅行時間 C_k^{rs} の計算がフロー伝播（FIFO 制約）を介してネットワーク全体の時刻依存フロー $\boldsymbol{f}$ に複雑に依存する点にある．

2.6.2　動的利用者均衡（DUE）の定式化

DUE 条件の相補性表現

静的 UE と同様に，DUE 条件は相補性スラック条件として書ける：各出発時刻 τ について需要 $q_{rs}(\tau) > 0$ が存在するとき，

$$u_{rs}(\tau) \leq C_k^{rs}(\boldsymbol{f}^*, \tau) \qquad \forall k \in K_{rs} \tag{2.59}$$

$$f_k^{rs}(\tau)\left[C_k^{rs}(\boldsymbol{f}^*, \tau) - u_{rs}(\tau)\right] = 0 \qquad \forall k \in K_{rs} \tag{2.60}$$

ここで $u_{rs}(\tau)$ は出発時刻 τ の均衡旅行時間，$C_k^{rs}(\boldsymbol{f}^*, \tau)$ は均衡フロー $\boldsymbol{f}^*$ のもとでの経路旅行時間である．式 (2.59) は「すべての経路の旅行時間は均衡値以上（未使用経路では不等号が成立し得る）」，式 (2.60) は「使用経路の旅行時間は均衡値に一致」という条件を表す点で，静的 UE の KKT 条件（第 2.3 節 第 2.3.2 小節参照）と構造的に同一である．

変分不等式定式化

Friesz et al.（1993）は，この DUE 条件が次の**変分不等式**（Variational Inequality, VI）として等価に表現できることを示した [11]：

$$\langle C(\boldsymbol{f}^*),\, \boldsymbol{f} - \boldsymbol{f}^* \rangle \;\geq\; 0 \qquad \forall \boldsymbol{f} \in \Lambda \tag{2.61}$$

ここで $\langle \cdot, \cdot \rangle$ は関数内積（各出発時刻 τ についての積分），$C(\boldsymbol{f})$ は経路旅行時間ベクトル，Λ は FIFO 条件・非負条件・需要充足条件を満たす実行可能フロー集合である．式 (2.61) の直感は，「均衡フロー $\boldsymbol{f}^*$ において，他の実行可能フロー $\boldsymbol{f}$ の方向へ移動しても旅行時間コスト $\langle C(\boldsymbol{f}^*), \boldsymbol{f} - \boldsymbol{f}^* \rangle$ は減少しない」という条件である．

静的 UE との本質的な相違

静的 UE は Beckmann 変換によってポテンシャル関数を持つ凸最適化問題（最小化問題）に帰着できた（第 2.3.2 小節）．これはリンク費用行列 $\partial C_k / \partial f_l$ が対称であることに起因する．一方，DUE では時刻依存のフロー伝播を通じて，ある経路・出発時刻の変化が他の経路・出発時刻の旅行時間に**非対称**に影響する．このため費用ヤコビアンは一般に非対称となり，ポテンシャル関数が存在しない．

変分不等式定式化はこの非対称性を正面から受け入れることで，Beckmann 変換なしに解の**存在**を議論できる枠組みを与える：C の連続性と Λ のコンパクト凸性から不動点定理によって少なくとも 1 つの解の存在が保証される（表 2.14）．ただし**一意性**については，C の強単調性 $\langle C(\boldsymbol{f}) - C(\boldsymbol{g}),\, \boldsymbol{f} - \boldsymbol{g} \rangle > 0$（$\boldsymbol{f} \neq \boldsymbol{g}$）が追加で成立する場合にのみ保証され，一般の DTA では保証されない．

表 2.14: 静的利用者均衡（UE）と動的利用者均衡（DUE）の比較

比較項目	静的 UE（第 2.3 節）	動的 UE（本節）
需要の表現	一定 q_{rs}	時刻依存 $q_{rs}(\tau)$
フロー変数	f_k^{rs}（スカラー）	$f_k^{rs}(\tau)$（関数）
フロー伝播	なし	FIFO 制約あり
費用行列	対称（凸ポテンシャル存在）	一般に非対称
定式化	凸最適化（Beckmann 変換）	変分不等式（VI）
解の存在	凸問題ゆえ保証	不動点定理で保証
解の一意性	費用関数が狭義単調増加なら一意	強単調性が必要（一般不保証）
計算コスト	低〜中	高（時間分解で緩和可）

2.6.3　時間分解と効率的アルゴリズム

基本的考え方

DTA 問題を「全時刻帯を一括で解く」のではなく，**時刻スライスを順番に処理する**方式（時間分解，Time Decomposition）に変換することで計算を大幅に効率化できる [14]．この方式の成立根

拠は FIFO 条件（第 2.6.1 小節）にある：出発時刻 $\tau_1 < \tau_2$ の 2 台の車両について退出時刻の順序 $\theta_a(\tau_1) \le \theta_a(\tau_2)$ が保存されるため，早い時刻に出発した車両がリンクに与えた影響は，遅い時刻の出発を処理する段階ですでに確定している．したがって，**各時刻スライスは前スライスまでの処理結果を入力として受け取るだけでよい**という時間方向の一方向依存性が成立し，全時刻帯を連立させることなくスライスを逐次的に処理できる．

時刻スライスを $\tau_1 < \tau_2 < \cdots < \tau_T$（スライス幅 $\Delta\tau$）とし，各スライスで出発する需要をリンク遅延モデルを通じて伝播させる．以下では，ある固定した起点 $r \in \mathcal{R}$ について 1 スライスずつ処理する基本手順を示す．

アルゴリズムの詳細

Kuwahara & Akamatsu（2001）は，物理的キューを伴う多対多 OD 問題に対して時間分解アルゴリズムを厳密に定式化した [14]．以下では，$\mathcal{O}_i = \{j \in \mathcal{N} : (i,j) \in \mathcal{A}\}$（ノード i の後継ノード集合），$\mathcal{I}_i = \{j \in \mathcal{N} : (j,i) \in \mathcal{A}\}$（ノード i の先行ノード集合）と記す．

Step 1（**初期化**）　$\ell \leftarrow 1$（最初の時刻スライス）を選択する．全リンク・全時刻において進入フローをゼロに初期化する：

$$x_a(t) = 0, \qquad \forall a \in \mathcal{A},\ t \in [0, T_{\max}]$$

各リンクのリンク遅延関数 $t_a(\cdot,\cdot)$ および自由走行時間 t_a^0 を入力データとして読み込む [1]．

Step 2（**リンク旅行時間の更新**）　時刻スライス ℓ の処理開始時点において，$\tau_1, \dots, \tau_{\ell-1}$ に出発した車両によって各リンクに蓄積されたフロー $\{x_a(t)\}$ をもとに，**リンク遅延モデル**を用いて各リンク a の退出時刻を更新する：

$$\theta_a(t) = t + t_a\big(x_a(t),\, t\big) \qquad \forall a \in \mathcal{A},\ \forall t \tag{2.62}$$

物理的キューを陽に扱う場合は，LWR 理論に基づく累積車両台数曲線（Newell–Daganzo モデル）を用いて $\theta_a(t)$ を計算する [2]．

Step 3（**時刻依存最短経路計算**）　起点 r からの出発時刻を τ_ℓ に固定して，**時刻依存 Dijkstra 法**にて各ノード i への最早到着時刻 $\pi_r(i)$ を計算する．ラベル更新規則は，静的 Dijkstra 法（第 2.2 節）においてリンク費用を「進入時刻に依存する退出時刻」に置き換えたものである：

$$\pi_r(r) = \tau_\ell, \qquad \pi_r(j) = \min_{a=(i,j)\in\mathcal{A}} \theta_a\big(\pi_r(i)\big) \tag{2.63}$$

ノード i に時刻 $\pi_r(i)$ に到着した車両はリンク $a=(i,j)$ に進入し，時刻 $\theta_a(\pi_r(i))$ にノード j へ退出するとみなす．FIFO 条件 $\partial\theta_a/\partial t \ge 0$ のもとではラベル値が単調増加し，静的 Dijkstra 法と同様の貪欲選択が有効であるため，一度確定したラベルを再更新する必要がない．各ノード j で先行ポインタ $F_j \leftarrow \arg\min_{(i,j)} \theta_a(\pi_r(i))$ を記録する [3]．

---

1) Step 1 の計算量は $O(|\mathcal{A}|)$（全リンクのフロー初期化）．

2) Step 2 の計算量は $O(|\mathcal{A}|)$（全リンクのリンク遅延関数更新）．

3) Step 3 の計算量は $O\big((|\mathcal{N}| + |\mathcal{A}|)\log|\mathcal{N}|\big)$（優先度付きキューを用いた Dijkstra 法）．FIFO 条件が破れる場合は動的計画法により $O(T \cdot |\mathcal{A}|)$ となる．

Step 4（**AoN 配分**） Step 3 で得た最短経路ツリーに需要 $q_{rs}(\tau_\ell)$ を集中させる（All-or-Nothing 配分，第 2.2 節参照）．最短経路ツリーを $\pi_r(j)$ の降順（起点 r から遠い順）にたどりながら，先行ポインタ F_j で定まるリンク $a_j = (F_j,\, j)$ への配分フロー y_{a_j} を集計する：

$$y_{a_j} = q_{rj}(\tau_\ell) + \sum_{m \in \mathcal{O}_j^{\mathrm{tree}}} y_{a_m} \tag{2.64}$$

ここで $q_{rj}(\tau_\ell)$ は起点 r・目的地 j・出発時刻 τ_ℓ の OD 需要流率，$\mathcal{O}_j^{\mathrm{tree}}$ は最短経路ツリー上でノード j から流出するリンクの終点集合である（式 (2.5) と同構造）[4]．

Step 5（**フロー伝播**） Step 4 で各リンク a に配分された進入フロー y_a を FIFO 制約のもとで前方伝播させ，リンク交通量 $x_a(t)$ を更新する．経路上のリンクを出発地側から目的地側へ順に走査し，車両がリンク $a = (i,j)$ に進入する時刻 t_a^{in}（直前リンク $a^- = (h,i)$ の退出時刻 $\theta_{a^-}(t_{a^-}^{\mathrm{in}})$）においてフローを加算する：

$$x_a\big(t_a^{\mathrm{in}}\big) \;\mathrel{+}=\; y_a \tag{2.65}$$

フロー加算後，式 (2.62) によりリンク遅延関数 θ_a が更新され，後続リンクへの影響が順次前方に伝播する（連鎖更新）．この処理は経路上のリンクを順に走査するだけで完了する [5]．

Step 6（**繰り返し**） $\ell < T$ ならば $\ell \leftarrow \ell + 1$ として Step 2 へ戻る．全スライス $\tau_1, \dots, \tau_T$ を処理したら終了する．

全スライスの処理完了後，各スライスで蓄積されたリンクフロー $x_a(t)$ が出発時刻ごとの動的 AoN 配分結果を表す．DUE 条件を達成する均衡解は，本アルゴリズムを内側ルーティングとして，外側で MSA 等の均衡探索反復と組み合わせることで求める（詳細は 第 2.6.4 小節 参照）．

時刻依存 Dijkstra と静的 Dijkstra の比較

Step 3 の時刻依存 Dijkstra 法は，静的 Dijkstra 法（第 2.2 節）とアルゴリズム構造を共有しつつ，表 2.15 の点で異なる．

表 2.15: 静的・時刻依存 Dijkstra 法の比較

比較項目	静的 Dijkstra（第 2.2 節）	時刻依存 Dijkstra（Step 3）
ラベル $\pi(i)$ の意味	起点からの最短費用	起点からの最早到着時刻
リンク評価値	定数リンク費用 t_{ij}	退出時刻 $\theta_a(\pi_r(i))$
ラベル更新式	$\pi(j) \leftarrow \pi(i) + t_{ij}$	$\pi(j) \leftarrow \theta_a\big(\pi_r(i)\big)$
貪欲選択の正当性	リンク費用の非負性	FIFO 条件（$\partial\theta_a/\partial t \geq 0$）
計算量	$O\big((\lvert\mathcal{N}\rvert + \lvert\mathcal{A}\rvert)\log\lvert\mathcal{N}\rvert\big)$	同じ（FIFO 成立時）

FIFO 条件が破れる状況（例：過飽和キューでの到着順序逆転）では貪欲選択が有効でなくなり，動的計画法を用いた $O(T \cdot |\mathcal{A}|)$ のアルゴリズムに切り替える必要がある [11]．

4) Step 4 の計算量は $O(|\mathcal{N}|)$（最短経路ツリー上の後退走査）．
5) Step 5 の計算量は $O(|\mathcal{N}|)$（経路長 $|P| \leq |\mathcal{N}| - 1$ のリンク走査とフロー加算）．

多対多 OD への拡張

複数起点 $r \in \mathcal{R}$ を持つ多対多 OD 問題では，**起点別分解**と**流量加算原理**を用いてアルゴリズムを $|\mathcal{R}|$ 回のループとして実装できる．各起点 r について Step 3〜5 を独立に実行して起点別リンクフロー $x_a^{(r)}(t)$ を求め，全起点にわたって累計する：

$$x_a(t) = \sum_{r \in \mathcal{R}} x_a^{(r)}(t) \tag{2.66}$$

この重ね合わせは，リンクフローが各起点フローの線形和として定義されるという加法性に基づいており，FIFO 条件とは独立に成立する．FIFO 条件が起点別分解に必要な理由は，同一リンク上で各起点のフローが整合したリンク遅延関数 θ_a のもとで伝播することを保証するためである．一方，リンク旅行時間 $t_a(x_a(t), t)$ は合計フロー $x_a(t)$ に依存するため，Step 2 のリンク旅行時間更新には累計フローを使用する点に注意する．この分解による全体の計算量は，1 スライス・1 起点あたりの支配的コスト（Step 3 の Dijkstra 法）に $T \times |\mathcal{R}|$ を乗じて次式で与えられる [14]：

$$O\Big(T \cdot |\mathcal{R}| \cdot \big(|\mathcal{N}| + |\mathcal{A}|\big) \log |\mathcal{N}|\Big). \tag{2.67}$$

ネットワーク規模 $|\mathcal{N}|, |\mathcal{A}|$ を定数とみなせば，T と $|\mathcal{R}|$ に対して線形であり，全時刻帯を連立して解く一括法（サイズ $T \cdot |\mathcal{A}|$ の連立方程式を解く方式）と比べて計算量を大幅に抑えられる．

2.6.4　解法と静的 TA との比較

主な解法アプローチ

DTA 問題に対しては，静的 TA と同様に反復的な数値解法が用いられる．代表的なアプローチを表 2.16 に示す．

表 2.16: DTA の主な解法アプローチ

アプローチ	代表手法	特徴
逐次代入法	MSA（逐次平均法）	実装が容易；収束保証が弱い
VI 解法	Gap function 最小化	理論的に厳密；計算コスト大
時間分解法	時刻スライス逐次処理	FIFO 前提；実用的な計算効率
シミュレーション基盤	MATSim，VISSIM	大規模ネットワーク向き

MSA（逐次平均法）

MSA は，Frank-Wolfe 法の線探索ステップサイズを $\alpha^{(n)} = 1/n$ に固定した特殊ケースと見なせる反復法である（第 2.3 節 第 2.3.3 小節参照）：

$$\boldsymbol{f}^{(n+1)} = \left(1 - \frac{1}{n}\right) \boldsymbol{f}^{(n)} + \frac{1}{n} \boldsymbol{h}^{(n)} \tag{2.68}$$

ここで $\boldsymbol{h}^{(n)}$ は現在のフロー $\boldsymbol{f}^{(n)}$ のもとでの All-or-Nothing 配分結果である．MSA は実装が単純なため広く用いられるが，ステップサイズが固定されているため，一般に最適収束率を保証しない．

Gap function による収束判定

DUE の収束指標として，静的 UE の相対ギャップ（第 2.3 節 第 2.3.3 小節参照）を時刻依存に拡張した **Gap function** が用いられる：

$$\mathrm{Gap}(\boldsymbol{f}) = \int \sum_{rs} \sum_{k} f_k^{rs}(\tau) \left[C_k^{rs}(\boldsymbol{f}, \tau) - u_{rs}(\tau) \right] d\tau \;\geq\; 0 \tag{2.69}$$

DUE 解 $\boldsymbol{f}^*$ では $\mathrm{Gap}(\boldsymbol{f}^*) = 0$ が成立する（式 (2.60) より）．数値解法では $\mathrm{Gap}(\boldsymbol{f}^{(n)}) / \int \sum_{rs} u_{rs}(\tau) q_{rs}(\tau)\, d\tau$ が規定の閾値 ε 以下になった時点で収束と判定する．

本節のまとめ

本節で示した主要な結果を整理する：

1. **動的交通配分**（DTA）は，時刻依存の需要 $q_{rs}(\tau)$，フロー伝播，リンクコスト $t_a(x_a(t), t)$ を明示的に扱い，時刻依存 Wardrop 条件を満たす動的利用者均衡（DUE）を求める枠組みである；FIFO 条件が時間方向の一方向依存性を保証し，問題の解析可能性を支える（第 2.6.1 小節）．
2. DUE 条件は**変分不等式** $\langle C(\boldsymbol{f}^*), \boldsymbol{f} - \boldsymbol{f}^* \rangle \geq 0$ として定式化でき（Friesz et al., 1993），C の連続性と Λ のコンパクト凸性から解の存在が保証される；ただし費用ヤコビアンの非対称性により，静的 UE と異なりポテンシャル関数は一般に存在せず，解の一意性も強単調性がなければ保証されない（第 2.6.2 小節）．
3. FIFO 条件に基づく**時間分解アルゴリズム**（Kuwahara & Akamatsu, 2001）は，全時刻一括処理を時刻スライスの逐次処理に変換し，計算量を時刻数と起点数に対して線形オーダーに抑える；多対多 OD への拡張では起点別分解と流量加算原理が成立根拠となる（第 2.6.3 小節）．
4. DTA の解法には逐次代入法（MSA），VI 解法（Gap function 最小化），時間分解法，シミュレーション基盤の 4 種類があり，それぞれ計算効率と理論的厳密さのトレードオフを持つ（第 2.6.4 小節）．

2.7　進化ゲーム理論と交通配分

第 2.3 節では，利用者均衡配分（UE）が Beckmann 変換によって凸最適化問題（UE/FD-Primal）と等価であることを示し，Frank-Wolfe 法によって数値的に解けることを確認した [3, 30]．しかし，静学的な UE 分析が答えるのは「均衡状態がどこにあるか」という問いのみであり，「利用者がどのようにして均衡に到達するか」という**ダイナミクス**の問いには答えない．

本節では，この問いに答えるための枠組みとして**進化ゲーム理論**を導入する [24, 23]．まず，UE がポテンシャルゲームの最小値と等価であるという静学的な関係を整理し（第 2.7.1 小節），この枠組みを進化的 Day-to-day ダイナミクスへ拡張する動機を説明する（第 2.7.2 小節）．続いて **集団ゲーム**（Population Game）の基本概念を導入し（第 2.7.3 小節），有限合理性のもとでの乗り換え行動を記述する**改訂プロトコル**（Revision Protocol）と**集団ダイナミクス**（Mean Dynamics）の理論的枠組みを示す（第 2.7.4 小節）．第 2.7.5 小節 では交通配分に関連する代表的な進化動学を紹介し，第 2.7.6 小節 で収束定理を述べ，第 2.7.7 小節 で Day-to-day ダイナミクスの交通政策への含意

を考察する．

本節で新たに用いる主要記号を表 2.17 にまとめる（基本記号は表 2.1 参照）．

表 2.17: 進化ゲームと Day-to-day ダイナミクスの主要記号

記号	意味	単位（例）
$\boldsymbol{x} \in X$	社会状態ベクトル（各戦略を選択している利用者数）	台/時
$F_i(\boldsymbol{x})$	戦略 i の利得関数（$= -c_k^{rs}$；負の旅行時間）	—
$\bar{F}(\boldsymbol{x})$	集団全体の平均利得 $= \sum_i x_i F_i / m$	—
$\rho_{ij}(F, \boldsymbol{x})$	改訂プロトコル（戦略 i から戦略 j への乗り換え率）	日 $^{-1}$
$\dot{f}_k^{rs}$	経路交通量の時間変化率（Day-to-day ODE の左辺）	—
$\boldsymbol{V}(\boldsymbol{f})$	集団ダイナミクスのベクトル場（$\dot{\boldsymbol{f}} = \boldsymbol{V}(\boldsymbol{f})$）	—

2.7.1　ポテンシャルゲームとしての UE

ポテンシャルゲームの概念

UE の最適化問題（第 2.3.2 小節）をゲーム理論の視点から捉え直すと，その背後に**ポテンシャルゲーム**（Potential Game）の構造が見えてくる [17].

ポテンシャルゲームとは，すべてのプレーヤーの「利得の変化」が，1 つのグローバルな関数 Q（**ポテンシャル関数**）の変化にきっかりと等しいゲームである．すなわち，プレーヤー a が戦略を i から j に変更したとき：

$$U_a(j,\, \boldsymbol{s}_{-a}) - U_a(i,\, \boldsymbol{s}_{-a}) = Q(j,\, \boldsymbol{s}_{-a}) - Q(i,\, \boldsymbol{s}_{-a}) \tag{2.70}$$

が成り立つ（$\boldsymbol{s}_{-a}$ は他のプレーヤーの戦略プロファイル）[17]．この条件のもとでは，Q を最小化する点が Nash 均衡と一致する——複数のプレーヤー間の均衡を，単一の最小化問題に置き換えられる点がポテンシャルゲームの大きな強みである．

交通配分のポテンシャル関数

n 人のドライバー（プレーヤー）が OD ペア rs 間の経路を選択するゲームを考える．ドライバー a が選ぶ経路 k の費用（負の利得）は経路旅行時間 $c_k^{rs}(\boldsymbol{f})$ である（第 2.3 節参照）．

ポテンシャル関数として Beckmann 変換の目的関数

$$Z(\boldsymbol{x}) = \sum_{a \in A} \int_0^{x_a} t_a(w)\, dw \tag{2.71}$$

を採用すると，式 (2.70) の条件が満たされることが示せる [17, 23]．離散 n 人ゲームの対応物として，1 人のドライバーが経路 k から経路 l に変更したときの Z の変化量は，連続近似のもとでそのドライバーの旅行時間変化 $c_l^{rs} - c_k^{rs}$ に等しい（n が十分大きい極限での議論は第 2.12 節で別途扱う）．

結論として，

- Z を最小化する経路フロー $\boldsymbol{f}^*$ が Nash 均衡 $=$ UE と一致する，
- どのドライバーも Z が最小の状態では経路変更のインセンティブを持たない，

という 2 つの事実が成り立つ [23].

UE をポテンシャルゲームの最小値として捉える利点は静学的解析に留まらない．次節以降で見るように，この構造は進化動学の **Lyapunov 関数**としての役割を果たし，ダイナミクスの安定性分析を可能にする．

2.7.2　進化ゲームと Day-to-day ダイナミクス

静学モデルの限界

混雑ゲームの静学的 Nash 均衡分析（第 2.3 節，第 2.12 節）は「均衡状態はどこか」という問いには答えるが，次の問いには答えない：

- 利用者は均衡状態を知っているのか？
- 均衡への収束は保証されるか？
- 均衡から外れた初期状態からどのような経路を辿って均衡に到達するか？

UE モデルは「完全情報」と「完全合理性」を前提とする（第 2.3.1 小節参照）．しかし現実の交通行動では，利用者は前日の旅行時間をもとに翌日の経路を決定するなど，**有限合理性**（Bounded Rationality）に基づいた漸次的な調整行動が観察される．これが **Day-to-day ダイナミクス**の研究動機である．

進化ゲーム理論による拡張

進化ゲーム理論（Evolutionary Game Theory）は，もともと生物進化（Taylor & Jonker 1978）や社会学習（Smith 1984）を記述するために発展した理論であり，プレーヤー集団の戦略分布が時間とともに変化するダイナミクスを扱う [28, 27].

交通工学における Day-to-day ダイナミクス（day-to-day adjustment process）は，進化ゲーム理論の枠組みと自然に対応する：

進化ゲーム理論	Day-to-day ダイナミクス
プレーヤー集団	通勤者・ドライバー集団
戦略	経路選択
集団状態	経路交通量 f_k^{rs}
利得	負の旅行時間（$-c_k^{rs}$）
戦略改訂	日々の経路変更行動
Nash 均衡	Wardrop UE

Sandholm（2001, 2010）は，ポテンシャルゲームに対して適切な条件を満たす進化動学を導入することで，**合理性の完全性を仮定しなくても長期的に UE へ収束すること**を理論的に証明した [23, 24]．本章の後半でこの収束定理を詳しく述べる（第 2.7.6 小節）．

2.7.3　集団ゲームの枠組み

集団ゲームの基本要素

Sandholm（2010）の**集団ゲーム**（Population Game）$\mathcal{G}$ は次の 3 要素で定義される [24]：

社会状態 $\boldsymbol{x} \in X$　n 個の戦略（純粋戦略）のそれぞれを選択している利用者数を並べたベクトル：

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^n_{\geq 0} \;\middle|\; \sum_{i=1}^{n} x_i = m \right\} \tag{2.72}$$

ここで $m > 0$ は集団の規模（総プレーヤー数）を表す．交通配分では $m = q_{rs}$（OD 交通量），$x_i = f_k^{rs}$（経路 k の交通量）に対応する．

利得関数 $F\colon X \to \mathbb{R}^n$　社会状態 $\boldsymbol{x}$ に応じた各戦略の利得を表す写像．$F_i(\boldsymbol{x})$ は戦略 i を選択した利用者が得る利得である．交通配分では，負の旅行時間 $F_i(\boldsymbol{x}) = -c_k^{rs}(\boldsymbol{f})$ が利得に対応する（コストが小さいほど選好される）．

Nash 均衡 $\boldsymbol{x}^* \in X$　どの利用者も戦略を変更することで利得を改善できない状態：

$$x_i^* > 0 \implies F_i(\boldsymbol{x}^*) \geq F_j(\boldsymbol{x}^*) \quad \forall j \tag{2.73}$$

すなわち，正の流量を持つ戦略は必ず利得を最大化している．交通配分では，条件 (2.73) は Wardrop UE 条件（式 (2.20)）と一致する．

交通配分との対応

単一の OD ペア rs に対して，経路集合 K_{rs} の要素（経路）を戦略とし，集団の規模を OD 交通量 q_{rs} とすると，集団ゲームの各要素は次のように対応する：

集団ゲーム要素	交通配分の対応	式
戦略 i	経路 $k^{rs} \in K_{rs}$	
集団規模 m	OD 交通量 q_{rs}	
社会状態 x_i	経路交通量 f_k^{rs}	式 (2.23)
実行可能集合 X	フロー保存集合 $\sum_k f_k^{rs} = q_{rs}$	
利得 $F_i(\boldsymbol{x})$	負の経路旅行時間 $-c_k^{rs}(\boldsymbol{f})$	
Nash 均衡	Wardrop UE	式 (2.20)

複数 OD ペアが存在する場合は，OD ペアごとに独立な集団ゲームを設定し，各集団のダイナミクスを並列に扱う．

2.7.4　改訂プロトコルと進化動学

限定合理性の定式化

完全合理性を仮定する Nash 均衡分析では，プレーヤーは常に最良の戦略を選択することが前提となっている．しかし Day-to-day モデルでは，利用者は情報制約や慣性のもとで**確率的・漸次的に戦略を変更する**ことが現実的である．

この行動を定式化するのが**改訂プロトコル**（Revision Protocol）である [24].

改訂プロトコル $\rho_{ij}(F,\boldsymbol{x}) \geq 0$ 現在，戦略 i を選択しているエージェント（利用者）1 人が，単位時間あたりに戦略 j へ乗り換える条件付き確率（率）：

$$\rho_{ij}(F,\boldsymbol{x}) \geq 0 \qquad \text{(乗り換え率)} \tag{2.74}$$

ρ_{ij} は現在の利得のベクトル $F(\boldsymbol{x})$ と社会状態 $\boldsymbol{x}$ の関数として与えられる．限定合理性の具体的なモデル（どの情報をどう使うか）は ρ_{ij} の選び方によって規定される．

集団ダイナミクス（Mean Dynamics）

個人レベルの確率的乗り換え行動から，集団全体の平均的な運動方程式（**Mean Dynamics**）を導出できる [24]．戦略 i の利用者数 x_i の時間変化率は，「他の戦略から i への流入」マイナス「i から他の戦略への流出」で与えられる：

$$\dot{x}_i = \sum_{j\neq i} x_j\,\rho_{ji}(F,\boldsymbol{x}) - x_i \sum_{j\neq i} \rho_{ij}(F,\boldsymbol{x}) \tag{2.75}$$

式 (2.75) の右辺：

- 第 1 項：戦略 j を採用している x_j 人のうち，率 ρ_{ji} で戦略 i に乗り換える者の流入量．
- 第 2 項：戦略 i を採用している x_i 人のうち，率 ρ_{ij} で他の戦略 j に乗り換える者の流出量．

交通配分の文脈では，戦略 i を経路 k^{rs} と読み替え，$x_i = f^{rs}_k$，$F_i = -c^{rs}_k$ とすることで，Day-to-day 調整過程の常微分方程式（ODE）が得られる．

フロー保存の維持

式 (2.75) において，$\sum_i \dot{x}_i = 0$ が任意の時刻で成立すること，すなわち社会状態が常に実行可能集合 X（式 (2.72)）内に留まることが確認できる [24]：

$$\sum_i \dot{x}_i = \sum_i \sum_{j\neq i} x_j\rho_{ji} - \sum_i x_i \sum_{j\neq i} \rho_{ij} = \sum_{i\neq j} x_j\rho_{ji} - \sum_{i\neq j} x_i\rho_{ij} = 0$$

（添字の付け替えによる）．このため，初期条件 $\boldsymbol{x}(0) \in X$ を設定すれば，すべての時刻で $\boldsymbol{x}(t) \in X$ が保証される．

2.7.5 代表的な進化動学

解析的・計算的に扱いやすいいくつかの進化動学を紹介する．各動学は改訂プロトコル ρ_{ij} の選び方によって定まり，異なる限定合理性モデルに対応する [24]．

Smith ダイナミクス

Smith ダイナミクス（Smith Dynamics）は，乗り換え率をコスト差に比例させるモデルであり，交通工学の文脈では Smith（1984）[27] が提案したことでよく知られる．

改訂プロトコル

$$\rho^{\mathrm{S}}_{ij}(F,\boldsymbol{x}) = \big[F_j(\boldsymbol{x}) - F_i(\boldsymbol{x})\big]_+ = \big[-c^{rs}_j + c^{rs}_i\big]_+ \tag{2.76}$$

ここで $[\cdot]_+ := \max(\cdot, 0)$ は正値部分（負の値をゼロとする操作）を表す．すなわち，乗り換え先 j のほうが現在の戦略 i より利得が大きい（コストが低い）場合のみ，コスト差に比例した率で乗り換えが生じる．

集団ダイナミクス（経路交通量表示）

$$\dot{f}_k^{rs} = \sum_{l \neq k} f_l^{rs} \left[c_l^{rs} - c_k^{rs}\right]_+ - f_k^{rs} \sum_{l \neq k} \left[c_k^{rs} - c_l^{rs}\right]_+ \tag{2.77}$$

右辺第 1 項は「経路 l より経路 k が安価」なとき l から k への乗り換え（流入），第 2 項は「経路 k が他の経路より高価」なとき k から他の経路への乗り換え（流出）を表す．

レプリケーター動学

レプリケーター動学（Replicator Dynamics）は，進化生物学において適応度が高い形質が集団内で増加する過程（Taylor & Jonker 1978）[28] を記述するモデルで，交通行動の「平均旅行時間より良い経路は利用者を引き付ける」という直感を表す：

$$\dot{x}_i = x_i\big(F_i(\boldsymbol{x}) - \bar{F}(\boldsymbol{x})\big) \tag{2.78}$$

ここで $\bar{F}(\boldsymbol{x}) = \sum_k x_k F_k / m$ は集団全体の平均利得（平均旅行時間の負値）である．平均利得より高い利得を持つ戦略（自己の旅行時間が平均より短い経路）は流量が増加し，逆は減少する．ただし，レプリケーター動学では境界（$x_i = 0$）の点が不変集合となるため，もともと利用されていない経路には新規の交通が流入しないという制約がある [24].

最良応答動学

最良応答動学（Best Response Dynamics）は，利用者が常にその時点での社会状態 $\boldsymbol{x}$ に対して最良の戦略（最短経路）へ即座に乗り換えると仮定するモデルである：

$$\dot{\boldsymbol{x}} \in \mathrm{BR}(\boldsymbol{x}) - \boldsymbol{x} \tag{2.79}$$

ここで $\mathrm{BR}(\boldsymbol{x}) = \arg\max_i F_i(\boldsymbol{x})$ は現在の状態での最良応答対応である．この動学は数理的には「右辺不連続な微分包含（Differential Inclusion）」となり，他の動学と比較して最も合理的な乗り換え行動を仮定するが，その分実装や解析がやや複雑になる．

代表的な進化動学の比較

表 2.18: 代表的な進化動学の比較

動学	乗り換え率 ρ_{ij}	ODE の形	特徴
Smith	$[F_j - F_i]_+$	$\dot{x}_i = \sum_j x_j[F_i - F_j]_+ - x_i \sum_j [F_j - F_i]_+$	コスト差比例；連続
レプリケーター	$[F_j - \bar{F}]_+ / m$	$\dot{x}_i = x_i(F_i - \bar{F})$	平均との差；境界不変
最良応答	$\mathbf{1}[j \in \mathrm{BR}]$	$\dot{\boldsymbol{x}} \in \mathrm{BR}(\boldsymbol{x}) - \boldsymbol{x}$	不連続；最速収束

2.7.6　安定性と収束定理

進化動学に課す条件

ここでは，経路フローの時間発展を記述する常微分方程式（ODE）

$$\dot{\boldsymbol{f}} = \boldsymbol{V}(\boldsymbol{f}) \tag{2.80}$$

を考える．$\boldsymbol{V}(\boldsymbol{f}) = \left(V_k^{rs}(\boldsymbol{f})\right)_{rs,k}$ は各経路フローの変化率を与えるベクトル場であり，式 (2.75) の右辺に対応する（Smith ダイナミクスでは式 (2.77)，レプリケーター動学では式 (2.78) の右辺がこれにあたる）．

Sandholm（2001, 2010）は，以下の 5 つの条件を満たす進化動学であれば，ポテンシャルゲームの Nash 均衡（＝ UE）へ収束することを証明した [23, 24].

(L) リプシッツ連続性 $\boldsymbol{V}(\boldsymbol{f})$ はリプシッツ連続である（解の一意存在を保証）．

(N) 非負性 $f_k^{rs} = 0$ ならば $V_k^{rs}(\boldsymbol{f}) \geq 0$（流量ゼロの経路への流入のみ許容）．

(C) フロー保存 $\displaystyle\sum_{k \in K_{rs}} V_k^{rs}(\boldsymbol{f}) = 0,\ \forall rs \in \Omega$（OD 交通量が時間不変）．

(PC) 正相関　（Positive Correlation）

$$\sum_{rs \in \Omega} \sum_{k \in K_{rs}} V_k^{rs}(\boldsymbol{f})\, c_k^{rs}(\boldsymbol{f}) \leq 0 \tag{2.81}$$

旅行費用が高い経路は流量が減少する方向への変化（コスト差に反応する合理性を最低限保証）．

(NS) Nash 固定性　（Nash Stationarity）$\boldsymbol{V}(\boldsymbol{f}^*) = \boldsymbol{0}$ ならば $\boldsymbol{f}^*$ は Nash 均衡（UE）．

Smith ダイナミクス（式 (2.77)）はこれら 5 つの条件をすべて満たす [27, 24]．レプリケーター動学（式 (2.78)）も (L)(N)(C)(PC)(NS) を満たすことが確認できる．

ポテンシャル関数の Lyapunov 性

収束定理の核心は，Beckmann 変換のポテンシャル関数 $Z(\boldsymbol{x})$ が進化動学の **Lyapunov 関数**として機能することにある．

命題（Lyapunov 減少）　正相関条件 (PC) を満たす進化動学のもとでは，

$$\frac{d}{dt} Z(\boldsymbol{f}(t)) = \sum_{rs} \sum_{k} \frac{\partial Z}{\partial f_k^{rs}}\, V_k^{rs}(\boldsymbol{f}) = \sum_{rs} \sum_{k} c_k^{rs}(\boldsymbol{f})\, V_k^{rs}(\boldsymbol{f}) \leq 0 \tag{2.82}$$

が任意の $\boldsymbol{f}(t) \in X$ で成立する（等号は Nash 均衡のみ）．

ここで，$\partial Z / \partial f_k^{rs} = c_k^{rs}$ は Beckmann 変換の勾配計算（第 2.3.2 小節参照）から直接得られる：リンク交通量を通じた連鎖律が経路旅行時間を与えるためである（KKT 条件；式 (2.27) 参照）．不等式は正相関条件 (2.81) そのものであり，Z は単調非増加であること，すなわちポテンシャルゲームの「谷」（Nash 均衡）に向かって系が流れることを意味する．

図 2.13 に 2 経路モデルでの Z の形状と，進化動学の収束軌跡のイメージを示す．

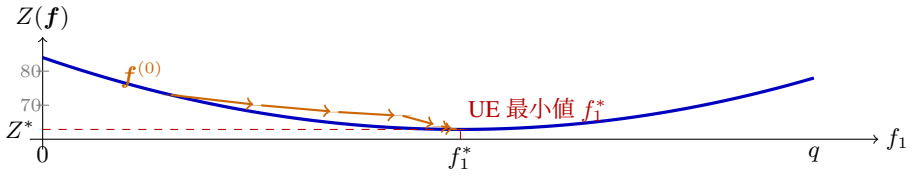


図 2.13: 2 経路モデルにおけるポテンシャル関数 $Z(f_1)$ と収束軌跡の概念図（$c_1(f_1) = 10 + f_1$, $c_2(f_1) = 23 - 3f_1$, $q = 6$）．Z は $f_1^* = 3.25$ を最小値とする凸関数であり，正相関条件を満たす進化動学（橙色の点列）は Z を単調に減少させながら UE に収束する（式 (2.82)）．

収束定理

以上の議論を整理すると，次の定理が得られる．

定理（Sandholm, 2001; 2010）　リンク費用関数 $t_a(\cdot)$ が連続かつ単調非減少であるとき，交通配分の集団ゲームはポテンシャルゲームである．この集団ゲームに対して条件 (L)(N)(C)(PC)(NS) を満たす任意の進化動学 $\dot{\boldsymbol{f}} = \boldsymbol{V}(\boldsymbol{f})$ について，

1. すべての解軌道 $\boldsymbol{f}(t)$ は Nash 均衡（UE）の連結部分集合に収束する，
2. リンク費用関数 $t_a(\cdot)$ が**狭義単調増加**であれば UE は一意に定まり，任意の初期条件から出発した解軌道がその唯一の Nash 均衡（UE）へ収束する（global attractor）．

[23, 24]

BPR 関数（第 2.1.3 小節）のような狭義単調増加・狭義凸なリンク費用関数のもとでは Z も狭義凸となり (ii) の条件が満たされるため，UE の一意性と大域的収束が保証される [31].

図 2.14 に Smith ダイナミクス（式 (2.77)）の 1 次元位相図（$\dot{f}_1$ vs f_1）を示す．$f_1 < f_1^*$ では $\dot{f}_1 > 0$（経路 1 へ流量が増加），$f_1 > f_1^*$ では $\dot{f}_1 < 0$（経路 1 から流量が減少）となり，UE への収束が確認できる．

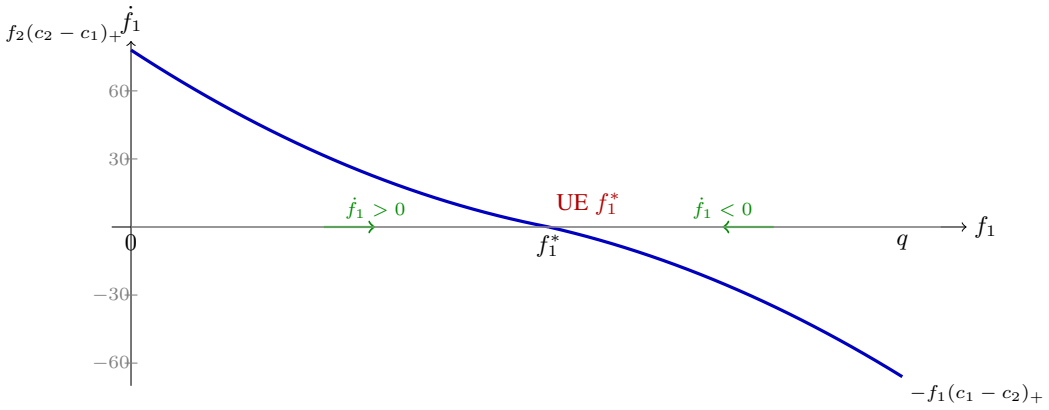


図 2.14: Smith ダイナミクスの 1 次元位相図（2 経路モデル）．$f_1^* = 3.25$ が UE（Nash 均衡点），$\dot{f}_1 = 0$．左側（$f_1 < f_1^*$）では $\dot{f}_1 > 0$，右側（$f_1 > f_1^*$）では $\dot{f}_1 < 0$ となり，全初期条件から UE への大域的収束が見て取れる．

2.7.7　Day-to-day ダイナミクスへの応用

Day-to-day 調整過程の解釈

収束定理は次のことを意味する：**「完全合理性（全情報・完全先読み）を前提としなくても，正相関条件（コストが高い経路を避ける傾向）を持つ日々の経路変更行動が続けば，長期的には Wardrop UE が実現する.」**

具体的に Smith ダイナミクス（式 (2.77)）を Day-to-day 調整過程として解釈すると：

- 各エージェント（通勤者）は毎日，現在使用している経路のコストと対比して，コストが低い代替経路が存在すれば，そのコスト差に比例した確率で経路を変更する.
- 個人レベルでは確率的で局所的な情報処理にすぎないが，集団全体では $\dot{Z}(\boldsymbol{f}(t)) \leq 0$ が保たれ，長期的に UE に収束する.

このメカニズムは，交通行動の観測結果と整合的であり，交通管理・情報提供の効果評価に応用できる.

混雑課金・情報提供の動的評価への含意

静学的な UE 分析では，混雑課金や情報提供の「均衡状態における効果」は評価できるが，「均衡への収束速度や収束の経路（過渡状態）」は評価できない．Day-to-day 進化動学の枠組みでは，次のような問いに対する分析基盤が与えられる：

1. **収束速度への政策効果**: 混雑課金（第 2.5 節 第 2.5.3 小節）は，課金なしの自由均衡（UE）ではなく SO を実現する料金体系を課す．Day-to-day ダイナミクスのもとでは，課金によって経路間コスト差が縮小ないし再構成されるため，収束速度や過渡状態の交通量が変化しうる [32].
2. **情報提供による乗り換え誘導**: 可変情報板（VMS）やナビゲーションシステムによりリアルタイムの経路情報が提供されると，ρ_{ij} の構造が変化し，より正確な情報（コスト差への反応の鋭化）が得られると改訂プロトコルが強化される．これは収束速度の改善として現れるが，過剰反応（全員が同一経路へ殺到する現象）を引き起こすリスクも生じる [24].
3. **Braess のパラドックスとダイナミクス**: 第 2.3.4 小節で示した Braess のパラドックスは，静学的に「新規リンク追加後の UE が悪化する」ことを示す．Day-to-day ダイナミクスの観点では，過渡状態でいったん既存リンクの混雑が緩和するように見えながら，最終的に悪化した均衡に落ち着く動的挙動が確認できる [24, 5].

クイズ 9：毎朝の「昨日より速い道」探しは，どこへ収束する？

山手通り周辺に通勤する多数のドライバーが，毎朝「昨日より旅行時間が短い経路があれば少しずつ乗り換える」という Smith ダイナミクス（改訂プロトコル $\rho_{ij} \propto [F_j - F_i]^+$）を繰り返しています．この「毎日の微修正」を長期間繰り返すと，最終的にどの状態に収束するでしょうか？

(ア) ランダムに変動し続け，安定した状態には到達しない

(イ) 利用者均衡（UE）——誰も旅行時間を短縮できる乗り換え先がない状態

(ウ) システム最適（SO）——総旅行時間が最小の状態

(エ) 全員が同じ経路に集中する

→ 正解・解説は節末を参照

本節のまとめ

本節で示した主要な結果を整理する：

1. UE は**ポテンシャルゲーム**の Nash 均衡であり，Beckmann 変換の目的関数 $Z(\boldsymbol{x})$ がポテンシャル関数に対応する（第 2.7.1 小節）.
2. 進化ゲーム理論の**集団ゲーム・改訂プロトコル**の枠組みにより，有限合理性のもとでの Day-to-day 調整過程を ODE として定式化できる（第 2.7.3 小節，第 2.7.4 小節）.
3. Smith ダイナミクス，レプリケーター動学など多様な進化動学が正相関条件等を満たしており，これらは**交通量配分の Day-to-day モデル**として利用できる（第 2.7.5 小節）.
4. 正相関条件を満たす進化動学はポテンシャル関数 Z を Lyapunov 関数として Nash 均衡（UE）に収束する——完全合理性を仮定しなくても長期均衡は静学的 UE と一致する（第 2.7.6 小節）.
5. Day-to-day 進化動学の観点から，混雑課金・情報提供・ネットワーク設計の**動的評価**（収束速度・過渡状態への影響）が可能となる（第 2.7.7 小節）.

クイズ 9 正解・解説

正解：(イ)

ポテンシャルゲームとしての UE の構造から，Smith ダイナミクスはグローバルに UE に収束することが保証されます．毎朝の「賢い乗り換え」は最終的に，Wardrop が 1952 年に定義した均衡条件を動的に「発見」するプロセスです.

2.8　ピグー課金と混雑課金の一般理論

第 2.5 節では，システム最適配分（SO）と混雑課金の基本的な考え方を導入し，最適課金額 $\tau_a^* = x_a^{SO} t_a'(x_a^{SO})$ という結果を示した（第 2.5.3 小節）．しかし，現実の政策場面では「なぜその課金額が正当化されるのか」という経済学的根拠，「すべてのリンクに同時に課金できない場合はどうなるか」という制度的制約，「時間帯によって需要が異なる場合の動的課金」など，より踏み込んだ議論が必要になる．

本節は第 2.5 節を発展させ，混雑課金の理論を体系的に整理する．第 2.8.1 小節 では，外部性という概念からピグー税の経済的根拠を整理する．第 2.8.2 小節 では複数リンクネットワークへの第一最善課金の定式化を行う．第 2.8.3 小節 では課金可能なリンクが限られる第二最善問題を扱う．

第 2.8.4 小節 では，出発時刻選択を扱う Vickrey ボトルネックモデルと最適動的課金を説明する．最後に 第 2.8.5 小節 では，課金収入の再分配と公平性の問題を論じる．

本節で新たに用いる主要記号を表 2.19 にまとめる（基本記号は表 2.1，表 2.10 参照）．

表 2.19: ピグー課金・ボトルネックモデルの主要記号

記号	意味	単位（例）
V	単一リンクの交通量（本節での連続変数表記）	台/時
MEC	限界外部費用 $= V\,t'(V)$（単一リンク表記）	分
$A^T \subseteq A$	課金可能なリンクの集合（第二最善課金で $A^T \subsetneq A$）	—
$\tilde{t}_a$	課金後の一般化リンク費用 $= t_a(x_a) + \tau_a$	分
s	ボトルネックの処理容量（最大サービス率）	台/時
$q(t)$	時刻 t における待ち行列長	台
$w(t) = q(t)/s$	時刻 t における待ち時間	時
N	通勤者総数	人
t^*	希望到着時刻（始業時刻）	—
$C(t)$	時刻 t に出発する通勤者の一般化費用（α-β-γ 体系）	円
α	待ち時間の価値（時間費用係数；表 2.1 の α とは異なる）	円/時
β	早着スケジュール遅延費用係数（$\beta < \alpha$）	円/時
γ	遅着スケジュール遅延費用係数（$\gamma > \alpha$）	円/時
$\tau(t)$	時刻 t における動的課金額	円

2.8.1　外部性とピグー税の考え方

混雑の外部性とは何か

ある利用者が混雑したリンクに追加的に入ると，そのリンクを走行している他の利用者全員の旅行時間が増加する．しかし意思決定を行う利用者はこの影響を自己の費用として算入しない．これが**混雑の外部性**（Congestion Externality）である [20].

単一リンク，交通量 V（連続変数），旅行時間関数 $t(V)$ を仮定すると，各費用成分は次のように整理される：

$$\text{私的限界費用 (PMC)} = t(V) \tag{2.83}$$

$$\text{社会的費用の全体} = V \cdot t(V) \tag{2.84}$$

$$\text{社会的限界費用 (SMC)} = \frac{d}{dV}\big[V \cdot t(V)\big] = t(V) + V\,t'(V) \tag{2.85}$$

$$\text{外部費用 (MEC)} = V\,t'(V) = \text{SMC} - \text{PMC} \tag{2.86}$$

外部費用 MEC $= Vt'(V)$ は，1 台が追加参入したことによって既存の V 台すべての旅行時間が $t'(V)$ だけ延びることから生じる．$t(V)$ が単調増加（$t'(V) > 0$）である限り，MEC > 0，すなわち混雑は**負の外部性**をもたらす．

ピグー税による外部性の内部化

Pigou（1920）は，負の外部性が「市場の失敗」の一形態であり，外部費用に等しい税を課すことで個人の意思決定を社会的最適に誘導できるという**ピグー税**（Pigouvian Tax）の概念を提唱した [20].

混雑課金の文脈では，ピグー税率を

$$\tau^* = V^* \cdot t'(V^*) \tag{2.87}$$

と定める（$V^* = V_{SO}$ は社会的最適交通量）．この課金を課すと，利用者が直面する費用は PMC$+\tau^* = t(V^*) + V^* t'(V^*) = \mathrm{SMC}(V^*)$ となり，個人の私的最適化と社会最適が一致する（図 2.9 参照）．

ピグー税の要点

- 課金なし（UE）では PMC のみを参照 $\rightarrow$ SMC $>$ PMC により過剰交通量
- ピグー税 $\tau^* = \mathrm{MEC}(V^*)$ を課すと，私的費用 = SMC
- 利用者の自発的な経路選択が社会最適（SO 解）と一致する
- 課金収入（$= \tau^* \times V^*$）は政府が徴収し，再分配可能（第 2.8.5 小節参照）

2.8.2　最適課金の定式化（第一最善）

第 2.5 節 第 2.5.3 小節 では単一リンクの例でピグー税を導出した．ここでは**多リンクネットワーク**に拡張し，**第一最善**（First-Best）課金の定式化を体系的に示す [25, 31].

第一最善課金の定義

「ネットワーク上のすべてのリンクに，任意の大きさの課金を設定できる」という理想的な条件下での最適課金を**第一最善課金**という．

各リンク $a \in A$ に料金 $\tau_a \geq 0$ を設定するとき，利用者が直面する一般化費用は $\tilde{t}_a(x_a) = t_a(x_a) + \tau_a$ となる．利用者はこの一般化費用に基づいて UE 原則で経路選択するため，均衡リンク交通量 $\boldsymbol{x}(\boldsymbol{\tau})$ は $\boldsymbol{\tau}$ の関数となる．

最適料金の導出

社会的費用を最小化するためには，均衡が SO 解 $\boldsymbol{x}^{SO}$ に対応するよう $\boldsymbol{\tau}$ を選べばよい．第 2.5 節（第 2.5.1 小節，式 (2.50)）で示したとおり，SO-Frank-Wolfe においてリンクコストとして用いるのは社会的限界費用 SMC であった．これを各リンクの課金後費用が SMC に等しくなるよう設定すると：

$$t_a(x_a^{SO}) + \tau_a^* = t_a(x_a^{SO}) + x_a^{SO}\, t_a'(x_a^{SO}) \tag{2.88}$$

よって

$$\boxed{\tau_a^* = x_a^{SO}\, t_a'(x_a^{SO}) \qquad \forall\, a \in A} \tag{2.89}$$

この第一最善料金 τ_a^* をすべてのリンクに設定したとき，修正費用 $\tilde{t}_a$ に基づく UE の解は SO 解と一致する，すなわち $\boldsymbol{x}_{UE}^{\mathrm{toll}} = \boldsymbol{x}^{SO}$[3].

BPR 関数への適用

BPR 関数 $t_a(x_a) = t_{a0}\left\{1 + \alpha(x_a/C_a)^\beta\right\}$（第 2.1.3 小節）に対しては，式 (2.53) から：

$$\tau_a^* = t_{a0}\,\alpha\,\beta\,\frac{(x_a^{SO})^\beta}{C_a^\beta} \tag{2.90}$$

混雑が激しく V/C 比 x_a^{SO}/C_a が大きいリンクほど，第一最善料金が高くなる．$\beta = 4$（通常値）では，V/C が 2 倍になると τ_a^* は $2^4 = 16$ 倍になる．

余剰分析との接続

第 2.5 節 第 2.5.4 小節 で整理したとおり，第一最善課金によって UE の死荷重損失（図 2.9 の紫の三角形）が完全に解消され，社会的余剰が最大化される．課金収入 $R_a = \tau_a^* \cdot x_a^{SO}$ は政府に帰属し，適切に再分配されれば原理的にパレート改善が実現する（第 2.8.5 小節参照）．

2.8.3　第二最善課金とネットワーク制約

第二最善課金の必要性

現実の政策環境では，技術的・政治的制約から**すべてのリンクに同時に課金することは困難**である：高速道路には料金所を設けられるが，一般道への課金は難しい，あるいは都心に入る一部コリドーのみ課金する「コードン型課金」（都市の周囲に料金関門を設ける方式）を採用する，などの状況がある [31].

課金可能なリンクの集合を $A^T \subseteq A$（$|A^T| < |A|$），課金できないリンクの集合を $A \setminus A^T$ とする．このとき，A^T のみに課金しての最適料金設計問題を**第二最善課金**という．

問題の複雑さ：バイレベル計画問題

第二最善課金の導出は，第一最善とは構造的に異なる難しさを持つ．一部のリンクのみに課金すると，利用者は課金リンクを回避し，未課金リンクへ迂回するため，かえって社会的損失が増大しうる [25].

最適な第二最善料金 $\boldsymbol{\tau}^{A^T}$ を求めるには，以下のような**バイレベル計画問題**（Bilevel Programming Problem）を解く必要がある：

上位問題（プランナー）

$$\min_{\boldsymbol{\tau}^{A^T} \geq \mathbf{0}} \sum_{a \in A} x_a \, t_a(x_a) \qquad \text{（社会的純損失の最小化）} \tag{2.91}$$

下位問題（利用者）

$$\boldsymbol{x} = \boldsymbol{x}_{UE}(\boldsymbol{\tau}^{A^T}) \qquad \text{（課金後の UE 配分）} \tag{2.92}$$

上位問題は社会的費用の最小化，下位問題は利用者の均衡反応を表す．両者が入れ子になっているため，一般に非凸最適化問題となり，大域的な最適解の探索が困難である [25].

第二最善課金の特性

- **未課金リンクへの転換効果**：課金リンクの料金を上げると，利用者が未課金リンクへ移動し，未課金リンクの混雑が増大する．二次的な外部費用が生じるため，最適な第二最善料金は単純な限界外部費用 $Vt'(V)$ より小さくなることが多い．

- **パレート改善の非保証**：第一最善課金ではパレート改善（収入再分配で全員改善）が原理的に可能だが，第二最善課金では課金収入が少額になるうえ，迂回交通量の増大により一部の利用者の厚生が低下しうる．

- **実用的アプローチ**：コードン型課金（ロンドン 2003 年〜，ストックホルム 2006 年〜，シンガポール 1998 年〜 ERP）では，都心部への流入リンクのみに課金し，外部性全体を近似的に内部化するとともに政治的受容性を確保している [29].

クイズ 10：首都高だけに課金したら，環七はどうなる？

首都高速と並走する環状七号線（環七通り）は無料です．仮に首都高にのみ最適課金 $\tau^* = 1$（前章参照）を設定した場合，次のうち最も起きやすいのはどれでしょうか？

(ア) 首都高の混雑が解消され，環七も含めたネットワーク全体が改善される

(イ) 課金を嫌ったドライバーが環七に流れ込み，環七の混雑が増し，課金による社会的改善効果が大きく削がれる

(ウ) ドライバーは料金差より利便性を重視するため，流出はほとんど起きない

→ 正解・解説は節末を参照

2.8.4　Vickrey ボトルネックモデルと動的課金

これまでの議論は**静的モデル**（時間を固定したスナップショット）を前提としていたが，現実のラッシュ時交通需要は時間帯によって大きく変動し，ボトルネックでは行列（キュー）が時系列で形成・消滅する**動的プロセス**が重要となる．Vickrey（1969）の**ボトルネックモデル**は，この動的側面を分析するための代表的な枠組みである [29].

モデルの設定

ボトルネック． 固定容量 s（台/時）の単一ボトルネック（例：市内への入口に立地する橋）を考える．時刻 t における交通需要が容量 s を超えると，行列（待ち行列）$q(t)$（台）が形成される．単位時間あたりの待ち時間は $w(t) = q(t)/s$ である．

希望到着時刻と通勤費用． N（人）の同質な通勤者全員が，職場の始業時刻 t^*（例：午前 9 時）をちょうど “希望到着時刻” とする．通勤者の一般化費用は α-β-γ **体系** [26, 2]:

$$C(t) = \alpha\, w(t) + \beta\, \max(t^* - t_{\text{arr}}, 0) + \gamma\, \max(t_{\text{arr}} - t^*, 0) \tag{2.93}$$

- $\alpha > 0$：行列での待ち時間（旅行時間）の価値（時間費用，[円/時]）
- $\beta > 0$：希望到着時刻より**早着**した場合のスケジュール遅延費用 ($\beta < \alpha$)
- $\gamma > 0$：希望到着時刻より**遅着**した場合のスケジュール遅延費用 ($\gamma > \alpha$)
- t_{arr}：実際の職場到着時刻

$\gamma > \alpha > \beta$ の仮定は，「遅刻は待ち時間より嫌」「しかし早着より遅着のほうが嫌」という一般的な実証結果に整合している．

無課金均衡（Nash 出発時刻均衡）

課金なしの状態では，通勤者は一般化費用が最小となる出発時刻を選ぶ．均衡では，ラッシュ時間帯（$[t_0, t_1]$）に出発するすべての通勤者の費用が等しくなる：

$$C(t) = C^* = \text{const.} \qquad \forall\, t \in [t_0, t_1] \tag{2.94}$$

この均衡では：

- t^* より**前**に出発するほど行列が長く（待ち時間が大），到着が早い（早着費用が小）；費用均等条件 $\alpha\, w(t) + \beta(t^* - t_{\text{arr}}) = C^*$ が成立．
- t^* より**後**に出発するほど行列が短く，到着が遅い（遅着費用が大）；費用均等条件 $\alpha\, w(t) + \gamma(t_{\text{arr}} - t^*) = C^*$ が成立．
- 結果として行列（キュー）が形成され，待ち時間 $w(t) > 0$ が発生する．この待ち時間は**純粋な社会的損失**（旅行時間が浪費され，誰も便益を得ない）である．

最適動的課金

最適動的課金では，各時刻 t における課金額 $\tau(t)$ を設定して，行列（キュー）を完全に消滅させることを目標とする：

$$w(t) = 0 \quad \forall\, t \in [t_0, t_1] \tag{2.95}$$

行列 $q(t) = 0$ を実現するには，ボトルネックへの到着率がつねに容量 s に等しくなるよう出発時刻の分布を誘導する必要がある．このとき，費用均等条件から最適料金プロファイルが決まる：

$$\tau(t) = \begin{cases} C^* - \beta\,(t^* - t_{\text{arr}}) & \text{早着側 } (t \le t^*) \\ C^* - \gamma\,(t_{\text{arr}} - t^*) & \text{遅着側 } (t \ge t^*) \end{cases} \tag{2.96}$$

料金プロファイルは「早着側は増加，遅着側は減少」という**山型**（逆 V 字型）の形状となる（図 2.15）．

最適課金の嬉しい性質

1. **キュー消滅**：最適課金によって $w(t) = 0$ が実現し，行列による時間浪費という社会的損失が完全に解消される．通勤者の到着時刻パターンは課金前と同一であり，スケジュール遅延費用は不変のままである（費用均等条件から明らか）．
2. **福祉改善の規模**：最適課金で解消できる社会的損失の割合は最大 50%（ケースによる）であることが Arnott, de Palma and Lindsey（1990）によって示されている [2]．正確には，福祉改善率は β/α と γ/α の比率に依存し，$\beta \to 0$ かつ $\gamma \to \infty$ の極限で改善率が最大になる．
3. **課金収入**：$\tau(t) \cdot s\, dt$ を積分した課金総収入は，課金前に発生していた行列での待ち時間費用の合計に等しく，利用者へ還付することで厚生損失なしに同一の効用水準を達成できる．

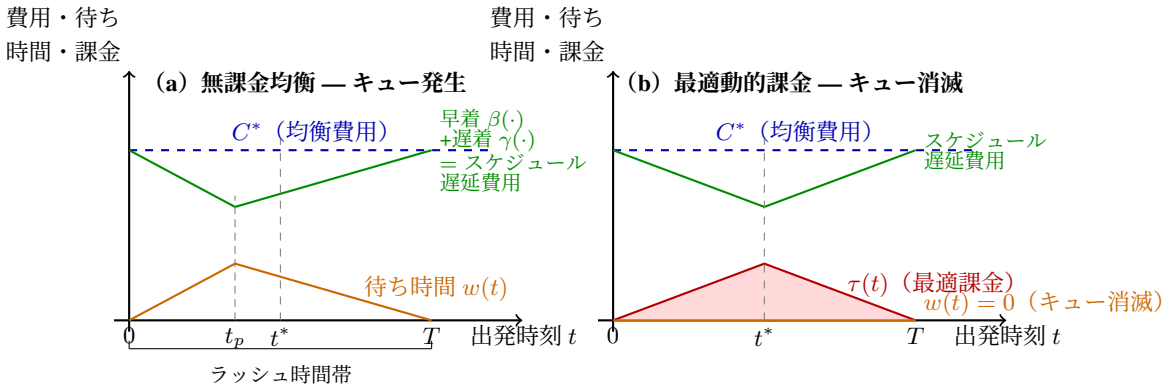


図 2.15: Vickrey ボトルネックモデルにおける出発時刻選択．縦軸は費用・待ち時間・課金額，横軸は出発時刻 t，t^* は希望到着時刻．t_p は定時到着する最後の出発者の出発時刻（$t_p + w(t_p) = t^*$）であり，待ち時間が正である限り $t_p < t^*$ となる．(a) 無課金均衡：行列 $w(t) > 0$ が形成される（純粋な社会的損失）．(b) 最適動的課金：山型（逆 V 字型）の料金 $\tau(t)$ により行列が消滅（$w(t) = 0$）し，通勤費用は均衡値 C^* のまま変化しない．なお，非混雑時の旅行時間（フリーフロー所要時間）は 0 と仮定している．(b) の赤い三角形の面積は $\int \tau(t)\,dt$ であり，ボトルネック容量 s を乗じた $s\int \tau(t)\,dt$ が総課金収入に等しい．

時間帯別料金との関係

実務では連続的な料金プロファイル $\tau(t)$ の設定は困難なため，時間帯を数個に区切った**時間帯別料金**（Time-of-Day Pricing）で近似される．例えば「ピーク前（やや高）→ ピーク（最高）→ ピーク後（やや高）→ 夜間（無料）」という 3〜4 段階の料金体系が実用的である．シンガポールの ERP（Electronic Road Pricing）は 30 分単位で料金を変動させる実装例として知られている．

2.8.5　収入の再分配と公平性

課金収入の規模

最適課金を実施した場合の収入は実務的に相当な規模になりうる．

- ロンドン中心部課金（2003 年導入）：年間約 2.3 億ポンドの総収入（純収入は約 1.5 億ポンド，2018/19 年度）
- ストックホルム中心部課金（2006 年導入，2007 年恒久化）：年間約 15 億クローナ（2017 年度，2016 年の税率引上げ後）
- シンガポール ERP：道路整備・公共交通補助の主要財源の一つ（収入額は詳細非公表）

収入の使途（再分配方式）

課金収入の使途によって，社会的公正（equity）への影響が大きく異なる（表 2.20）．

表 2.20: 課金収入の主な使途と公平性への影響

方式	内容	公平性への影響
一律還付（lump-sum）	全市民に等額還付（「交通券」「現金還付」など）	低所得者は支払い分よりも受取分が相対的に大きい場合があり，逆進性の緩和につながりうる
公共交通投資	鉄道・バスの改善，自転車インフラ整備に充当	自動車を持たない低所得層・高齢者の移動手段を拡充
既存税の減税	所得税・法人税などの歪んだ税を引き下げ	**二重配当効果**（double dividend）：環境・混雑効率 + 租税効率の同時改善
燃料税の代替	燃料税廃止と引き換えに混雑課金を導入	既存負担との整合性が政治的受容性を向上させる

公平性の懸念

混雑課金には逆進性の懸念がある：低所得者ほど収入に対する課金額の割合が大きく（**逆進的**），かつ「通勤のために車を使わざるを得ない」層に負担が集中しやすい [20].

しかし次の点に留意が必要である：

- **パレート改善の可能性**：課金収入を適切に再分配すれば，理論的には全員の厚生が（少なくとも）課金前と同水準以上に改善できる．すなわち**パレート改善**が可能である．混雑課金を「運転者への負担増」としてではなく，「交通システム全体の効率化で生じた余剰の再分配」として捉え直すことが重要である．
- **車を持たない市民への便益**：課金によって交通量が減少し，バスの定時性や自転車の安全性が向上するため，自動車非保有者（低所得層・高齢者に多い）に対しても直接的便益がある．
- **政治経済学的側面**：「混雑課金」という表現への心理的抵抗から，「快適ドライブ料金」「交通スムーズ料金」などの呼称変更やパイロット実験段階的導入が，政治的受容性を高める効果があることがストックホルムの事例で実証されている [29].

二重配当効果

課金収入を既存の歪んだ税（所得税, 法人税）の引き下げに充当すると，**二重配当**（double dividend）が生じる [20]：

1. **第一配当**：混雑外部性の内部化 → 交通効率の改善（死荷重損失の解消）
2. **第二配当**：租税の歪み（超過負担）の縮小 → 経済全体の効率改善

両配当が同時に実現するため，混雑課金はただの「税収確保策」ではなく，「経済政策」として正当化できるという論拠になる．

本節のまとめ

本節で示した主要な結果を整理する：

1. **外部性の内部化**：ピグー税 $\tau^* = V^* t'(V^*)$ で私的限界費用（PMC）を社会的限界費用（SMC）に引き上げることで，個人の私的最適化が社会最適と一致する（第 2.8.1 小節）．

2. **第一最善課金**：全リンクに $\tau_a^* = x_a^{SO} t'_a(x_a^{SO})$ を設定することで UE 解が SO 解に一致し，死荷重損失が解消される（第 2.8.2 小節）．
3. **第二最善課金**：課金可能リンクが限定される場合，バイレベル計画問題を解く必要があり，迂回交通への対応を含む単純な限界外部費用課金では不十分である（第 2.8.3 小節）．
4. **動的課金**：Vickrey ボトルネックモデルにおける山型（逆 V 字型）料金プロファイルにより行列（キュー）が消滅し，最大 50% の社会損失削減が可能である（第 2.8.4 小節）．
5. **収入の再分配**：適切な再分配によりパレート改善が可能であり，二重配当効果によって混雑課金を経済政策として正当化する論拠も生まれる（第 2.8.5 小節）．

クイズ 10 正解・解説

正解：(イ)
これが第二最善課金の本質——課金可能リンクが $A^T \subsetneq A$ に限定されると，課金外リンクへの交通転換が新たな外部性を生みます．「首都高は有料・一般道は無料」という現実の制度は，まさにこの第二最善問題を毎日実演しています．

2.9　シミュレーション

前節（第 2.8 節）まで，交通ネットワークの均衡・最適化・課金理論を解析的・数値的手法で扱ってきた．これらの理論モデルは均衡条件や最適解を明示的に与え，解釈性に優れる反面，大規模ネットワーク上での動的な交通変動，信号制御・情報提供といった施策の時空間的影響，あるいは個々の車両挙動をきめ細かく評価するためには，**シミュレーション**（Simulation）が不可欠である．

交通シミュレーションは，その**空間分解能**と**計算粒度**に応じて，大きく 2 種類に区別される（表 2.21）：

- **マクロシミュレーション**：交通流を時間・空間で集計した量（密度，流量，速度）で記述する．計算コストが低く，都市・地域スケールへの応用が可能である．
- **ミクロシミュレーション**：個々の車両を独立したエージェントとして逐次シミュレートする．信号制御や車線変更など細粒度の現象を詳細に再現できるが，計算コストが高い．

本節では，第 2.9.1 小節 でマクロシミュレーションの中核をなす**マクロ基本ダイアグラム**（MFD）とゾーンベースモデルを，第 2.9.2 小節 でミクロシミュレーションの基礎となる**車両追従モデル**（GHR モデル）をそれぞれ解説する．

本節で用いる主要記号を表 2.22 にまとめる．

2.9.1　マクロシミュレーション（MFD）

マクロシミュレーションでは，交通流を個々の車両ではなく**連続体**として捉え，密度・流量・速度などの集計量によってその挙動を記述する．本節では，この考え方の理論的背景から始め，都市ネットワーク規模で成立することが実証されている**マクロ基本ダイアグラム**（Macroscopic Fundamental

表 2.21: マクロシミュレーションとミクロシミュレーションの比較

項目	マクロシミュレーション	ミクロシミュレーション
記述対象	集計交通量（密度・流量）	個々の車両
状態変数	$n_i(t)$, $G_i(n_i)$	各車両の位置・速度
計算粒度	ゾーン × 時間ステップ	車両 × タイムステップ（秒）
計算コスト	低	高
空間詳細度	低	高
時間詳細度	低	高
適用スケール	都市・地域レベル	交差点・区間レベル
代表手法	ゾーンベースモデル（MFD）	車両追従モデル（GHR 等）

表 2.22: シミュレーションの主要記号

記号	意味	単位（例）
マクロシミュレーション（*LWR* モデル・*MFD*）		
$\rho(x,t)$	交通流密度	台/km
$q(x,t)$	交通流量（LWR モデル）	台/時
$Q(\rho)$	基本図（密度から流量を決める関数）	台/時
n_i	ゾーン i の Accumulation（ゾーン内車両台数）	台
$G_i(n_i)$	ゾーン i の Production（MFD；単位時間の走行距離総和）	台 ·km/時
$\bar{v}_i$	ゾーン i の平均速度 $= G_i(n_i)/n_i$	km/時
n_i^*	臨界 Accumulation（Production が最大となる車両台数）	台
ϕ_{ij}	ゾーン i からゾーン j への方向別流動シェア	—
ミクロシミュレーション（*GHR* 車両追従モデル）		
$v_i(t)$	時刻 t における車両 i（後続車）の速度	km/時
Δx_i	車両 i と前方車両の車間距離	m
Δv_i	車両 i と前方車両の相対速度	km/時
Δt	ドライバーの反応遅れ時間（GHR モデル）	秒

Diagram, **MFD**）の概念と，それを活用したゾーンベースシミュレーションの定式化を解説する [6, 13].

マクロ交通流の理論的背景：LWR モデル

マクロシミュレーションの理論的礎石は，Lighthill, Whitham (1955) および Richards (1956) によって同時期に独立に提案された **LWR（Lighthill-Whitham-Richards）モデル**に遡る [15, 22]．LWR モデルは，リンク上の交通流を 2 式で記述する：

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0 \quad \text{（車両保存則）} \tag{2.97}$$

$$q = Q(\rho) \quad \text{（基本図：閉合仮定）} \tag{2.98}$$

ここで $\rho(x,t)$ は密度 [台/km], $q(x,t)$ は流量 [台/h], $Q(\rho)$ は密度から流量を決定する単峰型関数（**基本図**, Fundamental Diagram）である．式 (2.97) と (2.98) を組み合わせると $\partial\rho/\partial t + Q'(\rho)\,\partial\rho/\partial x = 0$ という一階双曲型 PDE が得られ，渋滞波（ショック波）が Rankine-Hugoniot 条件 $v_s = \Delta q/\Delta\rho$ に従って後方へ伝播する現象を記述できる．

LWR モデルはリンクスケールの連続体描写であるが，これをネットワーク全体に拡張・集約した概念として MFD が位置づけられる．

MFD の定義と実証

マクロ基本ダイアグラム（MFD）は，ゾーン内の交通状態を次の 2 つの集計量で表す [6]：

Accumulation n_i **[台]** ゾーン i 内に現在存在する車両総数．$n_i = \bar{\rho}_i \cdot L_i$（平均密度 $\bar{\rho}_i$ とネットワーク総延長 L_i の積）．

Production $G_i(n_i)$ **[台・km/h]** ゾーン i 内で単位時間内に車両が走行する総距離．$G_i(n_i) = \bar{q}_i \cdot L_i$（平均流量 $\bar{q}_i$ とネットワーク総延長の積）．また, 平均ネットワーク速度 $\bar{v}_i$ は：$\bar{v}_i = G_i(n_i)/n_i$.

MFD は，Accumulation n_i を横軸，Production $G_i(n_i)$ を縦軸にとったとき，**単峰型**（unimodal）の関係として現れる（図 2.16）．臨界集積台数 n_i^* でスループットは最大値 $G_{\max}$ に達し，これを超えると混雑が広がりスループットが低下する．原点から MFD 上の各点への直線の傾きは平均ネットワーク速度 $\bar{v}_i \propto G_i(n_i)/n_i$ を表す．

MFD の**実証的存在**は Geroliminis & Daganzo (2008) によって確認された [13]．横浜市中心部（2003 年）のループ検知器と GPS プローブデータを用いた実測実験で，都市ネットワーク規模で安定した単峰型の MFD が存在することが示された．ただし，混雑が空間的に均一に分布しているほど MFD のバラつきは小さく安定し，局所的な混雑集中が生じると MFD は不安定になる（ヒステリシスや散乱が増大する）ことも同研究で指摘されている．

ゾーンベースシミュレーション

MFD を利用したゾーンベースシミュレーションは，ネットワークを複数の**ゾーン**（同質性を持つ部分ネットワーク）に分割し，ゾーン i の車両台数（Accumulation）$n_i(t)$ の時間発展を次の**離散時間方程式**で記述する [6]：

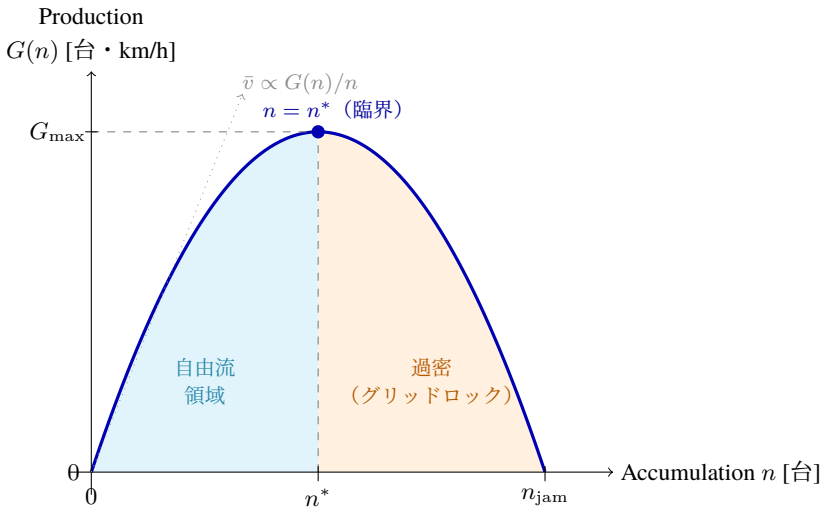


図 2.16: マクロ基本ダイアグラム（MFD）の典型的形状．横軸は Accumulation n，縦軸は Production $G(n)$．臨界集積 n^* でスループットが最大となり，これを超えると過密（グリッドロック）領域に入る．

$$n_i(t+1) = n_i(t) - n_{i,\text{out}}(t) + n_{i,\text{in}}(t) \tag{2.99}$$

ここで $n_{i,\text{out}}(t)$ はゾーン i からの総流出台数，$n_{i,\text{in}}(t)$ はゾーン i への総流入台数である．式 (2.99) は，Daganzo (2007) の連続時間モデル $\dot{n} = q_{\text{in}} - G(n)$ を 1 ステップ時間単位で離散化したものに対応する．

ゾーン i からゾーン j へのリンク流量は，ゾーン間の方向別流動シェア ϕ_{ij}（$\sum_{j \neq i} \phi_{ij} = 1$，OD パターンなどから与えられる定数）と MFD 流出関数 G_i を用いて次のように定式化される：

$$n_{ij,\text{out}}(t) = \phi_{ij} \cdot G_i\big(n_i(t)\big) \tag{2.100}$$

したがってゾーン i への総流入は $n_{i,\text{in}}(t) = \sum_{j \neq i} n_{ji,\text{out}}(t) = \sum_{j \neq i} \phi_{ji}\, G_j(n_j(t))$ となり，式 (2.99)–(2.100) を全ゾーンに対して連立することでネットワーク全体の動的状態が記述される．

応用：ペリメーターコントロール

MFD に基づく代表的な制御手法として**ペリメーターコントロール**（Perimeter Control）がある [6]．これは，都市中心部（対象ゾーン）の境界での流入量 q_{in} を信号制御や IC 制御によって制限し，ゾーン内の Accumulation $n_i(t)$ を臨界値 n_i^* 以下に保つ戦略である．n_i が n_i^* を超えると $G_i(n_i)$ が低下し，最悪の場合ネットワーク全体が機能停止するグリッドロック状態に陥るため，事前に流入制御することで G_i を効率的な領域（$n_i \leq n_i^*$）に維持できる．

更に，式 (2.99)(2.100) を最適制御問題（$n_i(t) \to n_i^*$ を目標とするフィードバック制御など）と統合することで，リアルタイム交通管理システムへの応用が可能である．

計算効率上の利点

ゾーンベースシミュレーションの最大の利点は，リンクレベルの詳細シミュレーションと比較した**低い計算コスト**である．ゾーン数 $|\mathcal{I}|$ の連立 ODE を解くだけで都市・地域スケールの動的交通

状態を推定できるため，最適制御などとのリアルタイム統合にも適している．一方，ゾーン内の空間均質性が損なわれると MFD が安定しないという制約もあり，ゾーン分割の適切な設計が性能に影響する．

2.9.2　ミクロシミュレーション

ミクロシミュレーションでは，ネットワーク上を走行する**個々の車両**を明示的に表現し，各タイムステップ（$\Delta t \approx 0.1$〜1 秒）ごとに全車両の挙動を逐次計算する**エージェントベース**の手法をとる．マクロシミュレーションが集計量の時間変化を追うのに対し，ミクロシミュレーションは個別車両の位置と速度を直接追跡する点で根本的に異なる．

エージェントベースシミュレーションの設計

典型的なミクロシミュレーターは，1 タイムステップにつき以下の処理を全車両に対して繰り返す：

1. **経路選択**：各車両は交通情報や自身の学習に基づいて，目的地への（暫定的な）経路を選択または更新する（均衡配分理論（第 2.3 節）を背景に もつステップ）．
2. **車両挙動計算**：前方車両との距離・相対速度などをもとに，車両追従モデル（本節後述）や車線変更モデルを用いて加速度 $\dot{v}_i$ を決定する．
3. **状態更新**：$v_i(t+\Delta t) = v_i(t) + \dot{v}_i \cdot \Delta t$，$x_i(t+\Delta t) = x_i(t) + v_i(t)\cdot\Delta t$ により速度・位置を更新する．

この繰り返しによって，信号での停止・発進，車線変更，合流・分岐など，マクロモデルでは平均化されて消えてしまう細粒度の交通流が再現される．

車両追従モデル（GHR モデル）

車両追従モデル（Car-Following Model）は，後続車が前方車の挙動（速度・距離）に反応して加速・減速する様子を記述する微分方程式系であり，ミクロシミュレーションの中核をなす．

代表的モデルの一つが Gazis, Herman & Rothery (1961) による **GHR モデル**（General car-Following / Gazis-Herman-Rothery Model）である [12]．GHR モデルは「後続車の加速度は，前方車との**相対速度**（刺激）に感応係数を乗じた形で決まる」という**刺激-反応フレームワーク**に基づく（図 2.17）：

$$\dot{v}_i(t) = \alpha \, \frac{v_i^{\beta}(t)}{\Delta x_i^{\gamma}(t-\Delta t)} \, \Delta v_i(t-\Delta t) \tag{2.101}$$

式 (2.101) の各記号の意味は以下のとおりである：

- $v_i(t)$：時刻 t における車両 i（後続車）の速度 [km/h].
- $\Delta x_i(t-\Delta t)$：時刻 $t-\Delta t$（反応遅れ前）における車両 i と前方車両 $i-1$ の**車間距離** [m].
- $\Delta v_i(t-\Delta t) = v_{i-1}(t-\Delta t) - v_i(t-\Delta t)$：時刻 $t-\Delta t$ における**相対速度** [km/h]（正値 = 前方車がより速い，負値 = 前方車がより遅い）．
- Δt：ドライバーの**反応遅れ時間** [s].

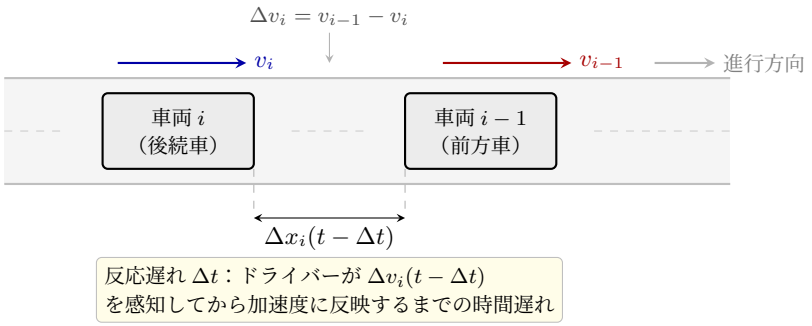


図 2.17: 車両追従モデル（GHR モデル）の概念図．後続車（車両 i）は，前方車（車両 $i-1$）との車間距離 Δx_i および相対速度 $\Delta v_i = v_{i-1} - v_i$ を反応遅れ Δt ののち感知し，加速度を調整する．

- α：感応係数（感度のスケールを決定）．
- β, γ：速度・車間距離に対する感応の非線形性を表す指数パラメータ．

式 (2.101) の直感的解釈：前方車が速い（$\Delta v_i > 0$）ならば $\dot{v}_i > 0$（加速），前方車が遅い（$\Delta v_i < 0$）ならば $\dot{v}_i < 0$（減速）という自然な挙動を表現する．$v_i^{\beta}/\Delta x_i^{\gamma}$ 項は，「高速領域では感応が高まる，車間距離が近いほど感応が高まる」という非線形な感応特性を規定する．

パラメータと特殊ケース

GHR モデルはパラメータ組 (α, β, γ) によって複数の既存モデルを統合する汎化枠組みである [12]：

- $(\beta = 0,\ \gamma = 1)$：Greenshields の線形速度-密度モデル $v = v_f(1 - k/k_j)$ に対応する巨視的基本図が導出される．
- $(\beta = 0,\ \gamma = 2)$：Greenberg の対数速度-密度モデル $v = c\ln(k_j/k)$ に対応する巨視的基本図が導出される．

また，GHR モデルを多数の車両に対して集計すると，マクロスケールの速度-密度関係（基本図）が導出されることが示されており [12]，ミクロ記述とマクロ記述（第 2.9.1 小節）を架橋する理論的役割ももつ．現地データ（Lincoln Tunnel・Holland Tunnel の実測データ）を用いたパラメータ推定によって (α, β, γ) の値域が実証的に検証されている [12]．

代表的なシミュレーションソフトウェア

現在，学術・実務での利用実績がある代表的なミクロシミュレーターとして次のものが挙げられる：

MATSim ETH チューリッヒおよびベルリン工科大学が中心となって開発するオープンソースの大規模エージェントベース交通シミュレーター（Multi-Agent Transport Simulation）．都市・地域スケールで数百万エージェントを扱うことができ，経路選択・様式選択・活動スケジューリングの繰り返し最適化（Day-to-day 調整）に対応する．

SUMO ドイツ航空宇宙センター（DLR）が開発するオープンソースのミクロシミュレーター（Simulation of Urban MObility）．自動車・公共交通・歩行者・自転車を含むマルチモーダルシミュレーションが可能で，TraCI（Traffic Control Interface）を通じた外部プログラムとのリアルタイム連携（信号制御プログラムや強化学習エージェントとの統合など）に広く利用される．

HONGO 東京大学で独自に開発された交通ミクロシミュレーター．国内の道路構造・交通規則・運転行動特性を反映したキャリブレーションが施されており，日本の都市交通に特有の課題（密集交差点，路地交通，混在交通など）のシミュレーション研究に活用されている．

利点と課題

マクロシミュレーション（第 2.9.1 小節）と比較したミクロシミュレーションの**主な利点**は，以下のとおりである：

- 信号制御戦略の精密な評価（ステップレベルの車両通過・停止の再現）が可能．
- 車線規制，インフラ改良（料金ゲート設置，交差点形状変更等）の局所的な影響を詳細に評価できる．
- 自動運転・コネクテッドカー（ACC，CACC など）の導入効果や混合交通下の挙動分析に適している．

一方，**主な課題**は以下のとおりである：

- **計算コストの高さ**：車両数・タイムステップ数に比例して計算量が増大するため，大規模ネットワーク全体への適用は困難なことが多い．マクロシミュレーションとの**ハイブリッド手法**（主要区間のみミクロ，残余はマクロ）も研究・実用化されている．
- **パラメータキャリブレーションの困難さ**：GHR モデルのパラメータ (α, β, γ) など，多数のパラメータを対象ネットワークの実測データにより推定・検証するキャリブレーション作業が必要であり，工数・データ収集コストが大きい．
- **モデルの仮定と現実の乖離**：車両追従モデルは追従時の挙動を記述するが，現実のドライバーの異質性（高齢者・攻撃的ドライバー等）や緊急回避挙動などを完全には再現できない．

マクロシミュレーションとミクロシミュレーションはいずれも一方が他方を代替するものではなく，分析目的・スケール・利用可能データに応じて使い分けるべき相補的な手法である．均衡配分理論（第 2.3 節〜第 2.5 節）・進化ダイナミクス（第 2.7 節）・混雑課金（第 2.8 節）といった本テキストの理論的成果を現実のシステムで検証・活用する際に，シミュレーションは不可欠なブリッジとして機能する．

本節のまとめ

本節で示した主要な結果を整理する：

1. **マクロシミュレーション（MFD）**は，LWR モデルを理論的基礎として交通流を密度・流量の集計量で記述し，都市ネットワーク規模の MFD を利用したゾーンベースモデルによって大規模シミュレーションとペリメーターコントロールなどの交通管理政策評価が可能になる（第 2.9.1 小節）．
2. **ミクロシミュレーション**は，GHR モデルなどの車両追従モデルを用いて個々の車両の加速・減速挙動をエージェントとして逐次シミュレートし，信号・交差点制御や自動運転などの細粒度の交通現象を詳細に評価できる；一方，計算コストの高さやパラメータキャリブレーションの工数が課題である（第 2.9.2 小節）．
3. マクロ・ミクロの 2 手法は相補的であり，分析目的・スケール・利用可能データに応じて使い分けることが求められる；均衡配分理論（第 2.3 節〜第 2.5 節）・進化ダイナミクス（第 2.7 節）・混雑課金（第 2.8 節）といった理論的成果を現実のシステムで検証・活用する際の不可欠なブリッジとして機能する．

コラム：東京都心部の道路網における second-best pricing – 鷲野萌花

東京都心部（山手線の内側）の道路ネットワークを対象に，以下の分析を行った．
(1) Day-to-day ダイナミクスシミュレーションに基づき，収束後の UE 状態における総旅行時間，リンクごとの旅行時間および交通量を算定した．
(2) 総旅行時間の短縮（SO 配分の目的関数）に基づいて Pigouvian 課金を導出し，その課金を導入した状態で Day-to-day ダイナミクスシミュレーションを行った場合，総旅行時間が SO 配分と一致することを確認した．
(3) 課金可能リンクに限定して Pigouvian 課金を導入した場合について，総旅行時間や社会的余剰の変化を分析した．
なお，使用したデータは山手線内側周辺の交通ネットワークデータ（幹線道路）と 2019 年 PT 調査を基にした OD 交通量データである．PT 調査は標本調査であるため，PT によって得られた OD 需要は 300 倍している．どの地域からの出発が多いか図示してみたところ，文京区のあたりが最も多くなっていた．

図 2.18: 発地分布

(1) Day-to-day ダイナミクスによる UE 状態の算定
今回，day-to-day ダイナミクスは

$$x^{t+1} = (1-\eta)x^t + \eta \cdot \mathrm{AON}(c(x^t) + \tau)$$

という更新則に基づく．これは，前日の経路コストを踏まえて AON 配分が行われるが，実際にその経路に動くのは一部（今回であれば，10%）であるということを示す．
AON 配分は離散的な配分であり，AON 配分されるリンクが変化するたびに総旅行時間は均衡点の付近でジグザグと変化する．
実際にこの更新則のもとで計算を回すと，23100 の総トリップは図のように収束し，総旅行時間は 2150.1078 [h· 台] となる．

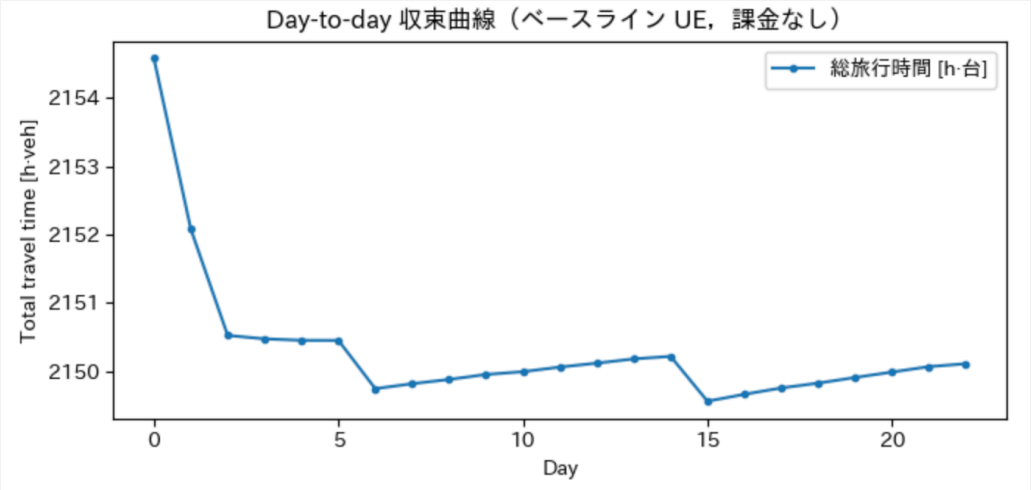


図 2.19: Day-to-day ダイナミクスの収束

また，均衡に達した際の道路の混み具合（赤が混んでいる）を示すマップは以下のようになった．

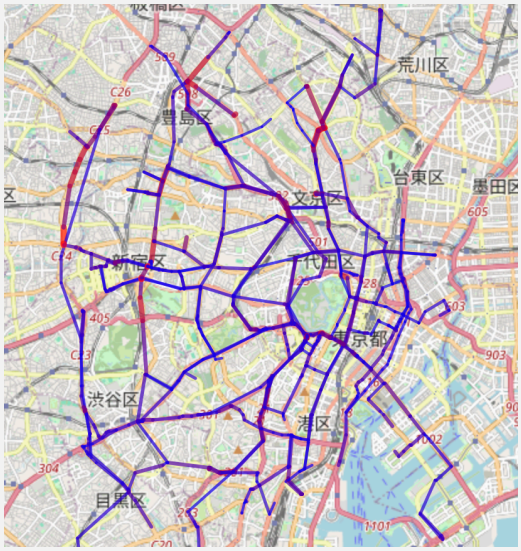


図 2.20: UE 状態におけるリンク混雑度

さらに，最も混雑していた文京区向丘周辺を拡大すると，以下のようになった．

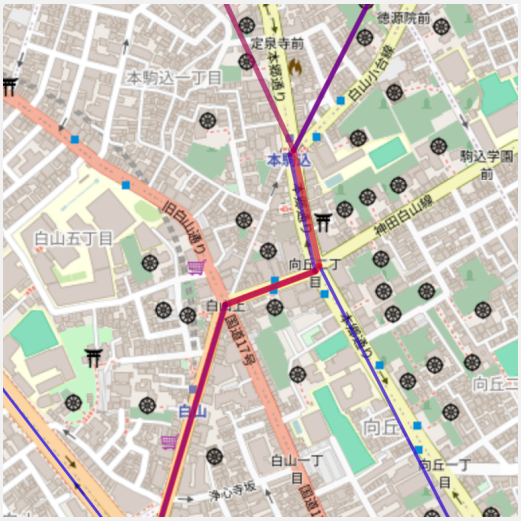


図 2.21: 文京区向丘周辺の混雑状況

混雑しているリンクはどちらかといえば北側に集中しており，この過程のもとで最も混雑しているリンクは文京区向丘周辺となった．
東京都心部の道路ネットワークは千代田区を中心に放射状に広がっており，それらをつなぐ環状線など，複数の道路が入り込む場所の道路容量が少ない場合，混雑する傾向にあり，それが需要が多い北側であるとより顕著になる．

(2) SO 配分と課金状態での Day-to-day ダイナミクス
先ほどとは異なり，総旅行時間の最小化を目的関数にとると，

$$t_a(x_a) + t'_a(x_a) \cdot x_a$$

をリンクコストとして Frank-Wolfe 法を用い均衡を求めることができる．
総旅行時間は

$$T_{\mathrm{SO}} = 2146.8598 \ [\mathrm{h} \cdot \text{台}]$$

となり，その均衡を達成するために必要な課金額の最大値は時間で表すと 0.17 分となる．
リンクごとに適切な課金を施し，day-to-day ダイナミクスを回していくことで SO と同じ配分結果を得ることができる．これはほかでもなく，課金によって外部性が内部化されている事を表す．

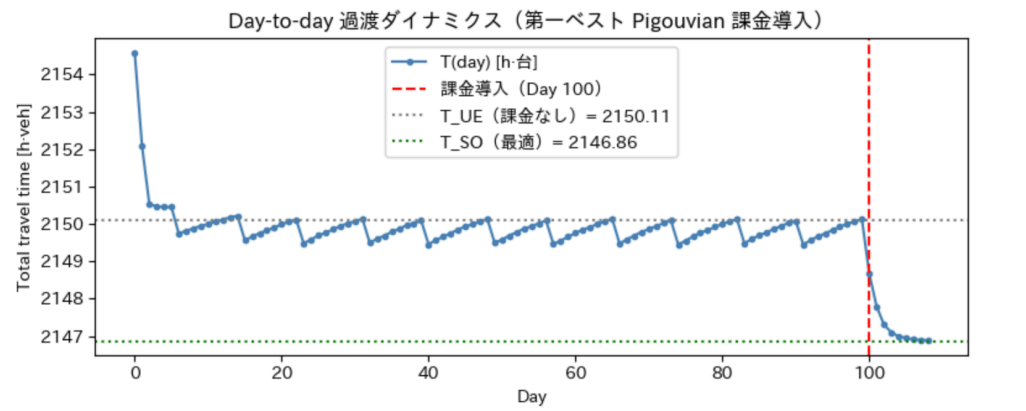


図 2.22: First-best 課金下での Day-to-day ダイナミクス

この仮定のもとで最も課金されたリンクは文京区本駒込ー動坂間のリンクであった．
文京区からの発地が多かったことやこの辺りのリンクの capacity が小さいことなどから，混雑し，かつ重要であるリンクにより多くの課金がされることが確認できる．

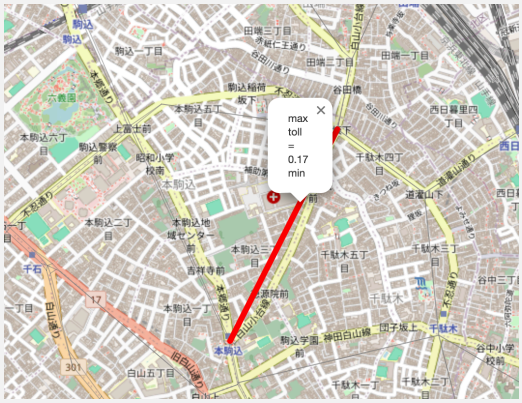


図 2.23: 最大課金リンク周辺

(3) 課金可能リンクへの限定

今回，課金可能なエッジ数は全体のリンクの 28%であった（国道などの幹線道路及び高速道路）．
それらのリンクに先ほど求めた First-best な課金を施すと，総旅行時間は 2150.0788 [h· 台] となり，UE の時とほぼ変わらなくなってしまった．
先ほどのように狭くて需要が集中しているリンクに直接の課金ができないためであると考えられる．

2.10　補足 A　黄金分割法（ステップサイズ探索）

Frank-Wolfe 法（第 2.3 節 第 2.3.3 小節 参照）の各反復では，補助解 $\boldsymbol{y}^{(n)}$ が得られた後，現在の解 $\boldsymbol{x}^{(n)}$ から $\boldsymbol{y}^{(n)}$ 方向への最適なステップサイズ $\alpha^* \in [0,1]$ を 1 次元最適化によって求める：

$$\alpha^* = \underset{0\le\alpha\le 1}{\arg\min}\, Z\big(\boldsymbol{x}^{(n)} + \alpha\,(\boldsymbol{y}^{(n)} - \boldsymbol{x}^{(n)})\big). \tag{2.102}$$

目的関数 Z は BPR 関数のような凸関数の積分であるため，式 (2.102) の右辺は α に関して一変数の凸関数となる．したがって，区間二分法や**黄金分割法**（golden section search）によって高精度かつ効率的に解を求めることができる [25].

2.10.1　黄金分割法のアルゴリズム

準備

黄金分割法は，一変数の**単峰性（unimodal）な関数** $\phi : [a,b] \to \mathbb{R}$ の最小値（または最大値）を求める区間縮小法である．ϕ の評価のみに基づき（微分情報を使わず），各反復で区間長を一定の比率で縮小して解に収束させる．

ステップサイズ探索に適用する場合は，$\phi(\alpha) = Z\big(\boldsymbol{x}^{(n)} + \alpha(\boldsymbol{y}^{(n)} - \boldsymbol{x}^{(n)})\big)$，$[a,b] = [0,1]$ として出発する．

黄金分割比

黄金分割法のアルゴリズムでは，区間 $[a_k, b_k]$ の内側に 2 点

$$g_k = b_k - s\,(b_k - a_k), \tag{2.103}$$

$$h_k = a_k + s\,(b_k - a_k) \tag{2.104}$$

を配置する．ここで s は**黄金分割比**（golden ratio）と呼ばれる定数であり，

$$s = \frac{\sqrt{5}-1}{2} \approx 0.618 \tag{2.105}$$

と定義される．s は $s^2 + s - 1 = 0$ の正の根であり，$s + s^2 = 1$，すなわち $1 - s = s^2 \approx 0.382$ が成り立つ．

この比率を選ぶ動機は，次の反復で区間の内点を 1 点だけ新たに評価すれば，もう 1 点は前の反復から流用できるという**計算効率の最大化**にある．

反復手順

1. **初期化**：$a_0 = 0$，$b_0 = 1$，$\phi(g_0)$，$\phi(h_0)$ を計算する（$k = 0$）．
2. **縮小**：

   - $\phi(g_k) > \phi(h_k)$ の場合： 最小値は $[g_k, b_k]$ にある．$[a_{k+1}, b_{k+1}] \leftarrow [g_k, b_k]$.
     次の反復では $g_{k+1} = h_k$（流用），h_{k+1} のみ新たに計算する．

- $\phi(g_k) < \phi(h_k)$ の場合： 最小値は $[a_k, h_k]$ にある．$[a_{k+1}, b_{k+1}] \leftarrow [a_k, h_k]$.
  次の反復では $h_{k+1} = g_k$（流用），g_{k+1} のみ新たに計算する．
- $\phi(g_k) = \phi(h_k)$ の場合： どちらの縮小を行ってもよい（通常は $[a_k, h_k]$ に縮小）．

3. **収束判定**：$b_{k+1} - a_{k+1} < \epsilon$（許容誤差）であれば終了し，$\alpha^* \approx (a_{k+1} + b_{k+1})/2$ を出力する．
   そうでなければ $k \leftarrow k+1$ として 2. へ戻る．

2.11 補足 B All-or-Nothing 配分の効率的アルゴリズム

Frank-Wolfe 法の各反復（第 2 ステップ）では，現在の旅行時間 $t_a(x_a^{(n)})$ を確定的費用とみなして**All-or-Nothing（AoN）配分**を実行し，補助解 $\boldsymbol{y}^{(n)}$ を求める（第 2.3 節 第 2.3.3 小節 参照）．AoN 配分は「各 OD ペアの全需要を最短経路に集中させる」操作であり，以下で述べる効率的な実装が計算コストを大幅に削減する．

2.11.1 直感的アイデア

ネットワーク上に $|R|$ 個の起点が存在し，各起点 r についてすべての目的地への最短経路を求めた場合，得られる最短経路の全体は**最短経路ツリー（shortest path tree）**と呼ばれる根付き木をなす．

このツリー構造を活用すると，経路を陽に列挙することなく，ツリー上の葉ノードから根ノード（起点 r）方向へ**逆向きに流量を集計**することで，各リンクへの AoN 配分量を効率的に計算できる．

2.11.2 アルゴリズムの詳細

前準備：記号の定義

$c(j)$ 起点 r からノード j への最短経路費用（ダイクストラ法で計算）

$\mathrm{pred}(j)$ ノード j への最短経路上の直前ノード（先行ポインタ）

O_j 最短経路ツリー上でノード j の直接の子ノード集合，すなわち $O_j = \{m \mid \mathrm{pred}(m) = j\}$

q_{rj} 起点 r，目的地 j の OD 交通量

y_{ij} ノード $i \to j$ リンクを経由する AoN 配分量の作業変数

アルゴリズム本体

1. **初期化**：すべてのリンクの補助フローを 0 に設定する：$y_{ij} \leftarrow 0 \ \ (\forall (i,j) \in A)$.
2. **起点ループ**：各起点 $r \in R$ について以下を実行する：
   (a) **最短経路の計算**：ダイクストラ法によって，起点 r から全ノード $j \in N$ への最短経路費用 $c(j)$ と先行ポインタ $\mathrm{pred}(j)$ を求める．
   (b) **ノードの逆順処理**：$c(j)$ の**降順**（目的地 j が起点 r から遠い順）に各ノード j を処理し，リンクフロー $y_{\mathrm{pred}(j),j}$ を次式で更新する：

$$y_{\mathrm{pred}(j),\,j} \;\leftarrow\; y_{\mathrm{pred}(j),\,j} + q_{rj} + \sum_{m \in O_j} y_{jm}. \tag{2.106}$$

式 (2.106) の右辺は，(i) 目的地 j 向けの需要 q_{rj}，および (ii) j の下流ノードからの合計フロー $\sum_{m\in O_j} y_{jm}$，を合算した値である（目的地が j の利用者と，j を通過してさらに下流を目指す利用者の両方をリンク $(\mathrm{pred}(j), j)$ に集計する）．

3. **補助フローの確定**：起点 r のループが終わったら，y_{ij} を All-or-Nothing 補助解 $y_{ij}^{(n)}$ としてリンクフローに加算する：$x_{ij}^{\mathrm{AoN}} \leftarrow x_{ij}^{\mathrm{AoN}} + y_{ij}$ $(\forall (i,j) \in A)$.

計算量

起点ごとのダイクストラ法の計算量は $O(|N|^2)$（優先度付きキューを用いれば $O((|N|+|A|)\log|N|)$）であり，ツリー上の集計は $O(|N|)$ で完了する．これを $|R|$ 起点に対して繰り返すため，全体の計算量は $O(|R|\cdot(|N|+|A|)\log|N|)$ となる．経路を用いて直接配分する単純実装（$O(|K_{rs}|\cdot|A|)$ のオーダーで経路数が指数的に多くなる）と比べて，実用的なネットワーク規模で大幅な高速化が可能である．

2.12　補足 C　利用者均衡と Nash 均衡の等価性

第 2.3 節では「UE 解は交通配分の Nash 均衡に対応する」と述べた．本補足では，まず Wardrop 均衡と Nash 均衡の定義を対比的に整理し（第 2.12.1 小節），次に n 人非協力ゲームの枠組みで等価性を厳密に定式化して命題として示す（第 2.12.3 小節）[25].

2.12.1　定義：Wardrop 均衡と Nash 均衡

交通配分における均衡概念として，以下の 2 つの定義が中心的な役割を果たす．

Wardrop 均衡（連続体モデル）

利用者を**無限小の原子なし連続体**として扱い，1 人の利用者がリンク交通量に及ぼす影響が厳密にゼロと仮定する．

定義（Wardrop 均衡）[30] 交通量配分 $\boldsymbol{f}^*$ が **Wardrop 均衡**（利用者均衡，UE）であるとは，任意の OD ペア $rs \in \Omega$ について以下が成立することをいう：

$$f_k^{rs} > 0 \;\Rightarrow\; c_k^{rs}(\boldsymbol{f}^*) = u_{rs}, \qquad f_k^{rs} = 0 \;\Rightarrow\; c_k^{rs}(\boldsymbol{f}^*) \geq u_{rs} \tag{2.107}$$

ここで $u_{rs} = \min_k c_k^{rs}(\boldsymbol{f}^*)$ は均衡最小経路費用である．すなわち，Wardrop の第 1 原則（第 2.3 節 第 2.3.1 小節参照）そのものである．

各利用者は**自分の選択がリンク費用に影響しない**と仮定して最短経路を選ぶ（価格受容者的行動）．フロー f_k^{rs} は確定量であり，混合戦略の概念を用いずに定式化される．

Nash 均衡（有限 n 人ゲーム）

利用者数 n を有限・明示的に扱い，各利用者の 1 回の経路変更がリンク交通量を $O(1/n)$ だけ変化させると仮定する．

定義（混合戦略 Nash 均衡）[18, 19] n 人非協力ゲームにおいて, 混合戦略プロファイル $(\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_n)$ が **Nash 均衡**であるとは，他のすべてのプレーヤーの戦略を所与として，どの利用者 a も自分の戦略を変更することで**期待旅行時間を改善できない**ことをいう：

$$\pi_{aj} > 0 \;\Rightarrow\; j \in \arg\min_{k} \mathbb{E}\big[g_k(\boldsymbol{h}) \,\big|\, \boldsymbol{\pi}_{-a}\big] \tag{2.108}$$

ここで $\boldsymbol{\pi}_{-a}$ は利用者 a 以外の戦略プロファイルを表す.

Nash 均衡では，各利用者は自分の選択が他者に与える影響（$O(1/n)$）を原理的に認識した上で意思決定する点が Wardrop 均衡と本質的に異なる.

2 定義の対比と連続体極限

	Wardrop 均衡	Nash 均衡（n 人）
プレーヤー数	連続体（無限大）	有限 n
1 人の影響	厳密にゼロ	$O(1/n)$
均衡変数	確定的フロー f_k^{rs}	混合戦略確率 π_{aj}
均衡条件	使用経路の費用が等しい（式 (2.107)）	期待費用最小の経路にのみ正確率（式 (2.108)）

$n \to \infty$ の**連続体極限**では，各プレーヤーの影響が無限小となる．このとき大数の法則より経路交通量 $h_j \approx n\pi_j$ が確定的に定まり，期待費用が確定的費用と一致するため，Nash 均衡条件 (2.108) は Wardrop 均衡条件 (2.107) と等価になる．この等価性を次の第 2.12.3 小節で命題として厳密に証明する.

2.12.2　設定

以下の設定を考える：

ネットワーク　単一の OD ペア (r, s)，経路集合 $K_{rs} = \{1, 2, \dots, J\}$.

利用者　n 人の同質な（homogeneous）通行者が存在し，全員が OD ペア (r, s) を利用する．したがって $q_{rs} = n$ とする.

経路費用　経路 j の費用は経路交通量ベクトル $\boldsymbol{h} = (h_1, \dots, h_J)$ に依存する連続関数 $g_j(\boldsymbol{h})$ で与えられる．ここで $h_j \geq 0$ は経路 j の交通量，$\sum_j h_j = n$ が成立する.

UE 条件　交通量ベクトル $\boldsymbol{h}^*$ が UE であるための必要十分条件は：

$$h_j^* > 0 \;\Rightarrow\; g_j(\boldsymbol{h}^*) = g_{\min}(\boldsymbol{h}^*), \qquad h_j^* = 0 \;\Rightarrow\; g_j(\boldsymbol{h}^*) \geq g_{\min}(\boldsymbol{h}^*) \tag{2.109}$$

ただし $g_{\min}(\boldsymbol{h}^*) = \min_k g_k(\boldsymbol{h}^*)$（均衡最小費用）.

2.12.3　Nash 均衡との等価性（命題 C.1）

命題 C.1　上記の設定のもとで，n **が大のとき確定論的利用者均衡（UE）は** n **人非協力ゲームの（対称）混合戦略 Nash 均衡と等価である.**

証明

ゲームの定式化.　利用者 a ($a = 1, \ldots, n$) が各経路 $j \in K_{rs}$ を選択する確率を π_{aj} とする．利用者の戦略は混合戦略 $(\pi_{a1}, \ldots, \pi_{aJ})$ であり，$\pi_{aj} \geq 0$，$\sum_j \pi_{aj} = 1$ を満たす．

利用者 a が経路 j を選択したときの旅行時間は $g_j(\boldsymbol{h})$ であり，$\boldsymbol{h}$ は全 n 人の戦略の実現値によって決まる確率変数である．n が十分大の極限では，大数の法則より

$$h_j \;\approx\; n\,\pi_j \tag{2.110}$$

が成立する（π_j は全利用者に共通の経路 j の選択確率）．式 (2.110) のもとで，経路 j の期待旅行時間は確定的に $g_j(\boldsymbol{h}^*)$ で近似される（$h_j^* = n\pi_j$）．

Nash 均衡条件.　利用者 a が旅行時間を最小化する合理的エージェントであるとすると，混合戦略 Nash 均衡においては：

$$\pi_j > 0 \;\Rightarrow\; g_j(\boldsymbol{h}^*) = \min_k g_k(\boldsymbol{h}^*), \qquad \pi_j = 0 \;\Rightarrow\; g_j(\boldsymbol{h}^*) \geq \min_k g_k(\boldsymbol{h}^*) \tag{2.111}$$

が成立しなければならない（最適反応条件）．

等価性.　式 (2.110) より $h_j^* = n\pi_j$ であるから，Nash 均衡条件 (2.111) を h_j^* で書き直せば，$h_j^* > 0 \Leftrightarrow \pi_j > 0$ および $h_j^* = 0 \Leftrightarrow \pi_j = 0$ より，UE 条件 (2.109) と完全に一致する．　□

含意

命題 C.1 は，Wardrop が（交通工学の観点から規範的に）定義した UE 条件が，ゲーム理論の意味でも Nash 均衡として正当化できることを示す．すなわち，多数の自律的な利用者がそれぞれ旅行時間を最小化した結果として到達する均衡状態が，まさに Wardrop 均衡に他ならない．

また，UE がポテンシャルゲームの最小化問題（Beckmann 変換）と等価であることは第 2.7 節でも改めて論じる（第 2.7.1 小節参照）．

2.13　補足 D　相補性条件と KKT 条件

本補足では, 不等式制約つき最適化問題の**最適性条件**として広く用いられる **Karush-Kuhn-Tucker（KKT）条件**を概説し，UE/FD-Primal（Beckmann 変換）および SUE/FD-path への適用を通じて，それぞれが Wardrop UE 条件・ロジット選択確率と等価であることを示す.

2.13.1　KKT 条件の一般形

最適化問題の標準形

不等式制約・等式制約を含む一般の最適化問題を以下の標準形で定義する：

$$\min_{\boldsymbol{z}} \; \phi(\boldsymbol{z}) \quad \text{s.t.} \quad g_i(\boldsymbol{z}) \leq 0 \; (i = 1, \ldots, m), \quad h_j(\boldsymbol{z}) = 0 \; (j = 1, \ldots, p). \tag{2.112}$$

ϕ, g_i が凸，h_j が線形のとき，KKT 条件は最適性の**必要十分条件**となる．

ラグランジアン

式 (2.112) に対して**ラグランジアン**を定義する：

$$L(\boldsymbol{z}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \phi(\boldsymbol{z}) + \sum_{i=1}^{m} \lambda_i \, g_i(\boldsymbol{z}) + \sum_{j=1}^{p} \nu_j \, h_j(\boldsymbol{z}). \tag{2.113}$$

ここで λ_i は**不等式制約**に対応する Lagrange 乗数（双対変数），ν_j は**等式制約**に対応する Lagrange 乗数である．

KKT 条件

条件名	式	意味
停留条件（stationarity）	$\nabla_{\boldsymbol{z}} L = \mathbf{0}$	目的関数の勾配が制約関数の勾配の線形結合で表される
主問題実行可能性	$g_i(\boldsymbol{z}) \le 0$，$h_j(\boldsymbol{z}) = 0$	元の制約をすべて満たす
双対問題実行可能性	$\lambda_i \ge 0$	不等式制約の Lagrange 乗数は非負
相補性条件	$\lambda_i \, g_i(\boldsymbol{z}) = 0$	「制約が活性」か「乗数がゼロ」の少なくとも一方が成立

相補性条件の直観

相補性条件 $\lambda_i \, g_i(\boldsymbol{z}) = 0$ は以下のように解釈できる：

- $g_i(\boldsymbol{z}) < 0$（制約が**非活性**，余白がある）ならば $\lambda_i = 0$：この制約は最適解に影響を与えないため，対応する「影の価格」はゼロ．
- $\lambda_i > 0$（最適解がこの制約によって**拘束**されている）ならば $g_i(\boldsymbol{z}) = 0$：制約は必ず等号で成立（活性制約）．

2.13.2　UE/FD-Primal への適用（Wardrop 条件との等価性）

問題の再掲

UE/FD-Primal（式 (2.22)）を再掲する：

$$\min_{\boldsymbol{f}} \; Z(\boldsymbol{f}) = \sum_{a \in A} \int_0^{x_a(\boldsymbol{f})} t_a(w) \, dw \tag{2.114}$$

$$\text{s.t.} \quad \sum_{k \in K_{rs}} f_k^{rs} = q_{rs} \;\; (\forall rs), \quad f_k^{rs} \ge 0 \;\; (\forall k, \forall rs), \quad x_a = \sum_{rs} \sum_{k} f_k^{rs} \, \delta_{a,k}^{rs}.$$

等式制約の Lagrange 乗数を u_{rs}（OD ペア rs ごと），非負制約 $-f_k^{rs} \le 0$ の乗数を $\mu_k^{rs} \ge 0$ とすると，ラグランジアンは

$$L = Z(\boldsymbol{f}) - \sum_{rs} u_{rs} \Big(\sum_{k} f_k^{rs} - q_{rs} \Big) - \sum_{rs} \sum_{k} \mu_k^{rs} \, f_k^{rs} \tag{2.115}$$

となる．

停留条件の導出

連鎖律より $\partial Z/\partial f_k^{rs} = \sum_a t_a(x_a)\,\partial x_a/\partial f_k^{rs} = \sum_a t_a(x_a)\,\delta_{a,k}^{rs} = c_k^{rs}$（経路 k^{rs} の旅行時間）である．停留条件 $\partial L/\partial f_k^{rs} = 0$ は

$$c_k^{rs} - u_{rs} - \mu_k^{rs} = 0 \quad \Longrightarrow \quad c_k^{rs} = u_{rs} + \mu_k^{rs}. \tag{2.116}$$

相補性条件の場合分けと Wardrop 条件

相補性条件 $\mu_k^{rs}\, f_k^{rs} = 0$（$\mu_k^{rs} \geq 0$，$f_k^{rs} \geq 0$）を式 (2.116) と合わせて場合分けすると：

- $f_k^{rs} > 0$（経路 k が**使用されている**）$\Rightarrow \mu_k^{rs} = 0 \Rightarrow c_k^{rs} = u_{rs}$
- $f_k^{rs} = 0$（経路 k が**使用されていない**）$\Rightarrow \mu_k^{rs} \geq 0 \Rightarrow c_k^{rs} = u_{rs} + \mu_k^{rs} \geq u_{rs}$

これはまさに Wardrop の第 1 原則（第 2.3 節 第 2.3.1 小節，式 (2.20)）に他ならない．u_{rs} が均衡最小旅行時間（均衡経路費用）に対応している．

結論（命題 D.1） BPR 関数のような単調増加・凸な費用関数のもとでは，UE/FD-Primal の KKT 条件と Wardrop の第 1 原則は完全に等価である．すなわち，UE/FD-Primal の最適解は Wardrop UE 解である．

2.13.3　SUE/FD-path への適用（ロジット確率との等価性）

問題の再掲

ロジット型 SUE の等価最適化問題（SUE/FD-path，第 2.4 節 第 2.4.2 小節参照）は，エントロピー項を加えた以下の形をとる：

$$\min_{\boldsymbol{f}} \quad Z(\boldsymbol{f}) = \sum_a \int_0^{x_a} t_a(w)\,dw - \frac{1}{\theta}\sum_{rs} q_{rs}\, H_{rs}(\boldsymbol{f}^{rs}), \tag{2.117}$$

$$H_{rs} = -\sum_k \frac{f_k^{rs}}{q_{rs}} \ln \frac{f_k^{rs}}{q_{rs}}, \quad \text{s.t.} \quad \sum_k f_k^{rs} = q_{rs},\ \ f_k^{rs} \geq 0. \tag{2.118}$$

SUE では全経路にフローが分散する（$f_k^{rs} > 0$，$\forall k$）ため，停留条件のみ導けばよい（非負制約は活性化しない：$\mu_k^{rs} = 0$）．

停留条件の導出

$\partial H_{rs}/\partial f_k^{rs} = -(\ln(f_k^{rs}/q_{rs}) + 1)/q_{rs}$ を利用すると，停留条件 $\partial Z/\partial f_k^{rs} = 0$ は

$$c_k^{rs} + \frac{1}{\theta}\left(\ln \frac{f_k^{rs}}{q_{rs}} + 1\right) = u_{rs} \tag{2.119}$$

となる（u_{rs} は等式制約 $\sum_k f_k^{rs} = q_{rs}$ の Lagrange 乗数）．

整理すると

$$\ln \frac{f_k^{rs}}{q_{rs}} = \theta(u_{rs} - c_k^{rs}) - 1 \quad \Longrightarrow \quad f_k^{rs} = q_{rs}\, e^{-1}\, e^{\theta u_{rs}}\, e^{-\theta c_k^{rs}}. \tag{2.120}$$

全経路 k' について式 (2.120) の両辺を和をとり，流量保存条件 $\sum_{k'} f_{k'}^{rs} = q_{rs}$ を代入すると，$e^{-1}e^{\theta u_{rs}}$ が消えて

$$\boxed{\frac{f_k^{rs}}{q_{rs}} = \frac{e^{-\theta c_k^{rs}}}{\displaystyle\sum_{k'} e^{-\theta c_{k'}^{rs}}}} \tag{2.121}$$

を得る．これはロジット型の確率的経路選択確率（第 2.4 節 第 2.4.1 小節，式 (2.42)）そのものである．

結論（命題 D.2） SUE/FD-path の KKT 停留条件はロジット選択確率と一致する．すなわち, SUE/FD-path の最適解は SUE 均衡であることが最適性条件から直接確認できる．

UE との対比

	UE/FD-Primal	SUE/FD-path
目的関数の違い	なし（エントロピー項なし）	エントロピー項 $-H_{rs}/\theta$ あり
KKT 停留条件	$c_k^{rs} = u_{rs} + \mu_k^{rs}$	$c_k^{rs} + \theta^{-1}(\ln p_k^{rs} + 1) = u_{rs}$
相補性条件	$\mu_k^{rs}\, f_k^{rs} = 0$（有意）	$\mu_k^{rs} = 0$（全経路正フロー）
均衡条件	Wardrop 第 1 原則	ロジット確率的選択
$\theta \to \infty$	—	UE に収束
$\theta \to 0$	—	均等配分に収束

参考文献

[1] Akamatsu, T.: Cyclic flows, Markov process and stochastic traffic assignment, Transportation Research Part B, Vol. 30, No. 5, pp. 369–386, 1996.

[2] Arnott, R., de Palma, A., Lindsey, R.: Economics of a bottleneck, Journal of Urban Economics, Vol. 28, No. 3, pp. 327–346, 1990.

[3] Beckmann, M., McGuire, C. B., Winsten, C. B.: Studies in the Economics of Transportation, Yale University Press, New Haven, CT, 1956.

[4] Bell, M. G. H., Cassir, C.: Risk-averse user equilibrium traffic assignment: an application of game theory, Transportation Research Part B, Vol. 36, No. 8, pp. 671–681, 2002.

[5] Braess, D.: Über ein Paradoxon aus der Verkehrsplanung, Unternehmensforschung, Vol. 12, No. 1, pp. 258–268, English translation: *Transportation Science*, 39(4), 446–450, 2005, 1968.

[6] Daganzo, C. F.: Urban gridlock: Macroscopic modeling and mitigation approaches, Transportation Research Part B, Vol. 41, No. 1, pp. 49–62, 2007.

[7] Daganzo, C. F., Sheffi, Y.: On stochastic models of traffic assignment, Transportation Science, Vol. 11, No. 3, pp. 253–274, 1977.

[8] Dial, R. B.: A probabilistic multipath traffic assignment algorithm which obviates path enumeration, Transportation Research, Vol. 5, pp. 83–111, 1971.

[9] Fisk, C.: Some developments in equilibrium traffic assignment, Transportation Research Part B, Vol. 14, No. 3, pp. 243–255, 1980.

[10] Frank, M., Wolfe, P.: An algorithm for quadratic programming, Naval Research Logistics Quarterly, Vol. 3, No. 1–2, pp. 95–110, 1956.

[11] Friesz, T. L., Bernstein, D., Smith, T. E., Tobin, R. L., Wie, B. W.: A variational inequality formulation of the dynamic network user equilibrium problem, Operations Research, Vol. 41, No. 1, pp. 179–191, 1993.

[12] Gazis, D. C., Herman, R., Rothery, R. W.: Nonlinear follow-the-leader models of traffic flow, Operations Research, Vol. 9, No. 4, pp. 545–567, 1961.

[13] Geroliminis, N., Daganzo, C. F.: Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings, Transportation Research Part B, Vol. 42, No. 9, pp. 759–770, 2008.

[14] Kuwahara, M., Akamatsu, T.: Dynamic user optimal assignment with physical queues for a many-to-many OD pattern, Transportation Research Part B, Vol. 35, No. 5, pp. 461–479, 2001.

[15] Lighthill, M. H., Whitham, G. B.: On kinematic waves. II. A theory of traffic flow on long crowded roads, Proceedings of the Royal Society A, Vol. 229, No. 1178, pp. 317–345, 1955.

[16] McFadden, D.: Conditional logit analysis of qualitative choice behavior, In Zarembka, P., ed.: Frontiers in Econometrics, Academic Press, New York, pp. 105–142, 1974.

[17] Monderer, D., Shapley, L. S.: Potential games, Games and Economic Behavior, Vol. 14, No. 1, pp. 124–143, 1996.

[18] Nash, J.: Equilibrium points in n-person games, Proceedings of the National Academy of Sciences, Vol. 36, No. 1, pp. 48–49, 1950.

[19] Nash, J.: Non-cooperative games, Annals of Mathematics, Vol. 54, No. 2, pp. 286–295, 1951.

[20] Pigou, A. C.: The Economics of Welfare, Macmillan, London, 1920.

[21] Pokutta, S.: The Frank–Wolfe algorithm: A short introduction, Jahresbericht der Deutschen Mathematiker-Vereinigung, Vol. 126, No. 1, pp. 3–35, 2024.

[22] Richards, P. I.: Shock waves on the highway, Operations Research, Vol. 4, No. 1, pp. 42–51, 1956.

[23] Sandholm, W. H.: Potential games with continuous player sets, Journal of Economic Theory, Vol. 97, No. 1, pp. 81–108, 2001.

[24] Sandholm, W. H.: Population Games and Evolutionary Dynamics, MIT Press, Cambridge, MA, 2010.

[25] Sheffi, Y.: Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods,

Prentice-Hall, Englewood Cliffs, NJ, 1985.

[26] Small, K. A.: The scheduling of consumer activities: work trips, American Economic Review, Vol. 72, No. 3, pp. 467–479, 1982.

[27] Smith, M. J.: The stability of a dynamic model of traffic assignment, Transportation Science, Vol. 18, No. 3, pp. 245–252, 1984.

[28] Taylor, P. D., Jonker, L. B.: Evolutionary stable strategies and game dynamics, Mathematical Biosciences, Vol. 40, No. 1–2, pp. 145–156, 1978.

[29] Vickrey, W. S.: Congestion theory and transport investment, American Economic Review, Vol. 59, No. 2, pp. 251–260, 1969.

[30] Wardrop, J. G.: Some theoretical aspects of road traffic research, Proceedings of the Institute of Civil Engineers, Part II, Vol. 1, No. 36, pp. 767–768, 1952.

[31] 土木学会: 交通ネットワークの均衡分析 – 最新の理論と解法, 土木学会, 1998.

[32] 赤松隆: 交通流の予測・誘導・制御と動的なネットワーク配分理論, 土木計画学研究・論文集, Vol. 13, pp. 23–48, 1996.

第 3 章

行動モデル

松永隆宏

序章 行動モデルとは

交通行動の記述

前章では，交通ネットワークに対して需要がどのように分布するかを記述する交通量配分問題について解説した．本章では焦点を個々の利用者の選択行動自体に移し，その記述のための理論的枠組みである**行動モデル**（Behavioral Model）を導入する．行動モデルとは，個人がどのようなプロセスを経て選択を行うのかを記述するモデルの総称であり，交通手段選択，目的地選択，経路選択，出発時刻選択，活動種類・時間配分など多様な選択行動を対象とする．

行動モデルの理論的基盤

行動モデルの理論的基盤はミクロ経済学における消費者行動理論に遡る．消費者行動理論では，個人（消費者）が財の組合せの中から自身の効用を最大化するように選択を行うと仮定する．交通行動に即していえば，「財」は交通手段や経路などの選択肢，「効用」は旅行時間・費用・快適さなどの属性から決まる満足度に対応する．

この効用最大化の枠組みにおいて，選択肢が離散的（鉄道かバスか，経路 A か経路 B か）である場合の需要関数（選択確率）を導出する理論は，大きく二つの系譜に整理できる（なお必ずしも二つの要素は並列関係にない．むしろ摂動効用理論の方が包括的な概念であるが，ここでは Logit モデルの選択確率 (3) の出自の観点から二つの系譜を紹介する）．

加法的ランダム効用最大化（ARUM） 各選択肢の効用に確率的な誤差項を加え，効用最大の選択肢が選ばれると仮定する．誤差項の分布形状を特定することで選択確率が導出される（例：ガンベル分布 $\rightarrow$ Logit モデル）．交通行動分析で最も広く用いられてきたアプローチである．

摂動効用理論（Perturbed Utility Theory） 財の期待効用と正則化項（摂動項）からなる直接効用関数を最適化する需要関数として選択確率を導出する．誤差項の分布ではなく正則化項の形状によってモデルが特徴づけられる．ARUM とは異なる出自を持つが，正則化項に Shannon entropy を仮定すると Logit モデルの選択確率と等価になるという双対的関係が知られている．

すなわち，交通行動分析で広く用いられてきた Logit モデルの選択確率

$$P(a) = \frac{\exp(V_a)}{\sum_{a'} \exp(V_{a'})}$$

は，ARUM における「ガンベル分布の仮定」と摂動効用理論における「Shannon entropy 正則化の仮定」という，共役関係にある二つの理論から同一の式として導出される．本章ではまず ARUM の枠組みから出発して Logit モデルの理論的基礎を解説するが，第 3 節では摂動効用理論の視点から経路選択問題を再検討し，選択肢集合の列挙を必要としないモデルへと議論を展開する．

行動モデルの対象とする選択行動は多岐にわたり，離散選択・連続選択，選択肢相関・集合生成など論点も多様であるため，「行動モデル理論」という単一の括りで整理しきれるものではない．本章は交通行動の記述に関する理論的な基礎を提供することを目的とし，上記の理論的枠組みに基づく行動モデルを中心に解説する．

本節の構成

本章の構成は以下の通りである．

第 1 節 ランダム効用最大化理論（RUM） 多くの行動モデルの基礎となるランダム効用最大化理論を導入する．また最も基本的なモデルとして多項選択モデル（Multinomial Logit Model; MNL）の定式化を行い，その特性を解説する．

第 2 節 選択肢相関や異質性を考慮したモデル MNL モデルでは説明できない選択肢間の相関や意思決定主体間の異質性を考慮するためのモデルを解説する．

第 3 節 選択肢非列挙型モデル 離散選択モデルはあらかじめ選択肢集合が規定されていることを前提とするが，例えば OD 間に無数の経路が存在する場合など，選択肢の列挙は必ずしも容易でない．本節では，この課題に対処するために提案されてきた選択肢非列挙型モデルを解説する．

第 4 節 アクティビティモデル 行動モデルが記述する選択行動は単独のトリップに限らない．最も包括的な行動表現は活動経路（Activity Path）であろう．本節では活動経路の選択行動を記述する Activity Based Models (ABMs) を解説し，付随する離散連続モデルについても触れる．

第 5 節 行動モデルの推定 行動モデルは，実際の交通行動データを用いてモデルパラメータを推定して初めて解釈や政策分析に利用できる．本節では，行動モデルの推定方法について，特に最大尤度法を中心に解説する．

3.1　ランダム効用最大化理論

本節では行動モデルの理解にあたり，ランダム効用最大化理論（Random Utility Maximization Theory, RUM）[13] を導入する．本教科書で扱う範囲ではより厳密には加法的ランダム効用最大化理論（Additive Random Utility Maximization Theory, ARUM）と呼ぶべきものであるが，以下では便宜上 RUM と呼ぶことにする．RUM はミクロ経済学の重要な仮定であり，個人は効用を最大化するような合理的な選択行動を行うとするものである．

本節の構成は以下の通りである．まず 3.1.1 項で効用関数の記述と誤差項の導入を行い，3.1.2 項で誤差項の独立同一性の仮定を述べる．続いて 3.1.3 項で誤差項の分布形状（正規分布・ガンベル分布）を具体的に示し，多項ロジット（MNL）モデルを導出する．最後に 3.1.4 項で，MNL モデルが持つ IIA 特性の問題と，次節以降での拡張への動機を述べる．

3.1.1 効用の記述とランダム性の導入

まず選択肢の効用評価の記述方法を述べる．各選択肢 a は分析者によって設定された説明変数 x_i および対応するパラメータ β_i を用いて評価される．その評価関数を効用関数と呼び，通常は線形な形式で表される．すなわち，選択肢 a の効用 U_a は以下のように表される．

$$U_a = \sum_i \beta_i x_i \tag{3.1}$$

なお本定式化では個人の異質性は考慮していない．例えば交通手段選択であれば，説明変数 $\boldsymbol{x}$ は「旅行時間」「旅行費用」「快適さ」などの属性を表し，パラメータ $\boldsymbol{\beta}$ はこれらの属性が効用に与える影響の大きさを表す．行動モデルの目的は，この効用関数を用いて個人の選択行動を記述することであり，それは各パタメータ β_i を推定することに帰着する．

個人 k の選択行動結果を y_k と表すと，効用最大化の仮定により

$$y_k = \arg\max_a U_a \tag{3.2}$$

となる．これは毎回確定的な選択が行われることを示すが，実際にはこのように確定的に対象の選択行動を説明することは困難である．なぜなら，(3.1) は各選択肢が分析者によって選定された特定の説明変数 $\boldsymbol{x}$ に基づいて評価されることを前提としているが，真の効用関数は選択者毎の私的情報そのものであり，分析者が全ての構成を把握することは原理的に困難であるためである．このような不確実性を考慮するために，効用関数 (3.1) にランダム性を加えるための誤差項 ε_a を導入する．すなわち，選択肢 a の効用は以下のように再定義される．

$$U_a = \sum_i \beta_i x_{a,i} + \varepsilon_a \tag{3.3}$$

ε_a は選択肢 a に特有のランダムな誤差項であり，上で述べた個人の選択行動に影響を与えるが分析者が観測できない要因を表す．なおこの誤差項に対し，従来の確定的な効用項 $V_a = \sum_i \beta_i x_{a,i}$ を確定項と呼ぶ．

3.1.2 誤差項の独立同一性の仮定

本項及び次項では誤差項に関する具体的な仮定を導入する．

まず，誤差項の独立同一性（Independent Identical Distribution, i.i.d.）に関する仮定を導入する．

効用関数 (3.1) は最も一般的には個人 k，選択肢 a に依存する値として表される．

$$U_{ka} = V_{ka} + \varepsilon_{ka} \tag{3.4}$$

独立同一性: Independent and Identically Distributed (i.i.d.) は以下の二つの要素からなる．

独立性　各個人および選択肢の組 k, a に対して ε_{ka} は互いに独立である．

同一分布　各個人および各選択肢の組 k, a に対して ε_{ka} は同一の確率分布に従う．

ここで独立とは，ある選択肢に関する未観測要因が他の選択肢の未観測要因と相関を持たないことを意味し，同一分布とは，それらの未観測要因のばらつき方がすべての選択肢・個人で同じであることを意味する．

3.1.3　誤差項の分布形状

続いて誤差項の分布形状を考える．誤差項によるランダム性の導入に伴い，(3.2) は次のように書き換えられる（個人 k の識別子は省略）．

$$y_k = \arg\max_a \left(\sum_i \beta_i x_{a,i} + \varepsilon_a \right) = \arg\max_a \left(V_a + \varepsilon_a \right) \tag{3.5}$$

選択肢 a の選択確率は次のように表される．

$$\begin{aligned} \Pr(y_k = a) &= \Pr\left(V_a + \varepsilon_a > V_{a'} + \varepsilon_{a'} \quad \forall a' \neq a \right) \\ &= \Pr\left(V_a - V_{a'} > \varepsilon_{a'} - \varepsilon_a \quad \forall a' \neq a \right) \\ &= \Pr_\varepsilon\left(V_a - V_{a'} > \varepsilon \quad \forall a' \neq a \right) \\ &= \int_{-\infty}^{V_a - V_{a'}} f(\varepsilon) d\varepsilon \\ &= F_\varepsilon(V_a - V_{a'}) \end{aligned} \tag{3.6}$$

従って選択確率は誤差項 ε の確率分布 $f(\varepsilon)$ および累積分布関数 $F_\varepsilon(\varepsilon)$ に明らかに依存する．

では続いてこの誤差項 ε の分布形状を具体的に与えてみよう．

(a)　正規分布の場合

誤差項に正規分布を仮定したものが **Probit Model** である．

まず二項選択の場合を考える．(3.6) において $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2)$ が 2 変数正規分布 $MVN(0, \Sigma)$ に従うと仮定する．ここで，

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} \tag{3.7}$$

である．分散共分散行列 Σ の非対角成分は選択肢間の誤差相関を表すが，これによって相関を許容できる点が Probit Model の利点の一つである．

$\varepsilon_2 - \varepsilon_1$ は $\sigma^2 = \mathrm{Var}(\varepsilon_2 - \varepsilon_1)$ の分散を持つ正規分布 $N(0, \sigma^2)$ に従うため，

$$P_1 = \Phi\left(\frac{V_1 - V_2}{\sigma} \right) \tag{3.8}$$

を得る．このように二項選択の場合は closed-form で表されるため，数値的な積分を必要とせず計算が容易である．

一方，多項の場合の選択確率は一般に多変量正規分布の積分として表されるため，二項の場合と異なり closed-form で簡潔に表せないことが多い．そのため実際の推定には数値積分やシミュレーションが必要となり，計算コストが高くなる．

(b) ガンベル分布の場合

これに対して，誤差項が先述した i.i.d. なガンベル分布に従うと仮定したものが **Logit Model** である．ガンベル分布は正規分布と類似の形状を持つが，累積分布関数が closed-form で表されるという特徴がある．ガンベル分布の累積分布関数は次のように表される．

$$F(x) = \exp\bigl(-\exp(-\mu(x-\eta))\bigr) \tag{3.9}$$

ここで η はロケーションパラメータ（分布の最頻値に一致），$\mu > 0$ はスケールパラメータ（分布のばらつきに関係）である．離散選択モデルにおいては，通常 $\eta = 0$，$\mu = 1$ に正規化する．ロケーションパラメータ η は効用の確定項 V_a に吸収でき，スケールパラメータ μ はパラメータ β_i と同時には識別できない（$\mu V_a = \mu \sum_i \beta_i x_{a,i}$ より，推定されるのは常に $\mu\beta_i$ の積である）ためである．

この正規化のもとで，選択確率は次のように表される．

$$\Pr(y_k = a) = \frac{\exp(V_a)}{\sum_{a'} \exp(V_{a'})} \tag{3.10}$$

これは closed-form で表されるため，数値的な積分を必要とせず計算が容易である．従って離散選択モデルにおいてはガンベル分布を使用することが多い．

一般に行動モデルの文脈で離散選択モデルを用いる際には，上記仮定を踏まえて誤差項に i.i.d. ガンベル分布を仮定することが多い．すなわち効用関数を以下のように宣言することが多い．

$$U_{ka} = V_{ka} + \varepsilon_{ka}, \quad \varepsilon_{ka} \sim \text{i.i.d. Gumbel} \tag{3.11}$$

なお選択肢が二つの場合は特に Binary Logit Model と呼び，複数の場合は **Multinomial Logit Model** と呼ぶ．

(c) 摂動効用理論との関連

なお序章で述べたように，Logit 選択確率は摂動効用理論の観点からも導出できる．具体的には，需要ベクトル p（各選択肢の選択確率）に対して，期待効用に Shannon entropy による正則化項を加えた

$$\max_{p_a \geq 0,\ \sum_a p_a = 1} \left\{ \sum_a p_a V_a + \mu H(p) \right\}, \qquad H(p) = -\sum_a p_a \ln p_a \tag{3.11}$$

を考えると，その最適解として Logit 選択確率が得られる．ここで，しばしば

$$H(p) = -\sum_a (p_a \ln p_a - p_a) \tag{3.12}$$

と等価に書かれることもあるが，これは制約 $\sum_a p_a = 1$ のもとで定数差しか持たず，最適解には影響しない．

このことは，ARUM において「誤差項に i.i.d. Gumbel 分布を仮定する」ことと，摂動効用理論において「Shannon entropy 型の正則化項を導入する」ことが，同一の Logit 選択確率を与える等価な表現であることを意味する．すなわち，一方は誤差項の分布仮定によって，他方は摂動項の形状によって，同じ行動仮説を表現していると解釈できる．この双対的な見方は，第 3 節で摂動効用理論に基づく経路選択モデルを導入する際に重要となる．

なお，ガンベル分布と正規分布の確率密度関数は図 3.1 のような形状をしている．

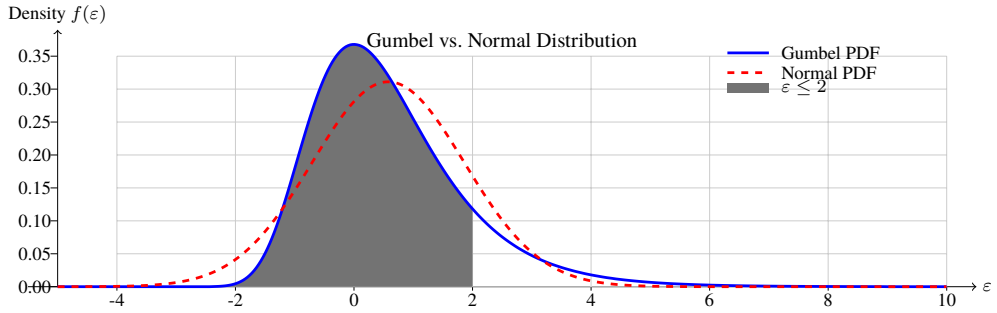


図 3.1: ガンベル分布の確率密度関数 (PDF) と，$\varepsilon \leq 2$ の領域

3.1.4　選択肢・個人間の相関・異質性の扱いと IIA 性の問題

上記では誤差項の独立同一性（i.i.d.）に関する仮定を導入したが，この仮定は現実の選択行動を記述する上でしばしば問題となる．例えば交通行動の文脈においては，互いに類似した交通手段や重複区間の多い経路どうしで未観測効用が似通うことが多く，独立性の仮定がしばしば問題となる．また隣接する複数の地点は，距離が離れた地点と比べて特徴が似通うことが多いため，目的地選択の文脈においても独立性の仮定が問題となる．さらに時間価値や混雑回避傾向は個人によって異なるため，個人間で同一の分布を仮定することも強い制約となる．こうした理由から，Nested Logit や Mixed Logit など，i.i.d. 仮定を緩和するモデルが導入されてきた．これらのモデルについては次節にて詳述する．

3.2　MNL からの拡張: 相関・異質性

本節では，前節で導入した MNL モデルからの拡張として，選択肢間の相関や意思決定主体間の異質性を考慮した Logit モデルを解説する．

本節の構成は以下の通りである．まず 3.2.1 項で MNL モデルの IIA 特性がもたらす限界を赤バス・青バス問題を通じて示す．次に 3.2.2 項で選択肢間の相関を部分的に許容する Nested Logit モデルを導入し，3.2.3 項で個人間の異質性を表現する Mixed Logit モデルを解説する．

3.2.1　IIA の限界

IIA（Independence of Irrelevant Alternatives）特性は直訳すると「無関係な選択肢との独立性」となるが，具体的には「任意の二つの選択肢の間の選択確率の比はその二つの選択肢の効用のみによって決まる」という性質のことである．例えば選択肢 a と b の間の選択確率の比は，以下の様に選択肢 a と b の効用のみによって決まる．

$$\frac{P(a)}{P(b)} = \frac{\frac{\exp(V_a)}{Z}}{\frac{\exp(V_b)}{Z}} = \exp(V_a - V_b) \tag{3.13}$$

ただし $Z = \sum_{a'} \exp(V_{a'})$ である．つまり選択肢 a, b いづれとも異なる第三の選択肢 c の効用 V_c はこの比に影響を与えない．この IIA 特性は，各選択肢効用の誤差項が独立であるとする i.i.d. の

仮定に起因するものであり，MNL モデルの重要な特徴である．

しかしながら選択肢間の相関，より厳密には誤差項の間に相関が存在する場合にはこの IIA 特性が崩れる．代表的な例として次の**赤バス・青バス問題**が知られている．

赤バス青バス問題

「車」「赤バス」の二つの選択肢から選ぶ二項 Logit モデルにおいて，効用の確定部分が全く同じであれば両者の選択確率は $1/2, 1/2$ となる．ここに「青バス」という選択肢を加えたとき，青バスの確定効用も他の二つの選択肢の確定効用と等しい場合，三つの選択肢の選択確率は $1/3, 1/3, 1/3$ となる．ところが，色違いのバスが登場しただけでもともと車を利用していた人の三分の一もがバスに転移するのは不自然である．

この問題は，車と赤バスの選択確率の比が青バスの有無によらず $1:1$ であるという IIA 特性によって生じる問題である．より根本的には，交通手段選択の文脈において暗に公共交通機関と自家用車という比較構造を上位に仮定する場合は，赤バスと青バスの効用の誤差項が独立であるという i.i.d. の仮定が現実には成立しないことに起因する．赤バス・青バスはともに車と比べて公共交通という性質を共有している点で類似した選択肢であるため，両者の効用の誤差項は相関していると考えるのが自然である．こうした選択肢間の相関を考慮するためには，**Nested Logit モデル**などのより発展的なモデルを用いる必要がある．

3.2.2 Nested Logit モデル

上述した誤差項の相関に起因する IIA 特性の限界を克服するためのモデルの一つが Nested Logit（NL）モデル（Ben-Akiva, 1973）である．NL モデルでは，誤差項が互いに相関関係にある選択肢同士を**ネスト**（グループ）にまとめることで，一部の選択肢間に相関がある場合でも多項選択を記述可能にしている．例えば先の赤バス・青バス問題においては，公共交通という性質を共有する赤バスと青バスを同一のネストにまとめることができる．この時，車と公共交通という誤差項が相関しない（と仮定できる）二つの選択肢を上位ネスト，赤バスと青バスという誤差項が相関する二つの選択肢を下位ネストに配置する二層のツリー構造を考えることができる．

より一般化して，上位ネストの選択肢を u、下位ネストの選択肢を l として選択肢 (u, l) の効用は以下のように定義される．

$$U_{ul} = V_u + V_l + V_{ul} + \epsilon_u + \epsilon_{ul} \tag{3.14}$$

ここで，V_u, V_l はそれぞれ選択肢 u, l に固有の効用の確定部分，V_{ul} は u, l の組合せで決まる効用の確定部分である．ϵ_{ul} は u, l の組合せで決まる効用の誤差項であり，スケールパラメータ μ の i.i.d. ガンベル分布に従うと仮定する．

ϵ_u は選択肢 u に固有の効用の誤差項であり，下位ネストの選択肢間で共通に持つ成分を表す．赤バス・青バス問題の例では，二つのバス間の誤差項のうち相関する部分が ϵ_u に吸収され，それぞれのバスに固有の誤差項が ϵ_{ul} で表される．このネスティングによって，誤差項が相関関係にある場合も，相関成分を上位ネストの誤差項 ϵ_u に吸収することで、下位ネスト内の選択肢の誤差項 ϵ_{ul} は互いに独立であると仮定できるようになる．

選択確率は条件付き確率と周辺確率の積として表す．

$$P(u,l) = P(l \mid u) \cdot P(u) \tag{3.15}$$

(a)　下位ネスト

まず下位ネスト内の条件付き確率 $P(l \mid u)$ を考える．ネスト u が与えられた条件のもとでは，ϵ_u は全ての選択肢 l に共通の定数として作用するため，選択肢間の比較において相殺される．したがって下位ネスト内の選択は ϵ_{ul}（i.i.d. ガンベル分布）のみに依存し，MNL と同様の形で与えられる．

$$P(l \mid u) = \frac{\exp\bigl(\mu(V_l + V_{ul})\bigr)}{\sum_{l'} \exp\bigl(\mu(V_{l'} + V_{ul'})\bigr)} \tag{3.16}$$

(b)　上位ネスト

次に，上位ネストの周辺確率 $P(u)$ を導出する．上位ネストの選択は「ネスト u を選んだときに得られる最大効用」同士の比較であり，次のように表される．

$$P(u) = \Pr\Bigl[\max_l U_{ul} \geq \max_l U_{u'l}, \quad \forall u' \neq u\Bigr] \tag{3.17}$$

ここで，ϵ_{ul} が i.i.d. ガンベル分布に従うとき，ガンベル分布の最大値もまたガンベル分布に従うという性質から，$\max_l(V_l + V_{ul} + \epsilon_{ul})$ はガンベル分布に従う．この項はネスト u を選んだときに得られる期待効用がガンベル分布に従うことを意味しており，この最大期待効用がロケーションパラメータに該当する．この値はログサム変数 V'_u として定義される．

$$V'_u \equiv \frac{1}{\mu} \ln \sum_l \exp\bigl(\mu(V_l + V_{ul})\bigr) \tag{3.18}$$

前節でガンベル分布を導入した際, ロケーションパラメータ η は効用の確定項に吸収できるため $\eta = 0$ に正規化したが，ここでも同じ操作を行う．$\max_l(V_l + V_{ul} + \epsilon_{ul})$ が従うガンベル分布のロケーションパラメータは V'_u であるから，これを確定項として分離し，残差を $\epsilon'_u \equiv \max_l(V_l + V_{ul} + \epsilon_{ul}) - V'_u$ とおけば，ϵ'_u はロケーション 0 のガンベル分布に従う．この分離により，上位ネストの比較は次のように書き直せる．

$$P(u) = \Pr\Bigl[V_u + V'_u + \epsilon_u + \epsilon'_u \geq V_{u'} + V'_{u'} + \epsilon_{u'} + \epsilon'_{u'}, \quad \forall u' \neq u\Bigr] \tag{3.19}$$

これは確定項 $V_u + V'_u$，誤差項 $\epsilon_u + \epsilon'_u$ をもつ離散選択問題の構造となっている．すなわち，下位ネストの選択において相殺された ϵ_u は，ここで上位ネストの誤差項の一部として機能する．ϵ_u がスケールパラメータ μ^u のガンベル分布に従うと仮定すれば，周辺確率も MNL と同様の形で与えられる．

$$P(u) = \frac{\exp\bigl(\mu^u(V_u + V'_u)\bigr)}{\sum_{u'} \exp\bigl(\mu^u(V_{u'} + V'_{u'})\bigr)} \tag{3.20}$$

(c)　NL モデルの選択確率

以上をまとめると，各選択肢 (u,l) の選択確率は次のようになる．

$$P(u,l) = \frac{\exp\big(\mu(V_l + V_{ul})\big)}{\sum_{l'} \exp\big(\mu(V_{l'} + V_{ul'})\big)} \cdot \frac{\exp\big(\mu^u(V_u + V'_u)\big)}{\sum_{u'} \exp\big(\mu^u(V_{u'} + V'_{u'})\big)} \tag{3.21}$$

なお，二つのスケールパラメータ μ^u と μ を同時に推定することはできないため，通常は条件付き確率のスケールパラメータ μ を 1 に正規化し，周辺確率のスケールパラメータ μ^u を推定する．ガンベル分布のスケールパラメータは分布のばらつきの程度を表しており，標準偏差に反比例する値であるため，μ^u/μ は以下の関係を満たす必要がある．

$$\frac{\mu^u}{\mu} = \sqrt{\frac{\mathrm{Var}(\epsilon_{ul})}{\mathrm{Var}(\epsilon_u) + \mathrm{Var}(\epsilon_{ul})}} \leq 1 \tag{3.22}$$

この条件はネスト構造が妥当であること，すなわち上位ネストの誤差分散が非負であることを保証するものである．$\mu^u/\mu = 1$ のとき上位・下位の誤差項間に相関がなくなり，NL モデルは MNL モデルに帰着する．

3.2.3 個人間の異質性と Mixed Logit モデルの導入

MNL モデルでは，パラメータは母集団で一定であると仮定しており，個人間の選好の違い（異質性）を表現できない．もっとも，性別や年齢層などのダミー変数を説明変数に導入し交互作用項を設けることで，グループ間の異質性をある程度考慮することは可能である．しかしこのアプローチでは，あくまでダミー変数で区分されたグループ単位での平均的な違いを捉えるにとどまり，同一グループ内での個人単位の選好のばらつきは依然表現できない．

Mixed Logit（MXL）モデルは，効用関数のパラメータそのものを確率分布に従うランダム係数として扱うことで，個人間の異質性を連続的に表現できるように拡張したモデルである．標準的な MNL モデルでは，誤差項 ϵ_{ka} は i.i.d. ガンベル分布に従うと仮定されるが，Mixed Logit ではこれに加えて，個人ごとに異なる係数ベクトルを導入することで，より柔軟な代替パターンやパネルデータにおける系列相関を表現できる．すなわち，Mixed Logit は「誤差項に正規分布を仮定した Probit モデルと Logit モデルの中間」と理解するよりも，**Logit 確率をランダム係数の分布で混合したモデル**と捉える方が適切である．

(a) Mixed Logit モデルの定式化

個人 k の選択肢 a に対する効用を，説明変数ベクトル $\boldsymbol{x}_{ka}$ を用いて

$$U_{ka} = \sum_i \beta_{ki} x_{kai} + \epsilon_{ka} = V_{ka} + \epsilon_{ka} \tag{3.23}$$

と定義する．ここで，ϵ_{ka} は i.i.d. ガンベル分布に従う誤差項であり，β_k は個人 k に固有のランダム係数ベクトルである．ランダム係数 β_k は母集団内である確率密度関数 $f(\beta \mid \theta)$ に従うと仮定し，θ はその平均や分散共分散行列などの母数を表す．

例えば，β_k を

$$\beta_k \sim N(\bar{\beta}, \Omega)$$

のような多変量正規分布に従うと仮定すれば，個人ごとの選好のばらつきや係数間の相関を表現することができる．このとき，β_k が与えられた条件のもとでの選択確率は，MNL モデルと同じく

$$P_{ka}(\beta_k) = \frac{\exp(V_{ka})}{\sum_{j} \exp(V_{kj})} \tag{3.24}$$

と表される．

しかし実際には β_k は観測されないため，その分布について積分することで無条件の選択確率を得る．すなわち，

$$P_{ka} = \int \frac{\exp(V_{ka})}{\sum_{j} \exp(V_{kj})} f(\beta \mid \theta)\, d\beta \tag{3.25}$$

である．このように，Mixed Logit は**条件付きでは Logit，無条件ではランダム係数の分布で平均化された選択モデル**と解釈できる．

なお，Mixed Logit は誤差成分表示として

$$U_{ka} = V_{ka} + \eta_{ka} + \epsilon_{ka} \tag{3.26}$$

と書かれることもある．ここで，V_{ka} は平均的な効用の確定項，η_{ka} は個人間の異質性や選択肢間の相関を表す追加的な確率項，ϵ_{ka} は i.i.d. ガンベル分布に従う誤差項である．ただしこの表現においても本質は，η_{ka} を通じて効用係数や誤差構造に柔軟性を与え，その結果として Logit 確率を混合している点にある．

(b)　シミュレーション推定

式 (3.25) の積分は一般に解析的に計算できないため，シミュレーションによる近似計算が必要となる．具体的には，分布 $f(\beta \mid \theta)$ から R 回乱数を抽出し，各ドロー $\beta^{(r)}$ に対して条件付き選択確率

$$P_{ka}^{(r)} = \frac{\exp\left(V_{ka}^{(r)}\right)}{\sum_{j} \exp\left(V_{kj}^{(r)}\right)} \tag{3.27}$$

を計算する．これらを平均することで，無条件の選択確率を

$$\hat{P}_{ka} = \frac{1}{R} \sum_{r=1}^{R} P_{ka}^{(r)} \tag{3.28}$$

と近似する．

このとき，シミュレーション対数尤度 $\ln SL$ は次のように書ける．

$$\ln SL = \sum_{k=1}^{K} \sum_{a=1}^{A} \delta_{ka} \ln \left(\frac{1}{R} \sum_{r=1}^{R} P_{ka}^{(r)} \right) \tag{3.29}$$

ここで，δ_{ka} は個人 k が選択肢 a を選んだとき 1，そうでないとき 0 をとる指示変数である．この対数尤度を最大化することで，ランダム係数分布 $f(\beta \mid \theta)$ の母数，すなわち平均や分散共分散行列の要素を推定することができる．

コラム：交通手段選択モデルの推定 – 石河万衣

行動モデルは政策議論等の基盤となるが、本稿ではその構築過程における試行錯誤に着目する。具体的には、松山市の PT 調査データを用い、以下の効用関数を設定して交通手段選択モデル（自動車、鉄道、バス）を推定した。

$$U_{train} = \beta_{time} x_{time}/10 + \beta_{fare} x_{fare}/100 + \beta_{train}$$
$$U_{bus} = \beta_{time} x_{time}/10 + \beta_{fare} x_{fare}/100 + \beta_{bus}$$
$$U_{car} = \beta_{time} x_{time}/10 + \beta_{fare} x_{fare}/100$$

以下の結果から、初期の MNL モデルでは運賃パラメータが正となり、運賃上昇が効用を増加させるという実態に反する結果を示した。

そこで運賃に対数変換を施し再推定したところ、尤度が改善し全パラメータの符号が整合した。これにより、低価格帯では運賃変動の影響が大きく、高価格帯では相対的に小さいという非線形の性質を適切に捉えることができた。

さらに、まず「自動車」か「公共交通」を選び、後者の場合に「鉄道」か「バス」を選ぶ階層的プロセスを想定し、NL モデルを推定した。MNL モデルより尤度がさらに向上し、スケールパラメータ λ も有意となった。これにより誤差項間の無相関仮説が棄却され、本対象におけるネスト構造導入の妥当性が統計的に実証された。

表 3.1: 交通手段選択モデル（N=1,902）

	MNL		MNL（運賃対数）		NL（運賃対数）	
	パラメータ	t 値	パラメータ	t 値	パラメータ	t 値
β_{train}	0.161	0.958	0.809	2.882 **	0.478	1.710
β_{bus}	-0.980	-5.609 ***	-0.310	-1.059	-0.258	-1.130
β_{time}	-0.431	-7.546 ***	-0.393	-7.509 ***	-0.361	-7.543 ***
β_{fare}	0.071	1.530	-0.672	-2.764 **	-0.364	-1.657
λ	-	-	-	-	0.551	4.687 ***
初期対数尤度 $L(0)$	-1599.214		-1599.214		-1599.214	
最終対数尤度 $L(\beta)$	-1080.426		-1077.551		-1072.256	
決定係数 ρ^2	0.324		0.326		0.330	
修正済み決定係数 $\bar{\rho}^2$	0.322		0.324		0.326	

***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$

3.3　選択肢非列挙型モデル

本節では，選択肢集合の問題と選択肢非列挙型の経路選択モデルについて解説する．

本節の構成は以下の通りである．まず 3.3.1 項で経路選択肢集合の爆発問題を示し，列挙型モデルの限界を説明する．次に 3.3.2 項で逐次的リンク選択に基づく Recursive Logit（RL）モデルとその一般化を導入する．3.3.3 項では，時間構造化ネットワークの導入によりサイクリック経路の扱いと動的計画法による効率的な求解が可能になることを示し，プリズム制約による選択肢集合の限定についても述べる．最後に 3.3.4 項で，摂動効用理論に基づく Perturbed Utility Route Choice（PURC）モデルを紹介し，RL モデルとの共役関係を論じる．

3.3.1　選択肢集合の爆発

経路選択肢集合は簡単に爆発する．例えばサイズ $n \times n$ のグリッドネットワーク（図 3.2）における単純な OD ペアに対して，同じ点を 2 度通らないような単純経路の数は表 3.2 のように n に対して急激に増加する．MNL モデルのような選択肢集合を全て列挙してから一括で経路を選択するモデル（選択肢列挙型モデル）ではこのような爆発する選択肢集合を扱うことは明らかに困難である．これに対する対象法として k 番目の最短経路までを選択肢集合とする方法などが用いられてきたが，k の決め方は恣意的であるため，選択肢集合の定義の難しさは根本的な問題として残る．

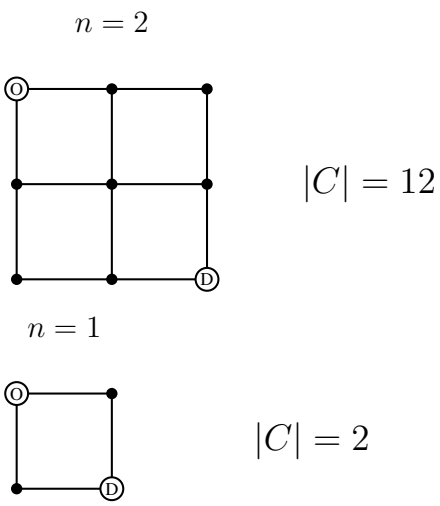


図 3.2: 格子ネットワークと単純経路数の例.

3.3.2　動的離散選択モデルと Recursive Logit モデル

以降では選択対象を経路に絞り，経路選択モデルの文脈において議論する．経路列挙の困難さに対処するために，経路非列挙型の経路選択モデルが提案されてきた．その基本的な着想は，経路を一体的な選択肢として扱うのではなく，交差点ごとに次のリンクを逐次的に選択するマルコフ過

表 3.2: 格子ネットワークにおける単純経路数の爆発

n	経路数
1	2
2	12
3	184
4	8,512
5	1,262,816
6	575,780,564
7	789,360,053,252
8	3,266,598,486,981,640
9	41,044,208,702,632,496,804
10	1,568,758,030,464,750,013,214,100
11	182,413,291,514,248,049,241,470,885,236
12	64,528,039,343,270,018,963,357,185,158,482,118
13	69,450,664,761,521,361,664,274,701,548,907,358,996,488
14	227,449,714,676,812,739,631,826,459,327,989,863,387,613,323,440
15	2,266,745,568,862,672,746,374,567,396,713,098,934,866,324,885,408,319,028

程として表現することにある．**マルコフ性**とは，次の選択が現在の状態にのみ依存し，過去の選択 $y_0, y_1, \ldots, y_{t-1}$ には依存しないという性質であり，

$$\Pr(y_{t+1} | y_t, y_{t-1}, \ldots, y_0) = \Pr(y_{t+1} | y_t) \tag{3.30}$$

と表される．図 3.2 のようなネットワークでの経路選択をこのマルコフ連鎖モデルで表現すると，毎回の選択肢集合は高々隣接ノード数に限定されるため，選択肢集合の爆発を回避できる．現象論的には，毎交差点で次の進行方向を選択するような意思決定過程を記述したモデルと言える．

しかしより現実的には，経路選択者は目の前の交差点における次のリンクだけでなく，目的地までの下流リンクの効用も考慮して経路を選択していると考えられる．この先読みを伴う逐次的な最適意思決定を定式化したモデルが，Fosgerau et al. (2013) [6] による Recursive Logit（RL）モデルである．

(a)　Recursive Logit モデルの定式化

有向ネットワーク $\mathcal{G} = (\mathcal{A}, \mathcal{N})$（$\mathcal{A}$: リンク集合，$\mathcal{N}$: ノード集合）を考える．RL モデルでは，経路 $\sigma = [a_1, \ldots, a_J]$ の選択確率を，マルコフ性の仮定のもとでリンク遷移確率の積として表す．

$$\mathbb{P}(\sigma = [a_1, \ldots, a_J]) = \prod_{j=1}^{J-1} p(a_{j+1} \mid a_j) \tag{3.31}$$

各リンク遷移において，リンク k にいるドライバーは接続リンク集合 $\mathcal{A}(k)$ から次のリンク a を選択する．このとき，リンク間遷移に関する直接的な効用（遷移効用）$v(a \mid k; \theta)$ と，リンク a から目的地 d までの最大期待効用 $V^d(a)$（価値関数）の和を最大化する．価値関数 V^d は以下の Bellman 方程式で定式化される．

$$V^d(k) = \mathbb{E}\left[\max_{a \in \mathcal{A}(k)} \left\{ v(a \mid k; \theta) + V^d(a) + \mu\varepsilon(a) \right\} \right], \quad \forall k \in \mathcal{A} \tag{3.32}$$

ここで $v(a \mid k; \theta) = \boldsymbol{\theta}^\top \boldsymbol{x}_{a|k}$ は遷移効用の確定項（$\boldsymbol{x}_{a|k}$: リンクの組 (k, a) の属性変数ベクトル，$\boldsymbol{\theta}$: 推定パラメータベクトル），ε は平均 0 の i.i.d. ガンベル分布に従う誤差項である．目的地リンク d

は後続リンクを持たないため $V^d(d) = 0$ とする．

誤差項にガンベル分布を仮定しているため，前節までと同様の Logit 構造が得られ，リンク k からリンク a への遷移確率は MNL 型の式で記述される．

$$p(a \mid k) = \frac{\exp\big(\frac{1}{\mu}\{v(a \mid k;\theta) + V^d(a)\}\big)}{\displaystyle\sum_{a' \in \mathcal{A}(k)} \exp\big(\frac{1}{\mu}\{v(a' \mid k;\theta) + V^d(a')\}\big)} \tag{3.33}$$

また，ガンベル分布の最大値の性質（ログサムの構造）により，式 (3.32) の価値関数は以下のように変形できる．

$$V^d(k) = \begin{cases} \mu \ln \displaystyle\sum_{a \in \mathcal{A}(k)} \exp\big(\frac{1}{\mu}\{v(a \mid k;\theta) + V^d(a)\}\big), & k \neq d \\ 0, & k = d \end{cases} \tag{3.34}$$

これは第 2 節で導入したログサム変数と同じ構造であり，ネスト内の選択肢の「魅力度」を集約する操作が，ここでは下流リンク群の期待効用の集約として再帰的に適用されている．

(b)　価値関数の行列表現と求解

リンク選択確率の計算には式 (3.34) の価値関数を求解する必要がある．式 (3.34) の両辺に指数関数を取り $z_k = e^{V^d(k)/\mu}$ とおくと，以下の行列表現が得られる．

$$\boldsymbol{z} = \boldsymbol{M}\boldsymbol{z} + \boldsymbol{b} \tag{3.35}$$

ここで $\boldsymbol{M}$（$|\mathcal{A}| \times |\mathcal{A}|$）の各要素は $M_{ka} = \delta(a \mid k) \exp\big(\frac{1}{\mu} v(a \mid k;\theta)\big)$ であり，$\delta(a \mid k)$ は $a \in \mathcal{A}(k)$ のとき 1，それ以外で 0 となる接続条件の指示変数である．$\boldsymbol{b}$ は目的地リンクに対応する要素のみ 1，それ以外は 0 のベクトルである．式 (3.35) は連立一次方程式であり，反復計算によって価値関数を求めることができる．

(c)　RL モデルの意義と一般化

RL モデルは，将来期待効用である価値関数をリンク効用に加えることで，逐次的な判断と先読み行動を同一の枠組みで記述するモデルである．ただし，標準的な RL モデルでは価値関数 V^d と遷移効用 v を等価な重みで評価するため，選択確率はプレトリップ型の MNL 経路選択モデルと一致することが知られている．すなわち，ネットワーク全域のリンク効用を大局的に認知・評価可能な状態での先読み行動を表しており，習慣的なドライバーの経路選択行動しか記述できない．

Oyama and Hato (2017) [14] はこの点に着目し，将来期待効用に空間割引率 β（$0 \leq \beta \leq 1$）を導入した一般化 RL モデルを提案した．Bellman 方程式は以下のように拡張される．

$$V^d(k) = \mathbb{E}\left[\max_{a \in \mathcal{A}(k)} \big\{v(a \mid k;\theta) + \beta V^d(a) + \mu\varepsilon(a)\big\}\right] \tag{3.36}$$

$\beta = 1$ のとき標準的な RL モデル（大局的な意思決定）に一致し，$\beta = 0$ のときドライバーは直近の遷移効用 v のみに基づいて近視眼的にリンクを選択する．$0 < \beta < 1$ では両者の中間的な意思決定を表現でき，β を推定することで意思決定の動学性を定量的に把握できる．

3.3.3　時間構造化ネットワークと Recursive Logit

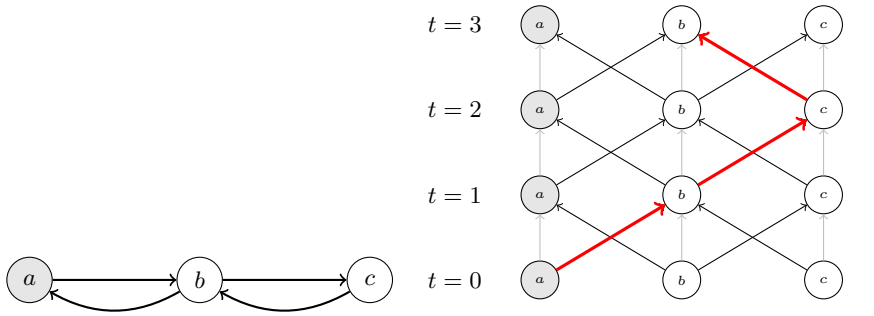
前節の RL モデルは強力な枠組みであるが，標準的な空間ネットワーク上の定式化にはいくつかの実用上の課題がある．第一に，ネットワークにサイクリック構造（同じノードを再訪する経路）が存在する場合，価値関数の求解が困難になる．式 (3.35) の行列 $\boldsymbol{M}$ のスペクトル半径が 1 以上となると，連立方程式の解が発散し得るからである．ここでスペクトル半径とは，行列 $\boldsymbol{M}$ の固有値の絶対値の最大値 $\rho(\boldsymbol{M}) = \max_i |\lambda_i|$ のことである．式 (3.35) を反復的に解く過程は $\boldsymbol{z}^{(n+1)} = \boldsymbol{M}\boldsymbol{z}^{(n)} + \boldsymbol{b}$ と表されるが，$\boldsymbol{M}$ を固有分解すると，各固有値成分が反復ごとに λ_i^n 倍される．したがって，$|\lambda_i| \geq 1$ となる固有値が一つでも存在すると，その成分が反復とともに減衰せず発散するため，解が収束しない．第二に，出発地と目的地が同じ回遊行動や，途中で立ち寄りを伴う経路など，サイクルを含む現実的な行動パターンを扱えない．第三に，時間帯による施設の利用可能性（店舗の営業時間など）や，時間制約のもとでの意思決定を明示的にモデルに組み込むことができない．

これらの問題を根本的に解決するのが，**時間構造化ネットワーク**（time-structured network）の導入である [21, 16].

(a)　時間構造化ネットワークの定義

時間構造化ネットワークとは，空間ネットワークに離散時間軸を加えて展開したネットワークである．空間集合 $\mathcal{A}$（ノードまたはリンク）と離散時点 $t = 0, 1, \ldots, T$ に対し，時間構造化ネットワーク $\mathcal{G}^{\mathrm{TS}} = (\mathcal{S}, \mathcal{E})$ を定義する．ここで $\mathcal{S}$ は**状態**（state）の集合であり，各状態は時点と空間位置の組 $s_t = (t, a)$（$a \in \mathcal{A}$）で表される．$\mathcal{E}$ はアーク（状態間の遷移）の集合であり，$e_t = (s_t, s_{t+1})$ の形をとる．T は時間制約（終了時点）を表す．

時間構造化ネットワーク上の経路は，状態の系列 $\psi = [s_0, s_1, \ldots, s_T]$ として記述される．ここで重要な性質は，各状態が時点 t を含むため，**いかなる経路においても同一の状態を 2 度以上通ることがない**という点である．すなわち，空間的には同じノードを複数回訪れるサイクリック経路（回遊行動）を表現できる一方で，時間構造化ネットワーク自体にはサイクリック構造が存在しない（図 3.3）.



(A) 空間ネットワーク（サイクルあり）　(B) 時間構造化ネットワーク（DAG）

図 3.3: 空間ネットワークと時間構造化ネットワーク．(A) では $a \to b \to a$ のようなサイクルが存在するが，(B) では時間軸の展開によりサイクルが除去される．赤線は活動経路 $[a, b, c, b]$ の例（$t = 2$ で c を訪問後，$t = 3$ で b に戻る）.

(b)　プリズム制約による選択肢集合の限定

時間構造化ネットワークのもう一つの重要な特徴は，**時空間プリズム制約**（time-space prism constraint）を自然に組み込めることである．時空間プリズムとは，Hägerstrand (1970) によって提唱された概念であり，起点 o から出発して終点 d に時刻 T までに到着するという制約のもとで，個人が到達可能な時空間領域を表す．

具体的には，起点 o から状態 a までの最小ステップ数 $D^o(a)$ と，a から終点 d までの最小ステップ数 $D^d(a)$ を用いて，時点 t における状態 a の**到達可能性**を指示変数 $I_t(a)$ で定義する．

$$I_t(a) = \begin{cases} 1, & \text{if } D^o(a) \le t \text{ かつ } D^d(a) \le T - t \\ 0, & \text{otherwise} \end{cases} \tag{3.37}$$

すなわち，時間制約 T のもとで o を出発して d に帰着可能な場合にのみ $I_t(a) = 1$ となる．さらに，状態 $s_t = a$ から $s_{t+1} = a'$ への遷移の存在条件は次のように書ける．

$$\Delta_t(a' \mid a) = I_t(a) \cdot \delta(a' \mid a) \cdot I_{t+1}(a') \tag{3.38}$$

$\Delta_t(a' \mid a) = 1$ のときのみ遷移が許容される．この制約により，到達不可能な状態が事前に除去され，計算対象となるネットワークの規模が大幅に削減される．

プリズム制約の利点は，物理的に到達不可能な状態を排除するだけでなく，以下のような現実的な制約を統一的に表現できる点にある．

バンドル制約　特定の時間帯に特定の場所に滞在しなければならない制約（例：会議への出席）を表現する．中間地点の通過条件を追加すれば，プリズムを分割して複数区間の制約を課すことができる．

ドメイン制約　特定の時空間領域への立入りを禁止する制約（例：店舗の閉店時間帯にその店舗状態への遷移を不許容とする）を表現する．$I_t(a) = 0$ と設定するだけで実装できる．

(c)　時間構造化ネットワーク上の Recursive Logit

時間構造化ネットワーク上では，RL モデルの Bellman 方程式を以下のように再定式化する．状態 $s_t = a$ における価値関数 $V_t(a)$ は，

$$V_t(a) = \mathbb{E}\left[\max_{a' \in \mathcal{A}_t(a)} \{ v_t(a' \mid a) + \beta V_{t+1}(a') + \mu \varepsilon_{t+1}(a') \} \right] \tag{3.39}$$

ここで $\mathcal{A}_t(a) = \{ a' \in \mathcal{A} \mid \Delta_t(a' \mid a) = 1 \}$ はプリズム制約を満たす遷移先の集合，β（$0 \le \beta \le 1$）は時間割引率である．前節と同様にガンベル分布の仮定のもとで，遷移確率は MNL 型となる．

$$p_t(a' \mid a) = \frac{\exp\big(\frac{1}{\mu}\{ v_t(a' \mid a) + \beta V_{t+1}(a') \}\big)}{\displaystyle\sum_{a'' \in \mathcal{A}_t(a)} \exp\big(\frac{1}{\mu}\{ v_t(a'' \mid a) + \beta V_{t+1}(a'') \}\big)} \tag{3.40}$$

価値関数のログサム表現は以下のようになる．

$$V_t(a) = \begin{cases} \mu \ln \sum_{a' \in \mathcal{A}} \Delta_t(a' \mid a) \exp\bigl(\frac{1}{\mu}\{v_t(a' \mid a) + \beta V_{t+1}(a')\}\bigr), & t \neq T \text{ かつ } a \neq d \\ 0, & t = T \text{ または } a = d \end{cases} \tag{3.41}$$

活動経路 $\psi = [s_0, \ldots, s_T]$ の選択確率は，各時点の遷移確率の積として $p(\psi) = \prod_{t=0}^{T-1} p_t(a_{t+1} \mid a_t)$ で与えられる．

(d) 動的計画法による求解

時間構造化ネットワーク上の RL モデルの最大の計算上の利点は，価値関数の求解が**動的計画法**（Dynamic Programming; DP）による単純な後ろ向き帰納法で完結する点にある．

標準的な空間ネットワーク上の RL モデルでは，ネットワークにサイクリック構造が含まれるため，式 (3.35) の連立方程式を逆行列計算や反復法で解く必要があった．このため，スペクトル半径の条件を満たすようにモデルを制約する必要があり，サイクリック経路を含む現実的なネットワーク構造を扱うことが困難であったのは前節で述べた通りである．しかし時間構造化ネットワークでは，時間軸の導入によりネットワークが有向非巡回グラフ（DAG: Directed Acyclic Graph）となるため，終了時点 T から初期時点 0 に向かって式 (3.41) を順次適用するだけで全ての価値関数が確定する．

具体的なアルゴリズムは以下の通りである．

1. **初期化:** $V_T(a) = 0$（$\forall a \in \mathcal{A}$）とする．
2. **後ろ向き計算:** $t = T-1, T-2, \ldots, 0$ の順に，式 (3.41) を用いて各状態の価値関数 $V_t(a)$ を計算する．
3. **遷移確率の計算:** 得られた価値関数を式 (3.40) に代入し，各時点の遷移確率を求める．

この後ろ向き帰納法の計算量は $O(T \cdot |\mathcal{A}|^2)$ であり，反復計算や収束判定が不要なため，実装も容易である．

(e) 時間構造化の意義のまとめ

以上をまとめると，時間構造化ネットワークの導入には以下の利点がある．

1. **サイクリック経路の扱い:** 空間的なサイクル（回遊行動）を保持しつつ，ネットワーク構造としては DAG となるため，価値関数の求解が保証される．
2. **DP による効率的な求解:** 反復計算や逆行列計算が不要となり，後ろ向き帰納法で厳密に解ける．
3. **時空間制約の明示的な導入:** プリズム制約・バンドル制約・ドメイン制約を指示変数 $I_t(a)$ の設定だけで柔軟に表現できる．
4. **時間依存の効用・パラメータ:** 遷移効用 v_t や割引率 β を時点に依存させることで，時間帯による行動変化や営業時間制約を自然に反映できる．

3.3.4　摂動効用型経路選択モデル

前節の Recursive Logit モデルは，ARUM の枠組みのもとで Bellman 方程式を用い，経路の明示的列挙を回避するアプローチであった．本項では，冒頭で導入したもう一つの理論的系譜である摂動効用理論の観点から，選択肢非列挙型の経路選択問題にアプローチするモデルを紹介する．

(a)　摂動効用理論の再訪：経路選択への動機

本章冒頭で述べたように，ARUM では個人の効用に確率的な誤差項を加え，効用最大の選択肢が選ばれる確率を導く．一方摂動効用理論では，需要ベクトル $\boldsymbol{p}$（各選択肢の選択確率）に対して，

$$\max_{\boldsymbol{p}\in\Delta}\left\{\sum_{a} p_a V_a + G^{\bullet}(\boldsymbol{p})\right\} \tag{3.42}$$

という摂動付き期待効用最大化問題を考える．ここで，Δ は単位シンプレックス，$G^{\bullet}$ は摂動項（あるいは正則化項）である．(3.12) と同様に

$$G^{\bullet}(\boldsymbol{p}) = -\mu\sum_{a}\left(p_a \ln p_a - p_a\right) \tag{3.43}$$

を用いると，その最適解として Logit 選択確率が得られる．

ARUM では各選択肢（経路）の効用を個別に評価する必要があるため，何らかの形で経路の列挙，あるいは再帰的構造を用いた計算（例えば Recursive Logit モデル）が必要となる．これに対し摂動効用理論では，決定変数を経路ではなくリンクフローとして定式化することで経路の明示的列挙を回避することができる．

(b)　Perturbed Utility Route Choice（PURC）モデル

Fosgerau et al. (2022) [7] は，この着想を経路選択問題に適用し，**Perturbed Utility Route Choice**（PURC）モデルを提案した．

(b).1　定式化　有向ネットワーク $\mathcal{G}=(\mathcal{N},\mathcal{A})$ 上で，起点 o から終点 d への経路選択を考える．PURC モデルでは，経路の選択確率を直接求めるのではなく，リンクフローベクトル $\boldsymbol{x}=(x_a)_{a\in\mathcal{A}}$ を決定変数とする次の最適化問題を解く．

$$\max_{\boldsymbol{x}\geq\boldsymbol{0}}\left\{\sum_{a\in\mathcal{A}} v_a x_a + G^{\bullet}(\boldsymbol{x})\right\}\quad \text{s.t.}\quad \boldsymbol{A}\boldsymbol{x}=\boldsymbol{b} \tag{3.44}$$

ここで，v_a はリンク a の効用，$G^{\bullet}(\boldsymbol{x})$ は摂動項，$\boldsymbol{A}$ はノード・リンク接続行列，$\boldsymbol{b}$ は各ノードでの外生的な流入出量を表すベクトルである．具体的には，起点 o で $+1$，終点 d で -1，その他のノードで 0 をとり，制約 $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ は各ノードでのフロー保存を表す．

特に，離散選択の場合と同様に Shannon entropy 型の摂動項

$$G^{\bullet}(\boldsymbol{x}) = -\mu\sum_{a\in\mathcal{A}}\left(x_a \ln x_a - x_a\right) \tag{3.45}$$

を用いると，誘導される経路フロー配分，ひいては経路選択行動は Logit 型の選択構造と整合的になる．

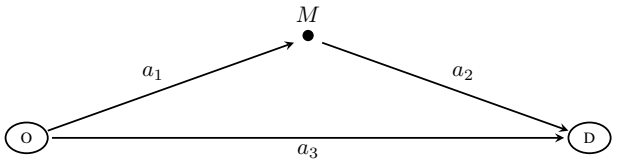


図 3.4: 3 ノード・3 リンクネットワークの例

なお，原著 [7] では，凸解析の都合上，G を凸関数として定義し，その凸共役を通じて間接効用や選択行動を記述している．本稿の記法では最大化問題に合わせて $G^{\bullet}$ を摂動項として書いているため，符号の取り方は原著と必ずしも一致しないが，役割としては同じである．また，原著では対応する凸関数に対して強凸性などの条件が課されている．

この定式化の本質は，**経路空間上の選択問題を，フロー保存制約のもとでのリンク空間上の最適化問題に変換している**点にある．リンク数はネットワーク規模に対して概ね線形に増加する一方，経路数は一般に指数的に増大しうる．したがって，この変換により，経路を明示的に列挙することなくネットワーク全体の選択行動を扱える点が，PURC モデルの大きな利点である．

(b).2　具体例：3 ノード・3 リンクネットワーク　図 3.4 のような単純なネットワークで各成分を確認する．ノード集合 $\mathcal{N} = \{O, M, D\}$，リンク集合 $\mathcal{A} = \{a_1, a_2, a_3\}$ とし，起点 O から終点 D への経路選択を考える．このネットワークには経路 1（$a_1 \to a_2$）と経路 2（a_3）の 2 本の経路が存在する．

リンクフローベクトル $\boldsymbol{x} = (x_{a_1}, x_{a_2}, x_{a_3})^{\top}$ に対し，ノード・リンク接続行列 $\boldsymbol{A}$ と需給ベクトル $\boldsymbol{b}$ は以下のように書ける．

$$\boldsymbol{A} = \begin{pmatrix} +1 & 0 & +1 \\ -1 & +1 & 0 \\ 0 & -1 & -1 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} +1 \\ 0 \\ -1 \end{pmatrix} \tag{3.46}$$

ここで $\boldsymbol{A}$ の各行はノード O, M, D に，各列はリンク a_1, a_2, a_3 に対応する．$A_{na} = +1$ はリンク a がノード n から流出することを，$A_{na} = -1$ はノード n へ流入することを表す．フロー保存制約 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ を成分ごとに展開すると，ノード O：$x_{a_1} + x_{a_3} = 1$（総流出量が 1），ノード M：$-x_{a_1} + x_{a_2} = 0$（すなわち $x_{a_1} = x_{a_2}$：通過フローの保存），ノード D：$-x_{a_2} - x_{a_3} = -1$（総流入量が 1）となる．例えば経路 1 に確率 q，経路 2 に確率 $1-q$ が配分される場合，$\boldsymbol{x} = (q,\, q,\, 1-q)^{\top}$ がフロー保存を満たすことが直ちに確認できる．このように，経路上の選択確率がリンクフローとして集約されており，式 (3.44) はこのリンクフロー空間上で直接最適化を行うため，経路の列挙が不要となる．

(b).3　解法：双対問題とノードポテンシャル　PURC モデルのもう一つの重要な特徴は，その解法が最適化問題の双対構造から自然に導かれる点にある．

式 (3.44) に対して，フロー保存制約 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ に対応するラグランジュ乗数 $\boldsymbol{\lambda} = (\lambda_n)_{n\in\mathcal{N}}$ を導入すると，ラグランジアンは

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \sum_{a\in\mathcal{A}} v_a x_a + G^{\bullet}(\boldsymbol{x}) + \boldsymbol{\lambda}^{\top}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) \tag{3.47}$$

と書ける．KKT 条件より，内点解 $x_a > 0$ では

$$v_a + \frac{\partial G^{\bullet}(\boldsymbol{x})}{\partial x_a} - (A^{\top}\lambda)_a = 0 \tag{3.48}$$

が成り立つ．摂動項が (3.45) で与えられるため

$$\frac{\partial G^{\bullet}(\boldsymbol{x})}{\partial x_a} = -\mu \ln x_a$$

であるから，

$$x_a^{*} = \exp\left(\frac{v_a - (A^{\top}\lambda)_a}{\mu}\right) \tag{3.49}$$

を得る．したがって最適リンクフローは，リンク効用 v_a とノードごとの双対変数 λ_n によって陽に表現される．

ここで λ_n はフロー保存制約に付随する Shadow price，つまりノードポテンシャルと解釈でき，Shannon entropy 型のもとでは RL モデルにおける価値関数と類似した役割を果たす（(3.34) 参照）．さらに，式 (3.49) をフロー保存条件 $\boldsymbol{A}\boldsymbol{x}^{*} = \boldsymbol{b}$ に代入することで，$\boldsymbol{\lambda}$ に関する非線形連立方程式が得られる．これを解くことで全てのリンクフローが確定するため，経路を明示的に列挙することなくネットワーク全体のフロー配分を直接求めることができる．この点は，経路集合や再帰方程式に基づいて解を構成する RL 型の経路選択モデルとは異なる，PURC モデルの重要な特徴である．

コラム：ログサム変数の導出 – 小西将真

発表時間内で紹介しきれなかったログサム分布の導出について記載する．最大値の分布もまたガンベル分布で記述できるため，閉じた形で計算できることが大きなメリットである．

1 ログサム分布の導出

1.1 最大効用の分布関数

各選択肢 j に対して効用を

$$U_j = v_j + \tilde{\varepsilon}_j$$

とし、最大効用を

$$M = \max_j U_j$$

とする。

任意の x に対して、

$$\Pr(M \leq x) = \Pr(v_j + \tilde{\varepsilon}_j \leq x,\ \forall j)$$

誤差項が独立であることから、

$$\Pr(M \leq x) = \prod_j \Pr(\tilde{\varepsilon}_j \leq x - v_j) = \prod_j F_j(x - v_j)$$

ここで $\tilde{\varepsilon}_j$ が Gumbel 分布に従うとすると、その累積分布関数は

$$F_j(z) = \exp\left[-\exp\{-\alpha(z-\beta)\}\right]$$

である。

したがって、

$$\Pr(M \leq x) = \prod_j \exp\left[-\exp\{-\alpha(x - v_j - \beta)\}\right]$$

$$= \exp\left[-\sum_j \exp\{-\alpha x + \alpha v_j + \alpha\beta\}\right]$$

ここで

$$\omega(V) := \sum_j e^{\alpha v_j}$$

とおくと、

$$\Pr(M \leq x) = \exp\left[-\exp\{-\alpha x + \alpha\beta + \ln\omega(V)\}\right]$$

1.2 位置母数へのログサムの出現

上式は次のように書き直せる：

$$\Pr(M \leq x) = \exp\left[-\exp\left\{-\alpha\left(x - \beta - \frac{1}{\alpha}\ln\omega(V)\right)\right\}\right]$$

したがって、M は再び Gumbel 分布に従い、その位置母数は

$$\beta^* = \beta + \frac{1}{\alpha}\ln\omega(V) = \beta + \frac{1}{\alpha}\ln\sum_j e^{\alpha v_j}$$

となる。
すなわち、ログサムは最大効用の分布における位置母数として現れる。

1.3 期待値としてのログサム

Gumbel 分布

$$\exp\left[-\exp\{-\alpha(x-\beta^*)\}\right]$$

の期待値は

$$E[M] = \beta^* + \frac{\gamma}{\alpha}$$

である。ただし γ はオイラー定数である。
よって、

$$E\left[\max_j(v_j + \tilde{\varepsilon}_j)\right] = \beta + \frac{\gamma}{\alpha} + \frac{1}{\alpha}\ln\sum_j e^{\alpha v_j}$$

を得る。

3.4 アクティビティモデル

本節では，行動モデルが記述する選択行動を単独のトリップから一日の活動スケジュール全体へと拡張する．前節までは交通手段選択や経路選択といった個別のトリップに関する離散選択を扱ってきたが，そもそもトリップは活動の派生需要であり，個人の行動をより包括的に記述するには活動スケジュール全体を対象とする必要がある．このような視点に立つモデルを総称して **Activity-Based Model; ABMs** と呼ぶ．

本節の構成は以下の通りである．まず 3.4.1 項で ABMs の基本的な考え方を導入する．3.4.2 項では前節までの経路選択モデルを活動スケジューリングへ拡張するアプローチを紹介し，3.4.3 項では日間変動を捉える活動ニーズモデルを概説する．最後に 3.4.4 項で，活動種類の選択（離散選択）と時間配分（連続選択）を同時に扱う離散連続モデルについて，構造型（MDCEV）と誘導型（Tobit）の両系譜を対比しながら解説する．

3.4.1 Activity-Based Models (ABMs) の基本的な考え方

従来の交通需要予測で広く用いられてきた四段階推定法は，生成・分布・分担・配分の各段階を順に解くことで交通需要を推定する．しかしこの枠組みではトリップを独立した単位として扱うため，一日の中で活動がどのように連鎖し，トリップがいかにして生じるかという行動の文脈を十分に捉えることができない．

ABMs は，トリップそのものではなく個人の活動スケジュールを分析の基本単位とする枠組みである．その基本的な考え方は以下の通りである．

派生需要としてのトリップ 個人は通勤，買い物，余暇などの活動に参加するためにトリップを行う．すなわちトリップは活動の派生需要であり，活動スケジュールが決まればトリップのパタンも決まる．

一日の活動スケジュール 個人の行動は「活動種類・開始時刻・終了時刻・活動場所」の系列として記述される．この系列を**活動経路**（Activity Path）と呼ぶ．

制約の考慮 活動スケジュールは時間的制約（一日は 24 時間），空間的制約（移動には時間がかかる），および個人属性による制約（自動車免許の有無など）のもとで決定される．

活動ベースアプローチの利点は，活動間の連鎖や時間配分の制約を明示的に扱える点にある．例えば，通勤帰りに買い物に立ち寄る行動（トリップチェイン）や，勤務時間の変更が余暇活動に与える影響などは，トリップ単位の分析では捉えにくいが，活動スケジュール全体を対象とすることで自然に記述できる．

3.4.2 経路選択モデルの拡張

これまでに導入した経路選択モデルを用いて活動スケジュールを記述することも可能である．

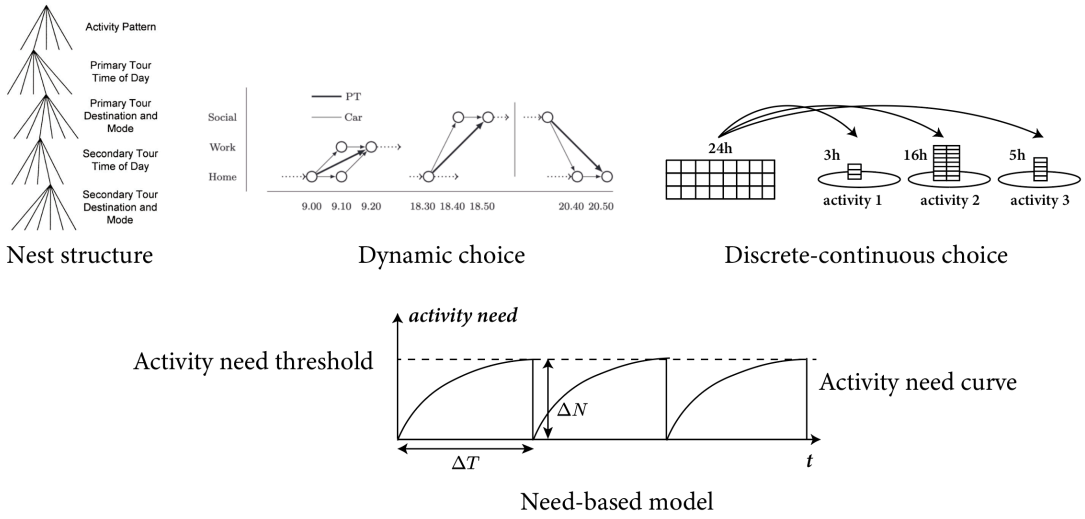


図 3.5: 様々なアクティビティモデルの例．上段左から階層的選択モデルの例（Bowman and Ben-Akiva [4]），逐次的選択モデルの例（Zimmermann et al. [19]），離散連続モデル（MDCEV）[3]），下段は活動ニーズモデル（閾値モデル）の例．

(a)　階層的選択モデル

Bowman and Ben-Akiva [4] に代表される階層的選択モデルは，活動スケジュールの決定を階層的な NL モデルとして記述する．具体的には，まず一日の活動パタン（在宅・通勤・買い物など）を 1 次ネスト，主活動の時間帯を 2 次ネスト，目的地や交通手段を 3 次ネストといった具合に階層構造を設定する．この枠組みは理論的整合性と解釈可能性に優れるが，階層構造の設定は分析者の事前知識に依存するため，複雑な活動パタンの表現には限界がある．

(b)　逐次的選択モデル

Zimmermann et al. [19] は，Recursive Logit モデル [6, 14] を活動スケジューリングに適用し，活動の逐次的な選択過程を記述するモデルを提案した．このアプローチでは，各時点において次に行う活動を逐次的に選択していくことで活動スケジュールを生成する．前節で紹介した経路非列挙型モデルと同様の発想であり，活動経路の列挙を必要とせず，選択肢集合の爆発を回避できるという利点がある．

3.4.3　日間変動を捉えるモデル：活動ニーズモデル

上記モデルは単独日の活動スケジューリングを対象としているが，実際の個人の活動スケジュールは日ごと (day-to-day) に変動する．例えば毎日の通勤パタンは概ね規則的であるが，買い物や余暇活動は曜日や直近の活動履歴によって変化する．このような日間変動を捉えるための代表的なアプローチとして，活動ニーズ（Activity Needs）に基づくモデルがある [2, 1]．このモデルでは，各活動種類に対して「前回の実施からの経過時間」に応じて活動ニーズが蓄積し，ニーズが閾値を超えたときに活動が発生すると仮定する．また Habib [8] は活動ニーズの充足を効用として定式化し，効用最大化の枠組みの中で活動発生を記述するモデルを提案した．これらのモデルは日間変動の構

造を明示的に表現できる一方，閾値やニーズ蓄積関数の設定が分析者の事前知識に依存すること，明示的な構造を仮定するために柔軟な代替パタンの表現が難しいことが課題として挙げられる．

3.4.4 離散連続モデル

前項までのモデルは主に「どの活動を行うか」という離散選択に焦点を当てていた．しかし，活動スケジューリングにおいては活動種類の選択（離散選択）と同時に各活動への時間配分（連続選択）も決定される．このような離散選択と連続選択を同時に扱うモデルを**離散連続モデル**（Discrete-Continuous Model）と呼ぶ．

(a) Corner Solution の問題

離散連続モデルの必要性を理解するために，まず **Corner Solution** の問題を考える．

個人が一日の時間を複数の活動種類に配分する状況を考えよう．全部で K 種類の活動が存在するとき，個人は各活動 k に対して時間 $t_k \geq 0$ を配分し，$\sum_{k=1}^{K} t_k = T$（T は総利用可能時間）という制約のもとで効用を最大化する．

ここで重要な点は，多くの活動種類に対して配分時間が $t_k = 0$ となることである．例えば，ある日に買い物をしなかった個人にとって，買い物への時間配分は $t_k = 0$ である．このように最適解が定義域の境界上に位置する状況を Corner Solution と呼ぶ．

通常の連続最適化では内点解を前提とするため，Corner Solution を適切に扱うことができない．離散連続モデルはこの問題に対処するために開発されたものであり，「どの活動を行うか（離散選択）」と「行う活動にどれだけ時間を配分するか（連続選択）」を統一的な枠組みで記述する．

(b) 離散連続モデルの系譜：構造型と誘導型

福田・力石 [22] による包括的なレビューに従えば，離散連続モデルは大きく二つの系譜に分類される（図 3.6）．

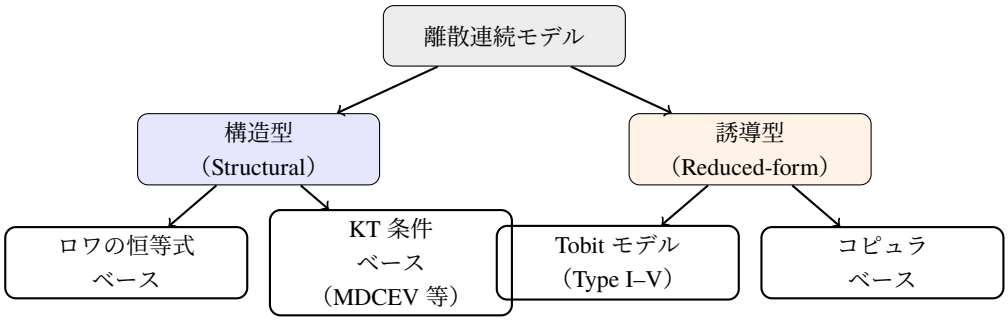


図 3.6: 離散連続モデルの系譜．福田・力石 [22] に基づく分類．

構造型（Structural Model） ミクロ経済理論に立脚し，資源制約下での効用最大化問題から演繹的にモデルを導出する．導出の方法として，条件付き間接効用関数にロワの恒等式を適用して需要関数を得る方法（Dubin and McFadden [5] 等）と，直接効用関数の最大化問題にキューン・タッカー（KT）条件を適用する方法（Wales and Woodland [18]，Bhat [3] 等）がある．経済理論との整合性が高い一方で，効用関数の関数形が制約されるため，現象記述の自由度が

限られる．特に，離散選択と連続選択に同一の評価関数を用いるため，「活動頻度は低いが 1 回あたりの活動時間は長い」といった離散・連続間の非対称な関係の記述が困難である．

誘導型（Reduced-form Model） 統計モデルによる現象の直接的記述に主眼を置く．代表的なものとして，Tobin [17] に端を発する Tobit モデルやその拡張（Type I–V），コピュラ関数を用いた誤差相関の柔軟な記述を行うモデルがある．資源制約を明示的に課さないため現象記述の自由度が高く，離散選択と連続選択に異なる評価関数を用いることもできる．一方で，経済理論との整合性は必ずしも保証されない．

以下ではこの分類に従い，構造型の代表として MDCEV モデルを，誘導型の代表として Type II Tobit モデルをそれぞれ解説し，両者の違いと使い分けを論じる．

(c)　構造型：MDCEV モデル

構造型離散連続モデルの中でも広く用いられているのが，Bhat [3] によって提案された **Multiple Discrete-Continuous Extreme Value**（MDCEV）モデルである．MDCEV モデルは KT 条件に基づく構造型モデルであり，K 個の財（活動種類）の消費量を資源制約のもとで同時に決定する．

個人 i は以下の直接効用関数を資源制約のもとで最大化する．

$$\max_{\boldsymbol{t}} \; U(\boldsymbol{t}) = \sum_{k=1}^{K} \frac{\gamma_k}{\alpha_k} \psi_k \left[\left(\frac{t_k}{\gamma_k} + 1 \right)^{\alpha_k} - 1 \right], \quad \text{s.t.} \quad \sum_{k=1}^{K} t_k = T, \; t_k \geq 0 \tag{3.50}$$

ここで $\boldsymbol{t} = (t_1, \ldots, t_K)$ は各活動への時間配分ベクトル，$\psi_k = \exp(\boldsymbol{\beta}^\top \boldsymbol{x}_k + \varepsilon_k)$ は活動 k の基本効用（ベースライン効用），$\gamma_k > 0$ は活動 k の消費単位を調整する変換（translation）パラメータ，$\alpha_k \leq 1$ は飽和パラメータである．

KT 条件を適用すると，活動 k に正の時間が配分される（$t_k > 0$）ための条件は，その活動の限界効用が Shadow price（ラグランジュ乗数）λ 以上であることとなる．

$$\frac{\partial U}{\partial t_k} = \lambda \quad \text{if } t_k > 0 \tag{3.51}$$

$$\frac{\partial U}{\partial t_k} < \lambda \quad \text{if } t_k = 0 \tag{3.52}$$

すなわち，限界効用が Shadow price に達しない活動には時間が配分されず（$t_k = 0$），Corner Solution が内生的に生じる．

この効用関数の特徴は以下の通りである．

飽和効果 $\alpha_k < 1$ のとき，活動 k への時間配分が増加するにつれて限界効用が逓減する．これにより，一つの活動に全ての時間を配分するのではなく，複数の活動に時間を分散させる行動が自然に表現される．

Corner Solution の許容 $t_k = 0$ が最適解となることを KT 条件によって内生的に許容する．基本効用 ψ_k が十分に小さい活動には時間が配分されないという行動を記述できる．

閉じた形の選択確率 誤差項 ε_k に i.i.d. ガンベル分布を仮定することで，どの活動の組合せが選択されるかの確率が閉じた形で得られる．

MDCEV モデルは活動種類と時間配分の同時決定を統一的に扱えるため，アクティビティモデルにおける時間配分の記述に広く応用されている [3, 9]．その一方で，**離散選択と連続選択が同一の効用関数から導出される**という構造的制約があり，両者に異なるメカニズムが働く現象の記述には注意が必要である．

(d) 誘導型：Tobit モデルの体系

誘導型の離散連続モデルの原型は，Tobin [17] が提案した Tobit モデルである．Tobit モデルは，潜在変数が閾値を超えた場合にのみ正の値が観測されるという**打ち切り**（censoring）の構造を持つ．福田・力石 [22] の整理に従えば，Tobit モデルには以下の類型がある（表 3.3）．

表 3.3: Tobit モデルの類型（福田・力石 [22] に基づく）

類型	離散問題と連続問題の関係	代表的研究
Type I	同一の評価関数・同一のパラメータ	Tobin (1958) [17]
Type II	異なる評価関数を許容	Heckman (1979) [10]
Type V	2 項選択それぞれに連続問題が観測	Lee (1983)

(d).1 Type I Tobit モデル は最も基本的な形であり，離散問題と連続問題を同一の評価関数で記述する．

$$t_k = \max(0,\, t_k^*), \quad t_k^* = \boldsymbol{\beta}^\top \boldsymbol{x}_k + \varepsilon_k \tag{3.53}$$

潜在変数 $t_k^* > 0$ のとき活動が実行され，その値がそのまま活動時間として観測される．端点解（$t_k = 0$）を離散選択問題と解釈すれば，KT 条件に基づく構造型モデルと同様の性質を持つ．しかし離散選択と連続選択に同一のパラメータ $\boldsymbol{\beta}$ を用いるため，「活動の実行しやすさ」と「実行した場合の時間の長さ」に異なる要因が影響する状況を記述できない．

(d).2 Type II Tobit モデル（Heckman モデル） は，離散問題と連続問題に**異なる評価関数**を許容するモデルであり，Heckman [10] によって提案された．離散選択（活動を実行するか否か）と連続選択（活動時間）にそれぞれ独立な潜在変数を設定する．

$$y_{\text{act}}^* = \boldsymbol{\alpha}^\top \boldsymbol{w} + \epsilon_{\text{act}} \tag{3.54}$$

$$y_{\text{time}}^* = \boldsymbol{\beta}^\top \boldsymbol{x} + \epsilon_{\text{time}} \tag{3.55}$$

$$y_{\text{time}} = \begin{cases} y_{\text{time}}^*, & \text{if } y_{\text{act}}^* > 0 \\ 0, & \text{if } y_{\text{act}}^* \le 0 \end{cases} \tag{3.56}$$

ここで y_{act}^* は活動実行に関する潜在変数（判別関数），y_{time}^* は活動時間に関する潜在変数であり，$\boldsymbol{w}$ と $\boldsymbol{x}$ はそれぞれ異なる説明変数ベクトル，$\boldsymbol{\alpha}$ と $\boldsymbol{\beta}$ は異なるパラメータベクトルである．誤差項 $\epsilon_{\text{act}}, \epsilon_{\text{time}}$ は二変量正規分布に従い，相関係数 ρ を持つ．

Type II Tobit モデルの重要な特徴は以下の点にある．

異なる評価関数 離散選択と連続選択にそれぞれ独立な説明変数とパラメータを設定できる．これにより，「活動の発生確率は高いが 1 回あたりの活動時間は短い」といった，離散・連続間の非対称な関係を自然に記述できる．

誤差相関 $\rho \neq 0$ のとき，離散問題と連続問題の誤差項に相関があり，活動実行の判断と活動時間の決定が共通の非観測要因によって結びつけられる．$\rho = 0$ かつ同一の評価関数を用いた場合，Type I に帰着する．

資源制約の不要 構造型のように総時間制約 $\sum t_k = T$ を課す必要がなく，個々の活動の実行・時間決定を逐次的に記述できる．

(e)　活動欲求を判別関数とした Type II Tobit モデルの応用

大山ら [20] は，Type II Tobit モデルの判別関数に**活動欲求**（Activity Needs）の概念を導入し，まちなかの回遊行動における活動発生メカニズムを分析した．

このモデルでは，個人がまちなかに流入してから退出するまでの一連の活動を対象とし，活動欲求の「閾値」と「充足度」の差として活動発生を記述する．n 回目の活動終了後の活動欲求 $V_{\mathrm{act},n}$ は以下で定義される．

$$V_{\mathrm{act},n} = N^* - N_n, \quad N_n = \sum_{t=1}^{n} \left(\beta_{\mathrm{act}} T_t + \sum_i \gamma_{ti} I_{ti} \right) + \sum_j \delta_{nj} s_{nj} \tag{3.57}$$

ここで N^* は活動欲求の閾値（個人属性・ツアー変数で決定），N_n は n 回の活動によって蓄積された充足度，T_t は t 回目の活動の滞在時間，I_{ti} は活動履歴変数，s_{nj} は空間特性変数である．$V_{\mathrm{act},n} > 0$ であれば活動欲求が残存しており次の活動が発生し，$V_{\mathrm{act},n} \leq 0$ であれば欲求が充足されて回遊が終了する．

この活動欲求 $V_{\mathrm{act},n}$ を Type II Tobit モデルの判別関数として用いることで，以下のモデルが定式化される．

$$y^*_{\mathrm{act}}(n) = V_{\mathrm{act},n} + \epsilon_{\mathrm{act}} \tag{3.58}$$

$$y^*_{\mathrm{time}}(n) = \boldsymbol{\beta}^\top_{\mathrm{time}} \boldsymbol{x}_{\mathrm{time},n} + \epsilon_{\mathrm{time}} \tag{3.59}$$

$$y_{\mathrm{time}}(n) = \begin{cases} y^*_{\mathrm{time}}(n), & \text{if } y^*_{\mathrm{act}}(n) > 0 \\ 0, & \text{if } y^*_{\mathrm{act}}(n) \leq 0 \end{cases} \tag{3.60}$$

尤度関数は，誤差項に二変量正規分布を仮定して以下のように導出される．

$$L = \prod_{y_{\mathrm{act}}=0} \left(1 - \Phi(V_{\mathrm{act},n})\right) \times \prod_{y_{\mathrm{act}}>0} \left[\frac{1}{\sigma} \phi\left(\frac{y_{\mathrm{time}} - \boldsymbol{\beta}^\top_{\mathrm{time}} \boldsymbol{x}}{\sigma} \right) \Phi\left(\frac{V_{\mathrm{act},n} + \sigma^{-1}\rho(y_{\mathrm{time}} - \boldsymbol{\beta}^\top_{\mathrm{time}} \boldsymbol{x})}{\sqrt{1-\rho^2}} \right) \right] \tag{3.61}$$

ここで $\Phi(\cdot), \phi(\cdot)$ はそれぞれ標準正規分布の分布関数と密度関数，σ は ϵ_{time} の標準偏差，ρ は ϵ_{act} と ϵ_{time} の相関係数である．

このモデルの特徴は，活動欲求の充足過程を判別関数に組み込むことで，**活動履歴が次の活動の発生確率に与える影響**と**活動時間の決定**を統一的に扱える点にある．推定の結果，買い物・娯楽活動の累積が次の活動発生を促進すること，主目的地（Main）での活動後に回遊が終了しやすいこと等が明らかにされた．

表 3.4: MDCEV モデルと Type II Tobit モデルの比較

	MDCEV（構造型）	**Type II Tobit**（誘導型）
理論的基盤	KT 条件に基づく効用最大化	統計モデルによる現象記述
評価関数	離散・連続に同一の効用関数	離散・連続に異なる関数を許容
資源制約	$\sum t_k = T$ を明示的に課す	資源制約を必要としない
誤差分布	ガンベル分布（閉じた形の確率）	正規分布（二変量正規で相関を記述）
Corner Solution	KT 条件から内生的に導出	潜在変数の閾値構造で記述
活動間相互作用	飽和パラメータで間接的に表現	判別関数への履歴変数導入で直接表現可能
適用場面	複数活動の同時配分	逐次的な活動実行・時間決定

(f) 構造型と誘導型の対比

以上をまとめると，MDCEV モデルと Type II Tobit モデルは表 3.4 のように対比される．

どちらのモデルを採用すべきかは対象とする現象に依存する [22]．一般に，(a) 経済理論との整合性を重視し，資源制約下での配分問題として記述する場合には KT 条件に基づく構造型が，(b) 現象記述の自由度や観測の不完全性への対処を重視する場合には誘導型が適している．さらに，(c) 両系譜の特徴を折衷した折衷型モデルの構築も試みられており，構造型の理論的基盤と誘導型の記述の柔軟さを両立させる方向が模索されている．

コラム：コロナ前後における活動の変化 – 饗場清日

（1）MDCEV モデルと Recursive Logit Model の比較

構造型モデルである **MDCEV モデル**と **Recursive Logit Model** の比較を行った．**MDCEV モデル**は，1 日を観測単位として，活動への時間配分を「離散的な活動選択＋連続的な時間配分」として定式化するものである．一方で，離散選択モデルである **Recursive Logit Model** は 1 日を時間ステップ列とみなし，1 ステップ列からの遷移列から学習するものである．場所次元と過去活動の効果は両方とも無視するものとし，「時間配分」と「時系列推移」の 2 つのモデルでそれぞれ何を表現するか，できるかを比較する．東京都市圏で実施された 4 つの PP データを使用するものとする．

period	期間	文脈	n_users (raw)
1	2019-07-08 .. 2020-02-11	**コロナ前**	305
2	2021-10-03 .. 2021-11-25	**コロナ規制明け直後**	115
3	2022-12-19 .. 2023-01-30	コロナ後・冬	182
4	2023-11-15 .. 2023-11-29	コロナ後・秋	223

表 3.5: 東京都圏で実施された 4 つの PP 調査

活動カテゴリとしては, H(home), S(Shopping), W(Commute/Work), W2(Business), O(Other), 予算制約は $T = 24\text{h}$ とする．

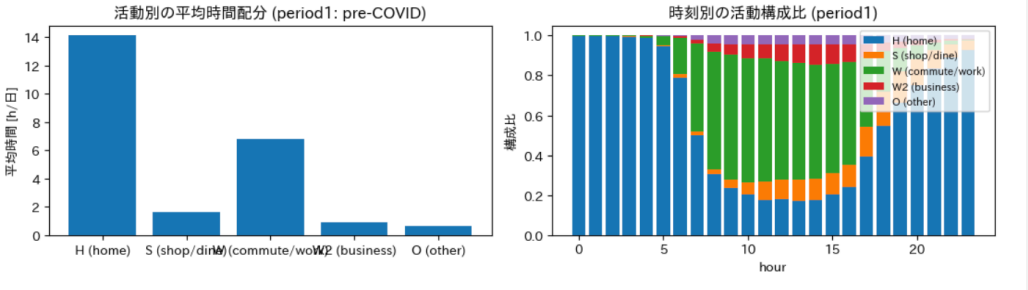


図 3.7: 実データの活動別グラフ

（2）MDCEV モデル（γ-profile, Bhat 2008）

各 out 活動について，

$$V_k = \psi_k - \log(x_k/\gamma_k + 1) \tag{3.62}$$

ここで，ψ_k：ベース効用，x_k：費やした時間，γ_k：飽和パラメータである．

選択確率は，以下の閉形式で表される．

$$P(x) = \left[\prod_{k \in M} \frac{1}{x_k + \gamma_k}\right] \cdot \left[\sum_{k \in M} (x_k + \gamma_k)\right] \cdot \left[\prod_{k \in M} \frac{e^{V_k}}{\sum_j e^{V_j}}\right] \tag{3.63}$$

（3）Recursive Logit Model

時間 t における活動，遷移効用を「1 時間ごとに次の行動を決める」形で捉えるものとする．

$$U_t(i \to j) = \alpha_j + \delta_{\text{stay}} 1[j = i] + \delta_{\text{morn}} 1[i = H, j \neq H, 6 \leq t < 10] + \delta_{\text{eve}} 1[i \neq H, j = H, 17 \leq t < 22] \tag{3.64}$$

（4）分析結果

MDCEV モデル

	param	estimate	std_err	t_value
0	psi_S	-0.7392	0.0506	-14.62
1	psi_W	0.0295	0.0465	0.63
2	psi_W2	-1.7580	0.0701	-25.09
3	psi_O	-1.9597	0.0758	-25.86
4	log_gamma_S	1.1533	0.0736	15.67
5	log_gamma_W	2.2314	0.0722	30.91
6	log_gamma_W2	1.8356	0.1202	15.27
7	log_gamma_O	1.3246	0.1281	10.34

図 3.8: MDCEV モデル pre コロナ_MDCEV モデル推定表

ベース効用：H は基準で 0 であり，ほとんどの活動で 0 以下なので何もしなければ選ばれにくいことがわかる．

飽和パラメータ：全てプラスの値であり，値が大きいほど，長く続く行動であることがわかる．

t 値：絶対値が 1.96 を越えていればそのパラメータは 5 %有意水準で有意である．

仕事のベース効用が唯一正の値であり大きいので，仕事に行く動機は他の活動と比べて強い．飽和パラメータが大きいので，仕事は一度始めると比較的長く続くが，買い物，その他は比較的短く切り上げる傾向がある．仕事の t 値は低いが，自宅との値の差が小さいためと考えられる．

Recursive Logit Model

	param	estimate	std_err	t_value
0	alpha_S	-0.2119	0.0102	-20.79
1	alpha_W	-0.0341	0.0062	-5.49
2	alpha_W2	-0.3249	0.0141	-22.97
3	alpha_O	-0.4137	0.0182	-22.79
4	delta_stay	3.6363	0.0251	145.12
5	delta_morn	2.3296	0.0608	38.35
6	delta_eve	2.3148	0.0592	39.08

図 3.9: RecursiveLogitModel_推定表

ベース効用：H が基準で 0 以下なので何もしなければ選ばれにくい．
活動継続のボーナス：同じ活動を続けることへのボーナスである．
朝の外出ダッシュ：6〜10 時の間に自宅を出るボーナスである．
夕方の帰宅ダッシュ：17〜22 時の間に自宅に帰るボーナスである．
δ_{stay} の t 値が突出して高く，一度始めた活動は継続し続ける傾向があると言える．また，朝の外出と夕方の帰宅に強いバイアスがあることがわかる．

再現性

モデルから観測時間配分の再現を試みると，RL では推定パラメータのもとで各 user, day を $t = 0$ から forward simulate する．ほとんど完全に表現できていると言える．朝の外出項などが時間制約の学習に役立っていることがわかる．

	observed [h]	RL simulated [h]
H (home)	14.23	14.69
S (shop/dine)	1.65	1.67
W (commute/work)	6.61	6.13
W2 (business)	0.92	0.93
O (other)	0.59	0.59

図 3.10: RL モデル推定表

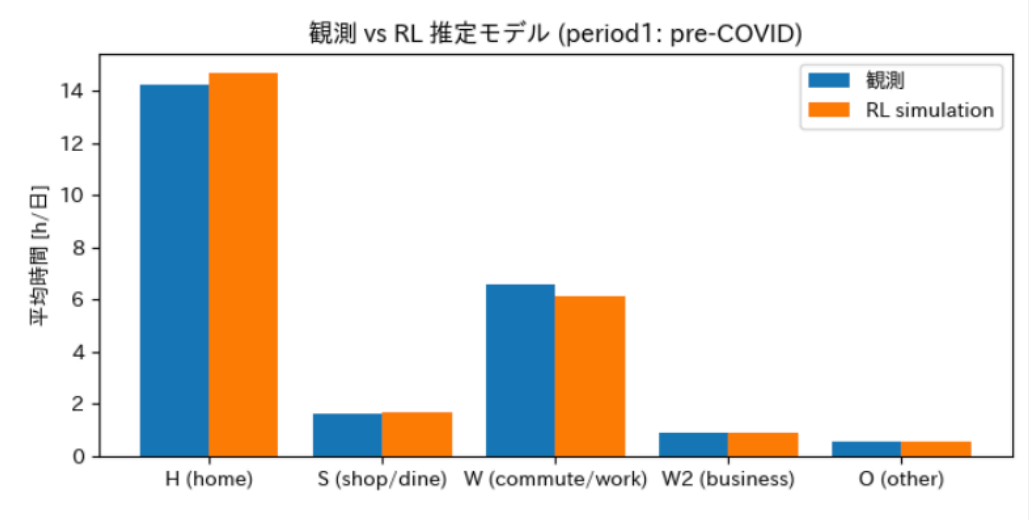


図 3.11: 観測と RL モデルの比較

MDCEV と RL の比較

両方のモデルから観測時間配分の再現を試す．MDCEV モデルで推定された γ_k から求まる期待時間と，RL モデルで推定された α_k を比較する．観測モデルが異なるため直接比較することはできないが，パラメータの符号と各活動の相対的な魅力度の傾向（γ_k，α_k）は一致する．

	param	estimate	std_err	t_value
0	psi_S	-0.7392	0.0506	-14.62
1	psi_W	0.0295	0.0465	0.63
2	psi_W2	-1.7580	0.0701	-25.09
3	psi_O	-1.9597	0.0758	-25.86
4	log_gamma_S	1.1533	0.0736	15.67
5	log_gamma_W	2.2314	0.0722	30.91
6	log_gamma_W2	1.8356	0.1202	15.27
7	log_gamma_O	1.3246	0.1281	10.34

	param	estimate	std_err	t_value
0	alpha_S	-0.2119	0.0102	-20.79
1	alpha_W	-0.0341	0.0062	-5.49
2	alpha_W2	-0.3249	0.0141	-22.97
3	alpha_O	-0.4137	0.0182	-22.79
4	delta_stay	3.6363	0.0251	145.12
5	delta_morn	2.3296	0.0608	38.35
6	delta_eve	2.3148	0.0592	39.08

図 3.12: MDCEV と RL 推定表の比較

（5）時期による比較

実データを用いた比較

・家にいる時間はコロナ規制後，直後ではなく少し経ってから 1 時間程度長くなっており，仕事・通勤時間は減少した．コロナ規制後から普及した各種オンラインサービスにより，対面での仕事が減少したのだろうと考えられる．

・Business（業務）は，コロナ規制直後〜翌年の冬にかけて半減していることがわかるが，翌年の秋には元の水準に復活している．

・買い物に使う時間は 30 分程度増加した．通勤時間が減り，通勤のついでに買い物に行くということが減って，必要な時間が増えたと考えられる．

	Phase1 コロナ前	Phase2 コロナ規制直後	Phase3 コロナ後冬	Phase4 コロナ後秋
サンプル数	6532	2299	1394	1581
observed H(home)	14.23	14.77	15.49	15.45
observed S(shop/dine)	1.65	1.84	1.77	2.19
observed W(commute/work)	6.61	6.54	5.56	4.73
observed O(other)	0.59	0.44	0.77	0.63

表 3.6: 各 Phase における観測活動時間

RL モデルを用いた比較

・コロナ規制後，Work の α 値は下がっている．

・2. コロナ規制直後，3. コロナ後において，ビジネスの α 値が下がっていることがわかる．コロナを受けて暫くは出張，訪問が減っていたと言える．

・4. コロナ後秋の結果を見ると，朝の出発に関する α 値がコロナ前と比べると下がっている．コロナを経てリモートワークが普及したことにより，朝出勤する人の割合が下がったといえる．

・パラメータで見ても Shopping の α_k の値は少し大きくなっている．

	param	estimate	std_err	t_value
0	alpha_S	-0.2119	0.0102	-20.79
1	alpha_W	-0.0341	0.0062	-5.49
2	alpha_W2	-0.3249	0.0141	-22.97
3	alpha_O	-0.4137	0.0182	-22.79
4	delta_stay	3.6363	0.0251	145.12
5	delta_morn	2.3296	0.0608	38.35
6	delta_eve	2.3148	0.0592	39.08

	param	estimate	std_err	t_value
0	alpha_S	-0.2225	0.0097	-22.91
1	alpha_W	-0.0580	0.0061	-9.47
2	alpha_W2	-0.5551	0.0237	-23.42
3	alpha_O	-0.5360	0.0227	-23.57
4	delta_stay	3.5142	0.0251	140.11
5	delta_morn	2.3911	0.0611	39.12
6	delta_eve	2.0468	0.0593	34.52

	param	estimate	std_err	t_value
0	alpha_S	-0.2339	0.0102	-22.85
1	alpha_W	-0.0825	0.0066	-12.46
2	alpha_W2	-0.5482	0.0238	-23.03
3	alpha_O	-0.4069	0.0167	-24.29
4	delta_stay	3.5265	0.0257	137.35
5	delta_morn	2.2846	0.0630	36.25
6	delta_eve	2.1253	0.0618	34.41

	param	estimate	std_err	t_value
0	alpha_S	-0.1874	0.0089	-21.09
1	alpha_W	-0.0906	0.0067	-13.59
2	alpha_W2	-0.3376	0.0137	-24.62
3	alpha_O	-0.4351	0.0179	-24.24
4	delta_stay	3.5514	0.0240	148.00
5	delta_morn	2.0919	0.0598	34.99
6	delta_eve	2.1750	0.0592	36.76

図 3.13: RL モデルによる時期の比較

実データと RL モデルの整合性

・全体的にどの時期のデータについても推定精度は高い.

・4 つ全ての時期のデータについて，H 在宅を 0.5h 程度高く見積もる傾向があった.

・4 つ全ての時期のデータについて，W 通勤・仕事を 0.5h 程度低く見積もる傾向にあった.

・S 買い物，W2 業務，O その他については極めて高い精度で予測できていた.

	observed [h]	RL simulated [h]
H (home)	14.23	14.69
S (shop/dine)	1.65	1.67
W (commute/work)	6.61	6.13
W2 (business)	0.92	0.93
O (other)	0.59	0.59

	observed [h]	RL simulated [h]
H (home)	14.77	15.25
S (shop/dine)	1.84	1.83
W (commute/work)	6.54	6.10
W2 (business)	0.42	0.38
O (other)	0.44	0.44

	observed [h]	RL simulated [h]
H (home)	15.49	15.88
S (shop/dine)	1.77	1.78
W (commute/work)	5.56	5.19
W2 (business)	0.43	0.41
O (other)	0.74	0.74

	observed [h]	RL simulated [h]
H (home)	15.45	15.84
S (shop/dine)	2.19	2.26
W (commute/work)	4.73	4.29
W2 (business)	0.99	1.00
O (other)	0.63	0.60

図 3.14: RL モデルによる時期の比較

（6）MDCEV モデルの forecast

Bhat(2008) の Forecasting Algorithm（決定論的最適配分）を用いる.

・MDCEV モデルでは W(通勤，仕事) を多く見積もる傾向がある.

・RL モデルの方が S，W2，O の活動時間を正確に見積もることができている.

	observed [h]	MDCEV pred (obs active set) [h]	diff
H	14.229	-0.000	-14.229
S	1.648	1.440	-0.208
W	6.615	7.621	1.007
W2	0.918	0.337	-0.581
O	0.590	0.372	-0.218

図 3.15: 実データと RL データの整合性

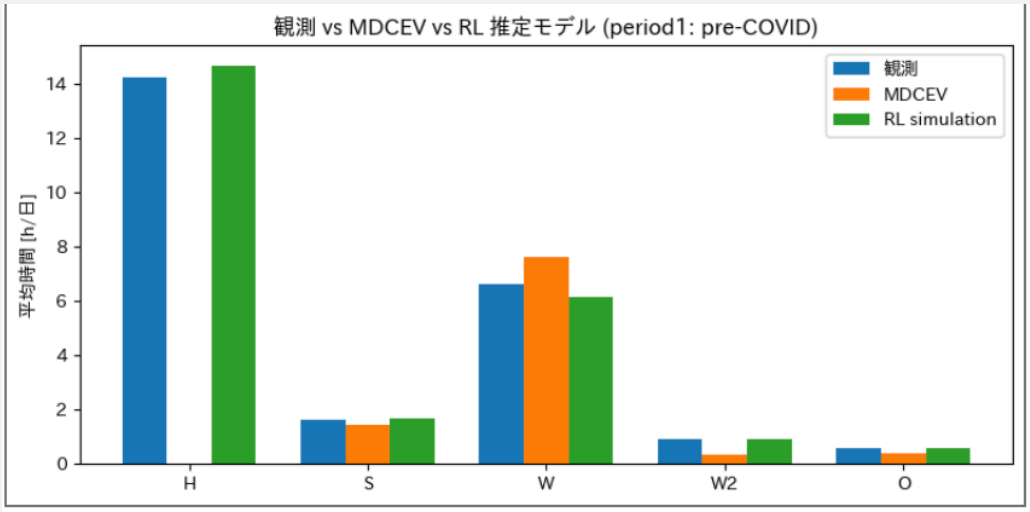


図 3.16: 実データと RL データの整合性

（7）Day-to-Day Dynamics

通常の Recursive Logit Model では，過去の活動履歴を考慮することはできないが，ここに**全ての活動に対して同じパラメータを追加する場合**を考える．1 日の各活動の最後に行った日からの経過日数も考慮に入れる．

$$U_t(i \to j) = [\text{baseline}] + \theta_{\text{rec}} \cdot \text{recency}_u(a(j)) \tag{3.65}$$

	param	estimate	std_err	t_value
0	alpha_S	-0.1449	0.0109	-13.27
1	alpha_W	0.0065	0.0068	0.96
2	alpha_W2	-0.2210	0.0156	-14.13
3	alpha_O	-0.3076	0.0194	-15.89
4	delta_stay	3.6110	0.0251	144.14
5	delta_morn	2.3335	0.0606	38.52
6	delta_eve	2.3115	0.0592	39.04
7	theta_recency	-0.0387	0.0028	-13.84

図 3.17: キャプション

$\theta_{\text{rec}} < 0$ より，普段やらない活動は将来も選ばれにくい傾向にあるといえる．recency を入れた方が，W の時間については正確に推定することができていた．

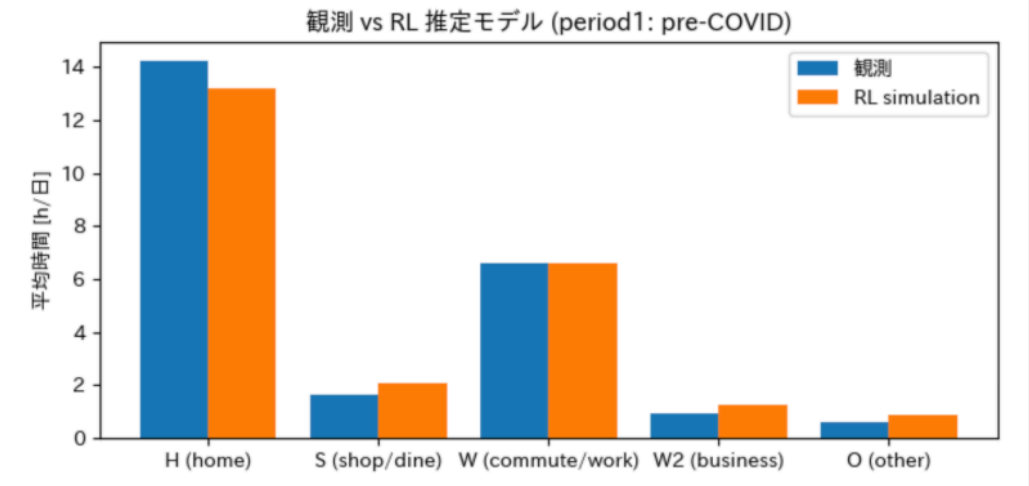


図 3.18: キャプション

活動種類ごとの Day-to-Day Recency を導入する場合

活動ごとの個別のパラメータの導入を行い，各活動ごとに違いを見た．レコードごとに効用が異なるので，N 個の異なる効用関数を解くことで，一人一人の現在の状況に合わせた

将来予測を計算する．

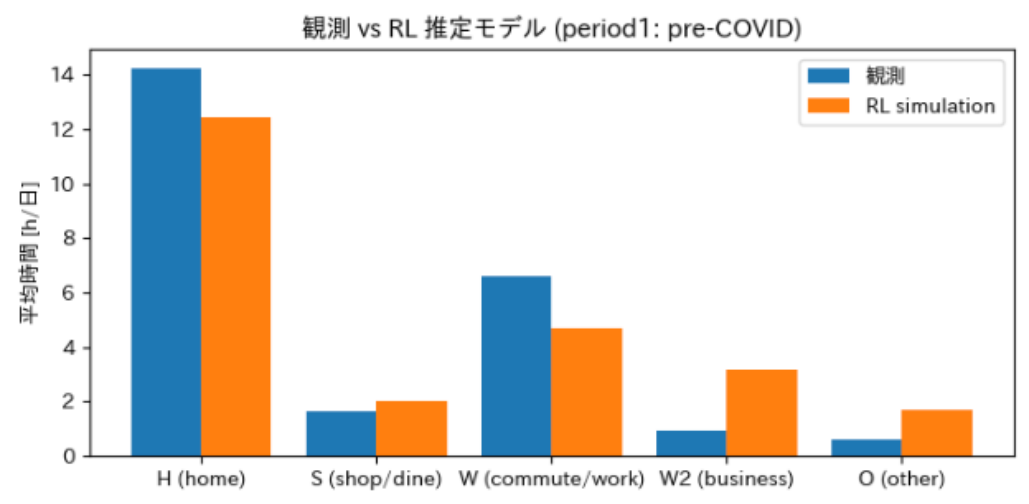


図 3.19: キャプション

モデルのはまり具合は悪化してしまったが，これは W2(Business) は他の活動と比べて発生頻度が低いため，負の相関を強く見積り過ぎた結果，シミュレーション上では W2 が過剰に選ばれる事態が発生してしまったのではないかと考えられる．さらに，サンプル数が少ないため，個人の極端な行動の影響を受けている可能性が高い．

3.5 モデルの推定と検定

前章までで MNL モデルをはじめとした行動モデルを構築したが，行動モデルを用いた分析の真髄は実データをもとにモデルパラメータを推定することにある．モデル推定の目標は効用関数内の未知パラメータ β_i を統計的に尤もらしい値として推定することであるが，同時にその値がどの程度信頼できるのかを検定することも重要である．本章では，効用最大化理論に基づく行動モデルの推定と検定について，特に離散選択モデルを中心に解説する．

本節の構成は以下の通りである．まず 3.5.1 項で尤度関数と最尤推定法の基礎を導入し，3.5.2 項でパラメータの識別可能性について述べる．3.5.3 項では推定結果の統計的信頼性を評価するための t 検定・尤度比検定・擬似決定係数を解説する．最後に 3.5.4 項では，前項までの議論が暗黙に仮定していた「観測データは真の行動を正確に反映している」という前提を問い直し，GPS や BLE などのセンサデータに含まれる観測誤差のもとでの推定手法——経路観測モデル，構造推定，量子化と観測多様体——を紹介する．

3.5.1 尤度と最尤推定

モデル推定の目標は，モデルの出力が実データにできるだけ近くなるようなパラメータ $\boldsymbol{\beta}$ を見つけることである．離散選択モデルでは，この問題は**尤度**を最大化するようなパラメータを求める**最尤推定法**によって解かれる．尤度とは，あるパラメータの値のもとで実際に観測されたデータが得られる確率のことを指し，より正解分布に近い出力を与えるパラメータがより高い尤度を持つ．

MNL モデルにおける尤度関数を導出する．M 人が N 個の選択肢の中から独立に選択行動を行うとし，k 番目の個人にとっての説明変数を $\boldsymbol{x_k}$，選択結果を y_k とする．各人の選択行動が独立であることから，観測データ全体の同時確率は各個人の選択確率の積として表される．

$$P(y_1;\boldsymbol{x_1},y_2;\boldsymbol{x_2},\ldots,y_M;\boldsymbol{x_M})=\prod_{k=1}^{M}P(y_k;\boldsymbol{x_k}) \tag{3.66}$$

これはパラメータ $\boldsymbol{\beta}$ の下で観測データが得られる確率そのものであるから，尤度関数 $L(\boldsymbol{\beta})$ として定義できる．MNL の選択確率を代入すると次のようになる．

$$L(\boldsymbol{\beta})\coloneqq\prod_{k=1}^{M}P(y_k;\boldsymbol{x_k})=\prod_{k=1}^{M}\frac{\exp(V_{y_k})}{\sum_{a=1}^{N}\exp(V_a)} \tag{3.67}$$

この尤度を最大化するパラメータ $\hat{\boldsymbol{\beta}}=\arg\max_{\boldsymbol{\beta}}L(\boldsymbol{\beta})$ を求める手法が最尤推定法である．

実際の計算では，尤度 $L(\boldsymbol{\beta})$ は 1 より小さい値を大量に掛け合わせるため非常に小さな値をとり，数値計算上アンダーフローを起こす恐れがある．そこで尤度の対数を取った**対数尤度**を最大化する（対数は単調増加関数であるため，最大化の結果は変わらない）．

$$LL(\boldsymbol{\beta})\coloneqq\log L(\boldsymbol{\beta})=\sum_{k=1}^{M}\log P(y_k;\boldsymbol{x_k}) \tag{3.68}$$

3.5.2　識別可能性

最尤推定が意味のある結果を返すためには，尤度を最大化するパラメータの組が一意に定まる必要がある．この条件が満たされないとき，尤度関数の最大化が収束せずパラメータ推定に失敗する．

離散選択モデルにおいて識別の問題が典型的に生じるのは，選択肢固有の定数項（ASC: Alternative-Specific Constant）の扱いである．例えば，鉄道 S，バス B，自家用車 P の 3 選択肢に対し，効用の確定項を以下のように立式したとする．

$$\begin{aligned} V_S &= \beta_{time} x_{S,time} + \beta_{cost} x_{S,cost} + \beta_{S,age} x_{age} + \beta_S \\ V_B &= \beta_{time} x_{B,time} + \beta_{cost} x_{B,cost} + \beta_{B,age} x_{age} + \beta_B \\ V_P &= \beta_{time} x_{P,time} + \beta_{cost} x_{P,cost} + \beta_{P,age} x_{age} + \beta_P \end{aligned} \tag{3.69}$$

MNL の選択確率は $\exp(V)$ の比で決まるため，全選択肢の ASC に同じ定数 c を加えても選択確率は変化しない．すなわち $(\beta_S, \beta_B, \beta_P)$ と $(\beta_S + c, \beta_B + c, \beta_P + c)$ は同一の選択確率を与え，尤度を最大化するパラメータの組が無限に存在してしまう．これを回避するには，1 つの選択肢を基準として ASC を 0 に固定し，自由度を 1 つ下げればよい．

$$\begin{aligned} V_S &= \beta_{time} x_{S,time} + \beta_{cost} x_{S,cost} + \beta_{S,age} x_{age} \\ V_B &= \beta_{time} x_{B,time} + \beta_{cost} x_{B,cost} + \beta_{B,age} x_{age} + \beta_B \\ V_P &= \beta_{time} x_{P,time} + \beta_{cost} x_{P,cost} + \beta_{P,age} x_{age} + \beta_P \end{aligned} \tag{3.70}$$

この問題は，第 1 節でガンベル分布のスケールパラメータ μ を 1 に正規化した理由と本質的に同じである．効用の絶対値には意味がなく，差だけが選択確率を決定するという離散選択モデルの根本的性質に起因する．

3.5.3　検定

パラメータの推定値が得られたとしても，その値が統計的に信頼できるかは別問題である．検定では大きく二つの問いに答える必要がある．第一に，推定された各パラメータが 0 と有意に異なるかどうか（個々の説明変数が選択行動に影響しているか）を判定する **t 検定**，第二に，推定モデル全体が Null model（全パラメータ 0 のモデル）に比べて有意に適合度が高いかを判定する**尤度比検定**である．いずれの検定も，推定量が確率変数であることを利用してその分布を考え，帰無仮説のもとで観測結果がどの程度起こりうるかを評価する．

以下ではまず検定の前提となる統計量と推定量の概念を整理し，次に t 検定と尤度比検定の手順を述べる．

(a)　推定値と推定量

前節で最尤推定法によって求めた $\hat{\boldsymbol{\beta}}$ の具体的な数値を**最尤推定値**と呼ぶ．しかし，最尤推定値は標本の選び方によって変わるため，「尤度を最大化する操作によって得られる量」として抽象化すれば，これは標本変量の関数すなわち**統計量**であり，確率変数としての分布を持つ．このように推定のために構成される統計量を**推定量**と呼び $\hat{\beta}$ と記す．最尤推定量は以下の漸近的な性質を満たすことが知られている．

一致性 標本数を無限に増やすと，最尤推定値はパラメータの真値に収束する．

漸近正規性 標本数が十分に多いとき，最尤推定量の分布は正規分布に近づく．

漸近有効性 標本数が十分に多いとき，最尤推定量の分散は不偏推定量の分散の下限（クラメール・ラオの下限）に近付く．

(b) 最尤推定量の分散とフィッシャー情報行列

推定量の期待値がパラメータの真値に一致するとき**不偏推定量**，さらにその不偏推定量が他のどの不偏推定量よりも分散が小さいとき**有効推定量**と呼ぶ．最尤推定量は厳密には不偏ではないが，標本が十分に多ければ漸近的に有効推定量と同等のふるまいを示す（漸近有効性）．不偏推定量の分散の下限を与えるのがクラメール・ラオの不等式である．

クラメール・ラオの不等式

$\hat{\boldsymbol{\beta}}$ をパラメータ $\boldsymbol{\beta}$ の不偏推定量とし，$\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ は $\hat{\boldsymbol{\beta}}$ の分散共分散行列，LL は対数尤度関数とする．このとき以下の行列に対する不等式が成り立つ．

$$I(\boldsymbol{\beta}) := -\mathbb{E}\left[\nabla\nabla^\top LL(\boldsymbol{\beta})\right] \tag{3.71}$$

$$\mathrm{Cov}(\hat{\boldsymbol{\beta}}) \geq I(\boldsymbol{\beta})^{-1} \tag{3.72}$$

ただし，(3.72) の不等号は，左辺から右辺を引いた行列 $\mathrm{Cov}(\hat{\boldsymbol{\beta}}) - I(\boldsymbol{\beta})^{-1}$ が半正定値行列であることを意味する [a]．

a) もし $\boldsymbol{\beta}$ が 1 変数のパラメータであれば，この不等号は単に実数の不等号と同じ意味になる．

$I(\boldsymbol{\beta})$ は**フィッシャー情報行列**と呼ばれる．なお，$\nabla\nabla^\top LL(\boldsymbol{\beta})$ は書き出すと下の (3.73) のようになり，特に行列部分には**ヘッセ行列**という名前がつけられている．

$$\nabla\nabla^\top LL(\boldsymbol{\beta}) = \begin{bmatrix} \frac{\partial^2 LL}{\partial {\beta_1}^2} & \frac{\partial^2 LL}{\partial\beta_1\partial\beta_2} & \cdots & \frac{\partial^2 LL}{\partial\beta_1\partial\beta_n} \\ \frac{\partial^2 LL}{\partial\beta_2\partial\beta_1} & \frac{\partial^2 LL}{\partial {\beta_2}^2} & \cdots & \frac{\partial^2 LL}{\partial\beta_2\partial\beta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 LL}{\partial\beta_n\beta_1} & \frac{\partial^2 LL}{\partial\beta_n\partial\beta_2} & \cdots & \frac{\partial^2 LL}{\partial {\beta_n}^2} \end{bmatrix} \boldsymbol{\beta} \tag{3.73}$$

等号が成立するとき，$\hat{\boldsymbol{\beta}}$ は分散の下限を達成しているので，$\hat{\boldsymbol{\beta}}$ は有効推定量となる．最尤推定量の場合は，漸近有効性により，標本が十分多ければ等号成立，すなわち

$$\mathrm{Cov}(\hat{\boldsymbol{\beta}}) \simeq I(\boldsymbol{\beta})^{-1} \tag{3.74}$$

としてよい．このように，フィッシャー情報行列を求めることで，最尤推定量の分散を求めることができる．フィッシャー情報行列を確率変数である $\boldsymbol{\beta}$ の分布から直接計算することもできるが，フィッシャー情報行列に代えて**観測された情報行列**を用いる方が計算が簡単かつ有用である．観測された情報行列 $J(\hat{\boldsymbol{\beta}})$ は次のように定義される．

$$J(\hat{\boldsymbol{\beta}}) \coloneqq -\nabla\nabla^{\top} LL(\hat{\boldsymbol{\beta}}) \tag{3.75}$$

(3.75) を用いると，(3.74) の代わりに次の式が得られる．

$$\mathrm{Cov}(\hat{\boldsymbol{\beta}}) \simeq J(\hat{\boldsymbol{\beta}})^{-1} \tag{3.76}$$

(c)　t 検定

t 検定では，各パラメータについて「$\beta_i = 0$」（その説明変数は効用に影響しない）という帰無仮説を立て，これを棄却できるかどうかを判定する．具体的には，(3.76) から得られる各パラメータの標準誤差 s_i を用いて推定量を標準化した **t 値**

$$t_i \coloneqq \frac{\hat{\beta}_i}{s_i} \tag{3.77}$$

を計算する．漸近正規性により，$|t_i| > 1.96$ であれば帰無仮説は 5% 有意水準で棄却され，$|t_i| > 2.58$ であれば 1% 有意水準で棄却される．

注意すべき点として，帰無仮説が棄却されなかった場合，それはその説明変数が「効かない」ことを証明しているわけではなく，「効くとは言い切れない」ことを意味するにすぎない．標本数が増えれば棄却される可能性もある．

(d)　尤度比検定

t 検定が個々のパラメータの有意性を問うのに対し，尤度比検定はモデル全体としての適合度を比較する．モデル A をモデル B に説明変数・パラメータを追加した上位モデルとするとき，両者の対数尤度の差 $LL(\hat{\boldsymbol{\beta}}_A) - LL(\hat{\boldsymbol{\beta}}_B)$ が十分に大きければ，追加した説明変数がモデルの改善に寄与していると判断できる．

尤度比検定では，次のような帰無仮説を立てる．

尤度比検定での帰無仮説

帰無仮説：モデル A とモデル B の尤度比は 0 である．

尤度が大きい方がより「もっともらしい」モデルなので，説明変数が多いモデル A の方が尤度が大きくなってほしいところである．この帰無仮説が棄却されればモデル A の方が有意に尤度が大きいということができ，二つのモデルの適合度の差が有意なものであるということができる．

標本が十分に多いとき，$2(LL(\hat{\boldsymbol{\beta}}_A) - LL(\hat{\boldsymbol{\beta}}_B))$ は自由度 $N_A - N_B$（N_A, N_B はそれぞれのパラメータ数）のカイ二乗分布に従うことが知られている．この統計量がカイ二乗分布の上側 5% 点を超えれば，帰無仮説「両モデルの適合度に差はない」を 5% 有意水準で棄却できる．特に，全パラメータを 0 としたモデル（**Null model**，対数尤度 $LL(\mathbf{0})$）を基準モデルとすれば，構築したモデルがランダムな予測より有意に優れていることを確認できる．なお，尤度比検定ではモデル A がモデル B を包含する入れ子の関係にある必要がある．

(e)　McFadden の擬似決定係数と自由度調整

尤度比検定のカイ二乗統計量は自由度に依存するため，異なるモデル間の適合度を直感的に比較しにくい．回帰分析における決定係数に対応する指標として，**McFadden の擬似決定係数** ρ^2 が広

く用いられている．

$$\rho^2 \coloneqq 1 - \frac{LL(\hat{\boldsymbol{\beta}})}{LL(\mathbf{0})} \tag{3.78}$$

ρ^2 は 0（Null model と同等）から 1（完全な予測）の範囲をとり，モデルの適合度を簡潔に示す．ただし ρ^2 はパラメータ数が増えるほど値が大きくなる傾向があるため，パラメータ数 N によるペナルティを加えた**自由度調整済み決定係数** $\bar{\rho}^2$ も併せて用いられる．あまり効かないような説明変数であったとしても，ないよりはあったほうがモデルが考慮できる情報は増え，モデルの説明性は向上するのが基本である．単純な ρ^2 ではこのような過度なパラメタライゼーションを助長する可能性があるため，$\bar{\rho}^2$ の方がより厳密な適合度の指標として用いられる．

$$\bar{\rho}^2 \coloneqq 1 - \frac{LL(\hat{\boldsymbol{\beta}}) - N}{LL(\mathbf{0})} \tag{3.79}$$

経験的に $\bar{\rho}^2$ が 0.2 以上であれば十分な適合度，0.15 以上でも概ね良好とされる．Logit モデルの推定結果を報告する際には，初期尤度 $LL(\mathbf{0})$，最終尤度 $LL(\hat{\boldsymbol{\beta}})$ とともに ρ^2 や $\bar{\rho}^2$ を併記するのが一般的である．

3.5.4　観測の不確実性とモデル推定

前節までの推定と検定の議論では，暗黙の前提として**観測データが真の行動を正確に反映している**ことを仮定していた．すなわち，個人の選択結果 y_k や経路 $r = [a_1, \ldots, a_T]$ は誤りなく観測されているものとして尤度関数を構成し，パラメータを推定した．

しかし実際の交通行動データ，特に GPS や BLE（Bluetooth Low Energy）などのセンサデータに基づく観測では，観測誤差が不可避的に存在する．GPS の測位精度は屋外でも数メートル，屋内やアーケード街ではさらに大きく劣化し，BLE ビーコンでは電波減衰・パケットロス・マルチパスなどの影響を受ける．このような状況で，観測データから「真の経路」を確定的に同定することは困難であり，同定された経路が誤っていれば，その誤りに基づいて推定されたモデルパラメータもバイアスを持つことになる．

本節では，この観測の不確実性に正面から向き合い，観測モデルと行動モデルを統合的に推定する枠組みを解説する．

(a)　従来の推定における暗黙の仮定

従来の経路選択モデルの推定では，センサデータからの経路推定（マップマッチング）とモデルパラメータの推定が**分離された二段階の手続き**として行われてきた（図 3.20）．

1. **前処理（経路推定）:** GPS の位置情報列からマップマッチング等により「真の経路」$\bar{r}$ を確定的に同定する．
2. **パラメータ推定:** 同定された経路 $\bar{r}$ を真の観測として扱い，最尤推定法により行動モデルのパラメータ θ を推定する．

この手続きの問題は，Step 1 で同定された経路 $\bar{r}$ が真の経路と異なる場合，Step 2 の推定結果が系統的にバイアスを持つことである．特に歩行者ネットワークのように密なネットワークでは経路候補が多く，観測誤差に対して頑健な経路同定が困難になるため，バイアスはより深刻になる．

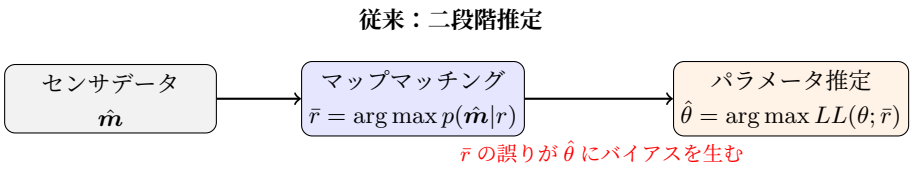


図 3.20: 従来の二段階推定の流れ．前処理で経路を確定的に同定するため，マップマッチングの誤りがパラメータ推定のバイアスに直結する．

(b)　経路観測モデルとリンク固有分散

Oyama and Hato (2018) [15] は，経路同定とパラメータ推定を分離せず，**観測モデル**と**行動モデル**を統合的に扱う枠組みを提案した．

その基本的な着想は，GPS の位置情報 $\hat{\boldsymbol{m}}$ と真の経路 r の関係を確率的に記述する**経路観測モデル**（route measurement model）を導入することである．リンク a 上の位置 x で観測された GPS 座標 $\hat{x}$ の観測確率を，リンク固有の分散 σ_a^2 を持つ正規分布でモデル化する．

$$p(\hat{x} \mid x, a; \sigma_a) = \frac{\|\hat{x} - x\|}{\sigma_a^2} \exp\left(-\frac{\|\hat{x} - x\|^2}{2\sigma_a^2}\right) \tag{3.80}$$

ここで $\|\hat{x} - x\|$ は観測位置と真の位置のユークリッド距離であり，レイリー分布の形をとる．

従来の観測モデルではネットワーク全体で一定の分散 σ を仮定していたが，実際にはアーケード街や高層ビル近辺では分散が大きく，見通しの良い道路では小さいといった**空間的な異質性**がある．リンク固有の分散 σ_a を導入することで，この異質性を明示的にモデルに組み込むことが可能になる．

観測モデルと行動モデルを結合した**リンク尤度**は，ベイズの定理により以下のように書ける．

$$p(a_t \mid \hat{\boldsymbol{m}}^t, a_{t-1}) = \frac{p(\hat{\boldsymbol{m}}^t \mid a_t; \sigma_{a_t})\, p(a_t \mid a_{t-1}; \theta)}{\sum_{a \in \mathcal{A}(a_{t-1})} p(\hat{\boldsymbol{m}}^t \mid a; \sigma_a)\, p(a \mid a_{t-1}; \theta)} \tag{3.81}$$

ここで $p(\hat{\boldsymbol{m}}^t \mid a_t; \sigma_{a_t})$ は観測モデル，$p(a_t \mid a_{t-1}; \theta)$ は行動モデル（RL モデルの遷移確率）である．この式は，各時点で「観測データがどの程度そのリンクと整合的か」（観測尤度）と「行動モデルに基づいてそのリンクがどの程度選ばれやすいか」（事前確率）の積によって，リンクの事後確率を計算する構造になっている．

(c)　構造推定：観測モデルと行動モデルの不動点

上述のリンク尤度は観測パラメータ σ と行動パラメータ θ の双方に依存するため，両者を同時に推定する必要がある．Oyama and Hato (2018) [15] はこの問題を，観測モデルと行動モデルの**不動点**を探索する**構造推定**（structural estimation）として定式化した．

1. **初期化:** 行動パラメータの初期値 $\theta^{(0)}$（例：無情報事前分布 $\theta^{(0)} = \mathbf{0}$）を設定する．
2. **観測モデルの推定:** 行動パラメータ $\theta^{(h)}$ を固定し，観測モデルの分散 σ を推定する．同時に推定された経路 $r^{(h)}$ を得る．
3. **行動モデルの推定:** 推定された経路 $r^{(h)}$ を用いて行動パラメータ $\theta^{(h+1)}$ を最尤推定する．
4. **収束判定:** $|\theta^{(h+1)} - \theta^{(h)}| < \xi$ であれば終了．そうでなければ Step 2 に戻る．

この反復過程は，情報幾何学における多様体学習の EM（em, expectation-maximization）アルゴリズム [11] と類似の構造を持つ．Step 2 は「現在の行動モデル（事前分布）のもとで観測データから経路を推定する」操作（E-step），Step 3 は「推定された経路のもとで行動パラメータを更新する」操作（M-step）に対応する．

不動点では $\theta^{(h)} = \theta^{(h+1)}$ が成立し，観測モデルが推定に用いる事前情報（行動モデル）と，推定結果から得られる行動パラメータが**整合的**（self-consistent）になる．

(d)　量子化と観測多様体

前項の構造推定は，各時点で「最も尤もらしい 1 本のリンク」を逐次的に同定する枠組みであった．しかし，屋内の BLE ビーコン観測のように観測点密度が低くかつ観測分散が大きい状況では上記の手法に基づいて算出される最尤のリンクですら信頼性が低く，逐次的な同定の誤りが後続の推定に波及するという問題が残る．

Matsunaga and Hato (2026) [12] は，この問題に対してリンクの確定的な同定を一切回避する**量子化**（quantization）のアプローチを提案した．「量子化」とは，物理学における量子力学のアナロジーであり，従来の確定的な状態同定（古典的な 0/1 の判定）を，複数の状態に対する確率分布へと緩和する操作を指す．

具体的には，従来の最尤推定における確定的な遷移の指示関数（クロネッカーのデルタ）

$$\delta_{S_T|S_{T-1}} = \begin{cases} 1, & \text{if } S_T \text{ が真の状態} \\ 0, & \text{otherwise} \end{cases} \tag{3.82}$$

を，観測データに条件づけられた**遷移尤度**（transition likelihood）$Q_{S_T,S_{T-1}}$ に置き換える．

$$Q_{S_T,S_{T-1}}(S^{\mathrm{obs}}) = p(S_T \mid S_{T-1}, S^{\mathrm{obs}}) \propto p(S^{\mathrm{obs}} \mid S_T;\sigma)\, p(S_T \mid S_{T-1};\theta) \tag{3.83}$$

ここで S^{obs} は BLE ビーコンの観測レコード集合，$p(S^{\mathrm{obs}} \mid S_T;\sigma)$ は観測モデル，$p(S_T \mid S_{T-1};\theta)$ は行動モデルの遷移確率である．

この量子化のもとで，対数尤度は以下の**一般化対数尤度**（generalized log-likelihood）に拡張される．

$$\tilde{LL}(\sigma,\theta) = -\sum_{i} \sum_{T=1}^{T^{\max}} Q_{S_T,S_{T-1}}\big(S^{\mathrm{obs}};\sigma,\theta\big) \ln p(S_T \mid S_{T-1};\theta) \tag{3.84}$$

通常の対数尤度では $\delta_{S_T|S_{T-1}} \in \{0,1\}$ であった重みが，$Q_{S_T,S_{T-1}} \in [0,1]$ という連続的な確率分布に置き換わっている．この Q の分布は観測データとモデルパラメータの双方に依存し，パラメータの更新に伴って変化する．この意味で Q は固定された点ではなく，パラメータ空間上の**多様体**（manifold）を形成する．Matsunaga and Hato (2026) [12] はこれを**観測多様体**（observation manifold）と呼び，EM アルゴリズムと同様に，観測多様体とモデル多様体の間の繰り返し射影によってパラメータを推定する手法を提案した．これは本質的には Oyama and Hato (2018) [15] の構造推定の一般化であり，経路同定を完全に回避したことで一つの目的関数 (3.84) を直接最大化することが可能になった．

(e)　潜在状態モデルとしての統一的理解

以上で紹介した枠組みは，いずれも「真の経路は直接観測できない**潜在変数**（latent variable）であり，観測されるのは観測誤差を含んだセンサデータである」という視点に立つ．Oyama and Hato (2018) [15] では経路列（クロネッカーデルタ）が潜在変数であったのに対し，Matsunaga and Hato (2026) [12] では遷移尤度 Q が潜在変数となっている．

重要な点は，この枠組みにおいて行動モデルと観測モデルが**相互に依存**していることである．行動モデルのパラメータが変われば，各時点でどのリンクが「もっともらしい」かの判断が変わり，結果として推定に用いられるデータ自体が変化する．逆に，観測モデルの精度が向上すれば，より正確な経路情報に基づいて行動パラメータを推定できる．構造推定や量子化の手法は，この相互依存性を不動点の探索として解決するものであり，「データは真である」という暗黙の仮定から解放された，より頑健な推定を可能にする．

クイズ 1：誤差項の解釈

効用関数 $U_a = V_a + \varepsilon_a$ の誤差項 ε_a が表すものとして最も適切なものはどれか．

(ア) 個人の非合理的な意思決定による揺らぎ

(イ) 観測データの観測誤差

(ウ) 分析者が観測できない，選択に影響を与える要因

(エ) 個人が選択肢を正確に比較できないことによる揺らぎ

クイズ 2：ログサム変数の直感的意味

NL モデルにおいて，公共交通ネスト（赤バスのみ）に「青バス」が新たに加わったとき，ログサム変数 $V'_{\text{公共交通}}$ はどう変化するか．

(ア) 変化しない（確定効用が同じ選択肢の追加は影響しない）

(イ) 増加する

(ウ) 減少する

(エ) 青バスの確定効用が赤バスより大きい場合にのみ増加する

クイズ 3：Mixed Logit の推定結果の解釈

Mixed Logit モデルで交通手段選択を推定した結果，旅行時間パラメータの標準偏差 σ が期待値 $\bar{\beta}$ と同程度に大きかった．この結果の解釈として最も適切なのはどれか．

(ア) 旅行時間は選択行動にほとんど影響していない

(イ) 旅行時間の観測データに大きな観測誤差がある

(ウ) 旅行時間を重視する人としない人が母集団内に混在している

(エ) モデルの当てはまりが悪く，推定が収束していない

クイズ 1 正解・解説

正解：(ウ)
ランダム効用最大化理論（RUT）は**完全合理性**を前提としている．すなわち，個人は自身の効用を完全に把握しており，常にそれを最大化するように選択を行う．この前提は誤差項 ε_a を導入しても崩れない．（ア）や（エ）のように個人の意思決定に非合理性や不完全性を認める立場は，限定合理性（bounded rationality）の考え方であり，RUT の枠組みとは異なる．
誤差項 ε_a が表すのは，選択に影響を与えるが**分析者が観測できない要因**である．例えば「今日は天気がいいから自転車にしよう」という判断は本人にとっては完全に合理的だが，分析者のモデルに天気が説明変数として含まれていなければ，その影響は ε に吸収され「ランダムに自転車を選んだ」ように見える．すなわち “Random Utility” の “Random” は個人の行動がランダムであることを意味するのではなく，分析者から見て効用がランダムに見えることを意味している．

クイズ 2 正解・解説

正解：(イ)
ログサム変数は $V_d' = \frac{1}{\mu}\ln\sum_i \exp\bigl(\mu(V_i + V_{di})\bigr)$ で定義される．exp の和に正の項が追加されるため，新たな選択肢の確定効用の大小によらず V_d' は必ず増加する．これが NL モデルにおける赤バス青バス問題の解決の仕組みである．青バスの追加により公共交通ネストのログサム変数（すなわちネスト全体の「魅力度」）が上昇し，上位ネストでの公共交通の選択確率が増加する．一方でこの増加は上位ネストの MNL 構造を通じて調整されるため，MNL モデルのように車の選択確率が 1/2 から 1/3 まで極端に低下することはない．

クイズ 3 正解・解説

正解：(ウ)
Mixed Logit モデルにおけるランダム係数の標準偏差 σ は，そのパラメータが母集団内でどれだけばらついているかを表す．σ が $|\bar{\beta}|$ と同程度に大きいということは，旅行時間を非常に重視する人からほとんど気にしない人まで幅広く混在していることを意味する．さらに，

正規分布を仮定している場合，$\sigma \approx |\bar{\beta}|$ であればパラメータの符号が反転する個人（旅行時間が長いほうを選好する人）が一定割合存在することも示唆される．これは MNL モデルでは表現できない情報であり，Mixed Logit モデルを用いる意義そのものである．

参考文献

[1] Arentze, T. A., Ettema, D., Timmermans, H. J.: Estimating a model of dynamic activity generation based on one-day observations: Method and results, Transportation Research Part B: Methodological, Vol. 45, No. 2, pp. 447–460, 2011.

[2] Arentze, T. A., Timmermans, H. J.: A need-based model of multi-day, multi-person activity generation, Transportation Research Part B: Methodological, Vol. 43, No. 2, pp. 251–265, Modeling Household Activity Travel Behavior, 2009.

[3] Bhat, C. R.: The multiple discrete-continuous extreme value (MDCEV) model: Role of utility function parameters, identification considerations, and model extensions, Transportation Research Part B: Methodological, Vol. 42, No. 3, pp. 274–303, A Tribute to the Career of Frank Koppelman, 2008.

[4] Bowman, J., Ben-Akiva, M.: Activity-based disaggregate travel demand model system with activity schedules, Transportation Research Part A: Policy and Practice, Vol. 35, No. 1, pp. 1–28, 2001.

[5] Dubin, J. A., McFadden, D. L.: An econometric analysis of residential electric appliance holdings and consumption, Econometrica, Vol. 52, pp. 345–362, 1984.

[6] Fosgerau, M., Frejinger, E., Karlstrom, A.: A link based network route choice model with unrestricted choice set, Transportation Research Part B, Vol. 56, pp. 70–80, 2013.

[7] Fosgerau, M., Paulsen, M., Rasmussen, T. K.: A perturbed utility route choice model, Transportation Research Part C: Emerging Technologies, Vol. 136, p. 103514, 2022.

[8] Habib, K. M., Miller, E. J.: Modelling daily activity program generation considering within-day and day-to-day dynamics in activity-travel behaviour, Transportation, Vol. 35, No. 4, pp. 467–484, 2008.

[9] Habib, K. M. N.: A random utility maximization (RUM) based dynamic activity scheduling model: Application in weekend activity scheduling, Transportation, Vol. 38, No. 1, pp. 123–151, 2011.

[10] Heckman, J. J.: Sample selection bias as a specification error, Econometrica, Vol. 47, pp. 153–161, 1979.

[11] ichi Amari, S.: Information geometry of the EM and em algorithms for neural networks, Neural Networks, Vol. 8, No. 9, pp. 1379–1408, 1995.

[12] Matsunaga, T., Hato, E.: Mean field dynamics for pedestrian interaction in quantized observation-based 3D route choice modeling, Transportation Research Part C: Emerging Technologies, Vol. 186, p. 105608, 2026.

[13] McFadden, D.: Modelling the choice of residential location, 1977.

[14] Oyama, Y., Hato, E.: A discounted recursive logit model for dynamic gridlock network analysis, Transportation Research Part C, Vol. 85, pp. 509–527, 12 2017.

[15] Oyama, Y., Hato, E.: Link-based measurement model to estimate route choice parameters in urban pedestrian networks, Transportation Research Part C, Vol. 93, pp. 62–78, 8 2018.

[16] Oyama, Y., Hato, E.: Prism-based path set restriction for solving Markovian traffic assignment problem, Transportation Research Part B: Methodological, Vol. 122, pp. 528–546, 2019.

[17] Tobin, J.: Estimation of relationships for limited dependent variables, Econometrica, Vol. 26, pp. 24–36, 1958.

[18] Wales, T. J., Woodland, A. D.: Estimation of consumer demand systems with binding non-negativity constraints, Journal of Econometrics, Vol. 21, pp. 263–285, 1983.

[19] Zimmermann, M., Mai, T., Frejinger, E.: Bike route choice modeling using GPS data without choice sets of paths, Transportation Research Part C: Emerging Technologies, Vol. 75, pp. 183–196, 2017.

[20] 大山 雄己, 福山 祥代, 羽藤 英二: 活動欲求を考慮した離散-連続モデルによる小滞在発生メカニズムの分析, 都市計画論文集, Vol. 49, No. 3, 日本都市計画学会, pp. 375–380, 2014.

[21] 大山 雄己, 羽藤 英二: 時空間制約と経路相関を考慮した歩行者の活動配分問題, 都市計画論文集, Vol. 51, No. 3, 日本都市計画学会, pp. 680–687, 2016.

[22] 福田 大輔, 力石 真: 離散-連続モデルの研究動向に関するレビュー, 土木学会論文集 D3（土木計画学）, Vol. 69, No. 5, pp. 497–510, 2013.

第 4 章
最適化

今村啓太

4.1 最適化とは

4.1.1 最適化の基本的な考え方

私たちは日常生活の中で，意識するかどうかにかかわらず，常に何らかの選択を行っている．例えば，大学へ行くときにどの経路を通れば最も早く到着できるかを考えることがあるし，限られた時間の中でどの作業から先に進めるべきかを判断することもある．また，企業であれば，複数の工場の中でどこにどれだけ生産を割り当てれば費用を最も小さくできるかを考えるし，物流事業者であれば，複数の配送先をどの順番で回れば移動距離を最も短くできるかを検討する．このように，いくつかの選択肢の中から，ある基準に照らして最も望ましいものを選ぶという考え方は，社会のさまざまな場面に現れている．

最適化とは，このような「最も良い選択」を，感覚や経験だけではなく，明確な基準と制約のもとで数理的に求めようとする考え方である．何を良いとみなすかは，問題の目的によって異なる．例えば，移動であれば時間が短いことが望ましいかもしれないし，輸送であれば費用が小さいことが重視されるかもしれない．場合によっては，安全性や公平性，環境負荷の低さなどが重視されることもある．したがって，最適化とは答えを計算する作業ではなく，まず何を評価したいのかを明確にし，そのうえで何が可能で何が不可能なのかを整理する作業でもある．

また，最適化は，図 4.1 に示すように現実社会の課題を解決するための体系的なアプローチとして位置づけられる [23]．すなわち，まず現実社会に存在する問題を整理し（課題の把握），それを数理モデルとして定式化することで扱いやすい形に落とし込む．次に，そのモデルに対して最適化計算を行い，最適解を導出する．さらに，得られた解について分析を行い，現実の状況に適用可能かを確認することで，具体的な解決施策へとつなげていく．このように最適化は，「課題の理解→定式化→解の導出→検証・適用」という一連のプロセスを通じて，現実の意思決定を支援する重要な枠組みである．

4.1.2 最適化問題の数学的定式化

最適化問題は一般に次のように書かれる．

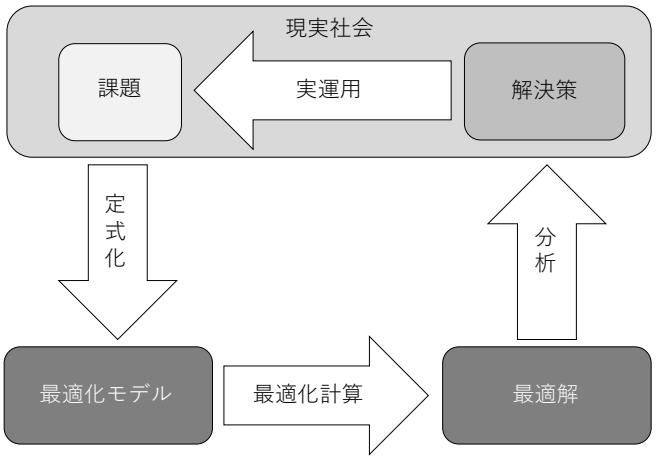


図 4.1: 最適化による課題解決のフレーム

$$\text{minimize} \quad f(x) \tag{4.1}$$

$$\text{subject to} \quad x \in X \tag{4.2}$$

ここで，x は決定変数と呼ばれ，私たちが選ぶ対象を表している．$f(x)$ は目的関数と呼ばれ，その選択がどれだけ良いか，あるいはどれだけ悪いかを数値として表したものである．どのような選択が望ましいのかを数量的に評価するためのものが目的関数である．

また，$x \in X$ は制約条件を表している．ここで，X は実行可能集合と呼ばれ，満たすべき条件をすべて満たすような x の集合である．例えば，資源の上限や容量制約，順序や論理関係など，現実の問題における制限はこの集合 X によって表現される．

このように，決定変数，目的関数，制約条件を組み合わせることで，最適化問題は一つの数理モデルとして定式化される．目的関数や制約条件に何を含めるかによって，得られる解の意味は大きく変わる．例えば，移動時間だけを目的関数に含めれば，費用や環境負荷は考慮されないことになる．逆に，制約を増やしすぎれば，現実には解けないほど複雑なモデルになってしまうこともある．したがって，最適化においてはどのように問題を表現するかが重要である．本書で扱うさまざまな問題も，この観点から理解することが重要である．

4.1.3　最大化問題と最小化問題

最適化問題には，大きく分けて，何かをできるだけ大きくしたい問題と，何かをできるだけ小さくしたい問題の二種類がある．前者は最大化問題，後者は最小化問題と呼ばれる．例えば，利益や満足度，利便性などを大きくしたい場合には最大化問題となる．一方で，費用，時間，損失，移動距離などを小さくしたい場合には最小化問題となる．ある関数 $f(x)$ を最大化する問題は，その符

号を反転させた $-f(x)$ を最小化する問題として書き直すことができる．すなわち，

$$\max f(x) \tag{4.3}$$

という問題は，

$$\min\{-f(x)\} \tag{4.4}$$

と等価である．したがって，理論的には最大化と最小化を厳密に区別する必要はなく，どちらか一方に統一して議論することができる．本書では，記述を簡潔にするため，主として最小化問題の形で説明を行う．

ただし，ここで重要なのは，最大化か最小化かという表面的な違いではなく，何を評価対象としているかである．例えば，ある物流問題において「利益最大化」と「費用最小化」は似た意味を持つかもしれないが，必ずしも同じではない．利益は収入から費用を差し引いたものであり，費用だけを最小化しても利益が最大になるとは限らないからである．

また，現実の問題では，評価したいものが一つだけとは限らない．例えば，都市交通では，移動時間を短くしたい一方で，費用も抑えたいし，環境負荷も小さくしたいというように，複数の目的が同時に存在することが多い．このような問題では，複数の目的をどう統合するかが新たな課題となる．例えば，時間と費用を重み付き和として

$$f(x) = \alpha T(x) + \beta C(x) \tag{4.5}$$

のように表し，一つの目的関数として扱うことがある．ここで，$T(x)$ は時間，$C(x)$ は費用，α, β はそれぞれの重要度を表す重みである．このような考え方は，多目的最適化となる．

このように，最大化問題と最小化問題は数学的には容易に変換できるが，目的関数に何を含めるかは決して自明ではない．したがって，「最大化か最小化か」という形式だけに注目するのではなく，その目的関数が何を意味しているのか，どのような価値判断を反映しているのかを意識して読むことが大切である．

4.1.4 連続最適化と離散最適化

最適化問題は，決定変数がどのような値をとるかによって，大きく連続最適化と離散最適化に分けられる．この区別は，問題の性質，理論，解法のすべてに大きな影響を与える．

連続最適化とは，決定変数が連続的な値，すなわち実数値をとる最適化問題である．例えば，生産量，速度，時間，投資割合のように，理論上は小数点以下を含む任意の値をとることができる変数を扱う問題がこれに該当する．変数 x が実数全体，あるいはその一部の区間に属するとき，これを記号で

$$x \in \mathbb{R}^n \tag{4.6}$$

のように表す．このような問題では，目的関数や制約関数がなめらかであれば，微分や勾配といった解析的な手法を用いて解を探すことができる．

例えば，単純な問題として，関数

$$f(x) = x^2 \tag{4.7}$$

を最小にする x を考える．この関数は放物線であり，$x = 0$ のとき最小値 0 をとる．微分を用いれば，

$$\frac{d}{dx} f(x) = 2x \tag{4.8}$$

であるから，これが 0 となる点，すなわち $x = 0$ が候補となる．このように，連続最適化では，関数の局所的な変化率に関する情報を使って最適解を見つけることができる．後に学ぶ最適性条件や KKT 条件は，まさにこの考え方を一般化したものである．

これに対して，離散最適化では，決定変数は飛び飛びの値しかとらない．例えば，ある道路を使うか使わないかという選択は，通常，0 か 1 で表される．また，複数の配送先をどの順番で回るかという問題では，順序そのものが変数となる．このような問題では，変数は整数や 0–1 値として表される．例えば，0–1 変数であれば，

$$x_{ij} \in \{0, 1\} \tag{4.9}$$

のように表される．

離散最適化が難しい理由の一つは，連続最適化のように微分を使うことができない点にある．変数が 0 か 1 しかとらないのであれば，「少しだけ増やす」という概念自体が意味を持たないからである．さらに本質的な難しさは，可能な解の数が急速に増加することにある．例えば，n 個の地点を一度ずつ訪問する順序を考えると，その候補数は $n!$ 通りある．$n = 5$ であれば $5! = 120$ 通りで済むが，$n = 10$ では $10! = 3,628,800$ 通りとなり，$n = 20$ ではその数は天文学的な大きさになる．これがいわゆる組合せ爆発であり，巡回セールスマン問題や配送計画問題が難しいとされる主な理由である．

連続最適化と離散最適化の違いは，解法にもはっきりと現れる．連続最適化では，単体法，内点法など，関数の構造を利用した方法が有効である．一方で，離散最適化では，分枝限定法，列生成法など，離散的な選択肢をうまく探索する方法が中心となる．

ただし，現実の問題は必ずしもこの二つのいずれかにきれいに分かれるわけではない．むしろ，両者が混在する場合のほうが多い．例えば，車両配送の問題では，ある顧客を訪問するかどうかは 0–1 変数で表される一方，到着時刻や待ち時間は連続変数で表されることが多い．このような問題は混合整数計画問題と呼ばれ，連続最適化と離散最適化の両方の性質を併せ持つ．

したがって，最適化を学ぶ際には，まず自分が扱っている変数が連続なのか離散なのかを見極めることが重要である．この違いを正しく理解することで，なぜ問題によって必要な理論やアルゴリズムが異なるのかが見えてくる．

4.1.5　実行可能領域と最適解

最適化問題を理解するうえで，実行可能領域と最適解という二つの概念は中心的な役割を果たす．これらをきちんと理解しないまま解法だけを学ぼうとすると，なぜその手法が成り立つのか，なぜある問題は解きやすく別の問題は解きにくいのかが見えにくくなる．

まず，実行可能領域とは，与えられた制約条件をすべて満たす解の集合のことである．例えば，二変数 x_1, x_2 に対して

$$x_1 + x_2 \leq q4, \qquad x_1 \geq q0, \qquad x_2 \geq q0 \tag{4.10}$$

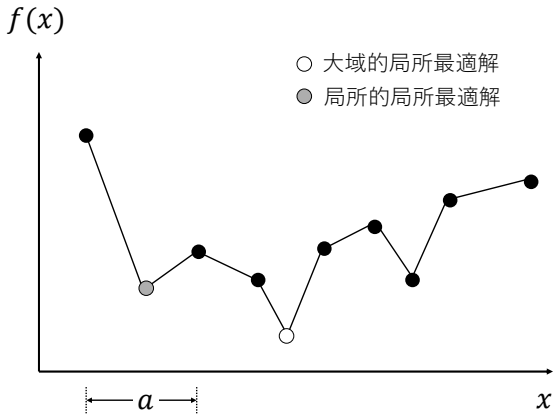


図 4.2: 局所的最適解と大域的最適解

という制約があるとする．このとき，これらの条件を同時に満たすすべての点 (x_1, x_2) の集まりが実行可能領域である．図で描けば，第一象限内のある三角形領域になる．

この実行可能領域の中で，目的関数を最大（または最小）にする点が最適解である．数式で書けば，x^* が最適解であるとは，

$$f(x^*) \leq f(x) \qquad \text{for all} x \in X \tag{4.11}$$

が成り立つことを意味する．ここで X は実行可能領域である．つまり，最適解とは，実行可能なすべての候補の中で最も優れた解である．

ただし，ここで注意しなければならないのは，「最適」という言葉には局所的な意味と大域的な意味があるということである．ある点 a の近くを少し動かしてもそれより良い解が見つからない場合，その点を局所的最適解と呼ぶ（図 4.2）．一方，実行可能領域全体を見渡して，それ以上良い解が存在しない場合，その点を大域的最適解と呼ぶ．連続最適化では，局所最適解はしばしば勾配が 0 になる点として現れるが，それが必ずしも大域最適解とは限らない．

例えば，非凸な関数では，複数の谷が存在することがある．その場合，ある谷の底に到達したとしても，もっと深い別の谷が離れた場所にあるかもしれない．局所的最適解と大域的最適解の違いは，まさにこのような状況に対応している．

ここで重要になるのが凸性である．実行可能領域 X が凸集合であるとは，X の任意の二点 x, y をとったとき，それらを結ぶ線分上のすべての点も X に含まれることを意味する．数式で書けば，任意の $\lambda \in [0, 1]$ に対して

$$\lambda x + (1 - \lambda)y \in X \tag{4.12}$$

が成り立つことである．さらに，目的関数 $f(x)$ も凸関数であれば，その問題は凸最適化問題とな

る．このとき，局所最適解は必ず大域最適解になる．これは最適化理論において最も重要な性質の一つであり，なぜ凸最適化が特別に重視されるのかを説明している．

また，最適解は常に一つとは限らない．例えば，線形計画問題では，目的関数の等高線が実行可能領域の辺と平行になる場合，その辺上のすべての点が最適解になることがある．このように，複数の最適解が存在する場合には，目的関数の値だけでは解が一意に定まらない．その場合，別の基準，例えば公平性や実務上の扱いやすさなどを追加して，最終的な解を選ぶことがある．

このように，実行可能領域と最適解の概念は，最適化問題の土台をなしている．制約がどのような形で実行可能領域を作り，その中で目的関数がどのように最適解を決めるのかを理解することは，後に学ぶ双対性やアルゴリズムの理解にも直結する．

クイズ 1：バス運行計画はどのタイプの最適化？

ある都市において，バス路線の運行計画を最適化する問題を考える．具体的には，各路線の運行頻度や経路を決定し，利用者の総移動時間の短縮と運行コストの削減，さらに環境負荷の低減を同時に考慮する．また，バスは各停留所を必ずしもすべて通過する必要はなく，運行経路の選択や停車の有無も意思決定の対象とする．
この問題に関する次の記述のうち，本文の内容として最も適切なものはどれか．

(ア) この問題は移動時間やコストといった連続量を扱うため連続最適化として定式化される．また，目的関数に複数の指標を含める場合でも，それらは単純に加算することで統一的に扱えるため，評価対象の選び方が解の意味に与える影響は限定的である．さらに，最大化問題として定式化する場合には最小化問題とは異なる理論体系を用いる必要がある．

(イ) この問題では，運行頻度や時間は連続的に扱われる一方で，経路の選択や停車の有無は離散的な意思決定となるため，連続最適化と離散最適化の性質が混在する．また，移動時間・コスト・環境負荷といった複数の評価指標は重み付けなどにより統合されることがあるが，その設定によって解の意味は変わり得る．さらに，最大化問題と最小化問題は数学的には変換可能であるが，何を目的関数として表現するかは問題の本質に関わる．

(ウ) この問題では，停車の有無や経路選択といった離散的な意思決定が含まれるため，解の候補数が非常に多くなる可能性がある．そのため，すべての候補を網羅的に探索することは現実的ではなく，何らかの探索戦略を用いることが重要となる．また，実行可能領域が与えられていれば，局所最適解であっても大域的最適解とみなしてよい．

→ 正解・解説は節末を参照

4.1.6 線形最適化と非線形最適化

最適化問題を分類するもう一つの重要な観点は，目的関数や制約条件の形が線形であるか，それとも非線形であるかという点である．この違いは，問題の扱いやすさや理論の美しさに大きく関係している．

線形最適化とは，目的関数と制約条件のすべてが変数に関して一次式で表される最適化問題である．典型的には，次のような形で書かれる．

$$\text{minimize} \quad c^T x \tag{4.13}$$

$$\text{subject to} \quad Ax \leq b \tag{4.14}$$

ここで，x は変数ベクトル，c は目的関数の係数ベクトル，A は制約係数行列，b は右辺ベクトルである．$c^T x$ は各変数に重みをかけて足し合わせた形であり，$Ax \leq b$ も各制約が一次不等式で表されている．

線形であることにより，実行可能領域は凸多面体という扱いやすい形になる．そして，このとき最適解はその凸多面体の頂点のいずれかに存在することが知られている．これは直感的には，平面で表される目的関数を一定方向に移動させていくと，最初に接する点が頂点になるという幾何学的な性質に対応している．この構造のおかげで，単体法や内点法のような強力なアルゴリズムが成立する．

一方で，目的関数や制約の中に二乗項，積，指数関数，対数関数，三角関数などが含まれる場合，その問題は非線形最適化となる．例えば，

$$f(x) = x^2 \tag{4.15}$$

や

$$f(x) = x_1 x_2 \tag{4.16}$$

のような関数は非線形である．非線形最適化では，線形最適化のような単純な幾何構造は一般には失われる．そのため，最適解を見つけることが急に難しくなる．

非線形最適化で重要なのは，局所最適解と大域最適解が一致しない場合があることである．例えば，

$$f(x) = x^4 - x^2 \tag{4.17}$$

という関数を考えると，この関数は複数の谷を持つ形をしており，局所的に低い点がいくつか存在する．このような関数では，ある解法が一つの谷の底にたどり着いたとしても，それが全体の中で最も低い点であるとは限らない．これが，非線形最適化，特に非凸最適化の難しさである．

ただし，非線形であっても凸性が保たれていれば，問題はかなり扱いやすくなる．目的関数が凸関数であり，実行可能領域も凸集合である場合，その問題は凸最適化問題と呼ばれる．このとき，任意の局所最適解は大域最適解でもある．これは理論的にも実務的にも重要な性質であり，現代最適化理論の中心をなしている．

都市交通や物流の分野では，現実の現象を忠実に表そうとすると，しばしば非線形性が現れる．例えば，交通量と旅行時間の関係，需要と料金の関係，混雑と効用の関係などは，一般に線形では表しにくい．そのため，実問題では非線形最適化が現れることが多い．しかし一方で，理論や計算のしやすさの観点から，まず線形モデルとして近似し，そのあと必要に応じて非線形性を導入するという進め方もよく行われる．

したがって，線形最適化と非線形最適化の違いを理解することは，その違いが，問題の構造，解の性質，利用できるアルゴリズムの種類にどのような影響を与えるのかを理解することが重要なのである．

4.1.7　都市交通における最適化の例

最適化という考え方は，都市交通や都市物流の分野において重要である．なぜなら，この分野では，限られた道路空間，時間，資源の中で，多数の利用者や車両が相互に影響し合いながら移動しているからである．そこでは常に，何らかの意味で「より良い運用」や「より効率的な配置」が求められている．

最も基本的な例の一つは，最短経路問題である．これは，道路ネットワーク上で出発地から目的地までの距離や時間を最も小さくする経路を求める問題である．道路ネットワークをグラフとして表し，交差点をノード，道路をリンクと考えると，各リンク (i, j) に対して移動コスト c_{ij} を定め，ある経路の総コストを最小化する問題として定式化できる．

次に，都市物流において中心的な問題となるのが配送計画問題（Vehicle RoutingProble m；VRP）である．この問題では，一つまたは複数の車両が，複数の顧客を訪問して荷物を配送する状況を考える．ここで決めなければならないのは，どの車両がどの顧客を担当するか，そして各車両がどの順序で顧客を訪問するかである．総移動距離や総配送時間を最小化したい一方で，各顧客はちょうど一度訪問されなければならず，各車両の容量制約も守らなければならない．このように，VRP は複数の制約の下で訪問順序を決める組合せ最適化問題であり，都市物流の効率化を考えるうえで不可欠なモデルである．

さらに，どこに物流拠点や駐車場を配置すべきかという施設配置問題も重要である．これは，都市空間の中でどの位置に拠点を置けば，全体として利用者へのアクセス性や配送効率が高くなるかを考える問題であり，長期的な都市計画とも密接に関わっている．

クイズ 1 正解・解説

正解：（イ）

本問題では，運行頻度や時間は連続変数である一方，経路選択や停車の有無は離散変数であるため，連続最適化と離散最適化が混在する混合的な問題となる．

また，移動時間・コスト・環境負荷など複数の評価指標は重み付き和などで統合されることがあるが，その重み設定により解の意味は変化する．したがって，目的関数の設計は本質的に重要である．

さらに，最大化問題は符号反転により最小化問題として表現できるため数学的には等価で

あるが，何を目的として定式化するかは意思決定そのものに関わる重要な問題である．
(ア) は，離散的意思決定の存在を無視している点，および目的関数の重要性を過小評価している点で不適切である．
(ウ) は前半の「組合せ爆発」や「探索戦略の必要性」は正しいものの，局所最適解と大域最適解を同一視している点が誤りである．この性質は凸最適化においてのみ成立する．

4.2 線形計画法

4.2.1 線形計画法とは

線形計画法は，目的関数と制約条件のすべてが線形で表される最適化問題であり，数理最適化の理論と応用の両面において中心的な役割を果たしてきた．実際，線形計画法は最適化理論の出発点であると同時に，現在でも生産計画，輸送計画，資源配分など，多くの分野で広く用いられている．

線形計画問題は，一般に次のように書かれる．

$$\text{minimize} \quad c^T x \tag{4.18}$$

$$\text{subject to} \quad Ax \leq b \tag{4.19}$$

$$x \geq 0 \tag{4.20}$$

ここで，$x \in \mathbb{R}^n$ は決定変数ベクトル，$c \in \mathbb{R}^n$ は目的関数の係数ベクトル，$A \in \mathbb{R}^{m \times n}$ は制約係数行列，$b \in \mathbb{R}^m$ は右辺ベクトルである．

多くの現実問題が少なくとも近似的には線形モデルとして表現できる．例えば，生産計画では，製品ごとの生産量を変数とし，総利益や総費用を線形に表すことができる．物流計画では，輸送量に比例してコストが増えると仮定すれば，やはり線形な目的関数が得られる．資源配分の問題でも，各活動が一定量の資源を消費し，その活動量に比例して利益や費用が変化するならば，線形計画として自然に定式化できる．このように，完全に厳密ではなくても，「まず線形近似から始める」という考え方は多くの分野で有効である．

また，線形計画問題は幾何学的にも代数的にも美しい構造を持っており，そのため強力な理論とアルゴリズムが成立する．一般の非線形最適化では，局所最適と大域最適が一致しないことや，複数の鞍点が現れることが問題を難しくする．しかし，線形計画では実行可能領域が凸多面体となり，目的関数も線形であるため，問題構造が整理される．その結果，後に見るように，双対問題が自然に定義できる，KKT 条件が必要十分条件になる，といった強い性質が得られる．

線形計画法を理解するためには，まず線形性が何を意味しているのかを押さえる必要がある．線形関数とは，変数の一次結合で表される関数であり，

$$f(x) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \tag{4.21}$$

のような形をしている．ここでは，変数同士の積や二乗項は現れない．したがって，各変数の影響は加法的であり，ある変数を増やしたときの目的関数への寄与は，他の変数の値に依存しない．制

約も同様に，

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i \tag{4.22}$$

のような一次不等式で表される．この単純さが，後に見る凸性，双対性，単体法などの理論を可能にしている．

4.2.2　線形計画問題の幾何学的理解

線形計画法の本質を理解するうえで，幾何学的な見方は重要である．2 変数あるいは 3 変数の場合には，制約条件を図で表すことができ，そのとき実行可能領域は半平面や半空間の共通部分として得られる．線形不等式で表される集合は凸集合であり，したがって，線形計画問題の実行可能領域は凸多面体となる．この「凸多面体」という性質こそが，線形計画法を特別に扱いやすいものにしている．

例えば，2 変数の問題

$$\text{minimize} \quad -x_1 - x_2 \tag{4.23}$$

$$\text{subject to} \quad x_1 + x_2 \leq 4 \tag{4.24}$$

$$x_1 \leq 3 \tag{4.25}$$

$$x_2 \leq 2 \tag{4.26}$$

$$x_1, x_2 \geq 0 \tag{4.27}$$

を考えると，実行可能領域は平面上の多角形として表される．各制約は一つの半平面を表し，その共通部分として，有限個の辺と頂点からなる領域が得られる．一方，目的関数

$$f(x_1, x_2) = -x_1 - x_2 \tag{4.28}$$

の等高線は，

$$x_1 + x_2 = \text{const.} \tag{4.29}$$

という直線の族となる．最適化とは，この等高線を一定方向に平行移動させていき，実行可能領域に最後まで接している点を求めることに対応する．

この説明から分かるように，線形計画法の目的関数は曲面ではなく平面であり，実行可能領域も凸な多面体である．したがって，最適化は「曲がった谷底を探す」作業ではなく，「平面を押し当てていって最後に接する場所を探す」作業とみなせる．このイメージは，非線形最適化と対比すると分かりやすい．非線形最適化では局所的なくぼみや鞍点が現れうるが，線形計画ではそのような複雑さはなく，幾何学的構造が整理されている．

この幾何学的な見方から分かる最も重要な事実は，線形計画問題の最適解は実行可能領域の頂点に存在するということである．厳密には，実行可能領域が空でなく，目的関数が下に有界であるならば，最適解は少なくとも一つの極点で達成される．これは，線形関数が凸多面体の内部で曲がることなく一定方向に変化するため，最小値は境界，さらにいえば頂点に押し出されるからである．

この事実は，線形計画法の理論とアルゴリズムの双方にとって決定的に重要である．もし最適解が一般に領域内部に現れるのであれば，すべての点を連続的に調べる必要があり，問題ははるかに

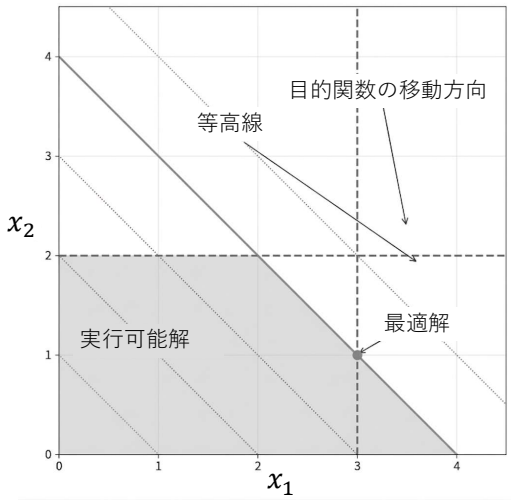


図 4.3: 線形計画問題における実行可能領域と最適解

難しくなる．しかし，最適解が頂点に存在するならば，頂点だけを調べればよいという発想が可能になる．もちろん，頂点の数も大きな問題では膨大になりうるが，「連続無限個の候補」ではなく「有限個の頂点」に問題が還元されることの意味は大きい．

また，線形計画では，複数の最適解が存在することもある．これは，目的関数の等高線が実行可能領域のある辺と平行になり，その辺全体で同じ目的関数値をとる場合に起こる．このとき，最適解は一つの頂点に限らず，辺上の無限個の点にわたって存在する．ただし，その場合でも，少なくとも一つの頂点は最適解である．この性質があるからこそ，頂点探索型のアルゴリズムが依然として有効である．

4.2.3 標準形と実行可能基底解

線形計画法では，問題をある決まった形に変換して扱うことが多い．その代表が標準形である．標準形は通常，

$$\text{minimize} \quad c^T x \tag{4.30}$$

$$\text{subject to} \quad Ax = b \tag{4.31}$$

$$x \geq 0 \tag{4.32}$$

という形を指す．一見すると，これは先ほどの一般形よりも特殊に見えるかもしれない．しかし，実際には多くの線形計画問題は，適切な変数変換によってこの形に直すことができる．そのため，理論を整理するときには標準形を前提にすると便利である．

もともとの不等式制約は，スラック変数を導入することで等式制約に変換できる．例えば，

$$a_1 x_1 + a_2 x_2 \leq b \tag{4.33}$$

という制約は，新たな変数 $s \geq 0$ を導入して

$$a_1 x_1 + a_2 x_2 + s = b \tag{4.34}$$

と書ける．この s をスラック変数と呼ぶ．スラック変数は，制約にどれだけ余裕があるかを表している．もし制約がちょうど等号で満たされていれば $s = 0$ であり，まだ余裕があれば $s > 0$ である．この意味で，スラック変数は制約の緩み具合を可視化する変数でもある．

一方，$\geq$ 型制約であれば，余剰変数を引くことで等式に直せるし，自由変数がある場合には，正負二つの非負変数の差として表現することができる．したがって，標準形は人工的な形ではあるが，一般性を大きく損なうことなく用いることができる．このように問題を標準形へ揃えることで，線形計画法の理論を統一的に展開できる．

標準形のもとでは，実行可能解の中でも基底解と呼ばれる解が重要になる．行列 A の列のうち，ランクと同数の列を選んで正則な部分行列を作り，それに対応する変数を基底変数，残りを非基底変数とする．非基底変数を 0 とおいて基底変数を解いたときに得られる解を基底解と呼ぶ．さらに，その基底解が $x \geq 0$ を満たしていれば，それは実行可能基底解である．

この定義は「自由に選ぶ変数」と「制約式によって自動的に決まる変数」を分けて考えている．標準形では等式制約があるため，すべての変数を自由に選べるわけではない．そこで，一部の変数を 0 に固定し，残りの変数を制約式から解く．その結果得られる特別な解が基底解である．これは代数的な視点から見た頂点の表現といえる．

この実行可能基底解は，幾何学的には実行可能領域の頂点に対応している．したがって，線形計画法の理論では，「頂点」と「実行可能基底解」は本質的に同じ対象を異なる見方で捉えたものと考えることができる．幾何学的には頂点，代数的には実行可能基底解，という二つの見方があることで，問題の理解は大きく深まる．

クイズ 2：信号制御

ある都市の四差路交差点において，1 サイクルの信号運用を設計する問題を考える．東西方向と南北方向に対して，それぞれ青時間を設定し，交差点全体のサイクル長は一定とする．また，各方向の青時間には安全上必要な下限制約があり，青時間の総和はサイクル長を超えてはならない．

ここで，車両遅延の評価を単純化し，各方向の総遅延は対応する青時間の一次式で表されるものとする．したがって，交差点全体の総遅延も，各方向の青時間の線形関数として表される．このとき，総遅延を最小化する信号設定を求めたい．

この問題に関する次の記述のうち，本文の内容として最も適切なものはどれか．

(ア) 目的関数と制約条件がいずれも線形であるので，この問題は線形計画問題として表すことができる．ただし，実行可能領域は連続な点の集まりであるから，最適解を求めるにはその内部も含めて全体を連続的に調べる必要があり，頂点に注目する考え方は本質的ではない．

(イ) 目的関数と制約条件がいずれも線形であるので，この問題は線形計画問題として表される．このとき，実行可能領域は線形不等式の共通部分として与えられる凸多面体となる．また，目的関数は線形関数であるため，最適解は実行可能領域の頂点の一つで達成される．さらに，目的関数の等高線が実行可能領域のある辺と平行な場合には，その辺上の複数の点が最適解となりうるが，その場合でも少なくとも一つの頂点は最適解である．

(ウ) 実行可能領域が凸集合であることから，その内部の任意の 2 点を結ぶ線分上の点も実行可能である．したがって，線形目的関数の値もその領域内で一定となり，頂点・辺・内部点のいずれを選んでも目的関数値は変わらない．このため，線形計画問題では最適解を特定の点として議論する必要はない．

→ 正解・解説は節末を参照

4.2.4 双対問題

線形計画法におけるもう一つの極めて重要な概念が双対性である．ある線形計画問題に対して，それと対をなす別の線形計画問題を定義することができる．もとの問題を主問題，対応して作られる問題を双対問題と呼ぶ [22].

例えば，主問題が

$$\text{minimize} \quad c^T x \tag{4.35}$$

$$\text{subject to} \quad Ax \geq b \tag{4.36}$$

$$x \geq 0 \tag{4.37}$$

であるとき，その双対問題は

$$\text{maximize} \quad b^T y \tag{4.38}$$

$$\text{subject to} \quad A^T y \leq c \tag{4.39}$$

$$y \geq 0 \tag{4.40}$$

となる．ここで，y は双対変数である．この関係を見ると，主問題と双対問題の間には明確な対応があることが分かる．主問題の制約の本数が双対問題の変数の数になり，主問題の変数の数が双対問題の制約の本数になる．また，不等号の向きや最大化・最小化の関係にも一定の規則がある．

双対変数は，各制約に対応する「影の価格」や「限界価値」として解釈される．例えば，ある資源制約に対応する双対変数が大きいならば，その資源を 1 単位増やしたときの価値が高いことを意味する．言い換えれば，その資源が現在の最適解において希少であり，ボトルネックになっていることを表している．この意味で，双対問題は主問題の背後にある資源配分や価格付けの構造を明らかにしてくれる．

双対性は，線形計画法において美しい形で成立する．まず弱双対性によれば，主問題の任意の実行可能解 x と双対問題の任意の実行可能解 y に対して，

$$b^T y \leq c^T x \tag{4.41}$$

が成り立つ．これは，双対問題の目的関数値が主問題の目的関数値の下界あるいは上界を与えることを意味する．最小化問題と最大化問題がこのような不等式で結ばれているという事実は，重要である．というのも，もしある主実行可能解と双対実行可能解について両者の目的関数値が一致すれば，その瞬間に両者が最適であることが分かるからである．

さらに，強双対性によれば，適当な条件のもとで主問題と双対問題の最適値は一致する．すなわち，

$$c^T x^* = b^T y^* \tag{4.42}$$

が成り立つ．この事実は，線形計画法を特別なものにしている核心である．なぜなら，主問題を直接見るだけでは分からない最適性を，双対問題を通じて理解できるからである．また，最適解の意味を「変数の値」だけでなく「資源の価値」という別の視点から捉えられるようになる．

後に学ぶ列生成法では，双対価格を用いて新しい列の有用性を判定する．ベンダーズ分解では，部分問題の双対情報がカットとして主問題へ返される．さらに，ネットワーク最適化や資源配分問題でも，双対変数の経済的意味づけは極めて重要である．したがって，双対問題の導入は，線形計画法の理解を深めるだけでなく，後続の多くの手法の基礎を与えるものである．

4.2.5　KKT 条件と相補性条件

最適化問題において,「この解が本当に最適か？」を判定するための条件として, KKT 条件（Karush–Kuhn–Tucker 条件）がある．これは，不等式制約を含む問題に対する最適性条件であり，ラグランジュの未定乗数法を拡張したものである．

一般に，次のような問題を考える：

$$\text{minimize} \quad f(x) \tag{4.43}$$

$$\text{subject to} \quad g_i(x) \leq 0 \quad (i = 1, \ldots, m) \tag{4.44}$$

$$h_j(x) = 0 \quad (j = 1, \ldots, p) \tag{4.45}$$

このとき，適当な条件のもとで，最適解 x^* は次の 4 つの条件（KKT 条件）を満たす：

(1) 停留条件「目的関数と制約をすべて考慮したとき，これ以上改善できない状態」

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{p} \mu_j \nabla h_j(x^*) = 0 \tag{4.46}$$

(2) 実行可能性「もとの制約を満たしている」

$$g_i(x^*) \leq 0, \quad h_j(x^*) = 0 \tag{4.47}$$

(3) 双対実行可能性「不等式制約に対応する係数（ラグランジュ乗数）は負にならない」

$$a_i \geq 0 \tag{4.48}$$

(4) 相補性条件

$$\lambda_i \, g_i(x^*) = 0 \tag{4.49}$$

特に重要なのが，相補性条件である．これは，制約に余裕があるなら，その制約は最適解に影響していないということを表している．

実際，相補性条件 $\lambda_i g_i(x^*) = 0$ は，次のどちらかが必ず成り立つことを意味する：

- 制約に余裕がある（$g_i(x^*) < 0$）$\Rightarrow \lambda_i = 0$
- ラグランジュ乗数が正（$\lambda_i > 0$）$\Rightarrow$ 制約はぴったり効いている（$g_i(x^*) = 0$）

つまり，「効いている制約（境界にある制約）だけが最適解を決める」という構造になっている．

この考え方は重要であり，例えば線形計画では，最適解は「いくつかの制約がちょうど等号になる点（＝頂点）」に現れるが，これはまさに相補性条件の直感と一致している．

さらに，線形計画のような凸最適化問題では，KKT 条件は必要条件ではなく十分条件にもなる．したがって，KKT 条件を満たす解を見つける $ftrightarrow$ 最適解を見つけるという強い性質が成り立つ．

4.2.6 双対性と KKT 条件の関係

KKT 条件は，線形計画法において簡潔な形で現れる．線形計画問題は凸最適化問題であるため，KKT 条件は最適性の必要条件であるだけでなく十分条件でもある．つまり，KKT 条件を満たす点は最適解である．この点は，線形計画法が理論的に扱いやすい理由の一つである．

線形計画における KKT 条件を見れば，停留条件はまさに主問題と双対問題の制約に対応し，相補性条件は主変数と双対変数の間の関係を表している．この相補性条件は，相補スラックネスと呼ばれ，線形計画における最適性の判定条件として重要である．例えば，制約に余裕があるならば対応する双対変数は 0 であり，双対変数が正ならば対応する制約はちょうど等号で成立していなければならない．この関係は，KKT 条件の一般形を最も分かりやすく具体化したものといえる．

少し丁寧に見るために，主問題を

$$\begin{aligned} \text{minimize} \quad & c^T x && (4.50)\\ \text{subject to} \quad & Ax \geq b && (4.51)\\ & x \geq 0 && (4.52) \end{aligned}$$

とし，双対問題を

$$\begin{aligned} \text{maximize} \quad & b^T y && (4.53)\\ \text{subject to} \quad & A^T y \leq c && (4.54)\\ & y \geq 0 && (4.55) \end{aligned}$$

とする．このとき，相補スラックネスは概念的には，

$$y_i\big((Ax-b)_i\big)=0 \tag{4.56}$$

および

$$x_j\big((c-A^Ty)_j\big)=0 \tag{4.57}$$

のような形で表される．これは，主問題側で余裕のある制約には双対変数がつかず，双対問題側で余裕のある制約には主変数が正にならないことを意味する．

この関係を経済的に解釈すると分かりやすい．ある資源制約にまだ余裕があるならば，その資源は追加しても価値がないので，双対価格は 0 になる．逆に，双対価格が正であるならば，その資源はちょうど使い切られており，1 単位増やすことに価値がある．このように，相補スラックネスは「余っている資源には価格がつかず，価値のある資源は使い切られている」という直感を表している．

また，KKT 条件と双対性の関係を理解すると，最適性判定の考え方が明確になる．主問題の実行可能解と双対問題の実行可能解を一つずつ持ってきて，さらに相補スラックネスが成り立っていれば，その二つはともに最適解である．これは，弱双対性による上下界の関係と，KKT 条件による局所的な整合条件とが，線形計画法では一つに統合されていることを意味している．

したがって，双対性と KKT 条件の関係を理解することは，「最適性とは何か」を深く理解することにつながる．また，後に学ぶ列生成法やベンダーズ分解では，双対解がアルゴリズムの中で直接使われるため，ここで相補スラックネスや双対変数の意味をしっかり理解しておくことが重要である．

クイズ 3：双対性による最適性判定

ある都市の道路ネットワークにおいて，各リンクの交通流量（台/時）を連続変数とし，総走行コストを最小化する線形計画問題を考える．各リンクには容量制約があり，それを超えないように流量を決定する．
あるとき，この主問題に対して，実行可能な解 x と，対応する双対問題の実行可能解 y が得られたとする．それぞれの目的関数値を計算したところ，

$$c^Tx=b^Ty$$

が成り立っていることが確認された．
この状況に関する次の記述のうち，本文の内容として最も適切なものはどれか．

(ア) 主問題と双対問題の目的関数値が一致していても，それぞれが最適解であるとは限らない．最適性を判定するためには，さらに別の条件を確認する必要がある．

(イ) 主問題と双対問題の目的関数値が一致しているので，弱双対性によりこの関係は常に成立する．したがって，この一致から最適性について特別な結論を導くことはできない．

(ウ) 主問題と双対問題の実行可能解に対して目的関数値が一致しているので，それらはともに最適解である．

→ 正解・解説は節末を参照

4.2.7 代表的な解法

線形計画問題は凸多面体構造を持つため，その幾何学的性質を活かした効率的な解法が存在する．ここでは，代表的な解法である単体法と内点法について，その基本的な考え方と特徴を詳しく説明する．

(a) 単体法

単体法は，線形計画問題の最適解が実行可能領域の頂点に存在するという性質に基づくアルゴリズムである．基本的な考え方は，ある実行可能基底解（頂点）から出発し，隣接する頂点へと移動しながら目的関数値を改善していくことである．

標準形

$$\text{minimize} \quad c^T x \tag{4.58}$$

$$\text{subject to} \quad Ax = b \tag{4.59}$$

$$x \geq 0 \tag{4.60}$$

を考えると，基底変数と非基底変数に分けることで，基底解が定まる．単体法では，非基底変数のうち，目的関数を改善する方向に寄与するものを選び（入る変数），対応して基底変数の一つを外す（出る変数）ことで，新しい基底解へと移動する．

この操作はピボット操作と呼ばれ，基底の更新として代数的に実行される．幾何学的には，これは多面体の辺に沿って隣接する頂点へ移動する操作に対応している．

単体法の重要な特徴は，

- 常に実行可能解を保ちながら探索を行う
- 各ステップで目的関数値が単調に改善される

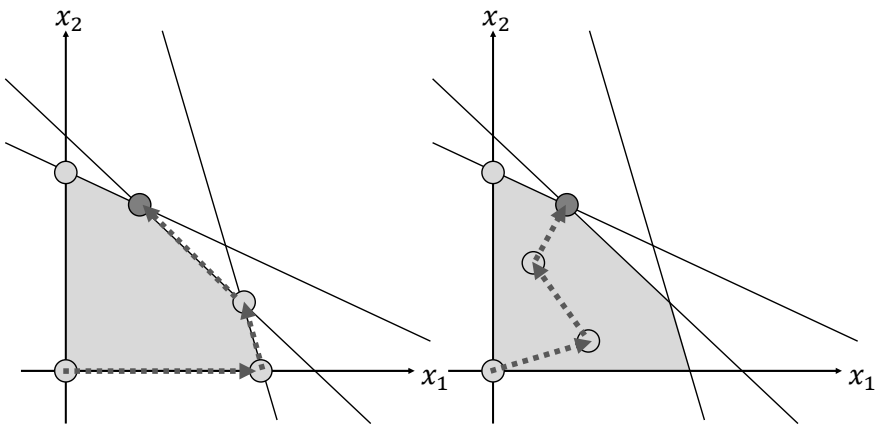
という点である．

また，最適性の判定は，すべての非基底変数に対して改善方向が存在しないこと，すなわち還元コストが非負であることによって行われる．この条件は双対問題の実行可能性とも対応しており，単体法は主問題と双対問題を同時に満たす方向へ進んでいると解釈できる．

理論的には，単体法は最悪の場合指数時間を要することが知られているが，実務上は高速に収束することが多く，長年にわたり広く用いられてきた．

(b) 内点法

内点法は，単体法とは異なり，実行可能領域の内部を通る経路をたどって最適解に近づくアルゴリズムである．



(a) 単体法　(b) 内点法

図 4.4: 単体法と内点法

基本的な考え方は，不等式制約を直接扱う代わりに，境界に近づくことを抑制するバリア関数を目的関数に加えた問題を解くことである．例えば，非負制約 $x_i > 0$ に対しては，

$$-\mu \sum_i \log x_i \tag{4.61}$$

のような対数バリア関数を導入する．

すると，元の問題は

$$\min\ c^T x - \mu \sum_i \log x_i \tag{4.62}$$

のような補助問題に置き換えられる．ここで，μ を徐々に小さくしていくことで，解は境界に近づき，最終的に元の線形計画問題の最適解へ収束する．

内点法では，各反復において，この補助問題の最適性条件（KKT 条件）を解くことで，解を更新していく．特に，ニュートン法に基づく方向を用いて，効率的に中心経路（centralpath）に沿って移動する．

単体法との大きな違いは，

- 頂点ではなく内部の点を通る
- 各反復で連続的に解を更新する

という点である．

理論的には，内点法は多項式時間で解けることが示されており，特に大規模かつ疎な問題に対して高い性能を発揮する．また，数値的に安定であるという利点もある．

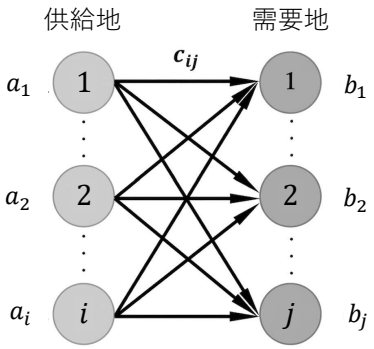


図 4.5: 輸送問題

4.2.8 線形計画法の例

ここでは，線形計画法がどのように現実問題に適用されるかを理解するために，代表的な二つの問題である輸送問題と生産計画問題を示す．

(a) 輸送問題

複数の供給地と需要地の間で物資を輸送する問題を考える [11]．供給地 i の供給量を a_i，需要地 j の需要量を b_j，供給地 i から需要地 j への単位輸送コストを c_{ij} とする．

変数 x_{ij} を「供給地 i から需要地 j に輸送する量」とすると，問題は次のように定式化される：

$$\text{minimize} \quad \sum_i \sum_j c_{ij} x_{ij} \tag{4.63}$$

$$\text{subject to} \quad \sum_j x_{ij} = a_i \quad (alli) \tag{4.64}$$

$$\sum_i x_{ij} = b_j \quad (allj) \tag{4.65}$$

$$x_{ij} \geq 0 \tag{4.66}$$

第 1 の制約は各供給地の供給量を使い切ることを表し，第 2 の制約は各需要地の需要を満たすことを表している．この問題はネットワーク構造を持つ線形計画問題の典型例であり，物流やサプライチェーン設計で広く用いられる．

(b)　生産計画問題

複数の製品を生産する際に，限られた資源をどのように配分するかを決める問題を生産計画問題という [7]．製品 j の生産量を x_j，単位利益を c_j，資源 i の利用可能量を b_i，製品 j が資源 i を消費する量を a_{ij} とする．

このとき，総利益を最大化する問題は次のように書ける：

$$\text{maximize} \quad \sum_j c_j x_j \tag{4.67}$$

$$\text{subject to} \quad \sum_j a_{ij} x_j \le b_i \quad (alli) \tag{4.68}$$

$$x_j \ge 0 \tag{4.69}$$

ここで，各制約は資源制約を表している．例えば労働時間，原材料，設備能力などがこれに対応する．双対変数は，これらの資源の「限界価値（影の価格）」として解釈できる．

この問題は，企業の生産計画だけでなく，エネルギー供給，交通容量配分，都市物流における資源配分など，多様な分野で基本モデルとして用いられている．

クイズ 2 正解・解説

正解：（イ）

この問題では，各方向の青時間が変数であり，総遅延はそれらの一次式として表されている．また，サイクル長制約や各方向の下限制約も線形である．したがって，この問題は目的関数と制約条件がともに線形で与えられる線形計画問題である．

線形計画問題では，実行可能領域は線形不等式の共通部分として与えられるため，凸多面体となる．さらに，目的関数も線形であるため，最小値は実行可能領域の頂点の少なくとも一つで達成される．この「最適解は頂点に存在する」という性質が，線形計画法の理論や単体法のようなアルゴリズムの基礎になっている．

また，複数の最適解が存在する場合もある．たとえば，目的関数の等高線が実行可能領域のある辺と平行であれば，その辺上のすべての点が同じ目的関数値をとる．しかし，その場合でも少なくとも一つの頂点は最適解に含まれる．

(ア) が誤りである理由は，実行可能領域が連続集合であっても，線形計画問題では最適解を頂点に限定して考えられるという重要な性質を見落としている点にある．実行可能領域全体を「内部も含めて一様に調べる必要がある」という理解は，線形計画法の本質を外している．

(ウ) が誤りである理由は，凸集合であることと，目的関数値が領域全体で一定であることを混同している点にある．凸性が意味するのは，「実行可能な 2 点を結ぶ線分上の点も実行可能である」という幾何学的性質であって，目的関数値が同一になることではない．線形目的関数は実行可能領域上で一定とは限らず，方向に応じて増減する．そのため，最適解をどこで達成するかは重要な問題であり，頂点・辺・内部点を区別して考える必要がある．

クイズ 3 正解・解説

正解：(ウ)

この問題の本質は，双対性を用いた最適性判定である．

線形計画問題では，弱双対性により，任意の主問題の実行可能解 x と双対問題の実行可能解 y に対して，

$$b^T y \leq c^T x$$

が成り立つ．これは，双対問題の目的関数値が主問題の目的関数値の下界を与えることを意味する．

ここで，

$$c^T x = b^T y$$

が成立しているということは，この不等式が等号で成立していることを意味する．

したがって，これ以上目的関数値を改善する余地はなく，x と y はそれぞれ主問題と双対問題の最適解である．これは強双対性の帰結であり，線形計画法における最も重要な最適性判定の一つである．

(ア) は，等号成立の意味を理解しておらず誤りである．

(イ) は，弱双対性を誤解している．弱双対性は不等式を保証するものであり，等号が成立することは特別な状況であり，最適性を示す強い条件である．

コラム：バスのルート最適化の手法提案と比較 – 桑本球太郎

社会便益を最大化するバスの最適ルートを探索するための MFD を基礎としたヒューリスティクス手法の比較と，ミクロへの応用について研究を行った．用いたデータは 2007 年松山都市圏パーソントリップ調査の個票データ（出発時刻が 7 時から 9 時で代表交通利用手段の分類が自動車のもの）と，松山市全域（主要な道路，すべて 1 車線と仮定）と市街地（自動車通行可能なリンク，車線数考慮）の 2 種類の道路ネットワークを用いた．
はじめに，松山市全体の道路ネットワークを PT 調査のゾーンごとに分割し（図 4.6），各エリアの隣接関係からマクロネットワークを構築した（図 4.7）．

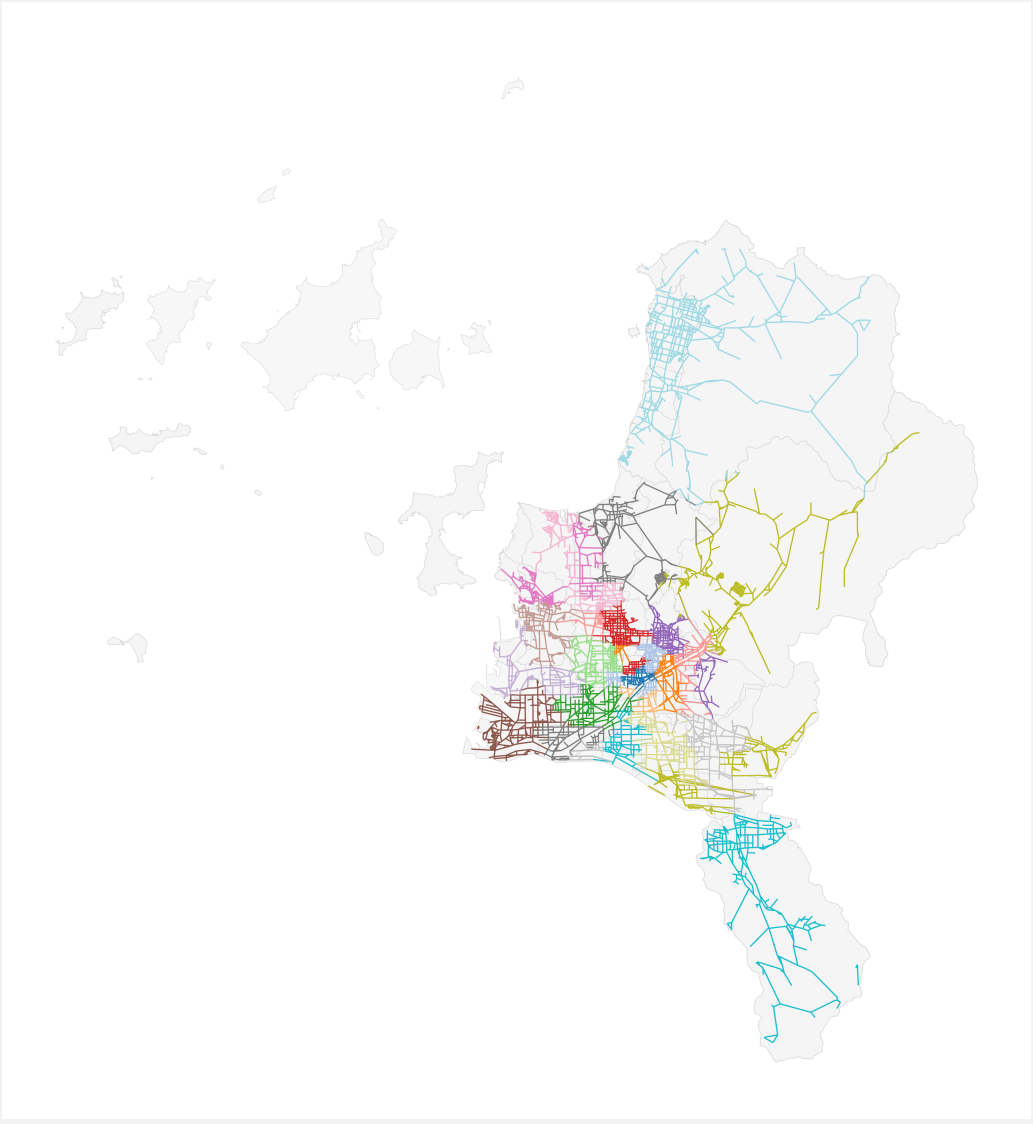


図 4.6: マクロゾーン分割

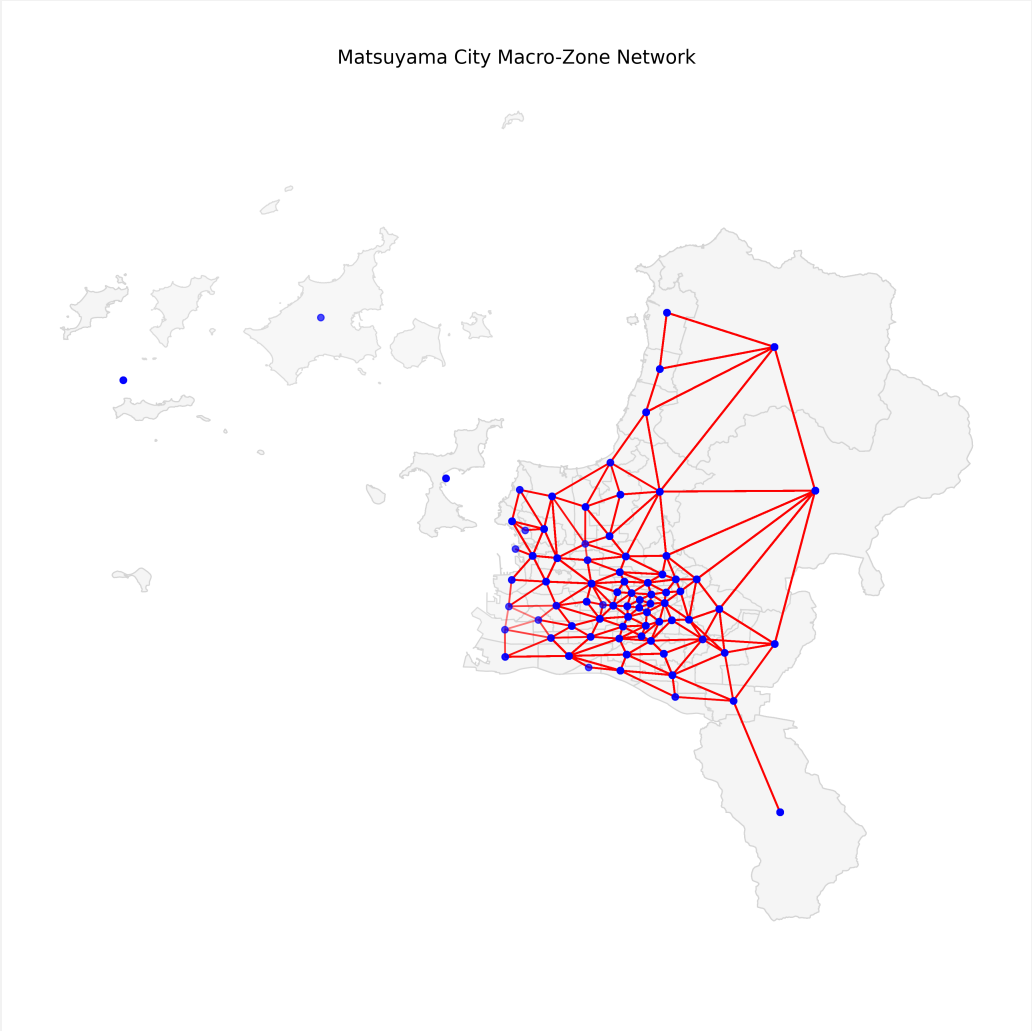


図 4.7: マクロネットワークの構築

その後，各ゾーンの総車線距離を計算し，そこに車の限界密度（1 キロに 120 台）を設定して乗じることで，ゾーンごとの最大滞在可能台数を算出した．また，各ゾーン内での動きをゾーン内完結，ゾーン出発，ゾーン到着，ゾーン通過の 4 つに分類しそれぞれの平均最短経路距離を算出した．

次に，これらを元に，実際の混雑状況の再現を試みた．具体的には，ゾーン内にいる車両数と最大滞在可能台数の比率から，

$$V = \frac{V_f}{1 + \alpha (n / N_{\max})^{\beta}} \tag{4.70}$$

という式 (4.70)（BPR 関数）を用いてそのゾーンでの速度を導出する．ここで，V はそのゾーンでの実際の速度，V_f は自由旅行速度（混雑がない状態での速度），n はゾーン内にい

る現在の車両数，$N_{\max}$ はゾーンの最大滞在可能台数（交通容量），α および β は交通特性を表すパラメータである．
先に求めた平均最短経路距離をこの速度 V で割ることで，そのゾーンの通過に何分かかるのかを計算する．このプロセスを 7:00 から 9:59 まで 1 分おきで行い，各ゾーンの通過にかかる時間を書き換え，その 1 分に発生する車（PT の時間平均とポアソン分布を用いて決定）の最短経路計算に用いることで，「車が増える $\rightarrow$ 速度が落ちる $\rightarrow$ 後から来た車が別のルートを選ぶ（ダイクストラ法）$\rightarrow$ 車が移動して別のゾーンが混む」というダイナミクスを表現し，1 分ごとの各状態でどのゾーンに何台の車がいたのかを記録した．以上の結果を用いて最適なバスのルートの最適化を考える．目的関数には，社会的便益を設定した．具体的には，新設される 10 ゾーンを巡回する環状バス路線に対し，各 OD の移動者が自動車とバスのどちらを選択するかを，所要時間と費用を変数としたロジットモデルを用いて確率的に推定する．バスへの転換によって得られる利用者全体のログサム便益から，バスの運行コストを差し引いた社会的便益 SW は次式で定義される．

$$SW = \sum_{od} \left(\frac{\ln(\exp(V_{car}) + \exp(V_{bus})) - \ln(\exp(V_{car}))}{|\beta_{time}|} \cdot VOT \cdot D_{od} \right) - C_{op} \cdot T. \quad (4.71)$$

ここで，V_{car}, V_{bus} はそれぞれ自動車とバスの効用関数，β_{time} は時間パラメータ，VOT は時間価値，D_{od} は OD 間の交通需要，C_{op} は単位時間あたりの運行コスト，T はバスの総運行時間を表す．本研究ではこの SW を目的関数とし，最適化のスコアとしている．
また制約条件としては以下を組み込んだ．

- **ルート長の固定:** 短すぎるルートでは意味をなさないので，10 ゾーンを巡るように固定した．
- **同一ゾーンの重複禁止:** 評価関数の冒頭では同一ゾーンを通っているものに極端なマイナス効用を与え，探索候補から除外されるように設計するとともに，GA は交差時に重複したノードはルート上にない他のゾーンから新しいものを選ぶようにし，SA と TS は Replace の構造上重複が起きないようにしている．
- **バスの輸送量制約:** 各区間の乗客数を集計し，最も混む区間の乗客数が 450 人を超えた場合は 450 人までしかバスに乗せないように実装している．
- **経路の連続性とペナルティ:** 空間的に隣接していないゾーン間をワープするような非現実的なルート生成を防ぐため，隣接していないゾーン間のつながりに対しては物理的距離に基づく迂回ペナルティを課した．
- **輸送容量の制約:** バスの最大定員を設定し，特定区間に需要が集中して定員を超過した場合は，超過分の転換行動を無効化することで便益の過大評価を防いだ．

- **混雑緩和のフィードバック（静的均衡へのすり合わせ）：**先の動的シミュレーションで得られた平均滞在台数を初期の背景交通量として設定した．バスへの転換によって自動車の交通量が減少すると，BPR 関数に従い道路の走行速度が向上する．速度が向上すると自動車の利便性が回復し，バスへの転換率が再び低下するという需要と供給の反復が発生する．本研究ではこの再計算を 3 回反復させている（前述の動的な交通配分のシミュレーションを毎回実行する計算コストを回避するため，ルート選択は無視し，モード選択にのみ注目している点に注意）．

上記の都市全体のゾーンから最適な 10 ゾーンの組み合わせと巡回順序を決定する問題は，膨大な探索空間を持つ組み合わせ最適化問題である．そのため本研究では，性質の異なる 3 種類のメタヒューリスティクス手法（GA，SA，TS）を実装し，探索性能と計算時間の比較を行った．公平な比較のため，各手法の評価回数が約 10,000 回となるようにパラメータを調整した（計算時間は環境に依存するため）．各手法のパラメータについて表 4.1 に示す．

表 4.1: メタヒューリスティクス各手法のパラメータ設定

手法	パラメータ	設定値と役割
GA	n（個体数）	100（一度に 100 個の異なるルートを保持）
	$ngen$（世代数）	100（100 個体 × 100 世代 = 10,000 回評価）
	$cxpb$（交差確率）	0.7（70%の確率で，親のルートを途中で切って繋ぎ合わせる）
	$mutpb$（突然変異確率）	0.2（20%の確率で，ルートの一部を入れ替えたり別のゾーンへ変更する）
	$tournsize$（選択圧）	3（ランダムに 3 つのルートを選び，一番点数が高いものを親とする）
SA	T（初期温度）	100000.0（序盤は改悪幅が大きくても確率的に許容する）
	終了条件	$T \leq 10.0$（温度が 10 以下になったら探索終了）
	$cooling_rate$（冷却率）	0.99908（1 回の評価ごとの冷却値．約 10,000 回で終了温度まで下がる）
	近傍生成	1（1 回のループにつき，変更を 1 つだけ提案させる）
TS	max_iter（最大反復数）	500（500 ステップ移動を繰り返す）
	$num_neighbors$（近傍サイズ）	20（1 歩進む前に，周りの 20 パターンの変更案をチェックする）
	$tabu_tenure$（タブー期間）	20（過去 20 回分の操作を記憶し，同じ操作を禁止する）

基礎概念：近傍生成

タブーサーチと焼きなまし法において，探索の要となるのが近傍生成の設計である．本モデルでは以下の 2 つの操作を組み合わせて新しいルートの候補を作成した．

- Swap（入れ替え）: ルート内の 2 つのゾーンの順番を入れ替える（例: A-B-C → A-C-B）．通過するエリアは変えずに，バスの巡回順序を変更する．
- Replace（差し替え）: ルート内の 1 つのゾーンを捨てて，まだ使われていない新しいゾーンを組み込む（例: A-B-C → A-B-Z）．ルートが通る「エリア」自体を変更する．

各手法は，この操作を活用して最適解へと迫っている．

遺伝的アルゴリズム（GA: Genetic Algorithm）

100 個体を保持し，100 世代にわたり以下の操作を繰り返す．交差（Crossover）: スコアの高かった優秀なルート A と，ルート B を選び，あるポイントで分割して繋ぎ合わせる．修復（Repair）: 繋ぎ合わせた結果，同じゾーンを 2 回通ってしまうルール違反が起きた場合，

重複したゾーンを削除し，まだ使っていないゾーンからランダムに引っ張ってきて穴埋めを行う．突然変異（Mutation）：世代交代の際，20%の確率で個体に対して Swap や Replace を強制的に行い，新しい変化をもたらすことで，多様性の喪失を防ぐ．

焼きなまし法（SA: Simulated Annealing）

1 ステップで Swap か Replace をランダムに行い，1 つの近傍解を作成する．序盤はスコアが悪化する「改悪解」も $\exp(\Delta/T)$ という確率で許容することで局所最適解に陥ることを防いだ．徐々に冷却率に従って温度を下げることで，最終的に最適な環状ルートへと収束させた．

タブーサーチ（TS: Tabu Search）

現在のルートの近傍を，Swap か Replace をランダムに行うことで毎ステップ 20 個生成し，最良のものへ移動する局所探索を行った．その際，直近の改変履歴をタブーリストに保持し，次のステップで同じ操作を禁止することで探索のループを防ぎ，効率的に未探索エリアを開拓させた．

性質の異なる上記の 3 つのメタヒューリスティクス手法（GA，SA，TS）について，それぞれ 20 回の独立した試行を行い，「社会的便益（スコア）」と「実行時間」の比較を行った結果が図 4.8 および図 4.9 である．バスの運行形態として直線ルート（往復なし）を想定した初期モデルと，実用性を考慮して始点と終点を結んだ循環ルートを想定した提案モデルの 2 パターンで評価を実施した．

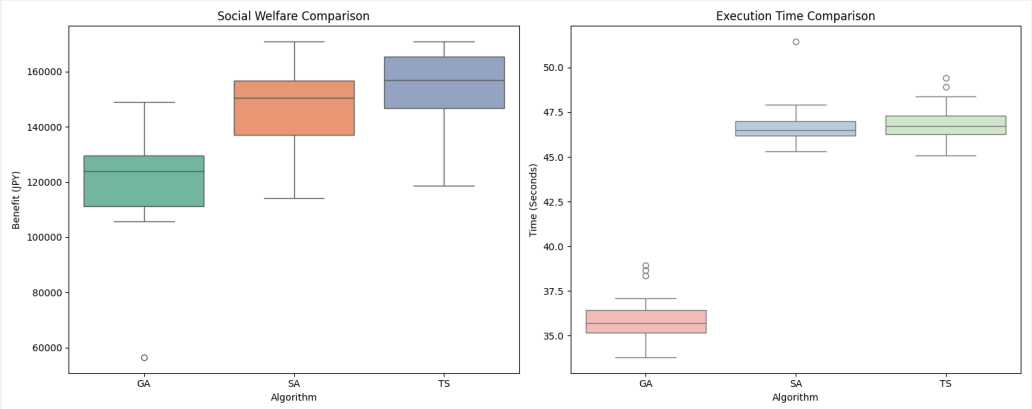


図 4.8: ループなしでの各手法の社会的便益と実行時間の比較

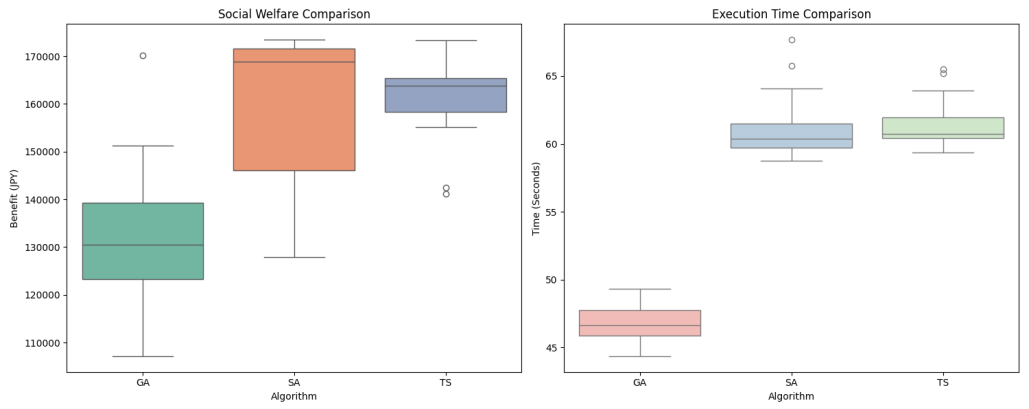


図 4.9: ループありでの各手法の社会的便益と実行時間の比較

結果を比較すると，ルートの前提条件によって最適な探索アルゴリズムが変化するという興味深い結果が得られた．直線ルート想定時には，TS が最も高いスコアを安定して記録した．TS は過去の改変履歴を記憶して重複探索を防ぐため，単純な一直線のルートを無駄なく効率的に探索し当てることに優れていたと考えられる．一方，本研究の最終的な提案モデルである循環ルート想定時においては，SA が最も高い便益（中央値 約 169,000 円，最大値 170,000 円超）を記録した．循環ルートは「始点と終点が繋がる」という制約が加わるため，探索空間の地形がより複雑になり，局所最適解に陥りやすくなる．SA の「序盤はスコアが悪化する変更も確率的に許容する」という物理的特性が，中途半端な局所解からの脱出を促し，最も広域の需要を拾える大域的最適解へ到達できたものと推察される．

同条件における TS は，箱ひげ図の四分位範囲が非常に狭く，安定して高い解を出力する堅牢さを見せたものの，絶対的な最大値や中央値では SA に及ばなかった．これは，TS が局所探索に優れる反面，循環ルートのような複雑な制約下では，タブーリストだけでは乗り越えられない大きな谷に直面し，そこそこの良いルートで探索が停滞してしまったためと考えられる．

GA は，両条件においてスコアの中央値が最も低く，試行ごとのばらつきも非常に大きかった．ルートの一部を入れ替える交叉を行う際，同じゾーンを 2 回通るルール違反が頻発し，それをランダムなゾーンで強制修復する処理が介入したことで，世代ごとの優秀なルートの継承が阻害されたためであると考えられる．強制修復の過程で需要量と距離などを反映させて探索することで改善できるとも考えられる．

また，実行時間のグラフからは，評価回数を揃えたことによる妥当性も確認できる．循環ルートにおける SA と TS は，どちらも実行時間が約 60 秒で完全に横並びとなっている．これは，両者が全く同じ回数（10,000 回）だけ重たい交通均衡フィードバックを実行したことを示しており，アルゴリズム間の純粋な「1 万回の探索の質」の比較が担保されていることの証明である．GA の実行時間が約 46 秒とやや短いのは，交差や突然変異によって生じた「重複ルート」が評価関数の冒頭で即座に弾かれるケースが含まれているためであり，実

質的な有効探索回数が他の 2 手法より少なかったことも，スコア低迷の一因と考えられる．
次にここまでで得られたマクロな最適ゾーン経路（回廊）を基敷とし，実際の交差点（ノード）と道路リンクからなる松山中心市街地のミクロレベルの道路ネットワークへの適用と，最終的なバス路線の再最適化を行った．
まず，マクロ最適化で得られたゾーン単位の経路を，実際の道路ネットワークへと落とし込んだ．具体的には，PT 調査のゾーンポリゴンとミクロな道路のノードの空間結合を実施した（図 4.10）．

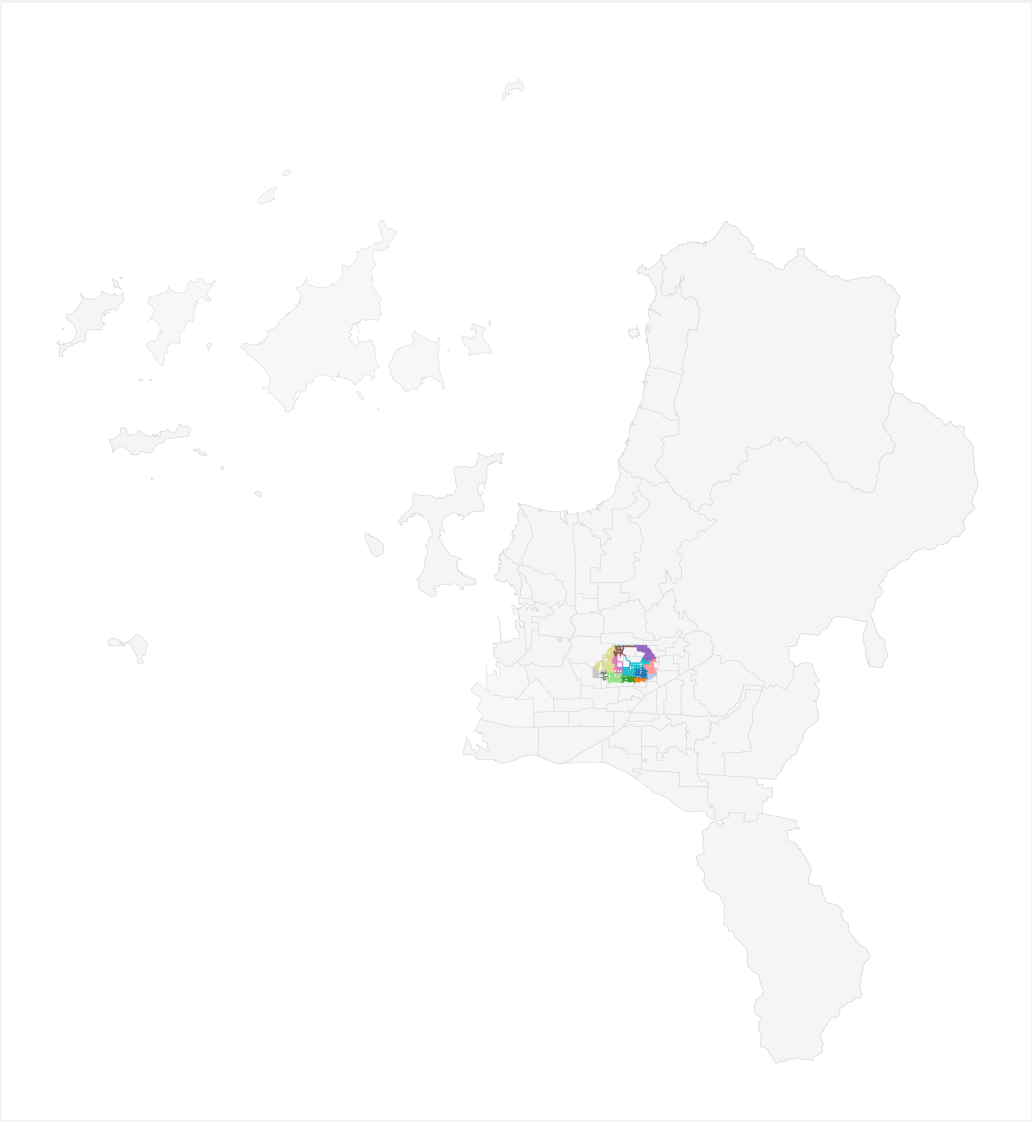


図 4.10: ミクロゾーンの空間結合

各ゾーン内において，自動車が通行可能なリンクに接続しており，かつ最も接続リンク数

の多いノードを代表ノードとして抽出し，マクロな経路を具体的な交差点の移動へと変換した．
ミクロレベルの評価においては，都市部の複雑な交通動態を反映するため，評価モデルに以下の精緻化を加えた．

- **実ネットワークでの渋滞考慮:** 各道路リンクの車線数から物理的な交通容量を算出し，実際の背景交通量データに基づく BPR 関数を適用した．これにより，幹線道路と生活道路のキャパシティの違いや，交差点間における慢性的な渋滞による遅延をシミュレーションに組み込んだ．
- **最低運行距離制約の導入:** ミクロ空間において厳密な運行コスト（1 分あたりの費用）を計上すると，探索アルゴリズムがコストを削るために，極端に近い交差点だけを巡回してすぐに車庫へ帰るという自明な局所解へ収束してしまう問題が生じる．これを防ぐため，最低運行距離制約を設け，それに満たないルートにはペナルティを課す設計とした．

ミクロレベルでの経路最適化には，マクロ評価で最も堅牢な性能を示したタブーサーチ（TS）を再度採用した．ミクロな道路網においては，GA の交叉処理（ルートの分断と結合）を用いると，物理的に走行不可能な経路や，同じ交差点を何度も往復するルール違反が大量発生し，探索効率が著しく低下してしまった．そのため，ルート内の交差点順序の入れ替え（Swap）と，近隣の別交差点への差し替え（Replace）による局所探索を反復する，階層型の TS モデルを構築した．
ミクロ道路網特有の性質と，階層型アプローチの有効性を検証するため，以下の 4 種類の初期解生成アプローチを用意し，十分な探索回数（300 イテレーション）の TS 探索を行うモンテカルロ評価（各 10 回試行）を実施した．

1. **Macro-TS（提案手法）:** マクロ最適解として選ばれたゾーンの代表ノードから探索を開始する．
2. **Greedy（貪欲法）:** 最も需要の高い交差点を起点とし，そこから物理的に最も近い交差点を順に繋いだ経路から開始する．
3. **Demand-Focus（需要特化）:** 距離を問わず，単純に需要が集中している上位の交差点を結んだ経路から開始する．
4. **Random（ランダム）:** 完全ランダムに選ばれた交差点の組み合わせから開始する．

実験の結果が以下の図 4.11 である．

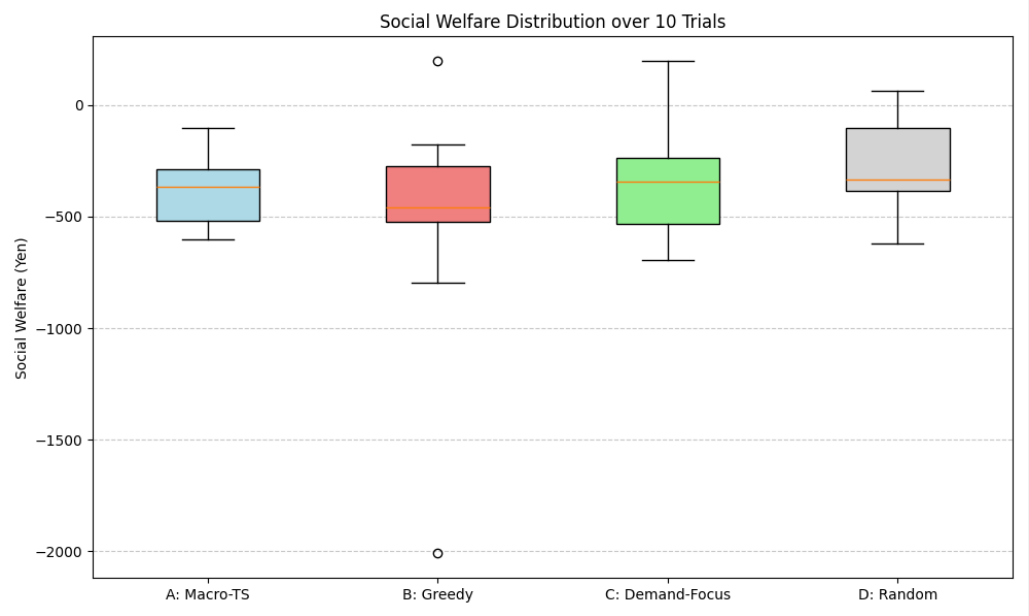


図 4.11: ミクロモデルにおける各初期解生成アプローチの社会的便益の比較

十分な探索回数を与えることで，ミクロ道路網内の渋滞や迂回を避け，社会的便益がプラスとなる実用的なバスルートが複数発見された．しかし，各手法の成績を比較すると，事前の予測に反して非常に興味深い知見が得られた．

第一に，平均スコアおよび勝率において Random が最も優れた成績を収めた点である．第二に，マクロ最適解を用いた「Macro-TS」が，唯一黒字化を一度も達成できず，赤字の局所最適解から抜け出せなかった点である．

この逆転現象は，マクロ最適解が抱えるミクロ空間の呪縛を強く示唆している．マクロレベルで有望とされたゾーン群は，需要が極めて高い反面，実際の道路網で見ると信号が多く，慢性的な渋滞が発生し，迂回を強いられる中心市街地の深部に該当することが多い．Macro-TS は，その渋滞エリアのど真ん中に初期解が固定されてしまうため，TS をもってしても根本的な遅延による運行コスト増大から抜け出せなかったと考えられる．

一方で，ランダム初期化は都市全体に散らばった状態から探索をスタートするため，初期解の多様性が担保される．TS の強力な局所探索能力と十分な探索回数を組み合わせることで，特定の渋滞エリアに過度に固執することなく，走りやすさと需要のバランスが取れた現実的な抜け道を柔軟に開拓できたことが，安定した高スコアに繋がったと推察される．なお，Greedy および Demand-Focus は，全試行の中で最高益を叩き出すポテンシャルを見せたものの，初期配置によっては行き止まりや大迂回の罠にハマり大赤字を出すハイリスク・ハイリターンな挙動を示した．

本研究の階層型シミュレーションを通じ，マクロな直線距離ベースの最適化は有望な需要回廊の発見に寄与するものの，それをそのままミクロな道路網へ適用すると，局所的な道

路構造や渋滞の影響で運行コストが想定以上に増大し，最適解から遠ざかることが判明した．現実のマイクロトランジット路線設計においては，需要の高いエリアをピンポイントで狙う戦略（マクロ最適解や需要特化型）に依存するのではなく，完全ランダム初期化による多様性を確保したマルチスタート型タブーサーチを用いることが，複雑な都市の道路網において高収益かつ安定したバスルートを構築する上で極めて有効であるのではないかと考えた．また，初期値によって分散が大きいものについては計算回数を増やすよりもさまざまな初期値から探索を行うことが有効であるのではないかとも考えられた．

4.3 組合せ最適化の基礎

4.3.1 組合せ最適化とは

最適化問題の中でも，変数が離散的な値をとる問題を，総称して組合せ最適化という．組合せ最適化では，変数は一般に 0–1 変数，整数，集合などとして表される．多数の離散的な候補の中から，ある制約のもとで最も良い組合せを選ぶ問題である．

ここで「組合せ」という言葉は，いくつかの要素を選ぶという意味に限らない．実際には，ある集合から要素を選ぶ問題もあれば，要素の並べ方を決める問題もあり，また，複数の対象をどのように割り当てるかを決める問題もある．例えば，目的地までの経路を決める問題では，どの経路を選ぶかが 0–1 変数で表される．施設配置問題では，どの候補地を選ぶかが 0–1 変数で表される．

組合せ最適化を特徴づける最も大きな点は，決定変数が連続的には動けないことである．連続最適化では，変数の値を少しだけ増やしたり減らしたりしながら，勾配や曲率を利用して最適解に近づくことができる．しかし，組合せ最適化では，変数は 0 か 1，あるいは整数であるため，「少しだけ動かす」という概念がそのままでは成り立たない．例えば，あるリンクを使うかどうかは 0 か 1 でしか表せず，その中間値をとることは意味を持たない．このため，組合せ最適化では，連続最適化で重要であった微分や勾配の考え方は直接には使えず，代わりに探索や列挙，あるいはそれを効率化する工夫が中心となる．

また，組合せ最適化では，解空間の構造そのものが豊かである．例えば，「ある都市の次にどの都市を訪れるか」という局所的な選択が，全体の巡回順序に大きく影響する．施設配置問題では，ある施設を選ぶかどうかが，その周辺の全体コストに波及する．スケジューリングでも，ある一つの仕事の順序変更が，後続の多数の仕事の開始時刻に影響する．このように，離散的な選択は局所的に見えても，全体としては強い相互依存関係を持つことが多い．この相互依存のために，単純な局所判断だけでは全体最適に到達できないことが多く，問題が難しくなる．

組合せ最適化が難しい理由は，候補数が急激に増大することにある．例えば，n 個の都市を一度ずつ訪問する順序の数は $n!$ 通りある．少し都市数が増えただけで候補数は爆発的に増加し，すべてを調べ尽くすことが現実的でなくなる．この現象が組合せ爆発である．重要なのは，この爆発が「候補数の多さ」ではなく，「問題サイズに対して指数的に増える」ことである．例えば，都市数が 5 から 10 に増えるだけで，順序の数は 120 通りから約 360 万通りに増える．さらに 20 都市になれば，その数は天文学的な規模に達する．

このような組合せ爆発のために，組合せ最適化では「最適解を求める」という目標そのものが計算上の困難と直結する．したがって，組合せ最適化を学ぶうえでは，問題を定義するだけでなく，「なぜ難しいのか」「どのような発想でその難しさに立ち向かうのか」を理解することが重要である [12].

4.3.2 整数計画と 0–1 計画

組合せ最適化問題は，しばしば整数計画問題として表現される．整数計画問題とは，決定変数の一部または全部に整数制約を課した最適化問題である．線形計画法では変数は連続的な実数値をと

ることができたが，整数計画では変数が整数値をとるように制約される．この違いは一見すると小さく見えるかもしれないが，問題の難しさを根本的に変える重要な違いである．

一般に，整数計画問題は

$$\text{minimize} \quad c^T x \tag{4.72}$$

$$\text{subject to} \quad Ax \le b \tag{4.73}$$

$$x_i \in \mathbb{Z} \quad (\text{一部または全部の } i) \tag{4.74}$$

のように表される．ここで，目的関数と制約が線形であっても，変数に整数制約が加わるだけで，問題は線形計画法とは全く異なる性質を持つようになる．線形緩和では滑らかな凸多面体の内部を動けたのに対し，整数計画ではその中の格子点だけが許される解になるからである．したがって，連続緩和では簡単に得られる構造が，整数制約によって急に壊れてしまう．

特に，

$$x_i \in \{0, 1\} \tag{4.75}$$

という形の変数を用いる問題は 0–1 計画と呼ばれ，多くの組合せ最適化問題の基本形となる．0–1 変数は，「選ぶか選ばないか」「使うか使わないか」「その条件を満たすか満たさないか」といった二者択一を自然に表現できるため，使いやすい．例えば，施設配置問題では，候補地 i に施設を設置するなら $x_i = 1$，設置しないなら $x_i = 0$ とすることで，配置選択を表現できる．配送計画問題では，リンク (i, j) を通るなら $x_{ij} = 1$，通らないなら $x_{ij} = 0$ として，ルート選択を記述できる．スケジューリングでは，ある作業をある時間帯に割り当てるかどうかを 0–1 変数で表すこともできる．

このように，0–1 変数は，現実の意思決定を直接的に表現することができる．線形計画法では，たとえば「0.3 個の施設を建てる」「0.5 本だけ道路を使う」といった解が数学的には許されてしまうことがあるが，現実にはそのような解は意味を持たない．整数制約や 0–1 制約は，まさにこの「現実に意味のある離散的選択」をモデルに反映するために導入される．したがって，整数計画は，現実問題をより忠実に表現するための自然な枠組みである．

しかし，整数制約を課すと，問題は急に難しくなる．連続変数だけなら線形計画として多項式時間で解ける場合でも，整数制約を加えると NP 困難になることが多い．この違いは，連続最適化と組合せ最適化の本質的な隔たりを示している．例えば，線形計画問題では実行可能領域全体が凸集合であり，最適解は頂点に存在するという美しい理論が成り立つが，整数計画では許される解はその凸多面体の中の離散点だけであり，もはや単体法のように滑らかに動くことはできない．

また，整数計画の重要な考え方として，線形緩和がある．これは，整数制約を一時的に外して，対応する線形計画問題を考えるものである．例えば，0–1 変数

$$x_i \in \{0, 1\} \tag{4.76}$$

を

$$0 \le x_i \le 1 \tag{4.77}$$

に緩めれば，連続問題になる．この線形緩和は，厳密解法における下界計算や，近似解法における解の評価において重要な役割を果たす．したがって，整数計画と線形計画は対立するものではなく，「現実性」と「扱いやすさ」の間をつなぐ関係にあると理解することが大切である．

4.3.3 NP 困難性と計算の難しさ

NP 困難性とは，大まかに言えば，「既知の多くの難しい問題と同程度に難しい」ことを意味する．後述する巡回セールスマン問題や一般の配送計画問題は，代表的な NP 困難問題である．この言葉はしばしば「難しい問題」という意味で使われるが，教科書としては，その意味をもう少し丁寧に理解しておく必要がある．

計算量理論では，ある問題が「サイズ n に対してどの程度の計算時間を必要とするか」を考える．ここで重要なのは，計算時間そのものの絶対値ではなく，問題サイズが大きくなったときにどのような速さで増えていくかである．例えば，計算時間が n^2 や n^3 のように多項式で増えるならば，比較的大規模な問題にも対応しやすい．一方，2^n や $n!$ のように指数的に増えるならば，少しサイズが増えただけで計算不能になる．NP 困難性が重要なのは，多くの組合せ最適化問題に対して，後者のような急激な計算量増大が避けられないと考えられているからである．

ここで重要なのは，NP 困難であることが「絶対に解けない」という意味ではないことである．小規模な問題であれば厳密解法で解けるし，特殊な構造を持つ問題では効率的なアルゴリズムが存在することもある．例えば，最短経路問題はネットワーク最適化の一つであるが，多項式時間で解ける．一方，同じネットワーク上の巡回セールスマン問題になると，急に難しくなる．したがって，NP 困難性は「すべてのケースで不可能」ということではなく，「一般の場合には効率的な厳密解法が期待しにくい」という意味で理解すべきである．

また，NP 困難性は理論的な分類であると同時に，実務上の解法選択にも大きな影響を与える．もし問題が多項式時間で解けるならば，原則として厳密アルゴリズムを設計することが最優先になる．しかし，NP 困難問題では，問題サイズが大きくなるにつれて厳密解法が現実的でなくなるため，近似解法，ヒューリスティクス，メタヒューリスティクス，分解法などをどう活用するかが中心的な課題となる．この意味で，NP 困難性は「なぜ解法を工夫しなければならないのか」を説明する根拠である．NP 困難性を知ることは，「なぜ厳密に最適解を求めることが難しいのか」「なぜ近似的な方法が必要になるのか」を理解するための出発点であり，後に近傍探索やメタヒューリスティクスを学ぶ際の土台となる．

4.3.4 最短経路問題

(a) 問題設定

最短経路問題は，最適化の中で基本的な問題の一つである．都市交通においては，道路ネットワーク上で出発地から目的地まで最も短い距離あるいは最も小さい所要時間を持つ経路を求める問題として現れる．

(b) 定式化

最短経路問題は，0–1 変数を用いて整数計画問題として表現できる．ネットワークは，通常，グラフ $G = (V, E)$ として表される．ここで，V はノード集合，E はリンク集合である．各リンク

$(i,j) \in E$ にはコスト c_{ij} が与えられており，リンク (i,j) を経路に含めるなら $x_{ij} = 1$，含めないなら $x_{ij} = 0$ とすると，目的関数は

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{4.78}$$

で与えられる．

制約は，始点からは 1 単位のフローが流出し，終点には 1 単位のフローが流入し，中間ノードではフロー保存が成り立つこととして記述できる．すなわち，

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = \begin{cases} 1 & (i = s) \\ -1 & (i = t) \\ 0 & (\text{otherwise}) \end{cases} \tag{4.79}$$

である．

(c)　Dijkstra 法

最短経路問題を解く代表的なアルゴリズムが Dijkstra 法である．この方法は，リンクコストが非負である場合に適用できる．基本的な考え方は，始点から各ノードまでの最短距離を，確定済みのノード集合を徐々に広げながら更新していくことである．

始点 s の距離を 0，それ以外のノードの距離を無限大として初期化する．その後，まだ確定していないノードのうち，暫定距離が最も小さいノードを一つ選び，そのノードを経由することで隣接ノードへの距離が改善されるかどうかを調べる．この操作を繰り返すと，各ノードまでの最短距離が確定していく．

図に示す例では，始点（0）から探索が開始される．まず，始点に隣接するノードに対して距離が計算され，それぞれの暫定距離が更新される．次に，その中で最も距離の小さいノード（例えば距離 1 のノード）が選ばれて確定される．続いて，そのノードを経由して到達可能なノードの距離が再計算され，より短い経路が見つかれば値が更新される．

このようにして，「最も小さい暫定距離を持つノードを確定し，その周囲を更新する」という操作を繰り返すことで，図のように確定ノードが徐々に広がっていく（太線で示された経路）．最終的には，すべてのノードについて最短距離と対応する経路が求められる．

Dijkstra 法が正しく機能する理由は，リンクコストが非負であるとき，いったん最小の暫定距離を持つノードは，それ以上改善されることがないからである．この単純な性質に基づいて，多項式時間で最短経路を求めることができる．

4.3.5　巡回セールスマン問題

(a)　問題設定

巡回セールスマン問題，すなわち TravelingSale smanProble m（TSP）は，組合せ最適化における代表的な問題の一つである．複数の都市を一度ずつ訪問して出発地点に戻る巡回路のうち，総移動距離が最も短いものを求める．

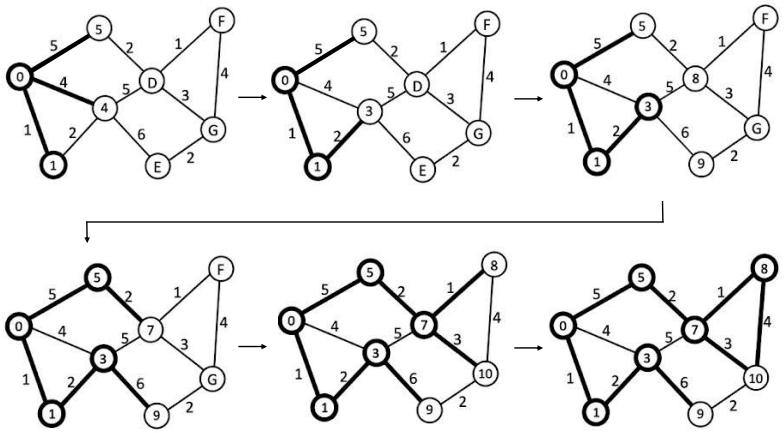


図 4.12: ダイクストラ法

(b) 定式化

TSP では，各都市をノード，都市間移動をリンクとしてグラフを構成し，都市 i から都市 j への移動コストを c_{ij} とする．リンク (i,j) を巡回路に含めるなら $x_{ij}=1$，含めないなら $x_{ij}=0$ とする．このとき目的関数は

$$\min \sum_i \sum_j c_{ij} x_{ij} \tag{4.80}$$

である．

制約として，各都市からはちょうど 1 本のリンクが出ていき，各都市にはちょうど 1 本のリンクが入ってこなければならない．したがって，

$$\sum_j x_{ij} = 1 \qquad (alli) \tag{4.81}$$

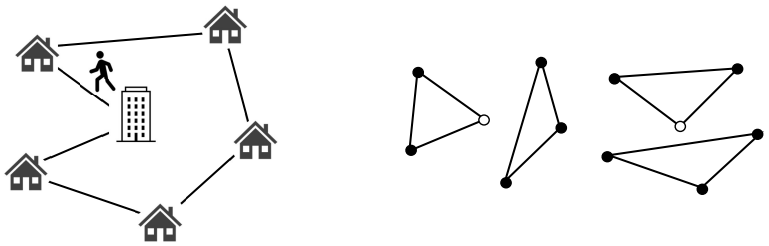
$$\sum_i x_{ij} = 1 \qquad (allj) \tag{4.82}$$

が必要である．

しかし，これだけでは問題は十分ではない．なぜなら，複数の小さな巡回路に分かれた解が許されてしまうからである．例えば，6 都市を 2 つの 3 都市サイクルに分けた解は，上の入次数・出次数制約を満たしてしまう．これを防ぐためには，部分巡回除去制約が必要となる．すなわち，任意の真部分集合 $S \subset V$ に対して，

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \tag{4.83}$$

を課す．この制約により，部分集合の中だけで閉じた巡回路が形成されることを防ぐことができる．



(a) TSP　　(b) 部分巡回路

図 4.13: TSP

(c)　TSP の難しさ

TSP が難しい理由は，訪問順序の数が爆発的に増加することにある．n 都市の巡回順序は，対称 TSP なら概ね $(n-1)!/2$ 通り，非対称 TSP なら $(n-1)!$ 通り存在する．このため，都市数が少し増えただけで全探索は不可能になる．

また，定式化の面でも，部分巡回除去制約は指数個存在する．したがって，単純にすべての制約を書き下して解くことは難しい．このことが，TSP に対して切除平面法や分枝限定法，列生成法などの高度な解法が発展した背景にある．

4.3.6　配送計画問題

(a)　問題設定

配送計画問題，すなわち Vehicle RoutingProble m（VRP）は，巡回セールスマン問題を複数車両に拡張した問題である [14]．TSP では，一台のセールスマンがすべての都市を訪問したが，VRP では，複数の車両がデポから出発し，複数の顧客を分担して訪問し，再びデポへ戻る．目的は一般に，総移動距離，総配送時間，総コストなどを最小化することである．

VRP が重要である理由は，現実の物流業務がまさにこの構造を持つからである．宅配便，食品配送，ごみ収集，保守点検など，多くの都市サービスは「複数の需要地点を複数車両で効率よく巡回する」という問題に帰着できる．

(b)　定式化

容量制約付き VRP（CapacitatedVRP;CVRP）を考える．

デポをノード 0，顧客集合を $N = \{1, 2, \ldots, n\}$ とする．ノード集合を $V = \{0\} \cup N$ とする．

- c_{ij}：ノード i から j への移動コスト
- d_i：顧客 i の需要量（$i \in N$）
- Q：車両容量

決定変数は以下の通り：

$$x_{ij} = \begin{cases} 1 & \text{車両が辺 } (i,j) \text{ を通るとき} \\ 0 & \text{それ以外} \end{cases} \qquad (i,j \in V,\ i \neq j) \tag{4.84}$$

$$u_i:\ \text{顧客 } i \text{ に到着したときの累積積載量} \qquad (i \in N) \tag{4.85}$$

目的関数：

$$\min \sum_{i \in V} \sum_{j \in V,\, i} c_{ij} x_{ij} \tag{4.86}$$

制約条件は以下の通りである：

(b).1 各顧客はちょうど一度訪問される

$$\sum_{j \in V,\, j} x_{ij} = 1 \quad (alli \in N) \tag{4.87}$$

$$\sum_{i \in V,\, i} x_{ij} = 1 \quad (allj \in N) \tag{4.88}$$

(b).2 デポに関するフロー保存（車両の出発と帰着）

$$\sum_{j \in N} x_{0j} = \sum_{i \in N} x_{i0} \tag{4.89}$$

(b).3 部分巡回の除去および容量制約（MTZ 制約）

$$u_i - u_j + Q x_{ij} \leq Q - d_j \quad (alli \in V,\, j \in N,\, i) \tag{4.90}$$

(b).4 累積荷量の範囲制約

$$d_i \leq u_i \quad (alli \in N) \tag{4.91}$$

(c) VRP と TSP の違い

TSP では，すべての都市を一つの巡回路で結ぶことが目標であった．これに対し VRP では，顧客集合を複数の車両にどのように割り当てるかという問題と，各車両の訪問順序をどう決めるかという問題が同時に現れる．

このため，VRP では経路生成だけでなく，顧客クラスタリングや車両割当ての問題も含まれる．さらに，現実の配送では，車両容量，時間窓，運転時間規制，複数デポ，積み下ろし時間など，多様な制約が加わる．その結果，VRP は TSP よりもはるかに複雑であり，実務上も理論上も重要な研究対象となっている．

(d)　VRP の代表的拡張

前節では，基本的な配送計画問題（VRP）について説明した．基本 VRP では，複数の顧客に対して，容量制約を持つ複数の車両をどのように運行させれば総移動距離や総配送時間を最小化できるかを考えた．しかし，現実の都市物流では，この基本形だけでは十分ではない．実際の配送では，顧客が受け取り可能な時間帯が決まっていたり，配送だけでなく集荷も同時に行ったり，複数の営業所や拠点から車両が出発する．

このように，現実の配送計画問題をより忠実に表現するためには，基本 VRP にさまざまな追加条件や構造を組み込む必要がある．本節では，このような VRP の派生問題を扱う．これらの問題は，単に制約が一つ増えただけの問題ではない．多くの場合，新しい制約が加わることで，問題の構造そのものが変化し，解法も大きく難しくなる．例えば，時間窓制約が加わると，もはや距離だけを短くすればよいわけではなく，訪問順序と待機時間を同時に考えなければならない．また，多デポの場合には，どの顧客をどの拠点が担当するかという割当ても同時に考える必要がある．

時間窓付き配送計画問題

現実の配送では，顧客がいつでも荷物を受け取れるとは限らない．例えば，商業施設への納品は営業開始前の限られた時間帯にしか認められないことがあるし，個人宅配送でも受取希望時間帯が指定されることが多い．このように，各顧客に対して訪問可能な時間帯が与えられている VRP を，時間窓付き配送計画問題（Vehicle RoutingProble mwithTimeWindows:VRPTW）という [13].

VRPTW では，各顧客 i に対して，訪問開始時刻が

$$a_i \leq t_i \leq b_i \tag{4.92}$$

を満たさなければならない．ここで，a_i は最も早い訪問可能時刻，b_i は最も遅い訪問可能時刻である．また，顧客でのサービスには一定の時間 s_i がかかるため，ある顧客を訪問した後は，移動時間に加えてサービス時間も次の顧客の到着時刻に影響する．したがって，VRPTW では，単にどの順番で顧客を回るかだけでなく，「何時にどこへ到着するか」という時間軸上の整合性を常に考えなければならない．

この問題が基本 VRP より難しいのは，経路の良し悪しが距離だけでは決まらないからである．例えば，移動距離が短い順に顧客を訪問するルートでも，ある顧客の時間窓に間に合わなければ実行不可能である．逆に，少し遠回りをしてでも，時間窓に適合する順序を選ばなければならない場合もある．また，ある顧客に早く着きすぎた場合には待機が発生することがあり，その待機時間も全体の効率に影響する．

都市物流において VRPTW が重要なのは，現実の配送業務の多くが時間制約の下で行われているからである．特に都心部では，交通混雑の時間変動も大きいため，所要時間自体が時間帯によって変わる場合がある．このような場合，単純な時間窓付き問題よりさらに複雑な時間依存 VRPTW が必要になることもある．したがって，時間窓付き VRP は，都市物流を実務に近い形で理解するための中心的なモデルの一つである．

集荷配送計画問題

物流では，物を届けるだけでなく，引き取る作業も同時に行われることがある．例えば，宅配便では荷物の配達と同時に返品や発送荷物の集荷を行うことがあるし，商業物流では納品後に空容器

や廃棄物を回収することもある．このように，配送と集荷の両方を同時に扱う問題を，集荷配送計画問題という [17].

この問題では，各顧客に対して，「届ける量」と「引き取る量」の両方が存在する．そのため，車両の積載量は経路の進行に伴って増減する．配送だけであれば，車両は出発時に荷物を多く積み，訪問を重ねるごとに荷量が減っていく．しかし，集荷があると，ある地点で荷物が減ったあと，別の地点でまた荷物が増える可能性がある．したがって，容量制約の確認は経路全体を通じて逐次的に行わなければならず，基本 VRP よりも複雑になる．

さらに，集荷と配送の間に優先関係がある場合もある．例えば，ある荷物をある地点で集荷したあと，別の地点へ配送しなければならない場合には，集荷地点の訪問が配送地点より先でなければならない．このような先行関係が加わると，問題は pickupanddeliveryproble m として知られる別の重要なクラスになる．この場合，顧客ペア間の順序制約を守りながらルートを設計しなければならない.

都市物流では，回収物流やリサイクル物流，宅配便の再利用容器回収など，多くの場面で集荷配送計画問題が現れる．したがって，配送を一方向の流れとしてだけでなく，都市内で物が循環するプロセスとして捉えるうえで，この問題は重要である.

多デポ配送計画問題

基本 VRP では，すべての車両が一つのデポから出発し，そこへ戻ることを仮定することが多い．しかし，現実の物流事業者や公共交通事業者は，複数の営業所や車庫，配送拠点を持っていることが一般的である．このように，複数のデポが存在する状況を扱うのが，多デポ配送計画問題（Multi-DepotVRP）である [5].

多デポ VRP では，どの顧客をどのデポから出る車両が担当するかを決める必要がある．したがって，問題は単なるルーティングではなく，「顧客割当て」と「ルート設計」が同時に現れる二重の構造を持つ．ある顧客にとって最も近いデポが常に最適とは限らない．近くてもそのデポの車両数が不足していたり，他の顧客との組み合わせが悪かったりすることがあるからである．そのため，多デポ問題では，空間的な近さだけでなく，全体としてのルート効率や負荷バランスも考慮しなければならない.

この問題は，都市物流において重要である．例えば，大都市圏では複数の配送センターや営業所が存在し，どのエリアをどの拠点が担当するかは，サービス水準や運行コストに大きく影響する．また，近年のマイクロデポや都市内小型拠点の導入により，多デポ的な構造はさらに一般的になっている.

理論的には，多デポ VRP は，基本 VRP の複雑さに加えて配置・割当ての要素を含むため，難しい問題である．施設配置問題と VRP の中間に位置する問題として理解することもでき，戦略的意思決定と運用的意思決定の接続を考えるうえで重要な題材となる.

新しいモビリティを含む配送計画問題

近年では，従来のトラックやバンだけでなく，ドローン，自動配送ロボット，小型 EV などの新しい配送手段を組み合わせた問題が注目されている．このような問題では，一種類の車両だけではなく，複数種類の移動主体が異なる能力や制約を持ちながら協調して配送を行う.

例えば，トラックが幹線的な移動を担い，最終的な顧客訪問は配送ロボットが歩道上で行うとい

うモデルがある．この場合，トラックの停止位置，ロボットの発進地点，顧客割当てを同時に決めなければならない [9]．また，ドローンを用いる場合には，飛行可能距離，積載重量，バッテリー容量などが新たな制約となる [18]．このように，新しいモビリティを含む VRP では，単なるルート順序だけでなく，異種車両間の同期や役割分担が中心的な課題になる．

クイズ 4：公共交通の相乗り

ある都市で，デマンド応答型交通（Demand Responsive Transport; DRT）を運行する問題を考える．1 台の車両がデポを出発し，複数の利用者の乗車地点（pickup）と降車地点（delivery）を巡回して，最後にデポへ戻る．各利用者については，対応する乗車地点を先に訪問し，その後に対応する降車地点を訪問しなければならない．また，車両には定員があり，どの時点でも乗車中の人数がその定員を超えてはならない．

この問題を VRP の拡張として定式化したい．ある学生は，「各ノードをちょうど 1 回訪問する制約」と「各ノードでのフロー保存制約」だけを入れれば，乗合型の運行計画として十分であると主張した．別の学生は，「それだけでは，相乗り交通として不適切な解が許されるので，pickup と delivery の先行関係や車内人数の推移を表す追加制約が必要である」と主張した．

この問題に関する次の記述のうち，最も適切なものはどれか．

(ア) 各ノードを 1 回ずつ訪問し，フロー保存が成り立っていれば，車両は必ず利用者を正しい順序で輸送する．したがって，pickup と delivery の先行関係を明示する制約は不要であり，容量制約も独立に考える必要はない．

(イ) 各ノードを 1 回ずつ訪問し，フロー保存が成り立っていても，ある利用者の delivery が pickup より先に訪問されるような不適切な解や，車内人数が定員を超える解が排除されるとは限らない．したがって，公共交通の相乗りを表現するには，通常の巡回制約に加えて，pickup と delivery の先行関係，および乗車人数の推移に基づく容量制約を導入する必要がある．

(ウ) pickup と delivery の先行関係だけを入れれば，利用者は必ず正しく輸送されるので，車内人数の推移を表す容量制約は不要である．なぜなら，各利用者は最終的に必ず 1 回だけ乗車し 1 回だけ降車するため，乗車人数が定員を超えることは構造上起こりえないからである．

→ 正解・解説は節末を参照

4.3.7　厳密解法と近似解法

組合せ最適化に対する解法は，大きく厳密解法と近似解法に分けられる．厳密解法とは，最適解を必ず見つける方法であり，代表例として分枝限定法，切除平面法などがある．これらの方法は理論的に最適性保証を持つが，大規模問題では計算時間が膨大になることがある．一方，近似解法や

ヒューリスティクスは，必ずしも最適解を保証しない代わりに，短時間で良い解を求めることを目的とする．

厳密解法は，「この解が最適である」と証明できる．例えば，分枝限定法では，探索木（問題の解空間を分岐によって体系的に列挙し，探索の過程を木構造として表したもの）を構築しながら上下界を用いて不要な枝を剪定し，最終的に残った解が最適であることを保証する．切除平面法では，線形緩和に有効なカットを追加することで，整数最適解を厳密に特定する．このような方法は，理論研究やベンチマーク作成，政策評価において極めて重要である．なぜなら，最適値が分かっていれば，ほかの手法がどれだけ良いかを客観的に比較できるからである．

近似解法という言葉は広い意味で使われるが，教科書では少なくとも二つに分けて考えると分かりやすい．一つは，理論的な近似保証を持つ近似アルゴリズムである．これは「最適値の何倍以内の解を必ず返す」といった性能保証を持つ方法であり，計算量理論との関係が深い．もう一つは，より実務的な意味でのヒューリスティクスである．ヒューリスティクスは，一般には理論的保証を持たないが，経験的に高品質な解を短時間で与えることを目的とする．

クイズ 4 正解・解説

正解：(イ)

本問の本質は，公共交通の相乗りを VRP として表現する際に，通常の巡回路制約だけでは不十分であることを理解しているかどうかである．

まず，「各ノードを 1 回ずつ訪問する制約」と「フロー保存制約」は，車両がネットワーク上を一つの連続したルートとして巡回することを表すためには重要である．しかし，それだけでは，ある利用者について delivery が pickup より先に訪問されるような解を排除できない．このような解は，数理的には巡回路であっても，交通サービスとしては意味を持たない．したがって，各利用者について pickup を delivery より先に訪問するという先行関係制約が必要である．

さらに，相乗り交通では，複数の利用者を同時に車内に乗せる可能性があるため，どの時点で何人が乗車中であるかを管理しなければならない．先行関係制約だけでは，「誰が先に乗って後で降りるか」という順序は表現できても，ある時点で車内に何人いるかは分からない．そのため，乗車人数の累積を表す変数や制約を追加し，常に定員以下であることを保証する必要がある．

(ア) は，巡回制約だけで輸送の意味的整合性まで保証できるとしている点で誤りである．巡回路としては成立していても，pickup より前に delivery を訪問するような不適切な解は別途排除しなければならない．

(ウ) は，先行関係があれば容量制約は不要であるとしている点で誤りである．実際には，複数の利用者を続けて pickup した後で順次 delivery するルートでは，pickup と delivery の順序関係は守っていても，途中時点で車内人数が定員を超える可能性がある．したがって，先行関係制約と容量制約は役割が異なり，どちらも必要である．

このように，公共交通の相乗り型 VRP では，単に巡回順序を決めるだけでは不十分であり，

「誰を先に乗せ，誰を後に降ろし，その途中で何人が同時に車内にいるか」まで一貫して表現する必要がある．

4.4　厳密解法

4.4.1　厳密解法の考え方

厳密解法とは，有限時間のうちに最適解であることを保証しながら問題を解く方法である．ここで重要なのは，単に「良い解」を出すのではなく，「これ以上よい解は存在しない」と理論的に示す点にある．ヒューリスティクスでは，得られた解がどれほど良いかを経験的にしか評価できない場合が多いのに対し，厳密解法では，解の最適性を数理的に証明できる．この違いは大きい．

厳密解法は，小規模問題に対する厳密解は，ヒューリスティクスの性能評価に不可欠である．新しいヒューリスティクスを提案したとしても，最適解との比較ができなければ，その有効性を客観的に示すことは難しい．また，厳密解法の内部で用いられる緩和，双対性，分解，列生成といった考え方は，そのままヒューリスティクスや近似解法の設計にも応用される．したがって，厳密解法は単独で重要なだけでなく，最適化全体の理論的基盤でもある．

厳密解法を理解するうえで最も重要なのは，「候補解を全部調べる」のではなく，「調べる必要のない部分を理論的に除外する」という発想である．もし可能な解を本当にすべて列挙して比較するならば，たいていの組合せ最適化問題はすぐに計算不能になる．したがって，厳密解法では，問題の構造を使って，探索空間を賢く絞り込む必要がある．この絞り込みを支えているのが，下界と上界，緩和問題，双対性，カット，分解といった概念である．

例えば，最小化問題では，ある実行可能解が見つかれば，その目的関数値は上界を与える．一方，制約を緩めた問題を解けば，真の最適値以下の値が得られることが多く，これが下界になる．もしある部分問題について，その下界がすでに得られている上界以上であれば，その部分問題の中にはより良い解が存在しないことが分かる．したがって，その部分問題はそれ以上調べる必要がない．このような上界と下界の比較によって探索空間を削減することが，厳密解法の中心的な考え方である．

本章では，代表的な手法として分枝限定法，切除平面法，列生成法，ベンダーズ分解等を学ぶ．いずれも「難しい問題を，緩和・分解・追加制約などによって扱いやすくし，そのうえで最適性を保証する」という共通の思想に基づいている．さらに最後には，厳密解法の意義と限界を整理し，次章のヒューリスティクスとの関係を明確にする．

4.4.2　分枝限定法

分枝限定法は，問題全体を部分問題に分けて探索しながら，各部分問題に対して「これ以上調べる価値があるかどうか」を理論的に判断し，不要な部分を剪定していく [6]．英語では BranchandBound と呼ばれ，現代の多くの整数計画ソルバーの基礎となっている．

分枝限定法を最も分かりやすく説明するために，0–1 整数計画問題を考える．例えば，

$$\text{minimize} \quad c^T x \tag{4.93}$$

$$\text{subject to} \quad Ax \leq b \tag{4.94}$$

$$x_i \in \{0, 1\} \qquad (i = 1, \ldots, n) \tag{4.95}$$

という問題があったとする．この問題をそのまま解くのは難しいが，まず整数制約

$$x_i \in \{0, 1\} \tag{4.96}$$

を一時的に外して，

$$0 \leq x_i \leq 1 \tag{4.97}$$

と緩めた線形緩和問題を解くことができる．この緩和問題は線形計画問題なので，比較的容易に解ける．

もし，緩和問題の解が整数解であれば，それは元の整数計画問題に対しても実行可能である．しかも，緩和問題は元の問題より制約が弱いので，その最適値は元の問題の最適値以下である．したがって，緩和問題の解が整数解であれば，それは元の問題の最適解である．これが分枝限定法の最も単純な成功例である．

しかし，多くの場合，緩和問題の解は整数にならない．例えば，ある変数 x_k が

$$x_k = 0.37 \tag{4.98}$$

のような分数値をとることがある．このとき，元の問題では x_k は 0 か 1 のどちらかでなければならないので，このままでは解として使えない．そこで分枝を行う．すなわち，元の問題を，

$$x_k = 0 \tag{4.99}$$

を追加した部分問題と，

$$x_k = 1 \tag{4.100}$$

を追加した部分問題の二つに分ける．これにより，元の探索空間は二つの部分空間に分割される．

各部分問題に対して再び線形緩和を解き，同じことを繰り返していくと，探索木が構築される．この木の各ノードは「ある変数の値がいくつか固定された部分問題」に対応している．ここで重要なのが bound，すなわち下界または上界の利用である．最小化問題では，各部分問題の緩和問題の最適値は，その部分問題の真の最適値に対する下界を与える．一方，探索中に整数実行可能解が見つかれば，その目的関数値は元の問題に対する上界を与える．この上界を暫定最良解，あるいは incumbent の値と呼ぶことがある．

もしある部分問題の下界が，すでに得られている上界以上であれば，その部分問題の中にはより良い解が存在しないことが分かる．したがって，その部分問題はそれ以上探索する必要がない．これを枝刈り，あるいは pruning という．また，部分問題の緩和解が非実行可能ならば，当然その枝も打ち切ってよい．さらに，緩和解が整数解ならば，その枝では最適解候補が得られたので，それ以上の分枝は不要である．このように，分枝限定法では，

1. 緩和問題を解く，
2. 分数解なら分枝する，
3. 下界と上界を比較して不要な枝を切る，

という操作を繰り返していく．「探索空間を木構造で分割し，下界と上界で不要部分を除外する」と理解することが重要である．

分枝限定法の性能を左右する最大の要因は，下界の強さである．緩和問題から得られる下界が弱ければ，多くの枝を切れず，探索木が巨大になる．逆に，下界が強ければ，探索初期の段階で多くの枝を除外できる．そのため，実務上のソルバーでは，より強い緩和を構成したり，良い上界を早く見つけるためのヒューリスティクスを内部で併用したりすることが多い．

また，どの変数で分枝するかも重要である．分枝変数の選び方によって探索木の大きさは大きく変わる．例えば，分数値が 0.5 に近い変数を優先して分枝する方法や，目的関数値への影響が大きい変数を優先する方法などが考えられる．このような分枝規則の工夫は，アルゴリズムの理論というより実装上の工夫に近いが，実際の計算性能には大きく影響する．

都市交通や物流の問題では，分枝限定法は，小規模から中規模の VRP，施設配置問題，時刻表設計問題などを厳密に解く際にしばしば用いられる．また，大規模問題では単独で解くことが難しくても，後に述べる切除平面法や列生成法と組み合わせることで，強力な解法の一部として機能する．したがって，分枝限定法は「単独の解法」というより，「現代的な厳密解法の骨格」として理解するのが適切である．

4.4.3　切除平面法

切除平面法は，整数計画問題に対するもう一つの代表的な厳密解法である [8]．分枝限定法が探索空間を木構造に分割することで整数性を実現しようとするのに対し，切除平面法は，緩和問題に新しい制約を追加することによって，分数解を排除しながら整数解に近づいていく方法である．

整数計画問題の難しさの一因は，線形緩和を解くと分数解が得られることにある．例えば，0–1 変数を持つ問題でも，緩和後には

$$x_1 = 0.5, _2 = 0.5 \tag{4.101}$$

のような解が現れることがある．この分数解は元の問題では許されないが，線形緩和では実行可能であり，しばしば緩和問題の最適解になってしまう．切除平面法の基本的な考え方は，このような「望ましくない分数解」を排除する制約を追加しながら，実行可能領域を徐々に整数解の凸包に近づけていくことである．

ここで追加される制約は，現在の分数解は排除するが，元の問題のすべての整数実行可能解は残すものでなければならない．このような制約をカット，あるいは切除平面という．もし誤って整数実行可能解まで削ってしまえば，元の最適解を失ってしまうので，カットは慎重に構成しなければならない．

切除平面法の流れは，概念的には次のように理解できる．まず，整数制約を緩和した線形計画問題を解く．もし得られた解が整数解ならば，それで終了である．しかし，分数解であれば，その解を排除するカットを一つ，あるいは複数生成し，それを元の緩和問題に追加する．すると，実行可

能領域は少し狭くなり，次に解いた緩和問題では，先ほどの分数解は現れなくなる．この操作を繰り返せば，理想的には最終的に整数解へ到達する．

切除平面法の本質は，「整数計画問題を直接解く」のではなく，「線形計画問題を繰り返し解きながら整数性を段階的に回復する」ことにある．このため，線形計画ソルバーの力を活用しやすいという利点がある．一方で，どのようなカットを作るかが性能を大きく左右する．弱いカットばかり追加しても改善は遅く，逆に強いカットを適切に生成できれば，効率的に整数解へ近づけることができる．

最も古典的な切除平面としては Gomory カットがある．これは単体表の分数解から機械的に生成できるカットであり，整数計画理論の歴史において重要である．ただし，実務上は Gomory カットだけでは十分でないことも多く，問題構造に応じた特殊なカットが広く用いられている．

TSP はその代表例である．TSP の基本定式化では，入次数・出次数制約だけでは部分巡回が生じてしまうことを 4.3.5 で学んだ．このとき，部分巡回除去制約は，まさにカットとして働く．すなわち，まず緩い緩和問題を解き，もしその解が複数の部分巡回から構成されていれば，その部分巡回を排除する制約

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \tag{4.102}$$

を追加する．すると，その部分巡回は次の反復では許されなくなる．このように，TSP における部分巡回除去制約は，切除平面法の最も分かりやすい具体例である．

切除平面法が強力であるのは，探索木を深くたどらなくても，緩和そのものを強化できる点にある．分枝限定法では，分数解が出るたびに分枝して探索木を伸ばしていく必要があるが，切除平面法では，一つのカット追加によって多数の分数解を一度に排除できることがある．そのため，強いカットをうまく設計できれば，探索効率は大きく改善する．

もっとも，切除平面法だけで常に効率よく解けるわけではない．追加すべきカットを見つけること自体が難しい場合もあるし，カットを加えすぎると線形計画問題が巨大化して 1 回の反復の計算が重くなることもある．また，分数解を排除しても，次に現れる解がまた別の分数解であることも多い．そのため，実際には切除平面法は単独で使われるより，分枝限定法と組み合わせた分枝カット法として用いられることが多い．現代の整数計画ソルバーの多くは，この分枝カット法を内部で実行している．

都市交通・物流の分野では，TSP や VRP の部分巡回除去制約，容量制約に基づくカット，不等式制約の強化などが典型的な応用例である．特に，配送計画問題では緩和が弱くなりやすいため，強いカットを設計することが解法性能の鍵になる．

4.4.4 列生成法

分枝限定法や切除平面法が，ある程度明示的に書き下せる定式化を前提にしているのに対し，列生成法は「定式化そのものが巨大すぎて，最初から全部の変数を用意できない」という状況に対処する方法である [20]．特に，配送計画問題の集合分割定式化のように，考えられるルート数が指数的に多い問題に対して強力である．

列生成法の核心は，「最初からすべての列を用意しない」という発想にある．ここでいう列とは，

線形計画の係数行列の列，すなわち各変数に対応する情報である．通常，線形計画や整数計画を解くときには，最初からすべての変数が明示されていることを前提とする．しかし，VRP では「可能なルート」を変数として持つ集合分割定式化を考えると，ルートの総数は指数的に増大し，現実的には列挙できない．このとき，少数の有望そうな列だけで始めて，必要になった列を逐次追加していく，という考え方が列生成法である．

代表的な例として，集合分割型の主問題を考える．各ルート r に対して変数 λ_r を導入し，そのルートを採用するなら $\lambda_r = 1$，採用しないなら $\lambda_r = 0$ とする．また，ルート r のコストを c_r，顧客 i をルート r が訪問するかどうかを a_{ir} とすると，本来の問題は概念的には

$$\text{minimize} \quad \sum_{r \in \mathcal{R}} c_r \lambda_r \tag{4.103}$$

$$\text{subject to} \quad \sum_{r \in \mathcal{R}} a_{ir} \lambda_r = 1 \qquad (\forall i) \tag{4.104}$$

$$\lambda_r \in \{0, 1\} \tag{4.105}$$

のような整数計画問題として書ける．ここで $\mathcal{R}$ は全ルート集合である．しかし，この $\mathcal{R}$ は通常きわめて巨大であり，すべてを最初から扱うことはできない．

そこで列生成法では，まずこの整数計画問題の線形緩和を考える．すなわち，

$$\text{minimize} \quad \sum_{r \in \mathcal{R}} c_r \lambda_r \tag{4.106}$$

$$\text{subject to} \quad \sum_{r \in \mathcal{R}} a_{ir} \lambda_r = 1 \qquad (\forall i) \tag{4.107}$$

$$\lambda_r \geq 0 \tag{4.108}$$

という形の問題を対象とする．列生成法が直接解いているのは，通常この線形緩和である．

実際の計算では，まず一部の列だけを持つ制限付き主問題（restricted master problem）を解く．この問題は小規模なので，通常の線形計画として解くことができる．その結果，双対変数が得られる．次に，この双対変数を用いて，「まだ導入していない列の中に，現在の目的関数値を改善できるものがあるか」を調べる．この探索問題を価格付け問題（pricing problem）という．もし改善可能な列が見つかれば，その列を主問題に追加して再び解く．改善可能な列が見つからなくなれば，現在の制限付き主問題の解は，元の巨大な線形緩和問題に対して最適である．

列生成法の数学的核心は reduced cost，すなわち被約費用にある．主問題の双対解を用いると，各未導入列が目的関数を改善するかどうかを，被約費用の符号によって判定できる．最小化問題では，被約費用が負の列があれば，その列を導入することで目的関数値を改善できる．逆に，すべての未導入列の被約費用が非負であれば，現在の解は線形緩和問題に対して最適である．この意味で，双対性は列生成法の中心的役割を果たしている．

列生成法の利点は，巨大な変数集合を最初から扱わずに済むことである．これにより，本来なら定式化すら困難な問題を，必要な列だけを生成することで解けるようになる．一方で，列生成法単体で得られる解は通常，線形緩和の最適解であって，整数最適解とは限らない．このため，厳密解を得るには，列生成法を分枝限定法と組み合わせた branch-and-price が必要になる．これは，分枝

によって整数性を実現しつつ，各ノードで列生成を行う方法であり，VRP の厳密解法として重要である．

都市交通・物流の分野で列生成法が重要なのは，ルートや運行パターンの数が指数的に大きくなる問題が多いからである．配送計画問題だけでなく，乗務員スケジューリング，車両運用計画，バスのダイヤ設計などでも，列生成法の発想が広く使われている．したがって，列生成法は単なる線形計画の一解法というよりも，「変数が多すぎる最適化問題をどのように扱うか」という観点から重要な役割を果たす方法であり，整数計画問題に対しても，その線形緩和を通じて強力な解法を与える枠組みなのである．

4.4.5 ベンダーズ分解

ベンダーズ分解は，大規模混合整数計画問題に対して用いられる代表的な分解法である [19]．列生成法が「変数が多すぎる」問題に対する方法であるのに対し，ベンダーズ分解は「問題の構造が二段階的・階層的である」場合に有効である．特に，ある一群の変数を固定すると，残りの問題がかなり容易になるような構造を持つとき，ベンダーズ分解は強力である．

典型的な状況として，変数を二つのグループに分けられる問題を考える．例えば，施設をどこに開設するかという離散的な設計変数と，その施設配置のもとでどのように物流を運用するかという連続または組合せ的な運用変数があるとする．このような問題では，設計変数を決めると，運用問題は比較的明確に解けることが多い．ベンダーズ分解は，この構造を利用して，元の大問題を主問題と部分問題に分割して解く．

概念的には，まず設計変数だけを扱う主問題を考える．主問題は最初はかなり粗い近似であり，運用変数の影響を完全には表していない．主問題を解くと，ある設計変数の候補解が得られる．次に，その候補解を固定したもとで，運用に関する部分問題を解く．もし部分問題が実行可能であれば，その目的値や双対情報を用いて，「この設計解のもとでは運用コストが少なくともこれだけかかる」という情報を主問題に返す．これがベンダー最適性カットである．一方，もし部分問題が実行不可能であれば，「この設計解ではそもそも運用が成立しない」という情報を主問題に返す．これがベンダー実行可能性カットである．

このようにして，主問題は反復的に洗練される．最初は粗い近似しか持っていなかった主問題も，カットを追加するたびに，部分問題の構造を少しずつ学習していく．十分多くのカットが加わると，主問題は元の問題を正確に表すようになり，最適解に到達する．この点で，ベンダーズ分解は「部分問題から得られる情報を切断面として主問題に戻す方法」と理解することができる．

ベンダーズ分解の魅力は，巨大な問題を無理に一つのまま解こうとしないことである．特に，設計変数が少数の整数変数であり，運用部分が線形計画やネットワーク問題として効率的に解ける場合には，うまく機能する．また，複数シナリオを持つ確率計画問題やロバスト最適化でも，各シナリオごとの部分問題を独立に解けるため，ベンダーズ分解の考え方が自然に現れる．

具体例として，施設配置と配送を同時に考える問題を挙げると分かりやすい．例えば，いくつかの候補地点の中から物流拠点を開設し，その後，選ばれた拠点のもとで顧客配送を行うとする．このとき，拠点の開設決定は離散的で長期的な意思決定であり，配送計画はその結果として定まる短期的な運用である．元の問題を一括で解くと極めて大規模になるが，拠点配置を主問題，配送計画

を部分問題として分けると，問題構造が見やすくなる．都市交通では，道路整備と交通配分，充電インフラ配置と EV 運用，ハブ配置と配送ルート設計など，多くの場面でこの階層構造が現れる．

もっとも，ベンダーズ分解も万能ではない．主問題に追加されるカットが弱いと収束が遅くなるし，部分問題が整数計画である場合には古典的ベンダーズ分解をそのまま適用できないこともある．そのため，実際には強化ベンダーカット，ロジックベンダーズ分解，組合せベンダーズ分解など，多くの発展形が研究されている．それでもなお，ベンダーズ分解の基本思想，すなわち「上位の意思決定と下位の運用を切り分け，部分問題からの情報で主問題を洗練する」という発想は，都市交通・物流の大規模最適化を理解するうえで重要である．

4.4.6　分枝カット法

切除平面法と分枝限定法は，それぞれ単独でも重要な厳密解法であるが，現代の整数計画ソルバーでは，この二つを組み合わせた分枝カット法（branch-and-cut）が標準的な解法枠組みとなっている [10]．分枝限定法は探索木を構築しながら整数性を実現する方法であり，切除平面法は緩和問題を強化して分数解を排除する方法であった．分枝カット法は，この二つの長所を同時に活用することで，より効率的に厳密解を求めようとする方法である．

分枝限定法だけを考えると，各ノードで解く緩和問題が弱い場合には，下界が十分に強くならず，多くの枝を探索しなければならない．その結果，探索木が巨大になり，計算時間が急増する．一方，切除平面法だけを考えると，緩和問題をどれだけ強化しても，最終的に整数解に到達するまでに多くの反復が必要になることがある．また，カットだけでは整数性を完全に実現するのが難しい場合も多い．そこで，探索木の各ノードで必要に応じてカットを追加し，緩和問題を強化しながら分枝を進める，という考え方が自然に生まれる．これが分枝カット法である．

分枝カット法の基本的な流れは次のように理解できる．まず，元の整数計画問題の線形緩和を解く．もし解が整数ならば終了である．もし分数解ならば，その解を排除する有効なカットを探す．適切なカットが見つかれば，それを追加して再び緩和問題を解く．この段階で緩和が十分に強化されれば，整数解に達したり，少なくとも下界が大きく改善されたりする．それでもなお分数解が残る場合には，分枝を行って部分問題を作る．そして，その各部分問題に対しても，再び緩和を解き，必要ならカットを追加し，さらに必要なら分枝する．このように，各ノードで「緩和強化」と「探索分割」を組み合わせながら計算を進める．

この方法の利点は明確である．カットを入れることで緩和が強くなれば，下界が改善され，多くの枝を早い段階で刈り取ることができる．一方で，カットだけで十分に整数性が回復しない場合には，分枝によって探索空間を確実に細分化できる．したがって，分枝カット法は，切除平面法の「緩和強化能力」と分枝限定法の「整数性実現能力」を併せ持つ方法といえる．

ただし，分枝カット法の性能は，どのカットをどのタイミングで追加するかに大きく依存する．弱いカットばかり追加しても効果は小さいし，逆にカットを追加しすぎると，各ノードで解く線形計画問題が巨大になってしまう．また，探索木のすべてのノードで同じように多くのカットを入れるのがよいとは限らず，どのノードでどの種類のカットを優先するかには工夫が必要である．このため，現代のソルバーでは，カット生成，分枝規則，ノード選択，ヒューリスティクスなどが複雑に組み合わされている．

TSP や VRP は，分枝カット法の典型的な応用対象である．例えば TSP では，緩和問題を解いたあとに部分巡回が見つかれば，その部分巡回除去制約をカットとして追加する．それでも分数解が残るなら分枝する．VRP でも同様に，容量制約や部分巡回を強化するための不等式をカットとして用いることがある．このように，問題構造に特有の強いカットを組み込むことが，分枝カット法の性能向上に直結する．

したがって，分枝カット法は「分枝限定法＋切除平面法」ではなく，現代的な厳密解法の中核的な枠組みであると理解すべきである．整数計画ソルバーが高性能であるのは，まさにこの枠組みの中で，多数の理論的工夫と計算技術が統合されているからである．都市交通・物流の研究においても，厳密解法で実験を行う場合，多くはこの分枝カット法の恩恵を間接的に受けている．

4.4.7 Branch-and-Price

列生成法は，変数の数が指数的に多い問題に対して有効であった．しかし，列生成法だけで解けるのは通常，線形緩和問題までである．配送計画問題の集合分割定式化のように，本来は整数性が必要な問題では，列生成だけでは得られた解が分数になることが多い．このとき，整数性を回復するために分枝限定法と列生成法を組み合わせた方法が必要になる．これが branch-and-price である [3].

branch-and-price は，分枝限定法の探索木の各ノードにおいて，単純な線形緩和ではなく，列生成によって解かれる制限付き主問題を扱う方法である．言い換えれば，「変数が多すぎて全部は持てない整数計画問題」に対して，分枝しながら必要な列だけをその都度生成していく厳密解法である．この方法は，VRP，乗務員スケジューリング，車両割当て問題などで強力である．

その考え方を理解するために，VRP の集合分割定式化をもう一度思い出そう．各実行可能ルート r を一つの列とみなし，そのルートを採用するかどうかを変数 λ_r で表すと，

$$\text{minimize} \quad \sum_{r \in \mathcal{R}} c_r \lambda_r \tag{4.109}$$

$$\text{subject to} \quad \sum_{r \in \mathcal{R}} a_{ir} \lambda_r = 1 \qquad (alli) \tag{4.110}$$

$$\lambda_r \in \{0, 1\} \tag{4.111}$$

のような整数計画問題が得られる．ここで $\mathcal{R}$ はすべての実行可能ルート集合である．しかし，$\mathcal{R}$ は膨大であり，最初からすべての列を列挙することはできない．そこで，まず少数の列だけで主問題の線形緩和を解き，双対情報に基づいて新しい有望列を価格付け問題から生成する．これが列生成法であった．

ただし，ここで得られるのは通常，λ_r が分数値をとる線形緩和解である．これは「ルートを半分使う」といった現実には意味のない解であり，そのままでは使えない．そこで，分枝限定法を導入する．ただし，ここで単純に元の変数 λ_r に対して分枝すると，列生成との相性が悪くなることがある．なぜなら，新しく生成される列が，分枝によって課された条件を破ってしまう可能性があるからである．

このため，branch-and-price では「何に対して分枝するか」が重要な設計課題となる．典型的に

は，ルート変数そのものではなく，元の問題に意味を持つ構造に対して分枝する．例えば VRP では，「顧客 i と顧客 j が同じルートで連続して現れるかどうか」「ある顧客対の接続を許すか禁止するか」といった形で分枝することがある．このように分枝すると，価格付け問題の中にもその条件を組み込むことができ，新しく生成される列も分枝条件を守るように制御できる．

branch-and-price の流れを概念的に述べると，まず根ノードで列生成を行い，線形緩和の最適解を求める．もしその解が整数ならば終了である．もし分数解ならば，適切な分枝規則に従って子ノードを作る．各子ノードでは，そのノードに固有の分枝制約を満たすように価格付け問題を修正し，再び列生成によって緩和問題を解く．この操作を探索木全体にわたって繰り返し，最終的に最適な整数解を見つける．

この方法の利点は，巨大な変数集合を最初から扱わずに済むことと，整数性も保証できることの両方を実現できる点にある．もし branch-and-price がなければ，VRP のような大規模ルート選択問題を集合分割定式化で厳密に解くのはほとんど不可能である．一方で，難しさも大きい．各ノードで列生成を行う必要があるため，1 ノードあたりの計算負荷が高い．また，価格付け問題自体が資源制約付き最短経路問題などの難しい問題になることが多い．さらに，分枝規則の設計を誤ると，列生成との整合が崩れ，計算が不安定になる．

それでもなお，branch-and-price は VRP 厳密解法の中核である．時間窓付き VRP や多デポ VRP など，複雑なルート制約を持つ問題では，ルートを列として扱う集合分割定式化が強い緩和を与えるため，branch-and-price の有効性が高い．この意味で，branch-and-price は「巨大なルート集合を持つ整数計画問題を厳密に解くための代表的枠組み」として理解するのがよい．

クイズ 5：解法の選択 1

ある都市鉄道会社では，朝ラッシュ時の列車ダイヤが所与のもとで，各車両編成に対して「どの列車をどの順に担当させるか」を決める車両運用計画を最適化したいと考えている．ここで，各車両編成の運用は，接続可能な列車にしか移れないこと，車庫への入出庫条件を満たすこと，連続運用時間の上限を守ることなど，複数の条件をすべて満たさなければならない．

この問題を厳密に解くために，各変数を「1 編成の実行可能な 1 日の運用パターン」として表す集合分割型の定式化を考えたところ，大都市圏では列車本数が非常に多く，実行可能な運用パターン数が膨大になり，**線形緩和問題でさえ，最初からすべての変数を明示的に持つことができない**ことが分かった．一方で，必要な制約は最初から明示できており，問題の難しさは制約の複雑さよりも，**変数数の爆発**にあると考えられる．

この状況に対して，次にとるべき行動として最も適切なものはどれか．

(ア) 分数解を避けることが重要なので，まずは現在の緩和問題に対して，分数解を排除する追加制約をできるだけ多く入れる方向を優先して検討する．

(イ) 問題の本質が「実行可能な運用パターンを表す変数が多すぎて，線形緩和ですら全列を一度に扱えないこと」にあると捉え，一部の変数だけで始めて問題を解き，その結

果を使って有望な変数を後から逐次追加する方向を検討する．

(ウ) 変数数の多さは一旦保留し，まずは意思決定部分と評価部分に機械的に分けて，前者の候補解を作ってから後者で妥当性を確認する枠組みを第一候補として検討する．

→ 正解・解説は節末を参照

クイズ 6：解法の選択 2

ある都市では，新たな LRT 停留所の設置計画を検討している．市はまず，予算制約のもとで「どの候補地点に停留所を設置するか」を決めたい．そのうえで，与えられた停留所配置のもとで，各居住地区の利用者をどの停留所へ割り当てるかを定め，総アクセス時間や停留所混雑の大きさを評価したいと考えている．
この問題を一つの大きな混合整数計画としてまとめて解くこともできるが，候補地点数や地区数が増えるにつれて計算が急速に重くなることが分かってきた．ただし，ここで重要なのは，**どの停留所を設置するかという離散的な設計判断を固定してしまえば，その下での利用者割当てと評価は比較的解きやすい問題として扱える**という点である．つまり，この問題の難しさは，変数数が多すぎることよりも，**上位の設計問題と下位の運用・評価問題が強く結びついている**ことにあると考えられる．
この状況に対して，次にとるべき行動として最も適切なものはどれか．

(ア) 候補地点の組合せ数が多い以上，各停留所配置パターンを一つの変数として表し，不足する変数だけを後から逐次追加する枠組みを第一候補として検討する．

(イ) 設計変数を固定すると下位の利用者割当て問題が解きやすくなるという構造に注目し，上位の設計判断を扱う問題と，その結果を評価する問題に分けて，後者から得られる情報で前者を少しずつ洗練する方向を検討する．

(ウ) まずは分数解を減らすことが重要なので，問題構造よりも先に，緩和問題に追加制約を入れて実行可能領域を狭める方向を中心に検討する．

→ 正解・解説は節末を参照

4.4.8　厳密解法の意義と限界

ここまで見てきたように，厳密解法には多様な方法がある．分枝限定法は探索空間を木構造で分割しながら上下界で不要部分を除外する方法であり，切除平面法は緩和問題に新しい制約を追加して整数性を強める方法であり，列生成法は巨大な変数集合を必要な分だけ生成する方法であり，ベンダーズ分解は階層構造を持つ問題を主問題と部分問題に分けて解く方法である．これらは見かけ上異なるが，いずれも「難しい問題を，構造を利用して扱いやすい部分に分け，しかも最適性保証を失わない」という共通思想に支えられている．

厳密解法の最大の意義は，言うまでもなく最適性保証である．得られた解が本当に最適であると分かることは，政策評価，理論研究，ベンチマーク作成において大きな価値を持つ．また，厳密解法は，単に答えを出すための方法というだけでなく，問題の構造を深く理解するための道具でもある．例えば，どの制約がボトルネックになっているのか，どの緩和が弱いのか，どの変数群が問題を難しくしているのかといったことは，厳密解法を設計・適用する過程で明らかになることが多い．

さらに重要なのは，厳密解法の考え方がヒューリスティクスや近似解法の設計にも生きていることである．良い上界を早く得るための構成法，双対価格を用いた局所探索，分解構造を利用した大域的な探索設計などは，厳密解法とヒューリスティクスの橋渡しとなる．したがって，厳密解法は「実務では使いにくい理論」ではなく，むしろ応用研究の土台である．

しかし，同時に限界も明確である．最も大きいのは計算負荷である．NP 困難問題に対しては，問題サイズが大きくなると分枝木が爆発的に大きくなったり，緩和問題自体が巨大化したりする．また，列生成法では価格付け問題が難しくなることがあり，ベンダーズ分解では収束に多くの反復が必要になることがある．現代のソルバーや計算機性能は大きく向上しているが，それでも都市交通・物流の現実的な大規模問題を常に厳密に解けるわけではない．

そのため，実務では厳密解法とヒューリスティクスを使い分けることが重要になる．小規模問題や戦略的設計問題では厳密解法が有効であり，大規模な日次運用問題ではヒューリスティクスが有効であることが多い．また，中間的な立場として，厳密解法の中にヒューリスティクスを埋め込んだり，ヒューリスティクスで得た良い解を厳密解法の上界として用いたりするハイブリッドな方法も広く使われている．

したがって，厳密解法を学ぶときには，「最適解を求める特別な方法」としてだけ見るのではなく，「最適化問題の構造を理解し，ほかの解法ともつながる中核的な考え方」として捉えることが重要である．次章では，この厳密解法と対比しながら，近傍探索やヒューリスティクスの考え方を学び，現実の大規模問題に対してどのように実用的な解法を設計するかを考えていく．

クイズ 5 正解・解説

正解：（イ）

この状況では，問題の本質的な難しさが**整数性そのものではなく，実行可能な運用パターンを表す変数が膨大で，線形緩和ですら全変数を最初から持てないこと**にあると明示されている．このようなときには，最初から全変数を列挙するのではなく，一部の変数だけで開始し，必要なものを後から追加していく発想が適切である．これは**列生成法**の基本的な考え方に対応している．

（イ）は適切である．問題の難しさを**変数数の爆発**として正しく捉えており，必要な変数だけを逐次導入する方向になっている．

（ア）は不適切である．追加制約による緩和強化は一般に重要であり，これは**切除平面法**の発想に対応する．しかし，この設問では**線形緩和問題でさえ全変数を持てない**ことが先に述べられているため，この段階で中心に置くべき論点ではない．

（ウ）も不適切である．問題を意思決定部分と評価部分に分ける見方は**ベンダーズ分解**に

つながる考え方であるが，この問題文では本質が**上位・下位の階層構造**ではなく，あくまで**変数数の爆発**にあると設定されている．したがって，まずこの方向を第一候補にするのは適切でない．

クイズ 6 正解・解説

正解：(イ)

この状況では，問題の本質的な難しさが**変数数の爆発ではなく，上位の設計判断と下位の運用・評価問題が強く結びついていること**にあると明示されている．さらに，設計変数を固定すると下位問題が比較的解きやすくなるとも書かれているため，問題を二つに分け，下位問題から得られる情報を上位問題へ返していく発想が適切である．これは**ベンダーズ分解**の基本的な考え方に対応している．

（イ）は適切である．この問題で強調されている**階層構造**を正しく捉えており，設計判断と評価問題を分けて扱う方向になっている．

（ア）は不適切である．不足する変数だけを後から追加するという考え方は**列生成法**につながるものである．しかし，この設問では難しさの中心が**変数として表すべきパターンの爆発**には置かれていない．本質は**設計と運用の階層構造**にあるため，適切でない．

（ウ）も不適切である．緩和問題に追加制約を入れて実行可能領域を狭める方向は**切除平面法**の発想に対応する．もちろん分数解や追加制約は重要になりうるが，この段階でまず着目すべきなのは，設計変数を固定したときに下位問題がかなり解きやすくなるという構造である．したがって，この方向を中心に据えるのは適切でない．

4.5　ヒューリスティクスと近傍探索

4.5.1　ヒューリスティクス

ヒューリスティクスとは，厳密な最適性保証は持たないが，経験的・構造的な工夫によって，短時間で良い解を得ることを目的とした解法の総称である．問題の構造や現象の特徴を利用しながら，探索を効率化するための体系的な考え方である．4.4 では，最適解を保証する厳密解法を学んだ．しかし，組合せ最適化問題の多くは NP 困難であり，問題規模が大きくなると，厳密解法だけで現実的な時間内に最適解を求めることが難しくなる．都市交通や物流の問題では，顧客数，車両数，時間帯，容量制約，駐車制約などが重なり，問題規模と複雑さが急速に増大する．このような状況では，「最適であること」そのものよりも，「十分に良い解を，限られた時間で得ること」が重要になる場合が多い．このとき重要になるのがヒューリスティクスである [2].

例えば，配送計画問題では，「近くにある顧客を続けて訪問したほうがよい」という直感がある．もちろん，常にそれが最適とは限らないが，少なくとも良い初期解を作る手がかりにはなる．また，すでに得られた解に対して，顧客訪問順序を少し入れ替えるだけで大きく改善することもある．このような問題特有の構造を利用することが，ヒューリスティクスの基本的な発想である．現

実の大規模問題では，最適性保証がなくても，十分に質の高い解を安定して出せることの方が実務上価値を持つことが多い．例えば，配送現場では，数時間かけて最適解を得るよりも，数秒から数分で良い配送計画を作れる方が望ましいことがある．また，不確実性が大きい環境では，厳密最適解そのものの意味が相対的に小さくなり，柔軟に計画を更新できることの方が重要になる．

そのため，ヒューリスティクスは厳密解法の代用品というよりも，別の目的に適した独立の解法体系として理解すべきである．本章では，まずヒューリスティクスの基本的な考え方を説明し，近傍探索という最も基本的な局所改善法を導入する．そのうえで，局所探索の限界とそれを乗り越えるメタヒューリスティクスの考え方を学ぶ．

4.5.2　初期解の構成

ヒューリスティクスの多くは，まず何らかの初期解を作り，その解を改善していくという流れをとる．したがって，どのような初期解を与えるかは，その後の探索結果に大きな影響を与える．近傍探索では，探索は通常，ある解の近傍から始まり，少しずつ改善を重ねるため，初期解が悪すぎると良い解に到達するまで時間がかかることがある．

初期解の構成法には，大きく分けて単純構成法と構成ヒューリスティクスがある．単純構成法とは，問題の制約を満たす解をできるだけ簡単に作る方法である．例えば，VRP であれば，顧客を入力順に割り当て，容量を超えたら新しい車両を使うといった方法でも，ひとまず実行可能解は得られる．このような方法は解の質は高くないかもしれないが，実装が容易であり，探索の出発点として使いやすい．

一方，構成ヒューリスティクスは，初期解の段階からある程度良い解を作ろうとする方法である．例えば，最短経路問題や TSP では，現在いる地点から最も近い未訪問ノードを選び続ける最近隣法がよく知られている．VRP では，顧客を地理的に近いもの同士でまとめたり，セービング法のように「別々のルートを結合するとどれだけ距離を節約できるか」を基準にルートを構成したりする方法がある．

初期解の役割は，単に探索の開始点を与えることだけではない．良い初期解があれば，その後の探索で到達できる解の質も高くなりやすい．また，厳密解法と組み合わせる場合には，良い初期解は上界として働き，分枝限定法などの探索効率を高める効果もある．したがって，初期解構成はヒューリスティクスの前処理ではなく，解法全体の性能を左右する重要な要素である．

4.5.3　近傍探索の基本概念

ヒューリスティクスの中でも最も基本的で重要な考え方が近傍探索である．近傍探索とは，現在の解に対して，それと「少しだけ異なる」解の集合，すなわち近傍を定義し，その中により良い解があれば移動するという操作を繰り返す方法である．

この考え方の根底には，「良い解の近くには，さらに良い解が存在することが多い」という経験的な事実がある．例えば，配送ルートがかなり良さそうに見えても，二つの顧客の訪問順序を入れ替えるだけで距離が短くなることがある．あるいは，顧客を一人だけ別の車両に移すことで，ルート全体のバランスが改善されることもある．このように，小さな変更で解が改善する可能性を系統的に調べるのが近傍探索である．

近傍探索を定義するには，まず解空間と近傍構造を定める必要がある．解空間とは，問題のすべての実行可能解の集合である．近傍構造とは，各解に対して，どの解を「近くにある」とみなすかを決める規則である．例えば，TSP では，巡回路中の二つの辺を交換して新しい巡回路を作る 2-opt 近傍が代表的である．VRP では，ある顧客を別の位置に挿入する挿入近傍や，二つの顧客を交換する交換近傍などが用いられる．

近傍探索の基本的な流れは単純である．まず初期解を用意し，その近傍の中から改善解を探す．改善解が見つかれば，そこへ移動し，再び新しい近傍を調べる．これを改善解が見つからなくなるまで繰り返す．この結果得られる解は，その近傍構造に関して局所最適解である．すなわち，定義した近傍の範囲内ではこれ以上改善できない解である．

この意味で，近傍探索は連続最適化における勾配法とよく似た位置づけを持つ．勾配法が局所的な微分情報を利用して目的関数を下げていくのに対し，近傍探索は離散的な近傍情報を利用して目的関数を下げていく．したがって，近傍探索は組合せ最適化における局所改善法の基本形とみなすことができる．

4.5.4 局所探索と局所最適解

近傍探索の最も基本的な形は局所探索である．局所探索では，現在の解の近傍を調べ，目的関数値が改善する解が見つかればそこへ移動する．改善解が複数ある場合には，最初に見つかったものへ移動する方法や，最も良い改善解へ移動する方法などがある．前者は firstimprovement，後者は bestimprovement と呼ばれることが多い．

局所探索は，実装が簡単でありながら，多くの問題でかなり良い解を与える．TSP に対する 2-opt 局所探索はその典型であり，単純な初期巡回路に対して辺の交換を繰り返すだけで，距離を大きく改善できることが多い．VRP でも，顧客の挿入や交換といった単純な近傍操作を組み合わせるだけで，初期解より大幅に良いルートが得られることが珍しくない．

しかし，局所探索には明確な限界もある．それは，局所最適解に到達すると，それ以上動けなくなることである．局所最適解とは，定義した近傍の中では改善解が存在しない解であるが，問題全体で見ればより良い解が離れた場所に存在することがある．この状況は，連続最適化における局所最適と大域最適の違いに対応している．

例えば，ある配送ルートが，どの単純な交換でも改善しない状態になっていたとしても，複数の顧客を同時に再配置すれば大きく改善することがある．しかし，近傍構造が単純すぎると，そのような解には到達できない．したがって，局所探索の性能は，どのような近傍を定義するかに強く依存する．近傍が小さすぎれば探索は速いが改善の余地が少なくなり，近傍が大きすぎれば改善の可能性は高まるが探索コストが増える．このバランスをどうとるかが，近傍探索設計の中心的な課題である．

4.5.5 代表的な近傍構造

近傍探索の性能を決める最大の要素は，近傍構造の設計である．近傍とは単に「少し異なる解」の集合であるが，その「少し」の定義が問題ごとに重要になる．良い近傍構造とは，一回の操作で意味のある改善が期待でき，しかも探索コストが過大にならないものである．

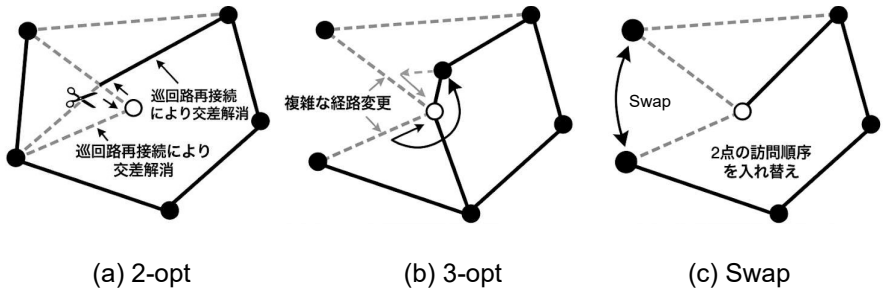


図 4.14: 代表的な近傍構造

TSP では，2-opt，3-opt，swap などが代表的な近傍である．2-opt は，巡回路中の二本の辺を選び，それらを切り替えることで巡回路を再接続する操作である．この操作は，交差している辺を解消する効果を持つことが多く，実際に有効である．3-opt はさらに三本の辺を交換するものであり，より大きな改善が期待できるが，探索量も増える．

VRP では，より多様な近傍が使われる．例えば，ある顧客を現在の位置から取り除いて別の場所に挿入する relocate 近傍，二つの顧客の位置を入れ替える swap 近傍，二つのルート間で顧客列を交換する 2-opt*近傍などがある．これらの操作は，ルート内部の順序改善だけでなく，ルート間の負荷バランスの調整にも役立つ．容量制約や時間窓制約がある場合には，近傍操作後に実行可能性を素早く判定する仕組みが必要になる．

4.5.6　貪欲法と局所改善の違い

貪欲法は，その時点で最も良さそうに見える選択を逐次積み重ねていく方法である [1]．例えば，最短経路問題では，ある条件のもとで「暫定的に最も近いノードを確定する」Dijkstra 法は貪欲法的な性格を持っている．TSP の最近隣法も，その時点で最も近い未訪問都市を選ぶという意味で貪欲法である．

これに対して局所改善法は，いったん解を作ったあとで，その解の局所的な修正を繰り返して改善していく方法である．したがって，貪欲法が「構成」を中心とするのに対し，局所改善法は「修正」を中心とする．貪欲法は速くて実装しやすいが，一度行った選択を後から見直さないため，初期の誤りがそのまま残ることがある．一方，局所改善法は後から修正できる柔軟性を持つが，初期解が必要であり，局所最適に陥る可能性がある．

実際には，この二つは対立するものではなく，むしろ組み合わせて使われることが多い．例えば，最近隣法で TSP の初期巡回路を作り，その後 2-opt で改善するという方法は典型的である．VRP でも，セービング法や挿入法で初期ルートを作り，そのあと relocate や swap による局所探索で磨き上げることが一般的である．このように，構成法と改善法をどう組み合わせるかが，ヒューリスティクス設計の重要な論点となる．

4.5.7　メタヒューリスティクスの考え方

局所探索は有用であるが，局所最適解にとどまってしまうという限界を持つ．この限界を乗り越えるために発展してきたのがメタヒューリスティクスである．メタヒューリスティクスとは，特定の問題に強く依存しない高水準の探索戦略であり，局所探索を拡張・制御することで，より広い解空間を探索しようとする枠組みである [21].

メタヒューリスティクスには，代表的な工夫や典型的な手法がいくつか存在する．マルチスタート（multi-start）は，初期解を一つに固定せず，複数の異なる初期解から局所探索を繰り返す手法である [4]．局所探索は初期解に強く依存するため，異なる初期点から探索を行うことで，多様な局所最適解を得ることができ，より良い解に到達する可能性が高まる．

また，改悪解の受け入れも重要な工夫の一つである [15]．通常の局所探索では，目的関数が改善する方向にのみ解を更新するが，これでは局所最適から脱出できない．そこで，一時的に目的関数が悪化する解への移動を許容することで，探索空間のより広い領域を探索できるようにする．焼きなまし法はその代表例であり，温度パラメータに基づいて悪化解の受け入れ確率を制御することで，探索の初期には広く探索し，徐々に収束させる仕組みを持つ．

さらに，可変近傍探索（Variable NeighborhoodSearch;VNS）は，単一の近傍構造に依存するのではなく，複数の異なる近傍構造を切り替えながら探索を行う手法である [16]．ある近傍で局所最適に到達した場合でも，別の近傍ではまだ改善の余地がある可能性があるため，近傍を体系的に変更することで，より広範な探索が可能となる．

メタヒューリスティクスは，パラメータ設定，近傍構造，初期解，停止条件などによって性能が大きく変化する．また，問題ごとに適した設計が必要であり，アルゴリズム名だけを知っていれば十分というわけではない．それでも，複雑で大規模な現実問題に対して，高品質な解を比較的短時間で得られることから，実務でも研究でも広く用いられている．

クイズ 7：解法の設計 1

ある都市で配送ルートを改善するために，まず単純な構成法で初期解を作成し，その後，顧客の挿入と交換からなる近傍を用いて局所探索を行った．その結果，計算開始直後には目的関数値が大きく改善したが，途中から長時間にわたって似たような形の解の周辺を探索するだけになり，最終的な解の質も実行ごとの差が大きかった．また，計算時間には一定の上限があり，探索量だけを無制限に増やすことはできない．
この状況に対して，解法改善のために次にとるべき行動として適切なものをすべて選べ．

(ア) 改善停滞の原因を近傍構造だけに限定せず，初期解の作り方，改善解の選び方，停止

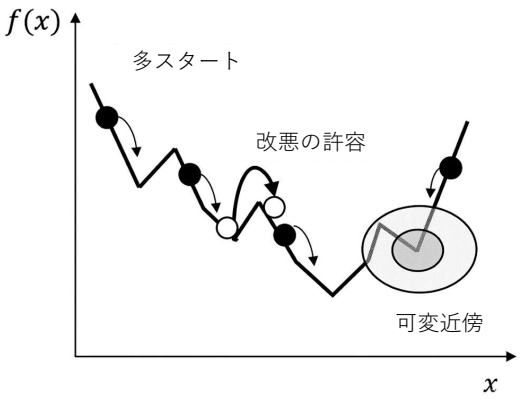


図 4.15: 代表的なメタヒューリスティクス

条件，探索の多様性不足なども含めて，探索設計全体のどこが律速になっているかを切り分けて確認する．

(イ) 近傍を大きくすれば現在の探索で届かない改善方向を扱える可能性があるので，まずは他の要因を保留し，近傍構造の強化に集中して挙動の変化を見る．その方が一度に一つの要因だけを変えることになり，原因特定も行いやすい．

(ウ) 初期解が探索の到達領域を制限している可能性はあるが，最終解のばらつきが大きい以上，それは主因ではないと考え，初期解の検討は後回しにしてよい．

(エ) 現行の停止条件や反復回数の設定を点検し，十分な探索が行われる前に打ち切っていないかを確認する．必要なら，探索時間の配分を変えずに停止条件だけを調整した比較実験を行う．

→ 正解・解説は節末を参照

クイズ 8：解法の設計 2

ある都市で配送ルートを改善するために，利用可能な計算時間を「初期解を丁寧に作る段階」と「その後の局所探索で改善する段階」のどちらに多く配分するかを検討している．精巧な初期解構成法を使えば探索開始時点の解の質は高くなるが，そのぶん改善に使える時間は短くなる．逆に，初期解を簡単に作れば探索時間は長く取れるが，出発点が悪くなり

やすい．予備実験では，問題によっては初期解の差がそのまま最終解に残る一方で，別の問題では探索段階で大きく挽回できることも分かっている．
この状況に対して，次にとるべき行動として適切なものをすべて選べ．

(ア) 初期解の質が高いほどその後の探索が有利になることは多いので，まずは構成法を強化し，残った時間の中で探索を行う設計を基本とする．問題ごとの差よりも，出発点を良くすることの効果を優先して考える．

(イ) 短時間の比較実験を行い，初期解の質の違いが最終解にどの程度残るのかを確認する．その結果に基づいて，構成法に時間をかける価値が高いのかを判断する．

(ウ) 近傍探索の一回あたりの計算コストや，初期解構成に要する時間を実測し，初期解に時間をかけた場合に改善反復回数がどれだけ減るのかを確認する．限られた計算時間の下では，こうした機会費用の把握が設計判断に直結する．

(エ) どの問題でも探索能力を優先すべきとまでは言えないが，初期解の比較実験は時間を要するため，まずは最も粗い初期解を固定して近傍探索だけを詳細に調べるのが効率的である．

→ 正解・解説は節末を参照

クイズ 9：解法の設計 3

ある都市で巡回セールスマン問題を解くために，単一の近傍による局所探索を異なる初期解から何度も実行したところ，最終的には似た水準の解へ収束することが多かった．一方で，ごく一部の実行では，探索の途中で一時的に目的関数が悪化したあと，それまで到達できなかった水準の解が得られた．ただし，計算時間は限られており，探索戦略を複雑化すると一回あたりの試行回数は減少する．
この状況に対して，解法改善のために次にとるべき行動として適切なものをすべて選べ．

(ア) 現在の単一近傍では表現できていない改善方向が存在する可能性を確認するために，異なる種類の近傍を導入して比較する．とくに，ルート内の順序変更とルート間の再配分では得意な改善が異なるため，近傍依存の限界かどうかを見極める必要がある．

(イ) 一時的な改悪を経てより良い解へ到達した実行が存在する以上，単調改善だけでは越えられない障壁がある可能性を考えるべきである．そのため，改悪解受け入れを含む戦略を候補に入れ，局所最適からの脱出能力を検証することには意味がある．

(ウ) 現在の探索でもごく一部では良い解が得られているので，探索戦略そのものを変えるより，同じ単一近傍・同じ受理規則のまま試行回数だけを増やして再現を狙う方が，実装の簡潔さを保てるという利点がある．

(エ) 複数近傍や改悪解受け入れを導入する場合でも，一回あたりの計算時間の増加によって総試行回数がどれだけ減るかを確認し，解の質向上と計算負荷増加のどちらが支配的かを比較しながら採用を判断する．

→ 正解・解説は節末を参照

4.5.8　ヒューリスティクスの評価

ヒューリスティクスの性能は，解の質，計算時間，安定性，実装容易性など，複数の観点から評価しなければならない．

まず最も基本的なのは，得られた解の質である．厳密解が分かっている小規模問題であれば，最適値との差，すなわち最適性ギャップを評価できる．最適解が分からない場合には，既存手法との比較や，下界との比較を行うことが多い．次に重要なのは計算時間である．実務では，同じ品質なら短時間で出る手法の方が価値が高い．また，ランダム性を含むアルゴリズムでは，複数回実行したときのばらつきも重要になる．平均性能だけでなく，最悪ケースや安定性を見る必要がある．

さらに，ヒューリスティクスの評価では，再現性も重要である．どのインスタンスで，どのパラメータで，どれだけの計算時間を使い，どのような比較をしたのかを明確にしなければ，性能比較は公平にならない．研究では，ベンチマーク問題を用いた比較や，複数のサイズ・条件での感度分析が求められる．この意味で，ヒューリスティクスの研究は，新しい近傍やアルゴリズムを提案するだけでなく，その有効性を説得的に示す評価設計まで含めて成立する．

クイズ 7 正解・解説

正解：(ア)，(エ)

この状況では，改善停滞の原因を一つに決めつけず，探索設計のどこが律速になっているかを見極めることが重要である．近傍構造の不足は有力な要因の一つではあるが，初期解の偏り，改善解選択規則，停止条件，探索の多様性不足なども同時に疑うべきである．

（ア）は適切である．探索設計全体を切り分けて確認するという姿勢は，最初にとるべき行動として妥当である．

（エ）も適切である．十分な探索の前に打ち切っているだけであれば，近傍を変える前に停止条件を見直す方が本質的な場合がある．

（イ）は一理あるが，この段階で近傍強化だけに集中するのは狭すぎる．原因候補が複数ある以上，近傍だけを優先して固定的に疑うのは適切ではない．

（ウ）も部分的にはもっともらしいが，最終解のばらつきが大きいことは，むしろ初期解依存性の存在を示唆する可能性もある．したがって，初期解の検討を後回しにしてよいとは言えない．

クイズ 8 正解・解説

正解：（イ），（ウ）

この状況では，初期解構成と探索改善のどちらを優先すべきかを，一般論で決めることはできない．問題によって，初期解の差が強く効く場合もあれば，探索段階で大きく挽回できる場合もあるためである．

（イ）は適切である．初期解の差がどれだけ最終解に残るかを実験的に確認することは，時間配分を決めるうえで本質的である．

（ウ）も適切である．限られた計算時間のもとでは，初期解構成に時間を使うことの機会費用を把握する必要がある．近傍探索の一回あたりコストを測ることはその判断に直結する．

（ア）は一理あるが，出発点を良くすることを常に優先すべきとは言えない．問題によっては，探索段階で大きく挽回できることがすでに示唆されているからである．

（エ）も，実験の簡素化という意味では理解できるが，初期解側の影響を見ないまま探索だけを詳しく調べても，時間配分の判断には不十分である．

クイズ 9 正解・解説

正解：（ア），（イ），（エ）

この状況では，単一近傍かつ単調改善のみの探索が，特定の局所最適群に捕まりやすい可能性が示唆されている．また，一時的な悪化を経てより良い解へ到達した例があることは，単調改善だけでは越えられない障壁が存在する可能性を示している．

（ア）は適切である．複数近傍を導入すれば，現在の近傍では表現できない改善方向を探索できるかもしれない．

（イ）も適切である．改悪解受け入れは，局所最適から脱出するための有力な戦略であり，観測結果とも整合している．

（エ）も適切である．探索戦略の強化には計算コストの増加が伴うため，解の質向上と試行回数減少のバランスを比較しながら導入する必要がある．

（ウ）は一理ある．同じ戦略で試行回数を増やせば，まれに良い解が再現される可能性はある．しかし，現状の観測は探索戦略そのものの限界を示唆しており，次にとるべき行動としては，戦略を変えずに回数だけ増やすよりも，脱出能力を高める方向の検討を優先する方が妥当である．

コラム：神奈川県央地域でのデマンド型交通の対象地区の選定 – 永井鴻平

神奈川県央に位置する海老名市, 綾瀬市, 大和市, 座間市の 4 市を対象に, オンデマンド交通の対象ゾーン選定の検討を行った.

対象ゾーン内の利用者は, ハブから出発した車両にピックアップされ, 一旦ハブへ集約, その後目的地へ送られることを想定する. ハブは, 4 市のおよそ中央に位置する, さがみの駅南口とした.

対象の需要は, 4 市内を相互発着する朝 7 時から 10 時台のトリップとし, 鉄道によるものを除いた. 対象のトリップ数は 1,723 件であった. 選定するゾーンの単位は東京都市圏パーソントリップ調査の地区コードとし, 4 市内のゾーン数は合計で 319 であった.

ゾーンの選定は, 目的関数

$$\sum_{i \in I} w_i \tilde{q}_i x_i - \nu k \tag{4.112}$$

を最大化する問題として定式化した. 決定変数はゾーン i が選定されるか否かを表す二値変数 x_i と, サービスに供される車両の台数 k である. ここで, I はゾーン全体の集合, w_i はゾーン i の重み, $\tilde{q}_i$ はゾーン i の有効需要, ν は車両 1 台あたりのコストである.

重み w_i は, 自身での運転や徒歩での移動が困難な利用者を重視したサービス設計のため, ゾーン i の対象トリップのうち, 60 歳以上または外出に身体的困難を伴う利用者によるトリップの割合 r_i を用いて,

$$w_i = 1 + \theta r_i \tag{4.113}$$

と定義した. θ は外出が困難な利用者をどの程度重視するかを表すパラメータである. $\theta = 1$, すなわちゾーン i の対象トリップが全て外出困難者によるとき, $w_i = 2$ となり, そのゾーンの重みが通常の 2 倍になる.

有効需要 $\tilde{q}_i$ は, 対象トリップのうち, サービスの初期利用者として 10%程度が期待できると仮定し,

$$\tilde{q}_i = 0.1 q_i \tag{4.114}$$

と定義した. q_i はゾーン i を発着する対象トリップ数である.

車両 1 台あたりのコスト ν は, サービス対象とする需要とのバランスを考慮し, 1 台あたりのコストが需要 20 件と対応するよう, $\nu = 20$ とした.

決定変数の整数制約条件として,

$$x_i \in \{0, 1\} \tag{4.115}$$

$$k \in \mathbb{Z}, 1 \leq k \leq 30 \tag{4.116}$$

を課した. ゾーン i がサービス対象に選定されるとき $x_i = 1$, 選定されないとき $x_i = 0$ となる. 車両の台数の上限は 30 台とした.

また, 容量制約

$$\sum_{i \in I} \tilde{q}_i s_i x_i \leq \eta H k, \tag{4.117}$$

およびサービス対象とする需要の最低カバー率制約

$$\sum_{i \in I} \tilde{q}_i x_i \geq \gamma \sum_{i \in I} \tilde{q}_i \tag{4.118}$$

を課した．ここで, s_i はハブからゾーン i を往復する時の所要時間であり, 平均走行速度 18 km/h, 乗り降りや車両の折り返しにかかるバッファ時間を 8 分と仮定して,

$$s_i = \varepsilon \left(\frac{2 d_i}{18} \times 60 + 8\zeta \right) \text{ [分]} \tag{4.119}$$

と定義した. d_i はハブからゾーン i までの走行距離である. 走行距離を求める際, ゾーンの代表点はゾーン内の対象トリップ発着点の重心とし, 実際の道路ネットワークに基づいた最短経路を用いた. ε は, 需要が集中したゾーンでの相乗り効果を考慮するための係数であり, ゾーン i の有効需要の密度 p_i [件/km^2] を用いて,

$$\varepsilon = 1 - 0.1565\sqrt{p_i} \tag{4.120}$$

と定義した. 選定対象のゾーンのうち, p_i 最大のゾーンで $\varepsilon \approx 0.3$ となるよう設定した.

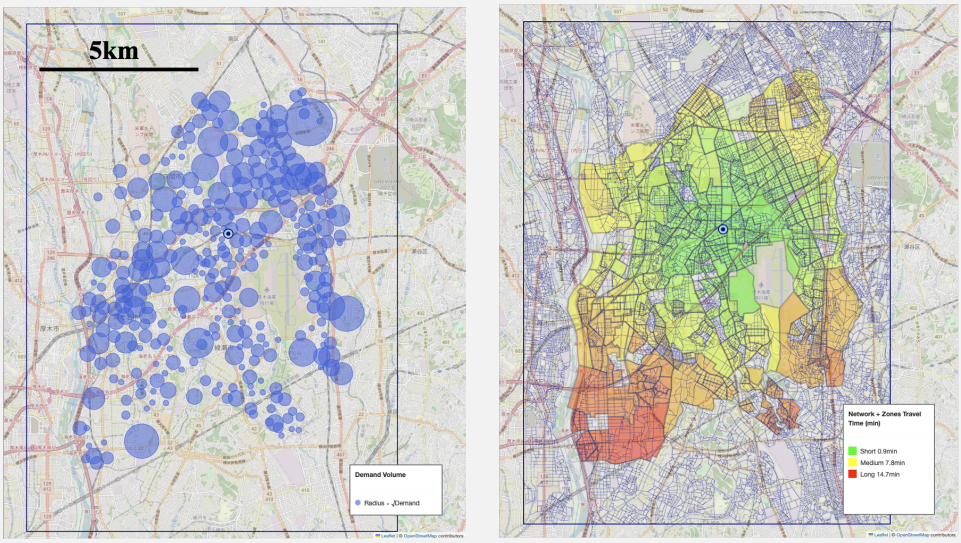
また, ζ は外出困難な利用者が利用する際, 乗降などにその他の利用者よりも時間がかかることを表現するための係数であり,

$$\zeta = 1 + 7 r_i \tag{4.121}$$

と定義した.

η は車両の稼働率, H はサービス運用時間, γ は需要の最低カバー率である. 需要の対象トリップの抽出時間に合わせ, $H = 240$ [分] とした. また, $\eta = 0.85$, $\gamma = 0.5$ とした.

この問題は, ゾーンの選定と車両台数の決定を同時に行う組合せ最適化問題であり, 整数計画問題として解くことができる. 対象の需要 $\tilde{q}_i$ の分布と, s_i/ε の分布, および d_i の算出に用いた道路ネットワークを図 4.16 に示す.



(a)　　(b)

図 4.16: ゾーンごとの (a) 需要 $\tilde{q}_i$ の分布, (b) ハブからの所要時間 s_i/ε の分布

$\theta = 10$ としたときのゾーンの選定結果を図 4.17 に示す. このとき, 車両台数 $k = 15$ [台] であった.

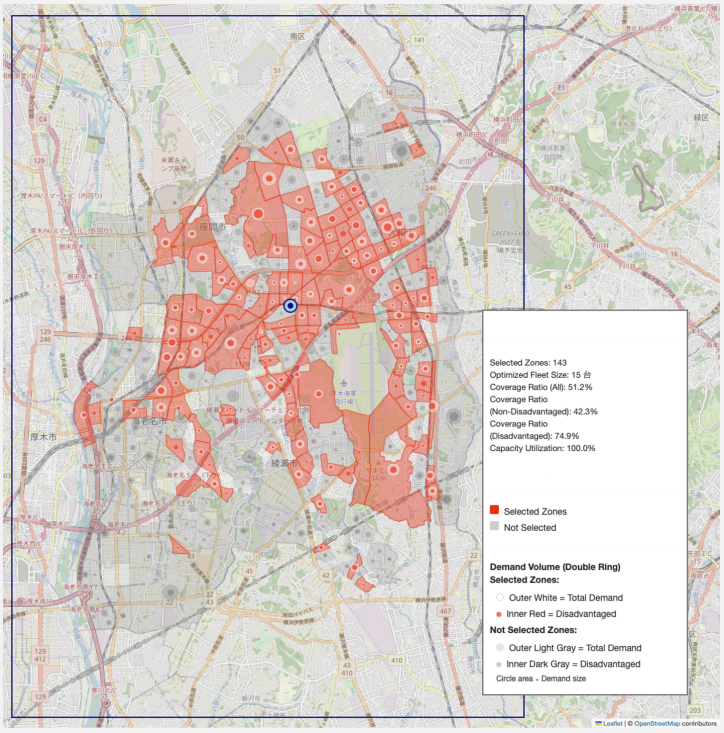


図 4.17: ゾーンの選定結果

結果から, ハブを中心にゾーンが選択されている様子が見られる. 一方で, ハブから遠いゾーンであっても, r_i が大きいゾーンは選択されやすく, ハブに近くても r_i が小さいゾーンは選択されていないものがあることもわかり, 重み w_i の設定や, ζ によるバッファ時間の増加がゾーンの選定に影響を与えていることが確認できた.

次に, サービスに供する車両の全てを置き換え, 外出が困難な利用者が利用するときもバッファ時間を増加させないようにした場合を考える. 展開式のステップや, 車椅子に乗ったまま乗降できるような設備を備えた高機能車両を導入することを想定し, 車両コストは相応に増加するものとした.

目的関数は, 式 (4.112) の第 2 項に対して通常仕様車に対する高機能車のコスト係数 β をかけ,

$$\sum_{i \in I} w_i \tilde{q}_i x_i - \beta \nu k \tag{4.122}$$

とした. また, 式 (4.119) において $\zeta = 1$ とし, 外出困難な利用者が利用するときもバッファ時間を増加させないようにした.

$\theta = 10, \beta = 1.2$ としたときのゾーンの選定結果を図 4.18 に, 車両台数やカバー率などを, 通常仕様車の結果と合わせて表 4.2 に示す.

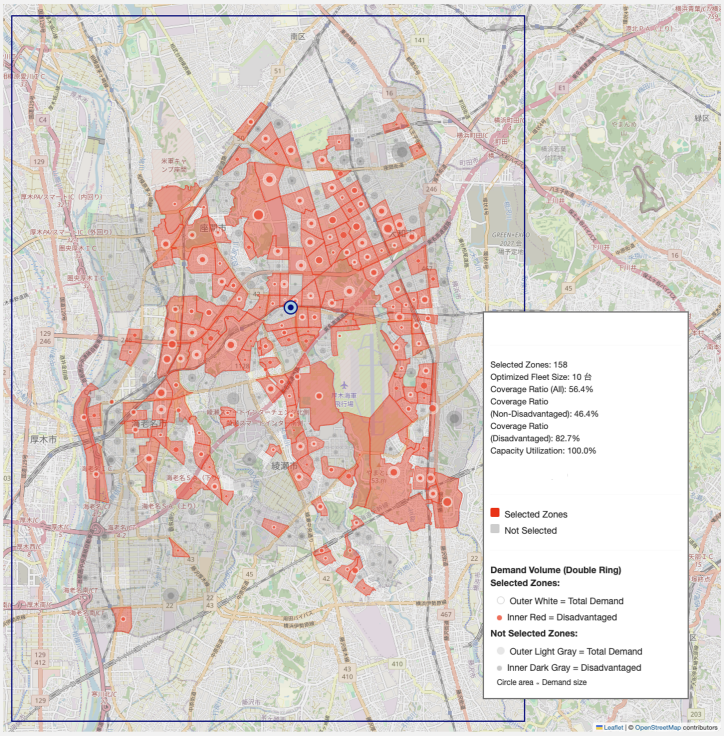


図 4.18: 高機能車両を導入した場合のゾーンの選定結果

表 4.2: 通常仕様車と高機能車両のゾーン選定結果の比較

	通常仕様車	高機能車両
車両台数	15	10
選定ゾーン数	143	158
需要カバー率	51.2%	56.4%
外出困難者のカバー率	74.9%	82.7%
その他の利用者のカバー率	42.3%	46.4%

結果を図 4.17 と比較すると, 高機能車を導入することで, 外出困難な利用者が利用するゾーンの選定がより促進されている様子が見られる. 表 4.2 から, 高機能車の導入によって需要のカバー率が向上したことがわかるが, 外出困難者のカバー率だけでなく, その他の利用者のカバー率も向上していることがわかる.

最後に, 高機能車を導入した場合について, 外出困難者の重視度 θ および高機能車のコスト係数 β を変化させてゾーン選定を行い, それぞれの係数の影響を調べた. 影響は選定されたゾーンに基づく需要のカバー率と, 決定された車両台数で確認した. 結果を図 4.19 と図 4.20 に示す.

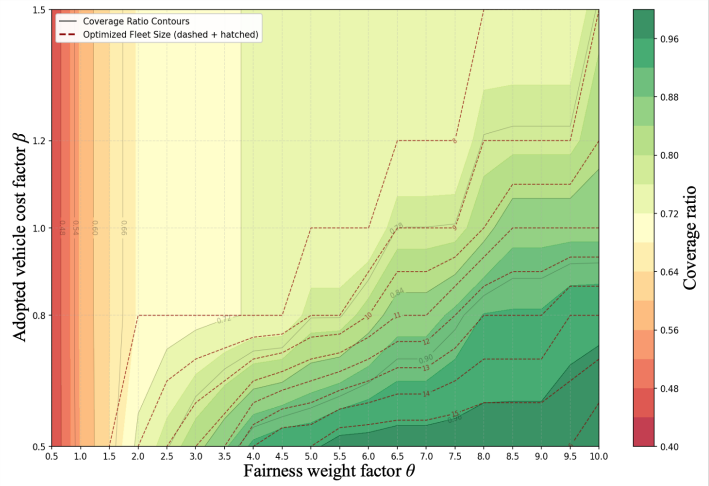


図 4.19: 外出困難者の重視度 θ と高機能車のコスト係数 β の影響 (外出困難者)

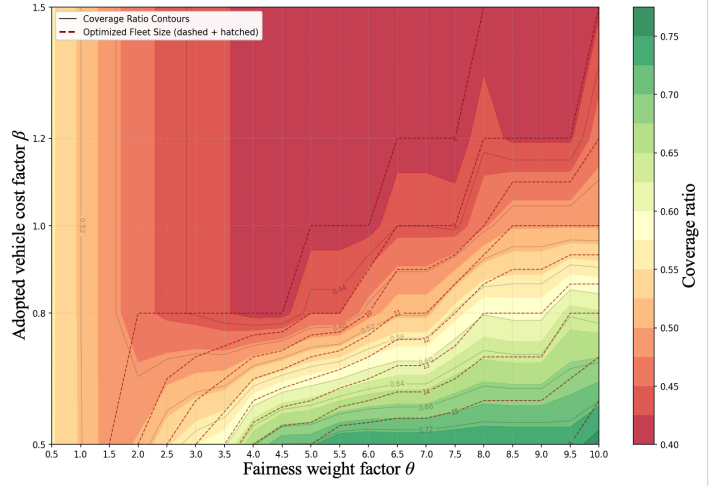


図 4.20: 外出困難者の重視度 θ と高機能車のコスト係数 β の影響 (その他の利用者)

当然, 外出困難者を重視するほど, また, 高機能車のコストが下がるほど, 外出困難者のカバー率は向上する. 通常仕様車のみでサービスを運営すると, 外出困難者を重視するにつれて車両容量が逼迫し, その他の利用者のカバー率が低下することが想定されるが, 高機能車の場合では, 双方のカバー率を両立させやすい. 特筆すべき現象として, θ も β も大きい, すなわち「車両は高いが外出困難者は重視したい」領域において, その他の利用者のカバー率が $\theta = 1$ のときよりも低下してしまうことがある. これは, 外出困難車を重視した結果, その他の利用者が不利益を被っている領域と解釈することができ, 公的サービスの運営としては望ましくない. 今回の検討条件 ($\beta = 1.2$) では, $\theta \leq 6$ ではこうした傾向が強く, 公平感を維持しつつ高機能車を導入することは難しいと考えられる. $\theta \geq 8$ 程度になると, 高機能車の導入によって利用者全体が恩恵を得やすくなり, 公平感を維持しつつ高機能車を導入することが現実的になる.

かなり大きな θ や ζ, 控えめな β を設定し, 高機能車を有利に扱う検討を行ったが, それでも高機能車を合理的に導入できる条件は限定的であった. この結果は, 経済的合理性によってオンデマンド交通のサービス性を向上することは難しく, いかに公的資金を投入するか, またその理由を社会に納得させるか, といった課題の重要性を示唆している.

コラム：高尾駅周辺における自動運転オンデマンド交通の拠点配置分析 – 名嘉山結月

近年、自動運転オンデマンド交通は、鉄道や幹線バスを補完する端末交通として期待されている。しかし、導入にあたっては、どの範囲を対象とするか、どこに拠点を置くか、どの程度の車両台数で需要をカバーできるかを検討する必要がある。本ミニ研究では、高尾駅周辺 3km 圏を対象に、複数拠点配置と需要ゾーン割当を組み合わせた最適化モデルを構築し、サービス設計上の示唆を得ることを目的とした。

(a)データと前提条件　使用したのは 2019 年東京都市圏パーソントリップ調査データであり、高尾駅から半径 3km 圏に関係するトリップを抽出した。対象需要は 500m メッシュで集計し、各メッシュを需要ゾーンとして扱う。図 4.21 に対象地域と 500m メッシュ需要分布を示す。需要量 d_i はゾーンごとに算出し、高齢者や免許非保有、自由に使える自動車の有無、外出困難、高齢者同行などをもとに公平性指標 $fairness_ratio$ *を設定した。潜在需要は需要量の 20%と仮定した。高尾駅を主拠点とし、追加拠点候補は 3km 圏内を空間的に分散してカバーするように設定している。

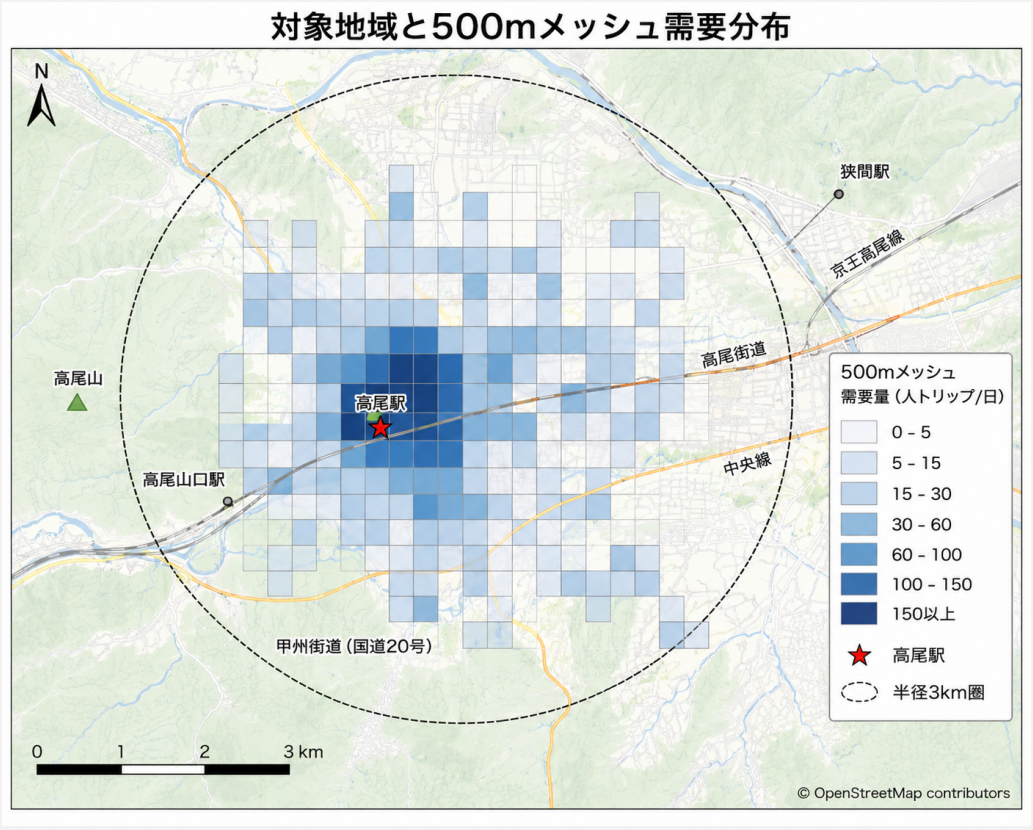


図 4.21: 対象地域と 500m メッシュ需要分布

*公平性指標は、各個人の支援必要度をもとにゾーン単位で集計した。

$$s_i = w_1 E_i + w_2 L_i + w_3 C_i + w_4 D_i + w_5 A_i$$

$$fairness_ratio_z = \frac{1}{N_z} \sum_{i \in z} s_i$$

ここで、E_i は高齢者、L_i は免許非保有、C_i は自由に使える自動車なし、D_i は外出困難、A_i は高齢者同行を表す 0/1 変数である。なお、本分析における fairness_ratio は厳密な福祉指標ではなく、移動制約のある需要を相対的に重視するための proxy として扱っている。

(b)初期分析：3 種類のサービス設計　まず、設計思想の違いを比較するため、以下の 3 種類のモデルを比較した。

- 半径型（radius）：高尾駅から一定距離圏内を一括してサービス対象とする。実装容易性を重視した単純な構造。
- 分散型（demand-based）：需要の大きいゾーンを個別に選択し、効率性を重視する構造。
- カバー率型（coverage）：一定割合以上の需要を必ずカバーし、公平性・公共性を重視する構造。

比較の結果、カバー率を高めるほどサービス時間が増加し、効率性と公平性の間に明確なトレードオフが存在することが確認された。また、需要の大きいゾーンを優先する分散型では効率性が高い一方、空間的な偏りが生じやすいことが示唆された。表 4.3 に 3 種類のサービス設計比較を示す。

モデル	特徴	主な示唆
半径型	実装容易性重視	シンプルだが公平性が不明確
分散型	効率性重視	需要偏在で空間的偏りが生じやすい
カバー率型	公共性重視	coverage 増加で service time が増加

表 4.3: 最適化結果比較

(c)モデル化　今回は、本格的な VRP ではなく、複数拠点配置＋需要ゾーン割当問題として定式化した。各需要ゾーンは、選択された拠点のいずれかに割り当てられる。高尾駅は必ず使用する拠点とし、追加拠点を候補集合から選択する。
サービス時間は、拠点からゾーンへの往復時間に加えて、乗降・待機バッファおよび高尾駅との接続コストを含む近似値として定義した。

$$t_{ij} = 2 \times \frac{l_{ij}}{v} \times 60 + b + \gamma \times \frac{h_j}{v} \times 60$$

ここで、l_{ij} は拠点 j からゾーン i までの距離 [km]、v は平均速度 [km/h]、b は乗降・待機バッファ [min]、h_j は拠点 j と高尾駅との距離 [km]、γ は駅接続コストの重みである。
この式は、端末交通として各拠点が高尾駅と接続される必要性を近似的に表現したものである。
解法には 0-1 整数計画（MILP）を用い、目的関数は総サービス時間の最小化とした。
決定変数は以下のとおりである。

$$x_j = \begin{cases} 1 & \text{拠点 } j \text{ を使用する場合} \\ 0 & \text{それ以外} \end{cases}, \qquad y_{ij} = \begin{cases} 1 & \text{ゾーン } i \text{ を拠点 } j \text{ に割り当てる場合} \\ 0 & \text{それ以外} \end{cases}$$

ここで、d_i はゾーン i の有効需要（effective demand）、t_{ij} は拠点 j がゾーン i を担当する場合のサービス時間 [min]、y_{ij} はゾーン i を拠点 j に割り当てるかを表す 0/1 変数である。
なお、繰り返しになるが、有効需要は各ゾーン需要の 20%と仮定した。また、f_i はゾーン i の公平性指標、C_j は拠点 j の処理可能容量を表す。
目的関数は次の形を想定した。

$$\min \sum_i \sum_j d_i t_{ij} y_{ij}$$

制約としては、以下の通り、coverage 制約、fairness 制約、拠点数制約、容量制約、割当制約を設定した。

$$\sum_i d_i z_i \geq \alpha \sum_i d_i \qquad \text{(coverage 制約)}$$

$$\sum_i d_i f_i z_i \geq \beta \sum_i d_i f_i \qquad \text{(fairness 制約)}$$

$$\sum_j x_j = P \qquad \text{(拠点数制約)}$$

$$\sum_j y_{ij} \leq 1 \qquad \forall i \quad \text{(割当制約)}$$

$$y_{ij} \leq x_j \qquad \forall i, j$$

$$\sum_i d_i t_{ij} y_{ij} \leq C_j x_j \qquad \forall j \quad \text{(容量制約)}$$

(d)計算条件　高尾駅は必須拠点とし、追加拠点数 P を変化させる。coverage threshold は 0.4、0.5、0.6 を比較し、総車両台数 fleet を変更した。さらに、nearest-k 制約を導入し、$k = 1$ に近づくほど各拠点の担当圏域が独立する条件を検討した。

項目	内容
対象地域	高尾駅周辺 3km 圏
需要単位	500m メッシュ
effective demand	demand の 20%
fleet	18 / 24
必拠点	高尾駅
追加拠点数 P	1-4
coverage threshold	0.4, 0.5, 0.6
fairness_threshold	0.3
nearest-k 制約	$k = 1$ に近い条件

表 4.4: 主要パラメータ一覧

(e)計算結果　最適解の一例として、$P = 3$ の場合には「高尾駅 + candidate_6 + candidate_8」が選択された。図 4.22 に最適拠点配置結果を示す。このとき、coverage は約 0.5、capacity utilization は約 1.0 となり、車両容量制約が強く効いていることが確認された。図 4.23 に示すように，coverage を高めると総サービス時間が大きく増加し、効率性と公共性のトレードオフが確認された。また、fleet 数を減少させた場合には、coverage 制約を満たす解が存在しない infeasible ケースが確認された。これは、自動運転オンデマンド交通の成立性が、需要量だけでなく投入可能な車両規模にも強く依存することを示している。一方で、fleet 数を増加させても、拠点配置や coverage 条件が同一である場合には、総サービス時間の改善は限定的であった。したがって、fleet 数は「効率性」よりも「サービス成立性」を左右する要因として機能したと考えられる。

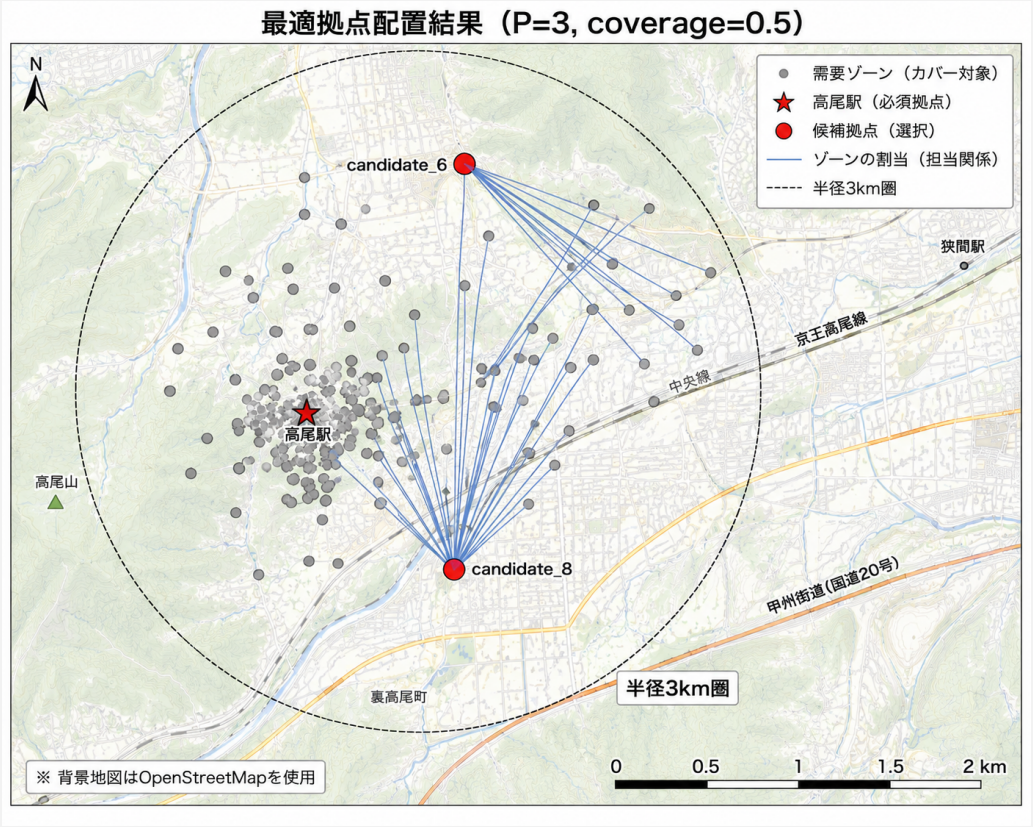


図 4.22: 最適拠点配置結果

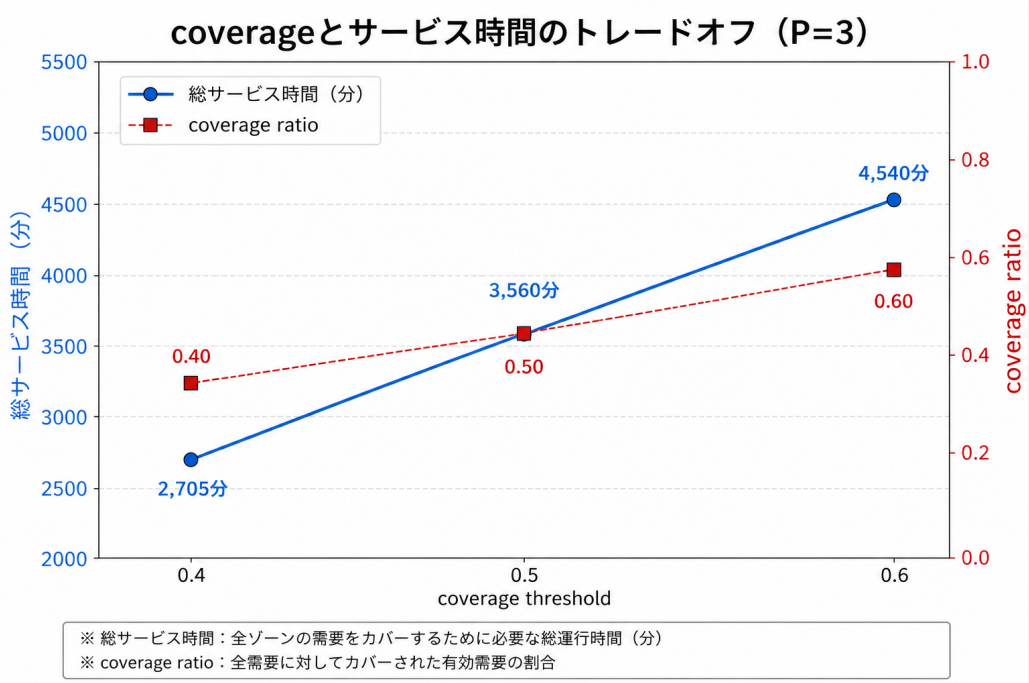


図 4.23: coverage と service time のトレードオフ

ケース	選択拠点数	カバー率	公平性	総サービス時間 (分)	平均拠点容量利用率
P1_k8	1	0.5	0.526	3006.6	0.651
P2_k8	2	0.5	0.517	3167.1	0.651
P3_k8	3	0.5	0.515	3633.2	0.651
P4_k8	4	0.5	0.511	3884.2	0.732
cov40_k8	3	0.4	0.4	2805.2	0.52
cov50_k8	3	0.5	0.515	3633.2	0.651
cov60_k8	3	0.6	0.618	4407.1	0.781
beta00_k8	3	0.5	0.515	3633.2	0.651
beta05_k8	3	0.5	0.515	3633.2	0.651
beta10_k8	3	0.5	0.515	3633.2	0.651

表 4.5: 最適化結果比較

(f)分析と考察　本分析では、表 4.5 に示す通り、拠点数 P を増加させても効率性が大きく改善するとは限らず、むしろ coverage の設定や車両容量制約が支配的であることが示された。一般には拠点数を増やすことで各ゾーンまでの距離が短縮される可能性があるが、本

分析では、追加拠点と高尾駅との接続コストや coverage 制約の影響により、総サービス時間が必ずしも減少しなかった。したがって、単に拠点数を増加させるのではなく、拠点配置と担当圏域を一体的に設計する必要があると考えられる。
都市部では需要密度が高いため、「どこをカバーするか」よりも「限られた車両容量でどれだけ効率的に運用できるか」が主要な制約となる可能性がある。一方、地方部では需要分布がより分散するため、拠点配置そのものが支配的な問題となることが考えられる。

(g)今後の展望　本分析を発展させるためには、道路ネットワークを用いた厳密なルーティングや公共施設を考慮した拠点配置、nearest-k 制約の厳密化が求められる。さらに、地方部への展開や避難・物流との統合的分析も今後の課題である。

参考文献

[1] Alidaee, B., Kochenberger, G. A., Amini, M. M.: Greedy solutions of selection and ordering problems, European Journal of Operational Research, Vol. 134, No. 1, pp. 203–215, 2001.

[2] Ball, M. O.: Heuristics based on mathematical programming, Surveys in Operations Research and Management Science, Vol. 16, No. 1, pp. 21–38, 2011.

[3] Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., Vance, P. H.: Branch-and-Price: Column Generation for Solving Huge Integer Programs, Operations Research, Vol. 46, No. 3, pp. 316–329, 1998.

[4] Bräysy, O., Hasle, G., Dullaert, W.: A multi-start local search algorithm for the vehicle routing problem with time windows, European Journal of Operational Research, Vol. 159, No. 3, pp. 586–605, 2004.

[5] Chan, Y., Carter, W. B., Burnes, M. D.: A multiple-depot, multiple-vehicle, location-routing problem with stochastically processed demands, Computers Operations Research, Vol. 28, No. 8, pp. 803–826, 2001.

[6] Doig, A. G., Land, A. H.: An Automatic Method of Solving Discrete Programming Problems, Econometrica, Vol. 28, No. 3, pp. 497–520, 1960.

[7] Gelders, L. F., Van Wassenhove, L. N.: Production planning: a review, European Journal of Operational Research, Vol. 7, No. 2, pp. 101–110, Fourth EURO III Special Issue, 1981.

[8] Gomory, R. E.: An Algorithm for Integer Solutions to Linear Programs, Recent Advances in Mathematical Programming, pp. 269–302, 1963.

[9] Heimfarth, A., Ostermeier, M., Hübner, A.: A mixed truck and robot delivery approach for the daily supply of customers, European Journal of Operational Research, Vol. 303, No. 1, pp. 401–421, 2022.

[10] Hernández-Pérez, H., Salazar-González, J.-J.: A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery, Discrete Applied Mathematics, Vol. 145, No. 1, pp. 126–139, Graph Optimization IV, 2004.

[11] Hitchcock, F. L.: The Distribution of a Product from Several Sources to Numerous Localities, Journal of Mathematics and Physics, Vol. 20, No. 1-4, pp. 224–230, 1941.

[12] Hromkovič, J.: 計算困難問題に対するアルゴリズム理論（組合せ最適化・ランダマイゼーション・近似・ヒューリスティクス）, 丸善出版, 和田幸一，増澤利光，元木光雄，穴井宏和，斉藤努 訳, 2012.

[13] Koskosidis, Y. A., Powell, W. B., Solomon, M. M.: An Optimization-Based Heuristic for Vehicle Routing and Scheduling with Soft Time Window Constraints, Transportation Science, Vol. 26, No. 2, pp. 69–85, 1992.

[14] Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms, European Journal of Operational Research, Vol. 59, No. 3, pp. 345–358, 1992.

[15] Li, H., Lim, A.: Local search with annealing-like restarts to solve the VRPTW, European Journal of Operational Research, Vol. 150, No. 1, pp. 115–127, O.R. Applied to Health Services, 2003.

[16] Mladenović, N., Hansen, P.: Variable neighborhood search, Computers Operations Research, Vol. 24, No. 11, pp. 1097–1100, 1997.

[17] Mosheiov, G.: Vehicle routing with pick-up and delivery: tour-partitioning heuristics, Computers Industrial Engineering, Vol. 34, No. 3, pp. 669–684, 1998.

[18] Murray, C. C., Chu, A. G.: The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery, Transportation Research Part C: Emerging Technologies, Vol. 54, pp. 86–109, 2015.

[19] Rahmaniani, R., Crainic, T. G., Gendreau, M., Rei, W.: The Benders decomposition algorithm: A literature review, European Journal of Operational Research, Vol. 259, No. 3, pp. 801–817, 2017.

[20] Vanderbeck, F., Savelsbergh, M. W.: A generic view of Dantzig–Wolfe decomposition in mixed integer programming, Operations Research Letters, Vol. 34, No. 3, pp. 296–306, 2006.

[21] 久保 幹雄, Pedroso, J. P.: メタヒューリスティクスの数理, 共立出版, 2009.

[22] 寒野 善博, 土谷 隆: 基礎系数学最適化と変分法, 丸善出版, 2014.

[23] 穴井 宏和, 斉藤 努: 今日から使える! 組合せ最適化: 離散問題ガイドブック, 講談社, 2015.

第 5 章

機械学習の基礎

古橋郁一

5.1 機械学習の基礎

機械学習 (machine learning) は，観測データから規則性を抽出し，未知の入力に対する予測や意思決定を可能にする方法論である．しかし，その中心課題は単なる経験的な当てはめではない．観測標本を生み出した確率分布を，どのようなモデル族によって近似し，その近似をどのような最適化問題として解くかという二つの問いが，多くの手法の背後にある．本章では，この視点を軸として，損失関数，勾配法，誤差逆伝播法，ニューラルネットワーク，さらに画像・系列・言語を扱う代表的構造へと議論を進める．

本章の目標は三つある．第一に，機械学習を「確率分布の学習と最適化の問題」として統一的に捉えることである．第二に，教師あり学習 (supervised learning) を中心として，経験リスク最小化 (empirical risk minimization; ERM) と汎化 (generalization) の基礎概念を定式化することである．第三に，現代的な深層学習 (deep learning) の主要要素である確率的勾配降下法，誤差逆伝播法，畳み込みニューラルネットワーク (convolutional neural network; CNN)，再帰型ニューラルネットワーク (recurrent neural network; RNN)，Transformer が，それぞれどのような問題意識と計算上の制約から導かれたかを理解することである．

5.1.1 機械学習の歴史

機械学習の歴史を学ぶ必要があるのは，現在広く用いられている手法が，単に性能競争の結果として現れたのではなく，表現力，計算資源，データ量，最適化の難しさという四つの制約に対する応答として形成されてきたからである．ある時代の主流手法は，その時点で利用可能な計算機とデータのもとで，どの程度複雑な関数を学習でき，どの程度安定に最適化できるかという条件によって規定されていた．この見取り図を先に持つことは，後に学ぶ CNN，RNN，Transformer を個別の技巧としてではなく，問題構造に応じた帰結として理解するために重要である．本章の議論は Bishop [4]，Murphy [21]，Goodfellow [7] などの標準的教科書を踏まえつつ，本書の関心に沿って構成する．

初期のパターン認識 (pattern recognition) では，主たる課題は観測ベクトルからクラスを識別する判別規則を設計することであった．ここでは，特徴抽出は人手で設計されることが多く，識別器は

線形判別，最近傍法，判別分析など比較的単純な形をとった．この段階では，モデルの表現力は限定的であったが，そのぶん計算量は制御しやすく，少量のデータでも扱いやすかった．他方で，入力表現そのものを学習する能力が弱く，複雑な知覚課題には限界があった．

これに対して統計的学習 (statistical learning)[35, 9] は，有限標本からの推定と汎化誤差の制御を理論的に扱う枠組みを整備した．経験リスク最小化，正則化，構造的リスク最小化，カーネル法などは，モデルの容量と汎化性能の折り合いをどのように付けるかという問いに答えるものである．この流れでは，表現力の拡張と同時に，過学習を避けるための理論的統制が重視された．すなわち，機械学習は単に当てはまりのよい関数を探す問題ではなく，未知データに対しても安定に振る舞う関数を推定する問題であるという認識が明確になった．

ニューラルネットワーク (neural network) の発想自体は古いが，1980 年代に誤差逆伝播法が広まったことで，多層モデルの学習が現実的な手続きとして定着した．ここで重要なのは，ニューラルネットワークが高い表現力を持ちながら，同時に最適化の難しさを強く抱え込む点である．浅いモデルでも非線形関数近似は可能であったが，計算資源が限られ，データ量も小さく，初期化や活性化関数の設計も未成熟であったため，深いモデルは長らく安定に学習しにくかった．したがって，表現力だけでは実用的成功は保証されず，それを支える最適化技術と計算基盤が不可欠であった．

2000 年代後半以降の深層学習の成功は，この四つの制約が同時に緩和されたことによって説明できる．第一に，GPU などの計算資源の発展により，大規模な行列演算を反復する訓練が実用化した．第二に，Web やセンサの普及によって大規模データが利用可能になった．第三に，ReLU 系活性化関数，適切な初期化，正規化，SGD の改良などにより，深いネットワークの最適化が容易になった．第四に，深い階層構造が画像・音声・言語における多段の表現学習に適していることが経験的に確認された．この結果，特徴量設計の重心が人手から学習へと移った．

現代の基盤モデル (foundation model) や大規模モデル (large-scale model) は，この延長線上に位置する．大規模事前学習によって汎用的な表現を獲得し，その後に少量の追加データや指示で多様な下流課題へ適応するという発想は，モデルの表現力とデータ量を極限まで押し広げた結果である．ただし，その基盤にあるのは依然として，条件付き分布や系列分布の学習を，巨大な最適化問題として解くという構図である．

この歴史的流れを各構造に結びつけると，CNN は画像に固有な局所性と平行移動対称性を取り込むために現れ，RNN は系列データの時間依存を表現するために導入され，Transformer は RNN の逐次性と長距離依存性の弱さを克服するために発展した．したがって，以後の節で扱う個々のモデルは，すべて「どのような構造仮定を置けば，限られた計算のもとで確率分布をうまく学習できるか」という共通問題に対する別々の答えである．次節では，この問題を教師あり学習の基本設定として明示的に定式化する．

5.1.2　機械学習の基本設定

機械学習の議論を精密に進めるためには，まず何を観測し，何を予測し，何を最適化するのかを明確にしなければならない．直感的には，入力から出力への対応を学ぶことが目的であるが，その背後では未知のデータ生成分布のもとで予測誤差を小さくするという問題が立てられている．とりわけ教師あり学習では，入力と正解出力が対になった標本が与えられるため，確率的定式化と最適

化問題との対応がもっとも見えやすい．

定義　教師あり学習では，訓練データセットを

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n} \tag{5.1}$$

と書く．ここで，$x_i \in \mathcal{X}$ は第 i 標本の入力，$y_i \in \mathcal{Y}$ は対応する出力，n は標本数，$\mathcal{X}$ と $\mathcal{Y}$ はそれぞれ入力空間と出力空間である．通常は，各対 (x_i, y_i) が未知の同時分布 $p_{\mathrm{data}}(x, y)$ から独立同分布に従って生成されたと仮定する．この仮定は理想化ではあるが，訓練データと将来のデータが同種の母集団から来るという汎化の前提を数学的に表現している．

学習器はパラメータ (parameter)θ を持つモデル f_θ として表される．$f_\theta(x)$ は入力 x に対する予測値であり，その形は問題設定に応じて異なる．回帰 (regression) では $\mathcal{Y}$ は実数または実ベクトルであり，$f_\theta(x)$ は平均値や連続量の予測を返す．分類 (classification) では $\mathcal{Y}$ は有限個のクラス集合であり，$f_\theta(x)$ はクラス確率やロジット (logit) を返すことが多い．いずれにせよ，θ はモデルの自由度を定める未知量であり，学習とは θ をデータから推定することである．

実際の学習では，データは訓練集合，検証集合，テスト集合に分割される．訓練集合はパラメータ更新に用いられ，検証集合はハイパーパラメータの選択や早期終了の判断に用いられ，テスト集合は最終的な汎化性能の評価に用いられる．この分割が必要なのは，同じデータでモデルの選択と性能評価を同時に行うと，評価が楽観的に歪むからである．とくに深層学習では，アーキテクチャ，学習率，正則化強度など多くの設計選択があるため，検証集合の役割は本質的である．

教師あり学習の中心概念は経験リスク最小化である．損失関数 (loss function) を $\ell(y, f_\theta(x))$ と書くと，経験リスクは

$$\hat{R}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) \tag{5.2}$$

で定義される．ここで ℓ は予測 $f_\theta(x_i)$ が正解 y_i からどれだけ外れているかを測る関数であり，$\hat{R}_n(\theta)$ は訓練データ上の平均誤差である．他方，真に最小化したい量は未知分布に関する期待損失

$$R(\theta) = \mathrm{E}_{(x,y)\sim p_{\mathrm{data}}}\left[\ell(y, f_\theta(x))\right] \tag{5.3}$$

であり，これを期待リスクまたは汎化誤差と呼ぶ．経験リスク最小化は，観測できない $R(\theta)$ を，その標本平均である $\hat{R}_n(\theta)$ に置き換えて最適化する考え方である．

このとき重要なのは，モデルが単に点予測関数としてではなく，条件付き分布 $p(y|x)$ の近似として理解できることである．すなわち，$f_\theta(x)$ はしばしば直接の予測値ではなく，条件付き分布 $p_\theta(y|x)$ を指定するためのパラメータを返す．たとえば回帰では，$f_\theta(x)$ を条件付き平均 $\mu_\theta(x)$ とみなして $p_\theta(y|x)$ をガウス分布で表すことが多い．分類では，$f_\theta(x)$ がクラスごとのロジットを返し，それを softmax に通してクラス確率 $p_\theta(y = k|x)$ を得る．この対応により，損失関数は確率モデルの負の対数尤度として自然に導かれる．

さらに，損失関数と条件付き分布の関係を理解すると，最適予測器の意味も明確になる．たとえば二乗損失のもとでは，各入力 x に対する最適予測は条件付き期待値 $\mathrm{E}[Y|X = x]$ である．分類で 0-1 損失を考えれば，最適予測は事後確率 $p(y|x)$ が最大のクラスを返す規則になる．したがって，

機械学習の目標は単なるラベルの記憶ではなく，条件付き分布の構造を十分によく近似し，適切な決定規則へ変換することである．

以上の定式化により，問題は「どの損失関数を選び，どのような確率モデルを仮定するか」に帰着する．回帰と分類が異なるのは出力空間の性質だけではなく，仮定する確率モデルと対応する損失が異なるためである．次節では，この対応を最尤推定と最大事後確率推定の観点から詳しく導き，損失最小化がどのような最適化問題になるかを整理する．

5.1.3　損失関数と最適化

損失関数を導入する理由は，学習の目的を数理的に定義し，パラメータ推定を最適化問題へ変換するためである．経験的には二乗誤差やクロスエントロピーがよく用いられるが，これらは便宜的に選ばれているわけではない．適切な確率モデルを仮定すると，それらは最尤推定 (maximum likelihood estimation; MLE) や最大事後確率推定 (maximum a posteriori; MAP) の帰結として現れる．したがって，損失の選択は「どの誤差を罰したいか」という感覚的判断だけではなく，「データ生成過程をどのようにモデル化するか」という統計的判断でもある．

定義　最尤推定では，パラメータ θ を，訓練データの条件付き尤度を最大化する量として定める．すなわち

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta} \prod_{i=1}^{n} p_\theta(y_i \mid x_i) = \arg\max_{\theta} \sum_{i=1}^{n} \log p_\theta(y_i \mid x_i) \tag{5.4}$$

である．対数は単調増加であるため，尤度最大化は負の対数尤度の最小化と等価である．以後，負の対数尤度を損失関数として用いると，確率推定と最適化が一つの枠組みに統合される．

回帰において，出力 y_i が平均 $\mu_\theta(x_i)$，分散 σ^2 のガウス分布に従うと仮定しよう．ここで $\mu_\theta(x_i)$ はモデルが入力 x_i から出力する条件付き平均，σ^2 は観測雑音の分散である．このとき

$$p_\theta(y_i \mid x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu_\theta(x_i))^2}{2\sigma^2} \right) \tag{5.5}$$

であるから，負の対数尤度の総和は定数項を除いて

$$\sum_{i=1}^{n} -\log p_\theta(y_i \mid x_i) = \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu_\theta(x_i))^2 + \mathrm{const.} \tag{5.6}$$

となる．したがって，平均二乗誤差 (mean squared error; MSE)

$$\mathcal{L}_{\mathrm{MSE}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu_\theta(x_i))^2 \tag{5.7}$$

は，ガウス雑音仮定のもとでの最尤推定に対応している．ここで n は標本数であり，各誤差を平均することでデータ数に依らない尺度を得ている．

分類では，出力が K 個のクラスのいずれかをとるとする．モデルが入力 x_i に対してクラスごとのロジット $z_k(x_i)$ を返すとき，softmax により

$$\pi_{\theta,k}(x_i) = \frac{\exp(z_k(x_i))}{\sum_{j=1}^{K} \exp(z_j(x_i))} \tag{5.8}$$

を定義すると，$\pi_{\theta,k}(x_i)$ はクラス k の条件付き確率 $p_\theta(y_i = k|x_i)$ と解釈できる．正解ラベルを one-hot ベクトル $y_i = (y_{i1}, \dots, y_{iK})$ で表し，$y_{ik} \in \{0, 1\}$ を「標本 i がクラス k に属するか」を表す指示変数とすると，負の対数尤度は

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log \pi_{\theta,k}(x_i) \tag{5.9}$$

となる．これはクロスエントロピー損失 (cross-entropy loss) であり，真のラベル分布とモデル分布のずれを測る量として理解できる．

MAP 推定では，さらにパラメータの事前分布 $p(\theta)$ を導入し，

$$\hat{\theta}_{\mathrm{MAP}} = \arg\max_{\theta} \left\{ \sum_{i=1}^{n} \log p_\theta(y_i|x_i) + \log p(\theta) \right\} \tag{5.10}$$

を解く．これは負の対数をとれば，データ適合項に加えて $-\log p(\theta)$ という正則化項を課すことに等しい．たとえば $p(\theta) \propto \exp(-\lambda \|\theta\|_2^2/2)$ というガウス事前分布を置けば，$\lambda\|\theta\|_2^2/2$ という L_2 正則化が現れる．同様に，ラプラス事前分布を置けば L_1 正則化が現れる．したがって，正則化は経験的な細工ではなく，パラメータに対する事前信念を導入した統計推定として理解できる．

もっとも，損失関数が定まれば問題が簡単になるわけではない．線形回帰やロジスティック回帰の一部では凸最適化 [5] が得られるが，深いニューラルネットワークでは一般に目的関数は非凸 (nonconvex) である．すなわち，勾配がゼロとなる停留点の周囲には，局所最小点，鞍点 (saddle point)，あるいは曲率が小さい平坦領域が存在しうる．ヘッセ行列を $H(\theta)$ と書けば，停留点で $H(\theta)$ の固有値がすべて正なら局所最小，正負が混在すれば鞍点である．深層学習においては，悪い局所解そのものよりも，高次元空間に多数現れる鞍点や，曲率の偏りによって進行が遅くなる平坦領域のほうが，しばしば実践上の難所となる．

さらに，最小化された訓練損失が小さいことと，汎化性能が高いことは同義ではない．平坦な最小点はしばしば汎化に有利だと経験的に言われるが，その因果関係は単純ではなく，パラメータ化の仕方や最適化アルゴリズムにも依存する．ここで重要なのは，機械学習における最適化は，解析的に唯一解を求める問題ではなく，巨大な非凸目的関数の中から，計算可能で汎化も良好な解を探索する過程だという点である．

この探索を現実的な時間で実行するために，深層学習では勾配法が中心的役割を果たす．とくに大規模データでは，全データに対する勾配を毎回計算することは高価すぎる．次節では，この問題に対する標準的解法である確率的勾配降下法を導入し，その統計的性質と実装上の意味を整理する．

5.1.4 確率的勾配降下法 (SGD)

損失関数が定まった後に必要なのは，その最小値へ向かってパラメータを更新する具体的な計算法である．とくに深層学習では，パラメータ数もデータ数も大きいため，厳密解を求めるよりも反復的な近似最適化が現実的である．このとき，全訓練データを毎回用いる方法は統計的には素

直であっても，計算量とメモリ使用量の点でしばしば実用的でない．確率的勾配降下法 (stochastic gradient descent; SGD) は，この制約のもとで勾配情報を効率よく利用するための基本アルゴリズムであり，その源流は Robbins [24] の確率近似法にさかのぼる．

訓練目的関数を

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta), \qquad \ell_i(\theta) = \ell(y_i, f_\theta(x_i)) \tag{5.11}$$

と書く．ここで $L(\theta)$ は平均損失，$\ell_i(\theta)$ は標本 i に対する個別損失，n は訓練標本数である．バッチ勾配降下法 (batch gradient descent) では，各反復 t における更新は

$$\theta_{t+1} = \theta_t - \eta_t \nabla L(\theta_t) = \theta_t - \eta_t \frac{1}{n}\sum_{i=1}^{n} \nabla \ell_i(\theta_t) \tag{5.12}$$

で与えられる．ここで θ_t は時刻 t のパラメータ，$\eta_t > 0$ は学習率 (learning rate) である．この更新は目的関数の真の勾配を用いるため安定であるが，大規模データでは毎回すべての標本を読む必要がある．

これに対して SGD では，訓練集合からサイズ b のミニバッチ B_t を無作為に選び，

$$g_t = \frac{1}{b}\sum_{i \in B_t} \nabla \ell_i(\theta_t), \qquad \theta_{t+1} = \theta_t - \eta_t g_t \tag{5.13}$$

と更新する．ここで b はバッチサイズ，g_t はミニバッチ勾配である．B_t を一様にサンプリングすれば，条件付き期待値について

$$\mathrm{E}[g_t \mid \theta_t] = \nabla L(\theta_t) \tag{5.14}$$

が成り立つ．すなわち，ミニバッチ勾配は真の勾配の不偏推定量である．この事実が，SGD を単なる近似計算ではなく，統計的に意味のある勾配推定法として位置づけている．

ミニバッチ化の効果を理解するには，勾配の分散に注目するとよい．個別勾配 $\nabla \ell_i(\theta_t)$ がある共分散行列 $\Sigma(\theta_t)$ を持つと近似すれば，復元抽出に基づく単純化のもとで

$$\mathrm{Var}(g_t \mid \theta_t) \approx \frac{1}{b}\Sigma(\theta_t) \tag{5.15}$$

である．したがって，バッチサイズ b を大きくすると勾配推定の分散はおおむね $1/b$ に比例して減少する．これは，大きなバッチでは更新方向が安定し，小さなバッチでは雑音が強くなることを意味する．雑音は収束を遅らせうる一方で，鞍点や鋭い谷から抜け出す助けになる場合もあるため，バッチサイズの選択は統計的効率と探索性の折り合いである．

実装上，ミニバッチ化が重要なのは，第一にメモリ制約のためであり，第二に計算資源の並列利用のためである．GPU や TPU は多数の標本に対する同型演算をまとめて実行することで高いスループットを発揮するが，全データを一度に載せることは通常できない．ミニバッチは，行列演算の効率を確保しつつ，メモリ内で処理可能な単位にデータを分割する役割を持つ．また，オンラインに到着するデータや巨大データセットに対しても，局所的な勾配情報だけで更新を続けられる点で，大規模学習の前提をなしている．

アルゴリズム SGD の基本形は単純である．各エポックでデータをシャッフルし，ミニバッチごとに順伝播，損失計算，勾配計算，パラメータ更新を行う．ここで「エポック」とは訓練集合全体を一巡する単位を指す．実際には，学習率の減衰，勾配クリッピング，重み減衰などを組み合わせることが多いが，中心にあるのは不偏な勾配推定を反復的に用いて目的関数を下げるという構図である．

SGD の派生として Momentum は過去の更新方向を指数移動平均し，振動を抑えながら進行方向を安定化する．Adam は一次モーメントと二次モーメントの推定に基づき，各パラメータに適応的な学習率を割り当てる．これらは実務上きわめて有用であるが，深層学習の根幹がミニバッチ勾配に基づく SGD にあることは変わらない．勾配法が機能する前提は，各反復でパラメータに関する勾配を効率よく計算できることであり，そのための決定的技術が誤差逆伝播法である．次節では，連鎖律に基づく勾配計算の仕組みを導く．

5.1.5 誤差逆伝播法

深いネットワークを学習するには，多数のパラメータそれぞれについて損失の勾配を求める必要がある．もし各パラメータを独立に微分し，そのたびに順方向計算をやり直すなら，計算量は現実的でなくなる．誤差逆伝播法 (backpropagation)[25] が必要なのは，複合関数として構成されたモデルの勾配を，計算グラフと連鎖律を用いて一度の順伝播と一度の逆伝播で効率よく求めるためである．この方法により，パラメータ数が多くても，計算量はグラフの辺数にほぼ線形に比例する．

計算グラフ (computational graph) では，各演算をノード，依存関係を辺として表す．L 層の全結合ネットワークを例にとると，入力を $h^{(0)} = x$ として，各層 $l = 1, \dots, L$ について

$$a^{(l)} = W^{(l)} h^{(l-1)} + b^{(l)}, \qquad h^{(l)} = \phi^{(l)}(a^{(l)}) \tag{5.16}$$

と書ける．ここで $W^{(l)}$ は第 l 層の重み行列，$b^{(l)}$ はバイアスベクトル，$a^{(l)}$ は活性化関数適用前の線形結合，$h^{(l)}$ は層出力，$\phi^{(l)}$ は活性化関数である．教師信号を y，最終損失を $\mathcal{L}(h^{(L)}, y)$ と書けば，求めたいのは各 $W^{(l)}$ と $b^{(l)}$ に関する偏微分である．

定義 逆伝播では，各層の前活性化 $a^{(l)}$ に関する損失の勾配

$$\delta^{(l)} = \frac{\partial \mathcal{L}}{\partial a^{(l)}} \tag{5.17}$$

を誤差信号と呼ぶ．出力層では

$$\delta^{(L)} = \frac{\partial \mathcal{L}}{\partial h^{(L)}} \odot \phi^{(L)\prime}(a^{(L)}) \tag{5.18}$$

であり，$\odot$ は要素ごとの積，$\phi^{(L)\prime}$ は活性化関数の導関数である．中間層では連鎖律により

$$\delta^{(l)} = \left((W^{(l+1)})^{\top} \delta^{(l+1)} \right) \odot \phi^{(l)\prime}(a^{(l)}), \qquad l = L-1, \dots, 1 \tag{5.19}$$

という再帰式が成り立つ．すなわち，上位層から伝わってきた誤差を重み行列の転置で引き戻し，その層の局所微分で修正することで，下位層の勾配情報が得られる．

この誤差信号が得られれば，各パラメータに関する勾配は直ちに計算できる．単一標本に対しては

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \delta^{(l)} (h^{(l-1)})^{\top}, \qquad \frac{\partial \mathcal{L}}{\partial b^{(l)}} = \delta^{(l)} \tag{5.20}$$

である．ミニバッチ学習では，各標本に対するこれらの勾配を平均したものがバッチ勾配となる．ここで $h^{(l-1)}$ が必要なのは，重み行列 $W^{(l)}$ が前層出力と線形に結びついているからであり，誤差信号 $\delta^{(l)}$ と入力 $h^{(l-1)}$ の外積として勾配が現れる．

順伝播時の中間値を保存する必要があるのも，この式から理解できる．逆伝播では $\phi^{(l)\prime}(a^{(l)})$ や $h^{(l-1)}$ が必要になるため，前活性化 $a^{(l)}$ と層出力 $h^{(l)}$ を保持しておかないと，逆向き計算のたびに順方向を再計算しなければならない．したがって，誤差逆伝播法は計算量を節約する代わりに，中間活性を記憶するメモリを要求する．この時間と空間の交換は，深いモデルや長い系列を扱う際にとくに重要である．

逆伝播が効率的である理由は，逆向きモード自動微分 (reverse-mode automatic differentiation) と同じ原理を用いているからである．出力がスカラー損失一つであるとき，各ノードから損失への感度を一度だけ伝播すれば，すべてのパラメータの勾配が得られる．これに対して前向きモードで各パラメータごとに微分を流すと，パラメータ数に比例する回数の計算が必要になる．深層学習では通常，出力次元よりもパラメータ数のほうがはるかに大きいため，逆伝播の利点は決定的である．

もっとも，深いネットワークでは勾配そのものが安定に伝わらないことがある．層ごとのヤコビアンを $J^{(l)}$ と書けば，下位層への勾配は概ねそれらの積

$$\frac{\partial \mathcal{L}}{\partial h^{(l)}} \propto J^{(l+1)} J^{(l+2)} \cdots J^{(L)} \tag{5.21}$$

に支配される．この積のノルムが繰り返し 1 より小さければ勾配消失 (vanishing gradient)，大きければ勾配爆発 (exploding gradient) が起こる．深層化と長系列化に伴ってこの問題は顕著になり，活性化関数，初期化，正規化，残差接続，ゲーティングなどの工夫が必要になる．

このように，誤差逆伝播法は深層学習を可能にした計算技術であるが，その効果はネットワークの構造と強く結びついている．次節では，そもそもニューラルネットワークがどのような関数族を表し，なぜ非線形性が不可欠なのかを整理する．

5.1.6　ニューラルネットワークの基本構造

ニューラルネットワークを学ぶ必要があるのは，深層学習におけるほとんどすべてのモデルが，その変形または拡張として理解できるからである．CNN，RNN，Transformer はいずれも特殊な構造を持つが，その基本単位は線形変換と非線形活性化の反復にある．したがって，まずパーセプトロン，多層パーセプトロン，活性化関数，出力層の意味を押さえることが，後続の複雑なアーキテクチャを理解する前提となる．

パーセプトロン (perceptron) は，もっとも基本的な線形判別器である．入力 $x \in \mathcal{X}$ をベクトルとし，重みベクトルを w，バイアスを b とすると，その決定関数は

$$\hat{y} = \operatorname{sign}(w^{\top} x + b) \tag{5.22}$$

と書ける．ここで $w^\top x$ は入力と重みの内積，sign は符号関数であり，超平面による二値分類を表す．パーセプトロンは線形分離可能な問題には有効であるが，入力空間における複雑な非線形境界を直接表すことはできない．この限界を超えるために，中間表現を導入した多層パーセプトロン (multilayer perceptron; MLP) が用いられる．

MLP では，各層が

$$h^{(l)} = \phi^{(l)}(W^{(l)} h^{(l-1)} + b^{(l)}) \tag{5.23}$$

という形で定義される．ここで $h^{(0)} = x$ は入力，$h^{(l)}$ は第 l 層の隠れ表現，$W^{(l)}$ と $b^{(l)}$ は学習されるパラメータ，$\phi^{(l)}$ は活性化関数である．最終層の出力は課題に応じて変わる．回帰では線形出力をそのまま使うことが多く，二値分類では sigmoid 関数で確率に写し，多クラス分類では softmax を用いてクラス確率へ変換する．

非線形性が必要であることは，数式から直ちに分かる．もしすべての活性化関数が恒等写像，すなわち $\phi^{(l)}(u) = u$ であるなら，二層以上のネットワークでも

$$h^{(L)} = W_{\mathrm{eff}} x + b_{\mathrm{eff}} \tag{5.24}$$

という単一のアフィン変換にまとめられる．ここで W_{eff} は各層の重み行列の積，b_{eff} はバイアス項をまとめたベクトルである．したがって，線形変換を何層重ねても，表現力は本質的に一層の線形モデルを超えない．非線形活性化は，関数クラスを真に拡張し，高次の特徴や複雑な決定境界を表現するために不可欠である．

活性化関数にはそれぞれ役割と副作用がある．sigmoid 関数

$$\sigma(u) = \frac{1}{1 + \exp(-u)} \tag{5.25}$$

は出力を $(0, 1)$ に収めるため確率解釈と相性がよいが，大きな正負の入力で勾配が小さくなりやすい．双曲線正接関数 $\tanh(u)$ は出力が $(-1, 1)$ に中心化されるため学習しやすい場面があるが，やはり飽和による勾配消失を起こしうる．ReLU は

$$\mathrm{ReLU}(u) = \max(0, u) \tag{5.26}$$

で定義され，正の領域で勾配が保たれるため深層学習を大きく前進させた．近年の Transformer では，標準正規分布の累積分布関数を Φ として $\mathrm{GELU}(u) = u\Phi(u)$ と定義される GELU がよく用いられる．これは入力を滑らかにゲートする働きを持ち，深いモデルの最適化と相性がよい．

普遍近似定理 (universal approximation theorem) は，適切な非線形活性化を持つ十分広い一隠れ層ネットワークが，コンパクト集合上の連続関数を任意精度で近似できることを述べる．この結果はニューラルネットワークの表現力を保証するが，それだけで深いネットワークの優位性が自動的に示されるわけではない．実際には，同じ関数を浅いネットワークで表すには非常に大きな幅が必要になることがあり，深さは合成構造を持つ関数を効率よく表す手段として重要である．すなわち，深いモデルの意義は「表現できるか否か」だけでなく，「どれだけ少ないパラメータで，どれだけ学習しやすく表現できるか」にある．

以上より，ニューラルネットワークの基本単位は，線形写像，非線形活性化，目的に応じた出力層から成ることが分かる．しかし高い表現力は同時に過学習の危険を伴う．次節では，訓練誤差の最小化と未知データへの汎化との緊張関係を整理し，正則化の役割を考える．

5.1.7　正則化と汎化

機械学習で本当に必要なのは，訓練データによく合うモデルではなく，未知データに対しても安定に機能するモデルである．正則化 (regularization) を学ぶ理由は，モデルの表現力が高くなるほど，訓練誤差を小さくするだけでは不十分になるからである．とくに深層学習では，過剰な自由度が過学習を招く一方で，十分な表現力がなければ複雑な分布を捉えられない．正則化はこの緊張関係を制御し，汎化性能を改善するための原理である．

過学習 (overfitting) とは，モデルが訓練集合に特有の偶然的変動や雑音まで取り込んでしまい，訓練誤差は小さいにもかかわらずテスト誤差が大きくなる現象をいう．逆に，モデルが単純すぎて訓練誤差すら十分に下がらない状態をアンダーフィット (underfitting) という．前者ではモデル容量がデータ量に比して大きすぎ，後者では表現力が不足している．したがって，良い学習とは，単に大きなモデルを選ぶことでも，小さなモデルを選ぶことでもなく，課題に見合った容量制御を行うことである．

この関係を理解する一つの手がかりが，二乗損失におけるバイアス・バリアンス分解 (bias-variance decomposition) である．真の回帰関数を $f^\star(x) = \mathrm{E}[Y \mid X = x]$，学習データ $\mathcal{D}$ から得られる推定器を $\hat{f}_{\mathcal{D}}(x)$，観測雑音の分散を $\sigma_\varepsilon^2 = \mathrm{Var}(Y \mid X = x)$ とすると，固定した入力 x における予測誤差の期待値は

$$\mathrm{E}_{\mathcal{D},Y}\left[(Y - \hat{f}_{\mathcal{D}}(x))^2\right] = \sigma_\varepsilon^2 + \left(\mathrm{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)] - f^\star(x)\right)^2 + \mathrm{E}_{\mathcal{D}}\left[\left(\hat{f}_{\mathcal{D}}(x) - \mathrm{E}_{\mathcal{D}}[\hat{f}_{\mathcal{D}}(x)]\right)^2\right] \tag{5.27}$$

と書ける．右辺の第一項は不可避な雑音，第二項は系統的なずれであるバイアス，第三項は学習データの揺らぎに対する感度であるバリアンスを表す．一般に，単純なモデルは高バイアス・低バリアンス，複雑なモデルは低バイアス・高バリアンスの傾向を持つ．

もっとも基本的な正則化は，目的関数にパラメータノルムの罰則を加える方法である．L_2 正則化は $\lambda \|\theta\|_2^2 / 2$ を，L_1 正則化は $\lambda \|\theta\|_1$ を損失に加える．ここで $\lambda > 0$ は正則化強度であり，$\|\theta\|_2$ はユークリッドノルム，$\|\theta\|_1$ は各成分の絶対値和である．前節で見たように，これらはそれぞれガウス事前分布とラプラス事前分布を課した MAP 推定として解釈できる．L_2 はパラメータを滑らかに縮小し，L_1 は疎な解を促しやすい．いずれも，無制約に複雑な関数を選ぶことを抑える容量制御とみなせる．

深層学習では，明示的な罰則以外にも多くの正則化手段が使われる．Dropout[31] では，訓練時に各隠れユニットを確率的に無効化し，ネットワークが特定の経路に過度に依存することを防ぐ．記号的には，隠れ表現 h に対してベルヌーイマスク m を掛けた $\tilde{h} = m \odot h$ を用いる操作とみなせる．Early Stopping は，検証誤差が悪化し始める前に訓練を止める方法であり，反復最適化それ自体を暗黙の正則化として利用する．Batch Normalization[14] は，ミニバッチ平均 μ_B と分散 σ_B^2 を用いて

$$\hat{a} = \frac{a - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \qquad y = \gamma \hat{a} + \beta \tag{5.28}$$

と正規化する．ここで a は正規化前の活性，ϵ は数値安定化のための小定数，γ, β は学習可能なスケールとシフトである．これは主として最適化を安定化する技法であるが，ミニバッチ統計量の揺らぎを通じて一定の正則化効果も持つ．

Data Augmentation は，入力にラベル不変な変換を加えることで，モデルに望ましい不変性を学習させる方法である．画像であれば平行移動，反転，切り出し，色変換などが典型であり，系列や言語でもノイズ付与やマスク化などが利用される．これは，単に標本数を人工的に増やす手法ではなく，「この変換では意味が変わらない」という事前知識を学習問題に埋め込む操作と解釈できる．その意味で，アーキテクチャ設計と同様に，分布に関する帰納バイアス (inductive bias) を与える手段である．

近年は，モデルサイズを増やすとテスト誤差が一度悪化した後で再び改善する double descent も観測されている．これは古典的な「モデルが複雑になるほど汎化は単調に悪化する」という単純図式が十分でないことを示している．しかし，この現象もまた，補間可能な過剰パラメータ領域であっても，どの解が最適化によって選ばれるかが重要であることを示すにすぎない．すなわち，汎化は依然として，データ，モデル容量，最適化，正則化の相互作用として理解すべき問題である．

ここまでの議論は，主として全結合ネットワークを念頭に置いていた．しかし現実のデータには，画像の局所性や系列の時間依存のような明確な構造が存在する．こうした構造を活用することで，表現効率と汎化性能を同時に高めることができる．次節では，画像に対する代表的な帰納バイアスを実装した CNN を扱う．

5.1.8 CNN

画像に対して全結合層だけを用いることが非効率である理由は，画像が高次元でありながら，局所的で平行移動に対して類似した構造を持つからである．たとえば，高さ H，幅 W，チャネル数 C の画像を一つのベクトルに平坦化して全結合層へ入力すると，隠れ層のユニット数に比例して巨大な数のパラメータが必要になる．しかも，この表現では近くにある画素と遠くにある画素の区別が消え，画像の空間構造が活かされない．CNN は，この問題を局所結合と重み共有によって解決する．

畳み込み層では, 各出力ユニットが入力全体ではなく局所領域だけを見る．これを受容野 (receptive field) と呼ぶ．さらに，入力の異なる位置で同じカーネルを使い回すことで重み共有を行う．二次元入力 X とカーネル K に対する離散畳み込みを，出力位置 (i, j) で

$$(X * K)_{i,j} = \sum_{u=0}^{r-1} \sum_{v=0}^{s-1} K_{u,v} X_{i+u, j+v} \tag{5.29}$$

と定義する．ここで r, s はカーネルの高さと幅，u, v はカーネル内部の添字，i, j は出力位置を表す．実際の画像ではチャネル方向の和も加わるが，本質は局所領域に対して同一の線形演算を空間全体で適用する点にある．

この構造は二つの利点をもたらす．第一に，パラメータ数が大幅に削減される．全結合層では入力次元に比例して重み数が増えるのに対し，畳み込み層ではカーネルサイズに比例するだけであ

る．第二に，平行移動等変性 (translation equivariance) が得られる．平行移動演算を T_Δ，畳み込み演算を $\mathcal{C}$ と書けば，

$$\mathcal{C}(T_\Delta X) = T_\Delta(\mathcal{C}(X)) \tag{5.30}$$

が成り立つ．すなわち，入力をずらせば出力も同じだけずれる．この性質により，同じ局所パターンを画像中のどこでも検出できる．

プーリング (pooling) は，局所領域の代表値をとることで空間解像度を下げ，わずかな位置ずれに対する不変性を高める操作である．最大値をとる max pooling と平均をとる average pooling が典型である．畳み込みとプーリングを重ねると，浅い層ではエッジや局所テクスチャ，深い層ではより抽象的な形状や部品が表現されるようになる．また，層を重ねるにつれて受容野が広がるため，局所情報から大域的文脈へと表現が統合されていく．

CNN の歴史的発展も，構造仮定と計算資源の相互作用として理解できる．LeNet[19] は文字認識で畳み込みの有効性を示した初期例であり，AlexNet[17] は GPU と ReLU を活用して ImageNet で深層 CNN の優位を広く認知させた．VGG は小さなカーネルを多層に積む設計で深さの効果を明確にし，ResNet[10] は

$$h^{(l+1)} = h^{(l)} + F(h^{(l)}) \tag{5.31}$$

という残差接続によって，より深いネットワークの最適化を可能にした．ここで F は残差写像であり，恒等写像を近道として残すことで勾配伝播を助ける．

もっとも，CNN の帰納バイアスは画像のような格子状データには適していても，系列のように順序依存が本質となる問題にはそのままでは十分でない．時系列，文章，音声のようなデータでは，過去の情報をどのように状態として保持し，将来の予測に反映させるかが中心問題となる．次節では，そのために導入された RNN を扱う．

5.1.9　RNN

系列データを扱うためには，観測が独立ではなく，時間や位置に沿って依存していることを明示的に表現しなければならない．文章，音声，交通流時系列などでは，現在の観測や予測が過去の文脈に依存するため，入力を単に固定長ベクトルとして扱うだけでは不十分である．RNN が必要なのは，系列の結合分布を連鎖律に従って分解し，その文脈を隠れ状態として逐次的に要約できるからである．

長さ T の系列 $x_{1:T} = (x_1, \dots, x_T)$ の結合分布は，確率の連鎖律により

$$p(x_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_{1:t-1}) \tag{5.32}$$

と書ける．ここで $x_{1:t-1}$ は時刻 1 から $t-1$ までの過去系列である．条件付き生成や系列ラベリングでも同様に，現在の出力は過去の入力や過去の出力に依存する．この依存構造を有限次元の状態で近似するため，RNN は各時刻 t に隠れ状態 h_t を持ち，

$$h_t = \phi(W_{xh} x_t + W_{hh} h_{t-1} + b_h) \tag{5.33}$$

と更新する．ここで x_t は時刻 t の入力，h_t はその時点までの文脈を要約した隠れ状態，W_{xh} と W_{hh} は入力と前時刻状態に対する重み行列，b_h はバイアス，ϕ は活性化関数である．出力 o_t は通常

$$o_t = W_{hy} h_t + b_y \tag{5.34}$$

とし，必要に応じて softmax などへ通す．

この再帰式の直感は明快である．各時刻で新しい観測 x_t を取り込み，それ以前の要約 h_{t-1} と結合して新しい要約 h_t を作るのである．したがって RNN は，可変長系列を固定次元の状態遷移として扱う動的システムとみなせる．重み行列が時刻をまたいで共有されるため，系列長が変わっても同じモデルを適用できる点も重要である．

学習は時間方向に展開した計算グラフに対する誤差逆伝播，すなわち BPTT(backpropagation through time)[38] によって行う．時刻ごとの損失を $\mathcal{L}_t$ とすると，総損失 $\mathcal{L} = \sum_t \mathcal{L}_t$ に対するある時刻 t の状態勾配は

$$\frac{\partial \mathcal{L}}{\partial h_t} = \frac{\partial \mathcal{L}_t}{\partial h_t} + \left(\frac{\partial h_{t+1}}{\partial h_t} \right)^{\top} \frac{\partial \mathcal{L}}{\partial h_{t+1}} \tag{5.35}$$

という再帰式に従う．ここで $\partial h_{t+1}/\partial h_t$ は時刻をまたぐヤコビアンであり，長い系列ではそれらの積が勾配を支配する．このため，固有値の大きさや活性化関数の導関数によって，勾配消失や勾配爆発が生じやすい．

LSTM(long short-term memory)[13] や GRU(gated recurrent unit)[6] が必要になったのは，この問題を緩和するためである．たとえば LSTM では，セル状態 c_t を

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \qquad h_t = o_t \odot \tanh(c_t) \tag{5.36}$$

と更新する．ここで f_t, i_t, o_t はそれぞれ忘却ゲート，入力ゲート，出力ゲート，$\tilde{c}_t$ は候補セル状態であり，いずれも時刻 t の入力と前時刻状態から計算される．重要なのは，セル状態が加法的な経路を持つため，勾配が長距離にわたって保たれやすいことである．GRU も同様に，より簡潔なゲーティングによって情報保持と更新を制御する．

RNN は長らく系列モデリングの中心にあったが，固定長の隠れ状態だけで長い文脈を圧縮することには限界がある．Encoder-Decoder 構造は，入力系列を文脈ベクトルに写し，それをもとに出力系列を生成することで機械翻訳などに成功したが，長文では情報ボトルネックが顕著になった．これを緩和したのが Attention[2] であり，各出力時刻が入力系列全体の関連部分を直接参照できるようにした．この発想がさらに発展し，再帰そのものを不要にしたのが Transformer[36] である．次節では，その構造を詳しく見る．

5.1.10 Transformer

Transformer を学ぶ必要があるのは，現代の大規模言語モデルや多くの生成モデルの中核がこの構造にあるからである．RNN は系列依存を表現できるが，計算が逐次的で並列化しにくく，遠く離れた要素間の依存を長い経路で伝えなければならない．この逐次性と長距離依存性の限界を克服するために，Transformer は系列内の要素同士を直接結び付ける Self-Attention を採用した．これにより，系列長に対する計算構造と表現の仕方が根本的に変わった．

長さ T の系列を，各トークンの埋め込みを並べた $T \times d$ の実数行列 H で表す．ここで T は系列長，d は埋め込み次元である．Self-Attention では，H からクエリ Q，キー K，バリュー V を

$$Q = HW^Q, \qquad K = HW^K, \qquad V = HW^V \tag{5.37}$$

により作る．W^Q, W^K, W^V は学習される重み行列である．各行はそれぞれ一つの位置に対応し，ある位置が他の位置をどれだけ参照すべきかを，クエリとキーの内積で決める．Scaled Dot-Product Attention は

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{5.38}$$

で定義される．ここで d_k はクエリとキーの次元，softmax は各行ごとに適用されて注意重みを確率分布に変換する．$\sqrt{d_k}$ で割るのは，内積の分散が大きくなりすぎて softmax が飽和するのを防ぐためである．

この式の意味は，各位置が他のすべての位置に対して重み付き平均をとり，新しい表現を得るということである．RNN では時刻 t の情報が時刻 $t+1, t+2, \ldots$ を経由して遠方へ伝わるのに対し，Self-Attention では任意の二位置間が一回の演算で直接相互作用できる．したがって，長距離依存性に対する経路長は理論的に短くなり，勾配伝播も容易になる．また，各位置の計算を同時に行えるため，訓練時の並列化効率が高い．

Multi-Head Attention は，この相互作用を単一の重み付けに限定せず，異なる関係を並列に学習するための仕組みである．ヘッド数を M とすると，第 m ヘッドは

$$\mathrm{head}_m = \mathrm{Attention}(HW_m^Q, HW_m^K, HW_m^V) \tag{5.39}$$

で与えられ，全体の出力は

$$\mathrm{MHA}(H) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_M)W^O \tag{5.40}$$

となる．ここで W_m^Q, W_m^K, W_m^V は第 m ヘッド専用の射影行列，W^O は結合後の出力射影，Concat は特徴次元方向の連結を表す．複数ヘッドを用いることで，あるヘッドは局所的対応，別のヘッドは長距離依存，さらに別のヘッドは構文的関係といった異なるパターンを同時に捉えうる．

Transformer には再帰がないため，位置情報を別途与えなければ系列の順序が失われる．これを補うのが Position Encoding である．代表的な正弦波型位置符号化では，位置 t，次元添字 i に対して

$$p_{t,2i} = \sin\left(\frac{t}{10000^{2i/d}}\right), \qquad p_{t,2i+1} = \cos\left(\frac{t}{10000^{2i/d}}\right) \tag{5.41}$$

を定め，入力埋め込みに加える．これにより，モデルは相対的・絶対的な位置関係を連続的な表現として受け取ることができる．実際には学習可能な位置埋め込みが使われることも多いが，要点は「順序を表現しない自己注意機構に，系列順序の手がかりを与える」ことである．

RNN と比較した Transformer の利点は，並列化しやすく，長距離依存を扱いやすい点にある．他方で，全位置対の相互作用を計算するため，Self-Attention の計算量とメモリ量は系列長 T に対して概ね $O(T^2)$ で増大する．このため非常に長い系列では効率上の課題が残り，疎注意や線形注意

など多くの改良が提案されている．それでも，自然言語処理を中心に Transformer が支配的となったのは，この計算コストを払ってなお得られる表現力と学習効率の利得が大きかったからである．

自己回帰モデル (autoregressive model) との関係も重要である．デコーダ型 Transformer では，未来のトークンを見ないように因果マスク (causal mask) を課し，

$$p_\theta(x_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{1:t-1}) \tag{5.42}$$

という系列分布を学習する．ここで各条件付き分布は，Self-Attention を通じて過去文脈全体に基づいて計算される．この枠組みを巨大データ上で事前学習したものが，大規模言語モデル (large language model; LLM) の基本形である．したがって Transformer は，本章の終点であると同時に，後に学ぶ生成モデルや強化学習と結び付く出発点でもある．

まとめ

本章では，機械学習を「確率分布の学習と最適化の問題」として捉える立場から，基礎概念を順に整理した．まず歴史的には，パターン認識から統計的学習，ニューラルネットワーク，深層学習へと進む流れを，表現力，計算資源，データ量，最適化という観点で読み直した．次に，教師あり学習の基本設定として，データ集合，モデル，パラメータ，経験リスク，汎化誤差，条件付き分布の対応を定式化した．そこから，損失関数が最尤推定や MAP 推定から自然に導かれること，深層学習の最適化が一般に非凸であることを確認した．

さらに，実際の学習を支える計算原理として，ミニバッチに基づく SGD と，連鎖律に基づく誤差逆伝播法を導入した．これにより，大規模データと多数のパラメータを持つモデルを現実的に学習できる理由が明確になった．そのうえで，ニューラルネットワークの基本構造を，線形変換，非線形活性化，出力層の組合せとして整理し，過学習と汎化の問題に対して正則化が果たす役割を考えた．最後に，画像の局所性を活かす CNN，系列の時間依存を表す RNN，長距離依存と並列化を両立する Transformer を順に見たことで，アーキテクチャとはデータ構造に応じた帰納バイアスの実装であることが理解できたはずである．これらの基礎は，強化学習や生成モデルへ進む際にも，その背後にある確率的定式化と最適化の考え方を見失わないための足場となる．

コラム：深層学習による交通流予測モデルの検討 – 李康民

研究の背景と目的

本研究では，高速道路の交通流データを用いて，将来の交通状態を予測するモデルを検討した．目的は大きく二つである．第一に，単純な Neural Network である Multi-Layer Perceptron（MLP）を用いて，交通量 Q，速度 V，交通密度 K の短期予測を行うことである．第二に，MLP を Moving Average および LWR 方程式に基づくモデルと比較し，どのようなデータ条件で Neural Network が有効に働くのかを確認することである．

ここで特に注目したのは，NN の性能を高めるためにはモデル構造だけでなく，入力データの性質や前処理がどの程度重要になるのか，という点である．

使用データと予測タスクの設定

使用したデータは，2024 年 1 月の東名高速道路における 5 分間隔の交通流データである．データには，時刻，KP，lane，交通量 Q，速度 V，交通密度 K が含まれている．まず時空間分布を確認すると，交通量 Q には明確な日周期性が見られた．夜間には交通量が低下し，日中に再び増加するというパターンが繰り返されている．一方，速度 V は多くの時間帯で比較的安定しており，全体としては周期性と短期的安定性が強いデータであると考えられる．

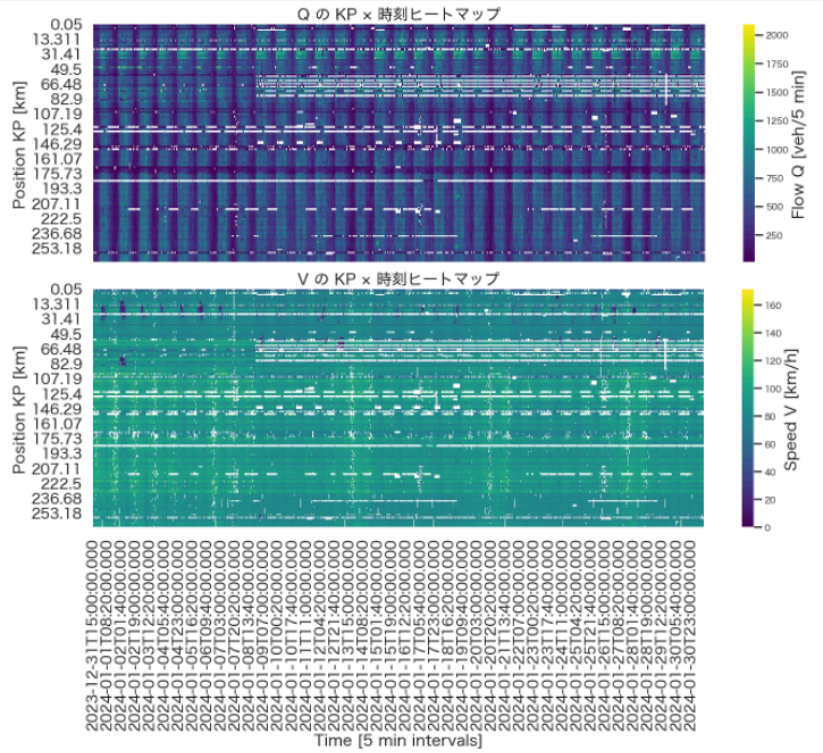


図 5.1: 交通量 Q と速度 V の時空間分布

lane ごとの集計では，交通量や速度の絶対値に大きな差が見られた．特に一部の lane では

交通量のピークが大きく，一方で別の lane では交通量が小さく，欠損率も高かった．欠損率の高い lane をそのまま学習に利用すると，モデルが本来の交通状態ではなく，データの欠損構造を学習してしまう恐れがある．

予測タスクとしては，Lookback を 12，Horizon を 12 に設定し，過去 60 分のデータから未来 60 分の交通状態を予測する問題を基本ケースとした．データは Train，Validation，Test に 0.6，0.2，0.2 の割合で分割し，Epoch は 12 回として MLP を学習した．

Baseline におけるモデル比較

表 5.1: 全 lane を用いた Baseline 条件での予測性能比較

Model	RMSE(Q)	MAE(Q)	RMSE(V)	MAE(V)	RMSE(K)	MAE(K)	F1
Moving Average	102.16	45.23	5.60	2.06	177.85	1.06	0.920
Simple MLP	138.13	89.58	8.61	4.59	256.16	2.17	0.129
LWR Godunov	350.83	194.02	60.70	43.41	478.87	19.96	0.166

上記の Baseline 条件における予測性能比較より，Moving Average が最も高い性能を示した．特に Q と V の RMSE，MAE は MLP よりも小さく，Congestion F1 も高い値となった．これは，今回の交通流データでは「現在の交通状態がしばらく継続する」という単純な仮定が非常に有効だったことを意味する．交通量には日周期性が強く，速度も短時間では大きく変化しにくいため，Moving Average のような単純な baseline でも十分に高い性能を示したと考えられる．

一方で，MLP は基本的な周期パターンを学習できているものの，Moving Average を上回ることはできなかった．LWR モデルは交通流理論に基づくモデルであるが，今回のデータでは欠損や観測点ごとのばらつきの影響を受け，誤差が大きくなった．

MLP による予測結果の特徴

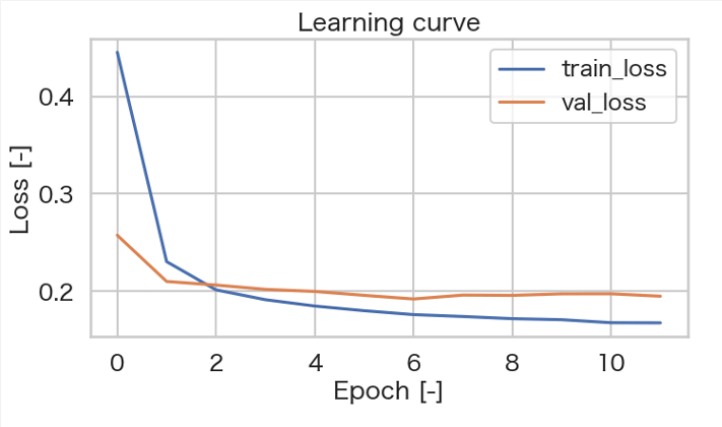


図 5.2: MLP の Learning Curve

MLP の Learning Curve を見ると，Epoch 0 から 2 付近で loss が急激に低下している．これは，MLP が交通データに含まれる基本的な周期性や平均的な傾向を早い段階で学習していることを示している．一方で，その後の validation loss の改善は小さく，学習を続けても大

きな性能向上は見られなかった．

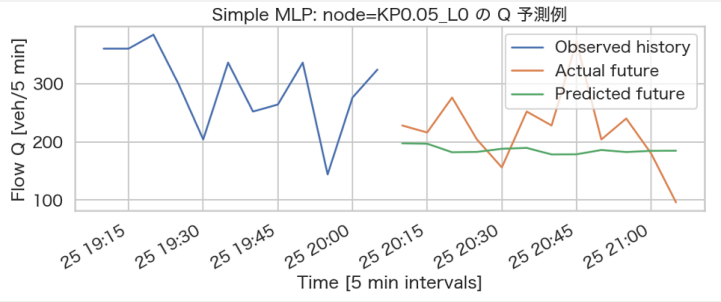


図 5.3: Simple MLP による交通量 Q の予測

また，予測例を見ると，MLP の予測値は実測値の細かな上下変動を追いかけるというよりも，平均的な値に近づく傾向があった．これは，MSE を最小化する学習では，不確実な変動を無理に予測するよりも，平均的な値を出す方が損失を小さくしやすいためである．特に MSE を損失関数として用いた場合，予測が困難な急激変動を無理に追従するよりも，平均的な値へ収束する方が損失を安定して小さくできるため，結果として平坦な予測が生成されやすい．今回の MLP は安定して学習していたが，突発的な変動や局所的な非線形パターンを十分に再現するには限界があった．

欠損率の高い lane を除外した場合

次に，欠損率の高い lane を除外して同じ予測タスクを行った．これは，欠損の多いデータが MLP の学習にノイズとして作用しているかを確認するためである．

表 5.2: 欠損率の高い lane を除外した場合の予測性能比較

Model	RMSE(Q)	MAE(Q)	RMSE(V)	MAE(V)	RMSE(K)	MAE(K)	F1
Simple MLP	119.27	75.57	7.84	4.31	228.56	1.82	0.371
Moving Average	131.91	76.69	7.27	3.60	210.24	1.68	0.751
LWR Godunov	429.36	313.80	65.25	50.68	662.99	38.50	0.314

欠損率の高い lane を除外した場合の予測性能比較を見ると，Simple MLP の Q に対する RMSE と MAE が Baseline より改善している．これは，欠損率の高い lane を除くことで，MLP がより安定した交通パターンを学習できるようになったためである．一方で，Congestion F1 では Moving Average の方が高く，渋滞状態の判定では単純な平均ベースの手法が依然として強い結果となった．この結果から，NN の性能改善には，モデル構造の変更だけでなく，入力データの品質管理が重要であることが分かる．

Lookback の変更と時間依存性

さらに，Lookback を 12 から 6 に短くし，過去 30 分のデータから未来 60 分を予測する設定を試した．これは，MLP が長い過去履歴を利用しているのか，それとも直近の交通状態に主に依存しているのかを確認するためである．

表 5.3: Lookback を 6 に変更した場合の予測性能比較

Model	RMSE(Q)	MAE(Q)	RMSE(V)	MAE(V)	RMSE(K)	MAE(K)	F1
Simple MLP	113.33	73.03	7.65	4.24	799.57	2.95	0.408
Moving Average	120.66	69.66	7.09	3.50	209.86	1.59	0.766
LWR Godunov	429.19	313.71	65.24	50.68	662.88	38.50	0.314

Lookback を 6 に変更した場合の予測性能比較を見ると，MLP の Q に対する誤差は大きく悪化せず，むしろ Baseline より改善した．このことから，今回の MLP は長い過去履歴よりも，直近の交通状態や時間帯ごとの平均的なパターンに強く依存して予測していたと考えられる．

ただし，K の RMSE は不安定であり，交通密度の予測には外れ値やデータのばらつきが強く影響していた可能性がある．

また，Lookback を短くした背景には，入力履歴を過度に長くすると，MLP が局所的なノイズや短期変動まで過学習してしまう可能性があると考えたためである．一方で，今回の結果では Lookback を変更しても性能差は限定的であり，現在の MLP は長期履歴よりも直近状態へ強く依存している可能性が示唆された．なお，交通流の周期性をより明示的に学習させるためには，24 時間単位などより長い Lookback を設定する方法も考えられるが，計算量やメモリ使用量の増加が課題となる．また今回は実装できなかったが，単純に直前時刻の系列だけを入力するのではなく，同一曜日・同一時間帯における過去データを特徴量として利用する方法も，交通流の周期性を捉える上で有効であると考えられる．

特徴量重要度と MLP の学習傾向

Moving Average は，直近の交通状態をそのまま利用する単純なモデルであるが，今回のデータでは高い予測性能を示した．これは，交通流データに強い短期安定性と周期性が存在しており，「現在の状態がしばらく継続する」という仮定が有効であったためであると考えられる．一方で，Moving Average は急激な交通状態変化や局所的なピーク変動を表現することが難しく，予測結果が平滑化されやすいという特徴も確認された．

SHAP による特徴量重要度を確認すると，速度 V が最も重要な特徴量として利用されていた．速度変化は混雑状態や流量変化を早く反映するため，短期予測において重要な情報になったと考えられる．また，直近 step の特徴量が強く利用されており，MLP が主に短期的な交通状態に依存して予測していることが分かる．これは，Lookback を短くしても性能が大きく変化しなかった結果とも整合的である．

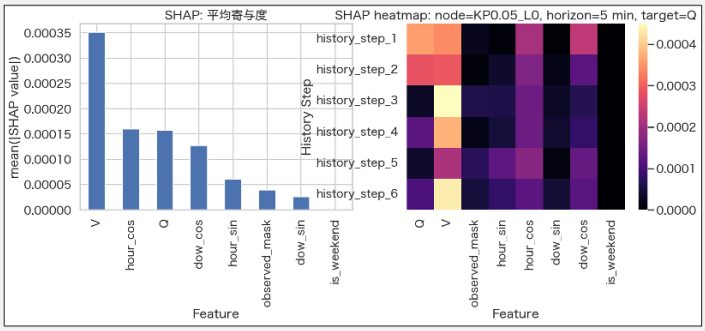


図 5.4: SHAP による特徴量重要度

結びに

本研究では，東名高速道路の交通流データを用いて，MLP，Moving Average，LWR モデルによる短期予測を比較した．その結果，MLP は周期的な傾向を学習できるものの，予測値が平均的で平坦になりやすく，Moving Average を明確に上回るには至らなかった．

また，Lookback を短くしても性能が大きく変化しなかったことや，SHAP で直近 step の重要度が高かったことから，今回の MLP は長期依存関係よりも，直近の交通状態に強く依存して予測していたと考えられる．

また，欠損率の高い lane を除外することで MLP の性能が改善したことから，NN の性能はモデル構造だけでなく，入力データの品質に大きく依存することが確認された．今回のデータでは，交通状態そのものが比較的予測しやすく，複雑な非線形性が強く現れていなかった可能性がある．本研究を通じて，交通流予測においては単純にモデルを複雑化するだけでは十分ではなく，交通データが持つ周期性，欠損構造，lane 構造，短期安定性といった特性を理解した上で，適切な前処理や特徴設計を行うことが重要であることが確認された．今後は，RNN や Transformer のような時系列モデルとの比較に加え，交通流の周期性や lane 間の関係性をより適切に表現できる特徴設計についても検討する必要があると考えられる．

5.2 強化学習

強化学習（reinforcement learning）は，環境との相互作用を通じて意思決定規則を学習し，長期的な累積報酬を最大化するための理論である．教師あり学習では入力と正解の対応が与えられ，教師なし学習では観測データの潜在構造の抽出が中心となるのに対し，強化学習では各時刻の行動に対して正解が即座に与えられるとは限らない．ある行動の評価は，その後に続く状態遷移と報酬列を通じて初めて定まり，その意味で遅延報酬を本質に含んでいる．

本章の目標は，強化学習を単なるアルゴリズムの集合としてではなく，確率的環境のもとで価値関数または方策を学習し，累積報酬を最適化する統一理論として理解することである．標準的な教科書的整理としては Sutton [33] が広く参照されるが，本章ではそれを踏まえつつ，交通・都市工学の逐次意思決定問題にも接続しやすい形で議論を組み立てる．まず歴史的流れから問題意識を整理し，次にマルコフ決定過程によって標準的定式化を与える．そのうえで，多腕バンディット問題によって探索と活用の本質を切り出し，動的計画法，価値ベース手法，方策勾配法へと進む．最後に，標準的な設定だけでは捉えきれない発展的話題を概観する．

5.2.1 強化学習の歴史

強化学習の歴史を学ぶ必要があるのは，この分野が単に高性能な手法を寄せ集めて成立したのではなく，逐次意思決定に固有の困難に対する応答として発展してきたからである．信号制御，車両経路選択，需要応答交通の運行計画のような問題では，現在の行動が将来の混雑や待ち時間を変化させ，評価は複数時刻にまたがって現れる．したがって，入力と正解の対応をそのまま学べばよい教師あり学習とは異なり，行動の価値を試行錯誤から推定しなければならない．

このとき難しさの中心にあるのが，遅延報酬，探索と活用のトレードオフ，そして方策依存のデータ生成である．第一に，最終的な成果にどの行動が寄与したかを識別する信用割当ての問題が生じる．第二に，より良い行動を見つけるためには未知の選択肢を試す探索が必要だが，探索を増やせば当面の報酬は失われる．第三に，データは受動的に与えられるのではなく，学習者自身の行動によって生成される．これらの特徴が，強化学習に独自の理論を要請する．

歴史的源流は，Bellman による動的計画法と最適制御にある [3, 22]．ここで与えられた基本原理は，長期最適化問題を現在の価値と将来の価値の再帰関係として表すベルマン方程式であった．ただし，この枠組みは環境モデル，すなわち状態遷移確率と報酬関数が既知であることを前提としていた．理論的には明快であっても，未知環境から経験を通じて学ぶという強化学習本来の状況には，そのままでは適用できなかった．

この不足を埋めたのが時間差分学習（temporal-difference learning）である．Sutton [32] は，将来の完全な帰結を待たず，次時刻の価値推定を足場にして現在の価値を更新するブートストラップの考え方を定式化した．これにより，モンテカルロ法のようにエピソード終了まで待たなくても逐次的に価値を学習できるようになった．ベルマン方程式は，解析のための式であるだけでなく，学習更新式の源泉になったのである．

さらに，Watkins [37] による Q 学習は，行動価値関数を直接学ぶことで，環境モデルなしに最

適方策へ近づく道を与えた．ここで重要なのは，価値推定と制御の問題が統合されたことである．TD 学習が与えられた方策の価値推定を中心にしていたのに対し，Q 学習は最大演算を通じて最適方策そのものを目指した．

状態空間が高次元化すると，表形式の価値関数では表現能力が不足する．この制約を緩和したのが深層強化学習であり，とくに DQN は画像入力から行動価値を学ぶ代表的成功例となった [20]．AlphaGo 以降は，価値学習，方策学習，探索，計画を組み合わせる深層強化学習が急速に発展した [28]．ただし，この発展は流行紹介として理解するべきではない．本質は，既知モデルから未知モデルへ，表形式から関数近似へ，価値推定から方策最適化へと理論の焦点が拡張された点にある．

同時に，価値ベース手法だけでは連続行動空間や確率的方策の直接最適化に限界があることも明らかになった．Williams [39] に始まる方策勾配法は，方策を直接最適化対象とする考え方を与え，後に Actor-Critic や PPO のような実用的手法へ発展した [26, 27]．この歴史を貫いているのは，強化学習を「環境との相互作用のもとで累積報酬を最大化する確率的意思決定問題」としてどう定式化し，どう近似的に解くかという問いである．次節では，その標準的枠組みとしてマルコフ決定過程を導入する．

5.2.2　マルコフ決定過程

強化学習を厳密に論じるには，何が状態であり，何が行動であり，何が確率的に変化し，何を最大化したいのかを明示する必要がある．そのための標準的枠組みがマルコフ決定過程（Markov decision process; MDP）である．MDP によって，遅延報酬，価値関数，方策最適化という本章の主要概念を一つの数理的言語で記述できる．

定義　MDP は，状態空間 $\mathcal{S}$，行動空間 $\mathcal{A}$，遷移確率 $P(s' \mid s, a)$，報酬関数 $r(s, a, s')$，割引率 γ からなる組

$$(\mathcal{S}, \mathcal{A}, P, r, \gamma) \tag{5.43}$$

として与えられる．ここで $s \in \mathcal{S}$ は現在状態，$a \in \mathcal{A}$ は現在行動，$s' \in \mathcal{S}$ は次状態を表す．遷移確率 $P(s' \mid s, a)$ は，状態 s で行動 a を選んだときに次状態が s' になる条件付き確率であり，報酬関数 $r(s, a, s')$ はその遷移に対応する即時報酬の期待値である．割引率 $\gamma \in [0, 1)$ は，将来報酬をどの程度現在価値として重視するかを表す係数である．

時刻 t に状態 s_t を観測し，行動 a_t を選び，報酬 r_t を受け取って次状態 s_{t+1} へ移るとする．このとき，時刻 t 以降の累積割引報酬，すなわちリターン G_t を

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{5.44}$$

と定義する．添字 k は現在から何ステップ先の報酬かを表し，γ^k はその報酬に掛かる割引である．有限ホライズンではこの和は終端時刻までの有限和となり，無限ホライズンでは $\gamma < 1$ によって収束が保証される．

MDP の核心はマルコフ性にある．これは，次状態と次報酬の条件付き分布が，過去全体ではなく現在状態と現在行動だけで決まるという仮定であり，

$$\mathbb{P}(s_{t+1}, r_t \mid s_0, a_0, \ldots, s_t, a_t) = \mathbb{P}(s_{t+1}, r_t \mid s_t, a_t) \tag{5.45}$$

と書ける．この仮定が重要なのは，現在状態が将来予測に必要な情報を十分に要約しているとみなせるからである．その結果，長期最適化問題を再帰式として記述できる．

定義　方策（policy）$\pi(a \mid s)$ は，状態 s のもとで行動 a を選ぶ条件付き確率である．決定論的方策では各状態に一つの行動が対応し，確率的方策では複数の行動に確率が割り当てられる．方策 π が与えられたとき，状態価値関数（state-value function）$V^\pi(s)$ と行動価値関数（action-value function）$Q^\pi(s,a)$ を

$$V^\pi(s) = \mathbb{E}_\pi[G_t \mid s_t = s], \tag{5.46}$$

$$Q^\pi(s,a) = \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a] \tag{5.47}$$

と定義する．$\mathbb{E}_\pi[\cdot]$ は，方策 π と遷移確率 P のもとでの期待値である．$V^\pi(s)$ は状態 s にいることの長期的な良さを，$Q^\pi(s,a)$ は状態 s で行動 a を取ることの長期的な良さを表す．

これらの価値関数は，ベルマン期待方程式を満たす．状態価値関数については

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} P(s' \mid s, a) \left[r(s,a,s') + \gamma V^\pi(s') \right] \tag{5.48}$$

が成り立つ．同様に，行動価値関数は

$$Q^\pi(s,a) = \sum_{s' \in \mathcal{S}} P(s' \mid s, a) \left[r(s,a,s') + \gamma \sum_{a' \in \mathcal{A}} \pi(a' \mid s') Q^\pi(s', a') \right] \tag{5.49}$$

を満たす．ここで $a' \in \mathcal{A}$ は次状態 s' における次の行動である．価値が即時報酬と将来価値の和として再帰的に分解されることが，強化学習の理論的出発点である．

最適方策 π^* は，全状態で価値を最大化する方策として定義される．対応する最適状態価値関数 $V^*(s)$ と最適行動価値関数 $Q^*(s,a)$ は

$$V^*(s) = \max_\pi V^\pi(s), \tag{5.50}$$

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \tag{5.51}$$

であり，ベルマン最適方程式

$$V^*(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s' \mid s, a) \left[r(s,a,s') + \gamma V^*(s') \right], \tag{5.52}$$

$$Q^*(s,a) = \sum_{s' \in \mathcal{S}} P(s' \mid s, a) \left[r(s,a,s') + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a') \right] \tag{5.53}$$

を満たす．最大演算は，将来にわたって最も価値の高い行動を選ぶという最適制御の原理を表している．

有限ホライズンと無限ホライズン，エピソード型と継続型の区別も重要である．有限ホライズンでは終端時刻が存在し，無限ホライズンでは割引率が重要になる．エピソード型では終端状態に達

すると課題が終了するのに対し，継続型ではタスクが終わらないため，交通流制御のような問題では長期的安定性をどう定義するかが論点になる．次節では，状態遷移という複雑さを一度外し，探索と活用の本質だけを切り出した多腕バンディット問題を考える．

5.2.3　多腕バンディット問題

MDP は強化学習の標準形であるが，状態遷移まで含めると探索と活用の本質が見えにくくなることがある．そこで，多腕バンディット問題（multi-armed bandit problem）を導入する．これは状態遷移を持たず，未知の行動価値を試行錯誤によって学ぶという構造だけを残した問題であり，強化学習の基礎概念を理解するうえで有用である [1, 18].

定義　K 本の腕を持つバンディットを考える．各腕 $i \in \{1, \dots, K\}$ を選ぶと，未知分布に従う報酬が得られ，その期待値を μ_i とする．時刻 t に選ばれる腕を A_t，得られる報酬を r_t と書く．この問題には状態遷移がなく，意思決定は「次にどの腕を引くか」だけである．最適腕の期待報酬を $\mu^* = \max_i \mu_i$ とすると，目標は時刻 T までの累積期待報酬を最大化することである．

このとき，性能評価には後悔（regret）がよく用いられる．累積後悔 R_T を

$$R_T = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right] \tag{5.54}$$

と定義すると，これは「常に最適腕だけを引いた場合に比べてどれだけ損をしたか」を表す．強化学習全体では累積報酬最大化が中心目的であるが，バンディットでは後悔最小化の形で理論解析が明確になる．

最も単純な戦略の一つが ϵ-greedy である．これは確率 $1-\epsilon$ で現在までに推定された最良の腕を選び，確率 ϵ で一様ランダムに探索する方法である．ここで $\epsilon \in [0,1]$ は探索率を表す．直感的には，既知の有望な腕を主に活用しつつ，未知の腕をときどき試すことで誤った早期収束を防ぐ．ただし，不確実性の大きさを腕ごとに使い分けないため，探索の効率は高くない．

これに対して，上側信頼限界法（upper confidence bound; UCB）は，推定平均と不確かさの両方を用いる [1]．腕 i の時刻 $t-1$ までの経験平均を $\hat{\mu}_i(t-1)$，その腕を選んだ回数を $N_i(t-1)$ とすると，典型的には

$$A_t = \arg\max_i \left[\hat{\mu}_i(t-1) + c\sqrt{\frac{\log t}{N_i(t-1)}}\right] \tag{5.55}$$

で腕を選ぶ．ここで $c > 0$ は探索の強さを調整する定数である．右辺第二項は未観測性に対応し，試行回数が少ない腕ほど大きくなる．したがって UCB は，「現在よさそうな腕」だけでなく「まだ十分試していないので本当は良いかもしれない腕」も積極的に選ぶ．

もう一つの代表的方法が Thompson sampling である [34]．これは，各腕の平均報酬に関する事後分布を持ち，そこから仮想的な平均報酬 $\tilde{\mu}_i$ をサンプルして

$$A_t = \arg\max_i \tilde{\mu}_i \tag{5.56}$$

とする方法である．たとえばベルヌーイ報酬では，成功確率にベータ分布を置くことで計算が容易になる．Thompson sampling の特徴は，探索が外的なランダム化としてではなく，パラメータ不確実性のベイズ的表現から自然に導かれる点にある．

バンディット問題が MDP より簡単なのは，状態遷移がないため，長期的な信用割当てを考える必要がないからである．一方で，本質的に共通している点も明確である．すなわち，未知の環境で累積報酬を最大化するには，不確実性のもとで探索と活用を両立させなければならない．この意味で，多腕バンディットは強化学習の縮約版である．次節では，再び状態遷移を導入し，ただし環境モデルが既知である場合の計画法として動的計画法を扱う．

5.2.4 動的計画法

動的計画法（dynamic programming）が必要になるのは，環境モデルが既知であれば，経験から学ぶ前に理論上の最適解を計算できるからである．これは現実にはしばしば非現実的な仮定であるが，強化学習アルゴリズムの基準点としてきわめて重要である．モデル未知の場合の TD 学習や Q 学習は，多くの場合この枠組みをサンプル近似したものと理解できる．

定義　方策 π に対するベルマン作用素 T^π を

$$(T^\pi V)(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} P(s' \mid s, a) \left[r(s, a, s') + \gamma V(s') \right] \tag{5.57}$$

と定義する．ここで V は任意の状態価値関数である．同様に，最適ベルマン作用素 T_* を

$$(T_* V)(s) = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s' \mid s, a) \left[r(s, a, s') + \gamma V(s') \right] \tag{5.58}$$

と定義する．ベルマン期待方程式は $V^\pi = T^\pi V^\pi$，ベルマン最適方程式は $V^* = T_* V^*$ という不動点方程式になる．

アルゴリズム　方策評価（policy evaluation）は，固定された方策 π の価値関数 V^π を求める手続きである．初期関数 V_0 を任意に取り，

$$V_{k+1} = T^\pi V_k \tag{5.59}$$

と反復すると，V_k は V^π に近づく．直感的には，価値は終端報酬や高報酬状態から逆向きに伝播する．まず 1 ステップ先の価値が更新され，その情報がさらに前の状態へ波及していくのである．

方策改善（policy improvement）は，現在の価値関数にもとづいてより良い行動を選ぶ段階である．すなわち，各状態 s について

$$\pi'(s) \in \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s' \mid s, a) \left[r(s, a, s') + \gamma V^\pi(s') \right] \tag{5.60}$$

とすれば，新しい方策 π' は少なくとも π と同程度，通常はそれ以上の価値を持つ．これは，現在の価値評価に照らして各状態で最善の 1 ステップ行動を選べば，方策全体が改善することを意味している．

方策反復（policy iteration）は，方策評価と方策改善を交互に行う方法である．

1. 現在の方策 π_k を評価して V^{π_k} を求める．
2. V^{π_k} に対して貪欲化を行い，新しい方策 π_{k+1} を得る．
3. 方策が変化しなくなるまで反復する．

価値反復（value iteration）は，方策評価を完全に解かずに最適ベルマン作用素を直接適用する方法であり，

$$V_{k+1} = T_* V_k \tag{5.61}$$

を反復する．その後，各状態で価値を最大化する行動を選べば最適方策が得られる．価値反復は「いま分かっている将来価値に照らして最善の 1 ステップ選択」を毎回価値へ織り込む方法であり，計画と改善が一体化されている．

定理　ベルマン作用素が収束をもたらす理由は，割引率 $\gamma < 1$ のもとで縮小写像になるからである．最大ノルム $\|V - W\|_\infty = \max_{s \in \mathcal{S}} |V(s) - W(s)|$ を用いると，

$$\|T^\pi V - T^\pi W\|_\infty \le \gamma \|V - W\|_\infty, \tag{5.62}$$

$$\|T_* V - T_* W\|_\infty \le \gamma \|V - W\|_\infty \tag{5.63}$$

が成り立つ．誤差が反復ごとに少なくとも γ 倍に縮むため，不動点に向かって収束する．直感的には，遠い将来の影響が割引によって弱まり，誤差が増幅しないことを意味する．

動的計画法は理論上の理想解法である一方，遷移確率と報酬関数が既知でなければ使えない．そのため，モデル未知の状況では期待値計算を実際の観測遷移で置き換える必要がある．次節では，その橋渡しとして TD 学習から Q 学習へ至る価値ベース手法を扱う．

5.2.5　価値ベース手法：Q 学習, Deep-Q 学習

現実の強化学習では，環境モデルが未知であることが多い．したがって，動的計画法のように遷移確率の期待値を厳密に計算することはできず，経験から価値を学習しなければならない．価値ベース手法は，価値関数を学ぶことを通じて最適方策を間接的に得る方法であり，モデル未知環境における強化学習の中心的流儀である．

出発点となるのは TD 学習である．状態価値関数 V に対する TD(0) の更新は

$$V(s_t) \leftarrow V(s_t) + \alpha \left[r_t + \gamma V(s_{t+1}) - V(s_t) \right] \tag{5.64}$$

で与えられる．ここで $\alpha \in (0, 1]$ は学習率である．角括弧内の量

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{5.65}$$

を TD 誤差という．δ_t は，現在の価値予測が 1 ステップ先の観測を踏まえてどれだけ過小または過大であったかを示す量であり，価値修正の方向と大きさを与える信号である．

制御問題では，状態そのものの価値だけでなく，各行動の価値を知る必要がある．そこで行動価値関数 $Q(s, a)$ を学習する．Q 学習は

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right] \tag{5.66}$$

という更新式で定義される [37]．ここで $a' \in \mathcal{A}$ は次状態 s_{t+1} における候補行動であり，$\max_{a'} Q(s_{t+1}, a')$ は将来に最善行動を選ぶと仮定した価値である．この最大演算を通じて，Q 学習は最適方策へ向かう．

Q 学習の重要な特徴は off-policy 学習である点にある．すなわち，実際にデータを収集する行動方策と，更新式の内部で想定している目標方策が一致していなくてもよい．たとえば ϵ-greedy によって探索を行いながら，更新時には貪欲方策を想定できる．これに対し SARSA は

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \tag{5.67}$$

と，実際に選ばれた次行動 a_{t+1} を用いて更新する on-policy 手法である．Q 学習が将来の最良行動に基づく強気の更新であるのに対し，SARSA は探索を含んだ現在の方策をそのまま評価する．

表形式の Q 学習は，状態空間と行動空間が小さいときには有効であるが，高次元状態では破綻する．画像入力や多数の交通状態特徴量のような状況では，各 (s, a) に対して表を持つことができない．このとき必要になるのが関数近似であり，Deep Q-Network（DQN）はニューラルネットワーク $Q(s, a; \theta)$ によって行動価値関数を近似する [20]．

DQN では，TD ターゲットを

$$y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) \tag{5.68}$$

と置き，損失関数

$$L_t(\theta) = \left[y_t - Q(s_t, a_t; \theta) \right]^2 \tag{5.69}$$

を最小化する．ここで θ は学習中のネットワークの重み，θ^- はターゲットネットワークの重みである．もし右辺のターゲットまで同じ重みで毎回更新すると，目標そのものが激しく動くため学習が不安定化する．ターゲットネットワークはこの問題を緩和し，更新先を一時的に固定する役割を果たす．

もう一つの重要要素が経験再生（experience replay）である．これは，過去の遷移 (s_t, a_t, r_t, s_{t+1}) をバッファに蓄積し，そこから無作為にミニバッチを抽出して学習する仕組みである．経験再生が必要なのは，連続する遷移が強く相関しているため，そのまま学習すると勾配推定が不安定になるからである．無作為抽出により相関を弱められ，同じ経験を複数回再利用できるのでデータ効率も向上する．DQN の安定化は，この経験再生とターゲットネットワークに大きく依存している．

その後，Double DQN は最大演算に由来する過大評価を軽減し，dueling network は状態価値とアドバンテージを分離して表現することで学習効率を高めた．しかし中心的な論点は，TD 学習の再帰構造を深層関数近似とどう両立させるかにある．価値ベース手法の長所は，離散行動空間では各行動の価値を直接比較できること，ベルマン方程式との結びつきが明瞭であることである．一方で，連続行動空間では $\max_{a'} Q(s_{t+1}, a')$ の計算が難しく，確率的方策の直接制御や制約付き最適化にも不向きである．この限界が，次節で扱う方策勾配法の必要性を生む．

5.2.6　方策勾配法：Actor-Critic, PPO

価値ベース手法の限界が明確になるのは，連続行動空間や確率的方策の直接最適化を考えるときである．ロボット制御のように行動が実数ベクトルで表される場合，離散行動に対する最大化は自然ではない．また，探索を方策の分布そのものに埋め込みたい場合には，価値関数を介した間接的な最適化より，方策を直接最適化する方が適切である．この問題意識から方策勾配法が導入される．

パラメータ θ を持つ方策を $\pi_\theta(a \mid s)$ とし，期待リターンを

$$J(\theta) = \mathbb{E}_{\pi_\theta}[G_0] \tag{5.70}$$

と定義する．ここで $\mathbb{E}_{\pi_\theta}[\cdot]$ は，方策 π_θ と環境ダイナミクスに従って生成される軌道に関する期待値である．方策勾配法の目標は，$J(\theta)$ を最大化するように θ を更新することである．

定理　方策勾配の基本形は

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\, G_t\right] \tag{5.71}$$

で与えられる [39]．ここで ∇_θ はパラメータ θ に関する勾配，$\log \pi_\theta(a_t \mid s_t)$ は選択された行動の対数確率，G_t は時刻 t 以降のリターンである．この式は，良い帰結をもたらした行動の確率を高め，悪い帰結をもたらした行動の確率を下げるという学習規則を正当化する．

REINFORCE はこの勾配を標本平均で近似する最も基本的なアルゴリズムであり，

$$\theta \leftarrow \theta + \alpha \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) G_t \tag{5.72}$$

と更新する．ここで α は学習率である．REINFORCE は不偏推定量を与えるが，G_t の分散が大きく，学習が不安定になりやすい．とくにホライズンが長い問題では，分散の削減が不可欠になる．

そこでベースライン $b(s_t)$ を導入し，

$$\nabla_\theta J(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\,(G_t - b(s_t))\right] \tag{5.73}$$

と書き換える．$b(s_t)$ が状態のみに依存する限り，勾配の期待値は変わらず，分散だけを減らせる．とくに $b(s_t) = V^\pi(s_t)$ と置くと，アドバンテージ関数

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t) \tag{5.74}$$

が現れる．これは，状態 s_t において行動 a_t が平均的行動よりどれだけ良かったかを表す量である．

Actor-Critic はこの考え方を実装する枠組みである．Actor は方策 $\pi_\theta(a \mid s)$ を表し，Critic は価値関数，典型的には $V(s; w)$ を推定する．ここで w は Critic のパラメータである．Critic は TD 学習により

$$\delta_t = r_t + \gamma V(s_{t+1}; w) - V(s_t; w) \tag{5.75}$$

を計算する．この TD 誤差 δ_t は，1 ステップのアドバンテージ推定として機能する．したがって Actor は

$$\theta \leftarrow \theta + \alpha\, \delta_t\, \nabla_\theta \log \pi_\theta(a_t \mid s_t) \tag{5.76}$$

と更新できる．δ_t が正なら行動確率を増やし，負なら減らすので，Critic の価値推定が Actor の改善方向を与えることになる．

実際の深層強化学習では，方策更新が大きすぎると性能が急激に悪化することがある．TRPO は新旧方策の距離に制約を課すことでこの問題に対処したが [26]，計算は比較的重い．これに対し PPO は，同じ問題意識をより簡潔な目的関数で扱う [27]．

PPO では，古い方策を $\pi_{\theta_{\text{old}}}$，新しい方策を π_θ とし，確率比

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \tag{5.77}$$

を考える．さらに A_t を推定アドバンテージとすると，clip された目的関数は

$$L^{\text{clip}}(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)A_t,\ \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right) A_t\right)\right] \tag{5.78}$$

と書かれる．ここで $\epsilon > 0$ は更新幅を制御するハイパーパラメータであり，$\text{clip}(\cdot)$ は値を区間 $[1-\epsilon, 1+\epsilon]$ に切り詰める演算である．新方策が旧方策から過度に離れると，利得の増加が頭打ちになるため，大きすぎる更新が抑制される．

PPO が実用上の標準手法とみなされるのは，信頼領域の思想を保ちつつ，実装が容易でミニバッチ最適化とも相性がよいからである．とくに連続行動空間，大規模関数近似，確率的方策の直接制御が必要な問題では，方策勾配法の重要性は大きい．価値ベース手法が価値推定を通じて行動を導くのに対し，方策勾配法は望ましい行動分布そのものを直接整形する．この違いが，現代の深層強化学習における両者の役割分担を定めている．

5.2.7 発展的話題

標準的な MDP，価値学習，方策勾配法は強力な枠組みであるが，現実世界の課題をそのまま捉えられるわけではない．実応用では，環境モデルは未知であり，観測は不完全であり，安全制約があり，データ収集が高価であることが多い．この節では，そのような制約のもとで基本理論がどのように拡張されるかを概観する．

(a) モデルベース強化学習

モデルベース強化学習（model-based RL）が必要になるのは，サンプル効率を高めたいからである．モデルフリー手法は柔軟であるが，多数の試行を要する．実機ロボットや都市交通制御では，試行そのものが高価であり，環境モデルを全く使わない学習は非現実的であることが多い．

基本的な発想は，遷移モデル $\hat{P}$ と報酬モデル $\hat{r}$ をデータから学習し，その近似モデルの内部で計画やロールアウトを行うことである．モデルが十分正確なら，実環境での試行を減らしつつ方策改善ができる．一方で，モデル誤差は長期予測で蓄積しやすいため，短い計画区間に限定する，モデル不確実性を推定する，モデルフリー更新と併用するといった工夫が必要になる．

(b) オフライン強化学習

オフライン強化学習（offline RL）が必要になるのは，新たな探索を現場で行えない状況があるからである．医療，金融，公共交通運用のような領域では，学習のために試しに危険な行動を取ることは許されない．そのため，過去に収集された固定データ集合

$$\mathcal{D} = \{(s, a, r, s')\} \tag{5.79}$$

だけを用いて方策を学習する．

ここでの難しさは，データに含まれていない行動を価値関数が過大評価しやすい点にある．すなわち，分布外の行動に対する楽観主義が性能劣化を招く．このため，オフライン強化学習では，行動方策に近い範囲で保守的に学習する，価値推定を意図的に低めに補正する，行動分布そのものを制約する，といった工夫が中心課題となる．

(c)　部分観測 MDP

標準的 MDP では現在状態が将来予測に必要な情報を十分含むと仮定したが，現実には状態そのものが直接観測できないことが多い．部分観測 MDP（partially observable MDP; POMDP）は，この状況を扱うための拡張であり，エージェントは状態 s_t の代わりに観測 o_t だけを得る．

このとき単一の観測だけでは意思決定に必要な情報が不足することがある．たとえば需要の潜在状態や他主体の意図は直接見えない．そこで，過去観測から信念状態を更新して隠れ状態の分布を内部状態として扱う方法や，再帰型ネットワークによって観測履歴を潜在表現へ圧縮する方法が用いられる．POMDP は，マルコフ性を放棄するのではなく，観測から実質的なマルコフ表現を再構成する問題として理解できる．

(d)　探索の高度化

疎な報酬や長いホライズンを持つ問題では，単純な ϵ-greedy では有益な経験に到達しにくい．このため，探索そのものを主題化する必要がある．代表的な考え方の一つが内発的動機づけ（intrinsic motivation）であり，新規性や予測誤差にもとづく探索ボーナスを外的報酬に加える．

たとえば修正報酬を

$$r'_t = r_t + \beta b_t \tag{5.80}$$

と書けば，$\beta > 0$ は探索ボーナスの重み，b_t は新規性や予測困難性を表す量である．ほかにも，不確実性の大きい行動を積極的に高く見る楽観主義や，事後分布から方策をサンプルする posterior sampling がある．探索は単なるランダム化ではなく，情報獲得をどう最適化するかという設計問題へ発展している．

(e)　多エージェント強化学習

多エージェント強化学習（multi-agent RL; MARL）が必要になるのは，環境が単一主体ではなく複数主体の相互作用で構成されるからである．交通流や配車市場では，各主体が学習することで他主体にとっての環境も変化する．その結果，各エージェントから見れば環境は非定常となり，単一エージェント MDP の仮定がそのままでは通用しにくい．

ここでの問題は，協調と競争をどう扱うかである．協調型では局所情報から全体効率を高める共同方策を構成する必要があり，競争型では相手の学習を見越した戦略的適応が求められる．実用的には，学習時には全体情報を利用し，実行時には各主体が局所観測だけで動く centralized training with decentralized execution が広く用いられる．

(f)　模倣学習と逆強化学習

現実の問題では，報酬関数を適切に設計すること自体が難しい．自動運転や運行制御では，「安全」「快適」「効率」といった複数の目的を単一の数値報酬に落とし込むのが容易ではない．このと

き，人間や既存システムの行動データを利用する方法として，模倣学習（imitation learning）と逆強化学習（inverse RL）が重要になる．

模倣学習は専門家の行動系列から直接方策を学ぶ方法であり，探索コストを大きく削減できる．逆強化学習は，専門家行動の背後にある報酬関数を推定し，その推定報酬のもとで方策最適化を行う方法である．前者が「どう行動するか」を学ぶのに対し，後者は「何を目的としているか」を推定する．いずれも，標準的な MDP ＋価値学習＋方策勾配だけでは扱いにくい現実的制約を補う拡張である．

5.2.8 まとめ

本章では，強化学習を環境との相互作用のもとで長期的な意思決定を最適化する理論として体系的に整理した．まず，教師あり学習や教師なし学習では十分に捉えられない遅延報酬，試行錯誤，探索と活用の問題から強化学習が必要になることを述べ，その歴史が動的計画法，TD 学習，Q 学習，深層関数近似，方策勾配法へと展開してきた流れを確認した．

続いて，マルコフ決定過程によって状態，行動，遷移確率，報酬，割引率，方策，価値関数を定義し，ベルマン方程式が強化学習全体の再帰的骨格であることを示した．多腕バンディット問題は，探索と活用の本質を状態遷移なしに抽出する縮約版として位置づけられ，動的計画法は環境モデル既知の場合の理論的基準点を与えた．そのうえで，モデル未知の状況では TD 学習と Q 学習が経験から価値を推定し，DQN が高次元状態へ拡張した．さらに，価値ベース手法の限界を受けて，方策勾配法，Actor-Critic，PPO が方策そのものを直接最適化する枠組みを与えることを見た．最後に，モデルベース強化学習，オフライン強化学習，POMDP，探索の高度化，多エージェント強化学習，模倣学習と逆強化学習を通じて，現実世界では未知性，部分観測，安全制約，データ収集制約が基本理論の拡張を要請することを確認した．以上を通じて，強化学習は「環境との相互作用のもとで価値関数または方策を学習し，長期的な意思決定を最適化する理論」として理解されるべきである．

コラム：探索と活用のジレンマ（多腕バンディット問題の実験）– 長田晴人

実験設定と評価指標

強化学習の重要なテーマの一つに，**探索と活用のジレンマ**がある．ここでは，第第 5.2.3 小節小節で取り上げた多腕バンディット問題を用いて，いくつかの探索手法を同一条件で比較し，パラメータの違いが学習曲線にどのように現れるかを確認する．

バンディット問題の名前は，スロットマシンが “one-armed bandit” と呼ばれていたことに由来する．レバーを引くと確率的に報酬が得られるが，どのレバーの期待報酬が高いかはプレイヤーには分からない．本実験では，6 本の腕 $a \in \{1, \ldots, 6\}$ を持つバンディットを考え，各腕を選ぶと確率

$$p = \left(\frac{1}{20}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{2}, \frac{3}{4} \right)$$

で報酬 1 が得られ，それ以外では報酬 0 が得られるとする．学習者はこの真の確率を知らず，実際に得られた報酬から各腕の行動価値を推定する．この問題では第 6 腕が最適腕であるが，第 5 腕も比較的高い報酬確率を持つため，初期の偶然によって第 5 腕を過大評価し，第 6 腕を十分に試さないまま学習が進む可能性がある．

比較する手法は，ε-greedy，softmax，UCB，optimistic 初期値，Thompson sampling である．このコラムの目的は，平均報酬と累積後悔の推移を通じて，**探索の強さ**や**価値推定の方法**が学習過程に与える影響を観察することである．確率的なばらつきを抑えるため，各条件について 1000 回の独立試行を行い，その平均を学習曲線としてプロットする．

まず，行動価値の更新方法と腕の選択方法を分けて整理する．表 5.4 は「得られた報酬をどのように価値推定へ反映するか」を表し，表 5.5 は「推定された価値にもとづいて次にどの腕を選ぶか」を表す．

表 5.4: 行動価値の更新方法

手法・式	備考
標本平均手法 $Q_t(a) = \frac{1}{N_t(a)} \sum_{\tau<t:A_\tau=a} r_\tau$	$N_t(a)$ は，時刻 t より前に腕 a を選んだ回数である．すなわち $N_t(a) = \sum_{\tau<t} \mathbf{1}\{A_\tau = a\}$ である．過去に腕 a から得た報酬の平均をそのまま行動価値とする．定常環境では自然な推定方法であるが，過去の観測をすべて同じ重みで扱うため，非定常環境では古い情報を引きずりやすい．
逐次的な標本平均 $Q_{t+1}(a) = Q_t(a) + \frac{1}{N_t(a)+1}\{r_t - Q_t(a)\} \quad (A_t = a)$	標本平均手法を逐次更新の形で書いたものである．時刻 t に腕 a を選び，報酬 r_t を得た後，その観測を加えた平均として $Q_{t+1}(a)$ を更新する．新しい報酬の重みは $1/(N_t(a)+1)$ であり，選択回数が増えるほど，新しい観測の影響は小さくなる．
固定学習率手法 $Q_{t+1}(a) = Q_t(a) + \alpha\{r_t - Q_t(a)\} \quad (A_t = a)$	$\alpha \in (0, 1]$ は学習率である．時刻 t に腕 a を選び，報酬 r_t を得た後，新しい報酬を一定の重みで反映する．α が大きいほど最近の観測を重視し，小さいほど推定は安定する．標本平均手法と異なり，古い観測の影響が相対的に小さくなるため，非定常環境に追従しやすい．

表 5.5: 多腕バンディット問題における腕の選択方法

手法・式	備考
greedy $A_t = \arg\max_{a \in \mathcal{A}} Q_t(a)$	常に現在の推定価値 $Q_t(a)$ が最大の腕を選ぶ．探索を行わないため，初期の偶然によって悪い腕を過大評価すると，その腕を選び続ける可能性がある．
ε**-greedy** $A_t = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_t(a), & \text{確率 } 1-\varepsilon, \\ \text{ランダムな腕}, & \text{確率 } \varepsilon \end{cases}$	$\varepsilon \in [0,1]$ は探索率である．確率 ε でランダム探索を行う．ε が大きいほど未知の腕を試しやすいが，学習後も低報酬の腕を選ぶ確率が残る．$\varepsilon = 0$ では greedy と一致する．
softmax $\Pr(A_t = a) = \dfrac{\exp(Q_t(a)/\tau)}{\sum_{b \in \mathcal{A}} \exp(Q_t(b)/\tau)}$	$\tau > 0$ は温度パラメータである．τ が大きいほど選択確率は一様に近づき，τ が小さいほど greedy に近づく．価値差をどの程度鋭く選択確率に反映するかを決めるスケールパラメータとして解釈できる．
UCB $A_t = \arg\max_{a \in \mathcal{A}} \left\{ Q_t(a) + c\sqrt{\dfrac{\log t}{N_t(a)}} \right\}$	$c > 0$ は探索ボーナスの強さ，$N_t(a)$ は腕 a の選択回数である．ただし，$N_t(a) = 0$ では第 2 項が定義できないため，本実験では各腕を最初に 1 回ずつ選んだ後にこの式を用いる．選択回数が少ない腕ほど第 2 項が大きくなるため，未探索の腕を体系的に試す．c が大きいほど探索的になり，小さいほど現在の推定価値を重視する．
optimistic 初期値 $Q_0(a) = Q_{\text{init}} \qquad (Q_{\text{init}} \text{ を大きく設定})$	Q_{init} は初期行動価値である．行動選択規則そのものではなく，初期価値を高く置くことで探索を促す方法である．まだ試していない腕が楽観的に評価されるため，明示的なランダム探索を入れなくても複数の腕を試しやすい．
Thompson sampling $\tilde{\theta}_a \sim P_t(\theta_a), \qquad A_t = \arg\max_{a \in \mathcal{A}} \mu(a; \tilde{\theta}_a)$	$P_t(\theta_a)$ は, 腕 a の報酬分布のパラメータ θ_a に関する事後分布である．各腕の事後分布からパラメータをサンプリングし，そのもとでの期待報酬 $\mu(a;\tilde{\theta}_a)$ が最大の腕を選ぶ．今回は Bernoulli 報酬なので, 成功確率 p_a の事後分布としてベータ分布 $\mathrm{Beta}(\alpha_a, \beta_a)$ を用いる．

累積後悔についても簡単に述べておく．平均報酬が「どれだけ報酬を得られたか」を表すのに対し，累積後悔は「最初から最適腕を知っていた場合に比べて，どれだけ報酬を取り逃したか」を表す．時刻 t に選んだ腕を A_t，その真の報酬確率を p_{A_t}，最適腕の報酬確率

を $p^* = 3/4$ とすると，時刻 T までの累積後悔は

$$R_T = \sum_{t=1}^{T} (p^* - p_{A_t})$$

で定義される．この指標を見ることで，探索のために低確率の腕を引いたコストや，第 5 腕のような「そこそこ良い腕」に固定されたことによる損失を評価できる．

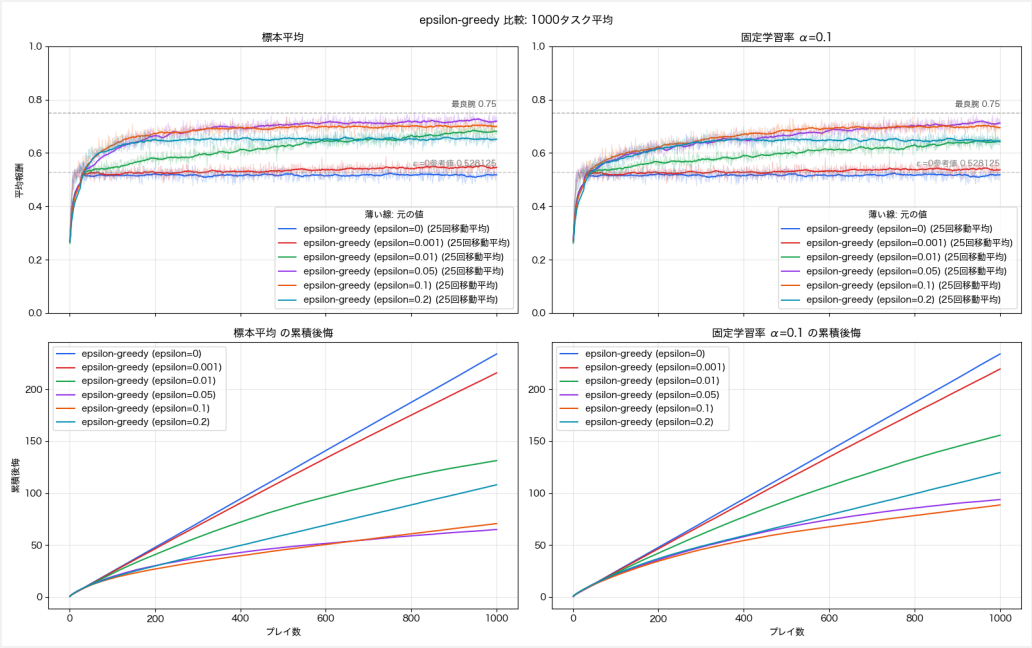


図 5.5: greedy 法と ε-greedy 法の学習曲線

greedy 法と ε-greedy 法

図 5.5 では，greedy 法と ε-greedy 法を比較した．greedy 法は

$$A_t = \arg\max_{a \in \mathcal{A}} Q_t(a)$$

により，常に現在の推定価値が最大の腕を選ぶ方法である．一方，ε-greedy 法では，確率 $1 - \varepsilon$ で greedy な腕を選び，確率 ε でランダムに腕を選ぶ．

$\varepsilon = 0$ の greedy 法では，初期値をすべて $Q_0(a) = 0$ とすると，最初に報酬 1 を出した腕に固定されやすい．報酬が出るまでは同点の腕をランダムに選ぶため，最初に成功する腕が a である確率は

$$q_a = \frac{p_a}{\sum_{b=1}^{6} p_b}$$

と近似できる．したがって，長期的な平均報酬は

$$\bar{r}_\infty = \sum_{a=1}^{6} q_a p_a = \frac{\sum_{a=1}^{6} p_a^2}{\sum_{a=1}^{6} p_a}$$

となる．今回の設定では

$$\sum_{a=1}^{6} p_a = \frac{8}{5}, \qquad \sum_{a=1}^{6} p_a^2 = \frac{169}{200}$$

であるから，

$$\bar{r}_\infty = \frac{169/200}{8/5} = \frac{169}{320} = 0.528125$$

となる．これは図中の $\varepsilon = 0$ の収束値とよく一致している．

一方，$\varepsilon = 0.2$ の場合，学習が十分進んで greedy 部分では第 6 腕を選べるようになったとしても，全体の 20% はランダムに腕を選ぶ．このとき長期的な平均報酬はおよそ

$$(1-\varepsilon)p^* + \varepsilon \frac{1}{6} \sum_{a=1}^{6} p_a$$

で与えられる．今回の設定では $p^* = 3/4$,

$$\frac{1}{6} \sum_{a=1}^{6} p_a = \frac{4}{15}$$

であるから，

$$(1-0.2) \cdot \frac{3}{4} + 0.2 \cdot \frac{4}{15} \simeq 0.653$$

となる．図で $\varepsilon = 0.2$ の平均報酬が最適腕の 0.75 より低い値に落ち着くのは，学習後も一定確率で低報酬の腕を選び続けるためである．つまり，ε **は小さすぎても大きすぎてもよくない**．今回の結果では，$\varepsilon = 0.05$ から 0.1 程度が，初期探索と後半の活用のバランスがよかった．

累積後悔のグラフは，この違いを別の形で示している．累積後悔の傾きは，各時刻でどれだけ報酬を取り逃しているか，すなわち瞬間的な後悔の大きさを表す．たとえば，greedy 法が平均報酬 $\bar{r}_\infty$ の腕選択に固定されているとき，累積後悔の傾きはおよそ

$$p^* - \bar{r}_\infty$$

となる．$\varepsilon = 0$ では

$$\frac{3}{4} - 0.528125 = 0.221875$$

となり，後悔が大きな傾きで増え続ける．これは，学習が速いというより，早い段階で誤った腕に固定されてしまっている状態である．

これに対して，$\varepsilon > 0$ では初期に探索を行うため，序盤の累積後悔は増えやすい．しかし，その探索によって最適腕を発見できれば，その後の累積後悔の傾きは小さくなる．また，累積後悔の傾きが時間とともに小さく変化し続けている場合には，腕の選択がまだ改善して

おり，学習が継続していると解釈できる．逆に，累積後悔がほぼ直線になった場合には，腕の選択傾向がほぼ固定され，その時点の方策で一定の損失を出し続けていると考えられる．これは，初期に損をしてでも正しい腕を見つける「学習の速さ」と，学習後に不要な探索をどれだけ減らせるかという「最終的な正確性」のトレードオフを表している．

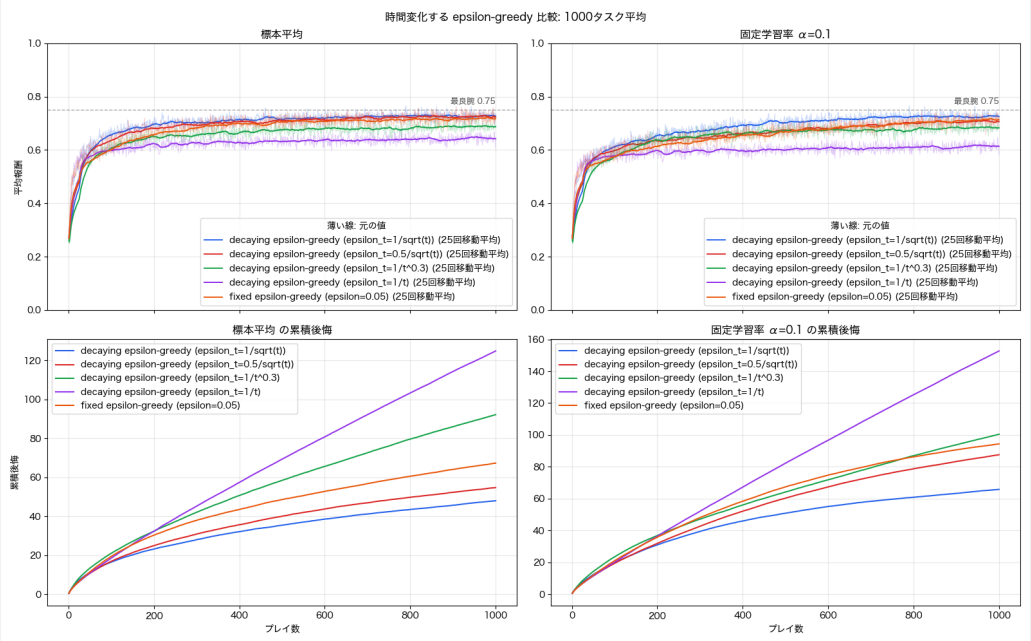


図 5.6: ε が時間変化する場合の学習曲線

時間変化する ε_t-greedy 法

固定した ε を用いる ε-greedy 法では，ε が小さすぎると探索が不足し，初期に高く評価された腕に固定されやすい．一方で，ε が大きすぎると，最適腕を見つけた後もランダム探索が残り，低報酬の腕を選ぶ損失が続いてしまう．そこで，学習初期には十分に探索し，学習後半では不要な探索を減らすために，時間とともに探索率を小さくする ε_t-greedy 法を考える．

ε_t-greedy 法では，

$$A_t = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_t(a), & \text{確率 } 1-\varepsilon_t, \\ \text{ランダムな腕}, & \text{確率 } \varepsilon_t \end{cases}$$

として，各時刻 t における探索率 ε_t を変化させる．本実験では，

$$\varepsilon_t = \frac{1}{\sqrt{t}}, \quad \varepsilon_t = \frac{0.5}{\sqrt{t}}, \quad \varepsilon_t = \frac{1}{t^{0.3}}, \quad \varepsilon_t = \frac{1}{t}$$

を比較し，固定値 $\varepsilon = 0.05$ の場合とも比べた．

図 5.6 を見ると，$\varepsilon_t = 1/\sqrt{t}$ は，平均報酬と累積後悔の両方で比較的良い挙動を示している．これは，時刻 T までに発生するランダム探索の期待回数から説明できる．ε_t-greedy 法では，各時刻に確率 ε_t で探索を行うため，探索回数 M_T の期待値は

$$\mathbb{E}[M_T] = \sum_{t=1}^{T} \varepsilon_t$$

である．腕の本数を $K = 6$ とすると，ランダム探索だけで各腕が試される回数の目安は

$$\frac{1}{K} \sum_{t=1}^{T} \varepsilon_t$$

となる．

$\varepsilon_t = 1/t$ の場合，

$$\sum_{t=1}^{T} \frac{1}{t} \simeq \log T$$

である．今回の実験では $T = 1000$ なので，$\log 1000 \simeq 6.9$ となり，各腕をランダム探索で試す回数は平均して 1 回程度にすぎない．そのため，早い段階で探索率が小さくなりすぎ，第 6 腕を十分に試す前に別の腕へ固定されやすい．

一方，$\varepsilon_t = 1/\sqrt{t}$ の場合，

$$\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \simeq 2\sqrt{T}$$

である．$T = 1000$ では $2\sqrt{1000} \simeq 63$ となるため，各腕はランダム探索だけでも平均して 10 回程度試される．今回の報酬確率では，第 6 腕の報酬確率 $p_6 = 3/4$ と第 5 腕の報酬確率 $p_5 = 1/2$ の差が

$$\Delta = p_6 - p_5 = \frac{1}{4}$$

と比較的大きい．そのため，各腕を十数回程度試せば，第 6 腕が有望であることを発見するきっかけを得やすい．さらに，$\varepsilon_t = 1/\sqrt{t}$ は時間とともに 0 に近づくため，最適腕を見つけた後には不要なランダム探索が減り，平均報酬を高く保ちやすい．

これに対して，$\varepsilon_t = 1/t^{0.3}$ の場合，

$$\sum_{t=1}^{T} \frac{1}{t^{0.3}} \simeq \frac{T^{0.7}}{0.7}$$

であり，$T = 1000$ ではおよそ 180 程度になる．これは $\varepsilon_t = 1/\sqrt{t}$ よりも探索回数が多く，最適腕を発見するうえでは有利である．しかし，学習後半にもランダム探索が多く残るため，低報酬の腕を引く損失が続き，累積後悔の増加を抑えにくい．つまり，$1/t$ は探索が減るのが速すぎ，$1/t^{0.3}$ は探索が残りすぎる．その中間にある $1/\sqrt{t}$ は，今回の設定では**探索不足と探索過多のバランス**がよかったと考えられる．

ここで，標本平均と固定学習率の違いにも注目したい．固定した ε の実験では両者の差は比較的小さかったが，時間変化する ε_t ではその差がより現れている．標本平均手法では

$$Q_{t+1}(a) = Q_t(a) + \frac{1}{N_t(a)}\{r_t - Q_t(a)\}$$

であり，腕 a を多く選ぶほど新しい報酬の重み $1/N_t(a)$ は小さくなる．一方，固定学習率では

$$Q_{t+1}(a) = Q_t(a) + \alpha\{r_t - Q_t(a)\}$$

であり，新しい報酬は常に重み α で反映される．実際，固定学習率の更新を展開すると

$$Q_n = (1-\alpha)^n Q_0 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} r_i$$

となり，最近の報酬ほど大きな重みを持つことが分かる．

時間変化する ε_t では，学習後半の探索機会が減るため，後から得られた少数の観測をどれだけ強く価値推定に反映できるかが重要になる．そのため，過去の観測を均等に平均する標本平均よりも，最近の結果を大きく見る固定学習率の方が，減衰スケジュールの違いに敏感に反応しやすい．このため，固定 ε では目立たなかった価値更新方法の違いが，時間変化する ε_t の実験ではよりはっきり現れたと考えられる．

ただし，この結果は $\varepsilon_t = 1/\sqrt{t}$ が一般に最適な減衰率であることを意味しない．適切な減衰率は，腕の本数，プレイ回数，最適腕と次善腕の報酬確率差，および価値更新方法に依存する．今回の 6 本腕バンディット問題では，$T = 1000$ という有限時間の中で，$1/\sqrt{t}$ が「十分に探索するが，探索し続けすぎない」スケールになっていたと解釈できる．

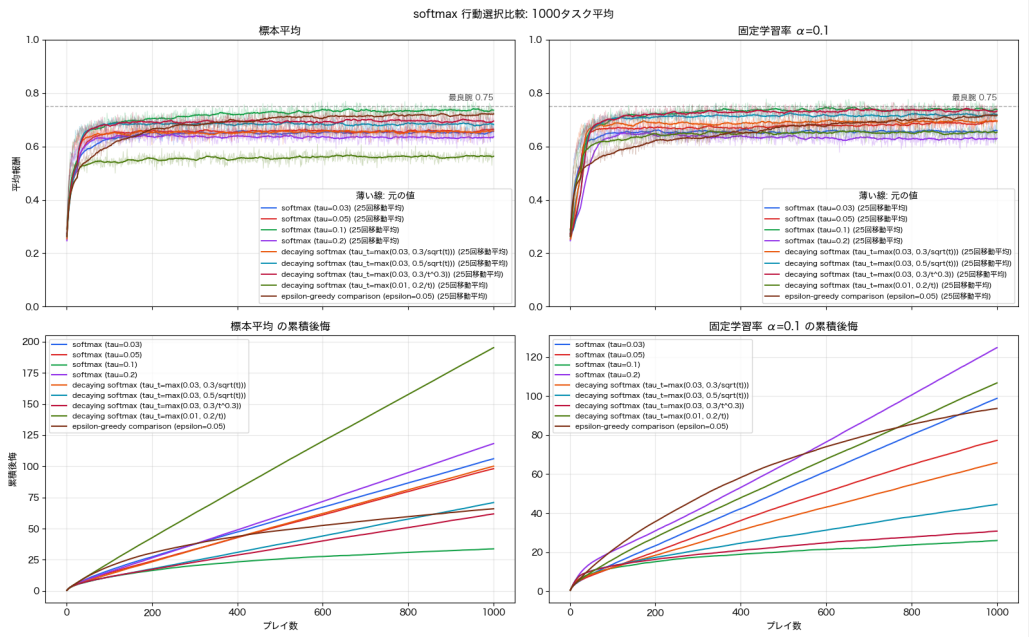


図 5.7: softmax 法の学習曲線

softmax 法

次に，softmax による行動選択を比較する．softmax 法では，各腕の推定行動価値 $Q_t(a)$ に対して

$$\Pr(A_t = a) = \frac{\exp(Q_t(a)/\tau)}{\sum_{b \in \mathcal{A}} \exp(Q_t(b)/\tau)}$$

によって選択確率を与える．ここで $\tau > 0$ は温度パラメータであり，τ が大きいほど各腕の選択確率は一様に近づき，τ が小さいほど greedy 法に近づく．したがって，τ は「価値の差をどれくらい鋭く行動選択に反映するか」を決めるスケールパラメータである．

固定温度の結果を見ると，今回の設定では $\tau = 0.1$ 付近が比較的よい性能を示している．この理由は，今回の真の報酬確率の差の大きさから説明できる．最適腕である第 6 腕と次善腕である第 5 腕の報酬確率差は

$$p_6 - p_5 = \frac{3}{4} - \frac{1}{2} = \frac{1}{4}$$

である．softmax では，2 つの腕 a, b の選択確率の比は

$$\frac{\Pr(A_t = a)}{\Pr(A_t = b)} = \exp\left(\frac{Q_t(a) - Q_t(b)}{\tau}\right)$$

となる．仮に推定値が真の報酬確率に近づき，$Q_t(6) - Q_t(5) \simeq 0.25$ であるとすると，$\tau = 0.1$ のとき

$$\frac{\Pr(A_t = 6)}{\Pr(A_t = 5)} \simeq \exp\left(\frac{0.25}{0.1}\right) = \exp(2.5) \simeq 12.2$$

となる．つまり，第 6 腕は第 5 腕より十分選ばれやすいが，第 5 腕が完全に排除されるほど極端でもない．この程度の鋭さが，今回の問題では探索と活用のバランスとしてちょうどよかったと考えられる．

一方，$\tau = 0.2$ では

$$\exp\left(\frac{0.25}{0.2}\right) = \exp(1.25) \simeq 3.5$$

となり，第 6 腕と第 5 腕の選択確率の差がそれほど大きくならない．そのため，学習後も第 5 腕や他の腕を選ぶ確率が残りやすく，平均報酬は最適値 0.75 から離れ，累積後悔も増えやすい．逆に，$\tau = 0.03$ のように温度が小さすぎると，

$$\exp\left(\frac{0.25}{0.03}\right) \simeq \exp(8.33)$$

となり，わずかな価値差でも選択確率が極端に偏る．学習が十分進んだ後であれば有利に働くが，初期には $Q_t(a)$ の推定誤差が大きいため，偶然高く評価された腕に選択が集中しやすい．

このように，softmax 法でも**温度は大きすぎても小さすぎてもよくない**．今回の 6 本腕バンディット問題では，最適腕と次善腕の差が 0.25 であるため，$\tau = 0.1$ 程度にすると

$$\frac{p_6 - p_5}{\tau} = \frac{0.25}{0.1} = 2.5$$

となり，softmax の指数関数によって最適腕を十分に優遇しつつ，完全な greedy にはならない程度の選択確率が得られる．これが，固定温度 $\tau = 0.1$ が今回うまく働いた主な理由である．

さらに，温度を時間とともに変化させる方法も試した．これは，学習初期には温度を高めにして広く探索し，学習が進むにつれて温度を下げることで，より価値の高い腕に選択を集中させる方法である．本実験では，たとえば

$$\tau_t = \max\left(0.03, \frac{c}{\sqrt{t}}\right)$$

のように，温度を減衰させつつ，下限値 0.03 を設けた．下限を置いたのは，温度が小さくなりすぎるとほぼ greedy になり，わずかな推定誤差にも過剰に反応してしまうためである．図 5.7 では，温度を適切に減衰させた場合，固定温度よりも累積後悔の増加を抑えられる場合がある．これは，序盤では探索によって第 6 腕を発見する機会を確保し，後半では温度低下によって第 6 腕への選択を強められるためである．

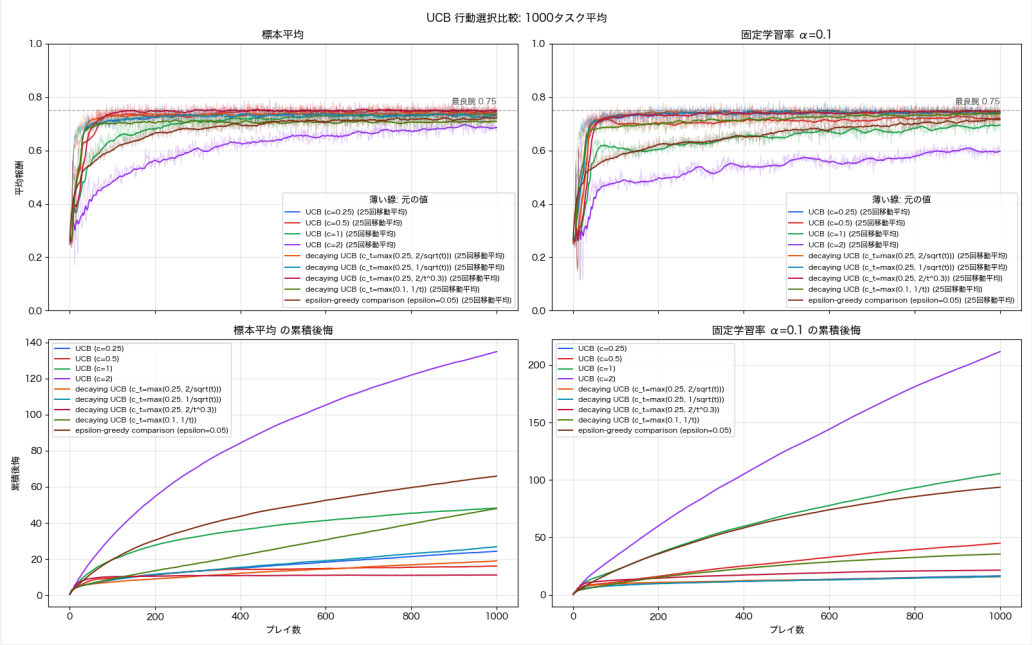


図 5.8: UCB 法の学習曲線

UCB 法

次に，UCB による行動選択を比較する．UCB は，現在の推定価値が高い腕だけでなく，ま

だ十分に試されていない腕も高く評価する方法である．ただし，未選択の腕では $N_t(a) = 0$ となり探索ボーナスが定義できないため，本実験では各腕を最初に 1 回ずつ選んだ後で，

$$A_t = \arg\max_{a \in \mathcal{A}} \left\{ Q_t(a) + c\sqrt{\frac{\log t}{N_t(a)}} \right\}$$

によって腕を選ぶ．ここで $N_t(a)$ は時刻 t までに腕 a を選んだ回数であり，$c > 0$ は探索ボーナスの強さを決めるパラメータである．第 2 項は，選択回数の少ない腕ほど大きくなるため，UCB は単なるランダム探索ではなく，**まだ不確実な腕を意図的に試す**方法と解釈できる．

図 5.8 を見ると，c が大きすぎる場合には性能が悪化している．特に $c = 2$ では平均報酬が低く，累積後悔も大きい．これは，探索ボーナス

$$c\sqrt{\frac{\log t}{N_t(a)}}$$

が大きすぎるために，すでに低報酬だと分かりつつある腕も繰り返し試してしまうからである．今回の問題では，最適腕と次善腕の報酬確率差は

$$\Delta = p_6 - p_5 = \frac{1}{4}$$

である．したがって，探索ボーナスがこの差 Δ より十分大きい間は，第 6 腕が高い推定価値を持っていても，試行回数の少ない腕が選ばれやすい．つまり，c が大きすぎると，UCB は慎重すぎる学習者になり，不確実性を減らすための探索コストが累積後悔として残る．

一方で，c が小さい場合には，探索ボーナスが弱くなり，現在の推定価値 $Q_t(a)$ をより重視する．図では，$c = 0.25$ や $c = 0.5$ が比較的良い結果を示している．これは，今回の 6 本腕バンディット問題では報酬確率の差が比較的大きく，少ない試行回数でも第 6 腕の有利さを見つけやすかったためだと考えられる．

UCB の特徴は，累積後悔の傾きにも現れている．c が適切な場合，初期には全ての腕を試すために後悔が増えるが，第 6 腕が有望であると分かるにつれて選択が集中し，累積後悔の傾きが小さくなる．一方，$c = 2$ のように探索が強すぎる場合には，後半になっても低報酬の腕を試す影響が残るため，累積後悔の傾きが十分に小さくならない．

また，時間変化する UCB 係数 c_t も比較した．これは，学習初期には大きめの探索ボーナスで各腕を広く試し，学習後半では探索ボーナスを小さくして，推定価値の高い腕をより重視する方法である．たとえば

$$c_t = \max\left(0.25, \frac{1}{\sqrt{t}}\right)$$

のように設定すると，初期には強めに探索しつつ，後半には c_t が下限値 0.25 に近づく．UCB の探索ボーナスそのものは

$$c_t\sqrt{\frac{\log t}{N_t(a)}}$$

であるため，c_t を小さくすることで後半の過剰な探索をさらに抑えられる．ただし，c_t を小さくしすぎると，UCB の利点である「試行回数の少ない腕を体系的に試す」効果も弱くなる．したがって，UCB でも探索の強さを問題のスケールに合わせることが重要である．

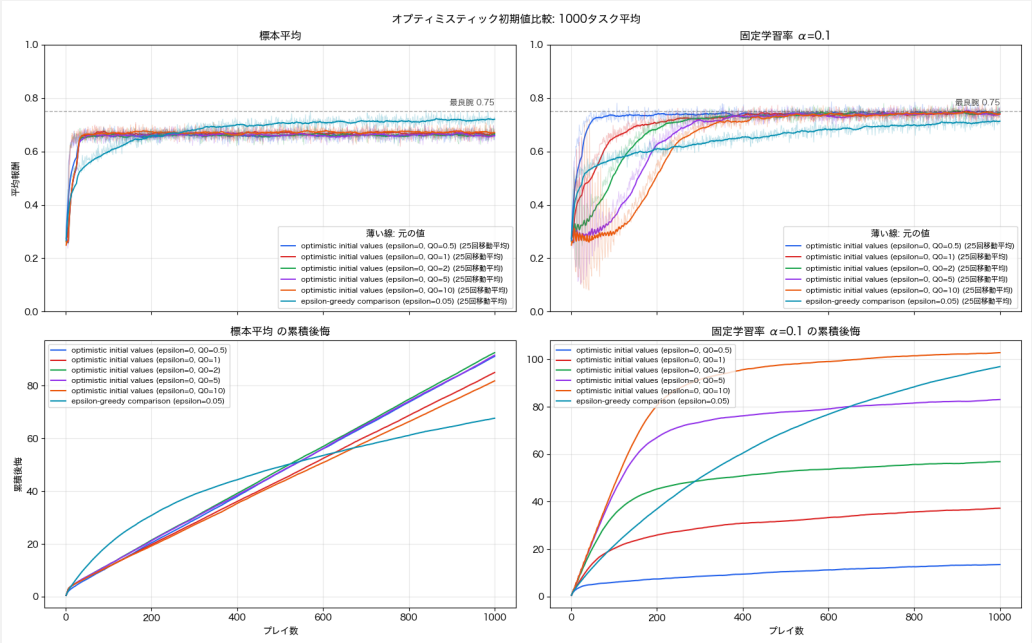


図 5.9: オプティミスティック初期値の学習曲線

optimistic 初期値

次に，optimistic 初期値による探索を比較する．これは，初期の行動価値 $Q_0(a)$ を高めに設定することで，まだ十分に試していない腕を魅力的に見せる方法である．本実験では $\varepsilon = 0$ とし，明示的なランダム探索は行わず，初期値の楽観性だけで探索がどの程度生じるかを調べた．

図 5.9 を見ると，標本平均を用いた場合には，Q_0 を変えても最終的な結果はどれも似たように悪くなっている．これは，標本平均手法では，腕 a を初めて選んだ時点で

$$Q_1(a) = r_1$$

となり，初期値 $Q_0(a)$ の影響がほとんど消えてしまうためである．つまり，Q_0 をいくら大きくしても，その腕を一度試すと，価値推定は実際に得られた 0 または 1 の報酬に置き換わる．そのため，optimistic 初期値は「まだ一度も試していない腕を試させる」効果は持つが，その後の価値推定には残りにくい．結果として，標本平均では Q_0 の違いが学習曲線に大きく反映されず，どの初期値でも似た挙動になったと考えられる．

一方，固定学習率 $\alpha = 0.1$ を用いた場合には，初期値によって学習速度に大きな差が現れている．固定学習率の更新は

$$Q_n(a) = (1-\alpha)^n Q_0(a) + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} r_i$$

と展開できる．この式から分かるように，固定学習率では初期値の影響が $(1-\alpha)^n Q_0(a)$ として残る．したがって，Q_0 を大きくすると，その腕が何度か低い報酬を出しても，しばらくは高く評価され続ける．

図では，$Q_0 = 0.5$ の場合に平均報酬が早く高い値へ到達し，累積後悔も小さくなっている．これは，初期値が大きすぎず，初期の探索を促しつつ，低報酬の腕への過大評価が比較的早く解消されたためだと考えられる．逆に，$Q_0 = 5$ や $Q_0 = 10$ のように初期値が大きすぎると，低報酬の腕も長い間高く評価され続ける．たとえば報酬がしばらく 0 だった場合，固定学習率では

$$Q_n(a) = (1-\alpha)^n Q_0(a)$$

となる．$\alpha = 0.1$ では $(1-\alpha)^n = 0.9^n$ なので，Q_0 が大きいほど，その過大評価が十分に下がるまで多くの試行が必要になる．

以上より，optimistic 初期値は，標本平均よりも固定学習率と組み合わせたときに特徴が現れやすい．標本平均では初期値が一度の観測でほぼ消えてしまうのに対し，固定学習率では初期値の影響がしばらく残るため，初期値の大きさが探索の強さと学習速度を大きく左右する．ただし，初期値は大きければよいわけではなく，小さすぎると探索を促せず，大きすぎると悪い腕への過大評価が長く残る．今回の結果は，optimistic 初期値もまた，問題の報酬スケールに合わせて設定する必要があることを示している．

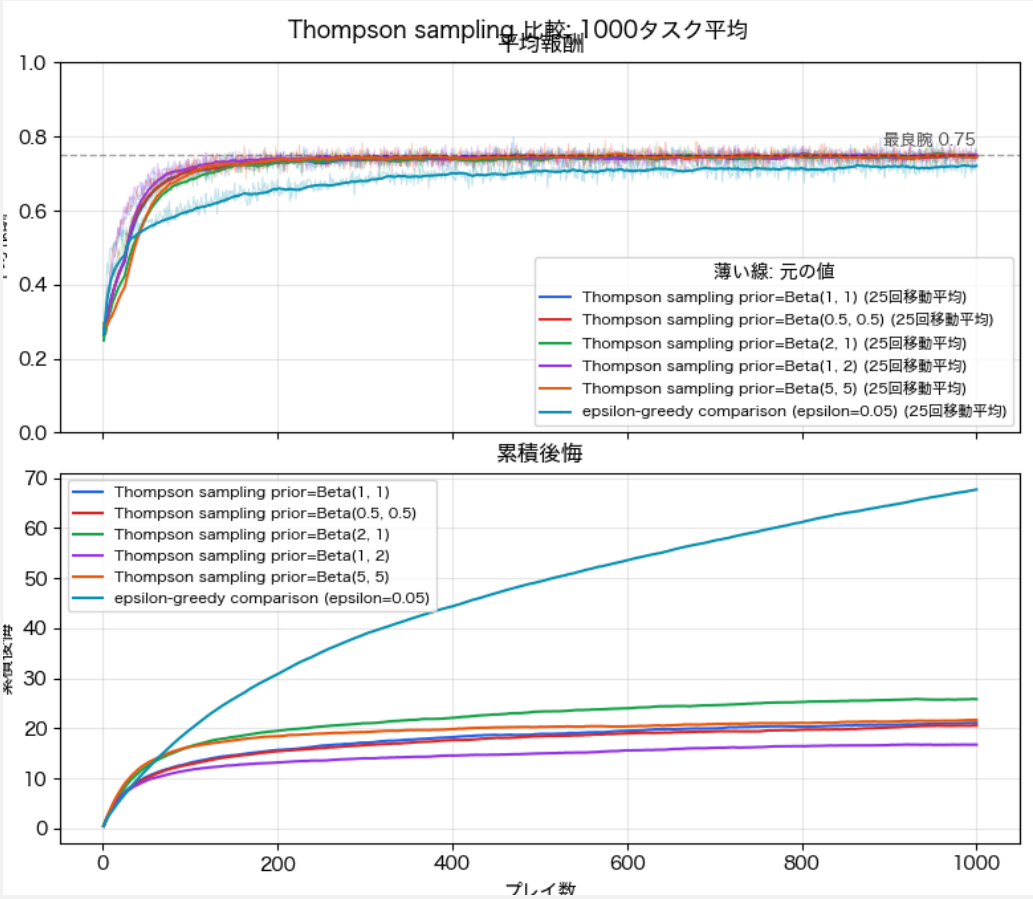


図 5.10: Thompson sampling の学習曲線

Thompson sampling

最後に，Thompson sampling による行動選択を比較する．Thompson sampling では，各腕の報酬確率そのものを一点推定するのではなく，その不確実性を確率分布として保持する．各時刻では，腕 a の報酬確率に関する事後分布から値をサンプリングし，そのサンプル値が最大となる腕を選ぶ．

今回の実験では報酬が Bernoulli(p_a) に従うため，各腕の成功確率 p_a に対してベータ分布を用いる．時刻 t における腕 a の事後分布を

$$p_a \sim \mathrm{Beta}(\alpha_a, \beta_a)$$

とし，そこから

$$\tilde{p}_a \sim \mathrm{Beta}(\alpha_a, \beta_a)$$

をサンプリングして，

$$A_t = \arg\max_{a \in \mathcal{A}} \tilde{p}_a$$

により腕を選ぶ．報酬 1 が得られた場合は α_a を増やし，報酬 0 が得られた場合は β_a を増やす．したがって，Thompson sampling では，探索は外から加えるランダム行動ではなく，**報酬確率に関する不確実性**から自然に生じる．

図 5.10 では，事前分布の違いによる学習曲線の変化を比較している．ベータ分布 $\mathrm{Beta}(\alpha, \beta)$ の平均は

$$\mathbb{E}[p] = \frac{\alpha}{\alpha + \beta}$$

であり，$\alpha + \beta$ が大きいほど事前分布は強くなる．つまり，(α, β) は単なる初期値ではなく，「はじめにどの程度の成功・失敗を仮想的に見たことにするか」を表している．

結果を見ると，Thompson sampling は全体として ε-greedy の比較対象よりも累積後悔が小さく，平均報酬も早く高い値に近づいている．これは，各腕の不確実性を分布として持つことで，まだ情報が少ない腕には自然に探索の余地が残り，成功が多く観測された腕には選択が集まりやすくなるためである．

事前分布の違いに注目すると，$\mathrm{Beta}(1, 2)$ が特に小さい累積後悔を示している．この事前分布の平均は $1/3$ であり，成功確率をやや低めに見積もる悲観的な事前分布である．今回の問題では低報酬の腕が多いため，成功確率をやや低く見る事前分布は，低報酬の腕を早めに見切る方向に働きやすい．その一方で，第 6 腕のように成功が多く観測される腕では，事後分布がすぐに高い値へ移動するため，選択が集中しやすい．

一方，$\mathrm{Beta}(2, 1)$ は平均 $2/3$ を持つ楽観的な事前分布である．この場合，各腕の成功確率を初期には高めに見積もるため，低報酬の腕についても「本当は良い腕かもしれない」という不確実性が残りやすい．その結果，低報酬の腕を見切るまでにやや時間がかかり，累積後悔が大きくなりやすい．これは，optimistic 初期値において初期値を大きくしすぎると悪い腕への過大評価が長く残ることと似ている．

また，$\mathrm{Beta}(0.5, 0.5)$ は U 字型の分布であり，成功確率が 0 付近または 1 付近にある可能性を比較的大きく見る事前分布である．そのため，初期にはサンプル値が極端になりやすく，不確実性に基づく探索がやや強く出る．一方，$\mathrm{Beta}(5, 5)$ は平均 $1/2$ の周辺に集中した事前分布であり，初期にはどの腕も中程度の成功確率を持つという強めの信念を置く．この場合，観測が少ない段階では事前分布の影響が残り，第 6 腕の優位性を反映するまでに少し時間がかかる．

この結果から，Thompson sampling における事前分布は，単なる初期設定ではなく，探索の性質そのものを決める要素であることが分かる．中立的な事前分布では観測に素直に反応し，楽観的な事前分布では探索が長く残りやすく，悲観的な事前分布では低報酬の腕を早く見切りやすい．ただし，悲観的すぎる事前分布では本当に良い腕を十分に試す前に候補から外してしまう危険もあるため，事前分布も問題の報酬スケールに依存して調整される．

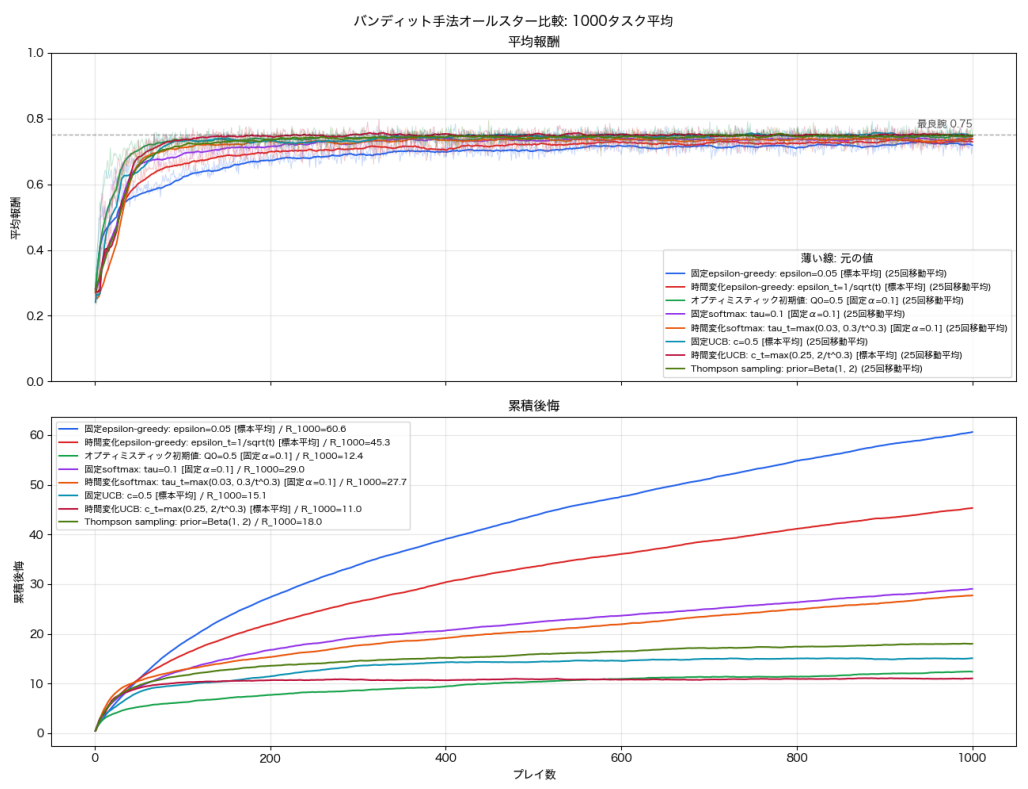


図 5.11: 全手法の，最も累積後悔が小さくなるようなパラメータでの学習曲線

手法間の比較とまとめ

最後に，各手法について今回の問題で比較的よい性能を示したパラメータを選び，同一の図で比較した．図 5.11 を見ると，平均報酬の推移だけでは，多くの手法が最終的に最適腕の期待報酬 0.75 付近へ近づいており，大きな差は見えにくい．しかし，累積後悔を見ると，各手法の違いはかなり明確に現れる．これは，平均報酬が「最終的にどこまで到達したか」を見る指標であるのに対し，累積後悔は「そこに至るまでにどれだけ損をしたか」を記録する指標だからである．

今回の結果では，optimistic 初期値，時間変化する UCB，固定 UCB が特に小さい累積後悔を示している．optimistic 初期値は，初期価値を高く設定することで未試行の腕を自然に試させるため，初期の探索を素早く行うことができる．今回のように定常的で，かつ最適腕と次善腕の差が比較的大きい問題では，一度よい腕を見つければそのまま活用しやすく，累積後悔が小さくなったと考えられる．また UCB は，試行回数の少ない腕に探索ボーナスを与えるため，ランダムに探索するのではなく，不確実な腕を体系的に試すことができる．その結果，探索の無駄が少なく，早い段階で第 6 腕へ選択を集中できたと解釈できる．

一方で，固定 ε-greedy は累積後悔が大きくなっている．これは，最適腕を見つけた後も確

率 ε でランダム探索を続けるためである．学習が十分に進んだ後の平均報酬は

$$(1-\varepsilon)p^* + \varepsilon \frac{1}{K}\sum_{a=1}^{K} p_a$$

で近似されるため，$\varepsilon > 0$ である限り，低報酬の腕を引く損失が残る．時間変化する ε_t-greedy は，初期には探索し，後半には ε_t を小さくすることでこの問題を緩和している．

softmax でも，固定温度では学習後も確率的な選択が残る．温度 τ が大きいと探索が残りすぎ，温度が小さいと初期の推定誤差に過敏になる．今回の比較では，時間変化する softmax は固定 softmax よりよい結果を示しており，温度を下げながら探索から活用へ移行することの有効性が見られる．ただし，softmax は価値差を指数関数で選択確率に変換するため，適切な温度スケールは報酬確率差に強く依存する．今回うまくいった温度が，他の報酬スケールでもそのまま有効とは限らない．

Thompson sampling も比較的良い結果を示している．探索率や温度を外から指定するのではなく，各腕の報酬確率に関する事後分布からサンプリングすることで探索が生じるため，明示的な探索率を設定しなくても，探索と活用のバランスを比較的うまく取ることができたと考えられる．

ただし，ここまでの結果は**定常バンディット問題**における比較である．各腕の真の報酬確率 p_a が時刻によらず一定であるという仮定のもとでは，過去の観測を蓄積することに意味があり，標本平均，UCB，Thompson sampling のように，観測回数が増えるほど推定を確信していく手法が有効に働きやすい．しかし，非定常環境では古い観測が現在の環境を表さない可能性がある．そのため，過去の全データを同じ重みで用いる標本平均手法は，変化後の最適腕に追従しにくい．

この観点から見ると，定常環境で特に意味を持つ手法と，非定常環境でも活かしやすい手法を分けて考える必要がある．標本平均に基づく greedy，UCB，通常の Thompson sampling は，基本的には「真の報酬確率が一定であり，観測を重ねるほど正しい値に近づく」という考え方に支えられている．したがって，定常環境では強いが，非定常環境では古い情報を信じすぎる危険がある．特に UCB では，ある腕を十分に試した後は探索ボーナス

$$c\sqrt{\frac{\log t}{N_t(a)}}$$

が小さくなるため，環境変化後にその腕を再び探索する力は弱くなる．Thompson sampling でも，観測数が増えて事後分布が鋭くなると，過去の信念が強く残り，新しい変化に反応しにくくなる場合がある．

一方，固定学習率による価値更新や，探索率・温度・UCB 係数を時間変化させる設計は，非定常環境へ拡張する余地を持つ．固定学習率では

$$Q_n = (1-\alpha)^n Q_0 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} r_i$$

となり，最近の報酬ほど大きな重みを持つ．したがって，古い観測を徐々に忘れながら，新しい環境に追従しやすい．同様に，非定常環境では探索率を完全には下げきらず，一定の探索を残すことにも意味がある．定常環境では不要な探索は後悔を増やすが，非定常環境では，環境変化を発見するための観測機会になるからである．

以上の比較から，多腕バンディット問題における手法の良し悪しは，単に平均報酬の高さだけでは判断できないことが分かる．それぞれの違いは，最適腕をどれだけ早く見つけるか，低報酬の腕をどれだけ早く見切るか，そして学習後に不要な探索をどれだけ残すかという形で，累積後悔の曲線に現れる．

本実験を通じて，多腕バンディット問題は，強化学習における探索と活用を理解するための最小構成の実験場であることが確認できた．状態遷移を持たない単純な問題であっても，探索率，温度，探索ボーナス，初期値，事前分布，学習率の選び方によって，学習曲線は大きく変化する．したがって，より複雑な強化学習問題を扱う際にも，アルゴリズムを名前で選ぶのではなく，その手法が**何を不確実とみなし，どのように探索し，どの程度過去を信じるのか**を考えることが重要である．

5.3　拡散モデルと生成モデル

本章では，生成モデルを，観測データの分布あるいは条件付き分布を学習し，そこから新しいサンプルを生成する確率モデルとして統一的に捉える。画像生成や文生成の成功は広く知られているが，重要なのは，生成モデルが単なる出力装置ではなく，潜在変数を通じた表現学習，尤度に基づく学習原理，近似推論，逐次生成，さらには予測と計画へ接続される一般的な枠組みであるという点にある。

本章の目標は三つある。第一に，潜在変数モデルの学習という観点から，EM アルゴリズムと変分オートエンコーダ（variational autoencoder; VAE）を位置づける。第二に，ノイズ除去過程としての拡散モデルを，離散時間から連続時間へと見通しよく整理する。第三に，自己回帰モデルと世界モデルを通じて，生成モデルが系列生成や意思決定支援へどのように拡張されるかを理解する。以下では，「潜在変数モデルの学習」「逐次生成」「ノイズ除去過程」という三つの視点が一貫して現れるように議論を進める。

5.3.1　生成モデルの考え方

この概念が必要になるのは，分類や回帰だけでは扱えない問いが多く存在するからである。たとえば欠損補完，異常検知，反実仮想的なシミュレーション，都市活動の需要分布の生成，交通軌跡の合成などでは，入力に対してラベルを当てるだけでなく，データそのものがどのような確率法則に従って生じるかを記述する必要がある。生成モデルは，この要請に対して，観測 x の分布を直接あるいは間接に学習する立場をとる。

(a)　分布を学ぶとは何か

識別モデル（discriminative model）は，典型的には入力 x からラベル y への条件付き分布 $p_\theta(y \mid x)$ を学習する。ここで θ は学習すべきパラメータである。これに対して生成モデル（generative model）は，観測データ x の分布 $p_\theta(x)$ を学ぶか，あるいは条件情報 c が与えられたときの条件付き分布 $p_\theta(x \mid c)$ を学ぶ。記号 $p_{\text{data}}(x)$ は真のデータ分布を表し，学習の目的は $p_\theta(x)$ を $p_{\text{data}}(x)$ に近づけることである。条件付き生成では，たとえば文脈 c に応じて文章や画像や将来軌跡を生成するために $p_\theta(x \mid c)$ を考える。

生成モデルを学ぶ最も自然な原理は最大尤度推定（maximum likelihood estimation）である。独立同分布に従う観測データ $x^{(1)}, \dots, x^{(N)}$ が与えられたとき，尤度は

$$\prod_{n=1}^{N} p_\theta(x^{(n)}) \tag{5.81}$$

であり，対数尤度は

$$\sum_{n=1}^{N} \log p_\theta(x^{(n)}) \tag{5.82}$$

となる。ここで N は標本数である。対数をとると積が和に変わり，最適化や解析が行いやすくなる。したがって，生成モデルの中心課題は，対数尤度を大きくする θ を求めることだといえる。

しかし，この方針は概念的には自然であっても，計算の上では難しい。高次元データに対して $p_\theta(x)$ を直接評価すること自体が困難であり，とくに潜在変数を導入したモデルでは周辺化が障害となる。以下で見るように，EM アルゴリズム，VAE，拡散モデル，自己回帰モデルは，いずれもこの困難に対する異なる解法として理解できる。

(b) 潜在変数モデル

生成モデルで潜在変数（latent variable）z を導入する主な理由は，観測 x の背後にある低次元あるいは構造化された要因を分離し，表現学習を可能にするためである。潜在変数モデルの基本形は

$$p_\theta(x) = \int p_\theta(x \mid z)\, p(z)\, dz \tag{5.83}$$

で与えられる。ここで $p(z)$ は潜在変数 z の事前分布，$p_\theta(x \mid z)$ は z が与えられたときの観測 x の条件付き分布であり，しばしばデコーダとも呼ばれる。積分は，潜在変数 z を明示的には観測しないまま，その全ての可能性を足し合わせる周辺化を表す。z が離散変数なら積分は総和に置き換わる。

この式は，観測データの複雑な分布を，潜在空間での比較的単純な分布と条件付き生成機構の組合せとして表現することを意味する。たとえば顔画像なら，姿勢，照明，表情といった要因を z に担わせることが期待される。交通軌跡なら，目的地，出発時刻帯，混雑回避傾向などを潜在表現として持たせることができる。潜在変数は観測されないため，その推論が必要になるが，その代わりにデータ分布の説明力と表現の可搬性が得られる。

問題は，周辺尤度 $p_\theta(x)$ の計算が難しいことである。とくに $p_\theta(x \mid z)$ をニューラルネットワークで表すと，積分は解析的に解けないことが多い。また，学習中に必要な事後分布 $p_\theta(z \mid x)$ も容易には得られない。したがって，生成モデルにおける中心課題は，潜在変数を含む尤度最大化と近似推論をどのように両立させるかにある。

(c) 本章の見取り図

生成モデルには複数の流儀がある。敵対的生成ネットワーク（generative adversarial network; GAN）は識別器とのゲームを通じてサンプル品質を高める方法であり，自己回帰モデルは結合分布を条件付き分布の積に分解して逐次生成を行う。VAE は潜在変数モデルの学習を変分推論で実現し，拡散モデルはノイズ付加とノイズ除去の逐次過程として生成を定式化する。

本章では，まず潜在変数モデルを学習する古典的な方法として EM アルゴリズムを扱い，次に厳密事後分布が扱えない深層モデルへ進むために VAE を導入する。その後，ノイズ除去過程としての拡散モデルを離散時間と連続時間で整理し，最後に逐次生成の別系統として自己回帰モデルと LLM を概観する。さらに，生成モデルが表現学習と予測を通じて環境理解へ拡張される例として世界モデルを取り上げる。これにより，生成モデルを，単なるサンプル生成ではなく，潜在表現の獲得と確率的予測の一般原理として捉えられるようになる。

5.3.2 EM アルゴリズム

この概念が必要になるのは，潜在変数を含むモデルでは，尤度最大化がしばしば直接には解けないからである。観測データだけを見れば単純な最適化問題に見えても，実際には観測されていない変数を周辺化した対数尤度が現れ，対数と積分が絡み合うために計算が難しくなる。EM アルゴリ

ズム（expectation-maximization algorithm）は，この困難を「潜在変数の推定」と「パラメータの更新」に分けることで扱う。

(a)　完全データ尤度と不完全データ尤度

潜在変数を z，観測を x とすると，完全データ（complete data）は (x, z) であり，不完全データ（incomplete data）は x のみである。完全データの対数尤度

$$\log p(x, z \mid \theta) \tag{5.84}$$

は，モデルによっては扱いやすい。一方，不完全データの対数尤度は

$$\log p(x \mid \theta) = \log \int p(x, z \mid \theta)\, dz \tag{5.85}$$

であり，積分の外側に対数があるため，直接最適化しにくい。ここで θ はモデルパラメータであり，$p(x, z \mid \theta)$ は同時分布である。難しさの本質は，対数 $\log$ が線形ではないため，潜在変数の周辺化と素直に交換できないことにある。

この問題を緩和するために，任意の分布 $q(z)$ を導入して

$$\log p(x \mid \theta) = \log \int q(z) \frac{p(x, z \mid \theta)}{q(z)}\, dz \tag{5.86}$$

と書く。ここで $q(z)$ は潜在変数 z に関する補助的な分布であり，後で適切に選ぶ。対数関数は凹関数であるから，Jensen の不等式より

$$\log p(x \mid \theta) \geq \int q(z) \log \frac{p(x, z \mid \theta)}{q(z)}\, dz \tag{5.87}$$

が成り立つ。右辺は θ に関する下界であり，しばしば自由エネルギー（free energy）あるいは変分下界と呼ばれる。

(b)　EM の導出

上の下界を

$$\mathcal{F}(q, \theta) = \mathbb{E}_{q(z)}[\log p(x, z \mid \theta)] + H(q) \tag{5.88}$$

と書くことにする。ここで $\mathbb{E}_{q(z)}[\cdot]$ は分布 $q(z)$ に関する期待値であり，$H(q) = -\mathbb{E}_{q(z)}[\log q(z)]$ はエントロピーである。この量は

$$\log p(x \mid \theta) = \mathcal{F}(q, \theta) + \mathrm{KL}\big(q(z) \,\|\, p(z \mid x, \theta)\big) \tag{5.89}$$

と分解できる。右辺第二項の $\mathrm{KL}(\cdot\|\cdot)$ は Kullback–Leibler ダイバージェンスであり，二つの分布のずれを測る非負量である。したがって，$\mathcal{F}(q, \theta)$ は常に対数尤度の下界であり，しかも $q(z) = p(z \mid x, \theta)$ のときに下界は厳密に一致する。

EM アルゴリズムはこの事実を用いて，θ と q を交互に更新する。現在のパラメータを θ^{old} とすると，E-step では

$$q^{\mathrm{new}}(z) = p(z \mid x, \theta^{\mathrm{old}}) \tag{5.90}$$

と置く。これにより，下界は現在のパラメータに対して最もきつくなる。続いて M-step では，この $q^{\text{new}}(z)$ を固定して $\mathcal{F}(q^{\text{new}}, \theta)$ を θ について最大化する。エントロピー項 $H(q^{\text{new}})$ は θ に依らないため，M-step は

$$Q(\theta \mid \theta^{\text{old}}) = \mathbb{E}_{p(z \mid x, \theta^{\text{old}})}[\log p(x, z \mid \theta)] \tag{5.91}$$

を最大化する問題になる。ここで $Q(\theta \mid \theta^{\text{old}})$ は補助関数（auxiliary function）であり，古いパラメータで計算した事後分布の下で完全データ対数尤度の期待値をとったものである。

以上をアルゴリズムとしてまとめると，EM は次の二段階を反復する。

1. E-step: 現在のパラメータ θ^{old} の下で事後分布 $p(z \mid x, \theta^{\text{old}})$ を計算する。
2. M-step: 補助関数 $Q(\theta \mid \theta^{\text{old}})$ を最大化して新しいパラメータ θ^{new} を得る。

この反復により，不完全データ対数尤度は単調非減少となる。EM が有効であるためには，E-step で厳密な事後分布を計算でき，M-step でも十分に扱いやすい最適化ができることが望ましい。

(c)　GMM における EM の直感

混合ガウスモデル（Gaussian mixture model; GMM）は EM の典型例である。K 個のクラスタを考え，観測 x_n がクラスタ k に属する確率を混合係数 π_k で表す。潜在変数 z_n はデータ点 x_n のクラスタ割当を表し，$z_n = k$ のとき

$$p(x_n \mid z_n = k, \theta) = \mathcal{N}(x_n; \mu_k, \Sigma_k) \tag{5.92}$$

とする。ここで μ_k はクラスタ k の平均，Σ_k は共分散行列である。

E-step では，各データ点が各クラスタに属する責務（responsibility）

$$\gamma_{nk} = p(z_n = k \mid x_n, \theta^{\text{old}}) = \frac{\pi_k \mathcal{N}(x_n; \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n; \mu_j, \Sigma_j)} \tag{5.93}$$

を計算する。責務 γ_{nk} は，データ点 x_n がクラスタ k にどれだけ説明されるかを表す重みである。M-step では，この重み付き平均として

$$N_k = \sum_{n=1}^{N} \gamma_{nk}, \quad \pi_k = \frac{N_k}{N}, \quad \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n \tag{5.94}$$

および

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k)(x_n - \mu_k)^{\top} \tag{5.95}$$

を更新する。ここで N_k はクラスタ k に割り当てられた有効サンプル数である。

この例から分かるように，EM は「見えない割当を現在のモデルで推定し，その推定結果に基づいてモデルを更新する」反復である。したがって，EM は潜在変数モデルの最も基本的な学習原理の一つといえる。ただし，深層生成モデルでは事後分布 $p_\theta(z \mid x)$ を厳密に計算できないことが多い。その場合には，厳密事後分布を用いる EM の代わりに，近似事後分布を学習する変分推論が必要になる。次節では，この考え方をニューラルネットワークに組み込んだ VAE を扱う。

5.3.3 VAE

この概念が必要になるのは，深層潜在変数モデルでは真の事後分布 $p_\theta(z \mid x)$ がほとんどの場合に計算不可能だからである。EM は厳密な事後分布を扱えるときに強力だが，ニューラルネットワークを含む柔軟な生成モデルでは，E-step そのものが閉形式で書けない。VAE[16, 23] は，近似事後分布を別のパラメータで学習することにより，生成モデルの尤度最大化を実用的な最適化問題へ変換する。

(a)　変分推論と ELBO

VAE では，真の事後分布 $p_\theta(z \mid x)$ の代わりに，近似事後分布 $q_\phi(z \mid x)$ を導入する。ここで ϕ はエンコーダのパラメータであり，観測 x から潜在変数 z の分布を出力する。生成モデルは依然として

$$p_\theta(x, z) = p_\theta(x \mid z) p(z) \tag{5.96}$$

と書かれ，$p(z)$ は事前分布である。任意の $q_\phi(z \mid x)$ に対して，

$$\log p_\theta(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x, z)}{q_\phi(z \mid x)}\right] + \mathrm{KL}\big(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x)\big) \tag{5.97}$$

が成り立つ。右辺第二項は非負であるから，

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - \mathrm{KL}(q_\phi(z \mid x) \| p(z)) \tag{5.98}$$

を得る。右辺がエビデンス下界（evidence lower bound; ELBO）である。ここで第一項は再構成項，第二項は KL 正則化項である。

再構成項

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] \tag{5.99}$$

は，潜在変数 z から元の観測 x をどれだけよく説明できるかを測る。これはオートエンコーダにおける復元誤差に相当するが，確率的な形で定式化されている。一方，KL 正則化項

$$\mathrm{KL}(q_\phi(z \mid x) \| p(z)) \tag{5.100}$$

は，データごとの近似事後分布を共通の事前分布 $p(z)$ に近づける。これにより，潜在空間が無秩序に散らばることを防ぎ，サンプリング可能な構造を持たせる。VAE の学習は，この二つの要請，すなわち「よく再構成すること」と「潜在空間を整えること」の均衡である。

(b)　再パラメータ化トリックと学習

ELBO を確率的勾配法で最適化するには，$q_\phi(z \mid x)$ からサンプルした z を通じて ϕ に勾配を流す必要がある。しかし，単に「分布からサンプルする」と書くと，サンプリング操作が非微分的に見える。そこで再パラメータ化トリック（reparameterization trick）を用いる。

近似事後分布を平均 $\mu_\phi(x)$ と標準偏差 $\sigma_\phi(x)$ を持つ対角ガウス分布

$$q_\phi(z \mid x) = \mathcal{N}\big(z; \mu_\phi(x), \mathrm{diag}(\sigma_\phi(x)^2)\big) \tag{5.101}$$

とすると，潜在変数 z を

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{5.102}$$

と書ける。ここで $\odot$ は要素ごとの積，I は単位行列，ϵ は標準正規分布からの補助ノイズである。重要なのは，確率性を ϵ に押し込み，$\mu_\phi(x)$ と $\sigma_\phi(x)$ については決定論的な変換として z を表した点にある。これにより，ϕ に関する勾配を，サンプル平均を通じて安定に推定できる。

事前分布を $p(z) = \mathcal{N}(0, I)$ とし，近似事後分布を上の対角ガウスとした場合，KL 項は解析的に計算できる。潜在次元を $j = 1, \dots, d$ とすると，

$$\mathrm{KL}(q_\phi(z \mid x) \| p(z)) = \frac{1}{2} \sum_{j=1}^{d} \left(\mu_j(x)^2 + \sigma_j(x)^2 - \log \sigma_j(x)^2 - 1 \right) \tag{5.103}$$

である。ここで $\mu_j(x)$ と $\sigma_j(x)$ は潜在変数の第 j 成分に対応する平均と標準偏差である。この閉形式は，VAE の実装を著しく簡潔にする。

(c) 表現学習としての VAE と限界

VAE の重要な意義は，生成と表現学習を同時に行う点にある。潜在空間が事前分布により正則化されるため，近い潜在ベクトルはしばしば意味的に近いサンプルを生成する。したがって，潜在空間上の補間は滑らかな変化として解釈されやすく，表現の連続性が得られる。これは，画像や軌跡や都市形状のような高次元データを低次元の意味表現へ圧縮する上で有用である。

他方で，VAE には典型的な限界もある。第一に，デコーダが強すぎると近似事後分布 $q_\phi(z \mid x)$ が事前分布 $p(z)$ に近づきすぎ，潜在変数がほとんど情報を持たなくなることがある。これは事後崩壊（posterior collapse）と呼ばれる。第二に，尤度を重視した平均化的な学習により，再構成や生成結果がぼけやすい。とくに多峰的な分布を単純なガウス尤度で近似すると，複数の可能性を平均したような出力になりやすい。

それでも VAE は，EM の考え方を深層モデルへ拡張した重要な橋渡しである。EM が厳密事後分布を用いるのに対し，VAE は近似事後分布 $q_\phi(z \mid x)$ 自体を学習する。この違いにより，潜在変数モデルの学習は大規模データと表現学習へ開かれた。次節では，潜在変数の推論ではなく，ノイズ除去の逐次過程として生成を定式化する拡散モデルを扱う。

5.3.4 拡散モデル

この概念が必要になるのは，高品質なサンプル生成と安定な学習を両立させる新しい生成原理が求められたからである。VAE は尤度と潜在表現を扱いやすい一方で出力がぼけやすく，GAN[8] は鋭いサンプルを生成できる一方で学習が不安定になりやすい。拡散モデル（diffusion model）[11] は，データに徐々にノイズを加える前向き過程と，そのノイズを逆向きに除去する生成過程を組み合わせることで，尤度に根ざした学習と高品質生成を接続する。

(a) 前向き過程と閉形式

離散時間の拡散過程では，時刻 $t = 1, \dots, T$ に対して，観測 x_{t-1} に小さなガウス雑音を加えて x_t を得る。前向き過程は

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}\, x_{t-1}, \beta_t I) \tag{5.104}$$

で定義される。ここで $\beta_t \in (0,1)$ は時刻 t のノイズ分散を表すスケジュールであり，I は単位行列である。β_t が小さいほど，1 ステップ当たりの変化は小さい。しばしば $\alpha_t = 1 - \beta_t$ とおく。

この前向き過程を繰り返すと，元のデータ x_0 は徐々に情報を失い，十分大きい T では標準正規分布に近い雑音 x_T へ到達する。重要なのは，累積係数

$$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s \tag{5.105}$$

を用いると，x_0 から任意の時刻 t の状態 x_t を 1 ステップで直接サンプルできることである。具体的には

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1 - \bar{\alpha}_t) I) \tag{5.106}$$

となり，したがって

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{5.107}$$

と書ける。ここで ϵ は標準正規ノイズである。この閉形式は学習上きわめて重要であり，各時刻の中間状態 x_t を逐次シミュレーションせずに直接生成できる。

(b)　逆過程とノイズ予測

生成は前向き過程の逆，すなわち雑音 x_T から出発して徐々にノイズを除去し，最終的にデータ x_0 を得る過程として定式化される。逆過程のモデルを

$$p_\theta(x_{t-1} \mid x_t) \tag{5.108}$$

と書く。ここで θ はニューラルネットワークのパラメータであり，この条件付き分布が真の逆向き遷移を近似するように学習する。前向き過程の各ステップが十分小さいとき，逆向き条件付き分布も近似的にガウスになるため，平均や分散をネットワークで予測する設計が自然になる。

元来，拡散モデルは変分下界に基づいて導出されるが，実用上はノイズそのものを予測する学習目標に簡約されることが多い。すなわち，前向き過程で x_0 から x_t を生成する際に用いたノイズ ϵ を，ネットワーク $\epsilon_\theta(x_t, t)$ に予測させ，

$$\mathcal{L}_{\text{simple}} = \mathbb{E}\big[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\big] \tag{5.109}$$

を最小化する。ここで期待値は，データ x_0，時刻 t，およびノイズ ϵ に関する平均である。損失が平均二乗誤差になるのは，真のノイズを回帰問題として学習する立場をとっているからである。

この三者の関係を明確にしておこう。前向き過程 $q(x_t \mid x_{t-1})$ はデータを徐々に壊す既知の確率過程であり，逆過程 $p_\theta(x_{t-1} \mid x_t)$ は壊れたデータから元の構造を回復する学習対象である。そしてノイズ予測 $\epsilon_\theta(x_t, t)$ は，逆過程を実装するための実用的なパラメータ化である。すなわち，ネットワークが「いま見えている x_t の中にどれだけノイズが混ざっているか」を推定できれば，そこから一段前の状態 x_{t-1} を構成できる。

(c)　条件付き生成と位置づけ

条件付き生成では，条件 c を入力に加えて $\epsilon_\theta(x_t, t, c)$ を学習し，$p_\theta(x \mid c)$ に相当する生成を行う。画像生成ならテキスト条件，交通分野なら出発地，目的地，時刻帯，天候などを条件とできる。

さらに，分類器なしガイダンス（classifier-free guidance）[12] は，条件付き予測と無条件予測を組み合わせて条件への忠実度を調整する手法であり，生成品質と条件整合性の実用的な制御を可能にする。

拡散モデルの要点は，生成を一挙に行うのではなく，容易に記述できる破壊過程と，それを反転する学習可能な回復過程へ分解したことにある。この見方をさらに推し進めると，時間刻みを無限に細かくした連続時間の確率過程へ到達する。次節では，その連続時間化がどのように理論を統一するかを見る。

5.3.5 連続時間拡散モデル

この概念が必要になるのは，離散時間の拡散モデルをより一般的な確率過程として理解すると，理論とアルゴリズムの見通しが大きく改善するからである。離散時間のデノイジング拡散確率モデル（denoising diffusion probabilistic model; DDPM）では時刻 $t = 1, \dots, T$ の格子上でノイズを加えていたが，刻み幅を十分小さくすると，連続時間の確率微分方程式で前向き過程と逆過程を統一的に表現できる。

(a) 前向き SDE と逆時間 SDE

連続時間では，時刻を $t \in [0, T]$ とし，データ分布から出発する確率過程 x_t を確率微分方程式（stochastic differential equation; SDE）

$$dx = f(x, t)\, dt + g(t)\, dw \tag{5.110}$$

で表すことが多い。ここで $f(x, t)$ はドリフト項，$g(t)$ は拡散係数，w は標準 Wiener 過程である。ドリフト項は決定論的な流れを，拡散項はランダムなノイズ注入を表す。前向き SDE（forward SDE）は，時刻とともに分布を単純化し，最終的に扱いやすい既知分布へ近づける過程に対応する。

このとき，各時刻における周辺分布を $p_t(x)$ と書く。$p_t(x)$ は時刻 t での x_t の確率密度であり，初期時刻 $t = 0$ ではデータ分布，終端時刻 $t = T$ ではほぼガウス雑音になるように設計される。Song [30] が示した重要な結果によれば，適切な正則性の下で，時間を逆向きにたどる逆時間 SDE（reverse-time SDE）は

$$dx = \left[f(x, t) - g(t)^2 \nabla_x \log p_t(x) \right] dt + g(t)\, d\bar{w} \tag{5.111}$$

と書ける。ここで $\nabla_x \log p_t(x)$ は時刻 t の分布の対数密度勾配であり，スコア関数（score function）と呼ばれる。また，$\bar{w}$ は逆向き時間に対応する Wiener 過程である。つまり，逆過程を記述する鍵は，各時刻の密度そのものではなく，その勾配である。

(b) スコア関数とスコアベース生成

スコア関数

$$\nabla_x \log p_t(x) \tag{5.112}$$

は，空間の各点で「密度がどちらの方向に増えるか」を示すベクトル場である。スコアベース生成モデル（score-based generative modeling）[29] は，密度 $p_t(x)$ を明示的にモデル化する代わりに，このスコア関数をニューラルネットワーク $s_\theta(x, t)$ で近似する立場をとる。密度そのものを正規化定

数まで含めて求めるのは難しいが，対数密度勾配は学習しやすい場合があり，しかも逆時間 SDE の構成にはそれで十分である。

スコアマッチング（score matching）は，この勾配場を直接学習する枠組みである。とくにノイズを加えたデータに対するデノイジング・スコアマッチングでは，ノイズレベルごとのスコアを学習し，それを用いて逆向きにサンプリングする。離散時間の拡散モデルでノイズ ϵ を予測する手法は，連続時間のスコア推定と密接に対応している。実際，ガウス摂動の下ではノイズ予測とスコア推定は定数倍の違いを除いて等価になる。

(c)　離散時間との対応と統一的理解

連続時間化の意義は，離散時間モデルと別物の理論を導入することではなく，それらを一つの枠組みへ包摂することにある。離散時間の DDPM で用いた β_t スケジュールは，連続時間では拡散係数 $g(t)$ に対応し，逆過程の平均推定はスコア関数によるドリフト補正として理解できる。したがって，DDPM は連続時間 SDE の時間離散化の一つとみなせる。

さらに，reverse-time SDE と同じ周辺分布を持つ常微分方程式

$$\frac{dx}{dt} = f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x) \tag{5.113}$$

を考えることもでき，これは確率流常微分方程式（probability flow ODE）と呼ばれる。ここで常微分方程式はノイズ項を持たない決定論的な流れであり，尤度計算や高速サンプリングの解析に有用である。もっとも，本質は，連続時間化によって「ノイズ付加」「スコア推定」「逆過程サンプリング」が統一的に記述できるようになった点にある。

ここまでの拡散モデルは，ノイズ除去に基づく逐次生成であった。これに対して，系列の自然な順序に従って条件付き分布を直接学習する立場も存在する。次節では，言語モデルを代表例とする自己回帰生成を扱う。

5.3.6　自己回帰モデルと LLM

この概念が必要になるのは，系列データにはしばしば自然な因果順序があり，その順序に沿って結合分布を厳密に分解できるからである。文，音声，行動列，交通需要の時系列は，先行する要素が後続の要素に情報を与える。こうした場合には，潜在変数を積分消去するよりも，条件付き分布そのものを逐次的に直接学習する方が自然である。

(a)　連鎖律による生成

自己回帰モデル（autoregressive model）は，長さ T の系列 $x_1, \dots, x_T$ の結合分布を確率の連鎖律により

$$p(x_1, \dots, x_T) = \prod_{t=1}^{T} p(x_t \mid x_{<t}) \tag{5.114}$$

と分解する。ここで $x_{<t}$ は時刻 t より前の要素 $x_1, \dots, x_{t-1}$ をまとめて表す。左辺の結合分布を直接扱う代わりに，各時刻の条件付き分布を学べばよいという点がこの枠組みの核心である。

学習は通常，教師ありの次トークン予測（next-token prediction）として行われる。すなわち，系列全体の対数尤度

$$\log p(x_{1:T}) = \sum_{t=1}^{T} \log p(x_t \mid x_{<t}) \tag{5.115}$$

を最大化する。ここで $x_{1:T}$ は系列全体を表す略記である。自己回帰モデルでは，各条件付き分布が明示的に規格化されているため，尤度計算が比較的直接的である。その代わり，生成時には x_1 から順に 1 トークンずつサンプルしなければならず，本質的に逐次的である。

(b) n-gram から Transformer へ

自己回帰言語モデルの古典的な形は n-gram であり，現在の単語の確率を直前 $n-1$ 個の単語に条件づけて推定する。これは有限長文脈しか見られないという制約を持つが，連鎖律に基づく生成という発想自体はすでに備えている。次に再帰型ニューラルネットワーク（recurrent neural network; RNN）による言語モデルが現れ，隠れ状態に過去の情報を圧縮することで可変長文脈を扱えるようになった。

さらに，Transformer[36] は自己注意機構により，系列内の離れた位置同士の依存関係を効率的に表現できるようにした。マスク付き自己注意を用いれば，将来のトークンを見ずに $p(x_t \mid x_{<t})$ を学習できる。大規模言語モデル（large language model; LLM）は，この自己回帰 Transformer を大規模データと大規模計算資源で学習したものであり，本質的には巨大な条件付き分布推定器である。

LLM において，入力は通常，単語そのものではなく離散トークン列へ変換される。モデルはトークン埋め込みと位置情報を用いて文脈を表現し，次トークンの条件付き分布を出力する。パラメータ数とデータ量と計算量を増やすと性能が系統的に向上するというスケーリング則 [15] は，自己回帰モデルが高い表現力を持つことを示している。また，文脈内に与えられた例から振る舞いを調整するインコンテキスト学習（in-context learning）は，明示的なパラメータ更新なしに条件付き分布を柔軟に活用している現象として理解できる。

(c) 拡散モデルとの対比

自己回帰モデルと拡散モデルは，どちらも逐次生成を行うが，その逐次性の意味が異なる。自己回帰モデルでは，順方向の因果分解に基づいて $x_1, x_2, \ldots$ を一つずつ生成する。これに対して拡散モデルでは，ノイズの多い状態からノイズの少ない状態へと反復的に洗練する。前者は条件付き分布 $p(x_t \mid x_{<t})$ を直接学ぶ枠組みであり，後者は逆過程 $p_\theta(x_{t-1} \mid x_t)$ を学ぶ枠組みである。

この違いは，潜在変数や近似推論との関係にも表れる。自己回帰モデルは，潜在変数を明示せずに高次元分布を分解して扱う。一方，VAE や世界モデルでは潜在表現が前面に現れ，拡散モデルではノイズ状態という補助変数が逐次的に導入される。どの立場をとるかは，対象データの構造，必要とされる尤度計算，生成速度，条件付けの仕方に依存する。次節では，生成モデルがさらに一歩進み，環境の内部遷移そのものを学習する世界モデルへ向かう。

5.3.7 世界モデル

この概念が必要になるのは，生成モデルの役割が単なるサンプル生成にとどまらず，予測，計画，制御へ広がるからである。エージェントが環境中で行動するとき，本当に必要なのは「もっともらしい画像を生成すること」だけではない。観測の背後にある状態を圧縮して表現し，行動に応じて将来がどう変化するかを内部でシミュレーションできることが重要である。世界モデル（world model）は，この内部シミュレータを生成モデルとして構成する。

(a)　潜在動的モデルの基本形

時刻 t における観測を o_t，潜在状態を z_t，行動を a_t とする。世界モデルの基本構図は，観測列の背後に低次元の潜在状態列を仮定し，その時間発展を学習することである。典型的には

$$p(z_1)\,p(o_1 \mid z_1)\prod_{t=2}^{T} p(z_t \mid z_{t-1}, a_{t-1})\,p(o_t \mid z_t) \tag{5.116}$$

という形の潜在動的モデルを考える。ここで $p(z_t \mid z_{t-1}, a_{t-1})$ は遷移モデル，$p(o_t \mid z_t)$ は観測モデルである。遷移モデルは行動 a_{t-1} を受けて潜在状態がどう変わるかを記述し，観測モデルは潜在状態からセンサ入力や画像や計測値がどう生成されるかを表す。

この構図は，VAE の潜在変数モデルを時間方向へ拡張したものと見なせる。観測 o_t をそのまま次時刻へ渡すのではなく，まず圧縮された潜在表現 z_t を介して扱うことで，高次元観測の冗長性を除きつつ，力学的に意味のある状態遷移を学習できる。表現学習と動的予測が結合している点が世界モデルの本質である。

(b)　RSSM と想像上ロールアウト

実際のモデルでは，確率的な潜在状態だけでなく，決定論的な再帰状態を併用することが多い。たとえば再帰状態空間モデル（recurrent state-space model; RSSM）は，確率的潜在変数と RNN 的な隠れ状態を組み合わせ，部分観測環境でも十分な履歴情報を保持できるようにする。この種のモデルは潜在力学モデル（latent dynamics model）と総称され，モデルベース強化学習（model-based reinforcement learning）において重要な役割を果たしている。

学習された遷移モデルがあれば，実環境で行動せずとも，潜在空間内で将来軌跡を展開できる。これが想像上ロールアウト（imagined rollout）である。具体的には，現在の潜在状態推定 z_t から，候補行動列 $a_t, a_{t+1}, \ldots$ を仮定して

$$z_{t+1}, z_{t+2}, \ldots \tag{5.117}$$

を遷移モデルで生成し，必要なら観測や報酬もデコードする。これにより，エージェントは現実世界で高価な試行錯誤を行う前に，内部モデル上で計画や方策評価を行える。交通シミュレーションでいえば，仮想的な需要変動や信号制御の変更が将来流動へ与える影響を，学習済み潜在力学上で検討することに相当する。

(c)　生成モデルから意思決定支援へ

世界モデルは，生成モデルの役割を三方向へ拡張する。第一に，観測の生成という意味で，依然として確率的なデコーダを持つ。第二に，潜在状態の学習という意味で，表現学習の装置でもある。第三に，行動条件付きの将来予測を通じて，計画と意思決定を支援する。ここでは生成が，見た目のサンプル合成ではなく，「もしこの行動をとったら何が起こるか」を内的に予測する能力へ転化している。

したがって，世界モデルは，本章で見てきた諸概念の自然な統合点である。潜在変数，近似推論，逐次生成，条件付き分布，表現学習がすべて現れるからである。生成モデルは，データ分布を学ぶ技術から出発しながら，最終的には環境の理解と行動選択を支える内部表現へと発展する。本章の締めくくりとして，この拡張可能性を意識しておくことが重要である。

まとめ

本章では，生成モデルを，データ分布の学習という統一的な視点から整理した。まず，識別モデルとの対比を通じて，生成モデルが $p_\theta(x)$ や $p_\theta(x \mid c)$ を学び，新しいサンプルを生成するだけでなく，欠損補完や異常検知や合成データ生成に資する確率モデルであることを確認した。その上で，潜在変数モデル

$$p_\theta(x) = \int p_\theta(x \mid z)\, p(z)\, dz \tag{5.118}$$

を導入し，複雑な観測分布を背後の表現と生成機構へ分解することの意味を見た。

続いて，潜在変数を含む尤度最大化の困難に対して，EM アルゴリズムが厳密事後分布を用いて下界最大化を行う方法であることを示し，さらに VAE が近似事後分布 $q_\phi(z \mid x)$ を学習することで，深層潜在変数モデルへこの考え方を拡張することを見た。ここでは，尤度，近似推論，表現学習が密接に結びついている。次に，拡散モデルでは，生成がノイズ除去の逐次過程として再定式化され，離散時間の DDPM から連続時間の SDE とスコア関数へ至る統一的理解が得られた。さらに，自己回帰モデルでは，結合分布の因果分解に基づいて条件付き分布を逐次的に直接学習する立場を確認し，LLM がその大規模な実例であることを位置づけた。最後に，世界モデルを通じて，生成モデルが観測生成だけでなく，将来予測，想像上ロールアウト，計画へ拡張されることを見た。

以上を通じて，生成モデルは，データ分布の学習，潜在表現の獲得，逐次的・確率的生成，さらには予測と計画へ拡張される統一的枠組みとして理解できる。個々の手法は，潜在変数をどう導入するか，尤度や下界をどう最適化するか，生成をどの方向の逐次過程として捉えるかという点で異なるが，いずれも「データがどのように生じるか」を数理的に記述しようとする試みである。

コラム：Persona 駆動型 LLM による災害後居住地選択モデル – 三木隆斗

本章では，生成モデルを「観測データの背後にある構造を仮定し，その構造から観測が生じる」と捉えた．この考え方は，画像や文章の生成だけでなく，人間の選択行動のモデル化にも応用できる．ここでは，津波避難後の居住地選択を，persona を潜在表現として用いる LLM モデルで記述した例を紹介する．焦点は，LLM を単に分類器として使うのではなく，世帯ごとの見えにくい選好や判断基準を persona として表現し，それを loading 関数で選択する点にある．

研究背景

戦争，紛争，巨大災害の後には，被災者の避難，移住，再定住といった長期的な人口移動が発生する．とくに人口減少傾向にある地域では，災害後に誰がどこへ移るかが，その後の地域の人口構成，復興事業の進み方，公共住宅の需要，生活圏の再編に大きく影響する．したがって，災害後の居住地選択を理解することは，復興計画や住宅供給のマネジメントにとって重要である．

一方で，この現象は通常の交通行動モデルより扱いにくい．既存の交通計画理論では，通勤や買物，避難時の移動のように，比較的短期の選択を扱うことが多い．しかし，災害後の移住や定住は数年から数十年単位で進む長期現象であり，次のような難しさがある．

- 長期的な定量データを得にくい．
- 世帯ごとの居住履歴が不完全になりやすい．
- 復興制度，住宅供給，地域条件，家族事情が絡み合う．
- 観測属性だけでは，帰還意向や再建意欲などの判断基準を捉えにくい．

このような背景から，表形式の変数だけでなく，世帯の履歴や地域文脈を文章として扱えるモデルが必要になる．大規模言語モデル（large language model; LLM）は，人間の判断過程を文脈依存的に表現しうるため，従来の離散選択モデルでは表しにくい半構造的な意思決定文脈を入力できる．

使用データ

使用したデータは，2026 年 3 月に実施された被災者アンケートである．対象は，災害時の居住地域が宮城県・岩手県・福島県のいずれかであった世帯であり，個人属性，世帯属性，居住地選択履歴，住宅類型などの情報を含む．データの規模は次の通りである．

項目	件数
世帯数	5,002
各世帯の状態数	11,662
居住地移動サンプル数	6,660

学習・評価の分割は世帯単位で行った．これは，同一世帯の履歴が訓練データとテストデータにまたがると，履歴情報を通じた情報漏洩が起こりうるためである．LLM による評価は計算コストが大きいため，テストデータのうち先頭 200 件を用いた．

分割	移動サンプル数	世帯サンプル数
訓練データ	4,691	3,542
バリデーション	982	709
テスト	987	751
LLM 評価	200	151

予測タスク

予測タスクは，各世帯の時点 t までの情報から，次時点 $t+1$ の居住状態変化を予測する問題である．

$$x_{i,t} = \phi(s_{i,0:t}), \qquad y_{i,t} = g(s_{i,t}, s_{i,t+1}) \in \mathcal{Y}. \tag{5.119}$$

ここで $x_{i,t}$ は世帯 i について時点 t までに観測可能な属性・履歴から作る特徴量，$s_{i,t}$ は地域，住宅類型，復興段階，時点などを含む居住状態，$y_{i,t}$ は次状態への遷移を分類したラベルである．将来情報など，予測時点 t で本来分からない情報は入力から除外した．符号化前の推論時特徴量数は 31 である．

選択肢集合 $\mathcal{Y}$ は次の 6 種類である．

選択肢	サンプル数	比率
現在地にとどまる	519	7.8%
被災地域内の仮住まいへ移る	241	3.6%
被災地域外の仮住まいへ移る	1,676	25.2%
災害公営住宅に移る	340	5.1%
自力再建先に移る	3,286	49.3%
帰還する	598	9.0%

この分布から分かるように，「自力再建先に移る」がほぼ半数を占める．したがって，単純に多数派を選ぶモデルでも一定の正解率は出るが，少数クラスを識別できるか，また全体の移動分布を再現できるかが重要になる．

LLM を使う動機

多項ロジットモデルやニューラルネットワークは，表形式に整理された特徴量を入力として扱う．これに対して，災害後の居住地選択では，「なぜ帰還できないのか」「なぜ仮住まいを継続するのか」「自力再建を選ぶ条件は何か」といった説明的な文脈が重要である．これらは，単純な数値変数だけでは表しきれない．

そこで，LLM へのプロンプトには，世帯属性，居住履歴の文章要約，災害後の地域文脈，仮住まい・帰還・自力再建などの候補説明を含める．さらに，その世帯がどのような判断傾向を持ちそうかを表す persona を与える．LLM は，これらの文脈を読んだ上で，6 つの選択肢の中から次の居住地選択を出力する．

persona を潜在変数として見る

persona とは，個票データから直接観測される変数ではなく，世帯の行動傾向や判断基準を文章で要約した潜在表現である．たとえば，同じ「仮住まい中」という状態でも，早期帰還を望む世帯，自力再建を目指す世帯，災害公営住宅を待つ世帯では，次の選択が異なる可能性がある．この差は，年齢や世帯人数のような観測属性だけでなく，生活再建の方針や地域へのこだわりによっても生じる．

生成モデルの記法でいえば，観測された特徴量 x と選択 y の背後に，直接観測されない persona m があるとみなせる．

$$p(y \mid x) = \sum_{m} p(y \mid x, m)\, p(m \mid x). \tag{5.120}$$

ここで $p(m \mid x)$ は対象世帯にどの persona を割り当てるかを表す loading であり，$p(y \mid x, m)$ は，その persona を文脈として与えた LLM が各選択肢を選ぶ分布に対応する．これは，潜在変数モデル

$$p_\theta(x) = \int p_\theta(x \mid z) p(z)\, dz \tag{5.121}$$

において，観測されない z を周辺化する考え方と対応している．本研究では，persona がこの z に相当する役割を持つ．

persona bank を用いた推論の流れ

persona を使った推論は，次の手順で行う．

1. 訓練データの世帯属性・居住履歴を LLM に与え，世帯の行動傾向を説明する persona を作成する．
2. 作成した persona を persona bank として保存する．
3. 予測対象世帯の属性・履歴から，近い persona を loading 関数で選ぶ．
4. 現在状態，居住履歴，選択肢説明，選ばれた persona を LLM へのプロンプトに入れる．
5. LLM が 6 つの候補から次の居住地選択を出力する．

重要なのは，persona そのものが最終予測ではない点である．persona は，LLM が文脈依存的な判断を行うための補助的な潜在表現である．したがって，同じ persona が選ばれても，現在地，履歴，選択肢説明が異なれば，異なる choice が出ることが望ましい．

コサイン類似度による loading

もっとも単純な loading は，予測対象世帯と persona bank 内の各 persona をベクトル化し，類似度が最大の persona を選ぶ方法である．世帯属性や履歴要約をベクトル化する関数を $\psi(\cdot)$，persona m の表現を p_m とすると，

$$m^*(x) = \arg\max_{m} \mathrm{sim}\{\psi(x), \psi(p_m)\} \tag{5.122}$$

と書ける．コサイン類似度を使う場合，

$$\mathrm{sim}\{\psi(x), \psi(p_m)\} = \frac{\psi(x)^\top \psi(p_m)}{\|\psi(x)\| \|\psi(p_m)\|} \tag{5.123}$$

である．

この方法は直感的で実装しやすい．しかし，属性や履歴が近い persona が，選択行動の予測に最も有効とは限らない．たとえば，年齢や家族構成は似ていても，住宅再建への意向や

元居住地へのこだわりが異なれば，次の居住地選択は変わりうる．したがって，単純な類似度だけでは，choice に効く persona を選びきれない可能性がある．

EM アルゴリズム的な learned loading

そこで，persona 割当そのものを観測されない潜在変数の推論問題とみなし，loading 関数を学習する．真の「この世帯にはどの persona が対応するか」はデータに含まれていない．一方で，各世帯が実際にどの居住地選択をしたかは観測されている．したがって，persona の割当を，観測された choice をよく説明するように更新することが考えられる．

この発想は，EM アルゴリズムに似ている．EM では，観測されない潜在変数 z の事後分布を E-step で推定し，その推定結果を使って M-step でモデルパラメータを更新する．ここでは，persona m が潜在変数に相当し，loading 関数が $p_\alpha(m \mid x)$ を与えると考える．α は loading 関数のパラメータである．

まず，現在の loading 関数に基づいて，世帯 i が persona m に対応する重みを計算する．

$$r_{im} = p_\alpha(m \mid x_i) = \frac{\exp\{\mathrm{sim}_\alpha(x_i, p_m)\}}{\sum_{m'} \exp\{\mathrm{sim}_\alpha(x_i, p_{m'})\}}. \tag{5.124}$$

これは，各 persona に対する soft assignment であり，GMM の EM における責務 γ_{nk} と似た役割を持つ．これが E-step 的な部分である．

次に，LLM または選択モデルが，persona m を与えたときに観測 choice y_i をどれだけ説明できるかを評価する．理想的には，

$$\sum_i \sum_m r_{im} \log p_\theta(y_i \mid x_i, m) \tag{5.125}$$

が大きくなるように，loading 関数のパラメータ α を更新する．これが M-step 的な部分である．実装上は，LLM の内部パラメータを直接更新するのではなく，どの persona を選ぶと観測 choice への適合が高くなるかを用いて，類似度関数 $\mathrm{sim}_\alpha(\cdot, \cdot)$ を調整する．

この learned loading は，単に「属性が近い persona」を選ぶのではなく，「choice の予測に効く persona」を選ぼうとする点に特徴がある．生成モデルの言葉でいえば，persona の事後分布 $p(m \mid x, y)$ を完全には計算できないため，観測 choice を手がかりに近似的に loading を改善していると解釈できる．

評価指標と比較モデル

評価には，個票単位の正解率だけでなく，少数クラスや移動分布の再現性を見る指標を用いた．Accuracy は直感的だが，多数派クラスに引っ張られやすい．Macro F1 は各選択肢を同じ重みで評価するため，少数クラスを拾えているかを見やすい．また，Share MAE と JSD は，観測された選択肢シェアと予測シェアのずれを見る指標であり，人口移動分布の再現性を評価するために用いた．

$$\mathrm{Acc} = \frac{1}{N}\sum_{i}\mathbf{1}(\hat{y}_i = y_i), \tag{5.126}$$

$$\mathrm{MacroF1} = \frac{1}{|\mathcal{Y}|}\sum_{c\in\mathcal{Y}}\mathrm{F1}_c, \tag{5.127}$$

$$\mathrm{ShareMAE} = \frac{1}{|\mathcal{Y}|}\sum_{c\in\mathcal{Y}}|\pi_c - \hat{\pi}_c|, \tag{5.128}$$

$$\mathrm{JSD} = \frac{1}{2}D_{\mathrm{KL}}(\pi\|m) + \frac{1}{2}D_{\mathrm{KL}}(\hat{\pi}\|m), \qquad m = \frac{1}{2}(\pi + \hat{\pi}). \tag{5.129}$$

ここで N は評価サンプル数，$\hat{y}_i$ は予測ラベル，π_c と $\hat{\pi}_c$ はそれぞれ選択肢 c の観測シェアと予測シェアである．また，π と $\hat{\pi}$ は全選択肢にわたる分布，m はその平均分布であり，JSD は二つの選択肢分布の形のずれを測る．

比較モデルとして, 最頻クラスを選ぶ Most Frequent, 現在地ごとの多数派を選ぶ Current Area Majority，直近期の遷移シェアを使う Rolling Average，ルールベースの Domain Heuristic，多項ロジットモデル（MNL），多層パーセプトロン（NN），そして persona を用いる LLM モデルを用いた．LLM には Qwen2.5-3B-Instruct と Qwen2.5-7B-Instruct を用いた．

実験結果

まず 3B モデルでは, persona を与えても choice が十分に多様化しなかった．No-Persona LLM では self rebuilt が 98%を占め，Cosine Persona LLM や Learned Persona LLM でも self rebuilt と temporary within に強く偏った．この段階では，LLM が文脈を読んで推論しているというより，persona のあてはめや多数派選択に近い挙動になっていた．

一方, 7B モデルでは選択が多様化し, 外部一時避難や帰還も出力されるようになった．LLM 評価と同じ 200 件で比較した結果を次に示す．

モデル	Acc.	F1	Bal.	MAE	JSD
7B Learned	0.755	0.442	0.443	0.024	0.005
7B Cosine	0.735	0.316	0.385	0.022	0.008
Area majority	0.785	0.426	0.461	0.048	0.044
MNL	0.790	0.311	0.392	0.050	0.045
NN	0.725	0.317	0.388	0.050	0.047

Accuracy だけを見ると，MNL や Area majority が高い．しかし，Macro F1 では 7B Learned loading が 0.442 となり，比較モデルの中で最も高かった．また，JSD は 0.005 であり，選択肢シェアの分布再現性も高かった．これは，learned loading が少数クラスを含む選択傾向を捉える上で有効であったことを示している．

choice share を見ると，observed では self rebuilt が 0.590，temporary outside が 0.235 であったのに対し，learned loading では self rebuilt が 0.635，temporary outside が 0.250 となり，全体の分布形状を比較的よく再現していた．LLM 評価データには「現在地にとどまる」が含まれていなかったため 5 択分布での比較になったが，小さいシェアの選択肢も完全には潰れずに残った．

choice	observed	learned	cosine
public housing	0.095	0.045	0.035
self rebuilt	0.590	0.635	0.640
temporary outside	0.235	0.250	0.245
temporary within	0.025	0.025	0.030
return origin	0.055	0.045	0.050

persona 使用の集中と解釈

loading 方法ごとの persona 使用を見ると，Cosine-only では 4 種類の persona しか使われず，最上位 persona が 150 件（75.0%）を占めた．Learned loading では使用 persona 数は 5 種類，最上位 persona は 114 件（57.0%）であった．数値上は cosine よりやや分散しているが，依然として少数の persona に強く集中しており，loading 関数によって persona 集中の課題が解決されたとは言い難い．

一方で，learned loading では，同一 persona でも choice が完全には固定されなかった．これは重要である．もし同じ persona が常に同じ choice を出すなら，persona は単なるラベルであり，LLM は文脈を読んでいないことになる．しかし，7B learned loading では，persona は選択を一意に決めるものではなく，選択傾向の事前分布として働いていた．最終的な choice は，persona に加えて，現在状態，居住履歴，元居住地との関係，選択肢説明によって変化した．

考察と課題

この実験から，LLM を用いた居住地選択モデルには三つの示唆がある．第一に，災害後の居住地選択のような複雑な意思決定では，モデルサイズが結果に大きく影響する．3B では選択肢が多数派に偏りやすかったが，7B では選択が多様化した．第二に，persona は LLM に文脈依存的な推論を行わせるための潜在表現として有効である．第三に，観測 choice を用いた learned loading は予測分布の再現には寄与したが，persona の使用が少数に集中する問題までは十分に解消できなかった．

一方で，課題も多い．プロンプトを少し変えただけで結果が大きく変わるため，prompt 設計の頑健性を検証する必要がある．また，learned loading でも persona 使用はまだ少数に集中しており，十分に多様な潜在表現を使えているとは言い切れない．さらに，LLM がなぜその選択をしたのか，persona のどの記述が判断に効いたのかを検証する解釈性の分析も必要である．

今後は，復興制度，災害公営住宅の供給量，地域の復旧状況などを政策シナリオとして入力し，居住地選択分布がどのように変わるかを調べることが考えられる．その意味で，persona 駆動型 LLM モデルは，単なる分類器ではなく，潜在表現を介して災害復興の意思決定支援へ接続しうる生成モデル的な枠組みである．

参考文献

[1] Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-Time Analysis of the Multiarmed Bandit Problem, Machine Learning, Vol. 47, No. 2–3, pp. 235–256, 2002.

[2] Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, Proceedings of the 3rd International Conference on Learning Representations (ICLR), arXiv:1409.0473, 2015.

[3] Bellman, R.: A Markovian Decision Process, Journal of Mathematics and Mechanics, Vol. 6, No. 5, pp. 679–684, 1957.

[4] Bishop, C. M.: Pattern Recognition and Machine Learning, Springer, 2006.

[5] Boyd, S., Vandenberghe, L.: Convex Optimization, Cambridge University Press, 2004.

[6] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734, 2014.

[7] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, MIT Press, 2016.

[8] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Nets, Advances in Neural Information Processing Systems 27 (NeurIPS), pp. 2672–2680, 2014.

[9] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2 edition, Springer, 2009.

[10] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.

[11] Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models, Advances in Neural Information Processing Systems 33 (NeurIPS), pp. 6840–6851, arXiv:2006.11239, 2020.

[12] Ho, J., Salimans, T.: Classifier-Free Diffusion Guidance, arXiv preprint arXiv:2207.12598, 2022.

[13] Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory, Neural Computation, Vol. 9, No. 8, pp. 1735–1780, 1997.

[14] Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Proceedings of the 32nd International Conference on Machine Learning (ICML), pp. 448–456, 2015.

[15] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., Amodei, D.: Scaling Laws for Neural Language Models, arXiv preprint arXiv:2001.08361, 2020.

[16] Kingma, D. P., Welling, M.: Auto-Encoding Variational Bayes, arXiv preprint arXiv:1312.6114, 2014.

[17] Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25 (NeurIPS), pp. 1097–1105, 2012.

[18] Lattimore, T., Szepesvári, C.: Bandit Algorithms, Cambridge University Press, Cambridge, 2020.

[19] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278–2324, 1998.

[20] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level Control through Deep Reinforcement Learning, Nature, Vol. 518, No. 7540, pp. 529–533, 2015.

[21] Murphy, K. P.: Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

[22] Puterman, M. L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming, John Wiley & Sons, New York, 1994.

[23] Rezende, D. J., Mohamed, S., Wierstra, D.: Stochastic Backpropagation and Approximate Inference in Deep Generative Models, Proceedings of the 31st International Conference on Machine Learning (ICML), pp. 1278–1286, 2014.

[24] Robbins, H., Monro, S.: A Stochastic Approximation Method, The Annals of Mathematical Statistics, Vol. 22, No. 3, pp. 400–407, 1951.

[25] Rumelhart, D. E., Hinton, G. E., Williams, R. J.: Learning Representations by Back-Propagating Errors, Nature, Vol. 323, No. 6088, pp. 533–536, 1986.

[26] Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust Region Policy Optimization, Proceedings of the 32nd International Conference on Machine Learning, pp. 1889–1897, 2015.

[27] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms, arXiv preprint arXiv:1707.06347, 2017.

[28] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the Game of Go with Deep Neural Networks and Tree Search, Nature, Vol. 529, No. 7587, pp. 484–489, 2016.

[29] Song, Y., Ermon, S.: Generative Modeling by Estimating Gradients of the Data Distribution, Advances in Neural Information Processing Systems 32 (NeurIPS), pp. 11918–11930, arXiv:1907.05600, 2019.

[30] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., Poole, B.: Score-Based Generative Modeling through Stochastic Differential Equations, Proceedings of the 9th International Conference on Learning Representations (ICLR), arXiv:2011.13456, 2021.

[31] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, Vol. 15, pp. 1929–1958, 2014.

[32] Sutton, R. S.: Learning to Predict by the Methods of Temporal Differences, Machine Learning, Vol. 3, No. 1, pp. 9–44, 1988.

[33] Sutton, R. S., Barto, A. G.: Reinforcement Learning: An Introduction, 2 edition, The MIT Press, Cambridge, MA, 2018.

[34] Thompson, W. R.: On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples, Biometrika, Vol. 25, No. 3–4, pp. 285–294, 1933.

[35] Vapnik, V. N.: Statistical Learning Theory, Wiley, 1998.

[36] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I.: Attention Is All You Need, Advances in Neural Information Processing Systems 30 (NeurIPS), pp. 5998–6008, arXiv:1706.03762, 2017.

[37] Watkins, C. J. C. H., Dayan, P.: Q-learning, Machine Learning, Vol. 8, No. 3–4, pp. 279–292, 1992.

[38] Werbos, P. J.: Backpropagation Through Time: What It Does and How to Do It, Proceedings of the IEEE, Vol. 78, No. 10, pp. 1550–1560, 1990.

[39] Williams, R. J.: Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning, Machine Learning, Vol. 8, No. 3–4, pp. 229–256, 1992.