

論文 / 著書情報
Article / Book Information

題目(和文)	エージェントシミュレーションを用いた組織の効用と振る舞いに関する研究
Title(English)	
著者(和文)	小林知巳
Author(English)	Tomomi Kobayashi
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第8952号, 授与年月日:2012年9月25日, 学位の種別:課程博士, 審査員:寺野 隆雄
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第8952号, Conferred date:2012/9/25, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

エージェントシミュレーションを用いた
組織の効用と振る舞いに関する研究

総合理工学研究科

知能システム科学専攻

小林 知巳

概要

企業不祥事に代表される社会規範に違反した組織的な行為を組織体逸脱と呼ぶ。一方、企業組織における生産性向上のための改善活動も、既定の業務を改変する意味において規範からの逸脱である。既定の規範から逸れるという点で両者の作動原理は同様と捉えられるが、両者を一体的に扱い、改善と逸脱がどのような条件変化のもとでどのような過程を経て生ずるのかを明確化した先行研究は少ない。

本研究の目的は、改善と逸脱といった組織マネジメントにおける表裏一体的な現象が、どのような条件変化のもとでどのような過程を経て生ずるのかを明確化するモデルを構築することである。本モデルでは、多様な表裏一体現象を分析するために階層間の効用産出量の関係を分析の視点として取り入れる。その上で、組織と社会の効用を一致して増加させる過程を改善、組織の効用を増加させ社会の効用を低下させる過程を逸脱と定義する。これは、組織構成員が組織にとっての利益を認識する反面、自身の行為が社会に及ぼす影響への認識が不十分になるという、組織体逸脱の事例に共通する傾向に基づくものである。

モデルの構築に際しては、エージェント・ベース・モデリングの手法を活用し、3階層の効用ランドスケープ（社会、組織、個人）による不完全情報のもとで、自身の満足の向上を目指して探索学習と模倣学習を行うエージェントを設定する。実験においては、集団の多様性、インフォーマルネットワークの量、報酬配分の傾斜、の3つの条件の変化に伴って改善と逸脱の双方が生ずる過程を、効用産出量の変化の観察を通じて分析する。

また、モデルの動作を再現しつつ仮想のケースを記述することにより、モデルの説明範囲を確認し、実際の不祥事を記述したケースとの比較を行う。ケースとの接地によってモデルが現実を説明できる範囲を明確化するとともに、モデルに基づくケース記述の限界と利点を説明する。

本研究の主要な貢献は次の2点である。まず、階層的なランドスケープと集団の性質を設定し、組織構成員の学習によって、改善と逸脱の双方が生ずる過程と要因を分析するエージェント・ベース・モデルを構築したことである。次に、提案モデルに基づくケース記述によって、改善と逸脱の共通要素と差異要素を分離し、構成員の学習と行動変化の過程を説明したことである。

目次

第1章 序論	1
1.1 はじめに	1
1.2 研究の背景と動機.....	1
1.2.1 表裏一体的な現象の例.....	1
1.2.2 改善と逸脱の関係.....	1
1.2.3 階層間の効用の相反.....	2
1.3 研究の目的.....	3
1.4 論文の構成.....	4
第2章 研究対象と問題.....	6
2.1 はじめに	6
2.2 表裏一体的な現象を扱った先行研究.....	6
2.2.1 機能分析	6
2.2.2 組織体逸脱.....	7
2.2.3 プリンシパル・エージェント理論.....	7
2.2.4 オンラインコミュニティとサイバースケード.....	8
2.2.5 先行研究への問題意識.....	8
2.3 エージェント・ベース・モデリング.....	10
2.4 研究の位置づけ.....	10
2.5 問題の定義.....	11
2.6 研究方法論とモデルの要件.....	13
第3章 研究の方法論とモデル.....	15
3.1 はじめに	15
3.2 モデルの説明.....	16
3.2.1 全体像	16
3.2.2 基本原則(Basic Principles) : ランドスケープ	17
3.2.3 基本原則(Basic Principles) : NK モデル	18
3.2.4 集団(Collectives) : 効用配分	20
3.2.5 集団(Collectives) : 多様性.....	21
3.2.6 相互作用(Interaction).....	21
3.2.7 偶然性(Stochasticity)	22

3.2.8 適応(Adaptation).....	22
3.2.9 目標・対象(Objectives).....	22
3.2.10 学習(Learning).....	22
3.2.11 感知(Sensing).....	24
3.2.12 創発(Emergence).....	24
3.2.13 観察(Observation).....	24
3.3 シミュレーション.....	25
第4章 方法論とモデルについての議論.....	26
4.1 はじめに.....	26
4.2 方法論とモデルの評価基準.....	26
4.2.1 想定した条件でのモデルの振る舞いの予想.....	26
4.2.2 既知の事例・理論に基づく評価基準.....	27
4.3 評価基準に基づくモデルの振る舞いの確認.....	28
4.3.1 一致したランドスケープ.....	28
4.3.2 全エージェントが同質の状態.....	29
4.3.3 フォーマルネットワークのみの状態.....	29
4.3.4 報酬に差がつかない状態.....	30
4.3.5 まとめ.....	31
4.4 既知の事例・理論に基づく評価基準との対比.....	31
4.4.1 ランドスケープの矛盾.....	31
4.4.2 多様性とインフォーマルネットワーク.....	32
4.4.3 まとめ.....	33
4.5 ケースとの接地.....	33
4.5.1 ケースとの接地方法.....	33
4.5.2 ケースの設定.....	34
4.5.3 NKモデルに基づくケースの設定.....	37
4.5.4 モデルの動作の再現：ケースA.....	38
4.5.5 モデルの動作の再現：ケースB.....	41
4.5.6 モデルの動作の再現：ケースC.....	43
4.5.7 ケースの記述.....	45
4.5.8 ケースにおけるモデルの説明範囲.....	49
第5章 方法論の応用.....	53
5.1 はじめに.....	53
5.2 応用実験の構成.....	54

5.3 応用Ⅰ：集団の多様性による逸脱と改善の創発.....	54
5.3.1 実験の背景と目的.....	54
5.3.2 実験条件	55
5.3.3 実験結果：産出効用量の推移.....	55
5.3.4 実験結果：効用産出量の分布.....	57
5.3.5 考察	58
5.4 応用Ⅱ：インフォーマルネットワークと集団の多様性の関係.....	60
5.4.1 実験の背景と目的.....	60
5.4.2 実験条件	60
5.4.3 実験結果1：組織効用と社会効用の推移.....	62
5.4.4 実験結果1：個人効用と満足度の推移.....	63
5.4.5 実験結果2.....	64
5.4.6 考察	65
5.5 応用Ⅲ：報酬制度の変化に伴うエージェント行動の変化.....	67
5.5.1 実験の背景と目的.....	67
5.5.2 実験条件	67
5.5.3 実験結果	68
5.5.4 考察	70
第6章 結論と課題	72
6.1 本研究の結論と貢献.....	72
6.2 今後の課題.....	75
謝辞	77
参考文献	78
業績目録	82
付録.....	84

第 1 章 序論

1.1 はじめに

本章では、研究の背景と目的および論文の構成を述べる。企業不祥事に代表される社会規範に違反した組織的な行為を組織体逸脱と呼ぶ。一方、企業組織における生産性向上のための改善活動も、既定の業務を改変する意味において規範からの逸脱である。既定の規範から逸れるという点で両者の作動原理は同様であるとの前提のもと、本論文では、これがどのような条件変化のもとでどのような過程を経て生ずるのかを明確化するモデルを構築することを目的としている。そのために、階層間の効用産出量の関係性を分析の視点として取り入れ、その上で、組織と社会の効用を一致して増加させる過程を改善、組織の効用を増加させ社会の効用を低下させる過程を逸脱と定義している。

1.2 研究の背景と動機

1.2.1 表裏一体的な現象の例

企業経営や組織運営において数多く見られる表裏一体的な現象の一例として、企業不祥事が挙げられる。例えば、2007年に株式会社赤福が不適正表示問題によって営業禁止処分を受けた事例がある。同社は、商品の廃棄や品切れを防ぐために、製造した商品を計画的に冷凍保存し、需要動向に合わせて解凍する手法と設備を開発した。この方法の精度が高かったため、解凍後の商品の風味が損なわれることは無かったが、解凍日を基準に消費期限を設定したことが法律違反に問われた（樋口、2009）。赤福は、商品廃棄を大幅に低減するという改善効果を上げた一方で、法令違反によって処罰を受けることになった。

また、1999年に発生したJCO（ジャパンコンバージョン）東海村事業所での臨界事故では、燃料製造の生産性を向上させるために手間や時間のかかる工程を効率化する違法な工夫を重ねたことが要因として指摘されている。これらの工夫は、監督省庁への届を怠り、工程違反による臨界の危険性増加を黙認して推進された。結果的に、一人当たりの燃料生産量は約80%上昇したが、深刻な臨界事故を引き起こし、周囲の多くの住民に対する退避および屋内退避措置、農産物への風評被害といった甚大な影響が生じた（岡本、2001；岡本・今野、2003；岡本・今野、2006）。

両ケースに共通するのは、まず改善と逸脱の両側面を有することである。次に、社会の規範と組織の方針が乖離する環境において、構成員が組織にとっての利益を認識している反面、自身の行為が社会に及ぼす影響への認識が不十分になっていることも、共通の傾向である。

1.2.2 改善と逸脱の関係

社会規範に違反した組織的な行為は、組織体逸脱と呼ばれる（宝月，2004）。一方，企業組織において取り込まれる生産性向上活動である改善（Imai, 1986）も，既定の業務方法を改変する意味において，規範からの逸脱の一種と捉えられる。このように，既定の規範から逸れるという点で，両者の作動原理は同様である。しかし，両者を一体的に扱い，改善と逸脱がどのような条件変化のもとでどのような過程を経て生ずるのか，を明確化した先行研究は少ない。

企業経営において，社会に損失をもたらす逸脱行為を，罰則などの方法によって直接抑制する施策が取られることがあるが，これは組織の活力を弱める可能性もある（宝月，1988）。なぜなら，逸脱は必ずしも外部への不効用を意図した個人の行動によって引き起こされるとは限らず（Baucus, 1994），生産性向上などの改善の側面も持つからである。また組織体逸脱は，組織の効用拡大を目指した構成員の相互作用を通じて，逸脱的な活動を組み込んでいく過程（宝月，1988）と捉えることができる。しかし，不祥事などを記述したケースでは，組織構成員間の相互作用と学習によって逸脱が生ずる過程が十分に説明されていない。

本研究では，図 1.1 に示すように，組織体逸脱と改善の関係は表裏一体的であると捉え，両者を組織や社会にどのような効用を与えるかによって識別する。その上で，改善は組織と社会の効用を一致して増加させるが，逸脱は組織の効用は増加させつつ社会の効用を低下させる過程であると定義する。そして，改善と逸脱を生じさせる条件と構成員の行動の変化を分析する。

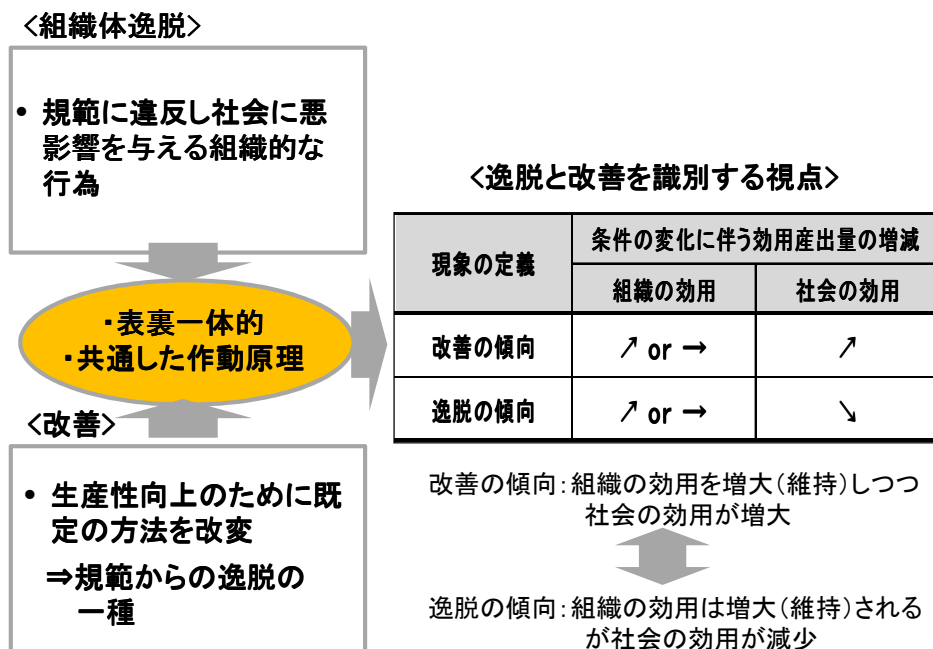


図 1.1 改善と逸脱の関係

1.2.3 階層間の効用の相反

表裏一体的な現象を，構成員の相互作用と学習によって生ずる階層間の効用の相反と捉えてモデル化することにより，図 1.2 に示したように組織マネジメントにおける様々なジレンマを階層間の効用産出量の関係の変化という共通の視点から分析できる．改善と逸脱に加えて，例えばネットコミュニティにおけるユーザー間の連帯と排他的な行動，あるいは組織運営の公式化による生産性向上と組織活力の減退などの表裏一体現象も，階層間の効用産出量の関係によって説明できる．よって以降の分析では，上位層と下位層の効用産出量の増減の過程によって，改善と逸脱の傾向を識別する．

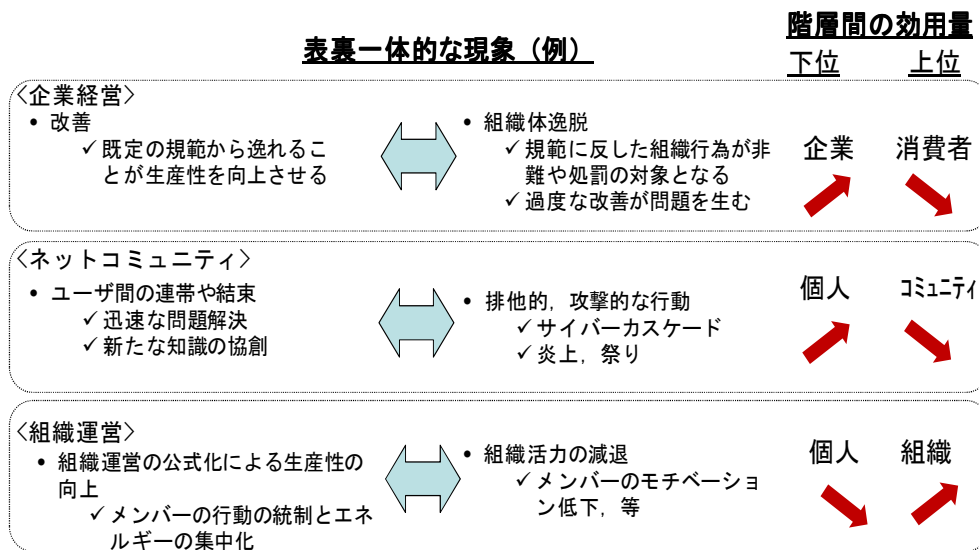


図 1.2 表裏一体的な現象と効用量の関係

1.3 研究の目的

以上より本研究の目的は，改善と逸脱に代表されるような組織マネジメントにおける表裏一体的な現象を分析するためのモデルを構築することである．本モデルでは，表裏一体現象を分析するために，社会，組織，個人という階層的なランドスケープを設定し，階層間の効用産出量の関係を分析の視点として取り入れる．これは，前述の不祥事の事例において，組織構成員が自組織にとっての利益を認識している反面，自身の行為が社会に及ぼす影響への認識が不十分であるという傾向を参考にしている．また，本研究で構築したモデルは，表裏一体的な現象である改善と逸脱が生ずる過程，および改善と逸脱を促進する要因，をその分析の対象とする．図 1.3 に，以上で述べた研究目的の構造をまとめた．また以降において，研究目的を踏まえて本論文の構成を述べる．

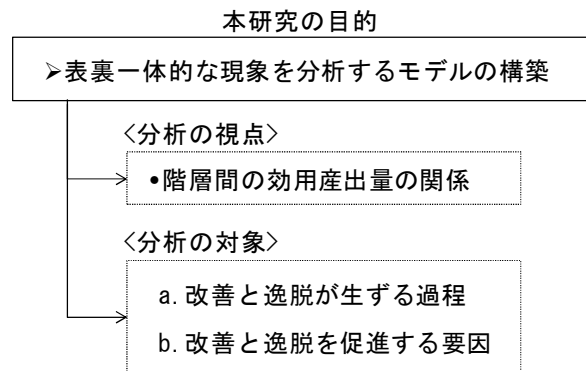


図 1.3 研究目的の構造

1.4 論文の構成

第 2 章以降の本論文の構成は次の通りである.

第 2 章：研究対象と問題

この章では、先行研究のサーベイをもとに、本研究の位置づけを明らかにした上で、研究において分析する問題を設定する。その上で、問題の分析に必要な研究方法論とモデルの要件を明らかにし、問題を分解して実験の構成を整理する。本研究が対象とする問題は、「不完全情報のもとでのエージェントの行動によって改善と逸脱が創発する過程と要因の分析」である。それを、1) 集団の多様性の変化が効用の産出に与える影響、2) インフォーマルネットワークの量が効用の産出に与える影響、3) 報酬による動機づけとエージェント行動との関係、の 3 つに分解する。

第 3 章：研究の方法論とモデル

この章では、2 章で設定した研究方法論とモデルの要件をもとに具体的なモデルを構築し、エージェント・ベース・モデルを記述するためのスタンダードに沿って説明する。

第 4 章：方法論とモデルについての議論

この章では、まず想定した条件下でのモデルの振る舞いを予想するとともに、モデルが生成する結果の妥当性を評価するために既知の理論と事例に基づく評価基準を設定する。その上で、予想と評価基準をもとに実験を行い、実験結果を予想や基準と比較する。これにより、モデルに設計の意図が反映されているか、および事前に設定した評価基準に照らして妥当な結果が生成されているか、を確認する。さらに、構築したモデルの要素をもとに仮想のケースを記述し、モデルの説明範囲の確認を行うと共に、モデルに基づくケースと事例に基づくケースとの相違を整理する。

第 5 章：方法論の応用

この章では、2 章で定義し分解した本研究の問題に沿って応用実験を行う。応用実験 I では集団の多様性、応用実験 II ではエージェント間のインフォーマルネットワークの量、応

用実験Ⅲでは成果に応じた報酬配分の傾斜度合いをパラメーターとして操作する。各パラメーターの操作による効用産出量の変化を観察し、エージェントの行動変化の過程を分析する。その上で、実験結果を各応用実験のテーマに関連する先行研究や事例と比較し考察する。

第6章：結論と課題

この章では、本研究の結果をとりまとめ、研究の貢献と今後の課題について述べる。2章において分解した3つの問題ごとに得られた結論を要約した上で、本研究が対象とする根本の問題に立ち返って研究の貢献と限界を整理する。また、それらをもとに今後の課題を提示する。

第2章 研究対象と問題

2.1 はじめに

本章では、先行研究のサーベイをもとに本研究の位置づけを示すとともに、対象とする問題を定義し、その接近法を論じる。関連研究として、機能分析、組織体逸脱、サイバークASCADE、などを挙げ、いずれも改善か逸脱の一方に焦点が当たり、一方から他方に転ずる過程の分析が不十分であるとの問題意識を示す。次に、エージェント・プリンシパル理論に言及し、集団の性質と表裏一体的な事象との関係が分析されていない点を問題意識として示した。本研究の接近では、改善と逸脱の双方が生ずる過程を、集団の性質（多様性、インフォーマルネットワークの量、成果貢献に基づく報酬配分の傾斜）を変数として操作しながら、効用産出量の変化の観察を通じて分析する。

2.2 表裏一体的な現象を扱った先行研究

2.2.1 機能分析

Merton (1949) は、顕在的順機能、顕在的逆機能、潜在的順機能、潜在的逆機能、という4種類の機能概念を提唱した。機能とは、「行為者の主観とは関わりなくその行為によって生じてくる客観的な結果」であり、当事者の意図や目的と区別される。事象Aが事象Bを促進する場合、AはBに対して順機能を持ち、阻害する場合は逆機能を持つ。また、行為によって意図されている結果が顕在機能であり、意図されていない結果は潜在機能と定義される。

Merton は、ある部族の雨乞いの儀式を例に、雨を降らせるという顕在的順機能を果たせてないものの、集団が一堂に会する機会を与えることで集団の同一性を強化する潜在的順機能を果たしていると指摘した。また、ホーソン西部電機会社での照明と生産性の関係に関する実験において、照明の条件とは無関係に生産性が向上したのは、実験そのものが持つ潜在的な機能によると主張した。労働者の生産性は、照明の加減ではなく、実験に伴う研究者との労働者間の関係の構築や集団構成員間の対人関係の変化という潜在的な社会的機能の影響を受けたことを指摘したのである。Mertonの研究の特徴は、生産性に機能を客観的な結果として捉え、当事者の意図や目的と区別したこと、および一見非合理的な行動が集団にとって機能的であることが多い点を指摘したことにある。しかしこの研究では、特に本研究の関心対象である潜在的逆機能が、どのような環境や条件のもとで生じやすいのか、については十分に言及されていない。

2.2.2 組織体逸脱

企業不祥事は、社会的な規範に違反する組織的な行為であり、広義には組織体逸脱と呼ばれる（宝月，2004）。逸脱の主な定義として、1) 人々や社会に有害な結果をもたらす病理的な特性を有する行為および行為者と捉える病理的定義、2) 規範や集合意識に違反した行為と捉える規範的定義、3) レッテルを貼ることによって逸脱が社会的に構築されるとする構成主義的定義が存在する（宝月，2004）が、本研究は、改善と逸脱の表裏一体的な関係を議論するために、2)規範的定義に依拠する。

組織体逸脱に対しては、例えば罰則の強化など逸脱行為を直接抑制する施策が、コンプライアンス強化の一貫として取られることがある。また企業不祥事の要因の一つとして、違法行為に対する給与や昇進などの報酬への期待が挙げられており（樋口，2009）、先行研究には、報酬やサンクションが組織体逸脱の継続を促進するとの主張もある。こうした観点から、過度な報酬期待を牽制するための直接的な抑制施策は有効である。一方で、このような施策は、組織の活力を弱め停滞をもたらす可能性もある（宝月，1988）。なぜなら、逸脱行為と改善活動の作動原理には共通性があり、規範からの逸脱が、「結果としてより妥当な制度の探索の機能を果たしているという側面がある」ためである（出口，2004）。また出口(2004)は、創造的逸脱と組織の失敗を招く逸脱の境界を引くことの難しさを指摘しており、本研究が対象とする逸脱と改善（生産性向上）との表裏一体的な側面に言及している。さらに、逸脱が学習の結果生ずることに着目し、学習の過程を解析することが組織逸脱の理解の上で重要であると述べており、この点は本研究が依拠する考え方である。しかし、他の多くの組織体逸脱の先行研究では、組織の失敗を招く逸脱に焦点が当てられ、改善との表裏一体的な関係への言及は少ない。加えて、いずれも組織的な事象に対する考察が中心であり、個々の構成員の学習の過程に対する分析は少なく、組織体逸脱と改善を生じさせる学習メカニズムのモデル化には至っていない。本間（2007）の研究は、組織体逸脱が発生する過程をモデル化している点が特徴だが、改善と逸脱との関係や両者を分かち条件については言及していない。

2.2.3 プリンシパル・エージェント理論

本理論では、経済問題の構造が委託者（プリンシパル）と代理人（エージェント）との関係として捉えられ、不完全情報におけるインセンティブ問題が扱われる（Salaniè，1997；伊藤，2003）。

財やサービスの質に対する情報が不完全な場合に良質な財やサービスの供給が妨げられるという逆選択問題は、不完全な情報環境における人々の行動が意図しない結果を引き起こすメカニズムを説明する。例えば、保険料が上昇すると、リスクの少ない加入者が減少し、リスクの大きい加入者が残るといった現象はその一例である。また、委託者が代理人の行動を正確に知ることができないという情報の非対称性が存在する場合には、代理人が契約に反して自己の利益を追求するというモラルハザード問題が発生し得る。保険加入者が

保険金を受け取るために事故を故意に生じさせることはその一例である。プリンシパルが自身の効用に関するエージェントの全行動を観察できるのであれば情報の非対称性は解消するが、そのための監督コストが極めて高くなるため現実的ではない。

プリンシパル・エージェント理論では、委託者と代理人の関係という階層的な構造と、委託者が代理人の特性を不完全にしか捉えられない不完全情報環境、という条件のもとで生ずる問題を分析しており、本研究のモデルが依拠する枠組みが提供される。一方で、当該理論では、エージェント間のネットワークやエージェントが構成する集団の性質については扱われておらず、どのような性質をもった集団において、モラルハザードがより生じ易いのかについては分析されていない。

2.2.4 オンラインコミュニティとサイバーカスケード

オンラインコミュニティは、利用者が共通の関心のもとに自主的に集いコミュニケーションを行う社会空間である（宮田，2005）。電子掲示板や SNS をはじめとするオンラインコミュニティは、共通する価値観をもつ個人同士が気軽に相互交流を図り共感を得る場として、急速に普及してきた。一方で、オンラインコミュニティには、ユーザー間の多様な知識の共有によって集合知を生みだす場としての側面もある（Wenger, 2002；三浦，2008）。集合知の充実、コミュニティの問題解決力や知識発信力を高めると考えられる。このため、分散している個人がコミュニティにおいて水平な立場でつながり、知識を共有することにより、様々な社会問題の解を見つけることができると期待されている。

また、ユーザーは新たな知識の獲得や問題の解決のみならず、連帯意識や共感などを求めてオンラインコミュニティに参加する。しかし、ユーザーが連帯意識や共感を追求すると、コミュニティ全体が素早く特定の信念や行動に飛びつくサイバーカスケード現象を生み、異なる考えを否定し極端な立場を取りやすい性質を強化することが指摘されている（Sunstein, 2003）。

オンラインコミュニティやサイバーカスケードの先行研究では、オンラインコミュニティにおける連帯と排他性の表裏一体的な関係を指摘しているが、一方が他方に転ずる条件や過程については十分な分析がなされていない。

2.2.5 先行研究への問題意識

以上で整理した、表裏一体的な現象を扱った先行研究のサーベイに基づき、表 2.1 に各先行研究への評価と本研究に取り入れる要素をまとめる。

まず本研究は、Merton (1949) から、行為者の意図と行為の結果を分離する考え方、およびプリンシパル・エージェント理論からは階層的な構造と不完全な情報環境をモデル構築の枠組みとして取り入れる。

一方で、先行研究に対する問題意識は次の通りである。組織体逸脱やサイバーカスケードの先行研究では表裏一体的な現象や要因が詳細に記述されているが、改善と逸脱の一方

に焦点が当たり、一方から他方へ転ずる過程についての議論が不足している。このため、どのような条件の変化に伴って改善が逸脱に転ずるのか、が十分に分析されていない。機能分析やプリンシパル・エージェント理論では表裏一体的な現象が類型化されており、現象が発生するメカニズムが説明されるが、表裏一体的な現象の発生に影響を与える集団の性質は考慮されていない。

前章の1.3で提示した、組織マネジメントにおける表裏一体的な現象が創発する過程と要因を、階層間の効用の産出量の関係の視点から分析するモデルを構築する、という本研究の目的は、以上の先行研究に対する問題意識に基づくものである。

表 2.1 先行研究の評価と本研究に取り入れる要素

先行研究	評価		本研究に取り入れる要素
機能分析 ある行為は、その意図に適合した結果(顕在的機能)のみならず、意図・意識せざる結果(潜在的機能)を生む(Merton, 1949)	特徴	・行為者の意図と行為の結果を分離 ・潜在的機能と行動の関係が詳細に記述される。	・行為者の意図とその結果との分離 ・集団の性質と潜在的逆機能の関係
	問題意識	・潜在的「逆」機能が、どのような環境や集団のもとで生じやすいのか、について十分言及されていない。	
組織体逸脱 組織逸脱への有効な規制は、社会的革新力を弱める可能性がある(宝月, 1988)。	特徴	・逸脱を生じさせる組織の状態や構成員の行動が詳細に記述されている。	/
	問題意識	・逸脱の側面に焦点が当たり、改善と対比した分析が不十分。 ・改善から逸脱に転ずる過程の分析が不十分。	
プリンシパル・エージェント 不完全な情報下において、モラルハザードや逆選択問題が生ずる(Salaniè, 1997)	特徴	・階層的な構造と不完全情報下における効用の増減を扱う。	・階層的な構造の枠組み、不完全な情報環境 ・構成員間のネットワークや集団の性質と効用との関係
	問題意識	・構成員間のネットワークなどの <u>集団の性質</u> については扱われていない。	
サイバースケード コミュニティにおける団結が、特定の信念への固執と排他性を高める(Sunstein, 2001)	特徴	・コミュニティメンバー間のコミュニケーションによる信念の伝播の過程が詳細に記述されている。	/
	問題意識	・排他性の側面に焦点が当たり、連帯がどのような条件と過程で、排他性に転ずるのかについては十分な分析がなされていない。	

2.3 エージェント・ベース・モデリング

エージェント・ベース・モデリングとは、「エージェント」と呼ぶ内部状態と意思決定力を備えた複数の主体についてモデル化を行い、エージェントの行動によって創発する現象とその創発過程を分析するための手法である（寺野，2004；2006；2007）。エージェント・ベース・モデリングは，事例分析から抽出したパラメーターを用いてモデル化を行い，そのモデルをもとにエージェントの行動をシミュレーションすることによって，社会的あるいはシステム的な課題への接近を試みる手法であり，数学的に記述された法則と，言葉で記述した事例分析の中間に位置する．モデルがコンピュータ・コードとして実行できるため再現性が高く，人々の間での伝達性が高いことが特徴である．また，エージェント・ベース・モデリングでは，合理的な意思決定の仮定を置かずに，多彩な学習を含む意思決定がエージェントに認められる（出口・木嶋，2009）．

エージェント・ベース・モデリングの適用領域について，Bonabeau (2002)は，組織の創発的な行動の分析が，有望な適用領域であると述べている．なぜなら，創発現象を捉えることができることが当モデリング手法の特長だからである．また，モデル構築の原則について Axelrod(1997)は，調査の対象は複雑であってもモデルの基礎となる前提は単純でなければならないと述べ，KISS 原理 (Keep it simple, stupid) を提唱し，その論拠として，当モデリング手法の目的が，特定の実験的な応用例そのものではなく，様々な応用例に表れる基本プロセスの理解にあることを強調した．一方で，寺野(2010)は，KISS 原理だけでは現実社会の複雑な現象のモデル化は不可能であることを指摘している．

本研究では，不完全情報下でのエージェントの相互作用を通じて，表裏一体的な現象が創発する過程を，様々な条件の下でシミュレーションするために，エージェント・ベース・モデリングの手法を取り入れる．モデルの構築に際しては，以上に整理した先行研究における議論を踏まえて，表裏一体的な組織現象に対する説明力とモデルとしての単純さとの両立を目指す．尚，本論文中に用いる「創発」は階層構造における創発の定義(Bass, 1992)に拠る．これは階層構造を持つ対象において，下位層の構造から説明できない上位層の特徴が生ずることを意味する．

2.4 研究の位置づけ

先行研究のサーベイをもとに，本研究の対象と位置付けを図 2.1 にまとめる．本研究は，表裏一体的な現象をモデル化して効用産出量という指標をもとに分析する観点において，事象そのものを記述する組織体逸脱や機能分析の先行研究と区別される．また，多様性や構成員間のネットワークなどの集団の性質を考慮しパラメーターとして変化させる点において，プリンシパル・エージェント理論の先行研究と区別される．

以上の位置づけの整理に基づく本研究の意義は次の二点である，一つは，様々な表裏一

体現象を、現象の記述ではなく、階層間の効用産出量の関係の観点から、統一的なモデルのもとに分析することである。

もう一つの意義は、エージェント・ベース・シミュレーションによって、集団の性質に関するパラメーターを操作しながら、様々な条件下で逸脱と改善の創発の過程を分析し、両者を促進する要因を見出すことである。

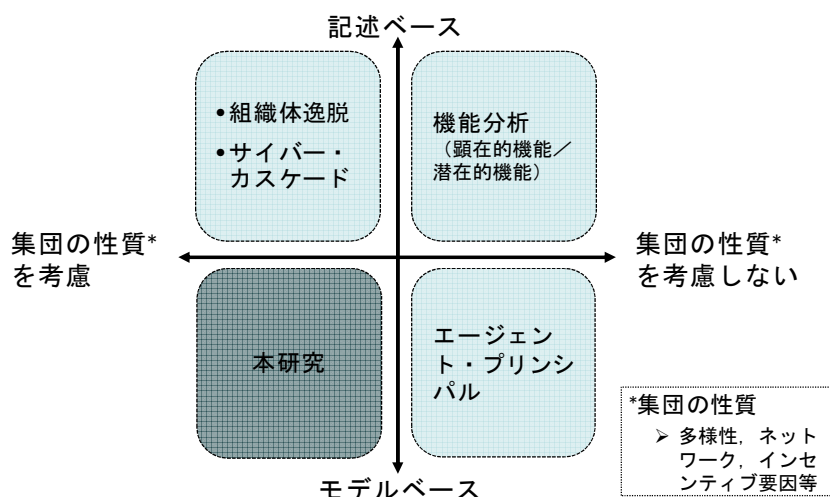


図 2.1 本研究の位置づけ

2.5 問題の定義

図 2.2 において、本研究が対象とする問題を定義し分解する。本研究の問題は「不完全情報のもとでのエージェントの行動によって改善と逸脱が創発する過程の分析」、および「改善と逸脱を促進する要因（集団の性質）の分析」である。後者の問題は、1) 集団の多様性の変化、2) 構成員間のネットワークの変化、および 3) 構成員個人の報酬による動機づけの変化の観点から、「Ⅰ. 集団の多様性の変化が効用の産出に与える影響」、「Ⅱ. インフォーマルネットワークの量が効用の産出に与える影響」、「Ⅲ. 報酬による動機づけの変化が効用の産出に与える影響」の 3 つに分解する。3 つの問題への分解は、組織体逸脱の先行研究や事例において取り上げられる、組織構成員の行動に影響を与える要因に基づくものである。

これら 3 つの問題ごとに、以降で応用実験のテーマを設定し、集団の性質の変化に伴うエージェントの行動とその結果である効用産出量の変化の過程を分析する。各応用実験では、それぞれのテーマである多様性、インフォーマルネットワーク、および報酬制度に関する先行研究や事例と、実験結果とを対比して考察を行う。分解した 3 つの問題と関連する先行研究の対応を図 2.3 に整理する。Ⅰ. 集団の多様性の変化が効用の産出に与える影響については、ダイバーシティ・マネジメント、組織体逸脱、およびオンラインコミュニティの各先行研究において、集団の多様化と同質化に伴う組織行動の変化が分析されている。

II. インフォーマルネットワークの量が効用の産出に与える影響については、組織ライフサイクルおよびインフォーマルコミュニケーションに関する先行研究で、インフォーマルコミュニケーションと組織構成員の行動との関係が分析されている。III. 報酬による動機づけの変化が効用の産出に与える影響については、報酬制度や組織体逸脱の先行研究において、成果主義型の報酬制度が組織構成員の行動に与える影響が分析されている。

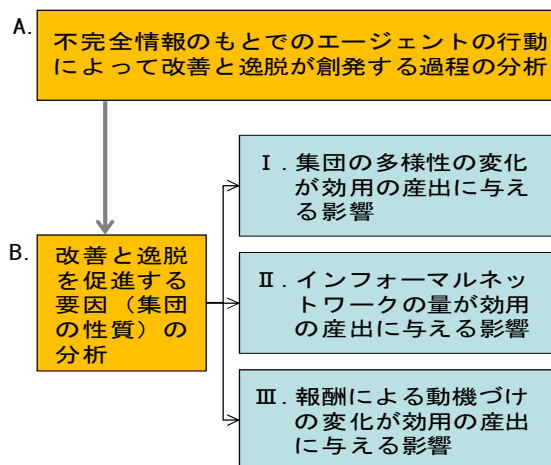


図 2.2 問題の定義と分解

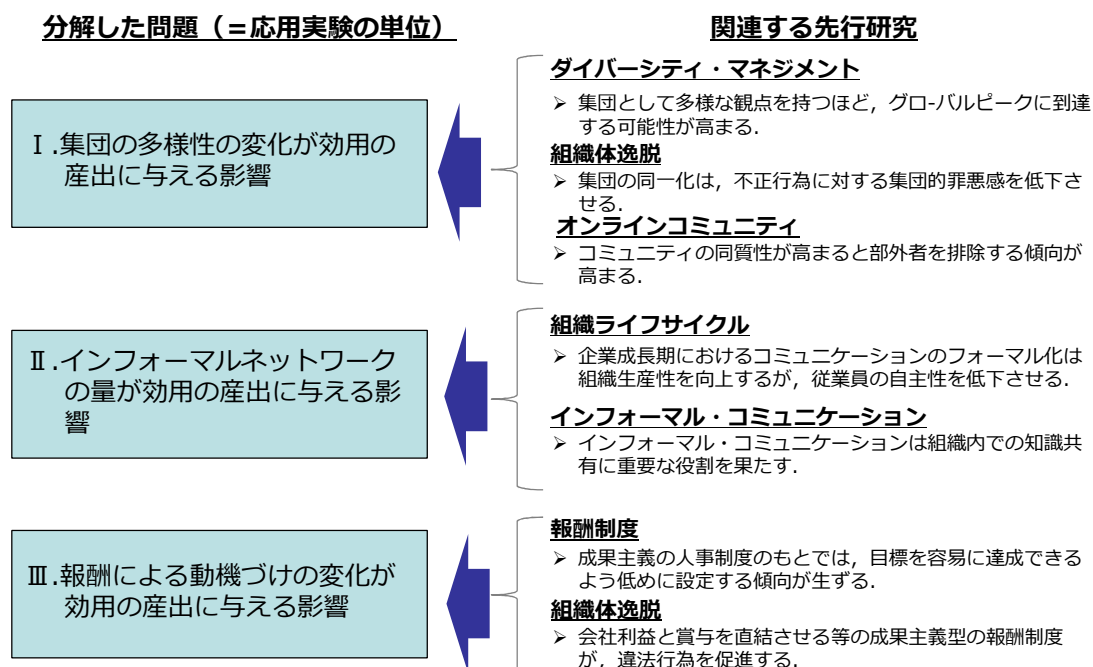


図 2.3 各問題と先行研究の関係

2.6 研究方法論とモデルの要件

2.5において定義および分解した本研究の問題を分析するためのモデルを構築するに際しては、図 2.4 に示した要件を満たす必要がある。

まず、エージェントにとっての不完全情報環境が実現されることが必要である。多くの組織において、構成員は自分が取った行動の結果を十分に予測できない。その不確実性のために意図しない結果が生じる。本モデルでは階層的な効用関数をランドスケープとして設定し、エージェントが各効用関数を十分に認識できない環境を構築する。これによって「不完全情報のもとでのエージェントの行動によって改善と逸脱が創発する過程の分析」という問題への対応を可能とする。次の要件は、集団の性質をパラメーター化して操作を可能にすることである。本モデルでは、1)多様性、2)エージェント間のネットワーク、および3)報酬配分の制度を、パラメーターとして設定し、操作を可能とする。これによって「改善と逸脱を促進する要因の分析」という問題への対応を可能とする。

さらに、エージェント・ベース・モデルを構築するためには、これらの不完全情報環境と操作された性質を持った集団の中での、エージェントの行動の規則を設定する必要がある。Axtell(2000)は、エージェントの状態と行動の規則の定義がモデル構築上の重要な要素であると述べている。本モデルでは、各エージェントが自身の満足を高めることを目的とした 2 種類の学習行動規則を設定する。一つはエージェントが現在の行動より高い満足をもたらす行動を探索するヒルクライム型の探索学習であり、もう一つは行動状態が近いか自身よりも報酬が多い友人の行動を模倣する模倣学習である。このような学習によるエージェントの行動の変化の過程は、各階層における効用産出量の変化として観察する。上位階層と下位階層の効用産出量がともに上昇する現象を改善と見なし、下位層の効用は上昇するが上位層の効用が下降する現象を逸脱と見なす。パラメーターの変化に伴う階層ごとの効用産出量の変化の傾向を比較することにより、改善と逸脱が生ずる過程とその要因を考察する。

エージェントの行動規則とその結果を含めた、本研究のモデル要件全体の構成を図 2.5 に示す。これらの要件に基づいて、第 3 章で研究モデルを構築し記述する。

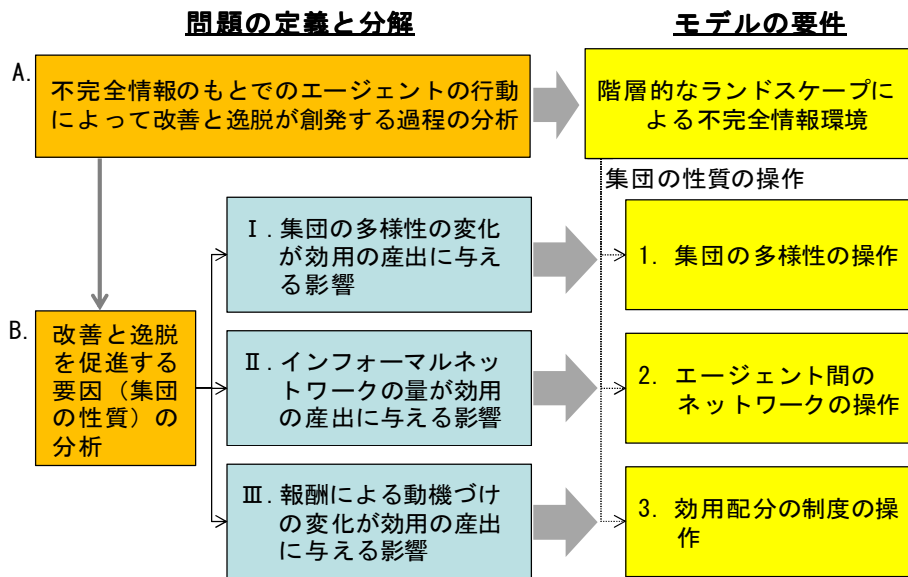


図 2.4 本研究の問題とモデル要件との関係

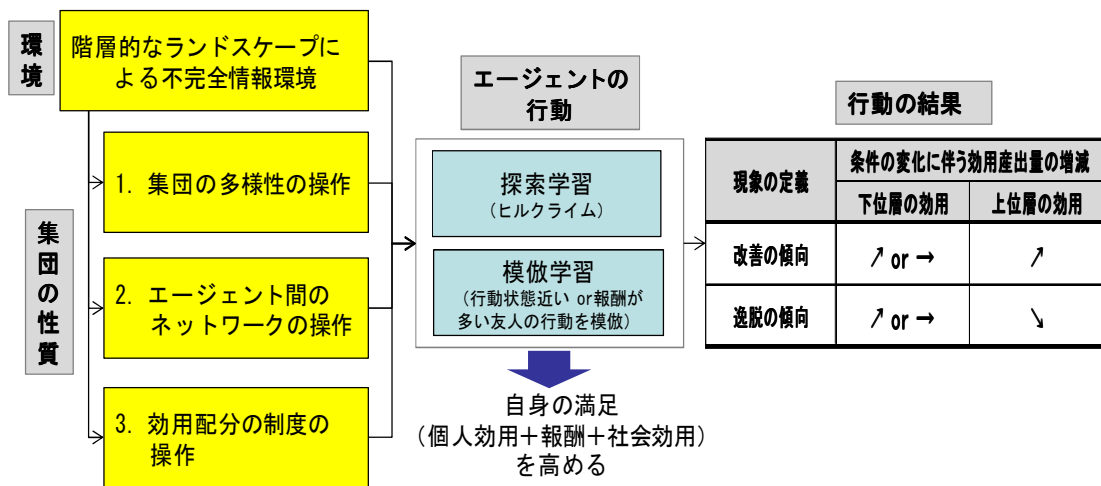


図 2.5 モデル要件の全体像

第3章 研究の方法論とモデル

3.1 はじめに

本章では、本研究で構築したエージェント・ベース・モデルを、モデル記述のスタンダードである ODD (Overview, Design concepts, and Details) に基づいて説明する。Grim et al. (2006, 2010) は、エージェント・ベース・モデルの説明の標準化と完全性の向上を目的に ODD プロトコルを提示した。これはエージェント・ベース・モデルが再現性に欠けるとの批判に応えたものであり、表 3.1 に示したように 11 個のデザインコンセプトから構成される。ODD プロトコルに立脚することにより、モデルの記述の厳密さが確保できる。

本研究モデルでは、3 階層の効用ランドスケープ (社会, 組織, 個人) による不完全情報のもとで、自身の満足の向上を目指して探索学習と模倣学習を行うエージェントを設定している。また効用関数には、Kauffman(1993, 1995)の NK モデルを用いる。NK モデルは抽象度が高いが、多峰性のランドスケープが設定し易く、かつ、階層的なランドスケープ間の相反関係を実現し易い、という利点があるため適用した。

表 3.1 ODD デザインコンセプト

No.	項目	概要
1	基本原則 (Basic Principles)	・モデルの土台にある基本コンセプト, 理論, 仮説 ・基本コンセプト間の関係
2	創発(Emergence)	・エージェントの適応特性や行動によって創発する結果は何か
3	適応(Adaptation)	・エージェントが、彼ら自身や環境の変化に反応して、意思決定し行動を変更するルール ・エージェントが目指す目標
4	目標・対象 (Objectives)	・エージェントが追求する成功の定義, 目標の測定方法
5	学習(Learning)	・エージェントが経験を通じて適応特性を変化させる方法
6	予測(Prediction)	・エージェントはどのように将来の状態を予測するか(環境および自分自身の将来の状態)
7	感知(Sensing)	・エージェントが感知し意思決定において考慮する内部および外部状態変数(state variables) ・他のエージェントから送られるシグナルを含む
8	相互作用 (Interaction)	・コミュニケーションを含むエージェント間の相互作用(直接/間接)
9	偶然性 (Stochasticity)	・ランダム, あるいは部分的にランダムなプロセスのモデル化
10	集団(Collectives)	・個人が形成するあるいは所属する集団(個人に影響を与え, あるいは影響される)の説明・表現 ・集団は個人の行動の結果として形成されるか, あるいはモデル設計者による定義か
11	観察(Observation)	・分析のためにABMから収集されるデータ

3.2 モデルの説明

3.2.1 全体像

本研究のモデルの要素を ODD の 11 個のデザインコンセプトに基づいて表 3.2 に整理した。ODD に基づくモデルの要素は次の通りである。

まず、基本原則(Basic Principles)として、社会、組織、個人という 3 階層のランドスケープと NK モデルに基づく効用関数を設定する。

集団(Collectives)の条件としては、集団の多様性とエージェントへの効用配分の方式を設定する。相互作用(Interaction)は、フォーマルおよびインフォーマルネットワークの構造として実現する。本モデルではこれらの操作によって集団の性質を変化させる。

偶然性(Stochasticity)は、ランダムなインフォーマルネットワークの設定によって友人選択に偶然性を持たせることで実現する。適応(Adaptation)は、所与の条件のもとで学習を行い、自律的に行動を変化させるエージェントの設定によって実現する。エージェントにとっての目標・対象(Objectives)は、社会および個人効用と組織から与えられる報酬の総和である満足度とする。学習(Learning)については、探索学習と模倣学習をエージェントの行動規則として設定する。感知(Sensing)については、自身の現在の行動状態の周囲やインフォーマルネットワークで繋がった友人からの情報のみに基づく意思決定により、感知の限界を持ったエージェントを設定する。以上の 6 つのデザインコンセプトによって、エージェントの行動規則とする。

以上のモデルの設定のもとでエージェントが相互作用をしながら学習と意思決定を行い、効用産出量を変化させた結果として、改善と逸脱が創発(Emergence)する。改善と逸脱の創発過程の観察(Observation)は、社会、組織、および個人の各階層の効用産出量の変化を通じて行う。

尚、本モデルにはデザインコンセプトの内「予測(Prediction)」に該当する要素は取り入れていないため、記述の対象外とした。

以降の 3.2.2 から 3.2.13 において、各デザインコンセプトに沿って本モデルの内容を説明する。

表 3.2 ODD デザインコンセプトに基づく本モデルの要素

分類	モデルの要素	
	デザインコンセプト(ODD)	要素
環境	基本原則 (Basic Principles)	階層的なランドスケープ: 社会, 組織, 個人の3階層のランドスケープ
		効用関数(NKモデル): 効用関数の多峰性
		効用関数(NKモデル): 階層間の効用関数の相反関係
集団の性質	集団(Collectives)	効用配分: 組織効用の産出量に応じた効用配分 集団を構成するエージェントの個人効用関数の多様性
	相互作用(Interaction)	ネットワーク構造 ①フォーマルネットワーク(上下間)を通じたインタラクション ②インフォーマルネットワーク(同位間)を通じたインタラクション (インタラクション: 行動状態と報酬情報の交換)
エージェントの行動規則	偶然性(Stochasticity)	友人選択: インフォーマルネットワークは, ランダムに選択されたエージェント間に形成
	適応(Adaptation)	行動状態の変化: ・エージェントは自身の満足の向上を目指して行動を変化
	予測(Prediction)	
	目標・対象(Objectives)	満足度の構成: 社会効用, 報酬, 個人効用の総和
	学習(Learning)	①個人の探索による学習(ヒルクライム) ②友人の模倣による学習(行動状態が近い, 報酬がより多い)
	感知(Sensing)	不完全情報による選択: ①自身の行動状態の周辺の効用の増減 ②友人の行動状態 ③友人が得た報酬
結果	創発(Emergence)	階層間の産出効用量の関係の変化
	観察(Observation)	①社会効用の産出量の変化の過程 ②組織効用の産出量の変化の過程 ③個人効用の産出量の変化の過程

3.2.2 基本原則(Basic Principles) : ランドスケープ

ランドスケープを中心としたモデルの全体像を, 図 3.1 に示す. 2.4 において設定したモデルの要件を図 3.1 の4つの要素として記述した. これは, モデルの具体的な構造を定式化するに先立ち, 概念モデル (National Research Council, 2008) を記述したものである. 尚, ランドスケープの概念は Axelrod(1997), および木島(2001, 2004)に基づいた.

まず, 1)階層的なランドスケープによる不完全情報環境は, 社会, 組織, および個人の3階層ごとに効用関数を設定することによって実現した. 社会効用関数は, 例えば消費者のニーズを表した関数であり, エージェントの行動がどの程度社会に効用をもたらすかを決定する. 組織効用関数は, 企業のビジネスモデルなどを表した関数であり, エージェントの行動が産出する組織効用を決定する. 組織効用と社会効用の関数は, 全エージェントに対して共通の形状とするが, 両者の間に一定の相反関係を持たせることで, エージェントにとってのジレンマを実現する. 個人効用関数は, 各エージェント個人のメンタルモデルや価値観を表した関数であり, エージェントにとって現在の状態がどの程度快適な状態にあるのかを決定する. エージェントは, 自身の現在の行動状態の周辺に限って効用関数の形状を把握することができる. よって, 効用ランドスケープ全体を把握した上で行動を選

択することはできず，不完全情報下に置かれる。

次に 2) 集団の多様性は，個人効用関数の多様性によって実現する。個人効用関数は各エージェントごとに設定し，異なる個人効用関数を持つエージェントが多いほど，集団の多様性が高い。

3) エージェント間のネットワークは，組織階層の上下間は固定的なフォーマルネットワークで結ばれる。一方，同位のエージェント間はインフォーマルネットワークによって結ばれており，インフォーマルネットワークの量を操作することによってエージェント間のコミュニケーションを変化させる。

最後に 4) 効用産出と配分について説明する。まず社会および組織は全エージェントが産出したそれぞれの効用を総和する。その上で，組織効用は報酬制度に基づいて報酬として各エージェントに配分され，社会効用は各エージェントの産出量そのまま配分される。組織からエージェントへの報酬は，金銭のみならず，仕事における権限拡大など非金銭的な要素を含むものとして位置づける。

以上に説明したモデルの主な構成要素である，効用関数，エージェントの学習と行動選択の方法，エージェント間のネットワーク，および報酬の配分方式について，3.2.2 から 3.2.6 で説明する。

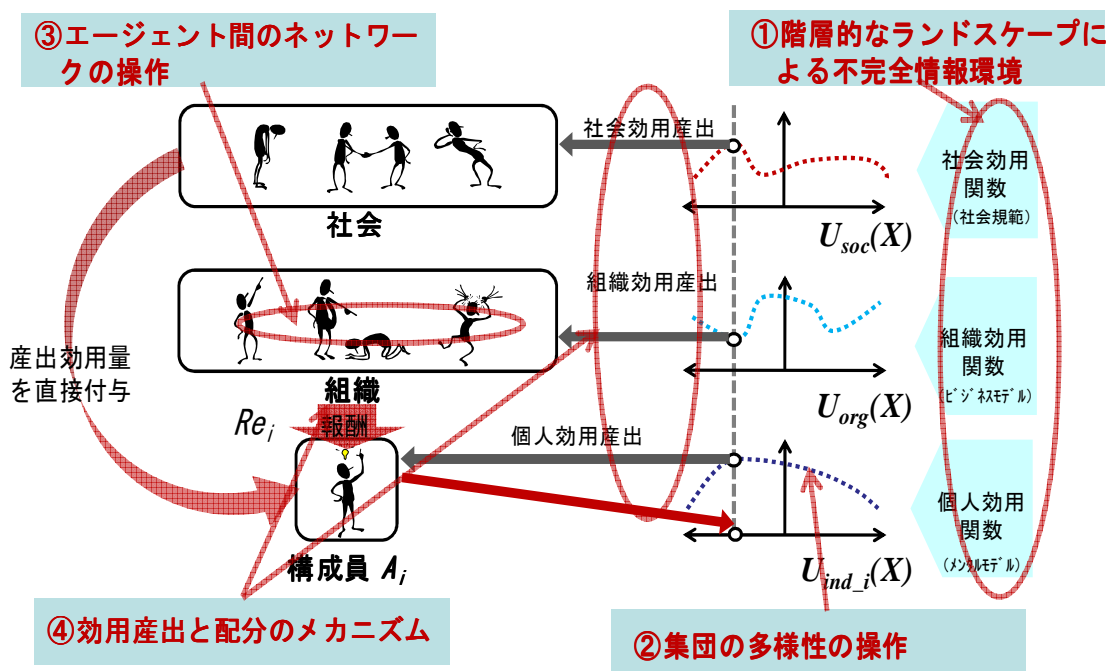


図 3.1 モデルの全体像

3.2.3 基本原則(Basic Principles) : NK モデル

本研究では，個人の価値観や，企業のビジネスモデル，消費者ニーズなどを効用ランドスケープとして実現するためにするために，Kauffman(1993, 1995)のNK 適応度地形モデル

(以下, NK モデル) を効用関数として用いた. NK モデルは, N 個の整数値の数字列を評価するモデルである. 各整数値は, 他の K 個の整数値との組み合わせによって評価値を決定する. 数字列全体の評価値は各整数値の評価値の平均とする (伊庭, 2002). 図 3.2 に $N=6$, $K=1$ の場合の数字列, 評価の依存関係, 整数値の組み合わせと評価値との対応, および評価値の計算例を示す.

Kauffman の生物進化の研究において, N 個の数字列は「生物の特質を決める遺伝子型」を示す. NK モデルは, 生物が持つある特質は K 個の他の特質と依存関係にあるとの考え方に立つ. 太い骨は重い生物にとっては有益だが, 足の速い生物にとっては有害になるのがその一例である.

本モデルでは, 効用関数間の違いを, この整数値の組み合わせと評価値の対応の違いによって定義する. 評価の際の整数値の依存関係の組み合わせは多様であるが, 本モデルでは単純化のため右隣の整数値に依存関係を持たせる方法を採用した. 整数値の各組み合わせに応じた評価値は, 0 から 1 の間の実数値でランダムに与えられる. K の大きさ, 依存関係の組み合わせ方, および整数値の範囲によって問題空間の局所解の数が増える. K の値が小さく (整数値間の依存関係が少なく), 整数値の取れる範囲が小さい場合, 局所解の数が少ない単純なランドスケープとなる. 逆に, K の値が大きく (整数値間の依存関係が多く), 整数値の範囲が大きい場合, 局所解の数が多くなる複雑なランドスケープとなる. 局所解を持った複雑なランドスケープを整数値の範囲と依存関係によって容易に実現できること, および整数値の組合せに対する評価値の設定によって, 階層間のランドスケープの相反関係を作り易いこと, が NK モデルを適用した理由である.

尚, 以降の応用実験では, $N=5$, $K=2$, 整数値の取れる範囲は 0 から 4 とし, 依存関係はランダムに設定した.

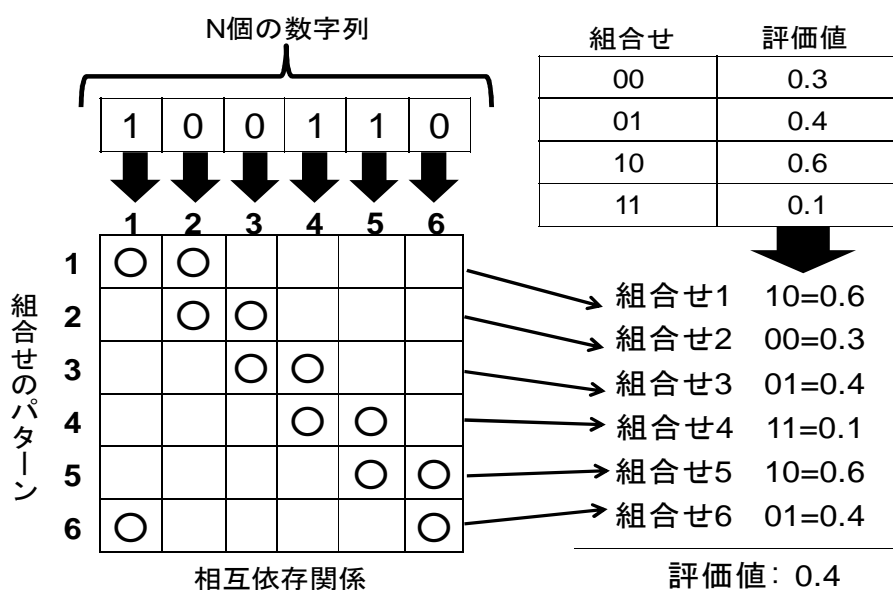


図 3.2 NK モデル

3.2.4 集団(Collectives)：効用配分

集団の構成員を動機づける報酬配分の方法は、均等配分から組織効用産出量に応じた傾斜の強い配分まで多様なパターンが考えられる。本モデルでは、図 3.3 のように報酬配分の傾斜の度合いを連続的に変化させながら、エージェントの行動の変化を分析する。

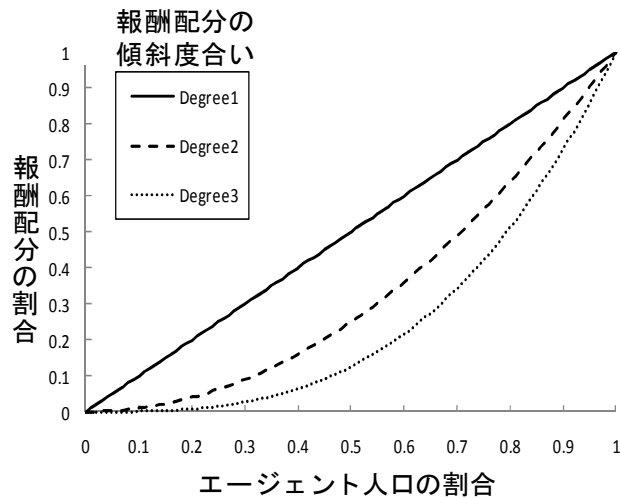


図 3.3 報酬配分のパターン

報酬配分の傾斜度合いを連続的に変化させるために、エージェントの累積報酬分布を式 3.1 から 3.3 によって設定した。これにより、均等配分を起点に傾斜の度合いを連続的に高める。ここで、 IR_e は累積報酬割合、 IP はエージェントの累積人口割合、 D は報酬配分の次数、 Ra_i はエージェント i が産出した組織効用の順位、 Z_D は報酬分布のジニ係数である。次数 D が大きい程、組織効用の産出量に応じた報酬配分の傾斜の度合いが高まる。

$$IR_e = IP^D \quad (3.1)$$

$$Re_i = D \left(\frac{n - Ra_i + 1}{n} \right)^{D-1} \frac{\sum_{i=1}^n U_{org}(X_i)}{n} \quad (3.2)$$

$$Z_D = \left(1 - \frac{2}{D+1} \right) \quad (3.3)$$

3.2.5 集団(Collectives) : 多様性

本モデルでは、集団の多様性を各エージェントの個人効用関数によって表す。Page (2007) は、集団の多様性を個々人の観点、解釈、ヒューリスティック、および予測モデルという要素の多様性として捉える枠組みを提示した。本モデルの個人効用関数は、Page (2007)が提示した要素に該当する。前述のように、異なる個人効用関数を持つエージェントが多いほど集団の多様性が高いと定義する。

3.2.6 相互作用(Interaction)

本モデルでは、多くの企業において見られる組織階層を表すため、三階層の階層型ネットワークを設定した。各階層のエージェント数は、図 3.4 の通りであり、全エージェント数は 39 人である。

ここで、上下階層の繋がりをフォーマルネットワークと定義し、同階層のエージェント間の繋がりをインフォーマルネットワークと定義する。第一階層のエージェント間の関係はフォーマルネットワークとする。

同階層のエージェント間のインフォーマルネットワークは、シミュレーション開始時に指定し、ランダムに選択されたエージェント間に形成される。また、エージェントはシミュレーション実行ステップごとに、友人変更確率 P_{cf} にもとづいて、友人関係を変更する。具体的には、現在の友人の 1 人とのインフォーマルネットワークを切断し、ランダムに選ばれたエージェント 1 人と新たなネットワークを形成する。これにより組織内のインフォーマルネットワーク数を一定に保ちながらエージェント間の関係を変化させる。ネットワークを切断するのは、現友人の中で行動状態の距離（相違）が最も大きいエージェントである。尚、実験では $P_{cf}=0.1$ とした。

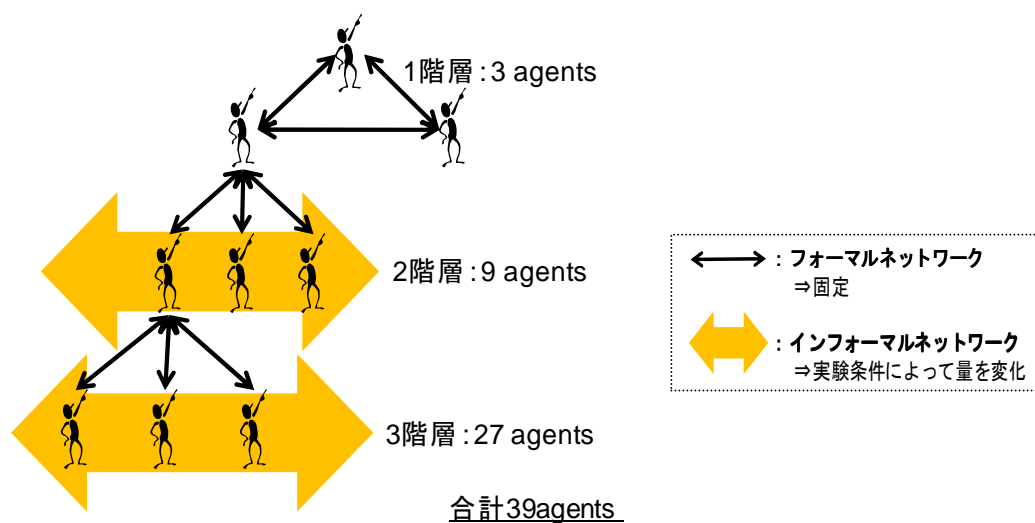


図 3.4 エージェント間のネットワーク

3.2.7 偶然性(Stochasticity)

3.2.6 で述べたように、エージェント間のインフォーマルネットワークは、シミュレーション開始時にランダムに選択されたエージェント間に形成される。これは、組織構成員間の偶然の出会いを表したものである。また、実験ではシミュレーションごとに効用関数をランダムに与えるため、エージェントが行動するランドスケープについての偶然性も実現する。

3.2.8 適応(Adaptation)

エージェントは、与えられたランドスケープと集団の性質のもとで、3.2.9 において述べる探索学習および模倣学習を行いながら自律的に行動状態を変化させる。

3.2.9 目標・対象(Objectives)

各エージェントは、自身の満足度を向上させるために学習しながら行動を変化させる。エージェントの満足度は以下の式 3.4 によって決定される。満足度は、個人効用、組織報酬、および社会効用が増加するほど高くなる。ここで、 Re は組織から各エージェントへの報酬を意味し、 X は各エージェントが選択した行動を意味している。 i は各エージェントの番号である。エージェントの満足の構成要素には、社会効用が含まれるが、これは社会効用に貢献したことへの満足感を表したものである。

$$S(U_{ind_i}(X), Re_i) = U_{ind_i}(X) + Re_i + U_{soc}(X) \quad (3.4)$$

3.2.10 学習(Learning)

本モデルにおいて、各エージェントは探索学習と模倣学習の2種類の学習を行う。

・探索による学習

エージェントは自分の行動を決定する際に、現在の行動状態の周辺で、最も自分の満足度が高まると予想される行動を選択する。本モデルでは、限られた情報の範囲において自身の満足を高めようとする制約された合理性（上田，2003）を持ったエージェントを設定した。具体的には、整数値の数字列で構成される現在の行動状態から1つの整数値を変更し、現在の値以外の選択肢の中で式 3.5 を最も高める行動を選択する。式 3.2 は、式 3.1 の組織報酬 Re を組織効用 $U_{org}(X)$ に置き換えたものである。行動の結果としてエージェントが得る満足度は、組織からの報酬に依存する。しかし、報酬の多寡は組織の制度によって決定されるため、エージェントは受け取るまでその値を知ることができない。そのため、組織報酬の決定要因である組織効用の産出量に基づいて自身の行動を変更する。これは、ヒルクライムアルゴリズムの一種であり、組織の構成員が自らの成果を自己評価しながら行動を変えることをモデル化したものである。

$$S(U_{ind_i}(X), Re_i) = U_{ind_i}(X) + U_{org}(X) + U_{soc}(X) \quad (3.5)$$

・模倣による学習

本モデルにおいて，エージェントの行動選択は長さ N の整数値の数字列で表現する．エージェントは，自分自身と行動状態が近く，多くの報酬を得ている友人の行動の一部を模倣しようとする．行動の結果の予測が不確実な状況では，成功したように見えるエージェントの行動を模倣することが優れた戦略になる（Axelrod and Cohen, 1999）との指摘を取り入れた．エージェント間の行動状態の距離（相違）は，数字列中の異なる数字の数とし，相互に異なる数字が少ないほど類似するものとする．距離が 3 となる行動選択の組み合わせの例を図 3.5 に示す．

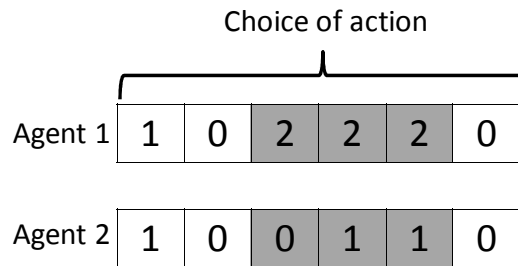


図 3.5 エージェント間の行動選択の相違

エージェント i が友人 j を模倣対象として選択する確率は式 3.6 の通りである．式 3.6 の分子は友人 j との行動状態の距離と報酬の積であり，分母はすべての友人達の行動状態の距離と報酬の積の総和である．これにより，エージェントはその友人と行動状態が近い程，そして組織から受け取る報酬が大きい程，模倣されやすくなる． L_{ij} は式 3.7 の通り，エージェント i とエージェント j の行動選択の近さを表しており，近い程 L_{ij} の値は大きい．式 3.7 において N は NK モデルにおける整数値の数字列の個数であり，行動状態 X の長さを示す． x_{ik} は，エージェント i の行動状態 X の k 番目の値であり， δ はクロネッカーのデルタである．エージェントが友人の行動をどの程度模倣するかは，行動状態の各整数値ごとに模倣確率によって決定される．本モデルでは，模倣確率を 0.5 とした．エージェントは，模倣後に自身の満足度が模倣前の満足度に比べて低下した場合は，元の行動状態に戻す．

$$P_j = \frac{Re_j \times L_{ij}}{\sum_{k \neq i} Re_k \times L_{ik}} \quad (3.6)$$

$$L_{ij} = \frac{\sum_{k=1}^N \delta(x_{ik}, x_{jk})}{N} \quad (3.7)$$

3.2.11 感知(Sensing)

エージェントは、行動を選択する際に、表 3.3 の情報を利用して学習を行い、行動選択の精度を向上させる。エージェントが利用する情報は、個人効用関数の評価値、組織効用関数の評価値、社会効用関数の評価値、インフォーマルネットワークによってつながった他エージェントの行動選択、他エージェントの報酬の 5 種類である。但し、利用できる情報の範囲は、自身の現在の行動状態の周辺やネットワークでつながった友人の情報に限定される。エージェントは、これらの情報をもとに各個人ごとの探索による学習、および友人の行動の模倣による学習という 2 種類の学習を行う。

表 3.3 行動選択に利用する情報

行動選択時に参照する情報	参照範囲
個人効用関数の評価値	自身の現在の行動の周辺のみ
組織効用関数の評価値	自身の現在の行動の周辺のみ
社会効用関数の評価値	自身の現在の行動の周辺のみ
他エージェントの行動選択	友人(非公式ネットワークで繋がったエージェント)のみ
他エージェントの報酬	友人(非公式ネットワークで繋がったエージェント)のみ

3.2.12 創発(Emergence)

本モデルを用いた実験において、エージェント行動の変化の結果は、社会、組織、個人の各階層ごとの効用産出量の変化として観察される。与えられた効用ランドスケープと集団の性質という条件の下でのエージェントの行動選択によって、階層ごとの効用産出量が変わり、階層間の効用量の相反関係が生ずる可能性がある。

3.2.13 観察(Observation)

本モデルを用いた実験によって収集されるデータは、社会効用産出量、組織効用産出量、個人効用産出量の 3 つである。本研究では、これらの効用産出量の絶対量を測定するのではなく、条件の変化に伴う産出量の変化の過程を観察対象とする。

3.3 シミュレーション

シミュレーションの進行は次の通りである。以下に示す Stage1 から Stage4 までを、全エージェントが順番に 1 回ずつ実行し、終了後に Stage5 を実行する。これを 1Step とし、Stage5 の終了後、Stage1 に戻って次の Step を開始する。

以降に 1 つの Step の流れを示す。

- Stage1 友人変更

友人とのインフォーマルネットワークを切断・形成する。

- Stage2 模倣行動成功判定

前 Step で模倣学習を選択したエージェントは現在の行動状態の模倣成功判定を実施する。

- Stage3 学習選択

模倣選択確率にもとづいて、探索学習を行うか(Stage4-1)、模倣学習を行うか(Stage4-2)を決定する。本実験では模倣選択確率を 0.3 に設定した。

- Stage4-1 探索学習

満足度の増加を目指して、現在の行動状態の周辺の範囲で行動を変更する。

- Stage4-2 模倣学習

行動状態が自身に近く報酬の多い友人の行動を優先的に模倣する。

- Stage5 報酬配分

エージェントの組織効用算出量に応じて、報酬制度をもとに組織が報酬を配分する。

第4章 方法論とモデルについての議論

4.1 はじめに

本章では、まず想定した条件下でのモデルの振る舞いを予想し、モデルが生成する結果を既知の理論と事例に基づいて評価する基準を設定する。その上で、設定した評価基準によるモデルの動作確認により、設計意図がモデルに反映していることを確認する。

次に、ケースとの接地によりモデルの説明範囲を確認する。ここでは、エージェントの探索学習と模倣学習の過程を手動で再現し、その上で、仮想の食品メーカーの設定でケースを記述し、モデルが現実を説明できる範囲を明確化する。さらに、記述した仮想ケースを実際の企業不祥事を記述したケースと比較し、モデルに基づくケース記述の限界と利点を説明する。

4.2 方法論とモデルの評価基準

ここで、構築したモデルの振る舞いの妥当性を評価するための基準を設定する。評価基準は、モデルが想定した条件における論理的な予想と整合した振る舞いを示すか、およびシミュレーションの結果が先行研究や事例で示された既知の理論や事実と整合するか、という二つの観点から設定し、次章において実験結果との比較を行う。

4.2.1 想定した条件下でのモデルの振る舞いの予想

まず、構築したモデルに設計者の意図が反映されていることを確認する。図4.1に本モデルの構成要素に対応して設定した極端な条件と、各条件についてモデルの構造から論理的に導かれる振る舞いの予想を示す。階層的なランドスケープによる不完全情報については、組織効用関数と社会効用関数が一致する条件を設定し、組織効用と社会効用の産出量が同量になるとの結果を予想した。集団の多様性については、全エージェントの個人効用関数が同一という同質の集団を設定し、同質でない条件と比較して個人効用の産出量が増加するという結果を予想した。エージェント間のネットワークについては、フォーマルネットワークのみ存在しインフォーマルネットワークが存在しない条件を設定し、インフォーマルネットワークが存在する条件よりも組織効用の産出量が低下するという結果を予想した。効用配分の制度については、組織効用への貢献によって報酬に差がつかない制度を条件として設定し、報酬に差がつく条件よりも組織効用の産出量が低下するという結果を予想した。

4.3において、それぞれの条件の下に実験を行い、モデルの実際の振る舞いが予想と整合することを確認する。

モデルの要素	想定した条件	振る舞いの予想
階層的なランドスケープによる不完全情報環境 ■ 三層構造の効用関数 ■ ピークを十分に認識できない	①ランドスケープが一致（組織と社会効用関数が一致）	➤ エージェントは、同量の組織効用と社会効用を産出
集団の多様性の操作 ■ 各エージェントごとの効用関数の設定による多様性のコントロール	②全エージェントが同質の状態（個人効用関数が一致）	➤ エージェント間の相互学習によって、個人効用関数のピークが探索され、個人効用の産出量が増加
エージェント間のネットワークの操作 ■ エージェントの相互コミュニケーション量を増減	③フォーマルネットワークのみの状態	➤ インフォーマルネットワークが多い状態より集団探索が抑制され、組織効用量は低下
効用配分の制度の操作 ■ エージェントは制度を通じて配分された報酬によって自身の行動の成果を間接的に認識	④組織効用への貢献によって、報酬（効用の配分）に差がつかない	➤ エージェントは報酬に差がつく条件下より組織効用への貢献を低下させる

図 4.1 想定した条件におけるモデルの振る舞いの予想

4.2.2 既知の事例・理論に基づく評価基準

次に、モデルが生成する結果と関連する領域における既知の結果との整合を確認するための評価基準を設定する。ここでは、組織体逸脱、ダイバーシティ・マネジメント、インフォーマルネットワーク、および報酬制度の先行研究における主張や事例を解釈し、モデルの実験結果と比較するための条件を図 4.2 に 2 つ設定した。

1 つ目は、矛盾するランドスケープのもとで成果主義的な報酬制度が取られた場合、エージェントが報酬を高めるために組織効用を重視し社会効用の産出量を低下させる、ということである（評価基準 a）。これは樋口（2009）で説明された食品会社および証券会社の不祥事の事例にもとづく。

2 つ目は、集団の多様性が高く且つインフォーマルコミュニケーションが多いほど組織効用の産出量を増加させる、ということである（評価基準 b）。これは、Page(2007)および西本(2006)と Kraut(1990)での指摘にもとづく。集団の多様性の増加やインフォーマルコミュニケーション量の増加が、集団内の異質な観点や知識の共有を促進し、生産性を向上させるという指摘である。

4.4 において、以上の 2 つの条件とシミュレーション実験の結果との整合を確認し考察する。

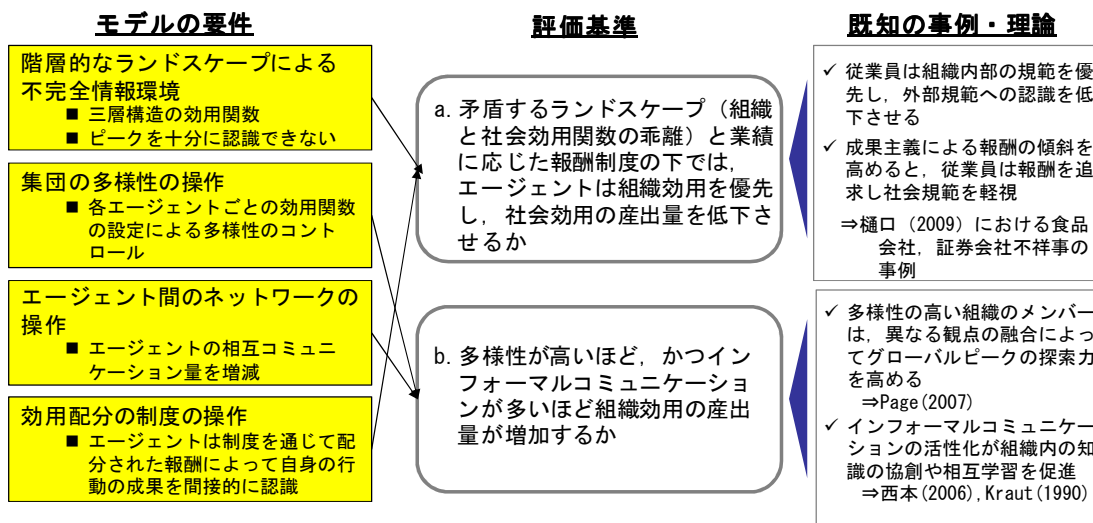


図 4.2 既知の事例・理論に基づくモデルの評価基準

4.3 想定した条件におけるモデルの振る舞いの確認

ここでは、4.2.1 で想定した 4 つの条件の下でシミュレーションを実施し、その結果を予想したモデルの振る舞いと比較する。

4.3.1 一致したランドスケープ

エージェントにランドスケープとして与えられる社会効用関数と組織効用関数を一致させた状態でシミュレーションを行う。関数を一致させた状態とは、3.3.2 で説明した NK モデルにおいて、社会効用と組織効用を表す N 個の整数値とその評価値が一致した状態である。他の実験条件については、報酬の傾斜次数 2、友人数 350、組織の多様度は 100% に固定した。

図 4.3 の実験結果の通り、一致したランドスケープの下では組織効用および社会効用ともに同量の効用が産出されている。これは 4.2.1 で予想したモデルの振る舞いと整合する。

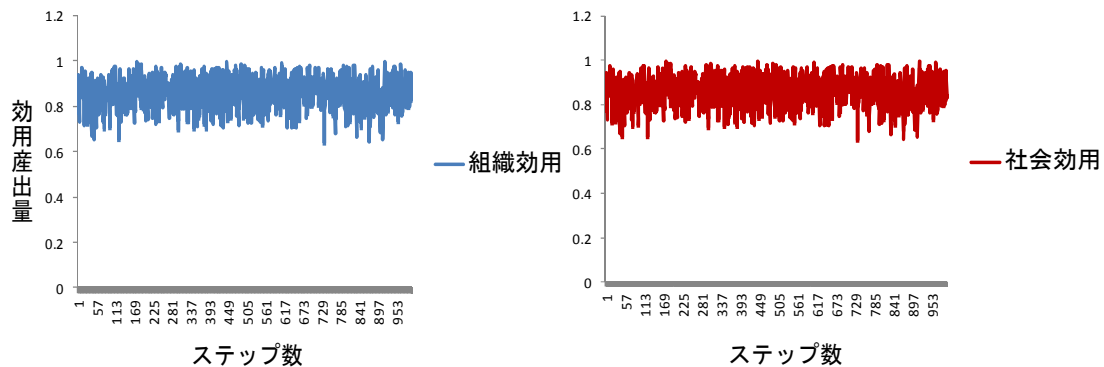


図 4.3 一致したランドスケープにおける効用産出量の比較

4.3.2 全エージェントが同質の状態

全エージェントの個人効用関数を一致させた状態でシミュレーションを行い、結果を全エージェントの個人効用関数が異なる状態の結果と比較する。図 4.4 に比較した結果を示す。多様度 0% が一致した状態（一様性の高い集団）であり、多様度 100% が異なる状態（多様性の高い集団）である。他の実験条件については、報酬の傾斜次数 2、友人数 350、組織効用と社会効用間の相反を 0.4 に固定した。

多様度 0% の方が、多様度 100% の状態よりも多くの個人効用を産出している。同質な集団の方が、エージェントが個人効用のピークの集団探索を強化していると解釈され、4.2.1 で予想したモデルの振る舞いと整合する。

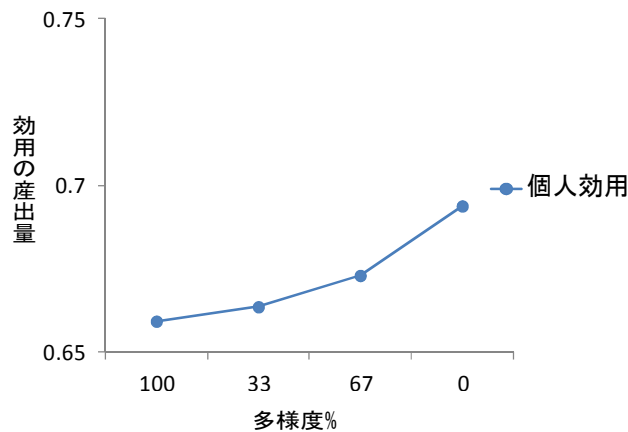


図 4.4 多様性の違いによるの個人効用産出量の比較

4.3.3 フォーマルネットワークのみの状態

同位のエージェントをインフォーマルネットワークで結ばずに、上下階層間のフォーマルネットワークのみの状態でシミュレーションを行う。インフォーマルネットワークの量は友人数で操作し、0 はフォーマルネットワークのみの状態、700 は約 90% のエージェント間にインフォーマルネットワークが設定されている状態である。他の実験条件については、報酬の傾斜次数 2、組織効用と社会効用間の相反を 0.4、多様度は 100% に固定した。

図 4.5 に友人数が 0 と 700 の状態の実験結果の比較を示す。友人数が多い（インフォーマルネットワークが多い）状態の方が、より多くの組織効用を産出している。これは、エージェントがインフォーマルコミュニケーション量の増加によって組織効用の集団探索を強化したためと解釈され、4.2.1 で予想したモデルの振る舞いと整合する。

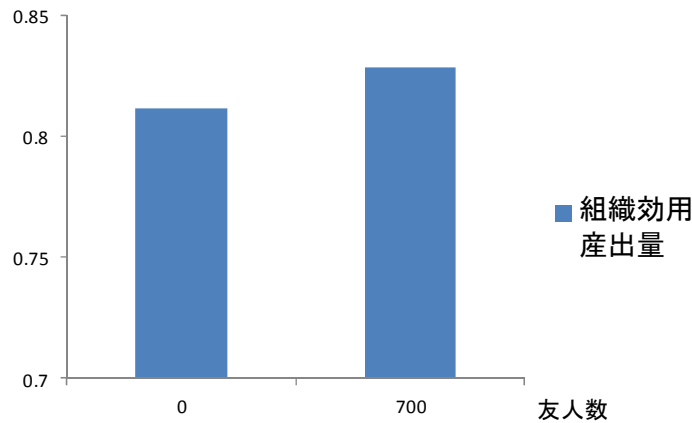


図 4.5 インフォーマルネットワークの違いによる組織効用産出量の比較

4.3.4 報酬に差がつかない状態

エージェントは組織効用の産出量に関係なく均等な報酬を与えられる状態でシミュレーションを行い、実験結果を組織効用の産出量に応じて報酬に一定の傾斜がつく状態と比較する。報酬の傾斜は次数で設定する。次数 1 は均等配分であり、次数 2 は報酬に一定の傾斜がつく状態である。他の実験条件については、組織効用と社会効用間の相反を 0.4、多様度は 100%、友人数 350 に固定した。

図 4.6 に、均等配分の状態と成果に応じて配分される状態とを比較した結果を示す。報酬の傾斜を示す次数 1 (均等配分) の方が、次数 2 (傾斜配分) よりも組織効用の産出量が少ない。これは、組織効用産出量によって報酬に差がつかないと、エージェントが組織効用の産出量の増加を追求しなくなると解釈され、4.2.1 で予想したモデルの振る舞いと整合する。

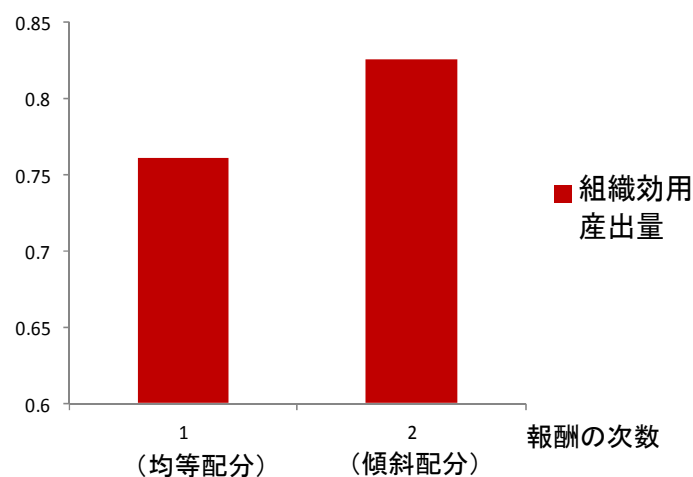


図 4.6 報酬制度の違いによる組織効用産出量の比較

4.3.5 まとめ

4.3.1 から 4.3.4 のシミュレーション結果はいずれも、想定した条件でのモデルの振る舞いの予想に合致した。よって、設計者の意図がモデルに反映されていることを確認することができた。

4.4 既知の事例・理論に基づく評価基準との対比

ここでは、4.2.2 において先行研究における主張や事例を解釈し、モデルの実験結果と比較するために設定した条件と、シミュレーションの結果とを比較し考察を行う。

4.4.1 ランドスケープの矛盾

まず、「矛盾するランドスケープのもとで成果主義的な報酬制度が取られた場合、エージェントが報酬を高めるために組織効用を重視し社会効用の産出量を低下させる」という条件と比較するために、社会効用関数と組織効用関数の相反度合いを変化させた実験を行う。相反度合い $x\%$ とは、社会効用関数と組織効用関数の評価値の内、 $x\%$ の評価値が 0.5 を基準として反転している状態をさす。評価値は 0~1 の乱数であるため、期待値 0.5 を基準に反転させる。3 章の図 3.2 で例示した評価値は、0.3, 0.4, 0.6, 0.1 の 4 個であるが、相反度合い 50% の場合は、この内 2 個の評価値が 0.5 を基準に反転する。この場合の組合せは 6 通りであり、(0.7,0.6,0.6,0.1) (0.7,0.4,0.4,0.1) (0.7,0.4,0.6,0.9) (0.3,0.6,0.4,0.1) (0.3,0.6,0.6,0.9) (0.3,0.4,0.4,0.9) となる（下線が反転した数値）。さらに、成果主義的な報酬制度を実現するため報酬の傾斜次数を 2 に設定する。他の実験条件については、友人数 350、組織の多様度は 100% に固定した。図 4.7 に実験結果を示す。社会効用関数と組織効用関数間の相反が高まるに従って、組織効用、社会効用ともに減少する結果となった。

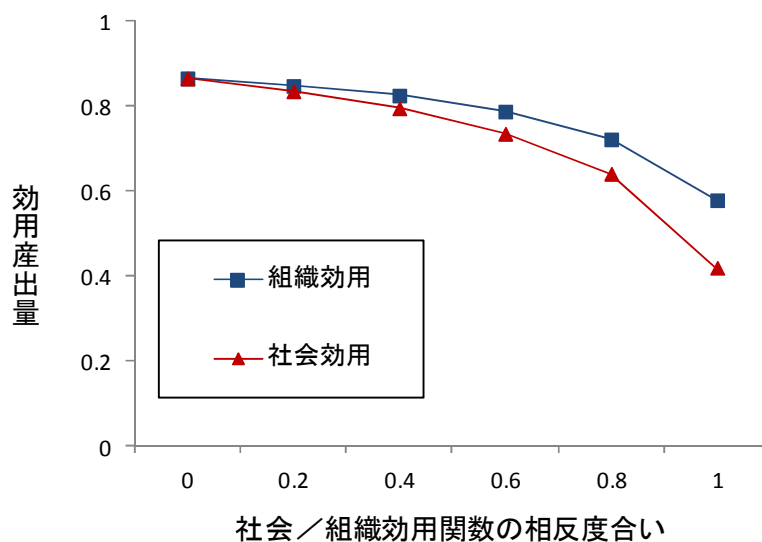


図 4.7 効用関数間の相反関係の強化に伴う効用産出量の変化

ここで、図 4.7 の実験結果の要因について考察する。まず、社会効用関数と組織効用関数の相反の高まりにより、エージェントは、組織効用と社会効用の産出量を同時に増加させる行動を見出し難いランドスケープ上に置かれる。よって、エージェントが組織効用量を高める行動を選択した場合は社会効用量が減少し、逆の場合は組織効用量が減少するという、ジレンマが常に生ずるため、組織効用量および社会効用量ともに減少したと推察する。この実験結果により、相反する効用ランドスケープによってジレンマ状態に置かれた集団は、社会効用量を低下させ逸脱に陥るだけでなく、組織効用量も低下させることが分かった。

4.4.2 多様性とインフォーマルネットワーク

次に、「集団の多様性が高く且つインフォーマルコミュニケーションが多いほど組織効用の産出量を増加させる」という条件と比較するために、集団の多様度とインフォーマルネットワークの量の双方を変化させた条件の下でシミュレーションを行う。集団の多様性は各エージェントの個人効用関数の相違で実現する。多様度 0% は全エージェントの個人効用関数が同一であり、100% は全て異なる状態である。また友人数 0 は上下階層間のフォーマルネットワークのみ存在する状態であり、友人数 700 は約 90% のエージェントが同位層間のインフォーマルネットワークで結ばれた状態である。

図 4.8 に多様度とインフォーマルネットワーク量の条件の 4 パターンの組合せの実験結果を示す。多様度が 100% で且つ友人数 700 の状態の組織効用産出量が、他の条件における産出量より多い結果となり、先行研究をもとに設定した条件の予想結果と整合した。

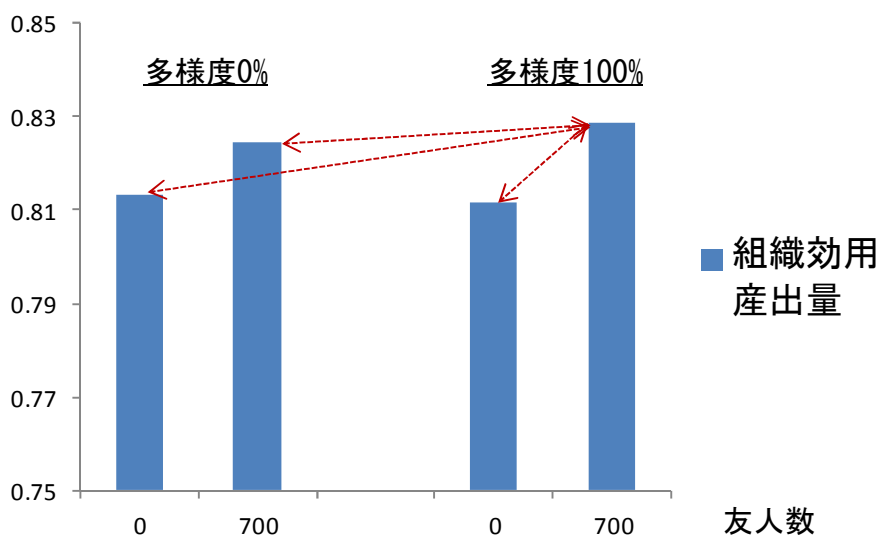


図 4.8 多様度とインフォーマルネットワークの違いによる効用産出量の変化

4.4.3 まとめ

4.4.1 の実験において、矛盾するランドスケープのもとで、社会効用の産出量が低下するのは予想した結果と一致するが、組織効用量の低下はこれまで明確に指摘されていなかったと認識している。よって、矛盾したランドスケープでのエージェントの行動をより具体的に分析するため、以降のケースとの接地において、エージェントの行動を手動で再現する。また 5 章の応用実験の条件として、一定のランドスケープの矛盾（社会効用関数と組織効用関数の相反度合い：0.4）を設定することとする。尚本章では、集団の性質に関する条件を固定してシミュレーションを行ったが、応用実験では、集団の性質をパラメーターとして変化させ、エージェントの行動の変化の過程を分析する。

4.4.2 の実験では、集団の多様性が高く且つインフォーマルネットワークの量が多い条件で組織効用の産出量が最も増加しており、予想した結果と一致した。但し、本章の実験は、多様性およびインフォーマルネットワークともに最大値と最小値の条件間の比較であるため、応用実験では中間的な状態を含めて効用産出量に変化する過程を分析する。

4.5 ケースとの接地

4.5.1 ケースとの接地方法

ここで、モデルの現実の事象に対する説明範囲の確認のために、構築したモデルについて仮想のケースとの接地を行う。モデルとケースとの接地ステップを図 4.9 に示した。

まずモデルの要素を企業経営の要素に置き換えた上で、ケースの骨格を表したテンプレートを設計し、テンプレートをもとにケースの条件を設定する。ここでは、仮想の洋菓子メーカーを条件として設定した。次に、社会、組織、個人の効用ランドスケープと集団の多様性については簡易な NK モデルに基づく設定を行う。その上で、仮想の洋菓子メーカーの従業員の行動を NK モデルに立脚して手動で再現する。ケースのストーリーは、テンプレートと再現した従業員の行動の変化の過程をもとに記述する。尚、当該ケースはモデルの要素に基づいて記述するが、ケースとしての完結性（読んで事象を理解できること）を確保するために必要な情報は、モデルの要素以外のものも必要に応じて加筆する。最後に、ケースに記述した要素とモデルの要素を比較し、モデルの説明範囲を確認する。さらに実際の事例を記述したケースとの比較も行い、モデルに基づくケース記述の限界と利点を明らかにする。

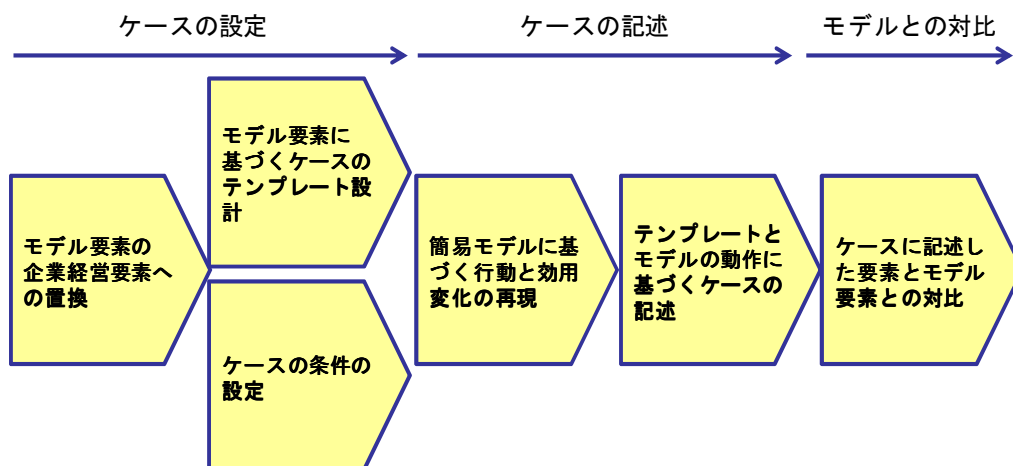


図 4.9 ケースとの接地のステップ

4.5.2 ケースの設定

ケース・テンプレートの設計に先立ち、まず、表 3.2 において説明した ODD プロトコルに基づくモデルの要素を、表 4.1 で企業経営の要素に次のように置き換える。

3 階層の効用関数のランドスケープは社会規範、企業のビジネスモデル、組織構成員個人の価値観に置き換える。NK モデルによって設定した多峰性の効用関数は、複雑な法規制や企業戦略の多様な選択肢を表す。また、階層ごとの効用関数間の相反関係は、企業目標達成と社会規範遵守とのジレンマを表す。効用配分の制度は、企業の評価・報酬制度に、エージェントの個人効用の多様性が、組織の多様性に置き換える。エージェント間のネットワーク構造は、組織内の公式の指示・報告システムや同僚間の非公式のコミュニケーションルートを示す。インフォーマルネットワークの設定ルールは、組織内での従業員間の偶発的な出会いを示す。エージェントが行動状態を変化させる要因となる満足度は、従業員の行動変化の動機づけ要因に置き換える。エージェントの探索学習と模倣学習は、各従業員の自律的な行動変化と模倣を通じた特定の行動パターンの伝播を示す。不完全情報に基づくエージェントの行動選択は、企業組織内で身近な情報や友人の行動に左右される従業員の意思決定に置き換える。これらの結果として生ずる階層間の効用産出量の変化が、企業組織における改善と逸脱の創発であり、社会、組織、個人の各効用産出量の変化は、それぞれ法令遵守度合いの変化、企業収益の変化、従業員のモチベーションの変化、に置き換える。

以上のようなモデルの要素の企業経営の要素への置換によって、モデルに基づくケースの記述を行い易くする。

表 4.1 モデルの要素の企業経営の要素への置換

分類	モデルの要素		企業経営の要素
	デザインコンセプト(ODD)	要素	
環境	基本原則 (Basic Principles)	階層的なランドスケープ: 社会, 組織, 個人の3階層のランドスケープ	社会規範, 企業のビジネスモデル, 個人の価値観
		効用関数(NKモデル): 効用関数の多峰性	複雑な規制, 企業戦略の多様な選択肢
		効用関数(NKモデル): 階層間の効用関数の相反関係	企業目標の達成と社会規範の遵守との間のジレンマ
集団の性質	集団(Collectives)	効用配分: 組織効用の産出量に応じた効用配分	人事制度(評価・報酬)
		集団を構成するエージェントの個人効用関数の多様性	組織の多様性
相互作用(Interaction)	ネットワーク構造 ①フォーマルネットワーク(上下間)を通じたインタラクション ②インフォーマルネットワーク(同位間)を通じたインタラクション (インタラクション: 行動状態と報酬情報の交換)		従業員間のコミュニケーション環境
エージェントの行動規則	偶然性(Stochasticity)	友人選択: インフォーマルネットワークは, ランダムに選択されたエージェント間に形成	従業員同士の出会い・接点
	適応(Adaptation)	行動状態の変化: ・エージェントは自身の満足の向上を目指して行動を変化	従業員の行動の変化 ・自身の成果への自己評価による報酬への期待に基づいて行動を変更
	予測(Prediction)		
	目標・対象(Objectives)	満足度の構成: 社会効用, 報酬, 個人効用の総和	従業員の行動変化の動機づけ要因
	学習(Learning)	①個人の探索による学習(ヒルクライム) ②友人の模倣による学習(行動状態が近い, 報酬がより多い)	①より満足を高める行動の探索 ②友人の行動の模倣
	感知(Sensing)	不完全情報による選択: ①自身の行動状態の周辺の効用の増減 ②友人の行動状態 ③友人が得た報酬	①各個人は身近な情報をもとに行動を決定 ②自分と似た行動を取る友人の行動を参考に自身の行動を決定 ③組織内で高く評価される者の行動を参考に自身の行動を決定
結果	創発(Emergence)	階層間の産出効用量の関係の変化	経営上の成果の変化
	観察(Observation)	①社会効用の産出量の変化の過程 ②組織効用の産出量の変化の過程 ③個人効用の産出量の変化の過程	①法令遵守の度合いの変化, 消費者の利益の変化 ②企業の収益の変化 ③従業員のモチベーションの変化

表 4.1 で置き換えた企業経営の要素を, ランドスケープ, 集団の性質, 構成員の行動, 行動の結果, の枠組みに沿って各要素間の関係を整理しながら配置し, ケースの骨格を表すテンプレートを作成する (図 4.10 参照).

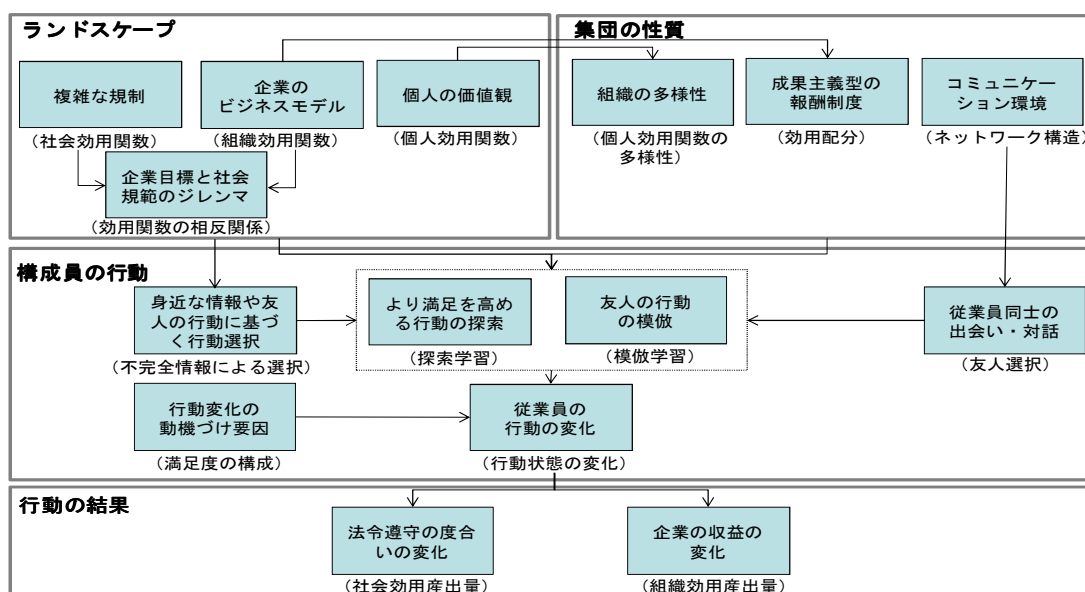
次に, テンプレートに沿って仮想のケースの条件を設定する. 本論文では, 仮想の洋菓子メーカーA社を対象としたケースを記述する. 株式会社不二家のケース (信頼回復対策会議, 2007), 株式会社赤福のケース (コンプライアンス諮問委員会, 2008 ; 樋口, 2009), 雪印乳業のケース (産経新聞取材班, 2002) を参考にした. 基本的な条件設定は次の通りである.

- ・ A社は洋菓子を中心とした菓子類を製造・販売する老舗の菓子メーカーである. 全国に数百店規模の洋菓子店舗およびレストランを展開する.
- ・ 同社は, 市場成長の停滞, プライベートブランドの台頭, および原材料費の上昇などによる利益率の低下が続いており, コストダウンによる利益率向上の必要性に迫られている.

- ・ A 社は、コストダウンを実現するために、まず工場の製品製造ラインにおける原価低減に優先して取り組むこととした。工場における製造工程には様々な原価低減の可能性が存在する一方で、やり方によっては製品の安全性を低下させるリスクもある。
- ・ 食品は、消費者が直接口にするものであるため、製品の安全性に対する規制は厳しく、消費者に健康被害を生じさせたり法令に違反する行為は厳しく処罰される。このため、A 社は、コストダウンを製品安全性の確保と法令遵守を考慮しながら進める必要がある。

以上の条件設定に基づくケースの骨格を図 4.11 に示した。集団の多様性を条件として変化させ、ケース A は一様な集団、ケース B は多様性の高い集団とする。さらに多様性の高い集団（ケース B）において、社会効用関数と組織効用関数の相反を高めたケース C を記述する。効用配分の傾斜は全ケース共通であり、最も組織効用産出量が多いエージェントに総組織効用量の 40% を報酬として付与し、2 番目に大きいエージェントが 35%、3 番目に 25% を付与する。ネットワーク構造も全ケース共通とし、全エージェント間にインフォーマルネットワークを設定する。

尚、ランドスケープと集団の多様性については、NK モデルに基づく条件設定を行う。NK モデルによるケースの条件設定は 4.5.3 において説明する。



() : モデルの要素名称

図 4.10 ケースのテンプレート

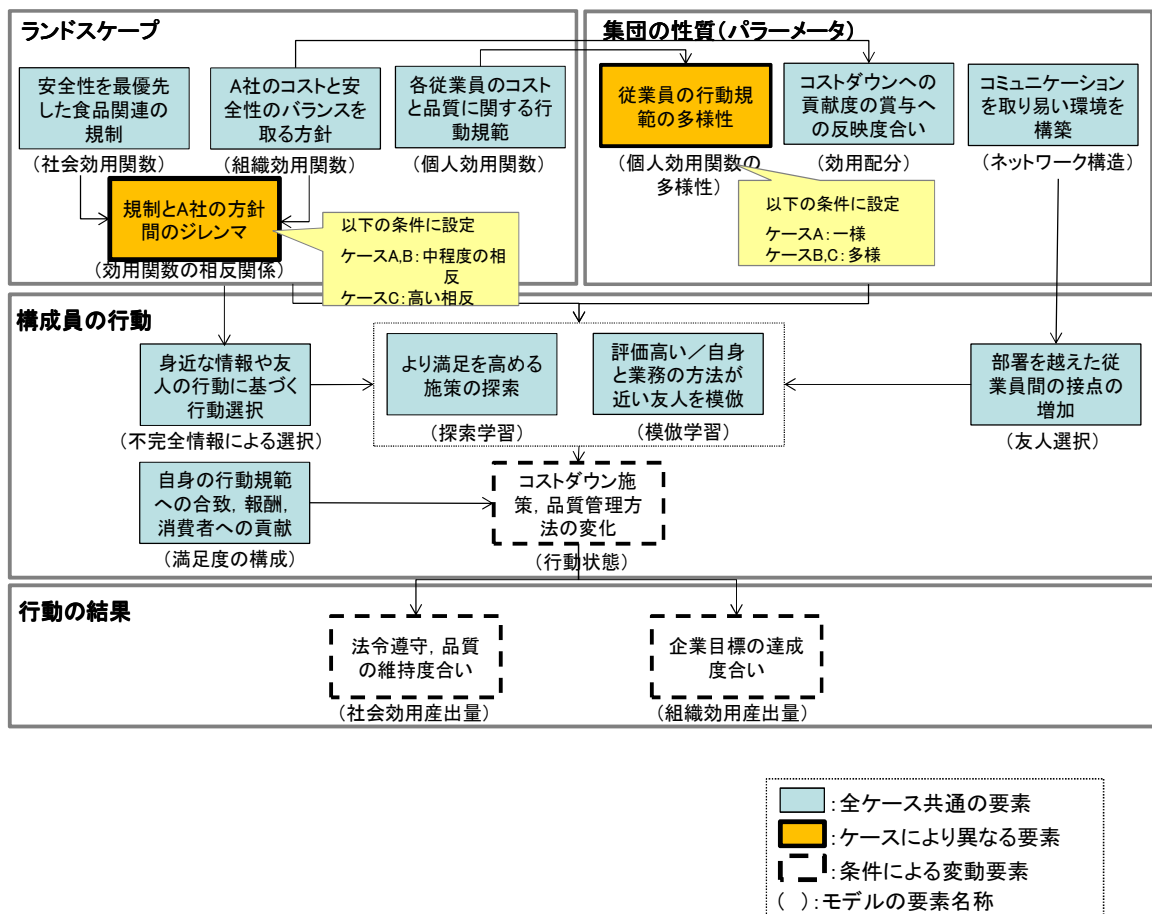


図 4.11 テンプレートに基づくケースの骨格

4.5.3 NK モデルに基づくケースの設定

NK モデルに基づいて洋菓子メーカーA社のケースを設定する。ここでは簡易化のため、 $N=3$ 、 $K=1$ に設定した。A社の工場において、各製品製造ラインが選択し得る施策のパターンを表 4.2 に示す。施策の種類は、1. コストダウン施策、2. 消費期限の設定、3. 品質管理の方法の3種類である。各種類の施策について2通りの選択肢が存在し、0か1の値で表現される。0は原価低減の効果が低いが製品の安全性を維持し易く、1はより原価低減効果が高いが安全性を低下させるリスクが高い選択肢である。表 4.3 に各施策間の依存関係を整理する。 $K=1$ の設定に基づき右隣の施策と依存関係を持たせる。0と0の選択肢の組み合わせは、製品の安全性は高いが原価低減効果が低い組み合わせであり、1と1はその逆の組み合わせである。0と1および1と0は、両者の中間であり安全性と原価低減のバランスを取る組み合わせである。例えば、現場判断によって柔軟に消費期限を設定する方法(選択肢1)は製品の安全性低下のリスクが高いが、細菌検査(選択肢0)という高精度の品質管理方法との組み合わせにより品質低下のリスクを低減できる。一方で、細菌検査の機器にかかるコストによって柔軟な消費期限設定による原価低減の効果は低下する。

表 4.2 NK モデルに基づくケースの設定：施策のパターン

整数値	施策パターン (N=3)					
	1. コストダウン施策		2. 消費期限の設定		3. 品質管理の方法	
	コストへの影響	品質への影響	コストへの影響	品質への影響	コストへの影響	品質への影響
0	製造工程の効率化による 労務費の削減		マニュアルに基づく設定		機材による細菌検査	
	原価低減効果は限定的	製造期間の短縮を伴うため品質を維持し易い	コスト増大の可能性あり	厳格な期限設定のため品質を維持し易い	コストが増大する	問題検出の精度が高いため品質を維持し易い
1	原料廃棄の抑制による 材料費の削減		現場判断に基づく設定		人手による風味検査	
	原価低減効果が高い	原料の使用期限超過により品質低下の可能性あり	柔軟な設定により原価低減の可能性あり	規定を越えた期限設定により品質低下の可能性あり	コストは増大しない	問題検出の精度が低いため品質低下の可能性あり

表 4.3 NK モデルに基づくケースの設定：施策間の依存関係

整数値の組合せ	00	01	10	11
安全性	高	やや高	やや低	低
原価低減効果	小	やや小	やや大	大

4.5.4 モデルの動作の再現：ケース A

4.5.3 で説明した NK モデルに基づいてランドスケープと集団の多様性の条件を設定し、エージェントの行動の変化の過程を再現する。

表 4.4 にケース A の条件を示した。社会は安全性を最優先するため、0 と 0 の施策の組み合わせの評価値が最も高く、1 と 1 への評価値が最も低い。一方、A 社はコストダウンを実現したいが安全性も考慮するため、0 と 1 および 1 と 0 という原価低減と安全性のバランスを取る組み合わせの評価値が高く、安全性と原価低減を追求する組み合わせへの評価値は低い。このように社会と組織の効用関数にはある程度の相反がある。また、工場の製造ライン長を 3 名設定するが、ケース A では全員がコストダウンを推進する行動規範を持つため、1 と 1 の組み合わせへの評価値が最も高い設定である。以上の設定に基づいて、各ライン長が選択する 3 種類の施策の各組み合わせに対する評価値を表 4.4 の下段に示した。

表 4.4 ランドスケープ：ケース A

組合せ (K=1)	評価値				
	社会	組織	個人		
	安全性を重視	低コスト志向だが 安全性も考慮	ライン長① 低コスト志向	ライン長② 低コスト志向	ライン長③ 低コスト志向
00	0.4	0.1	0.1	0.1	0.1
01	0.3	0.3	0.2	0.2	0.2
10	0.2	0.4	0.3	0.3	0.3
11	0.1	0.2	0.4	0.4	0.4
000	0.40	0.10	0.10	0.10	0.10
001	0.30	0.27	0.20	0.20	0.20
010	0.30	0.27	0.20	0.20	0.20
100	0.30	0.27	0.20	0.20	0.20
011	0.20	0.30	0.30	0.30	0.30
110	0.20	0.30	0.30	0.30	0.30
101	0.20	0.30	0.30	0.30	0.30
111	0.10	0.20	0.40	0.40	0.40

表 4.4 に設定したランドスケープのもとでの、各ライン長の行動変化の過程を表 4.5 に説明する。当初は 3 名が異なる施策を選択しているが、ライン長②と③は低コスト志向の個人効用関数と組織効用関数に沿ってより満足度を高める施策を探索した結果、Step2 において社会効用をより下げる施策を選択する。当初最も安全性の高い施策（000）を選択したライン長①は報酬が低いため、高報酬者の行動を模倣するモデルのルールに基づいてライン長②の施策を模倣する。その結果、Step2 でより社会効用を低下させる施策(001)を選択した。しかし、ライン長②と③がより組織効用の高い施策を探索したため報酬が相対的に低くなった①は、更にライン長②の施策を模倣する。Step3 以降では 3 名ともより満足度を高める施策を見いだせないため、施策の微動はあるがほぼ同じ行動状態に収束し、社会効用が低下した状態が継続した。

以上の各ライン長の行動変化の結果生ずる効用産出量の変化の過程を図 4.12 に示す。各ライン長が満足度の向上を目指して探索学習と模倣学習を反復した結果、初期状態よりも組織効用は上昇するが社会効用が低下する逸脱の傾向が生じた。全ライン長が低コスト志向の個人効用関数を持っており、A 社の方針も安全性を考慮はするがコストダウンを志向する組織効用関数であったため、結果的に安全性を重視する社会効用関数が軽視されていく過程が効用産出量の変化から把握できる。

この手動による各ライン長の行動と効用産出量の変化の傾向は、5 章の応用実験において、集団の多様性を変化させた場合のシミュレーションの結果で確認する。

表 4.5 構成員の行動変化の過程：ケース A

Step		1	2	3	4	5
状態の説明		・それぞれが異なったコストダウン施策を取る ・最も安全性を重視した施策を取る①の報酬が低く、満足度が低い	・①は高報酬の②の施策を模倣 ・②、③は更に満足度を上げるために自分のポリシーと企業方針に沿った施策を選択 ⇒その結果、①の報酬は余り上昇しない	・①は更に高報酬の②の施策を模倣 ・②は現状の施策を維持 ・③は周囲の②の施策を模倣 ⇒結果的に、全員同一のコストダウン施策を選択	・①、②はより満足度を上げる施策が見いだせないため現状を維持。 ・③は②の施策を模倣した結果、満足度が低下し、施策を元に戻す。	・①、②、③ともにこれ以上満足度を上げる施策が見いだせず現状を維持する。
ライン長①	行動変更	(初期状態)	模倣学習	模倣学習	現状維持	現状維持
	行動選択	000	001	011	011	011
	社会効用	0.40	0.30	0.20	0.20	0.20
	組織効用	0.10	0.27	0.30	0.30	0.30
	個人効用	0.10	0.20	0.30	0.30	0.30
	報酬	0.16	0.22	0.30	0.30	0.30
	満足度	0.66	0.72	0.80	0.80	0.80
ライン長②	行動変更	(初期状態)	探索学習	現状維持	現状維持	現状維持
	行動選択	001	011	011	011	011
	社会効用	0.30	0.20	0.20	0.20	0.20
	組織効用	0.27	0.30	0.30	0.30	0.30
	個人効用	0.20	0.30	0.30	0.30	0.30
	報酬	0.24	0.33	0.30	0.30	0.30
	満足度	0.74	0.83	0.80	0.80	0.80
ライン長③	行動変更	(初期状態)	探索学習	模倣学習	元に戻す	現状維持
	行動選択	100	110	011	110	110
	社会効用	0.30	0.20	0.20	0.20	0.20
	組織効用	0.27	0.30	0.30	0.30	0.30
	個人効用	0.20	0.30	0.30	0.30	0.30
	報酬	0.24	0.33	0.30	0.30	0.30
	満足度	0.74	0.83	0.80	0.80	0.80

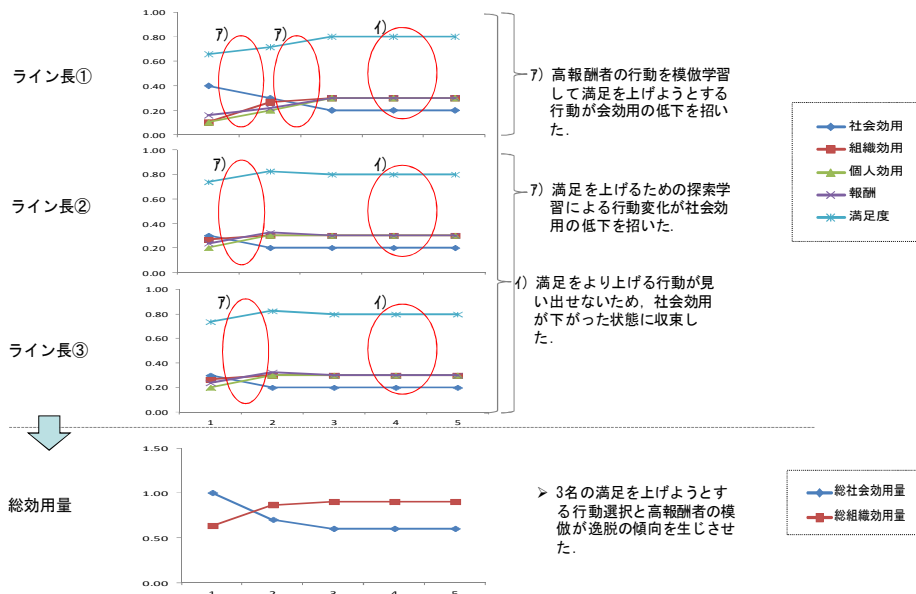


図 4.12 効用産出量の変化の過程：ケース A

4.5.5 モデルの動作の再現：ケース B

表 4.6 にケース B の条件を NK モデルに基づいて示した。社会と組織のランドスケープはケース A と同一の設定である。一方、3 名の製造ライン長の個人効用関数はケース A と異なり、全員異なる設定とする。ライン長①は安全性の高い 00 への評価値が最も高く、他組み合わせへの評価値は低い。③は逆に原価低減効果の高い 11 への評価値が最も高く、他は低い。②は原価低減と安全性のバランスを取る組み合わせへの評価値が高い、という設定である。以上の設定に基づいて、各ライン長が選択する 3 種類の施策の各組み合わせに対する評価値を表 4.6 の下段に示した。

表 4.6 ランドスケープ：ケース B

組合せ (K=1)	評価値				
	社会	組織	個人		
	安全性を重視	低コスト志向だが 安全性も考慮	ライン長① 安全性志向	ライン長② 両立志向	ライン長③ 低コスト志向
00	0.4	0.1	0.4	0.1	0.1
01	0.3	0.4	0.1	0.4	0.1
10	0.2	0.3	0.1	0.4	0.1
11	0.1	0.2	0.1	0.1	0.4
000	0.40	0.10	0.40	0.10	0.10
001	0.30	0.27	0.20	0.30	0.10
010	0.30	0.27	0.20	0.30	0.10
100	0.30	0.27	0.20	0.30	0.10
011	0.20	0.30	0.10	0.30	0.20
110	0.20	0.30	0.10	0.30	0.20
101	0.20	0.30	0.10	0.30	0.20
111	0.10	0.20	0.10	0.10	0.40

表 4.6 に設定したランドスケープのもとでの、各ライン長の行動変化の過程を表 4.7 で説明する。当初は 3 名とも原価低減効果が高いが安全性の低い施策(111)を選択している。ライン長①と②は満足度をより高める施策を探索し、Step2 においてそれぞれより組織効用と社会効用が高い施策を選択した。一方③は低コスト志向の個人効用関数を持つため、これ以上満足度を高める施策を見いだせず、Step2 では施策を変えない。しかし、①と②の施策の変化に伴って相対的に組織効用の産出量が低下したため、Step3 で②の施策を模倣し 110 に変更する。その間に①と②は、より満足度を高める施策を探索し、③は Step4 でその施策を更に模倣するが、満足度が低下して Step5 で施策を元に戻す。

以上の各ライン長の行動変化の結果生ずる効用産出量の変化の過程を図 4.13 に示す。各ライン長が満足度の向上を目指して探索学習と模倣学習を反復した結果、初期状態よりも組織効用および社会効用ともに上昇する改善の傾向が生じた。安全性を志向するライン長①と安全性と原価低減のバランスを志向する②が社会効用と組織効用を初期状態よりも上昇させる施策を探索し、低コスト志向の③がその施策を模倣したことにより組織全体として改善傾向が生じた。

表 4.7 構成員の行動変化の過程：ケース B

Step	1	2	3	4	5
状態の説明		<ul style="list-style-type: none"> ①、②は自身の行動規範(①安全性志向, ②両立志向)をもとに、より満足度を高める施策をそれぞれ探索して選択した結果、社会効用と組織効用を増加。 ③は満足度をより高める施策を見いだせずに現状を維持。 	<ul style="list-style-type: none"> ①、②は満足度をさらに高める施策を、自身の行動規範をもとに、探索し続け、施策を変更。 ③は、より高い報酬を得ている②の施策を模倣。 	<ul style="list-style-type: none"> ①と②はより高い報酬の③の施策を模倣。 ③は自身と近い施策を取っている②を模倣。 	<ul style="list-style-type: none"> ①、②、③は模倣した結果、満足度が下がったため、施策を元に戻す。
ライン長①	(初期状態)	探索学習	探索学習	模倣学習	元に戻す
行動変更	111	011	010	110	010
行動選択	111	011	010	110	010
社会効用	0.10	0.20	0.30	0.20	0.30
組織効用	0.20	0.30	0.27	0.30	0.27
個人効用	0.10	0.10	0.20	0.10	0.20
報酬	0.20	0.30	0.25	0.33	0.25
満足度	0.40	0.60	0.75	0.63	0.75
ライン長②	(初期状態)	探索学習	探索学習	模倣学習	元に戻す
行動変更	111	110	100	110	100
行動選択	111	110	100	110	100
社会効用	0.10	0.20	0.30	0.20	0.30
組織効用	0.20	0.30	0.27	0.30	0.27
個人効用	0.10	0.30	0.30	0.30	0.30
報酬	0.20	0.30	0.25	0.33	0.25
満足度	0.40	0.80	0.85	0.83	0.85
ライン長③	(初期状態)	現状維持	模倣学習	模倣学習	元に戻す
行動変更	111	111	110	100	110
行動選択	111	111	110	100	110
社会効用	0.10	0.10	0.20	0.30	0.20
組織効用	0.20	0.20	0.30	0.27	0.30
個人効用	0.40	0.40	0.20	0.10	0.20
報酬	0.20	0.20	0.33	0.22	0.33
満足度	0.70	0.70	0.73	0.62	0.73

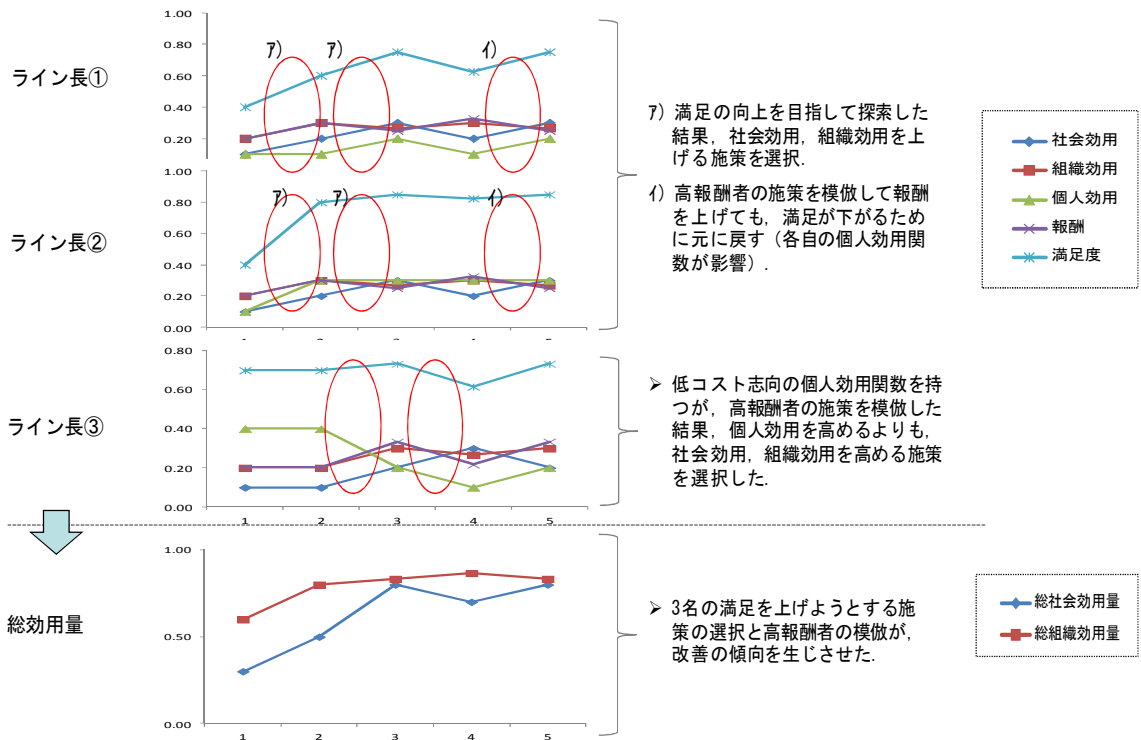


図 4.13 効用産出量の変化の過程：ケース B

4.5.6 モデルの動作の再現：ケース C

表 4.8 にケース C の条件を NK モデルに基づいて示した。製造ライン長 3 名の個人効用はケース B と同様に全員異なる設定とする。一方、社会と組織のランドスケープはケース A および B よりも相反の度合いを高める。社会は安全性を最優先するが、組織はコスト低減を最優先する設定とする。以上の設定に基づいて、各ライン長が選択する 3 種類の施策の各組み合わせに対する評価値を表 4.8 の下段に示した。

表 4.8 ランドスケープ：ケース C

組合せ (K=1)	評価値				
	社会	組織	個人		
	安全性を重視	低コスト志向	ライン長① 安全性志向	ライン長② 両立志向	ライン長③ 低コスト志向
00	0.4	0.1	0.4	0.1	0.1
01	0.3	0.2	0.1	0.4	0.1
10	0.2	0.3	0.1	0.4	0.1
11	0.1	0.4	0.1	0.1	0.4
000	0.40	0.10	0.40	0.10	0.10
001	0.30	0.20	0.20	0.30	0.10
010	0.30	0.20	0.20	0.30	0.10
100	0.30	0.20	0.20	0.30	0.10
011	0.20	0.30	0.10	0.30	0.20
110	0.20	0.30	0.10	0.30	0.20
101	0.20	0.30	0.10	0.30	0.20
111	0.10	0.40	0.10	0.10	0.40

表 4.8 に設定したランドスケープのもとでの、各ライン長の行動変化の過程を表 4.9 で説明する。Step1 において、3 名とも原価低減効果が高いが安全性の低い施策(111)を選択している。ライン長①と③は探索によってより満足度を高める施策を見いだせないため、当初の施策を維持し続ける。③は低コスト志向の個人効用を持つためであり、①は安全志向だが不完全情報のもとで現在の状態の周辺で満足度を高める施策を探索できない。一方、ライン長②はより満足度を高める施策を探索し、Step2 において社会効用を高める施策(110)を選択する。しかしその結果、組織効用の産出量がライン長①と③に比べて低下し、報酬も下がる。よって②はより高報酬の①の施策を模倣し、結果的に Step3 で元のより社会効用の低い施策を選択する。②は以降も社会効用の高い施策を探索するが、報酬が下がって高報酬者を模倣し、社会効用を下げる行動を反復する。

以上の各ライン長の行動変化の結果生ずる効用産出量の変化の過程を図 4.14 に示す。ケース B と同一の多様性を持った組織であっても、各ライン長は社会効用を高める施策にシフトしないため、当初の社会効用が低い状態が継続する。これは、社会と組織のランドスケープの相反が大きいことにより、社会効用を高める施策の選択が、報酬および満足度の低下を招くためである。このようなランドスケープのもとでは、個人効用が多様であっても改善行動が生じ難い。

表 4.9 構成員の行動変化の過程：ケース C

Step		1	2	3	4	5
状態の説明			・①、③は探索によってより満足度を高める施策を見いだせないため現状を維持。 ・②は探索によってより満足度を高める行動を見出す。 ⇒その結果、報酬が低下	・報酬が低下した②は、より高い報酬を得ている①の施策を模倣。 ・①、③は、より満足度を高める施策を見いだせないため現状を維持。	・②は、模倣によって満足度が下がり、施策を元に戻す。 ⇒その結果、報酬が低下 ・①、③は、より満足度を高める施策を見いだせないため現状を維持。	・報酬が低下した②は、より高い報酬を得ている①の施策を模倣。 ・①、③は、より満足度を高める施策を見いだせないため現状を維持。
	全員が、安全性が低くコストダウン効果が大きい施策を選択した状態を初期状態とする。					
ライン長①	行動変更	(初期状態)	現状維持	現状維持	現状維持	現状維持
	行動選択	111	111	111	111	111
	社会効用	0.10	0.10	0.10	0.10	0.10
	組織効用	0.40	0.40	0.40	0.40	0.40
	個人効用	0.10	0.10	0.10	0.10	0.10
	報酬	0.40	0.41	0.40	0.41	0.40
	満足度	0.60	0.61	0.60	0.61	0.60
ライン長②	行動変更	(初期状態)	探索学習	模倣学習	元に戻す	模倣学習
	行動選択	111	110	111	110	111
	社会効用	0.10	0.20	0.10	0.20	0.10
	組織効用	0.40	0.30	0.40	0.30	0.40
	個人効用	0.10	0.30	0.10	0.30	0.10
	報酬	0.40	0.28	0.40	0.28	0.40
	満足度	0.60	0.78	0.60	0.78	0.60
ライン長③	行動変更	(初期状態)	現状維持	現状維持	現状維持	現状維持
	行動選択	111	111	111	111	111
	社会効用	0.10	0.10	0.10	0.10	0.10
	組織効用	0.40	0.40	0.40	0.40	0.40
	個人効用	0.40	0.40	0.40	0.40	0.40
	報酬	0.40	0.41	0.40	0.41	0.40
	満足度	0.90	0.91	0.90	0.91	0.90

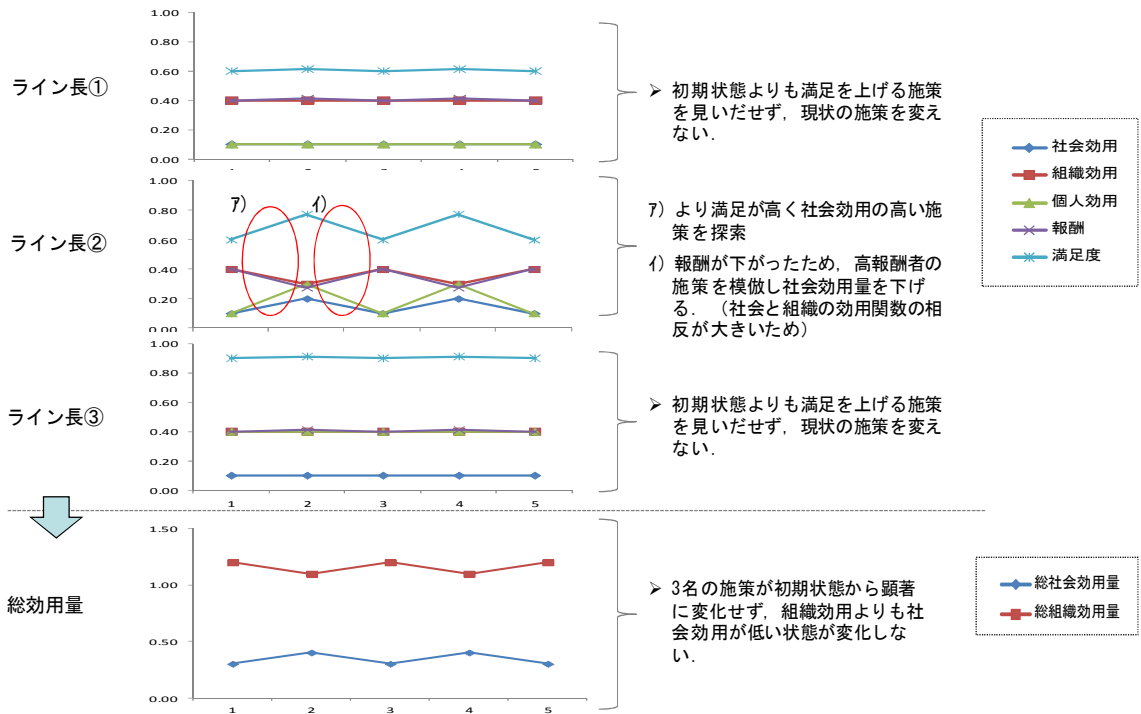


図 4.14 効用産出量の変化の過程：ケース C

4.5.7 ケースの記述

ここで、前述のテンプレートと主導による構成員の行動の再現結果をもとに、モデルの要素に基づいて仮想のケースを記述する。最初に基本的なケースの設定を記述した上で、同一の洋菓子メーカーにおける構成員の行動の変化の過程を、組織の多様性とランドスケープに関する条件を変えた3つのケースによって記述する。ケースAは各ライン長が一様に低コストを志向する設定、ケースBは各ライン長の行動規範が異なる設定、ケースCは各ライン長の行動規範はBと同様だが、社会と組織の方針の相反度合いがより高いという設定である。

尚、以降に記述するケース中の下線部はモデルの要素に基づく記述であり、カッコ内には該当するモデルの要素の名称を記載した。

<基本設定>

A社は洋菓子を中心とした菓子類を製造・販売する老舗の菓子メーカーである。全国に数百店規模の洋菓子店舗およびレストランを展開する。同社は、市場成長の停滞、プライベートブランドの台頭、および原材料費の上昇などによる利益率の低下が続いており、コストダウンによる利益率向上（組織効用産出量）の必要性に迫られている。A社は、コストダウンを実現するために、まず工場の製品製造ラインにおける原価低減に優先して取り組むことにした。工場における製造工程には様々な原価低減の可能性が存在する一方で、やり方によっては製品の安全性を低下させるリスクもある。食品は、消費者が直接口にするものであるため、製品の安全性に対する規制は厳しく（ランドスケープ：社会効用関数）、消費者に健康被害を生じさせたり法令に違反する（社会効用産出量）行為は厳しく処罰される。食品に関する法令としては、衛生上の危害防止を目的とした食品衛生法、および品質に関する適正表示を目的としたJAS法に加え、計量法や健康増進法などが存在している。このため規制が複雑化しており（ランドスケープ：社会効用関数）、従業員にとって規制や法令の全体像を掴むことは難しく、現場で共有されている情報の範囲で判断を行わなければならない（不完全情報）。

工場の各製造ラインで原価低減を推進するに際しては、コストダウン施策、消費期限設定、品質管理方法の3つの手法（効用関数：NKモデル）を検討し判断する必要がある。以下にそれぞれの手法について説明する。まずコストダウン施策は、製造工程の効率化による労務費の削減と、材料廃棄の抑制による材料費の削減に分類される。前者は、例えば生産工程間の無駄の排除であり、原価低減の効果は限定的だが、製造の期間短縮につながるため安全性の低下は生じにくい。後者は、例えば廃棄ロス率の目標管理や真空包装による使用可能期間の延長などであり、顕著な原価低減効果が期待できる反面、安全性を低下させる可能性がある。2番目の消費期限の設定方法には、社内マニュアルに厳密に基づく方法と、マニュアルに依拠しつつ現場の判断で一定の調整を行うやり方がある。3番目の品質管理の方法には、機材を用いた細菌検査と、担当者の経験や感覚に基づく風味検査の方法がある。3つの手法の組み合わせのパターンによって、原価低減の効果と安全性に与える影響が異なる

ってくる (効用関数：NK モデル)。例えば、材料廃棄の抑制は安全性を低下させる可能性があるが、マニュアルに基づく厳密な消費期限設定との組み合わせによって安全性低下のリスクを抑制できる。しかし、このような組み合わせは原価低減の効果を限定することになる。現場の判断で一定の調整を行って消費期限を設定する方法は製品廃棄を削減する効果が高いが安全性低下の可能性も高い。これを高精度の細菌検査と組み合わせることで、安全性低下のリスクを抑制できるが、検査機器のコストがかかるため原価低減効果も抑制される。A社の工場の各ラインでは、以上の手法の組み合わせによる原価低減効果と製品の安全性への影響を考慮しながら施策を決定し、コストダウンに取り組むことになった。

また A 社は、従来のトップダウンでの一律的な取り組みの限界を感じていたため、今回は現場主導での活動を促進する方針をとった。そこで、現場コストダウンに対する従業員の意欲を高めるために人事制度の改革を行った。各工場の製造ラインにコストダウン施策を決定する権限を委譲し、賞与額が会社目標への貢献度にある程度連動する制度を導入したのである (効用配分)。このため、従業員は四半期ごとの実績評価の結果に敏感になった。さらに同社は、人事制度の改革と並行して社内のコミュニケーション活性化の施策を打っていった。例えば、工場内にオープン型の喫煙スペースを設置し、喫煙者および非喫煙者を含めてコミュニケーションを取り易い環境を実現した。また、各工場の繁閑に合わせて、工場間で柔軟に人員を流動させる制度を構築した。これにより、部署を越えた従業員間の接点が増加し (コミュニケーション環境)、現場レベルの情報共有が促進されるようになった。

<ケース A>

A 社は、消費者が求める製品安全性の確保と法令遵を考慮しながらコストダウンを進める方針を打ち出した (ランドスケープ：組織効用関数)。同社は安全性と原価低減のバランスを取ろうとしたが、高い安全性を要求する法令の思想とは合致しない部分もあった (ランドスケープ：効用関数の相反関係)。

一方 A 社内では、各工場の従業員に対しコストダウンへの意識を強化するための教育プログラムが実施された。これにより、従業員はコストの削減を最優先する行動規範を共通して持つようになったのである (ランドスケープ：個人効用関数)。このような状況下で、同社のある工場では、次のような過程でコストダウンへの取り組みが進行した。

この工場では、主要な製造ラインを担当するライン長が 3 名おり、当初はそれぞれ異なる原価低減施策を推し進めていた。ライン長①は、最も安全を重視した施策を採用していた。工程の無駄を省く手法でコストダウンを図り、マニュアルに基づいて厳格に消費期限を設定し、細菌検査を行っていたのである (行動状態)。ライン長②は、①と同様のコストダウンと消費期限設定方法を採用していたが、細菌検査にかかる費用を懸念して人手による風味検査を行っていた (行動状態)。またライン長③は、材料廃棄の抑制施策を採用していたが、マニュアルによる消費期限と細菌検査によって製品の安全性を確保していた (行動状態)。このような当初の状態において、最も安全性を重視していた①は業績評価が②、③より低く

かったために、人事制度の改訂に伴って賞与が下がった（効用配分）。これに不満を抱いた①は、より高い賞与を得た②の方法を真似て、品質管理方法を風味検査に切り替え、細菌検査を取り止めたのである（模倣学習）。一方で、ライン長②と③も、より原価低減効果の高い方法を日々の探求を通じて見出し（探索学習）、消費期限の設定をマニュアルに依拠した方法から現場の判断で調整するやり方に変更した（行動状態）。この方法は、製品の安全性を低下させる可能性があるものだが、業績評価が上がる期待があり、且つコストダウンを最優先する自身のポリシー（個人効用関数）にも合致していた。

このように②と③が施策を変更して会社目標への貢献度を上げたために、①は②の方法を模倣してより原価を低減したにも関わらず業績評価は相対的に②、③よりも低くなってしまった（効用配分）。そこで①は、自分より高い評価を得ている②の方法を更に模倣することにした（模倣学習）。これと並行して③も日々のコミュニケーションを通じて把握していた自分とやり方の近い②の施策を取り入れることとし（模倣学習）、この時点で、全ライン長が、工程の無駄を削減しながら、消費期限を現場判断で調整し、人の感覚に依存した風味検査で品質を管理する方法をとるようになった（行動状態）。

3人は、変更した方法よりも製品の安全性を向上させる（社会効用産出量）方法があることは認識していた。しかしその方法に変更すると、会社からの評価が下がって賞与が減額される懸念や自身のポリシーに反して納得感が下がる（満足度の構成）可能性あることから、以降は施策を変更しなくなった。以上の過程を経て、当初よりも会社目標の達成度は向上（組織効用産出量）したが、製品の安全性が低下する（社会効用産出量）可能性の高い施策が工場全体として常態化した。

ある時、保健所の抜き打ち検査を受けた当該工場では、厚生労働省の定めたガイドラインである洋生菓子の衛生規範に定められた値を大幅に超過する細菌が検出された製品や、社内基準を超過した消費期限を表示した製品を出荷していたことが判明した。これらにより、A社は厳重注意を受けるとともに、現場レベルでの判断による衛生基準の緩和や従業員の消費期限への認識不足を問題点として指摘された。また、報道を通じて事情を知った消費者から苦情が殺到し、一定期間の商品販売停止を余儀なくされた。

〈ケース B〉

A社は専門的なノウハウの強化によって利益構造を転換することを目的に人材採用についての施策を打った。専門性を有する即戦力の中途採用者を増やし、多様な経験とスキルを持った人材を現場の要所に配置したのである。

当初、A社工場の製造ラインでは、皆が同じ方法で原価低減に取り組んでいた。いずれのラインでも、材料廃棄を抑制しながら現場判断で消費期限の設定を調整し、品質管理はコストのかからない人手での風味検査を行っていた（行動状態）。会社のコストダウン方針を重視し過ぎて、極端に原価低減を推し進める施策が推し進められていたのである。

そうした状況下で、新たに採用された人材が配属された結果、ある工場では3つの製造

ラインが、それぞれ独自の管理ポリシーをもった人材によって管理されることとなった。ライン長①は製品の安全性を最優先するタイプであり、②は原価低減と安全性のバランスを確保することを重視、③は原価低減を最優先し安全性を軽視するタイプ（ランドスケープ：個人効用関数）となったのである。このような体制のもと、同工場では次のような過程でコストダウンへの取り組みが進行した。まず①は、現状の方法が自身のポリシーと合わない（個人効用関数）ことが分かったが、その時点で考えられる施策変更の選択肢には自分のポリシーに合致するものが見い出せなかったため（不完全情報）、まずは、現施策より安全性が高く会社目標に貢献できる方法に変更した（行動状態）。その結果、自分自身の納得感は変わらなかったが（個人効用関数）、安全性が向上し報酬も増えたため満足度は向上した（満足度の構成）。一方②は、自分のポリシーにより合致し、かつ安全性と会社目標への貢献度をともに上げられる方法を見出し施策を変更した（行動状態）。その結果、満足度が大きく向上した。しかし③は、原価低減を最優先する自身の考え方に現状のやり方が合っていると感じた。このため、安全性や会社目標への貢献度をより向上させる選択肢があることは分かったものの、現状の方法を変更しないことにした（行動状態）。しかし、その結果①や②よりも業績評価が下がってしまった（効用配分）。そこで、③は日々のコミュニケーションを通じて（友人選択）把握していた②のやり方を真似て、より評価を上げようとした（模倣学習）。一方、①と②は既に変更した施策の実施を通じて更に自分の満足を高める方法を独自に見出し（探索学習）、会社目標への貢献度は若干低下するが、より安全性の高い施策に変更した（行動状態）。しかし、③の施策に比較して①と②のやり方は会社目標への貢献度が低くなり業績評価が下がったため、①と②はより高い評価を得た③のやり方を模倣（模倣学習）し、安全性が多少低い方法に変更した（行動状態）。これにより②と③の評価は上がったものの、安全性を下げたことによって自身の満足度は低下した（満足度の構成）ため、元の方法に戻した（行動状態）のである。この間③は、日々の接点を通じて把握（友人選択）していた②の施策を真似し（模倣学習）、より安全性の高い施策の実施を試みた（行動状態）が、評価がやや下がりかつ自身のポリシーとの乖離が大きくなったために満足度が低下し（満足度の構成）、元の施策に戻した（行動状態）。

このように、ライン長①と②は自身の満足をより高める施策を探求した結果、当初より製品の安全性が高く会社目標への貢献度も高い施策を見出した。ライン長③は初めのうちはコストダウンを最優先する自身のポリシーに合う安全性を軽視した施策を実施していたものの、②や③に比較して評価が下がったことなどにより、より安全性が高く会社目標に合致した施策を模倣した。結果的に、この工場は会社目標への貢献度を上げ（組織効用産出量）ながら、製品の安全性も当初の状態より高めた（社会効用産出量）のである。

〈ケース C〉

A社は、専門性を有する即戦力の中途採用者を増やし、多様な経験とスキルを持った人材を現場の要所に配置した。同時にA社は、コストダウンを徹底して追求する方針と目標を打ち出した（ランドスケープ：組織効用関数）。これにより同社の従業員は、高い製品安全性を要求

する法令とコストダウンを最優先する会社方針とが矛盾した状態（ランドスケープ：効用関数の相反関係）に置かれることとなった。

当初、A 社工場の製造ラインでは、皆が同じ方法で原価低減に取り組んでいた。いずれのラインでも、材料廃棄を抑制しながら現場判断で消費期限の設定を調整し、品質管理はコストのかからない人手での風味検査を行っていた（行動状態）。会社のコストダウン方針を重視し、極端に原価低減を推し進める施策が推し進められていた。

そうした状況下で、新たに採用された人材が配属された結果、ある工場では3つの製造ラインが、それぞれ独自の管理ポリシーをもった人材によって管理されることとなった。ライン長①は製品の安全性を最優先するタイプであり、②は原価低減と安全性のバランスを確保することを重視、③は原価低減を最優先し安全性を軽視するタイプ（ランドスケープ：個人効用関数）となったのである。このような体制のもと、ライン長②は、現状の方法が自身のポリシーと合わない（個人効用関数）ことを認識し、より安全性が高く自身のポリシーに合う方法である細菌検査を取り入れた（行動状態）。安全志向のライン長①も現状が自身のポリシーに反することを認識していたが、その時点で考えられる施策変更の選択肢には自分のポリシーに合致するものが見い出せなかったため（不完全情報）、やむなく現状の方法を維持した（行動状態）。一方低コスト志向の③は、今のやり方が最も自分のポリシーに合っており十分に満足していたため、現状を維持した（行動状態）のである。このような状況において、高精度だがコストのかかる細菌検査を取り入れた②の会社目標の達成度は①と③に比べて見劣りする結果となり、業績評価が下がってしまった（効用配分）。そこで②は、より高評価を得た①のやり方を真似ることにした（模倣学習）結果、精度の高い細菌検査を取り止めて元の風味検査に戻ってしまった（行動状態）のである。このように、A 社の工場には独自の管理ポリシーをもった人材が配属されたにも関わらず、工場全体として顕著な施策の変更は生じず、当初の極端な原価低減施策が維持され続けてしまった。

4.5.8 ケースにおけるモデルの説明範囲

以上のケースに記述された要素とモデルに含まれる要素とを表 4.10 において比較する。「モデル化の対象外」に記載した項目は、ケースとしての完結性を実現する上で、モデルの要素以外に追記した内容である。ケースに記述した要素の内、法令の具体的な内容、市場の説明、A 社の経営のパフォーマンス（利益率の低下）、報酬以外の人事施策（採用、教育）、コミュニケーション環境の具体的な内容、および逸脱の傾向が強まった結果実際に起きた事象、についてはモデルの対象外となっている。ケースとしての完結性を実現する上で、モデルの要素で説明し切れない範囲が生ずる点は、本モデルの説明範囲の限界である。

表 4.11 では更に、モデルに基づいて記述した仮想ケースと、株式会社不二家の不祥事事例を記述したケース（信頼回復対策会議，2007）との説明範囲を比較した。この比較の目的は、モデルに基づいて記述したケースと実例に基づくケースとの記述範囲の相違を分析することである。表中の網掛けは、相対的に記述が不足する部分を示している。比較した

結果、実ケースはモデルに基づく仮想ケースよりも、システムや工場の衛生状態などの環境に関する説明が充実していることが分かった。また従業員の違法行為の詳細な経緯が描かれており、最終的に企業が指摘された法令違反や措置の内容も詳細である。逆に見れば、これらはモデルに基づく仮想ケースの記述範囲の限界を示している。

一方でケースとの接地を通じて、モデルの要素に基づいてケースを記述するメリットは次の3点であると認識した。まず1点目は、改善のケースと逸脱のケースに共通する要素と異なる要素とを分離し識別できることである。実際の不祥事を記述したケースでは、逸脱の要素に焦点が当たり、改善と共通する要素が記述されることは少ない。2点目のメリットは、ランドスケープや集団の性質という条件下で、行動規範が異なる組織構成員の探索学習と模倣学習の過程を記述できることである。実ケースでは、従業員の違法行為の詳細や、その結果生じた法令違反の内容については詳細に記述されているが、個々の従業員が行った探索や模倣のプロセスとその動機づけ要因については記述されていない。また実ケースでは、個々の従業員の行動規範の相違が記述されず、組織全体としての「コストダウンへの強い意識」のみが指摘される。尚、組織的な不祥事の先行研究（間嶋，2007）でも、組織構成員の行動変化の過程の記述は行われているが、どのような探索学習や模倣学習が行われたのかまでは言及されない。これらと比較して、モデルに基づくケースでは、表4.5、4.7、4.9に示したように、各組織構成員の探索学習と模倣学習を通じて改善と逸脱が生ずる一連の過程を記述したことが利点である。3点目のメリットは、社会、組織、個人の3階層のランドスケープ間の関係が構成員の行動とその結果に影響することを記述したことである。ケースCに示したように、多様性の高い組織であっても、ランドスケープの相反度合いが高いと改善行動が生じ難い。実ケースでは、集団の性質と構成員の行動がもたらす効果が、階層的なランドスケープ間関係に依存することへの言及が不十分である。組織体逸脱の先行研究には、企業が置かれている環境要件が逸脱行為に与える影響への指摘があるが（宝月，2006）、社会、組織、個人のランドスケープ間関係については言及されていない。

表 4.10 記述した仮想ケースにおけるモデルの説明範囲

分類	デザインコンセプト(ODD)	企業経営の要素	ケースの要素	
			モデル化の対象	モデル化の対象外
環境	基本原則 (Basic Principles)	社会規範	食品に関する法規制の方針(安全性志向であること)	法令の具体的な内容(食品衛生法、JAS法、計量法、健康増進法)
		企業の戦略	コストと安全性のバランスを取るというA社の方針	
		個人の価値観	コストと品質に関する従業員の行動規範	
		市場		市場の停滞、プライベートブランドの台頭、原材料費の上昇
		企業目標と社会規範との間のジレンマ	業績不振によるコスト削減目標と、品質管理に関する法令遵守のジレンマ	
集団の性質	集団(Collectives)	企業組織の生産性		利益率の低下
		人事制度 -評価・報酬 -採用 -教育	企業目標への貢献度が評価され給与に反映される制度	
				専門性を有する中途採用者の増加
				コストダウンへの意識を強化する教育プログラム
組織の多様性	従業員の行動規範の多様化/一様化			
相互作用(Interaction)	従業員間のコミュニケーション環境	コミュニケーションを取り易い環境	コミュニケーション環境の具体的な内容(工場内のオープン型の喫煙スペース、工場間で人員を流動させる制度)	
エージェントの行動規則	偶然性(Stochasticity)	従業員同士の出会い・接点	部署を越えた従業員間の接点の増加	
	適応(Adaptation)	従業員の行動の変化	A社従業員のコストダウン施策、品質管理方法の変化	
	目標・対象(Objectives)	従業員の行動変化の動機づけ要因	自身の行動規範への合致、報酬、消費者への貢献	
	学習(Learning)	①より満足度を高める行動の探索 ②友人の行動の模倣	①各従業員は、より満足度を高める原価低減施策を追求 ②組織目標への貢献度が高く評価の高い従業員の行動を模倣	
	感知(Sensing)	①各個人は身近な情報をもとに行動を決定 ②自分と似た行動を取る友人の行動を参考に自身の行動を決定 ③組織内で高く評価される者の行動を参考に自身の行動を決定	①従業員は消費期限に関する法令を十分認識していない ②周囲の製造ラインでの消費期限延長方法 ③同僚の従業員の評価	
結果	創発(Emergence)	従業員の行動による効用の変化	改善と逸脱の創発	
	観察(Observation)	経営上の成果の変化	①法令遵守の度合いの変化、消費者の利益の変化 ②企業目標の達成度の変化	
		発生した事象		・洋生菓子の衛生規範に定められた値を大幅に超過する細菌が検出された製品の出荷 ・社内基準を超過した消費期限を表示した製品の出荷 ・保健所からの厳重注意 ・消費者からの苦情の殺到による商品販売停止

表 4.11 本モデルに基づく仮想ケースと実ケースの説明範囲の相違

分類	デザインコンセプト(ODD)	企業経営の要素	モデルをベースに記述したケース	実例をもとに記述したケース 株式会社不二家信頼回復対策会議 (2007)
環境	基本原則 (Basic Principles)	社会規範	食品に関する法規制の方針(安全性志向であること)	・法令の具体的な内容(食品衛生法, JAS法) ・生洋菓子の衛生規範(旧厚生省の指導基準)
		企業の戦略	コストと安全性のバランスを取るというA社の方針	期限切れ材料は使用しないという社内規定 (但し, マニュアル化に不備)
		個人の価値観	コストと品質に関する従業員の行動規範(個人単位)	・工場のコストダウンへの強い意識(組織単位) ・余った原料は捨てられないという意識(組織単位)
		企業目標と社会規範との間のジレンマ	業績不振によるコスト削減目標と, 品質管理に関する法令遵守のジレンマ	
		その他の環境要素		・生産管理システムの不備 ・工場の不衛生(ネズミや虫)
集団の性質	集団(Collectives)	人事制度 - 評価・報酬 - 教育	企業目標への貢献度が評価され賞与に反映される制度	
		組織の多様性	従業員の行動規範の多様化/一様化	社内規定に関する教育が不徹底 (※個人の行動規範が組織単位に記述されていることから, 一様化が推察される)
	相互作用(Interaction)	従業員間のコミュニケーション環境	コミュニケーションを取り易い環境	品質のチェック体制の不備(品質管理担当の人数が少ない)
エージェントの行動規則	偶然性(Stochasticity)	従業員同士の出会い・接点	部署を越えた従業員間の接点の増加	工場間で生産を相互補完する体制
	適応(Adaptation)	従業員の行動の変化	A社従業員のコストダウン施策, 品質管理方法の変化	商品廃棄に関する社内基準の延長の経緯(詳細に記述)
	目標・対象(Objectives)	従業員の行動変化の動機づけ要因	自身の行動規範への合致, 報酬, 消費者への貢献	
	学習(Learning)	①より満足度を高める行動の探索 ②友人の行動の模倣	①各従業員は, より満足度を高める原価低減施策を追求 ②組織目標への貢献度が高く評価の高い従業員の行動を模倣	
	感知(Sensing)	①各個人は身近な情報をもとに行動を決定 ②自分と似た行動を取る友人の行動を参考に自身の行動を決定 ③組織内で高く評価される者の行動を参考に自身の行動を決定	①従業員は消費期限に関する法令を十分認識していない ②周回の製造ラインでの消費期限延長方法 ③同僚の従業員の評価	・在庫処分ルールに関するマニュアル化の不備 ・消費期限に対する理解と認識の低さ ・細菌検査への理解不足と判断の甘さ ・衛生規範遵守への認識の甘さ
結果	創発(Emergence)	従業員の行動による効用の変化	改善と逸脱の創発	
	観察(Observation)	経営上の成果の変化	①法令遵守の度合いの変化, 消費者の利益の変化 ②企業目標の達成度の変化	
		発生した事象		・法的措置の具体的な内容, 違反した条項の詳細

□: 記述が無い, あるいは相対的に記述が不足

第5章 方法論の応用

5.1 はじめに

本章では、集団の多様性、エージェント間のインフォーマルネットワークの量、エージェントへの貢献度に応じた報酬配分の傾斜度合い、という3つの条件のもとに詳細なシミュレーション実験を実施し、その結果を関連領域の先行研究の主張と比較する。その上で、実験から見出した次の3点を明らかにする：1) 多様性の向上が改善の促進要因になる。一様な組織は改善状態と逸脱状態への分散が高い。2) インフォーマルコミュニケーションの増加は、多様性の高い集団では改善を促進するが、一様な集団では逸脱を促進する。3) 報酬の傾斜を一定水準まで高めるに伴い改善が生ずるが、一定水準を越えると逸脱が生ずる。

研究の対象となる問題と各応用実験との関係は図5.1の通りである。応用実験Ⅰでは集団の多様性、応用実験Ⅱではエージェント間のインフォーマルネットワークの量と集団の多様性、応用実験Ⅲでは成果に応じた報酬配分の傾斜度合いをパラメーターとして操作する。各パラメーターの操作による効用産出量の変化を観察し、エージェントの行動変化の過程を分析する。

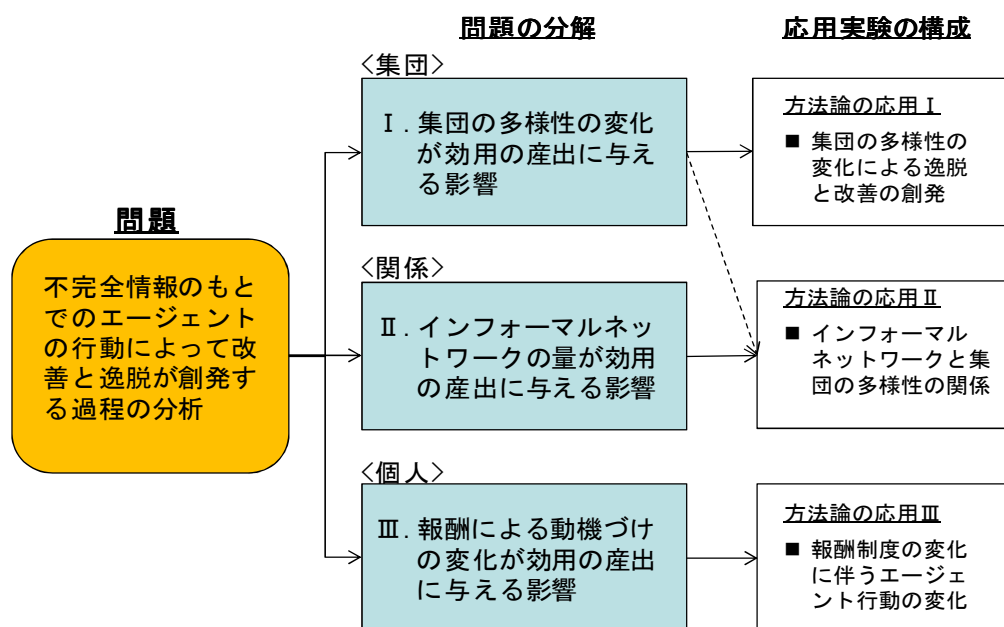


図 5.1 分解した問題と応用実験の関係

5.2 応用実験の構成

応用実験は、表 5.1 に示す通り対象とする問題ごとに 3 つのテーマから構成する。実験 I では、組織の多様性をパラメーターとして変化させた場合の組織効用と社会効用の産出量の変化を観察し、改善と逸脱の創発の過程を分析する。実験 II では、インフォーマルネットワークの量を変化させ、組織効用と社会効用の産出量の変化を観察する。加えて、集団の多様性を条件として組み合わせ、多様性の異なる集団でのインフォーマルネットワークの効果の相違を分析する。実験 III では、エージェントが産出する組織効用量に応じた報酬配分の傾斜を変化させ、組織効用と社会効用の産出量の変化を観察する。

表 5.1 各応用実験の目的

	応用実験 I	応用実験 II	応用実験 III
タイトル	組織の多様性の変化による逸脱と改善の創発	インフォーマルネットワークと集団の多様性の関係	報酬制度の変化に伴うエージェント行動の変化
目的	集団の性質の変化に伴う、各階層の効用産出量の変化から逸脱と改善が創発する過程とその要因を分析	インフォーマルネットワークの増減が効用産出に与える影響を、集団の多様性の変化と関連づけて分析	業績に応じた効用配分の傾斜度合いの操作による効用産出量の変化の過程とその要因を分析

5.3 応用 I : 集団の多様性の変化による逸脱と改善の創発

5.3.1 実験の背景と目的

応用実験 I では、集団の多様性を実験のパラメーターとして漸次変化させる。そして、多様性の変化に伴う各階層の効用産出量の増減の過程を分析し、逸脱と改善を促進させる要因を見出すことが、本実験の目的である。

集団の多様性を扱った研究としては、ダイバーシティ・マネジメントおよび組織体逸脱が挙げられる。

ダイバーシティ・マネジメントは、例えば、性別、年齢、人種などの観点で多様性の高い組織の優位性について分析している。多様性が高い集団は、能力の高い人々によって構成される集団よりも高いパフォーマンスを示すとの結果を導いた先行研究も存在する (Hong and Page, 2004; Page, 2007)。また、組織体逸脱の先行研究には、組織構成員の多様性が集団の凝集性を緩和し、組織体逸脱を抑制する要因になり得るとの指摘がある (岡本・今野, 2003)。前述した、赤福の不祥事のケースでは、法律違反に至るまでの過剰な材料廃

棄の回避を行った要因の一つとして、商品の廃棄を罪悪視する強力な企業文化の存在が指摘されており（樋口，2009）、組織の多様性が低下していた可能性がある。

多様性のもたらすメリットへの期待から、研究や企業経営でのダイバーシティ・マネジメントへの取り組みが増えているが、集団の多様性と組織のパフォーマンスとの関係は十分に分析されていないと認識している。先行研究の中にも、構成員の多様性と企業の財務などのパフォーマンス向上との関係を扱ったものは数多く存在するものの、「それは相関だけであって、根底となる理論が構築されているわけではない」との指摘がある（谷口，2005）。組織体逸脱やサイバースケードの研究では、集団の同質性が不祥事や排他的な行動の要因であることへの指摘はあるものの、多様である場合と同質である場合とで集団の構成員の行動がどのように異なるのかが分析されていない。

本実験では、以上の先行研究の内容と問題意識をもとに、集団の多様性と組織効用および社会効用の産出量との関係を分析する。

5.3.2 実験条件

実験の条件を表 5.2 に示す。

組織の多様性は、共通の個人効用関数を持つエージェント数を加減することによって変化させる。一様な組織では全エージェントの個人効用関数が共通しており、100%の多様性をもつ組織では全エージェントがそれぞれ固有の個人効用関数を持つ。尚、多様度以外の条件は、効用関数間の相反 0.4、友人数 350、報酬配分の傾斜次数は 2 に固定した。

表 5.2 応用実験 I の条件

実験条件	応用実験Iの設定	
ランドスケープ(効用関数の相反率)	0.4	固定
組織規模(エージェント数)	39	固定
組織ヒエラルキー	3階層	固定
多様性	0% - 100%	変動
報酬制度の傾斜次数	2	固定
非公式ネットワーク(友人数)	350	固定

5.3.3 実験結果：効用産出量の推移

図 5.2 は、多様性を次第に向上させた場合の組織効用量と社会効用量の推移である。組織の多様性を向上させるに従って、社会効用の平均産出量が増加し、組織効用の平均算出量も若干増加しており、改善現象が創発する傾向にある。

実験結果について考察する。まず組織効用量が若干増加したのは、多様化に伴って、報酬の多いエージェントの行動がより模倣されたためである。多様化によってエージェント

間の個人効用関数が異なると、個人効用が高まる行動状態がそれぞれ異なるため、行動状態に近いエージェントは少なくなる。そのため、行動状態の近さよりも報酬の多さが模倣の条件として優先される。

次に、社会効用量が増加したのは、多様化に伴って、報酬のみならず社会効用産出量の増加が、模倣成立の条件になり易くなったためである。本モデルにおいて、エージェントの満足度は、個人効用の産出量、組織から得る報酬（組織効用の産出量に応じて配分）、および社会効用量産出量の総和によって決定される。またエージェントは、模倣後において、模倣前より満足度が下がると、行動状態を模倣前に戻す。エージェントの多様性が高いと、個人効用が高まる行動状態がそれぞれ異なるため、図 5.3 に示したように、相互の模倣によって個人効用量が低下する。個人効用量の低下を補完して満足度を維持するために、報酬のみならず社会効用産出量の増加が、模倣成立の条件になる。その結果、社会効用産出量が高まり、逸脱が抑制される。以上の結果は、組織の多様性の強化が改善を促進すること、および社会効用と個人効用の産出量のトレードオフがその要因であることを示す。さらに、集団の多様性をパラメーターとして操作することで、改善と逸脱の双方が生じることを示している。

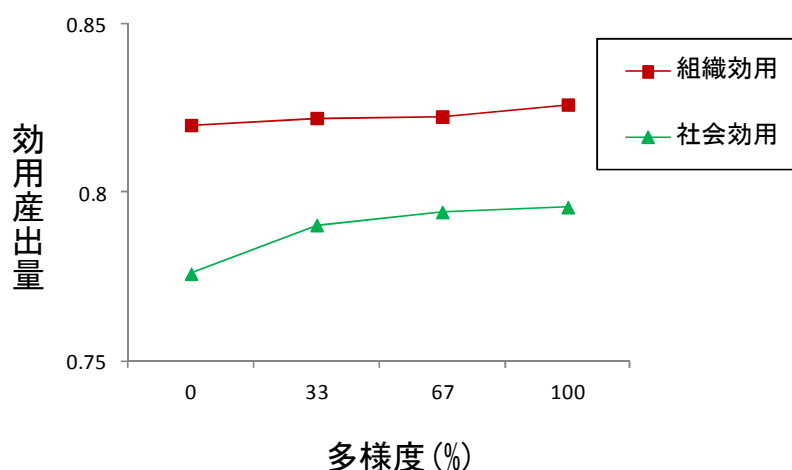


図 5.2 多様性の向上に伴う組織効用と社会効用産出量の変化

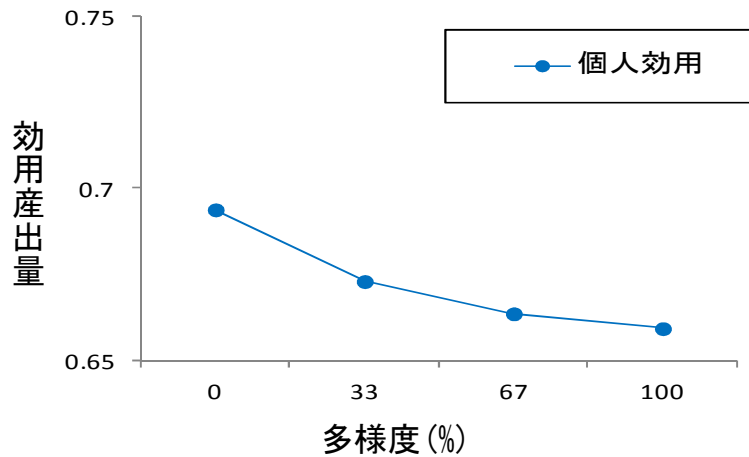


図 5.3 多様性の向上に伴う個人効用産出量の変化

5.3.4 実験結果：効用産出量の分布

次に、社会効用と組織効用産出量の各シミュレーションごとの分布をもとに、組織の多様性の相違による傾向を分析する。多様度 0%と 100%の組織効用と社会効用の産出量の分布を図 5.4 と図 5.5 に示す。シミュレーション回数は 1000 回である。分布を見ると、多様性が高い組織の方が一様な組織に比較して、収束傾向にある。

実験結果について考察する。前述のように、多様性が低いとエージェント間の模倣によって個人効用が高まり易いため、個人効用と報酬によって満足度が十分増加する。よって、社会効用量が低下しても模倣が成立し易くなる。

但し、個人効用関数と社会効用関数が一致している場合は、エージェント群が個人効用関数のピークに収束した結果、社会効用関数のピークも選択することになり、結果的に改善現象が生ずる。実験では、シミュレーションごとに効用関数をランダムに与えるため、一様な組織においては、効用関数の関係次第で、偶発的に極端な改善と組織体逸脱が生ずる。しかし、一様な組織のエージェントは、個人効用関数と社会効用関数が一致しない限り、社会効用を軽視する行動を取り易いため、1000 回のシミュレーションの平均を取ると社会効用産出量が低下する。

以上より、一様な組織は、多様性の高い組織よりも逸脱傾向が強いだけでなく、組織行動が逸脱と改善の両極に振れやすいことが分かった。

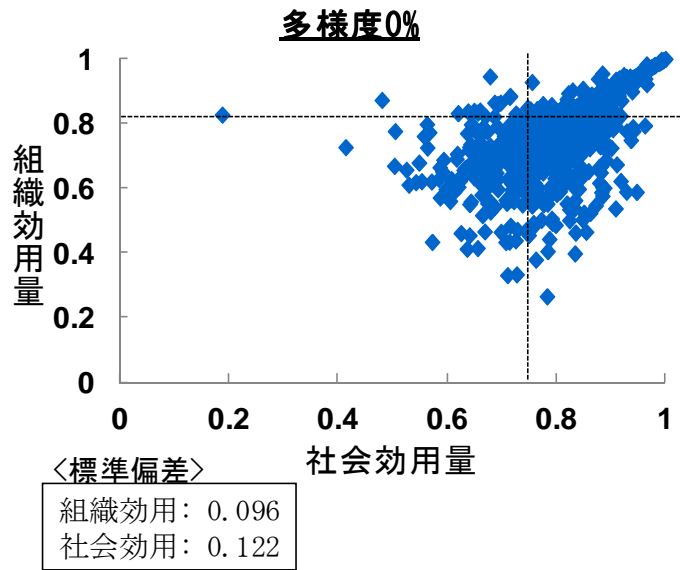


図 5.4 組織効用と社会効用産出量の分布（多様度 0%）

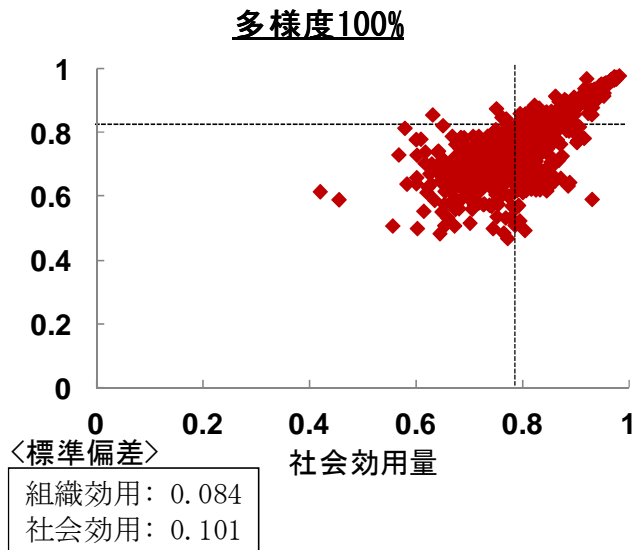


図 5.5 組織効用と社会効用産出量の分布（多様度 100%）

5.3.5 考察

本実験では、1つの集団において多様性が変化することに伴い、改善から逸脱へ、あるいは逸脱から改善へと構成員の行動が変化する過程を示した。また一様な集団は、逸脱を生ずる傾向があるが、社会のニーズと構成員の価値観が一致した状態では、顕著な改善を生むことも示した。更に、一様な集団ほどではないが多様な集団でも改善と逸脱の分散が見られることを示した。これらの結果を階層間の効用産出量の視点から表 5.3 にまとめる。但

し、これらの結果は、次の条件の下で得られたものである。

- ・満足度の構造： エージェントの満足度は、個人効用の産出量、組織から得る報酬、社会効用量産出量の総和によって決定される。
- ・報酬制度： 報酬は産出した組織効用の量に応じて傾斜配分される。
- ・エージェント間の模倣学習： 報酬の多いエージェントの行動はより模倣される。

また表 5.4 において、本実験の成果を関連する先行研究と比較して考察する。本研究の貢献は、集団の多様性をパラメーター化して操作し、多様性の変化によって改善と逸脱の双方が発生する過程を分析したことにある。先行研究は、多様性か一様性のいずれか、および改善と逸脱のいずれか、に焦点を当てている。一方本研究では、構成員の具体的な行動の記述は捨象しており、効用産出量の変化からの推測に止まる点が限界である。

表 5.3 多様性の変化に伴う階層ごとの効用産出量の傾向

	個人効用	組織効用	社会効用
多様性高	↘	↗	↗
多様性低	↗	↘	↘

表 5.4 応用実験 I と先行研究との関係

	集団の性質		構成員の行動	行動の結果	
	多様性	一様性		改善	逸脱
組織体逸脱 (樋口, 2009)		凝集性に焦点を当てる	不祥事の具体的な行動を記述		逸脱(集団愚考など)のみに焦点を当てる
ダイバーシティ・マネジメント (Page,2007; 谷口, 2005)	多様性(観点, 解釈, 予測)に焦点を当てる		構成員の多様性が優位性を生む現象を記述	改善(グローバルピークの探索)のみに焦点を当てる	
サイバーカスケード (Sunstein, 2001)		連帯, 共感に焦点を当てる	コミュニティメンバーが特定の信念へ傾倒する行動を記述		逸脱(カスケード, 集団分極化)のみに焦点を当てる
本研究: 応用 I	エージェント間の個人効用関数の相違によって多様と一様の双方を扱う		効用産出量の変化から類推	逸脱と改善を一体的に扱う	

5.4 応用Ⅱ：インフォーマルネットワークと集団の多様性の関係

5.4.1 実験の背景と目的

応用実験Ⅱでは、インフォーマルネットワークの量をパラメーターとして漸次変化させることに加え、集団の多様性も変化させる。ネットワークと多様性の条件の組合せに応じて、各階層の効用産出量に変化する過程を分析し、逸脱と改善を促進する要因を見出すことが、本実験の目的である。

インフォーマルなマネジメントの有効性については、古くから多くの指摘が行われている。Merton(1949)は公的な社会構造がもつ機能的欠陥を補完するために非公式な構造が必要であると主張した。近年では、組織運営においてフォーマルな組織の分業や効率性を維持しつつ同僚同士のインフォーマルな交流を活用することの有効性が指摘されている(Katzenba and Khan, 2010)。また、組織内の知識共有や協創を促進する手段としてインフォーマルコミュニケーションへの注目が集まっている(西本, 2006; Kraut, Fish, Root and Chalfonte, 1990)。これらの先行研究は、インフォーマルなマネジメントやコミュニケーションのメリットを具体的に記述するが、そのメリットがどのような条件において発揮されるのか、および条件によってはデメリットを生じさせる可能性は無いのか、については十分考察されていない。逆に、組織体逸脱の先行研究の中には、組織の上層への正式な報告というフォーマルなコミュニケーションが機能せず、現場レベルでのインフォーマルなコミュニケーションと判断に終始したことが工事ミスの隠ぺいという逸脱行為を生んだとの指摘がある(本間, 2007)。この研究の中では、インフォーマルコミュニケーションのデメリットが強調されている。一方、Kazanjian and Drazin(1990)や Quinn and Cameron(1983)らの組織ライフサイクル理論の領域では、企業の成長ステージによってはコミュニケーションや意思決定等の組織マネジメントの公式化が有効であり、インフォーマルなマネジメントが常に組織の成長に寄与するとは限らないことが指摘されてきた。組織の成長ステージによってインフォーマルなマネジメントの効果が異なるという指摘は重要である。

以上より、本実験ではインフォーマルコミュニケーションが条件によってはメリットとデメリットの双方を生む可能性があるとの仮定のもとに、応用実験Ⅰで分析した集団の多様性と、組織ライフサイクルから組織成長段階という概念を実験条件として取り入れ、インフォーマルネットワーク量の条件と組合せたシミュレーションを行う。

5.4.2 実験条件

応用実験Ⅱは実験ⅠとⅡとに分けて実施する。実験Ⅰの条件を表5.5に、実験Ⅱの条件を図5.6に示す。

実験Ⅰでは、インフォーマルネットワークの量を友人数によって操作する。友人数0は、上下階層間のフォーマルネットワークのみ存在する状態であり、700はエージェント間のネットワークの95%がインフォーマルネットワークである状態である。友人数は0から700

まで漸次変化させながら各階層の効用の産出量を観察する。また合わせて、集団の多様性が 0%と 100%の場合を条件に加えてシミュレーションを行う。尚、フォーマルおよびインフォーマルともに双方向の模倣学習が行われる。他の条件については、効用関数間の相反 0.4、報酬配分の傾斜次数は 2 に固定した。

実験 2 では、図 5.6 に示した組織成長段階のシナリオをにもとづき、エージェント数とエージェントの多様性の段階的に変化させる。組織ライフサイクル理論を参考に 4 つの成長ステージに沿って、エージェント数を 5 人から 50 人まで順次増やし、集団の多様性を 0%から 70%まで上昇させる。この成長段階を通じて、インフォーマルネットワーク率を 0%に設定した場合（条件 A）と、80%に設定した場合（条件 B）とで、個人効用と組織効用の産出量を比較する。インフォーマルネットワーク率が 0%とは上下間のフォーマルネットワークのみ存在する状態であり、80%とは約エージェント間のネットワークの 8 割がインフォーマルネットワークで結ばれている状態である。尚、実験 2 ではフォーマルなコミュニケーションとインフォーマルなコミュニケーションの差異を強くするために、フォーマルネットワークを通じた模倣学習は上意下達型とする。エージェントは、80%の確率でフォーマルネットワークでつながったエージェントの行動を模倣し、20%の確率でインフォーマルネットワークでつながったエージェントの行動を模倣する。

表 5.5 応用実験 II-1 の条件

実験条件	応用実験 II の設定	
ランドスケープ(効用関数の相反率)	0.4	固定
組織規模(エージェント数)	39	固定
組織ヒエラルキー	3階層	固定
多様性	0% , 100%	変動
報酬制度の傾斜次数	2	固定
インフォーマルネットワーク(友人数)	0 - 700	変動

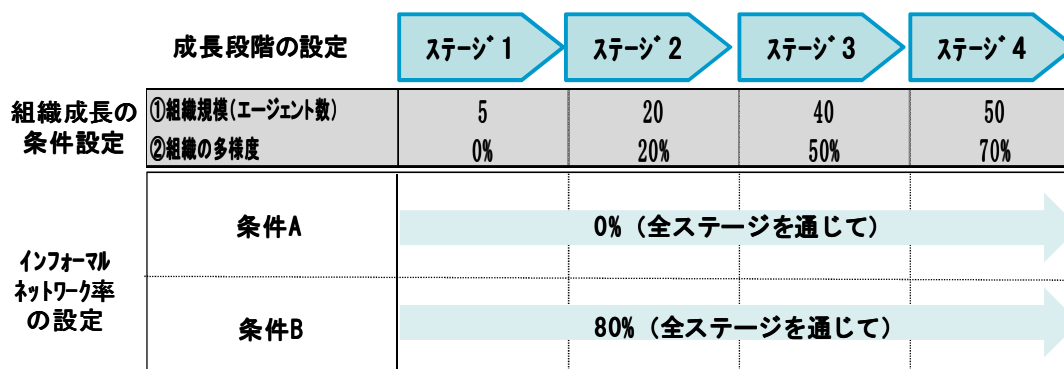


図 5.6 応用実験 II-1 の条件

5.4.3 実験結果 1：組織効用と社会効用量の推移

友人間のインフォーマルネットワーク数を変化させ、多様性の高い組織および一様な組織における効用産出量の推移を分析した。インフォーマルネットワークの量は友人数で表している。最小値 0 は友人間の繋がりが無く上下間の繋がりのみ存在する状態であり、最大値 700 はエージェント間のネットワークの 95%がインフォーマルネットワークである状態である。

図 5.7 に組織効用産出量，図 5.8 に社会効用産出量の推移を示した。インフォーマルネットワークが増加し友人間の繋がりが増えると，組織効用量は組織の多様性に関わらず増加する。一方，社会効用量は一様な組織において減少する傾向にあり，一様な組織ではインフォーマルネットワークの増加が，逸脱の促進要因として作用することを示している。

一様な集団で多様性の高い集団よりも，インフォーマルネットワークの増加によって社会効用産出量が低下する傾向にある要因の 1 つは，図 5.7 に示したように，インフォーマルネットワークの増加によって，エージェント間の模倣学習が促進され，組織効用の産出量が増加して報酬が増えていることである。組織効用量は，エージェントがより多くの報酬を受け取っている友人の行動を模倣しようとする性質により，模倣の促進によって漸増する。組織効用量の増加に伴う報酬の増加によってエージェントは社会効用量を増加させなくても満足度を維持することができる。

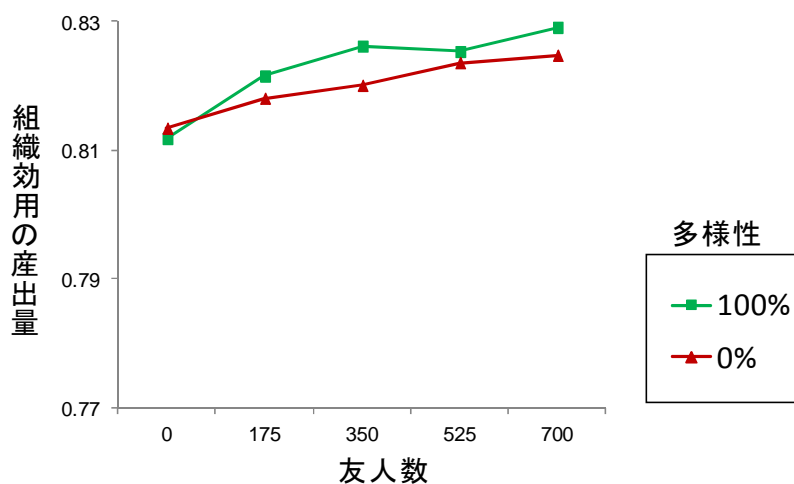


図 5.7 友人数の変化に伴う組織効用産出量の推移

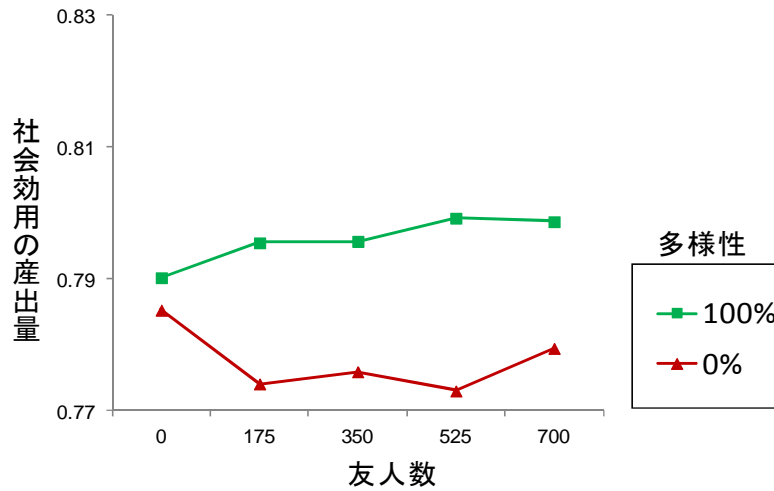


図 5.8 友人数の変化に伴う社会効用産出量の推移

5.4.4 実験結果 1：個人効用量と満足度の推移

次に、5.4.3 の結果を個人効用の産出量に関連付けて分析する。図 5.9 において、一様な集団の方が多様な集団よりも個人効用産出量が高く維持されている。個人効用関数が同一のエージェントで構成される一様な集団の方が個人効用のピークを探索しやすい。これにより、エージェントは社会効用量が減少しても、もともと高い個人効用量と、組織効用量の増加による報酬の増大によって満足度を維持できる（図 5.10 を参照）。一方、個人効用関数が異なるエージェントが構成する多様性の高い集団では、エージェント間の模倣によって個人効用のピークを探索し難いため、社会効用の産出量を増加させて、満足度を維持する。以上の傾向は、同じ価値観を持った構成員間のインフォーマルコミュニケーションが増加すると、価値観の共有による満足が優先され、社会への影響が軽視されることを示している。

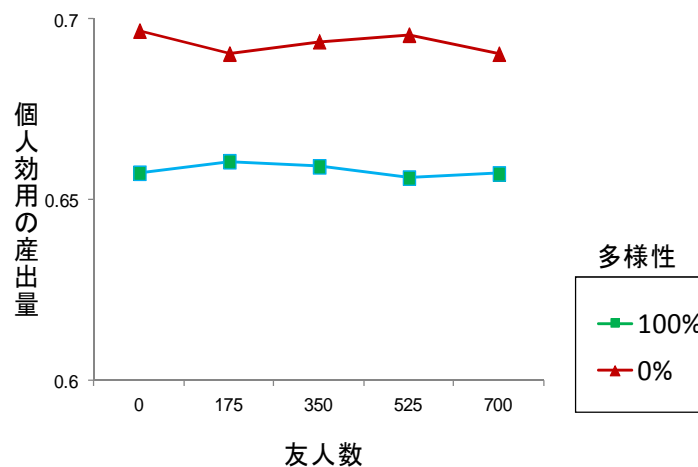


図 5.9 友人数の変化に伴う個人効用産出量の推移

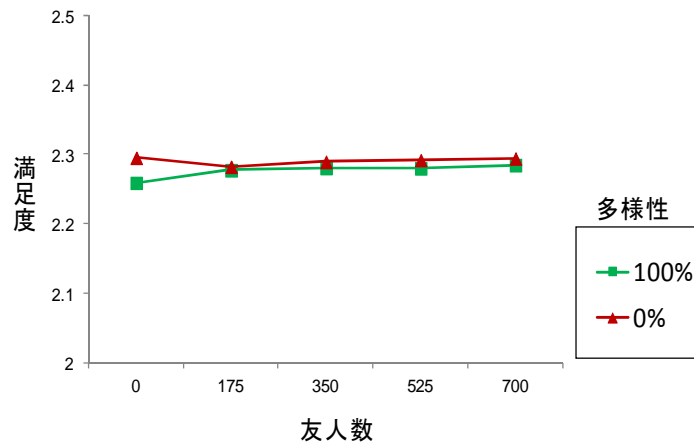


図 5.10 友人数の変化に伴う満足度の推移

5.4.5 実験結果 2

条件 A (インフォーマルネットワーク率 0%) の実験結果を図 5.11 に、条件 B (インフォーマルネットワーク率 80%) の実験結果を図 5.12 に示す。条件 B の組織効用の産出量が、条件 A よりも低いという結果になった。個人効用の産出量は、ステージ 2 までは条件 B の方が多いが、ステージ 3 と 4 では条件 A の方が高いという結果になった。

この結果を分析する。まずエージェント数が増加して多様性が増す組織成長プロセスを通じて、インフォーマルコミュニケーションが活発だと組織効用の量が低下するのは、エージェントが、常に新しく集団に参入する多様なエージェントとの相互学習を行うため、自身より多くの報酬を得ているエージェントかどうか判断できずに、組織効用関数のピークの集団探索力が弱まるためだと推察する。むしろ、限られた数のエージェントが上意下達型のコミュニケーションを取る方が、組織効用を多く産出し高い報酬を得ているエージェントの行動が模倣され易くなる。

次に、個人効用の産出量が成長ステージを追うごとに低下する傾向から、ステージ 2 までの多様性が比較的低い状態では、個人効用関数が同一の友人を見出し易いが、多様性が上がるにつれて見出し難くなり、個人効用関数のピークの集団探索力が弱まると推察する。このため、個人効用の産出量がステージ 2 まで増加しステージ 3 以降に減少する傾向は、インフォーマルネットワーク率が高い状態 (条件 B) においてより顕著となる。

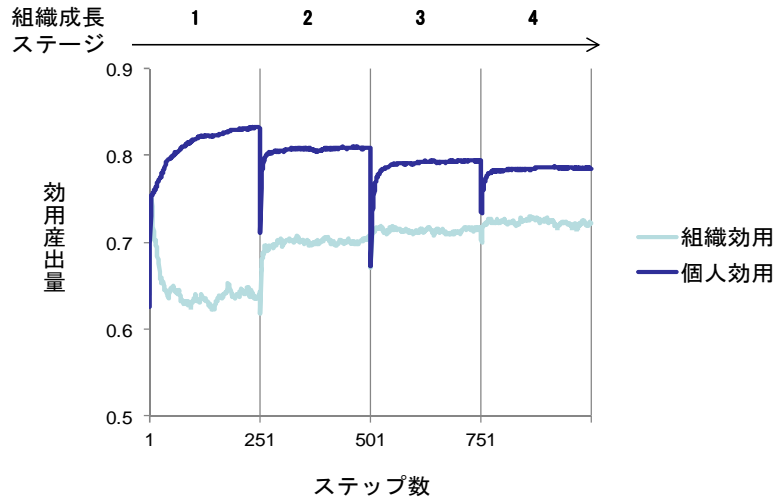


図 5.11 組織成長に伴う個人および組織効用産出量の変化
(条件 A : インフォーマルネットワーク率 0%)

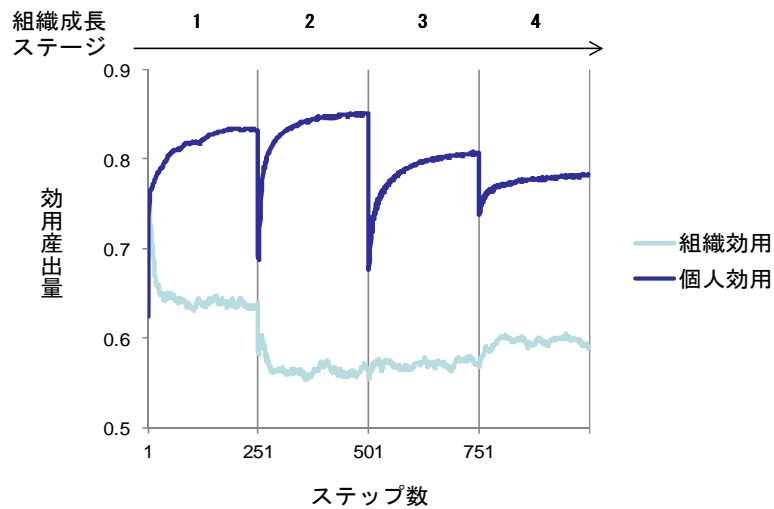


図 5.12 組織成長に伴う個人および組織効用産出量の変化
(条件 B : インフォーマルネットワーク率 80%)

5.4.6 考察

本実験では、インフォーマルコミュニケーションの増加が、多様性の高い集団では改善を促進するが、一様な集団では逸脱の要因となることを示した。この結果を階層間の効用産出量の視点から表 5.6 にまとめる。本実験ではさらに、多様な個人効用関数を持ち、報酬の実績が無い新規参入エージェントが増加する中で、インフォーマルコミュニケーションを活発化させると、模倣学習による個人および組織効用量の増加が困難であり、結果的に個人探索が増加して効用ランドスケープのピークに対する集団探索が弱まることも示した。

但し、これらの結果は、次の条件の下で得られたものである。

- ・エージェント間の模倣学習： 報酬の多いエージェントの行動はより模倣される。この模倣学習がインフォーマルネットワークの増加に伴って促進される。
- ・コミュニケーション方法： 組織成長段階を取り入れた実験では、エージェントは友人より上司の行動を高い確率で模倣（上意下達型コミュニケーション）。

表 5.7 において、本実験の成果を関連する先行研究と比較しながら考察する。インフォーマルコミュニケーションによる改善と逸脱の連続的な変化の過程を、階層間の効用産出量の観点から、集団の多様性に結び付けて分析したことが本研究の貢献である。先行研究の多くは、インフォーマルコミュニケーションによる改善か逸脱のいずれか一方に焦点を当てているためである。但し、集団の構成員の具体的なインフォーマルコミュニケーションの行動は捨象しており、効用産出量の変化からの推測に止まる点が本研究の限界である。

表 5.6 インフォーマルネットワーク増加に伴う階層ごとの効用産出量の傾向

インフォーマルネットワークを設定する集団の性質	個人効用	組織効用	社会効用
多様性が高い集団	→(低)	↗	↗
一様な集団	→(高)	↗	↘

表 5.7 応用実験Ⅱと先行研究との関係

	集団の性質とインフォーマルネットワークの効果の関係	具体的なコミュニケーション行動	行動の結果	
			改善	逸脱
インフォーマルコミュニケーション (西本, 2006; Kraut, 1990)	扱わない	具体的なインフォーマルコミュニケーション行動を記述	インフォーマルコミュニケーションが協業や相互学習を促進する側面に焦点を当てる	
組織ライフサイクル (Kazanjian&Drazin, 1990)	扱わない	具体的なコミュニケーション行動は記述しない	インフォーマルコミュニケーションの効用は組織の成長段階によることを指摘	
組織体逸脱 (本間, 2007)	扱う (上層部と現場との認識のギャップ)	インフォーマルコミュニケーションに基づく逸脱行為を記述		非公式なコミュニケーションによるミスの隠ぺいに焦点を当てる
本研究: 応用Ⅱ	扱う (集団の多様性)	効用産出量の変化から類推	階層間の効用産出量の関係から、インフォーマルコミュニケーションの効用と不効用を一体的に分析	

5.5 応用Ⅲ：報酬制度の変化に伴うエージェント行動の変化

5.5.1 実験の背景と目的

応用実験Ⅲでは、組織効用の産出量に応じた報酬配分の傾斜を実験のパラメーターとして漸次変化させる。そして、傾斜の変化が各効用産出量の変化に与える影響とその要因を見出すことが、本実験の目的である。

報酬制度が組織構成員の行動に与える影響を分析した先行研究としては、多くの企業で導入された成果主義型の評価報酬制度を批判したもの、および報酬制度と組織体逸脱との関係を分析したものが挙げられる。

高橋(2010)は、金銭的報酬とパフォーマンスが連動すると、従業員は一定の基準をクリアする働き方になりベストを尽くさなくなると指摘した。樋口(2006)も同様の主張を行っている。また城(2004)も、成果主義型の制度への移行によって従業員が設定する目標の水準が下がり、「仕事をやり遂げるといふ」意識から「単に目標を達成する」という意識に変化したことを実例に基づいて述べている。さらに、高評価者と低評価者への2極分化が発生し、後者は評価の上昇を諦めて残業を繰り返したため、組織全体の生産性が低下したことも指摘されている。樋口(2009)は、証券会社で起きた不祥事の事例において、従業員の賞与が会社の経常利益に直結していたことが違法行為に手を出した一因であると指摘している。報酬への意識が強くなったために、違法行為が露見する恐れを軽視するようになり、コンプライアンス意識が欠如していったとの分析が示されている。

5.5.2 実験条件

実験の条件を表 5.8 に示す。報酬配分の傾斜は、傾斜度合いを示す次数の操作によって変化させる。次数 1 は全エージェントに報酬が均等に配分される状態であり、以降次数が 36 まで増加するに従い組織効用を多く産出するエージェントがより多い報酬を得るようになる。尚、報酬配分の傾斜次数以外の条件は、効用関数間の相反 0.4、多様度は 100%、友人数は 350 に固定した。

表 5.8 応用実験Ⅲの条件

実験条件	応用実験Ⅱの設定	
ランドスケープ(効用関数の相反率)	0.4	固定
組織規模(エージェント数)	39	固定
組織ヒエラルキー	3階層	固定
多様性	100%	固定
報酬制度の傾斜次数	1 - 36	変動
インフォーマルネットワーク(友人数)	350	固定

5.5.3 実験結果

図 5.13 が示すように、報酬配分が均等配分の場合(次数=1 の場合)は、組織効用を産出するインセンティブが存在せず、組織効用より社会効用の産出量の方がやや大きい。報酬配分の傾斜を強めると、組織効用量は一旦顕著に増加して社会効用量を上回るが、その後減少し続ける。また、社会効用量は当初は若干増加するが、その後次第に減少する。

まず、組織効用量が一旦増加後に減少する原因としては、組織効用の産出量が相対的に評価され報酬に反映されることが挙げられる。報酬配分の傾斜強化によって、エージェントは他のエージェントよりも少しでも多くの組織効用を産出し順位を上げれば、より多くの報酬を得ることができるようになる。このため、エージェントは組織効用産出量の増加を十分に追求せずとも報酬が増加し、結果的に満足度の維持が可能となる。よって、一定以上に報酬配分の傾斜が強まると、組織効用のグローバルピークの探索が抑制される。

次に、社会効用産出量の傾向について考察する。エージェントは、自身より報酬を多く得ている他エージェントの行動を模倣するが、その結果満足度が低下した場合は、模倣前の行動状態に戻る。報酬配分の傾斜が一定以上に強いと、エージェントはより報酬の高いエージェントの行動の模倣によって社会効用量が減少しても、報酬量の増加がそれを補完して満足度を維持することができる。その結果、社会効用産出量が減少する。逆に傾斜が弱い場合は模倣によって報酬が十分に増加しないため、社会効用量を減少させる模倣は満足度の低下を招き、不成立になり易い。その結果、社会効用量の減少が抑制される。報酬の傾斜度合いの強化に伴う満足度の推移を図 5.14 に示す。図 5.13 において、報酬配分の傾斜強化により組織効用および社会効用産出量が減少しても、満足度は維持されている。

さらに、個人効用産出量との関係を分析する。図 5.15 に示すように、報酬配分の傾斜度合いの強化に伴い、個人効用の産出量は一旦低下するが、その後増加していく。それぞれの結果を総合すると、エージェントは報酬の傾斜が大きくなると、組織効用の増加を追求せず、個人効用量を増加させることによって満足度を維持すると推測できる。

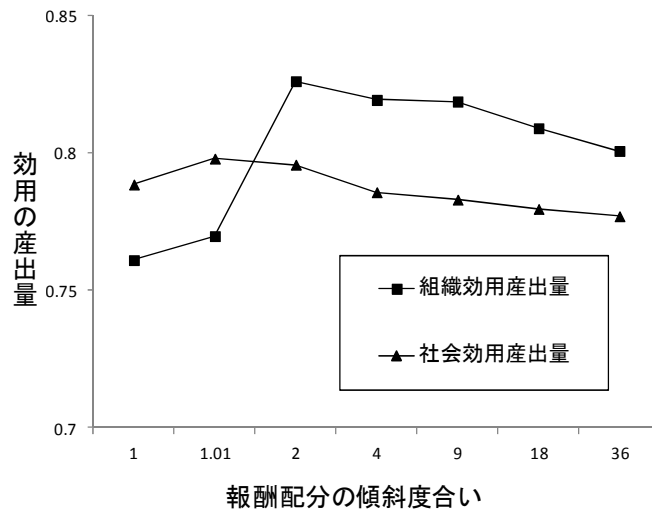


図 5.13 報酬の傾斜強化に伴う組織および社会効用産出量の推移

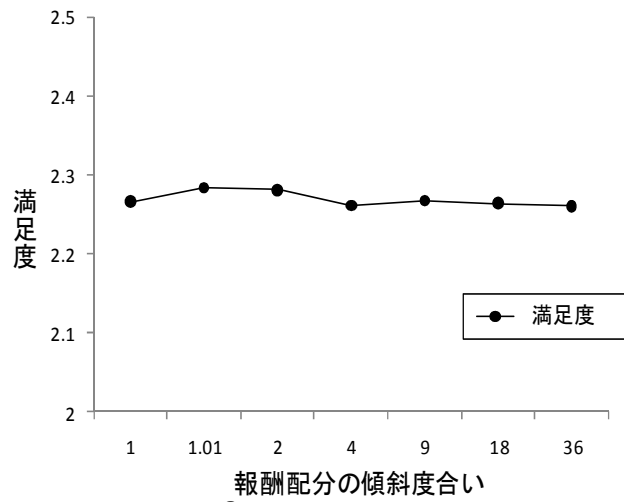


図 5.14 報酬の傾斜強化に伴う満足度の推移

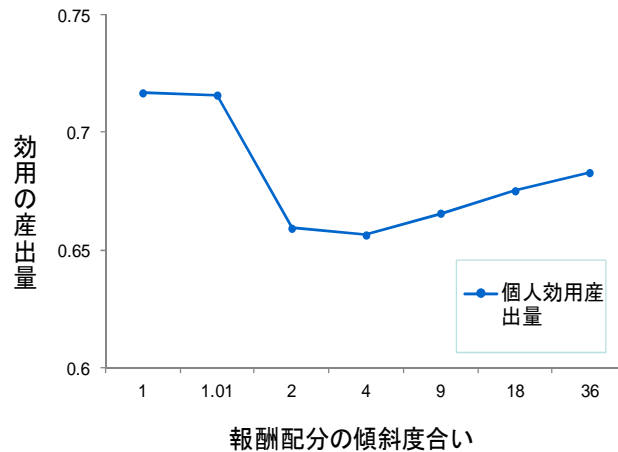


図 5.15 報酬の傾斜強化に伴う個人効用産出量の推移

5.5.4 考察

本実験では、報酬配分の傾斜の強さによって、改善と逸脱の双方が生ずることを示した。また、一定以上に報酬の傾斜を高めると逸脱が促進される要因として、エージェントは報酬の傾斜が大きくなると、組織効用の増加を追求せず、個人効用量を増加させることによって満足度を維持することを挙げた。この結果を、階層間の効用産出量の視点から表 5.9 にまとめる。但し、これらの結果は、次の条件の下で得られたものである。

- ・満足度の構造： エージェントの満足度は、個人効用の産出量、組織から得る報酬、社会効用量産出量の総和によって決定される。
- ・エージェント間の模倣学習： 報酬の多いエージェントの行動はより模倣される。
- ・報酬制度： 即時報酬であり、報酬の遅れを含めていない。

表 5.10 において、本実験の成果を関連する先行研究と比較して考察する。報酬制度や組織体逸脱に関する先行研究は、成果主義型の報酬制度が組織効用量や社会効用量の低下をもたらす逸脱を促進する側面に焦点を当てていた。本研究の貢献は、改善と逸脱の両側面を対象とし、報酬配分の傾斜度合いの強化に伴う効用産出量の連続的な変化から、改善から逸脱に転ずる過程を示したことである。

一方で本研究は、成果主義型の報酬制度の導入による組織構成員の具体的な行動の変化や違法行為の記述は捨象している。効用産出量の変化に基づく構成員の行動の推測に止まる点が先行研究と比較した本研究の限界である。また出口(2000)は、報酬に時間遅れがある場合にはエージェントは自身の成果に対する客観的な評価ができないことを指摘したが、本研究では報酬の遅れがエージェント行動に与える影響の分析を対象としていないことも限界である。但し、エージェントは自身の産出した組織効用量から得られる報酬を事前に把握できない条件を設定し、ある程度の不確実性を実現した。

表 5.9 報酬配分の傾斜の変化に伴う階層ごとの効用産出量の傾向

報酬配分の傾斜	個人効用	組織効用	社会効用
一定レベル	↘	↗	↗
一定レベル以上	↗	↘	↘

表 5.10 応用実験Ⅲと先行研究との関係

	評価報酬制度に基づく 組織構成員の行動	行動の結果	
		改善	逸脱
報酬制度・評価制度 (城, 2004; 高橋, 2010)	制度の変更に伴う構成員の行動の変化を記述		成果主義型の制度による仕事の質の低下に焦点を当てる
組織体逸脱 (樋口, 2009)	具体的な違法行為と報酬制度との関係を記述		成果主義型の報酬制度と違法行為との関係に焦点を当てる
本研究: 応用Ⅲ	効用産出量の変化から類推	成果に応じた報酬の傾斜の変化による効用の連続的な変化の過程を示す	

第6章 結論と課題

6.1 はじめに

本章では、本研究の貢献と今後の課題を論じる。本研究の主要な貢献は次の2点である。まず、1)階層的なランドスケープと集団の性質を設定し、組織構成員の学習によって、改善と逸脱の双方が生ずる過程と要因を分析するエージェント・ベース・モデルを構築したことである。次に、2)提案モデルに基づくケース記述によって、改善と逸脱の共通要素と差異要素を分離し、構成員の学習と行動変化の過程を説明したことである。

6.2 本研究の結論と貢献

本研究の結論を図6.1と6.2に示す。本研究は、組織活動における逸脱と改善の現象を取り扱うことが可能な統一的なモデルを提案し(図6.1)、その有効性を、計算機実験とケースとの接地によって確認したものである(図6.2)。各結論について、以下に説明する。

図6.1に示したように、本研究は、改善と逸脱のような組織活動における表裏一体的な現象が創発する過程と要因を、階層間の効用の産出量の関係の視点から分析するモデルを構築することを目的としたものである。この目的に照らした本研究の貢献は、階層的なランドスケープと集団の性質の変化のもとで、組織構成員の相互作用と学習を通じて改善と逸脱の双方が生ずる過程の分析を可能にするエージェント・ベース・モデルを構築したことである。

結論1

組織活動において表裏一体的な現象が創発する過程と要因を、階層間の効用の産出量の関係の視点から分析するモデルを構築。

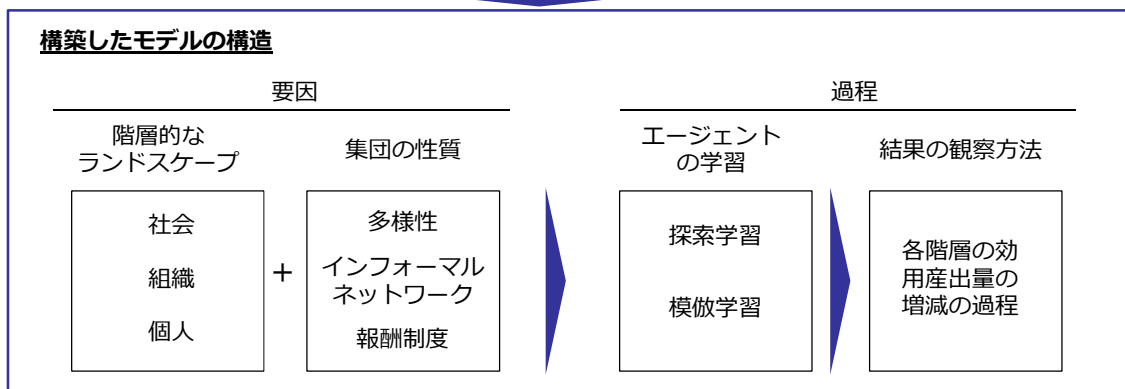


図6.1 本研究の結論の要約1

さらに、図 6.2 に示したように、計算機実験（シミュレーション）によって、条件を操作しながら改善と逸脱が生ずる過程を分析するによりモデルの有効性を確認した。また、ケースとの接地を通じて、構成員の行動変化の過程が分析できること、階層的なランドスケープ間の関係が構成員の行動に与える影響が分析できること、および改善と逸脱に共通する要素と異なる要素を識別できるという、モデルの有効性を明らかにしたことも本研究の貢献である。一方で、本モデルの説明範囲には具体性の面で限界があることが明らかになった。またモデルの構造についても、報酬遅れが取り込めておらず即時報酬である、ランドスケープを外生的に設定しておりエージェントの行動に伴う創発的な変化は対象外としている、という限界が明らかになった。

結論2

構築したモデルの有効性を、計算機実験とケースとの接地を通じて確認。

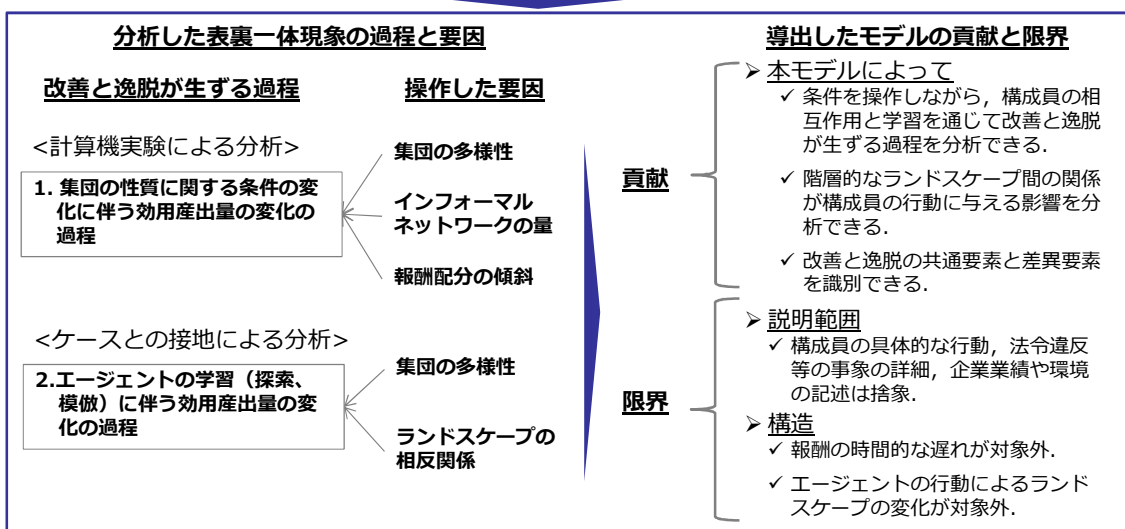


図 6.2 本研究の結論の要約 2

以下に、本論文の各章の結論をまとめる。

第 1 章では、研究の背景と目的を説明し、本研究の出発点となる「改善と逸脱は表裏一体であり共通の作動原理を持つ」という仮説を示した。

第 2 章では、先行研究への問題意識をもとに本研究の位置づけを整理し、モデルの要件と本研究が対象とする問題を定義した。根本となる問題は「不完全情報のもとでのエージェントの行動によって改善と逸脱が創発する過程の分析」である。これを、1) 集団の多様性の変化が効用の産出に与える影響、2) インフォーマルネットワークの量が効用の産出に与える影響、3) 報酬による動機づけが効用の産出に与える影響、の 3 つの問題に分解した。

第3章では、2章において設定した問題を分析するためのモデルを構築し、エージェント・ベース・モデルを記述するスタンダードである ODD(Overview, Design concepts, and Details) に沿って説明した。

第4章では、設定したモデルの振る舞いの予想と生成結果に対する評価基準に基づき、実際にシミュレーションを行った結果と、モデルの振る舞いや生成結果の予想とを比較した。これにより、モデルに設計の意図が反映されていること、および設定した評価基準に照らして妥当な結果が生成されていることを確認した。本章では、合わせてケースとの接地を行い、組織構成員の探索学習と模倣学習によって改善と逸脱が生ずる過程を再現した。これにより、モデルの説明範囲の限界を確認するとともに、モデルの要素に基づくケース記述のメリットを実際の事例に基づくケースと比較して明確化した。

第5章では、2章で分解した問題に沿ってエージェント・ベース・シミュレーションによる3種類の応用実験を行い、実験結果を関連する先行研究や事例と比較し考察した。第4章のケースとの接地で、組織構成員の学習行動に伴う改善と逸脱の過程を分析したことに対し、本章では集団の性質に関する条件変化に伴う改善と逸脱の過程を分析した。

応用実験Ⅰ：多様性の変化により改善と逸脱の双方が生ずることを示した。また、多様性の向上が改善の促進要因になること、および一様な組織は、改善状態と逸脱状態への分散が高いことを示した。

応用実験Ⅱ：インフォーマルコミュニケーションの増加は、多様性の高い集団では改善を促進するが、一様な集団では逸脱を促進することを示した。

応用実験Ⅲ：報酬配分の傾斜を一定水準まで高めるに伴い改善の傾向が生ずるが、一定水準を越えると逸脱の傾向が生ずることを示した。

但し第5章の結論は、あくまで本モデルの構造や条件のもとで、階層ごとの効用産出量の変化を観察することによって得たものである。

以上にまとめた各章の個別の結論の論理的な関係を図6.3に示した。本研究は、作業仮説のもとに問題とその分析のためのモデルを構築し、シミュレーションによるモデルの有効性と適用範囲拡大の余地の考察と、ケースとの接地を通じたモデルの説明範囲の限界と利点の明確化を経た上で、研究全体の貢献を整理するという、論理的な構成を取っている。

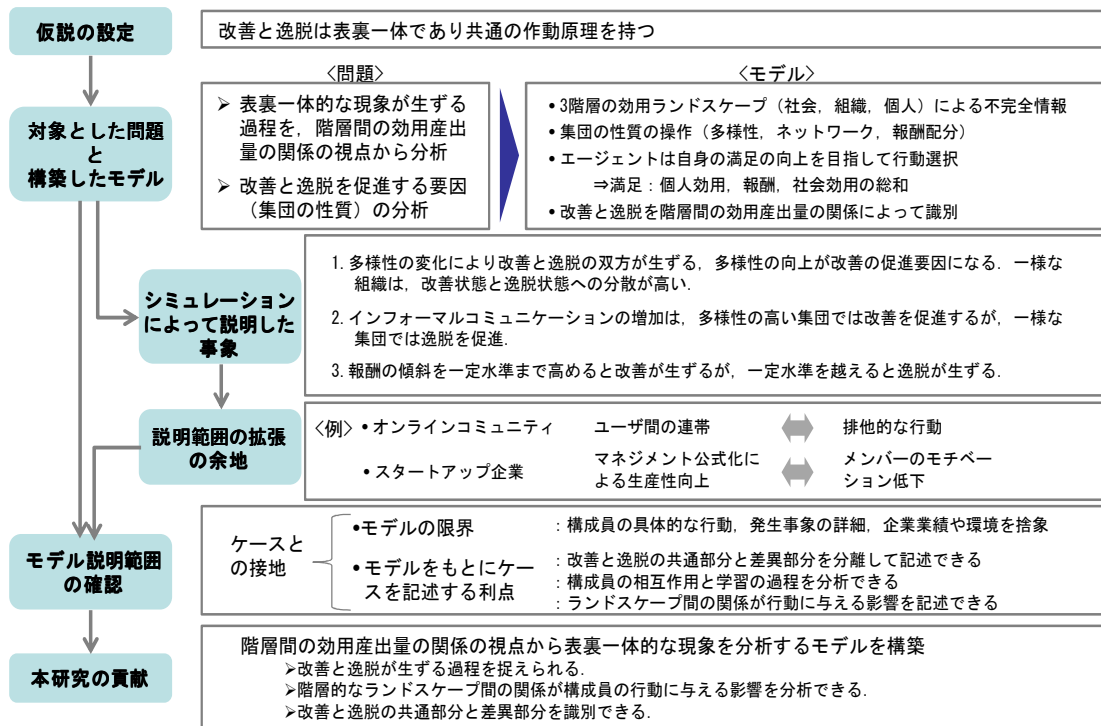


図 6.3 本論文の結論の論理構成

6.3 今後の課題

今後の課題として、以下の4つが挙げられる。課題1)は分析対象とする問題の拡大、課題2)と3)はモデルの拡張、課題4)は本モデルを活用した研究スコープの拡大についての課題である。

- 1) 今回対象とした3つの問題を越えた実験と分析の実施
- 2) 報酬の時間的な遅れによるエージェント行動の変化の分析
- 3) エージェントの行動によるランドスケープの変化の分析
- 4) モデルに基づいたケースの記述による将来的なシナリオの分析

課題1)について、本論文で定義し分解した問題は、主に経営学および組織論の分野の先行研究に基づいているが、表裏一体的な現象は、例えば経済学でも様々な議論がなされている。それらの議論の内容を取り入れ、より多様な条件下で、エージェントの探索学習や模倣学習による効用産出量の変化の過程を分析することには意義があると考えられる。今後は今回設定した問題以外にも分析の対象を拡大する。

課題2)は、本モデルではエージェントが産出する組織効用量をもとに報酬を即時に与えており、報酬の遅れを考慮していないというモデルの限界によるものである。本論文では、

産出した組織効用が報酬制度を通じて配分され、且つ制度を実験パラメーターとして変化させることによって不確実性を実現した。しかし、行動を実行した直後の報酬だけで成果を判断できないという、より不確実性の高い状況下でのシミュレーションが必要であり、今後モデルに取り入れる。

課題 3)について、本論文では社会、組織、および個人の各効用関数をランドスケープとして外生的に設定している。しかし現実においては、例えば企業が提供する製品・サービスや発信する情報によって消費者のニーズが変化するなど、エージェントの行動によってランドスケープが変化することも多い。また、逸脱行為が常態化すると従来の規範に対する逸脱が新たな規範になるとの指摘もある(出口, 2004; Reason, 1997)。今後は、エージェントの行動とランドスケープが相互に影響し合って変化するモデルを構築し、シミュレーションと分析を行う。

課題 4)は、モデルの適用範囲と限界を整理するために仮想のケースとモデルの要素とを接地させた結果、構成員の学習と行動変化の過程が分析できた、および表裏一体的な現象である改善と逸脱に共通する部分と差異部分とを分離し識別できたという、研究の成果に基づくものである。従来のケースは、過去に起きた事象をもとに記述されてきたが、モデルをベースにケースを記述することで、企業経営や組織マネジメントにおける将来のシナリオを考察することにつながる可能性がある。畑村(2000)は、起こり得る失敗を想定しながら頭の中の経験として習得する仮想失敗体験の必要性を指摘した。構築したモデルに基づいて、エージェント・ベース・シミュレーションとケース記述による考察とを組み合わせることで、実効性の高い仮想失敗体験が可能となり、不確実性の高い環境における想定範囲を拡大することが期待される。

謝辞

本論文の作成にあたり，多くの方々からご指導とご支援を頂きましたことに，この場をお借りして厚く御礼申し上げます。

東京工業大学大学院の寺野隆雄先生には，指導教官として研究テーマの選定から論文作成の細部に至るまで多岐に渡るご指導を頂戴し，研究を導いて頂きました。吉川厚先生には，大学院への入学前から研究の土台となる幅広い知識を授けて頂くと共に，研究方法の基本をお教え頂きました。出口弘先生，新田克己先生，山本学先生には，論文審査を通じて博士論文としての内容を固める上で大変貴重なご助言を頂戴しました。

特別研究員の國上真章さんには，投稿論文の共同執筆者として，研究や論文執筆が行き詰る度に的確なアドバイスを頂き，打開の道筋を見出す手助けをして頂きました。そして共同研究者である東京工業大学大学院の高橋聡さんには，モデル構築や実験を全面的に支えて頂いただけでなく，様々な情報や知識の提供を通じて研究活動をサポートし続けて頂きました。

また入学以来，寺野研究室のメンバーの方々には，様々な場面で温かいご支援とご助言を頂戴しました。山田隆志先生および山口ひとみさんには，研究活動の要所で親身なご支援を頂きました。研究に対する真摯な姿勢と楽しい語らいが同居する寺野研究室で学べたからこそ，博士課程を完遂することができたと実感します。改めて深く感謝申し上げます。

最後に，常に励ましを与えてくれた両親，長きに渡って筆者を支援し続けてくれた妻，そして心の支えとなってくれた二人の娘に感謝致します。

2012年8月末日

※各氏の所属，職責は2012年8月時点の表記です。

参考文献

- Axelrod, R.(1997) *The Complexity of cooperation*, *Princeton Univ. Press*, (寺野隆雄監訳(2003)『対立と協調の科学—エージェント・ベース・モデルによる複雑系の解明』, ダイヤモンド社.)
- Axelrod, R., Cohen, M. (1999) *Harnessing Complexity*, *Free Press*. (高木晴夫監訳, 寺野隆雄訳(2003)『複雑系組織論』, ダイヤモンド社.)
- Axtell, R. (2000) *Why Agents? On The Varied Motivations for Agent Computing in the Social Sciences*, *Center on Social and Economic Dynamics Working Paper*, No. 17.
- Bass N. (1992) *Emergence, Hierarchies, and Hyperstructures*, *Artificial Life III*, 515-537.
- Baucus, M. (1994) *Pressure, opportunity and predisposition: A multivariate model of corporate illegality*, *Journal of Management*, 20, 699-721.
- Bonabeau, E. (2002) *Agent-based modeling: Methods and techniques for simulating human systems*, *PNAS*, 99(3), 7280-7287.
- Grimm V., Berger U., Bastiansen F., et al. (2006) *A standard protocol for describing individual-based and agent-based models*, *ecological modeling* , 198, 115–126.
- Grimm V., Berger U., DeAngelis D., et al. (2010) *The ODD protocol: A review and first update*, *ecological modeling*, 221, 2760–2768.
- Hong L. and Page, S. (2004) *Groups of diverse problem solvers can outperform groups of high-ability problem solvers*, *PNAS*, 101(46), 16385-16389.
- Imai, M. (1986) *Kaizen: The Key To Japan's Competitive Success*,. *McGraw-Hill/Irwin*. (今井正明(1991)『カイゼン —日本企業が国際競争で成功した経営ノウハウ』, 講談社文庫.)
- Katzenbach J., & Khan Z. (2010) *Leading outside the lines: How to Mobilize the (in) Formal Organization Energize Your Team and Get Better Results*, *Booz & Company, Inc.* (ブーズ・アンド・カンパニー訳(2011)『インフォーマル組織力 —組織を動かすリーダーの条件』, 税務経理協会) .
- Kauffman S. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*, *Oxford University Press*.
- Kauffman S. (1995) *At Home in the Universe: The Search for Laws of Self-Organization and Complexity*, *Oxford University Press*. (米沢富美子訳 (2008)『自己組織化と進化の論理 —宇宙を貫く複雑系の法則』, ちくま学芸文庫.)
- Kazanjian, R., & Drazin, R. (1990) *A stage-contingent model of design and growth for technology-based new ventures*, *Journal of Business Venturing* 5, 137-150.
- Kijima K. (2001) *Generalized landscape theory: Agent-based approach to alliance formations in*

- civil aviation industry, *Journal of Systems Science and Complexity*, 14(2), 113-123.
- Kraut, R., Fish, R., Root, R., & Chalfonte, B. (1990) Informal communication in organizations: Form, function, and technology, In S. Oskamp & S. Spacapan (Eds.), *People's Reactions to Technology*, Sage Publications, 145-199.
- Merton R. (1949) Social Theory and Social Structure: Toward the Codification of Theory and Research, *The Free Press*. (森, 金沢, 森, 中島訳(1961)『社会理論と社会構造』, みすず書房.)
- National Research Council (2008) Behavioral Modeling and Simulation; From Individuals to Societies, *The National Academies Press*.
- Page, S. (2007) *The Difference*, Princeton University Press. (水谷淳訳(2009)『「多様な意見」はなぜ正しいのか ― 衆愚が集合知に変わるとき』, 日経 BP 社.)
- Quinn, R., & Cameron, K.(1983) Organizational life cycles and shifting criteria of effectiveness, *Management Science* 29, 33-51.
- Reason J. (1997) *Managing the Risks of Organizational Accidents*, Ashgate Publishing Limited. (塩見弘監訳, 高野研一・佐相邦英訳 (1999)『組織事故』, 日科技連) .
- Salaniè B.(1997) *The Economics of Contracts*, The MIT Press. (細江, 三浦, 堀訳(2000)『契約の経済学』, 勁草書房.)
- Sunstein, C. (2001) *Republic.com*, Princeton Univ. Dept. of Art. (石川幸憲訳(2003)『インターネットは民主主義の敵か』, 毎日新聞社.)
- Wenger, E., McDermott, R. (2002) *Cultivating Communities of Practice*, Harvard Business School Press. (野村恭彦監修, 櫻井祐子訳(2002)『コミュニティ・オブ・プラクティス ― ナレッジ社会の新たな知識形態の実践』, 翔泳社.)
- 伊藤秀史(2003)『契約の経済理論』, 有斐閣.
- 伊庭齊志(2002)『遺伝的アルゴリズム』, 医学出版.
- 上田泰 (2003)『組織行動研究の展開』, 白桃書房.
- 岡本浩一(2001)『無責任の構造 ― モラル・ハザードへの知的戦略』, PHP 新書.
- 岡本浩一, 今野裕之(2003)『リスクマネジメントの心理学』, 新曜社.
- 岡本浩一, 今野裕之(2006)『組織健全化のための社会心理学 ― 違反・事故・不祥事を防ぐ社会技術』, 新曜社.
- 株式会社赤福コンプライアンス諮問委員会 (2008)「報告書」, 株式会社赤福 Homepage (<http://www.akafuku.co.jp>).
- 株式会社不二家信頼回復対策会議 (2007)「品質管理の実態調査 ― 業務レベルにおける中間報告」, 株式会社不二家 Homepage (<http://www.fujiya-peko.co.jp>).

- 株式会社不二家信頼回復対策会議(2007) 「品質管理の実態調査 ―不二家における品質管理上の問題点と対策」, 株式会社不二家 Homepage (<http://www.fujiya-peko.co.jp>).
- 木嶋恭一(2004) 「ランドスケープ理論とその周辺」, 『オペレーションズ・リサーチ: 経営の科学』, 49(3), 137-142.
- 産経新聞取材班(2002) 『ブランドはなぜ墜ちたか ―雪印, そごう, 三菱自動車事件の深層』, 角川文庫.
- 城繁幸 (2004) 『内側から見た富士通 ―「成果主義」の崩壊』, 光文社.
- 高橋伸夫(2010) 『虚妄の成果主義 ―日本型年功制復活のススメ』, ちくま文庫.
- 谷口真美(2005) 『ダイバーシティ・マネジメント―多様性をいかす組織』, 白桃書房.
- 出口弘(2000) 『複雑系としての経済学 ―自律的エージェント集団の科学としての経済学を目指して』, 日科技連.
- 出口弘(2004) 「組織の失敗と評価のランドスケープ学習」, 『組織科学』, 38(2), 29-39.
- 出口弘, 木嶋恭一 編(2009) 『エージェントベースの社会システム科学宣言 ―地球社会のリベラルアーツめざして』, 勁草書房.
- 寺野隆雄(2004) 「エージェント・ベース・モデリングへの招待」, 『オペレーションズ・リサーチ』, 49(3), 131-136.
- 寺野隆雄(2006) 「エージェント・ベース・モデリングの技術と応用」, 『社会シミュレーションの技術動向と実務的課題』 計測自動制御学会システム工学部会, 経営情報学会合同シンポジウム資料, 1-8.
- 寺野隆雄(2007) 「学習と進化とシステム思考」, 『計測と制御』, 46(4), 274-279.
- 寺野隆雄(2010) 「なぜ社会システム分析にエージェント・ベース・モデリングが必要か」, 横幹, 4(2), 56-62.
- 西本一志 (2006) 「インフォーマル・コミュニケーションによる知識共創場の構築」, 『共創と複雑系シンポジウム予稿集』, 計測自動制御学会, 17-26.
- 畑村洋太郎(2000) 『失敗学のすすめ』, 講談社.
- 樋口晴彦(2006) 『組織行動の「まずい!!」学』, 祥伝社.
- 樋口晴彦(2009) 『不祥事は財産だ ―プラスに転じる組織行動の基本則』, 祥伝社.
- 宝月誠(1988) 「組織体逸脱の研究に向けて」, 『犯罪社会学研究』, 13, 12-18.
- 宝月誠(2004) 『逸脱とコントロールの社会学 ―社会病理学を超えて』, 有斐閣.
- 宝月誠(2006) 「食品の安全性と企業逸脱」, 『立命館産業社会論集』, 42(3), 1-23.

本間道子(2007)『組織性逸脱行為過程 ―社会心理学的視点から』, 多賀出版.

間嶋崇(2007)『組織不祥事 ―組織文化論による分析』, 文眞堂.

三浦麻子, 川浦康至(2008)「人はなぜ知識共有コミュニティに参加するのか ―質問行動と回答行動の分析」, 『社会心理学研究』, 23(3), 233-245.

宮田加久子(2005)『きずなをつなぐメディア ―ネット社会の社会関係資本』, NTT 出版.

業績目録

査読付原著論文

- 小林知己, 高橋聡, 國上真章, 吉川厚, 寺野隆雄 (2011) 「オンラインコミュニティにおける効用相反とそのハーネシングに関するエージェントシミュレーション」, 『情報社会学会誌』 5(3), 31-40.
- 小林知己, 高橋聡, 國上真章, 吉川厚, 寺野隆雄 (2011) 「組織逸脱と改善の分岐条件とそのハーネシングに関するエージェントシミュレーション」, 『電子情報通信学会論文誌』 94(11), 1825-1835.

国際学会発表

- Tomomi Kobayashi, Satoshi Takahashi, Masaaki Kunigami, Atsushi Yoshikawa, Takao Terano(2010) Unifying Organizational Deviation and KAIZEN Activities - An Agent Based Model. Ernst, A. & Kuhn, S. (Eds.). Proceedings of the 3rd World Congress on Social Simulation WCSS2010 (CD-ROM).
- Tomomi Kobayashi, Satoshi Takahashi, Masaaki Kunigami, Atsushi Yoshikawa, Takao Terano (2011) The Effect of Management Style Formalization with Growth of Organization - An Agent-Based Model. Proc. Knowledge Management in Organization 6-th Conference (KMO 2011). 20.
- Tomomi Kobayashi, Satoshi Takahashi, Masayuki Kunigami, Atsushi Yoshikawa, Takao Terano (2012) A Unified Agent-Based Model to Analyze Organizational Deviation and Kaizen Activities. in Francien Dechesne, Hiromitsu Hattori, Adriaan ter Mors, Jose Miguel Such, Danny Weyns, Frank Dignum (Eds.): Advanced Agent Technology -Proc. AAMAS 2011 Workshops-, LNAI 7068, Springer Verlag, pp. 384-395.

国内学会発表

- 小林知己, 高橋聡, 國上真章, 吉川厚, 寺野隆雄 (2010) 「組織逸脱と改善の分岐現象とそのハーネシングに関するエージェントシミュレーション」, Proc. JAWS2010 S-1, 2.
- 小林知己, 高橋聡, 國上真章, 吉川厚, 寺野隆雄 (2010) 「オンラインコミュニティにおける効用相反とそのハーネシングに関するエージェントシミュレーション」, 第3回知識共有コミュニティワークショップ論文集, pp.11-20.
- 小林知己, 高橋聡, 國上真章, 吉川厚, 寺野隆雄 (2011) 「組織逸脱と改善の分岐条件とハーネシング」, 『第4回 横幹連合コンファレンス』 (CD-ROM).

研究助成

- ・ 「ベンチャー企業の成長を促進する組織形成モデルの構築」, 科学技術融合振興財団, 平成 20 年度シミュレーション&ゲーミングの先進的独創的な手法の研究.

所属学会

- ・ 電子情報通信学会, 情報社会学会, 組織学会, 国際戦略経営研究学会, 日本ベンチャー学会.

付録

付録には、本研究モデルのソースコードの抜粋を掲載する。ソースコードは、ODD (Overview, Design concepts, and Details) プロトコルに基づき、モデルの内容を把握する上で重要性の高い部分を抜粋した。抜粋したソースコードと掲載ページを ODD との対応付けを含めて以下に整理する。

【Environment: ランドスケープの設定】 (Basic Principles に該当)	85
【NK Model Function: NK モデルの設定】 (Basic Principles に該当).....	87
【Agent: エージェントの定義】 (Collectives に該当)	89
【Hill Climb: 探索学習】 (Learning に該当)	91
【Evaluation Agent: 模倣対象の決定】 (Learning に該当).....	92
【Value: 満足度の判定】 (Objectives に該当).....	94

【付録 本研究モデルのソースコード抜粋】

Environment: ランドスケープの設定

```
package common;

import java.io.File;
import java.util.Random;
import common.IDistributor;
import common.IAgentCreator;
import common.IProblem;
import common.Value;
import common.Agent;
import common.ILogger;
import common.INetworkCreator;
import common.Parser;
import common.RandomGenerator;

public class Environment {

    private Agent[] agents;
    private int steps;
    private IAgentCreator agentCreator;
    private String[] property;
    private ILogger[] logger;
    private EvaluationAgent evaluationAgent;
    private IDistributor organizationalDistributor;
    private IDistributor socialDistributor;
    private int changeStart;
    private int[] formalLinkMap;
    private int sameNum;
    private int friendNum;
    private int allAgentNum;
    private IProblem organizationalProblem;
    private IProblem socialProblem;
    private IProblem personalProblems[];
    private double repulsionRate;

    public Environment(String[] property, File[]
    experimentDirectory) throws Exception {
        this.property = property;
        setRandomGenerator();
        changeStart = Parser.parseInteger(this.property, "
        Problem.Change.Start");
        sameNum = Parser.parseInteger(property,
        "Problem.Same.Num");
        friendNum =
        Parser.parseInteger(property, "Friend.Num");
        repulsionRate =
        Parser.parseDouble(property, "Repulsion.Rate");
        setFormalLinkMap();
        setUnformalLinkMap();
        organizationalProblem = Parser.<IProblem>
        parseObject(property,
        "Organization.Problem.ClassName", property);
        socialProblem =
        organizationalProblem.cloneDeep(property);
        socialProblem.repulsion(repulsionRate);
        setPersonalProblems();
        setSamePersonalProblems();
        setAgents();
        evaluationAgent = new
        EvaluationAgent(this.property);
        organizationalDistributor = Parser.<IDistributor>
        parseObject(this.property,
        "OrganizationDistributor.ClassName",
        this.property);
        socialDistributor = Parser.<IDistributor>
        parseObject(this.property,
        "SocialDistributor.ClassName", this.property);
        updateEvaluation();
    }

    public void initialize(){
        steps = 0;
        setRandomGenerator();
        setUnformalLinkMap();
        organizationalProblem.initialize();
        socialProblem =
        organizationalProblem.cloneDeep(property);
        socialProblem.repulsion(repulsionRate);
        for (int i = 0; i <
        personalProblems.length; i++){
            //
            //
            //
            personalProblems[i].initialize();
        }
        setPersonalProblems();
        setSamePersonalProblems();
        setAgents();
        organizationalDistributor.fusion(organizationalPr
        oblem, agents);
        socialDistributor.fusion(socialProblem, agents);
    }

    organizationalDistributor = Parser.<IDistributor>
    parseObject(this.property,
    "OrganizationDistributor.ClassName",
    this.property);
    socialDistributor = Parser.<IDistributor>
    parseObject(this.property,
    "SocialDistributor.ClassName", this.property);
    updateEvaluation();
}

public Environment(String[] property) throws Exception {
    this.property = property;
    setRandomGenerator();
    changeStart = Parser.parseInteger(this.property,
    "Problem.Change.Start");
    sameNum =
    Parser.parseInteger(property,
    "Problem.Same.Num");
    friendNum =
    Parser.parseInteger(property, "Friend.Num");
    logger = new ILogger[0];
    setFormalLinkMap();
    setUnformalLinkMap();
    organizationalProblem =
    Parser.<IProblem> parseObject(property,
    "Organization.Problem.ClassName", property);
    socialProblem =
    organizationalProblem.cloneDeep(property);
    socialProblem.repulsion(repulsionRate);
    setPersonalProblems();
    setSamePersonalProblems();
    setAgents();
    evaluationAgent = new
    EvaluationAgent(this.property);
    organizationalDistributor = Parser.<IDistributor>
    parseObject(this.property,
    "OrganizationDistributor.ClassName",
    this.property);
    socialDistributor = Parser.<IDistributor>
    parseObject(this.property,
    "SocialDistributor.ClassName", this.property);
    updateEvaluation();
}

}
```

```

        updateEvaluation();
    }

public void next(){
    for(int i=0;i < agents.length;i++){
        if(RandomGenerator.nextDouble() < 0.1){
            agents[i].changeFriend(agents);
        }
        agents[i].nextState(personalProblems[i],
            organizationalProblem, socialProblem);
    }
    organizationalDistributer.fusion(organizationalPr
        oblem, agents);

    socialDistributer.fusion(socialProblem, agents);
    updateEvaluation();
    if(steps > changeStart){
        organizationalProblem.change(steps);
    }
    for (int i = 0; i < logger.length; i++) {
        logger[i].logStep(agents, organizationalProblem,
            socialProblem, steps);
    }
    ++steps;
}

public void end(int iterationTime){
    evaluationAgent.evaluate(agents,
        organizationalProblem, socialProblem,
        iterationTime);
}

public double getAverageOrganizationalUtility(){
    return
        evaluationAgent.getAverageOrganizationalUtility
        ();
}

public double getAverageVarianceOrganizationalUtility(){
    return
        evaluationAgent.getVarianceOrganizationalUtilit
        y();
}

public double getStandardDeviationOrganizationalUtility(){
    return
        evaluationAgent.getStandardDeviationOrganizati
        onalUtility();
}

public double getAveragePersonalUtility(){
    return
        evaluationAgent.getAveragePersonalUtility();
}

public double getBestOrganizationalUtility(){
    return
        evaluationAgent.getBestOrganizationalUtility();
}

public double
    getEstimationAverageOrganizationalUtility(){
    return
        evaluationAgent.getEstimationAverageOrganizati
        onalUtility();
}

private void setRandomGenerator(){
    try {
        RandomGenerator.setRandomGenerator(Parser.<
            Random> parseObject(property,
            "Random.ClassName"));
    } catch (Exception e) {
        e.printStackTrace();
    }
    if (Parser.parseBoolean(property,
        "Random.UseSeed")) {
        RandomGenerator.setSeed(Parser.parseLong(prop
            erty, "Random.Seed"));
    } else {
        RandomGenerator.setSeed(System.currentTimeM
            illis());
    }
}

private void setFormalLinkMap(){
    INetworkCreator networkCreator = null;
    try {
        networkCreator = Parser.<INetworkCreator>
            parseObject(property,
            "Agent.Creator.Network.ClassName");
        allAgentNum = Parser.parseInteger(property,
            "Agent.Creator.Number");
    } catch (Exception e) {
        e.printStackTrace();
    }
    formalLinkMap =
        networkCreator.createNetwork(allAgentNum,
        property);
}

private void setUnformalLinkMap(){
    for(int i = 0 ; i < allAgentNum ;i++){
        for(int j = 0 ; j < allAgentNum ;j++){

            if(formalLinkMap[i][j] == 2){
                formalLinkMap[i][j] = 0;
            }
        }
    }

    int friendNum1;
    int friendNum2;
    int i=0;
    while(i < friendNum){
        friendNum1 =
            RandomGenerator.nextInt(allAgentNum);
        friendNum2 =
            RandomGenerator.nextInt(allAgentNum);
        while(friendNum1 == friendNum2){
            friendNum2 =
                RandomGenerator.nextInt(allAgentNum);
        }
        if(formalLinkMap[friendNum1][friendNum2] ==
            0){
            formalLinkMap[friendNum1][friendNum2] = 2;
            formalLinkMap[friendNum2][friendNum1] = 2;
            i++;
        }
    }
}

private void setLogger(File experimentDirectory){
    try {
        Object[] parameterLogger = { this.property,
            experimentDirectory };
        Object[] objectsLogger =
            Parser.parseObjects(property,
            "Log.Logger.ClassName", parameterLogger);
        logger = new ILogger[objectsLogger.length];
        for (int i = 0; i < objectsLogger.length; i++) {
            logger[i] = (ILogger) objectsLogger[i];
        }
    } catch (Exception e) {
        logger = new ILogger[0];
    }
}

private void setPersonalProblems(){
    try {

```

```

int dependNum = 2;
Object[] parameter = { property , dependNum};
Object[] personalObjects =
Parser.parseSameObjects(property,
"Problem.ClassName", parameter);
personalProblems = new
IProblem[personalObjects.length];
for (int i = 0; i < personalObjects.length; i++){
personalProblems[i] = (IProblem)
personalObjects[i];
}
} catch (Exception e) {
e.printStackTrace();
}
}
private void setSamePersonalProblems(){
int[] sameProblem = new int[allAgentNum];
for(int i=0;i < sameProblem.length;i++){
sameProblem[i] = i;
}
int tmp,num;
for(int i=0;i < sameProblem.length;i++){
num =
RandomGenerator.nextInt(sameProblem.length);
tmp = sameProblem[i];
sameProblem[i] = sameProblem[num];
sameProblem[num] = tmp;
}
for(int i=0;i < sameNum;i++){
personalProblems[sameProblem[i]] =
personalProblems[sameProblem[0]];
}
}
private void setAgents(){
Value[] initialState = new Value[allAgentNum];
int dimation = Parser.parseInteger(property,
"Problem.Dimation");
int limitValue = Parser.parseInteger(property,
"Problem.Limit.Value");
for (int i = 0; i < allAgentNum; i++) {
int[] temp1 = new int[dimation];
for (int j = 0; j < temp1.length; j++) {
temp1[j] =
RandomGenerator.nextInt(limitValue);
}
initialState[i] = new Value(temp1);
}
try {
agentCreator =
Parser.<IAgentCreator> parseObject(property,
"Agent.Creator.ClassName");
agents = agentCreator.createAgents(property, f
ormalLinkMap, initialState);
} catch (Exception e) {
e.printStackTrace();
}
}
private void updateEvaluation(){
double personalUtility;
double organizationalReward;
double socialReward;
for(int i=0;i < agents.length;i++){
personalUtility =
personalProblems[i].getEvaluation(agents[i]);
organizationalReward =
organizationalDistributer.getDistribution(i);
socialReward =
socialDistributer.getDistribution(i);
agents[i].updateEvaluation(personalUtility,
organizationalReward, socialReward);
}
}

```

```

}
}
public EvaluationAgent getEvaluationAgent(){
return evaluationAgent;
}
}

```

NK Model Function: NK モデルの設定

```

package problem;

import java.util.ArrayList;
import java.util.List;

import common.IProblem;
import common.Parser;
import common.RandomGenerator;
import common.Value;
import common.Agent;

public class NKModelFunctionNew implements IProblem {
private int dimation;
private int dependNum;

private int limitValue;
private int changeTerm;
private int changeStep;
private double[] wait;
private NK[] nk;

public NKModelFunctionNew(String[] property)throws
Exception{
dimation = Parser.parseInteger(property,
"Problem.Dimation");
dependNum = Parser.parseInteger(property,
"Problem.DependNum");
limitValue = Parser.parseInteger(property,
"Problem.Limit.Value");
changeTerm = Parser.parseInteger(property,
"Problem.Change.Term");
changeStep = 0;
wait = new double[(int) Math.pow(limitValue,
dependNum)];
for(int i=0;i < wait.length;i++){
wait[i] = RandomGenerator.nextDouble();
}
nk = new NK[dimation];
for(int i=0;i < nk.length;i++){
nk[i] = new NK(i);
}
}

public NKModelFunctionNew(String[] property, Integer
dependNum)throws Exception{
dimation = Parser.parseInteger(property,
"Problem.Dimation");
this.dependNum = dependNum;
limitValue = Parser.parseInteger(property,
"Problem.Limit.Value");
changeTerm = Parser.parseInteger(property,
"Problem.Change.Term");
changeStep = 0;
wait = new double[(int) Math.pow(limitValue,
dependNum)];
for(int i=0;i < wait.length;i++){
wait[i] = RandomGenerator.nextDouble();
}
}
}

```

```

        nk = new NK[dimension];
        for(int i=0; i < nk.length; i++){
            nk[i] = new NK(i);
        }
    }

    public NKModelFunctionNew(String[] property, NK[] _nk,
        double[] _wait) throws Exception {
        dependNum = Parser.parseInteger(property,
            "Problem.DependNum");
        dimension = Parser.parseInteger(property,
            "Problem.Dimension");
        limitValue = Parser.parseInteger(property,
            "Problem.Limit.Value");
        changeTerm = Parser.parseInteger(property,
            "Problem.Change.Term");
        changeStep = 0;
        wait = new double[_wait.length];
        for(int i=0; i < _wait.length; i++){
            wait[i] = _wait[i];
        }
        this.nk = _nk;
    }

    public void change(int step) {
        if(step%changeTerm == 0)
            wait[RandomGenerator.nextInt(wait.length)] =
                RandomGenerator.nextDouble();
    }

    public double getEvaluation(Value value) {
        double evaluation;
        double temp = 0;
        int num = 0;
        if(dependNum==1){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i);
                temp += wait[num];
            }
        }
        if(dependNum==2){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i)
                + limitValue *
                value.getInteger(nk[i].getDependNum(1));
                temp += wait[num];
            }
        }
        if(dependNum==3){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i)
                + limitValue *
                value.getInteger(nk[i].getDependNum(1))
                + limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(2));
                temp += wait[num];
            }
        }
        if(dependNum==4){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i)
                + limitValue *
                value.getInteger(nk[i].getDependNum(1))
                + limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(2))
                + limitValue * limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(3));
                temp += wait[num];
            }
        }
        if(dependNum==5){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i)
                + limitValue *
                value.getInteger(nk[i].getDependNum(1))
                + limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(2))
                + limitValue * limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(3))
                + limitValue * limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(4));
                temp += wait[num];
            }
        }
        evaluation = (double)temp / (double)dimension;
        return evaluation;
    }

    public double getEvaluation(Agent agent) {
        Value value = agent.getCurrentState();
        double evaluation;
        double temp = 0;
        int num = 0;
        if(dependNum==1){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i);
                temp += wait[num];
            }
        }
        if(dependNum==2){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i)
                + limitValue *
                value.getInteger(nk[i].getDependNum(1));
                temp += wait[num];
            }
        }
        if(dependNum==3){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i)
                + limitValue *
                value.getInteger(nk[i].getDependNum(1))
                + limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(2));
                temp += wait[num];
            }
        }
        if(dependNum==4){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i)
                + limitValue *
                value.getInteger(nk[i].getDependNum(1))
                + limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(2))
                + limitValue * limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(3));
                temp += wait[num];
            }
        }
        if(dependNum==5){
            for(int i=0; i < dimension; i++){
                num = value.getInteger(i)
                + limitValue *
                value.getInteger(nk[i].getDependNum(1))
                + limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(2))
                + limitValue * limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(3))
                + limitValue * limitValue * limitValue *
                value.getInteger(nk[i].getDependNum(4));
                temp += wait[num];
            }
        }
        evaluation = (double)temp / (double)dimension;
        return evaluation;
    }

```

```

        temp += wait[num];
    }
    }
    evaluation = (double)temp / (double)dimention;
    return evaluation;
}

public NKModelFunctionNew clone() {
    try {
        return (NKModelFunctionNew)super.clone();
    } catch (CloneNotSupportedException e) {
        return null;
    }
}

public NKModelFunctionNew cloneDeep(String[]
property) {
    NKModelFunctionNew clone;
    try {
        clone = new NKModelFunctionNew(property,
this.nk,this.wait);
        return clone;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

public void initialize() {
    for(int i=0;i < wait.length;i++){
        wait[i] = RandomGenerator.nextDouble();
    }
    for(int i=0;i < nk.length;i++){
        nk[i] = new NK(i);
    }
}

private class NK{
    int[] depend;
public NK(int num){
    List<Integer> numbers = new
ArrayList<Integer>(dimention);
    for(int i = 0 ; i < dimention ; i++){
        numbers.add(i);
    }
    depend = new int[dependNum];
    depend[0] = numbers.remove(num);
    int temp;
    for(int i = 1 ; i < depend.length ; i++){
        temp =
numbers.remove(RandomGenerator.nextInt(num
bers.size()));
        depend[i] = temp;
    }
}

public int getDependNum(int num){
    return depend[num];
}

public void repulsion(double rate) {
    for(int i=0;(i + 1) / (double)wait.length <
rate;i++){
        wait[i] = 1.0 - wait[i];
    }
}
}

```

Agent: エージェントの定義

```

package common;
import java.util.HashSet;
import java.util.Set;

import common.Parser;
import common.IUpdateRule;
import common.Value;
import common.IProblem;

public class Agent{

    private Set<Agent> formalLinkedList;
    private Set<Agent> unformalLinkedList;
    private Value currentState;
    private Value preState;
    private Value nextState;
    private double evaluation;
    private double preEvaluation;
    private double personalUtility;
    private double organizationalReward;
    private double socialReward;
    private int limitValue;
    private boolean hasJumped;
    private int rank;
    private double jumpProbability;
    private double jumpThreshold;
    private IUpdateRule updateRule;
    ISearch search;

    public Agent(Value currentState, double jumpProbability,
double jumpThreshold, String[] property) throws
Exception {
        this.currentState = currentState.clone();
        this.formalLinkedList = new HashSet<Agent>();
        this.unformalLinkedList = new
HashSet<Agent>();
        this.updateRule = Parser.<IUpdateRule>
parseObject(property, "IUpdateRule.ClassName", property);
        this.evaluation = 0;
        this.limitValue = Parser.parseInteger(property,
"Problem.Limit.Value");
        this.hasJumped = false;
        this.jumpProbability= jumpProbability;
        this.jumpThreshold = jumpThreshold;
        this.search = Parser.<ISearch>
parseObject(property, "Search.ClassName", property);
    }

    public void nextState(IProblem personalProblem, IProblem
organizationalProblem, IProblem socialProblem) {
        serchNeighboring(personalProblem, organizationalProblem,
socialProblem); updateRule.nextState(this); int[] updatedState
= this.currentState.getInteger();
        for(inti=0;i<updatedState.length;i++){
            if(limitValue <= updatedState[i] ||
updatedState[i] < 0){
                }
            }
        }
        private void serchNeighboring(IProblem personalProblem,
IProblem organizationalProblem, IProblem socialProblem){
            nextState =
search.maxNeighboring(personalProblem,
organizationalProblem, socialProblem, this);
        }

        public void updateEvaluation(double personalUtility, double
organizationalReward, double socialReward) {

```

```

        setOrganizationalReward(organizationalReward);
        setSocialReward(socialReward);
        setPersonalUtility(personalUtility);
        evaluation = personalUtility +
organizationalReward + socialReward;
    }
    public int[] getMyNetworkBestState(){
        Agent[] formalLinkedList =
getFormalLinkedList();
        Agent[] unformalLinkedList =
getUnformalLinkedList();
        Agent myNetworkBest = this;
        for(int i=0;i<formalLinkedList.length;i++){
            if(myNetworkBest.getOrganizationalReward() <
formalLinkedList[i].getOrganizationalReward()){
                myNetworkBest = formalLinkedList[i];
            }
        }
        for(int i=0;i<unformalLinkedList.length;i++){
            if(myNetworkBest.getOrganizationalReward() <
unformalLinkedList[i].getOrganizationalReward(
))){
                myNetworkBest = unformalLinkedList[i];
            }
        }
        return
myNetworkBest.getCurrentState().getInteger();
    }
    public int[] getMyNetworkRandomState(){
        Agent[] formalLinkedList =
getFormalLinkedList();
        Agent[] unformalLinkedList =
getUnformalLinkedList();
        Agent myNetworkRandom = this;
        double
agentSumClosenessOrganizationalReward =
getSumClosenessOrganizationalReward();
        double randomAgentNum =
RandomGenerator.nextDouble() *
agentSumClosenessOrganizationalReward;
        double tempAgentSumCloseness = 0;
        for(int i=0;i<formalLinkedList.length;i++){
            tempAgentSumCloseness += (1 -
Value.length(currentState,
formalLinkedList[i].getCurrentState()) *
formalLinkedList[i].getOrganizationalReward());
            if( tempAgentSumCloseness >
randomAgentNum){
                myNetworkRandom = formalLinkedList[i];
                return
myNetworkRandom.getCurrentState().getInteger(
);
            }
        }
        for(int i=0;i<unformalLinkedList.length;i++){
            tempAgentSumCloseness += (1 -
Value.length(currentState,
unformalLinkedList[i].getCurrentState()) *
unformalLinkedList[i].getOrganizationalReward(
));
        }
        if( tempAgentSumCloseness > randomAgentNum){
            myNetworkRandom = unformalLinkedList[i];
            return
myNetworkRandom.getCurrentState().getInteger(
);
        }
    }
    myNetworkRandom = formalLinkedList[0];
    return
myNetworkRandom.getCurrentState().getInteger(
);
    }
}

}

public Value getNextState() {
    return nextState;
}

public Value getCurrentState() {
    return currentState;
}

public Value getPreState() {
    return preState;
}

public double getEvaluation() {
    return evaluation;
}

public double getPreEvaluation() {
    return preEvaluation;
}

public double getPersonalUtility(){
    return personalUtility;
}

public double getOrganizationalReward(){
    return organizationalReward;
}

public double getSocialReward(){
    return socialReward;
}

public int getFriendNum(){
    return unformalLinkedList.size();
}

public double getSumLength(){
    Agent[] formalAgents = getFormalLinkedList();
    Agent[] unformalAgents =
getUnformalLinkedList();
    int lengthValue = 0;
    for(int i=0; i < formalAgents.length; i++){
        lengthValue += Value.length(currentState,
formalAgents[i].getCurrentState());
    }
    for(int i=0; i < unformalAgents.length; i++){
        lengthValue += Value.length(currentState,
unformalAgents[i].getCurrentState());
    }
    return lengthValue;
}

public double getSumCloseness(){
    Agent[] formalAgents = getFormalLinkedList();
    Agent[] unformalAgents =
getUnformalLinkedList();
    double closenessValue = 0;
    for(int i=0; i < formalAgents.length; i++){
        closenessValue += 1 - Value.length(currentState,
formalAgents[i].getCurrentState());
    }
    for(int i=0; i < unformalAgents.length; i++){
        closenessValue += 1 - Value.length(currentState,
unformalAgents[i].getCurrentState());
    }
    return closenessValue;
}

public double getSumClosenessOrganizationalReward(){
    Agent[] formalAgents = getFormalLinkedList();
    Agent[] unformalAgents =
getUnformalLinkedList();
    double closenessValue = 0;
    for(int i=0; i < formalAgents.length; i++){
        closenessValue += (1 - Value.length(currentState,
formalAgents[i].getCurrentState())) *

```



```

        formalAgents[i].getOrganizationalReward();
    }
    for(int i=0; i < unformalAgents.length; i++){
        closenessValue += (1 - Value.length(currentState,
        unformalAgents[i].getCurrentState())) *
        unformalAgents[i].getOrganizationalReward();
    }
    return closenessValue;
}
public void setCurrentState(Value currentState) {
    this.currentState = currentState.clone();
}
public void setOrganizationalReward(double
organizationalReward){
    this.organizationalReward =
    organizationalReward;
}
public void setSocialReward(double socialReward){
    this.socialReward = socialReward;
}
private void setPersonalUtility(double personalUtility){
    this.personalUtility = personalUtility;
}
public boolean setFormalLink(Agent agent){
    return formalLinkedList.add(agent);
}
public boolean setUnformalLink(Agent agent){
    return unformalLinkedList.add(agent);
}
public boolean removeUnformalLink(Agent agent){
    return unformalLinkedList.remove(agent);
}
public Agent[] getFormalLinkedList() {
    return formalLinkedList.toArray(new
    Agent[formalLinkedList.size()]);
}
public Agent[] getUnformalLinkedList() {
    return unformalLinkedList.toArray(new
    Agent[unformalLinkedList.size()]);
}
public void setPreState(){
    preState = currentState.clone();
}
public void setPreEvaluation(){
    preEvaluation = evaluation;
}
public void setJump(){
    hasJumped = true;
}
public void cancelJump(){
    hasJumped = false;
}
public boolean hasJumped(){
    return hasJumped;
}
public void setRank(int rank){
    this.rank = rank;
}
public int getRank(){
    return rank;
}
public double getJumpThreshold(){
    return jumpThreshold;
}
public double getJumpProbability(){
    return jumpProbability;
}
public void addRank(){
    rank++;
}

```

```

public void makeFriendRandom(Agent[] agents) {
    boolean makedFriend = false;
    while( !makedFriend){
        int friendNum =
        RandomGenerator.nextInt(agents.length);
        if(!formalLinkedList.contains( agents[ friendNum
        ] )){
            if(!unformalLinkedList.contains( agents[ friendN
            um ] )){
                setUnformalLink( agents[ friendNum ]);
                agents[ friendNum ].setUnformalLink(this);
                makedFriend = true;
            }
        }
    }
}
public void changeFriend(Agent[] agents) {
    Agent[] unformalAgents =
    getUnformalLinkedList();
    if(unformalAgents.length != 0){
        double length;
        double tempLength = 0;
        Agent deleteAgent;
        deleteAgent = unformalAgents[0];
        length = Value.length(currentState,
        deleteAgent.currentState);
        for(int i = 0; i < unformalAgents.length; i++){
            tempLength = Value.length(currentState,
            unformalAgents[i].currentState);
            if( length < tempLength){
                length = tempLength;
                deleteAgent = unformalAgents[i];
            }
        }
        deleteAgent.removeUnformalLink(this);
        unformalLinkedList.remove(deleteAgent);
        if(RandomGenerator.nextDouble() < 0.5){
            this.makeFriendRandom(agents);
        }else{
            deleteAgent.makeFriendRandom(agents);
        }
    }
}
}

```

Hill Climb: 探索学习

```

package updateRule;

import common.Agent;
import common.IUpdateRule;
import common.RandomGenerator;
import common.Value;
import common.Parser;

public class HillClimb implements IUpdateRule {

    private double imitationProbability;
    private int dependNum;

    public HillClimb(String[] property){
        imitationProbability =
        Parser.parseDouble(property,
        "Agent.ImitationProbability");
        dependNum = Parser.parseInteger(property,

```

```

        "Problem.DependNum");
    }

    public void nextState(Agent myself) {
        if(myself.hasJumped()){
            judgeJump(myself);
        }else if(RandomGenerator.nextDouble() <
            myself.getJumpProbability()){
            jump(myself);
        }else{
            walk(myself);
        }
    }

    public void walk(Agent myself){
        myself.setCurrentState(myself.getNextState());
    }
    public void jump(Agent myself){
        myself.setPreState();
        myself.setPreEvaluation();
        int[] currentState =
            myself.getCurrentState().getInteger();
        int[] myNetworkBestState =
            myself.getMyNetworkRandomState();
        for(int i=0;i < currentState.length;i++){
            if(imitationProbability >
                RandomGenerator.nextDouble()){
                currentState[i] = myNetworkBestState[i];
            }
        }

        myself.setCurrentState(new Value(currentState));
        myself.setJump();
    }

    public void judgeJump(Agent myself){
        double currentEvaluation =
            myself.getEvaluation();
        double preEvaluationTreshold =
            myself.getPreEvaluation() *
            myself.getJumpThreshold();
        if(currentEvaluation <
            preEvaluationTreshold){
            myself.setCurrentState(myself.getPreState());
        }
        myself.cancelJump();
    }
}

```

Evaluation Agent: 模倣対象の決定

```

package common;

import java.io.IOException;
import common.IProblem;
import common.Agent;
import common.Parser;

public class EvaluationAgent{
    private double[] allAveragePersonalUtility;
    private double[]
        allAverageOrganizationalUtility;
    private double[] allAverageSocialUtility;
    private double[] allBestOrganizationalUtility;
    private int iterationTime;

    public EvaluationAgent(String[] property) throws
        IOException{

```

```

        iterationTime = Parser.parseInteger(property,
            "Agent.Iterator.Time");
        allAveragePersonalUtility = new
            double[iterationTime];
        allAverageOrganizationalUtility = new
            double[iterationTime];
        allAverageSocialUtility = new
            double[iterationTime];
        allBestOrganizationalUtility = new
            double[iterationTime];
    }

    public void evaluate(Agent[] agents,IProblem
        organizationalProblem,IProblem
        socialProblem,int iterationTime) {
        double sum;
        double averagePersonalUtility;
        double averageOrganizationalUtility;
        double averageSocialUtility;
        double bestOrganizationalUtility = 0;
        sum = 0.0;
        for (int i = 0; i < agents.length; i++) {
            sum += agents[i].getPersonalUtility();
        }
        averagePersonalUtility = sum / agents.length;
        sum = 0.0;
        double tempOrganizationalUtility = 0;
        for (int i = 0; i < agents.length; i++) {
            tempOrganizationalUtility =
                organizationalProblem.getEvaluation(agents[i].ge
                    tCurrentState());
            sum += tempOrganizationalUtility;
            if(bestOrganizationalUtility <
                tempOrganizationalUtility){
                bestOrganizationalUtility =
                    tempOrganizationalUtility;
            }
        }
        averageOrganizationalUtility = sum /
            agents.length;
        sum = 0.0;
        double tempSocialUtility = 0;
        for (int i = 0; i < agents.length; i++) {
            tempSocialUtility =
                socialProblem.getEvaluation(agents[i].getCurrent
                    State());
            sum += tempSocialUtility;
        }
        averageSocialUtility = sum / agents.length;
        allAveragePersonalUtility[iterationTime] =
            averagePersonalUtility;
        allAverageOrganizationalUtility[iterationTime]
            =averageOrganizationalUtility;
        allAverageSocialUtility[iterationTime]
            =averageSocialUtility;
        allBestOrganizationalUtility[iterationTime] =
            bestOrganizationalUtility;
    }

    public double getAverageOrganizationalUtility()
    {
        double averageOrganizationalUtility = 0;
        for(int i = 0; i <
            allAverageOrganizationalUtility.length ; i++){
            averageOrganizationalUtility +=
                allAverageOrganizationalUtility[i];
        }
        return
            averageOrganizationalUtility/(double)iterationTi
            me;
    }
}

```

```

public double getBestOrganizationalUtility() {
    double bestOrganizationalUtility = 0;
    for(int i = 0; i <
        allBestOrganizationalUtility.length ; i++){
        bestOrganizationalUtility +=
        allBestOrganizationalUtility[i];
    }
    return
    bestOrganizationalUtility/(double)iterationTime;
}

public double getAveragePersonalUtility() {
    double averagePersonalUtility = 0;
    for(int i = 0; i <
        allAveragePersonalUtility.length ; i++){
        averagePersonalUtility +=
        allAveragePersonalUtility[i];
    }
    return
    averagePersonalUtility/(double)iterationTime;
}

public double getVarianceOrganizationalUtility() {
    double sum = 0;
    for(int i = 0; i <
        allAverageOrganizationalUtility.length ; i++){
        sum += allAverageOrganizationalUtility[i];
    }
    double averageOrganizationalUtility =
    sum/(double)iterationTime;
    sum = 0;
    for(int i = 0; i < a
        llAverageOrganizationalUtility.length ; i++){
        sum += (averageOrganizationalUtility -
        allAverageOrganizationalUtility[i])
        * (averageOrganizationalUtility -
        allAverageOrganizationalUtility[i]);
    }
    double variance = sum/(double)iterationTime;
    return variance;
}

public double getStandardDeviationOrganizationalUtility()
{
    double sum = 0;
    for(int i = 0; i <
        allAverageOrganizationalUtility.length ; i++){
        sum += allAverageOrganizationalUtility[i];
    }
    double averageOrganizationalUtility =
    sum/(double)iterationTime;
    sum = 0;
    for(int i = 0; i <
        allAverageOrganizationalUtility.length ; i++){
        sum += (averageOrganizationalUtility -
        allAverageOrganizationalUtility[i])
        * (averageOrganizationalUtility -
        allAverageOrganizationalUtility[i]);
    }
    double variance = sum/(double)iterationTime;
    return Math.sqrt(variance);
}

public double getEstimationAverageOrganizationalUtility()
{
    double sum = 0;
    for(int i = 0; i <
        allAverageOrganizationalUtility.length ; i++){
        sum += allAverageOrganizationalUtility[i];
    }
    double averageOrganizationalUtility =
    sum/(double)iterationTime;
    double estimation = 1.96 *
    Math.sqrt(variance/(double)iterationTime);
    return estimation;
}

double averageOrganizationalUtility =
sum/(double)iterationTime;
sum = 0;
for(int i = 0; i <
    allAverageOrganizationalUtility.length ; i++){
    sum += (averageOrganizationalUtility -
    allAverageOrganizationalUtility[i])
    * (averageOrganizationalUtility -
    allAverageOrganizationalUtility[i]);
}
double variance = sum/(double)iterationTime;
double estimation = 1.96 *
Math.sqrt(variance/(double)iterationTime);
return estimation;
}

public double getAverageSocialUtility() {
    double averageSocialUtility = 0;
    for(int i = 0; i < allAverageSocialUtility.length ;
        i++){
        averageSocialUtility +=
        allAverageSocialUtility[i];
    }
    return
    averageSocialUtility/(double)iterationTime;
}

public double getStandardDeviationSocialUtility() {
    double sum = 0;
    for(int i = 0; i < allAverageSocialUtility.length ;
        i++){
        sum += allAverageSocialUtility[i];
    }
    double averageSocialUtility =
    sum/(double)iterationTime;
    sum = 0;
    for(int i = 0; i < allAverageSocialUtility.length ;
        i++){
        sum += (averageSocialUtility -
        allAverageSocialUtility[i])
        * (averageSocialUtility -
        allAverageSocialUtility[i]);
    }
    double variance = sum/(double)iterationTime;
    return Math.sqrt(variance);
}

public double getEstimationAverageSocialUtility() {
    double sum = 0;
    for(int i = 0; i < allAverageSocialUtility.length ;
        i++){
        sum += allAverageSocialUtility[i];
    }
    double averageSocialUtility =
    sum/(double)iterationTime;
    sum = 0;
    for(int i = 0; i < allAverageSocialUtility.length ;
        i++){
        sum += (averageSocialUtility -
        allAverageSocialUtility[i])
        * (averageSocialUtility -
        allAverageSocialUtility[i]);
    }
    double variance = sum/(double)iterationTime;
    double estimation = 1.96 *
    Math.sqrt(variance/(double)iterationTime);
    return estimation;
}

public double[] getAllAverageOrganizationalUtility(){
    return allAverageOrganizationalUtility;
}

```

```

    }
    public double[] getAllAverageSocialUtility(){
        return allAverageSocialUtility;
    }
}

```

Value: 満足度の判定

```

package common;

public class Value implements Cloneable{

    private int[] value;
    private int[][] valueList;

    public Value(int size){
        value = new int[size];
        for(int i=0;i < value.length;i++){
            value[i] = 0;
        }
    }

    public Value(int[] value){
        this.value = value.clone();
    }

    public Value(int[] value1,int[] value2){
        int[] sum = new int[value1.length];
        for(int i=0;i < sum.length;i++){
            sum[i] = value1[i] + value2[i];
        }
        this.value = sum.clone();
    }

    public void plusVector(int multi,int[] value1){
        for(int i=0;i < value.length;i++){
            value[i] += value1[i] * multi;
        }
    }

    public int getInteger(int num){
        return value[num];
    }

    public int[] getInteger(){
        return value.clone();
    }

    public int getSize(){
        return value.length;
    }

    public String toString(){
        StringBuilder sb = new StringBuilder();
        for(int i=0; i < value.length; i++){
            sb.append(value[i]);
        }
        return sb.toString();
    }

    public void set(int num,int newValue){
        this.value[num] = newValue;
    }

    public static double length(Value value1, Value value2){
        int sum = 0;
        for(int i=0;i < value1.getSize();i++){

```

```

            if(value1.getInteger(i) != value2.getInteger(i)){
                sum++;
            }
        }
        double length = sum / (double)value1.getSize();
        return length;
    }

    public static double length(int[] value1,int[] value2){
        int sum = 0;
        for(int i=0;i < value1.length;i++){
            if(value1[i] != value2[i]){
                sum++;
            }
        }
        double length = sum / (double)value1.length;
        return length;
    }

    public void setValueList(int[][] valueList){
        this.valueList = valueList.clone();
    }

    public double getAverageLength(){
        double sum = 0;
        for(int i = 0; i < valueList.length; i++){
            sum +=length(this.getInteger(),valueList[i]);
        }
        double average = sum / valueList.length;
        return average;
    }

    public Value clone() {
        return new Value(getInteger());
    }
}

```