

## 修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院情報システム学研究科 社会知能情報学専攻 博士前期課程		
氏 名	布施 太章	学籍番号	1351016
論文題目	マルチエージェントプランニングにおける 動的協調形態変更メカニズムに関する研究		
<p>要 旨</p> <p>近年、一般家庭やオフィス等の日常生活環境に対するアンビエントシステムの構築に関する研究が注目されている。一般的に日常生活拠点では複数人が生活しており、その各個人の行動を円滑にサポートするインタラクションを自律的に実行することがアンビエントシステムの目的である。そのためには、生活拠点に設置された有限のインタラクティブデバイスを適切に操作できる能力が要求される。このアンビエントシステムを制御する方法として、各人ごとにインタラクティブデバイス制御のためのプランニングを行うエージェントがお互いに協調競合することが研究されている。</p> <p>本研究では、実世界でのデバイス運用のような環境が動的に変化する状況下においてマルチエージェントの協調形態を利用したプランニングの検討を行った。実環境では環境の変化に対応するプラン修正能力と実際にプランが行われている中でプランの競合を解消させる即応性を求められる。更に協調形態が複数共存させるモデルを考えた場合、直接協調型がプラン修正能力と即応性に長けた状況も生じると考えられる。</p> <p>具体的には、既存研究に対して、これらの共存モデルへの拡張及び、直接協調型の導入も含めた、中央制御型と直接協調型、間接協調型の 3 つの協調形態が共存するといったモデルを提案した。提案したモデルのうち、中央制御型と間接協調型の共存型に対して、マルチエージェントシミュレーションプラットフォームである <b>artisoc</b> にて仮想プランニング空間を構築して実験を行った。実験では、プランニング精度とプランニング実行時間の検証を行い、大規模空間になった際の中央制御型の統率を司るエージェントが複数存在する状況では、中央制御型と間接協調型の共存型を導入することでプランニング精度が向上する可能性を示した。</p>			



平成26年度  
修士論文

マルチエージェントプランニングにおける  
動的協調形態変更メカニズムに関する研究

電気通信大学  
大学院情報システム学研究科  
社会知能情報学専攻

学籍番号 : 1351016

氏名 : 布施 太章

主任指導教員 : 栗原 聡 教授

指導教員 : 田中 健次 教授

指導教員 : 大須賀 昭彦 教授

平成27年1月26日(月) 提出

## マルチエージェントプランニングにおける 動的協調形態変更メカニズムに関する研究

近年，一般家庭やオフィス等の日常生活環境に対するアンビエントシステムの構築に関する研究が注目されている．一般的に日常生活拠点では複数人が生活しており，その各個人の行動を円滑にサポートするインタラクションを自律的に実行することがアンビエントシステムの目的である．そのためには，生活拠点に設置された有限のインタラクティブデバイスを適切に操作できる能力が要求される．このアンビエントシステムを制御する方法として，各人ごとにインタラクティブデバイス制御のためのプランニングを行うエージェントがお互いに協調競合することが研究されている．

本研究では，実世界でのデバイス運用のような環境が動的に変化する状況下においたマルチエージェントの協調形態を利用したプランニングの検討を行った．実環境では環境の変化に対応するプラン修正能力と実際にプランが行われている中でプランの競合を解消させる即応力を求められる．更に協調形態が複数共存させるモデルを考えた場合，直接協調型がプラン修正能力と即応性に長けた状況も生じると考えられる．

具体的には，既存研究に対して，これらの共存モデルへの拡張及び，直接協調型の導入も含めた，中央制御型と直接協調型，間接協調型の3つの協調形態が共存するといったモデルを提案した．提案したモデルのうち，中央制御型と間接協調型の共存型に対して，マルチエージェントシミュレーションプラットフォームである *artisoc* にて仮想プランニング空間を構築して実験を行った．実験では，プランニング精度とプランニング実行時間の検証を行い，大規模空間になった際の中央制御型の統率を司るエージェントが複数存在する状況では，中央制御型と間接協調型の共存型を導入することでプランニング精度が向上する可能性を示した．

キーワード：MAS(Multi-Agent System：マルチエージェントシステム)，プランニング，協調

# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	研究背景	1
1.2	解決すべき課題	2
1.3	研究の目的	2
<b>第2章</b>	<b>エージェントとプランニング</b>	<b>4</b>
2.1	エージェント	4
2.1.1	エージェントと環境	5
2.1.2	エージェントプログラム	6
2.2	プランニング	10
2.2.1	熟考的プランニング	11
2.2.2	即応的プランニング	17
2.2.3	熟考的プランニングと即応的プランニングの統合	20
<b>第3章</b>	<b>関連研究：</b>	
	<b>MASにおける協調形態メカニズム</b>	<b>25</b>
3.1	中央制御型	25
3.2	直接協調型	26
3.3	間接協調型	26
3.4	従来研究に対する本研究の位置付け	27
<b>第4章</b>	<b>提案手法：</b>	
	<b>マルチエージェントプランニングにおける</b>	
	<b>動的協調形態変更メカニズム</b>	<b>29</b>
4.1	中央制御型と間接協調型の共存型	30
4.2	直接協調型の追加	32
<b>第5章</b>	<b>評価用シミュレーション環境の構築：</b>	
	<b>中央制御型と間接協調型の共存型</b>	<b>34</b>
5.1	マルチエージェントシミュレーション	34
5.2	エージェントの構成	35
5.3	実験環境	36
5.3.1	前処理過程	39
5.3.2	行動エージェントの振る舞い	39
<b>第6章</b>	<b>実験と考察：</b>	
	<b>中央制御型と間接協調型の共存型</b>	<b>43</b>

6.1	実験 1 :	
	各協調形態の共存の様子 . . . . .	43
6.2	実験 2 :	
	中央制御型と間接協調型の共存型の分析 . . . . .	48
6.2.1	プランニング精度 . . . . .	49
6.2.2	プランニング実行時間 . . . . .	50
<b>第 7 章</b>	<b>結論</b>	<b>52</b>
	<b>謝辞</b>	<b>53</b>
	<b>参考文献</b>	<b>54</b>
付 録 A	中央制御型と間接協調型の共存型 競合回避	58
付 録 B	中央制御型と間接協調型の共存型 分析	61

# 目次

2.1 エージェントと環境	4
2.2 エージェントプログラム	6
2.3 単純条件反射エージェント	7
2.4 モデルベース条件反射エージェント	7
2.5 目的ベースエージェント	8
2.6 効用ベースエージェント	8
2.7 プランニングを行うエージェントの主な構成	9
2.8 熟考的プランニング	10
2.9 即応的プランニング	10
2.10 積み木問題	12
2.11 巡回セールスマン問題	22
2.12 3T アーキテクチャ	23
2.13 MRR-プランニング	24
3.1 MASにおける協調メカニズムの特徴	27
3.2 中央制御型と間接協調型の変更型：アルゴリズム	28
4.1 中央制御型と間接協調型の変更型：概略図	29
4.2 中央制御型と間接協調型の共存：アルゴリズム	31
4.3 中央制御型と間接協調型の共存：概略図	31
4.4 MASにおける協調メカニズムの共存：アルゴリズム	33
4.5 MASにおける協調メカニズムの共存：概略図	33
5.1 プランニング例	35
5.2 シミュレーションの過程	37
5.3 実行動作例	38
5.4 競合解消フェーズ	40
5.5 実行フェーズ	42
6.1 実験1：プランニング精度	45
6.2 実験1：プランニング実行時間	46
6.3 実験2：プランニング精度	49
6.4 実験2：プランニング実行時間	51
A.1 中央制御型プランニングの競合回避の例	59
A.2 間接協調型プランニングの競合回避の例	60
B.1 artisoc 実行画面：行動エージェント数と $\theta$ の関係図	61
B.2 artisoc 実行画面： $\theta = 11 \sim 20$	62

B.3	artisoc 実行画面 : $\theta = 21 \sim 30$	63
B.4	artisoc 実行画面 : $\theta = 31 \sim 40$	64

## 表 目 次

1.1	各連携形態の特徴 . . . . .	3
2.1	タスクと環境の分類 . . . . .	6
2.2	初期条件およびゴール条件の状態記述 . . . . .	12
6.1	実験環境 1 . . . . .	43
6.2	実験 1：プランニング精度 . . . . .	45
6.3	実験 1：プランニング実行時間 . . . . .	46
6.4	実験環境 2 . . . . .	48



# 第1章 序論

## 1.1 研究背景

近年、自律行動を行うロボットのようなデバイスの研究が盛んに行われている [1][2][3][4]. 例えば、人がよりの確な情報をよりの確なタイミングにて獲得できることを実現する電子秘書システム [1] や、円滑な生活に加え、日常生活における安全安心を確保するための、ネットワーク機能をもつ高齢者福祉ペット型ロボットシステム [2] がある. どちらもロボット等による日常生活のサポートの実現とその質の向上を目指している. そして、共通する重要な機能が、「それぞれのシステムが、どの人に対してどのような働き掛けをどのような手段にて実行するのか」を決定する仕組みであり、言い換えると、「目的を達成するためにどのような手順で実行するのか」を決定する仕組みと言え、これはプランニングと呼ばれる. 特に複数の自律システムが連携してプランニングを行う枠組みは「マルチエージェントプランニング」と呼ばれる.

また、空間が大規模化すると、自律的な行動を行うデバイスが複数存在する場合は考えられる. この際に、それぞれ目的のための行動を行った結果、デバイス同士が競合を起こしてしまい、それぞれの行動を阻害してしまう可能性がある. 仮にデバイスがユーザ補助を目的とした行動を行うと計画していた時、デバイス制御の競合によって補助が正しく実行できない場合、環境内のユーザに対して有効な補助ができなかったり、最悪の場合事故が発生してしまうことも考えられる. このような事態を回避するために、デバイスの目的を安全に実行するためのプランニング技術が必要となる. さらに、人が活動するような環境は動的環境であるから、プランニング技術には適応性の高い、時間制約下での協調プランニングについて考えなければならない.

## 1.2 解決すべき課題

個々の携帯端末やロボットがそれぞれプランニングを実行することを考えると、自身の計算能力と実行するプランニングの規模の大きさに依存して、個々がそれぞれプランニングを実行する形態が好ましい場合や、計算能力の高いサーバ等で一括してプランニングを行う形態が望ましい状況もある。また、サーバへの負荷が過度に集中した場合や端末とサーバ間での通信品質が劣化した場合には、一部のシステムにおいては端末でのプランニングを行う形態の方が好ましい場合など、個々のシステムがそれぞれ最適なプランニングを実行するために他のシステムと連携する形態は状況により異なる。

更に、これら自律システムの導入が普及し日常生活の様々なところに偏在し、人のサポートを目的としてそれぞれが勝手にプランニングを行うことで、異なるシステムがそれぞれのプラン実行のために同一のデバイスの制御権の獲得を巡る競合問題も表面化する。例えば、介護ホーム等での利用を想定したロボット型電動車椅子のハードウェア開発が進んでいるが、個々の車いすが高齢者の意志のまま最短経路にて勝手に移動すれば、狭い廊下での鉢合わせや、トイレ等の目的地に同時に到着することによる競合等が発生してしまうことから、ナビゲーションのための車椅子同士の連携が必要となる。

そして、万が一にも、プランが実行できないことにより、有効な補助ができず事故が発生するような事態は避けなければならない。つまり、インタラクションデバイスのプランニング技術には以下の 2 点が必要である。

- 計画されたプランが環境変化によって実行不能にならないようにするための修正能力
- 時間制約下でプランニングする即応性

## 1.3 研究の目的

1.2 節のインタラクションデバイスのプランニング技術を満たす手法の 1 つに協調形態の利用が考えられる。

連携の仕方においては、その最適な連携形態が状況に応じて異なることから、動的に連携形態を変更できる仕組みが必要である。加えて連携形態を変更させる過程では、個々の

システムが立案中のプランや実行のプランの整合性を維持・保証できなければならない。連携形態としては、中央制御型、直接協調型、間接協調型等が代表的なものである。各特徴を簡単に表 1.1 に示す。

プランニングに関する研究の歴史は深く、前述したのような個々の連携形態についての研究は成熟されているといえる。しかし、連携形態そのものを動的に変更する枠組みについての手法はほとんど確立されておらず、本研究ではこれに着目する。

表 1.1: 各連携形態の特徴

	中央制御型	直接協調型	間接協調型
特徴	<ul style="list-style-type: none"> <li>・全体を統括しているエージェントは全体を把握できる</li> <li>・全体を統括しているエージェントに情報が集まる</li> </ul>	<ul style="list-style-type: none"> <li>・各エージェントが自律的に計算</li> <li>・各エージェントが自律的に計算</li> </ul>	<ul style="list-style-type: none"> <li>・各エージェントが自律的に計算</li> <li>・共有メモリにアクセスする回数はエージェント数に依存しない</li> <li>・全体を統括しているエージェントを介さない</li> </ul>
メリット	<ul style="list-style-type: none"> <li>・全体を見た最適解を見つけることが可能</li> </ul>	<ul style="list-style-type: none"> <li>・計算負荷の局所化の防止</li> </ul>	<ul style="list-style-type: none"> <li>・計算負荷の局所化の防止</li> <li>・エージェント同士の情報伝達の負担軽減</li> </ul>
デメリット	<ul style="list-style-type: none"> <li>・計算コストが全体を統括しているエージェントに集中</li> </ul>	<ul style="list-style-type: none"> <li>・通信コストが高い</li> </ul>	<ul style="list-style-type: none"> <li>・最適解の精度が低い</li> </ul>

## 本論文の構成

以下に本論文の構成を述べる。まず第 2 章では、マルチエージェントシステムプランニング技術に関してまとめる。次に第 3 章では、マルチエージェントシステムの協調形態についてまとめ、自身の研究の位置づけを述べる。第 4 章では、マルチエージェントプランニングにおける動的協調形態変更メカニズムを提案する。第 5 章では、提案手法を元に評価用シミュレーション環境の構築について述べる。第 6 章では、提案手法についての検証実験をまとめる。最後に第 8 章で本論文のまとめと今後の課題について述べる。

## 第2章 エージェントとプランニング

### 2.1 エージェント

自分の行動を環境とのインタラクションを通じて自ら決定できる自律行動主体をエージェントと呼ぶ。以下2.1節では，エージェントの概要を [5] を引用して述べる。

エージェントはある環境をセンサを通じて感じ取り，あるルールに則り，その環境にアクチュエータを通してはたらきかける (図2.1 参照)。一般的には，ある時点におけるエージェントの行為選択は，それまで観測された知覚系列<sup>1</sup>全体に依存する。エージェントの振舞いは知覚系列を行為へ対応づけるエージェント関数によって表されることになる。このエージェント関数は図2.1のルール部分にあたり，エージェントプログラムによって定義される。なお，エージェントプログラムの詳細については2.1.2節で述べる。

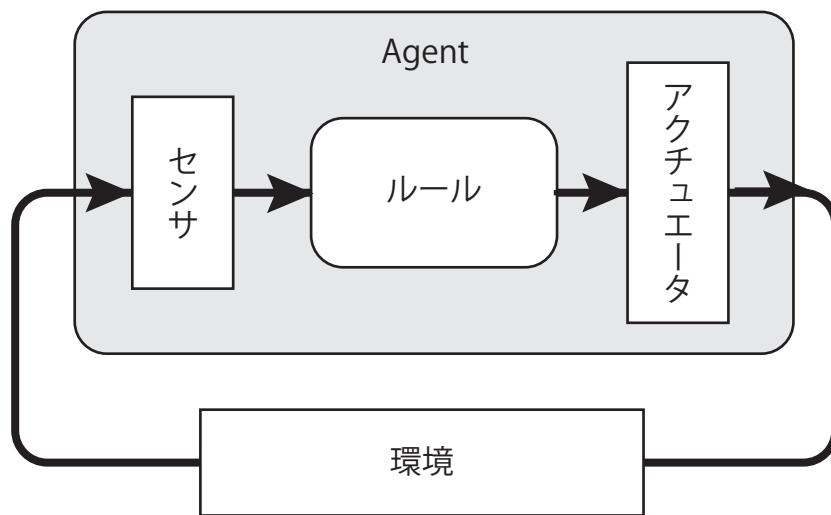


図 2.1: エージェントと環境

<sup>1</sup>知覚系列：エージェントが感じたこと (知覚) の完全な履歴

### 2.1.1 エージェントと環境

どんなエージェントを作るか、といったことを考える上で、エージェント関数以前に、前提条件が強く影響する。そのため、前提条件であるタスク環境を考慮する必要がある。

#### タスク環境の性質

タスク環境の主な性質を以下に示す。なお、それぞれ右に行くほど条件が厳しくなる。

観測可能性 完全観測可能 or 部分観測可能

決定性 決定的 or (戦略的<sup>2</sup> or) 確率的

時間依存性 挿話的<sup>3</sup> or 系列的

流動性 静的 or (準動的<sup>4</sup> or) 動的

連続性 離散的 or 連続的

構成数 単一エージェント or マルチエージェント

例えば、タクシーの自動運転を行うエージェントのタスク環境の性質の場合を述べる。観測可能性に関して、ドライバーは他のドライバーが何を考えているかを知ることにはできないということから、部分観測可能となる。決定性に関して、交通状況の正確な予測が困難で、さらに突然タイヤがパンクしたり、エンジンがオーバーヒートしたりすることもあるため、確率的である。時間依存性に関して、短期の行為が長期にわたり結果に影響を与えるため系列的である。流動性に関して、運転アルゴリズムが次に何をすべきか判断しあぐねているうちに、他の車両やそのタクシー自体が動きつづけるため動的である。連続性に関して、タクシーや他の車両のスピードや位置が連続な値の範囲にわたって滑らかに時間変化するように、連続状態と連続時間で表される問題である。そのため、タクシー運転

<sup>2</sup>他のエージェントの行為を除いて環境が決定的である場合を指す。

<sup>3</sup>単発的とも言う。

<sup>4</sup>環境自体は時間経過とともに変化することはないが、エージェントの性能指標(どれくらい成功したかを評価する基準)に時間経過が反映される場合を指す。

の行為もまた（ハンドルの角度などのように）連続的である。構成数に関して，衝突を避けることがすべてのエージェントの性能指標を最大化するので，部分的には協調的マルチエージェントである。

表 2.1 に，いくつか代表的な環境の性質を示す。

表 2.1: タスクと環境の分類

タスク \ 環境	観測可能性	決定性	時間依存性	流動性	連続性	構成数
クロスワードパズル	完全	決定的	系列的	静的	離散的	シングル
タクシードライバー	部分	確率的	系列的	動的	連続的	マルチ
英語教師エージェント	部分	確率的	系列的	動的	離散的	マルチ

### 2.1.2 エージェントプログラム

2.1 節の図 2.1 のルール部分はエージェントプログラムで定義される (図 2.2)。すべての

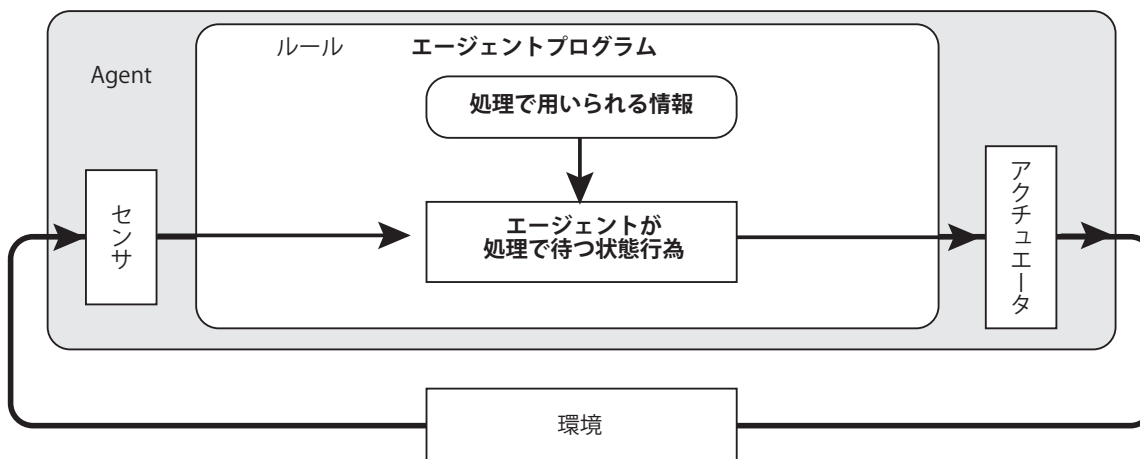


図 2.2: エージェントプログラム

知的エージェントが有する原理を含む基本的エージェントプログラムには，単純条件反射エージェント，モデルベース条件反射エージェント，目的ベースエージェント，効用ベースエージェントの4つがある。以下に簡潔に特徴を述べる。

## 単純条件反射エージェント

現在の知覚をもとに行動し過去の知覚履歴は無視する。したがって、現在の知覚のみに基づいて正しい判断が行われ造得る場合、すなわち環境が完全観測可能である場合に限り、正しく動作する (図 2.3).

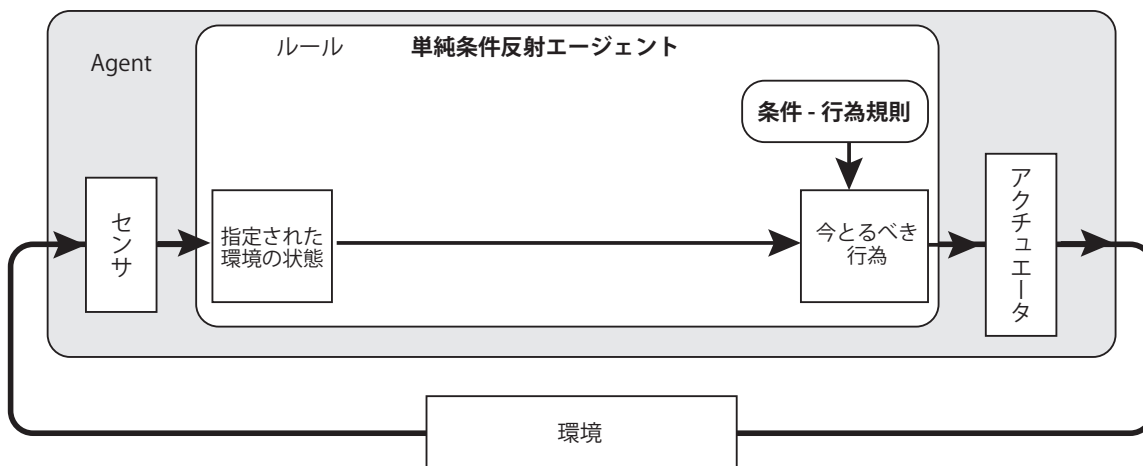


図 2.3: 単純条件反射エージェント

## モデルベース条件反射エージェント

エージェントが直接観測できない現在の状態部分を反映する内部状態を知覚履歴に基づいて管理することで、現在みることができないこの世の部分を追跡し、部分的観測可能性に対応する (図 2.4).

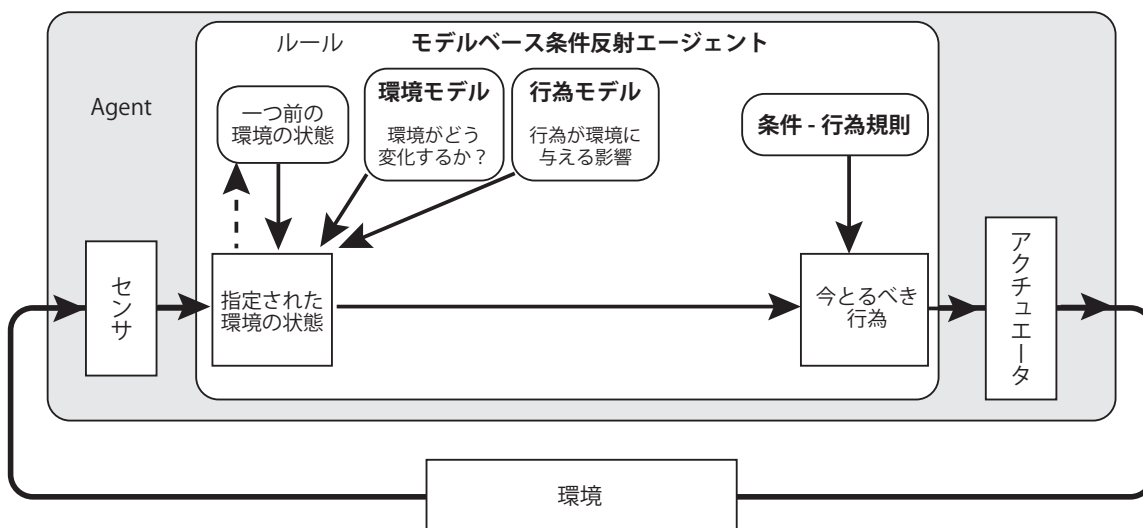


図 2.4: モデルベース条件反射エージェント

## 目的ベースエージェント

可能な行為の結果に関する情報（条件反射エージェントで内部状態を更新するために使われる情報と同じ）と目的状態を組み合わせることで、目的を達成するための行為を選択する (図 2.5).

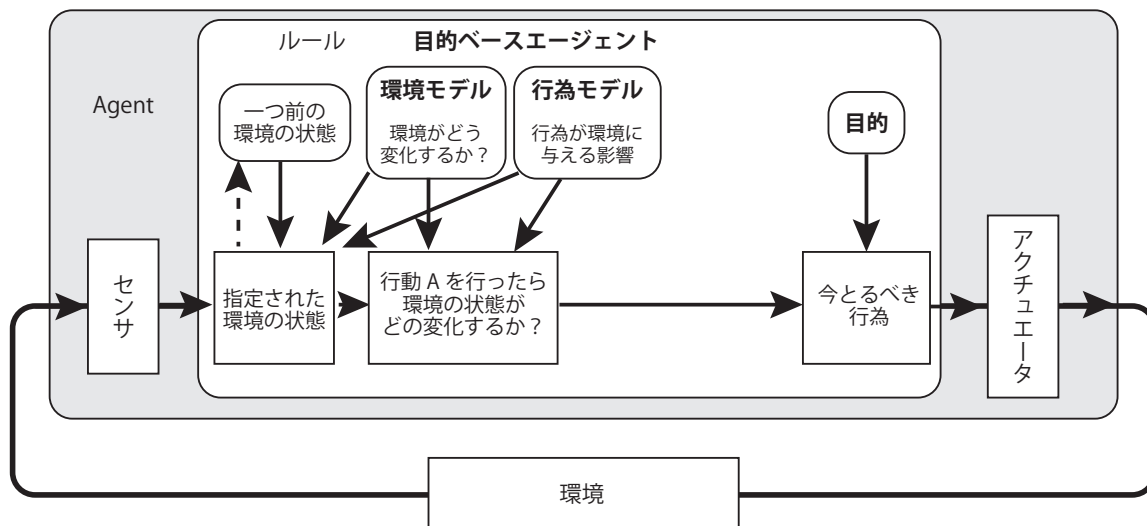


図 2.5: 目的ベースエージェント

## 効用ベースエージェント

目的が達成できる行動（行動系列）が複数ある、もしくは目的自体が複数ある場合に、行動の仕方や実現された状態の善し悪しの指標である効用を利用し、より高い効用を保ちつつ目的を達成する行為を選択する (図 2.6).

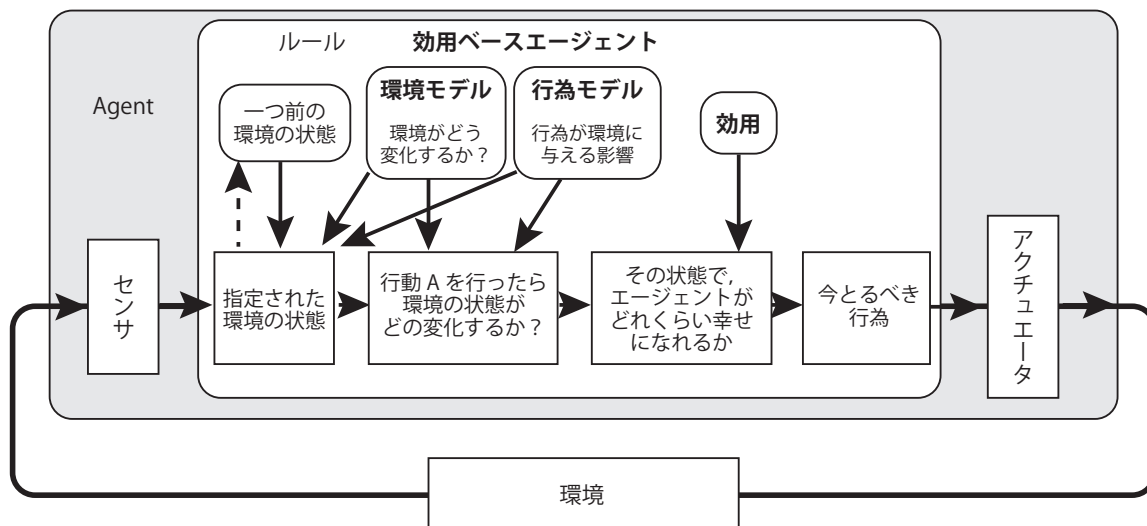


図 2.6: 効用ベースエージェント



人工知能の研究分野として、探索やプランニングがある。これらは、目的ベースエージェントで挙げられている目的を達成する行為系列を発見するための手法の1つである。プランニングを行うエージェントは、内部状態で記述された初期状態や現状態を認識する状況認識モジュール、内部状態を更新するために使われる情報である環境モデル、目的を達成するためのプランニング、ゴール状態である目的の4つから構成される(図2.7)。次節2.2では、プランニングについて詳細に述べる。

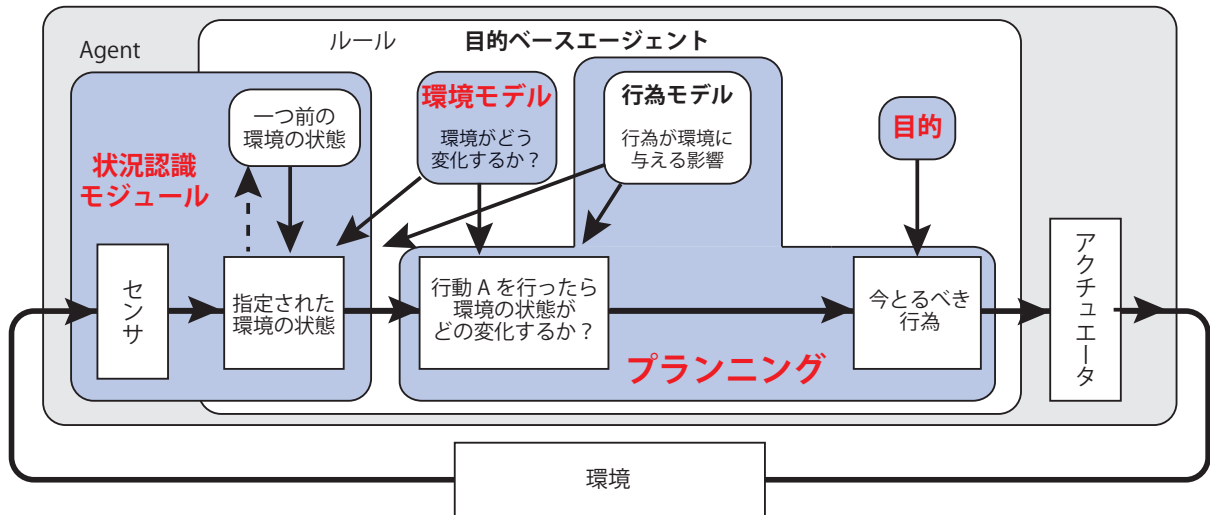


図 2.7: プランニングを行うエージェントの主な構成

## 2.2 プランニング

プランニングは人工知能研究における重要な研究テーマであり、これまで多くの研究が行われている<sup>5</sup>。プランニングは大きく熟考的な手法（以降、熟考的プランニングと呼ぶ）と即応的な手法（以降、即応的プランニングと呼ぶ）に分類することができる。その中でも、2.1.1 節で示したタスク環境が、完全観測可能、決定的で、挿話的 (単発的) で、静的で、かつ離散的な環境は、熟考的プランニング環境と呼ばれる (図 2.8)。対照的に、即応的プランニングは、部分観測可能であったり、確率的である環境に対するプランニングであり、異なったアルゴリズムとエージェント設計を必要とする。

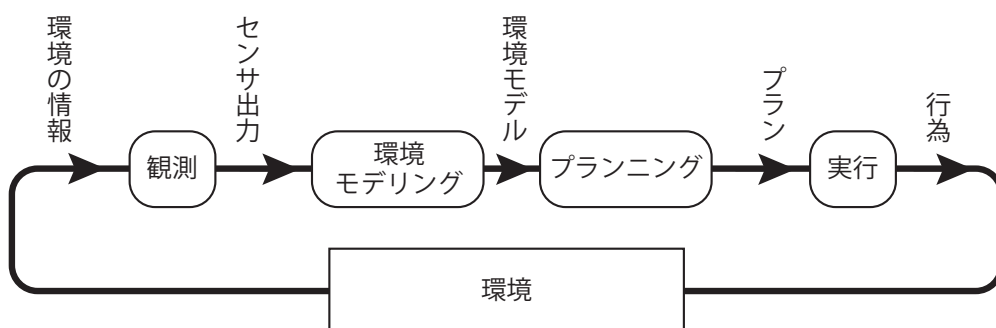


図 2.8: 熟考的プランニング

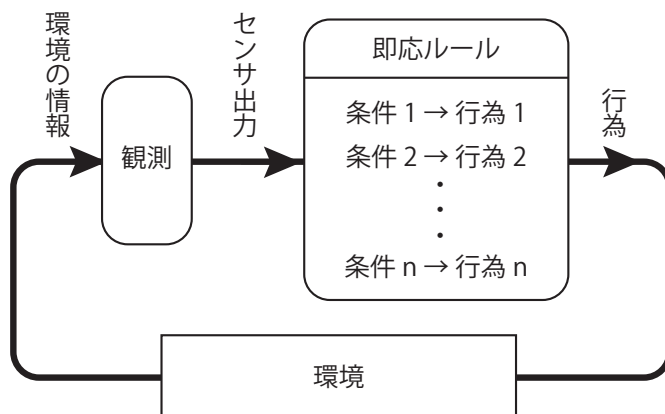


図 2.9: 即応的プランニング

<sup>5</sup>研究事例は本節の手法ごとに参考文献を参照する。

熟考的プランニングを行う自律行動主体の活動の流れは、図 2.8 のように観測→環境のモデリング→プランニング→実行の繰り返しになるが、前述のようにプランニングと環境のモデリングに多大な時間がかかるため、その動きが非常に緩慢になる。さらに、熟考的プランニングは、「生成されたプランが現実世界で確実に実行される」という前提を置いているが、実環境では、環境は容易に変化し、行為はしばしば失敗する [6]。

それに対して、即応的プランニングは、動的環境において目標を達成することを指向したプランニングである [7]。即応的プランニングの構成を図 2.9 に示す。その基本となる考えは、「自律行動主体が実世界で知的に行動するために、従来の多大な探索を伴うようなプランニングは必要でなく、その場の環境に応じて断片的な行動を行えばよい」というものである。

本節では、まずこれまでのプランニング研究の概要を簡単に述べる。

### 2.2.1 熟考的プランニング

これまで提案されてきたプランニング手法のほとんどが熟考的プランニングに属する。熟考的プランニングは古典的プランニングとも呼ばれる。以下に代表的な手法である、STRIPS と A\* アルゴリズムの概要を述べる。

#### STRIPS

最も有名な古典的プランニングが STRIPS (STanford Research Institute Problem Solver) である [8]。STRIPS の原理は今日のほとんどのプランニングで用いられており、センサから得た情報を用いて、内部状態で記述された初期状態（現状態）のモデルとゴール状態のモデルを構成する。そして、初期状態をゴール状態に変換するための単位行動系列を生成する (図??)。ロボットが何らかの行動を行うというのは、別の状態を作り出すことを意味し、その最小単位が単位行動である。最適な単位行動系列が得られれば、このプランニング結果に従って移動するロボットは最適な行動選択を行うことができる。このように STRIPS は時間をかけて精度の高いプランを生成することを目的とする熟考的プランニングである。以下 STRIPS の概要を述べる。

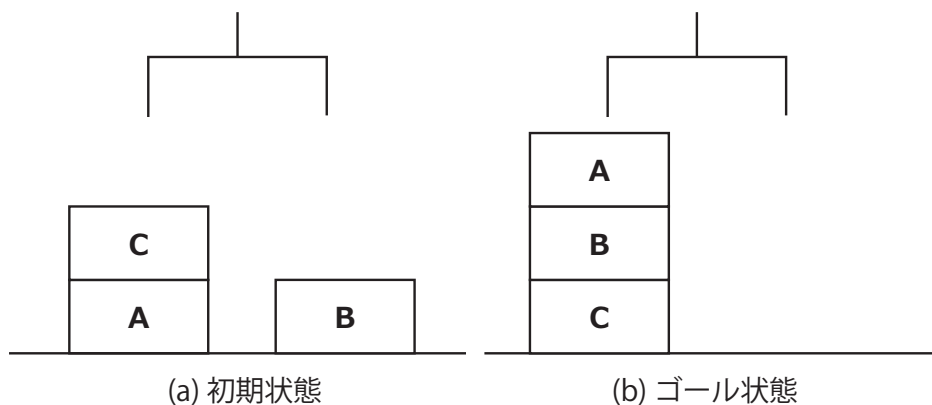


図 2.10: 積み木問題

STRIPS では、状態は関数を含まないリテラルで表現される。例えば図 2.10 のような簡単な積み木問題<sup>6</sup>を考える。図 2.10 の (a) 初期状態を  $I$ , (b) ゴール状態を  $G$  とすると、リテラルの連言で表され、次式 2.1, 2.2 になる。また、各々のリテラルの状態記述を表 2.2 に示す。

$$I = \text{CLEAR}(B) \wedge \text{CLEAR}(C) \wedge \text{ON}(C, A) \\ \wedge \text{ONTABLE}(A) \wedge \text{ONTABLE}(B) \wedge \text{HANDEEMPTY} \quad (2.1)$$

$$G = \text{ON}(B, C) \wedge \text{ON}(A, B) \quad (2.2)$$

表 2.2: 初期条件およびゴール条件の状態記述

初期条件		ゴール条件	
リテラル	状態記述	リテラル	状態記述
CLEAR(B)	B の上は空である	ON(B, C)	B が C の上にある
CLEAR(C)	C の上は空である	ON(A, B)	A が C の上にある
ON(C, A)	C が A の上にある		
ONTABLE(A)	A が机の上にある		
ONTABLE(B)	B が机の上にある		
HANDEEMPTY	手が空である		

<sup>6</sup>積み木：積み上げることができるが、1つの積み木の上には1つの積み木しか載らない。

アーム：積み木を持ち上げ、テーブルの上や他の積み木の上に移動することができる。また、一度に1つの積み木だけ持ち上げることができ、他の積み木が上に載っている積み木を持ち上げることはできない。

ロボットにとってこれらのリテラルで述べられていることだけが真で、そのほかのすべては偽とみなされる。単位行動は、それが適用可能であるための前提条件<sup>7</sup>、その適用によって削除される状態リスト<sup>8</sup>、および追加される状態<sup>9</sup>を表現した3種類のリテラルで記述する。例えば図 2.10 のような積み木問題の場合は以下の単位行動を組み合わせることで解くことができる。

---

**pickup**( $x$ ) : テーブル上の積み木  $x$  をアームに持つ

P<sup>7</sup> :  $\text{ONTABLE}(x) \wedge \text{CLEAR}(x) \wedge \text{HANDEEMPTY}$

D<sup>8</sup> :  $\text{ONTABLE}(x)$ ,  $\text{CLEAR}(x)$ ,  $\text{HANDEEMPTY}$

A<sup>9</sup> :  $\text{HOLDING}(x)$

**putdown**( $x$ ) : アームに持った積み木  $x$  をテーブル上に置く

P :  $\text{HOLDING}(x)$

D :  $\text{HOLDING}(x)$

A :  $\text{ONTABLE}(x)$ ,  $\text{CLEAR}(x)$ ,  $\text{HANDEEMPTY}$

**stack**( $x, y$ ) : アームに持った積み木  $x$  を積み木  $y$  の上に置く

P :  $\text{HOLDING}(x) \wedge \text{CLEAR}(y)$

D :  $\text{HOLDING}(x)$ ,  $\text{CLEAR}(y)$

A :  $\text{HANDEEMPTY}$ ,  $\text{ON}(x, y)$ ,  $\text{CLEAR}(x)$

**unstack**( $x, y$ ) : 積み木  $x$  の上の積み木  $y$  をアームに持つ

P :  $\text{HANDEEMPTY} \wedge \text{ON}(x, y) \wedge \text{CLEAR}(x)$

D :  $\text{HANDEEMPTY}$ ,  $\text{ON}(x, y)$ ,  $\text{CLEAR}(x)$

A :  $\text{HOLDING}(x) \wedge \text{CLEAR}(y)$

---

<sup>7</sup>P : Prerequisite condition

<sup>8</sup>D : Delete condition

<sup>9</sup>A : Add condition

STRIPS のプランニング手法としての特徴は Means-ends analysis (手段-目的解析) にある。これはゴール状態  $G$  と現在の状態  $S$  を比較し、その差を減少させることができるような単位行動を次々と適用させていくというものである。選ばれた単位行動の前提条件が満たされていない場合は、満たされていない部分の実現を中間目標として目標スタックに積み、再帰的に STRIPS を実行することで初期状態からゴール状態まで到達する。アルゴリズムを以下に示す。

---

```
STRIPS( $G$ ) {
  状態  $S$  の要素がゴール状態  $G$  の全ての要素を含むまで以下を繰り返す ;
  {
     $S$  の要素に含まれない  $G$  の要素を  $g$  とする ;
    単位行動のうち、追加リストに  $g$  の要素を含むものを  $f$  とする ;
     $f$  を適用するための前提条件を  $p$  とする ;
    if ( $p$  の全ての要素が  $S$  に含まれていない) then STRIPS( $p$ ) を実行する ;
     $S$  に  $f$  を適用し、 $S$  を書き換える ;
  }
  終了 ;
}
```

---

STRIPS はチェスや積み木問題など、環境の構造が単純で、しかもプランを生成してからロボットがプランを実行し終わるまで、環境（問題空間）が変化しないような状況において効果的な手法である。ところが、実世界を環境とする自律行動主体に対して STRIPS を適用することは、次の 2 つの問題により非常に困難である。

1. 実世界上の問題は構造が複雑なために記述するのが困難であり、これを内部表現に変換する作業の負荷、並びに変換された初期状態からゴール状態への最終経路探索の負荷が大きくプラン生成に膨大な時間を要してしまう。

2. 仮にプランニングが完了して、単位行動系列が得られ、それに基づき行動を開始したとしても、行動中に環境が変化してしまうと得られた単位行動系列が適用できなくなる可能性が出てきてしまう。

つまりプランニングを開始しても、長時間が経過しなければロボットは動作を開始することができない。また、動作を開始してから環境が変化してしまうと得られたプランでは、対応できなくなる場合もある。その時はプランの実行を中断しその時点で再びプランニングを最初から実行することになる。

### A\* アルゴリズム

ゲームや積み木問題において、初期状態からゴール状態に至るまでのすべての到達可能な状態を考え、それをグラフで表現すると、初期状態からゴール状態へのプランとは、初期状態からゴール状態へのグラフ空間における経路とみなすことができる。このように考えると、STRIPS におけるプラン生成過程は探索を行うことと同義である。探索も人工知能研究の重要な研究テーマの一つであり、これまで多くの探索手法が提案されているが、ここではヒューリスティック関数を用いた A\* アルゴリズムについて述べる。A\*[9] はヒューリスティック関数を用いることで探索コストを低く抑えつつも完全性が証明されているアルゴリズムであり、ゴール状態までの最短経路、すなわち最適なプランを求めることができる。

A\* では、節点、有向グラフ、コスト以外に、各節点に結合された、その節点からゴール節点に到達するまでの経路のコスト下限の推定値である  $\hat{h}(n)$  を用いる。 $(\hat{h}(\cdot))$  はヒューリスティック関数と呼ばれる)。例えば、節点を町、矢印のコストが鉄道線路の距離を表すとすると、 $\hat{h}(n)$  は町  $n$  から目標の町までの飛行機路線の距離であるかもしれないし、節点がパズルの位置であるならば、 $\hat{h}(n)$  はパズルが解かれるまでの最小手数かもしれない。以下に A\* のアルゴリズムを示す。

---

### A\* アルゴリズム

1.  $g(n)$  を START からノード  $n$  までの最良解経路のコストとし,  $g(\text{START}) = 0$  とする. OPEN というリストを考え, 最初は START ノードだけを含むような節点のリストとする. ここで  $\hat{h}(\text{START})$  を計算する.
2. 値  $[g(N) + \hat{h}(N)]$  が最小であるようなノード  $N$  を OPEN から選択する. もしも  $N$  がゴールノードならば  $N$  への経路が最良解経路であり, そのコストは  $g(N)$  である. もしも OPEN に節点がなければグラフに解となりうる経路が存在しないことを意味する.
3.  $N$  を OPEN から取り除く.  $N$  の継続節点をすべて見つけこれらを OPEN に追加する. 各継続節点  $S$  に対して  $g(S) = g(N) + (N \text{ から } S \text{ までのコスト})$  とする. ここで  $\hat{h}(S)$  を計算する.
4. ステップ 2 に進む.

---

A\* アルゴリズムの特徴は,  $h(n)$  を  $n$  からゴール節点までの最良解経路の実際のコストとしたとき, もしも全ての  $n$  について,  $\hat{h}(n) \leq h(n)$  であるならば, A\* は必ず最良解を見つけることができる. しかしながら, 例えば 24 パズルの最適解を A\* で求めようとすると, 現在のコンピュータの計算速度であっても膨大な時間を要してしまう. また STRIPS と同様, 解が求まってそれに従って自律行動主体が移動している最中にグラフ構造が変化してしまった場合には, その地点から再び探索を開始しなければならない. 一方, 我々人間は A\* のような複雑な探索行為を行うことなく, 適当にタイルを動かしながら, 決して最適解ではないもののある程度の時間でパズルを解いてしまうことがある. このように, 最適解を求めることを第一とせず, 適度な時間でとりあえず解が求まればよく, しかも環境が途中で変化しても対応できるようなプランニングに関する研究も行われており, 熟考的プランニングとは対照的に「即応的プランニング」と呼ばれている.



## 2.2.2 即応的プランニング

即応的プランニングは「動的環境下でのプランニング」と言い換えることもできる。以下に代表的な手法である、実時間  $A^*$  アルゴリズムとリアクティブプランニングの概要および即応的プランニングの具体例を述べる。

### 実時間 $A^*$ アルゴリズム

実時間  $A^*$  アルゴリズム [10] は、ゴール状態へのプランが得られなくても 1 回のプランニングを短時間で終了し、即実行することを繰り返すことでゴール状態に到達できればよいという方針でプランニングを行う。「1 回のプランニングに要する時間が環境の変化する間隔よりも短ければ、ゴール状態に到達できる可能性がある」という意味で即応性を有する（ただし最適なプランが得られる保証はない）。以下に実時間  $A^*$  アルゴリズムを示す。

---

### 実時間 $A^*$ アルゴリズム

1. 現状態  $x$  から遷移可能な全ての状態  $\{x'\}$  を求める。
2. 遷移可能なすべての状態  $\{x'\}$  に対し、ゴール状態までの推定距離  $f(x')$  を求める。  $f(\cdot)$  は  $A^*$  でも用いられたヒューリスティック関数である。
3. 現状態の推定距離  $f(x)$  を 2 番目に小さい  $(f(x') + d)$  に更新する。ここで  $d$  は現状態  $x$  から  $x'$  までの距離である（この更新方法は同じ状態を繰り返し通過することを防ぐ役割を持つ）。
4. 最小の推定距離を持つ状態  $(\min(f(x') + d))$  となる  $x'$  に実際に遷移する。
5. ステップ 1 に戻る。

---

実時間  $A^*$  アルゴリズムは短時間で次に行う動作を決定することができ、必要な記憶量もプランの長さに対して線形である。条件によっては完全性すなわち解があれば必ず見つけられるという性質があるが、 $A^*$  のような解の最適性は保証されない。また、実時間

$A^*$  アルゴリズムと類似する手法である移動目標探索アルゴリズム (MTS : Moving Target Search) [11][12] はロボットがゴール地点までの移動経路を探索して移動する問題において、ゴール地点そのものが移動してしまう場合でもロボットがゴール地点に到達することを可能とする。そして、ロボットの移動速度がゴールの変化する速度を上回れば、どのような問題空間であっても (有限であれば) 最終的にはロボットはゴールに到達できることを保証する。

### リアクティブプランニング

プランニングをいかに高速に行っても、環境の変化に追いつけないようではこれに対する対応することはできない。例えば、実時間  $A^*$  アルゴリズムにおいて 1 回のプランニングを完了する間に環境が変化してしまうと、実時間  $A^*$  アルゴリズムで移動するロボットであってもゴール地点に到達することはできない。このような場合、事前にいくつかのプランやプランの原型などを用意しておき、プランニングの最中はこれらの中から最適なものを選択することだけを行うことにより、さらなる変化への追従が期待できる。

リアクティブプランニングは、プランの実行時に環境の変化に応じて動的にプランを変更することができるプランニング手法の総称である。SRI<sup>10</sup> インターナショナルにて開発された PRS(Procedural Reasoning System)[13] は、手順的なプランを複数事前に用意しておき、実行中に適切なプランを状況に応じて適宜選択することで動的な変化に対応した。

このように、リアクティブプランニングにおけるプランニングは、これまで述べてきたプランニングのようにプランの生成を行うのではなく、「いくつかの予め用意された代替プランの中から状況に応じて適切なものを選択するプロセス」を意味する。

---

<sup>10</sup>SRI : Stanford Research Institute  
<http://www.sri.com/about>

## 即応プランニングの具体例

即応ルールを用いたプランニングでは、直接実行可能な断片的な行為からなる結論部と、センサ出力から直接判定可能な条件部で記述された即応ルールと呼ばれるルールを用いて、次にとるべき行為の選択を行う。即応プランニングの構成は前述の図 2.9 に示したとおりである。

観測されたセンサ情報が、複数の即応ルールの条件部に直接的に与えられ、それぞれのルールが並列にその条件の適用を判定する。そして、条件が満たされたルールが適切に選択、あるいは統合されて実行される。1つのルールが実行されると、また観測を行うというループが環境の変化に対して十分に短い周期で繰り返される。具体的な即応ルールの例として、壁に沿って移動する壁沿い移動がある。壁を左にして壁沿いをするための即応ルールは、以下のようなになる。

ルール A(壁に近づき過ぎたら離れる)

IF 壁に 5cm 以内に近づいた

THEN 右に 10 度ステアリングを切る。

ルール B(壁から離れすぎたら近づく)

IF 壁から 5cm 以上離れた

THEN 左に 10 度ステアリングを切る。

この 2つの即応ルールが交互に実行されるだけで、角のない壁への壁沿いが可能である。さらに角を壁沿いさせるには、それに対応した即応ルールを追加すればよい。この壁沿い移動において、実行時に壁のゆがみが生じた場合や壁に多少の障害物が置かれた場合などの環境変化に対しても対応できる。しかし、熟考的プランニングで壁沿いの移動を考えた場合、まず環境モデルを構築して、壁沿いを行うための完全なプランを生成してからそのプランを実行する手続きをとることになる。そして、もし実行時に壁のゆがみや障害物により、環境が変化したり、行為の実行が失敗したりした場合には、想定していない状態になるためプランの実行が難しくなる。このような意味で、即応的プランニングは環境の変化に対してロバストであるといわれる。

### 2.2.3 熟考的プランニングと即応的プランニングの統合

実環境への適用を想定したプランニング手法には、熟考的プランニングと即応的プランニングの両方の側面が必要である。熟考性と即応性を組み合わせるアプローチとして、適当な量のプランニングと実行を繰り返すインターリーブ、計算可能時間に応じて解の質を変化させる任意時間アルゴリズム、そして複数のプランニングモジュールを組み合わせる方法の3つの概要を述べる。

#### インターリーブ

プランニングと実行を交互に繰り返しつつ、ゴールを達成するようなプランニングをインターリーブプランニングと呼ぶ。実時間  $A^*$  アルゴリズムはプランニング部が一定時間で終了することが特徴であった。これに対し、熟考的プランニングをインターリーブプランニングに適用することを考えると、いつプランニングを停止して、それまでのプランを実行するかということが課題となり、この問題を解決することで熟考的プランニングに即応性を持たせることができる。山田ら [14] は次のようにプランの成功確率に応じてプランニングを停止させ実行を開始することでこの問題を解決する方法を提案している。まず、STRIPS 的な古典的プランニングによってプランを生成する。そのプランからベイジアンネットワーク [15] を生成し、その上で自律行動主体がいる環境の変化の程度に依存した入力確率を用いて、プランが完全に成功する確率を計算する。環境の観測はプランニングと平行して行われ、前向きビーム探索でプランを展開する。そして各レベルで最適なプランの成功確率が実行閾値より下回ると、その最適なプランの実行が開始される。なお、この閾値は予め与えておく。この手法の特徴は次の通りである。

- プランを実行する閾値を変えることで熟考と即応を調整することができる。
- 成功確率が環境の変化の程度によって変化するので（変化が激しくなると確率が下がる）、実行閾値が一定であっても環境の変化によって熟考が制御される。

この手法の問題点としては、確率を求めるための前処理が必要なことである。例えば、ロボットが未知の環境下で行動しなければならないときには、まず遭遇する可能性のある種々の状況におけるプランの成功確率を予め学習する必要がある、この学習プロセスが完了しないと正しくプランニングを行うことができないことが挙げられる。

## 任意時間アルゴリズム

任意時間アルゴリズムは、プランニングに与えられた時間の長短に関わらず、常にその時点での最善のプランを返すようなアルゴリズムの総称である。任意時間アルゴリズムは 3 つの性質からなる次の anytime 性を満たす。

### anytime 性

- プランの評価：プランの質を明確に評価できる。
- プランの単調性：プランの質が、時間の経過に対して単調に増加する。
- 割り込み可能性：任意のタイミングで有効なプランを返す。

プランニングに使用できる時間が少なく、即応的な反応が求められる場合にもプランを生成でき、十分な時間があれば熟考し、より最適なプランを生成できる任意時間アルゴリズムは、時間制約のある問題領域で多数使用されている [16][17]。任意時間アルゴリズムを実際の問題に適用する例を以下に述べる。

任意時間アルゴリズムは、組み合わせ最適問題として有名な巡回セールスマン問題に適用することができる。あるセールスマンが  $n$  都市を、同じ都市を二度訪ねることなく巡回する場合を考える。また出発した都市と巡回が終了した都市は一致しなければならない。都市  $i$  と  $j$  間の直接移動コストを  $Cost(i; j)$  とするとき、巡回にかかる総コストが最小となるパスを求めるのが巡回セールスマン問題である。巡回セールスマン問題は NP 完全問題として知られており、都市数  $n$  の増加に伴い問題解決が困難となる。図 2.11(a) は 6 つの都市を巡る巡回セールスマン問題で、 $a, b, c, d, e, f$  の順で都市を訪問するパスが得られている状態を示している。ここでこのアルゴリズムが任意に 2 本の都市間リンク  $(c, d)$ ,  $(f, a)$  を選択し、以下の条件 (式 2.3) が成り立つかどうかを確認する。

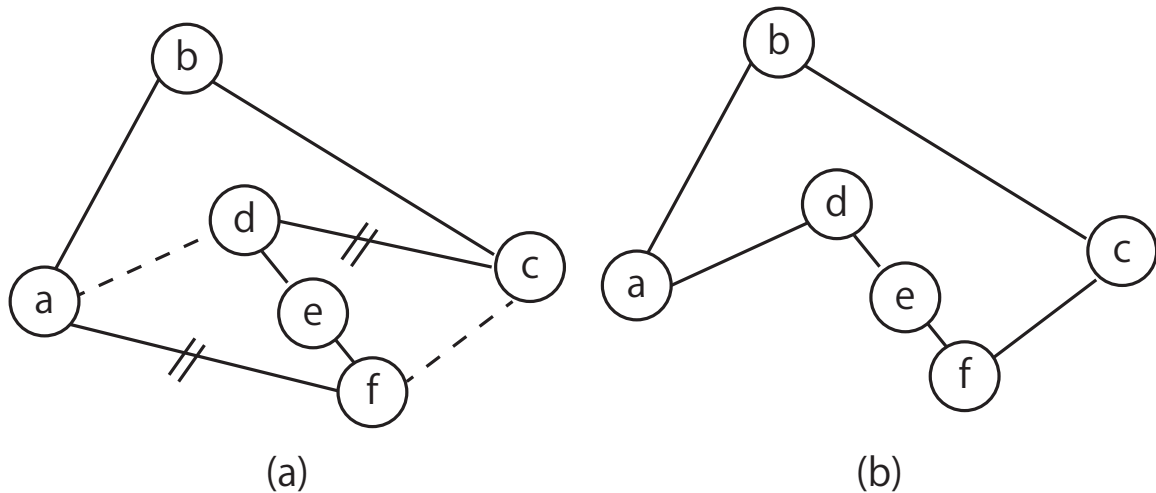


図 2.11: 巡回セールスマン問題

$$Cost(c; f) + Cost(d; a) \leq Cost(c; d) + Cost(f; a) \quad (2.3)$$

もしこの条件が成り立てば、既存のパスを新しいパスに置き換える (図 2.11(b)). この条件が新しいパスが既存のパスよりも低コストであることを保証している.

このアルゴリズムは、すべての都市を 1 回だけたどるツアーをランダムに生成し、過去の最良プランより良いプランが見つかったときに、その最良プランを更新する. そして、停止されたときは、その最良プランを出力する. この場合、解はツアーの長さにより明確に評価でき、最良ツアーの更新により解の単調性を満たし、さらに任意のタイミングで停止可能であり、最短ツアーを近似解として返すこともできるので割り込み可能性も満たすことになる. よってこのアルゴリズムは、任意時間アルゴリズムの 1 つとなっている.

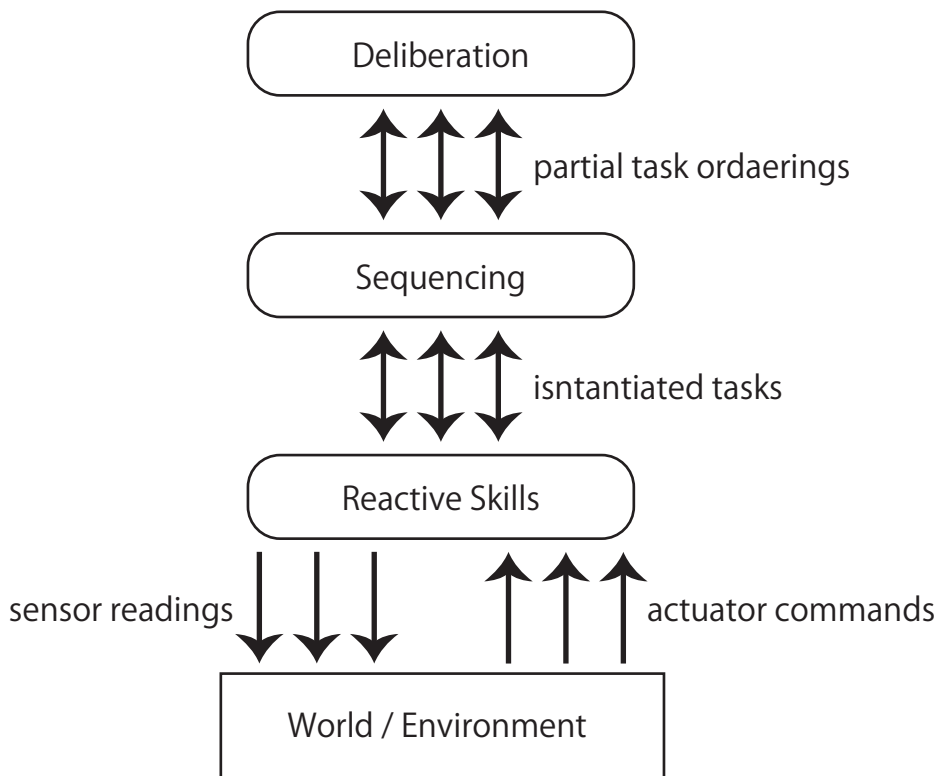


図 2.12: 3T アーキテクチャ

### 複数のモジュールを組み合わせるによる統合

即応的なプランニングを行うモジュールと熟考的なプランニングを行うモジュールという種類の異なるモジュールを組み合わせることで即応と熟考の統合を行うアプローチも数多く提案されている [18]. Bonasso らは複数の層からなるプランニングアーキテクチャ 3T を提案している [19]. 図 2.12 はその概念図である.

Reactive Skills 層は即応的なセンサ-行動ループ (skill と呼ぶ) を制御する. Deliberation 層は熟考的プランニングを行う. Sequencing 層は達成すべき目標を現在の状況を考慮して適切な skill の系列に変換して実行したり, 状況の変化に応じて実行を中止するなどの実行管理を行う. 同様のアーキテクチャとして ATLANTIS[20], NMRA[21] などがある. 栗原ら [22] は 3T のような階層的アーキテクチャには次の 2 つの問題があると指摘し, その問題を回避した上で即応と熟考の統合を可能にする MRR-プランニングを提案している.

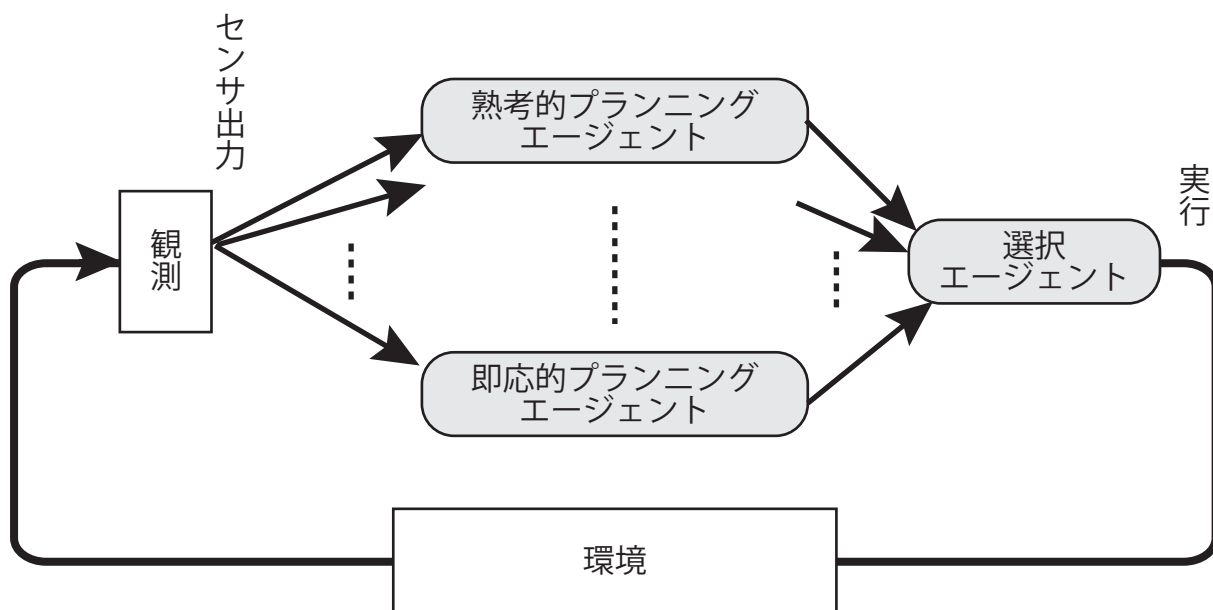


図 2.13: MRR-プランニング

- 各階層間が密接に結びつき複雑な通信を行うため、即応性が低下する可能性がある。
- ある層のモジュールが故障した場合に、その影響が他の層にまで波及し正しく動作できなくなる。

並列動作可能な複数のモジュールから成る MRR-プランニングのアーキテクチャを図 2.13 に示す。観測されたセンサ出力は、複数から成るプランニングモジュールに入力され、その情報に基づき各モジュールは即応的プランニングや熟考的プランニングを行う。なおこれらのモジュールにはどんなプランニングアルゴリズムを用いてもかまわない。各プランニングモジュールで生成されたプランは選択モジュールに渡され、そこで最終的に実行すべき行為を決定する。MRR-プランニングの各モジュールは自分の行動を環境とのインタラクションを通じて自ら決定できる自律行動主体、すなわちエージェントである。したがって各エージェントが自らの目標達成のみを考慮して考えたプラン同士をいかに組み合わせるかが重要となり、MRR-プランニングでは、選択エージェントがその重要な役割を担っている。



## 第3章 関連研究：

### MASにおける協調形態メカニズム

個々のエージェントが自身の問題解決をしつつも、複数のエージェント全体として問題解決を図るシステムをマルチエージェントシステム (MAS : Multi Agent System)[23] と呼ぶ。中でも MAS における協調メカニズムは、大域的視野からトップダウン的に協調動作を行う「中央制御型」、局所的視野を持つ個々のエージェントによるボトムアップ型の協調動作による「直接協調型」と「間接協調型」の3種類に大別できる。それぞれの協調形態について述べ、またそれらの研究に対する本研究の位置づけについて述べる。

#### 3.1 中央制御型

既存のトップダウン的なプランニングには、環境情報をすべて把握した上で、目的を達成させるために全体のプランニングを行う中央制御型のプランニングがある。このタイプでは、矛盾がなく合理的なプランを生成するために、各エージェントが生成したプランを一箇所に集約して協調を行う [24][25][26][27][28][29]。

Georgeff ら [27] や Elkawkagy ら [28] は、個々のエージェントが独立に生成したプランの実行において発生する競合を回避する方法に関して次のような提案をしている。まず STRIPS 形式により、各エージェントが独立にそれぞれの目的についてプランニングを完了する。そして生成された各エージェントのプランは、中央集権的なスケジューラにより競合解消が行われる。競合解消とは、各エージェントのプラン間において2つの行為が並列実行可能か、いずれかが先行すれば実行可能かなどを逐一調べ、プラン全体が安全に実行されるように調整を行うことである。環境情報を踏まえた上で目的を達成するためのプランニングを行うことから作成されるプランの精度は高くなる。しかし、全てのプランニ

### 3.2. 直接協調型

ングを集中的に行うことから計算的な負荷が大きくなり、プランニングに時間がかかる。

Wang ら [29] は、上記のような集中管理の方法のデメリットである計算量の多さを 2.1.2 節で述べたエージェントプログラムのルール部分をシンプルな目的・効用ベースにすることでアルゴリズムの完全性を維持しながら、速さも実現しようとした。しかし、細かいプランニングができないため無駄な動きが非常に多いアルゴリズムであり、動的な環境には適さない方法であった。

## 3.2 直接協調型

既存のボトムアップ的なプランニングには、各エージェントが得られる局所的な情報から独立してプランニングを行い、相互に情報を交換するなどして直接的な方法を用いて協調行動を行う直接協調型のプランニングがある。分散したエージェントがそれぞれにプランニングを行うため、各エージェントにかかる負荷は少なくて済む。そのため、計算負荷の低減をすることが可能である。その一方で、各エージェント同士の相互の通信の情報を集中的に集めてプランニングを行う中央制御型に比べると、少ない局所的な情報でプランニングを実行しなければならないことからプランニングの精度は劣ってしまう。

マルチエージェントプランニング研究における従来研究においては、直接協調型に関する提案が主であった [30]。直接協調型の研究では、各エージェントが生成した環境へのプランを実行に移すまえに、全部もしくは一部の他のエージェントにこれから自分が実行しようとするプランに関連したメッセージをやり取りする。このメッセージ交換は、競合がなくなるまで続けられ、その後プランの実行を行う。

## 3.3 間接協調型

既存のボトムアップ的なプランニングには、各エージェントが得られる局所的な情報から独立してプランニングを行い、共有メモリを用いるなどして間接的な方法を用いて協調行動を行う間接協調型のプランニングがある。直接協調型と同様に、分散したエージェン

## 3.4. 従来研究に対する本研究の位置付け

トがそれぞれにプランニングを行うため、各エージェントにかかる負荷は少なくて済む。そのため、計算負荷の低減をすることが可能である。また、間接的な方法で協調行動を行うため、直接協調型で指摘される通信コストの増加も回避できる。一方で、情報を集中的に集めてプランニングを行う中央制御型や情報を交換して集めてプランニングを行う直接協調型に比べると、更に少ない局所的な情報でプランニングを実行しなければならないことからプランニングの精度は劣ってしまう。

間接協調型の研究には、共有メモリを用いることでお互いのエージェントのプランニングと協調、プラン実行を独立して行うことを可能にし、プランニングの即応性を高めつつ、エージェント同士の協調を保持することを可能にした、次のような方法が提案されている。本田ら [31] は、エージェント同士の協調のために共有メモリ用いた手法を提案しており、直接メッセージを送るような方法よりもエージェント同士の独立性を高めることができ、協調にかかる時間が短縮されプランニング時間を短くすることが可能になった。

## 3.4 従来研究に対する本研究の位置付け

上記3種類の協調形態の特徴をまとめると図3.1のようになる。

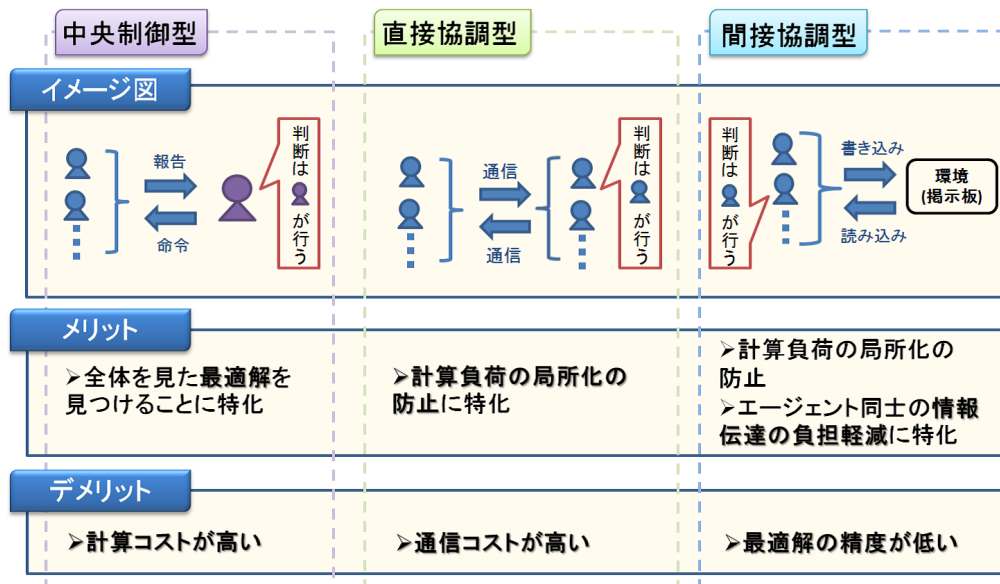


図 3.1: MAS における協調メカニズムの特徴

## 3.4. 従来研究に対する本研究の位置付け

プランニングに関する研究の歴史は深く、前述したような個々の連携形態についての研究は成熟されているといえる。しかし、連携形態そのものを動的に変更する枠組みについての手法はほとんど確立されておらず、本研究ではこれに着目する。

本研究と同様に連携形態そのものを動的に変更する枠組みとしては、中瀬ら [32] の手法がある。中瀬らは、プランニング可能な時間が十分な状況では中央制御型でプランニングを行い、中央制御型のプランニングが間に合わない状況では、間接協調型に切り替える方法を提案している (図 3.2)。この手法では、中央制御型での計算と間接協調型での計算を並列で行い、時々刻々の環境の変化に合わせて、どちらかの計算結果を用いるかを決定している。これは一方で、同じ時刻の同じ空間には、中央制御型と間接協調型のどちらかの計算結果しか採用しないことを示唆している。また、空間規模を大きくした場合、情報量と計算量が多くなり、結局間接協調型のみを選択してしまう。

しかし今後、アンビエントシステムがより多くの場所で構築された場合、大規模空間における実時間制御や局所的に協調形態を変更することが求められる。そこで、より柔軟な協調形態の選択ができる手法を構築することを本研究の位置付けとする。

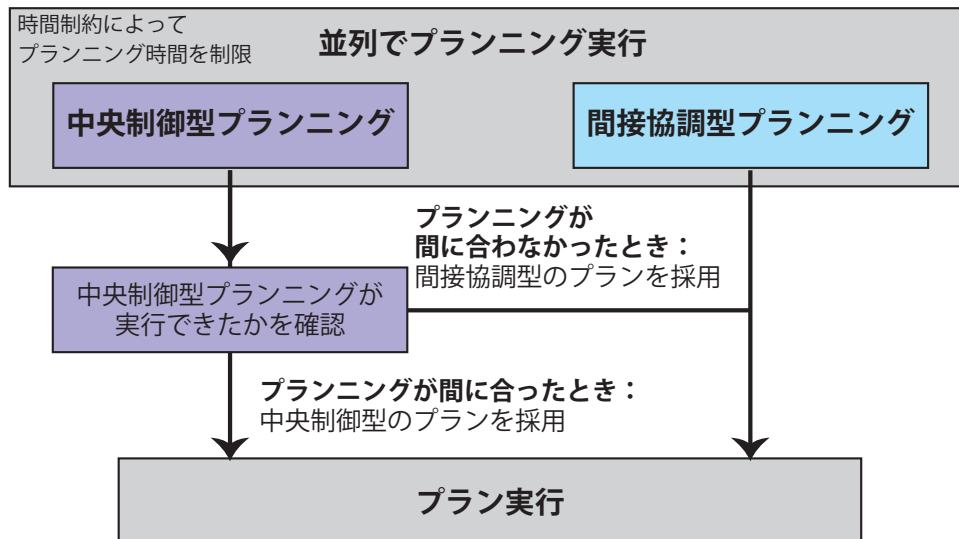


図 3.2: 中央制御型と間接協調型の変更型：アルゴリズム

## 第4章 提案手法：

### マルチエージェントプランニングにおける 動的協調形態変更メカニズム

3.4節にあるように中瀬らは、中央制御型と間接協調型に焦点を当てて協調形態メカニズムの変更型を提案しており、主に2つの協調形態の変更に特化した手法であった(図3.2)。この手法の概略図を図4.1に示す。

これに対して本研究では、空間をより広く考えた場合、同じ時刻の同じ空間においても、中央制御型の計算結果を採用した意思決定と間接協調型の計算結果を採用した意思決定、さらには直接協調型の計算結果を採用した意思決定が共存するといったモデルと提案する。本章では、提案手法であるMASにおける協調形態の共存型モデルについて述べる。



図 4.1: 中央制御型と間接協調型の変更型：概略図

## 4.1 中央制御型と間接協調型の共存型

空間が大規模化することを別の観点から見ると、複数の空間を1つの大規模空間と捉えることができる。このように想定すると、実世界ではそれぞれの空間において異なる協調形態を利用して意思決定をしている場合が考えられる。中瀬らの手法では、このような複数の協調形態が同時に存在している状態に対応することができなかった。そこで、提案手法では、複数の空間に対してそれぞれ中央制御型プランニングと間接協調型プランニングの変更型のアルゴリズムを適応する。以下に、提案手法のアルゴリズム (図 4.2) と概略図 (図 4.3) を示す。

ここでは、中央制御型プランニングに制御範囲を設定する。それぞれの空間において、中央制御型プランニングの制御範囲内に各エージェントがいるのかどうか確認する。制御範囲外の場合、中央制御型プランニングを介さずに間接協調型プランニングのみを実行する。

制御範囲内の場合、変更型のアルゴリズムへ移行する。中央制御型と間接協調型のプランニングを並行で実行させ、与えられた時間制約に従ってプランニングを終了する。中央制御型プランニングが実行できたかどうかを確認する。確認の方法はいろいろあるが、今回の研究では、中央制御型プランニングに制御可能範囲の他に制御可能エージェント数を設けた。この制御可能エージェント数以下ならば中央制御型プランニングの計算結果を採用する。一方で、制御可能エージェント数を超えると間接協調型プランニングの計算結果を採用する。

4.1. 中央制御型と間接協調型の共存型

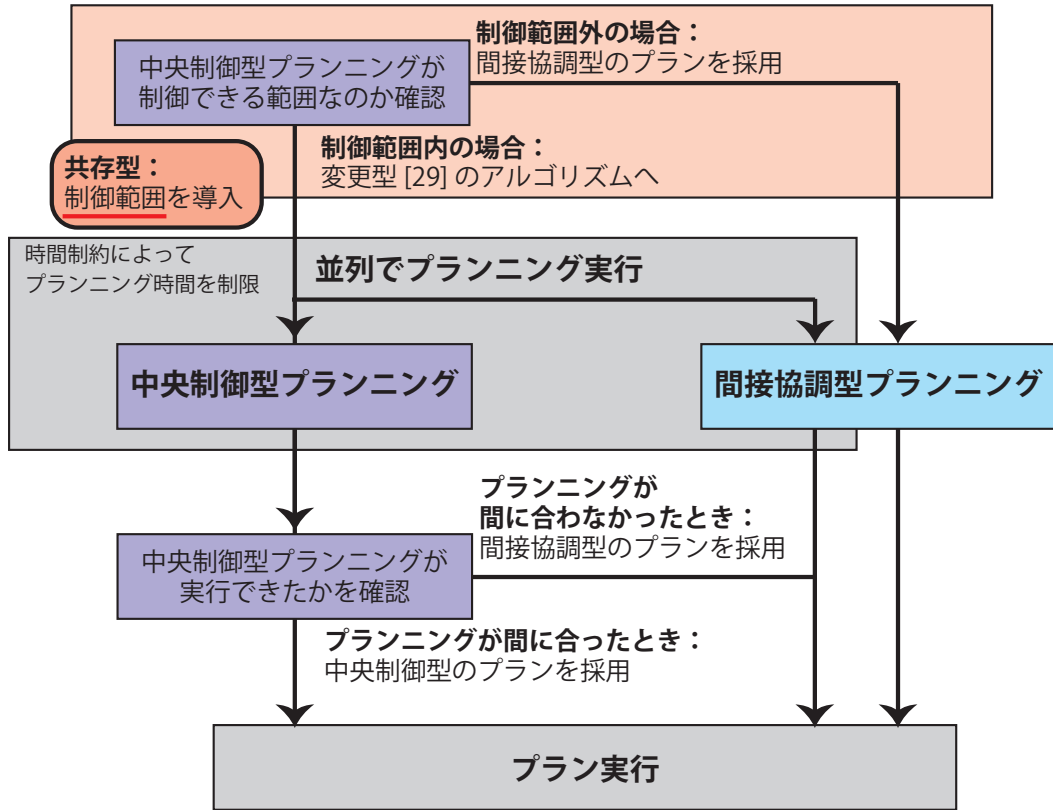


図 4.2: 中央制御型と間接協調型の共存：アルゴリズム

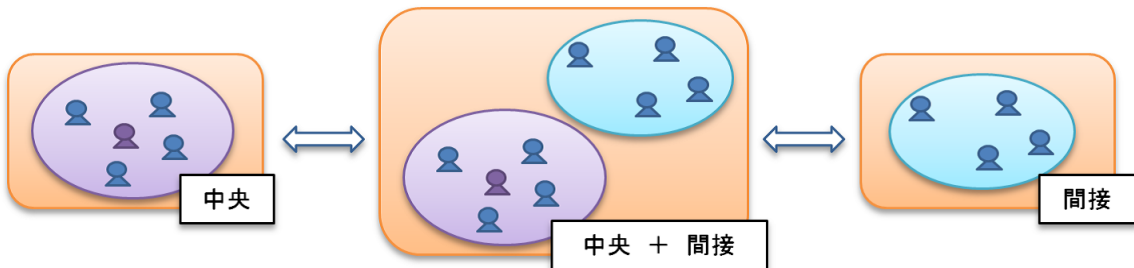


図 4.3: 中央制御型と間接協調型の共存：概略図

## 4.2 直接協調型の追加

また、4.1 節同様に、複数の空間に対して複数の協調形態による意思決定が共存する場合、次のような状況も考えられる。

1. 中央制御型プランニングの制御範囲外かつ、間接協調型プランニングが適応不可能。  
例えば、間接協調型プランニングにおける共有メモリが利用できない状況。
2. 中央制御型プランニングの制御範囲外かつ、エージェント数が非常に少ない。

つまり、エージェント同士の通信コストによる影響がほとんどない状況。

1 の場合は、直接協調型プランニングしか選択肢が残されていない。2 の場合は、間接協調型プランニングよりも直接協調型プランニングを用いたほうが合理的な場合も考えられる。つまり、直接協調型プランニングも含めた、3 つの協調形態が同じ時刻の同じ空間に共存することは十分に考えられる。以下に、1 の場合の提案手法のアルゴリズム (図 4.4) と概略図 (図 4.5) を示す。

ここでは、4.1 節と同様に、中央制御型プランニングに制御範囲を設定する。それぞれの空間において、中央制御型のプランニングの制御範囲内に各エージェントがいるのかどうか確認する。制御範囲外の場合、中央制御型プランニングを介さずに間接協調型プランニングもしくは直接協調型プランニングを実行する。どちらの協調形態を利用するかは、共有メモリを利用できるか否かで決定する。共有メモリを利用できる場合は、間接協調型プランニングを、利用できない場合は、直接協調型プランニングを採用する。

制御範囲内の場合、変更型のアルゴリズムへ移行する。中央制御型と間接協調型のプランニングを並行で実行させ、与えられた時間制約に従ってプランニングを終了する。中央制御型のプランニングが実行できたかどうかを確認する。確認の方法はいろいろあるが、今回の研究では、中央制御型プランニングに制御可能範囲の他に制御可能エージェント数を設けた。この制御可能エージェント数以下ならば中央制御型プランニングの計算結果を採用する。

一方で、制御可能エージェント数を超えると間接協調型プランニングもしくは直接協調型プランニングを実行する。どちらの協調形態を利用するかは、制御範囲外の場合と同様、共有メモリを利用できるか否かで決定する。共有メモリを利用できる場合は、間接協調型プランニングを、利用できない場合は、直接協調型プランニングを採用する。



4.2. 直接協調型の追加

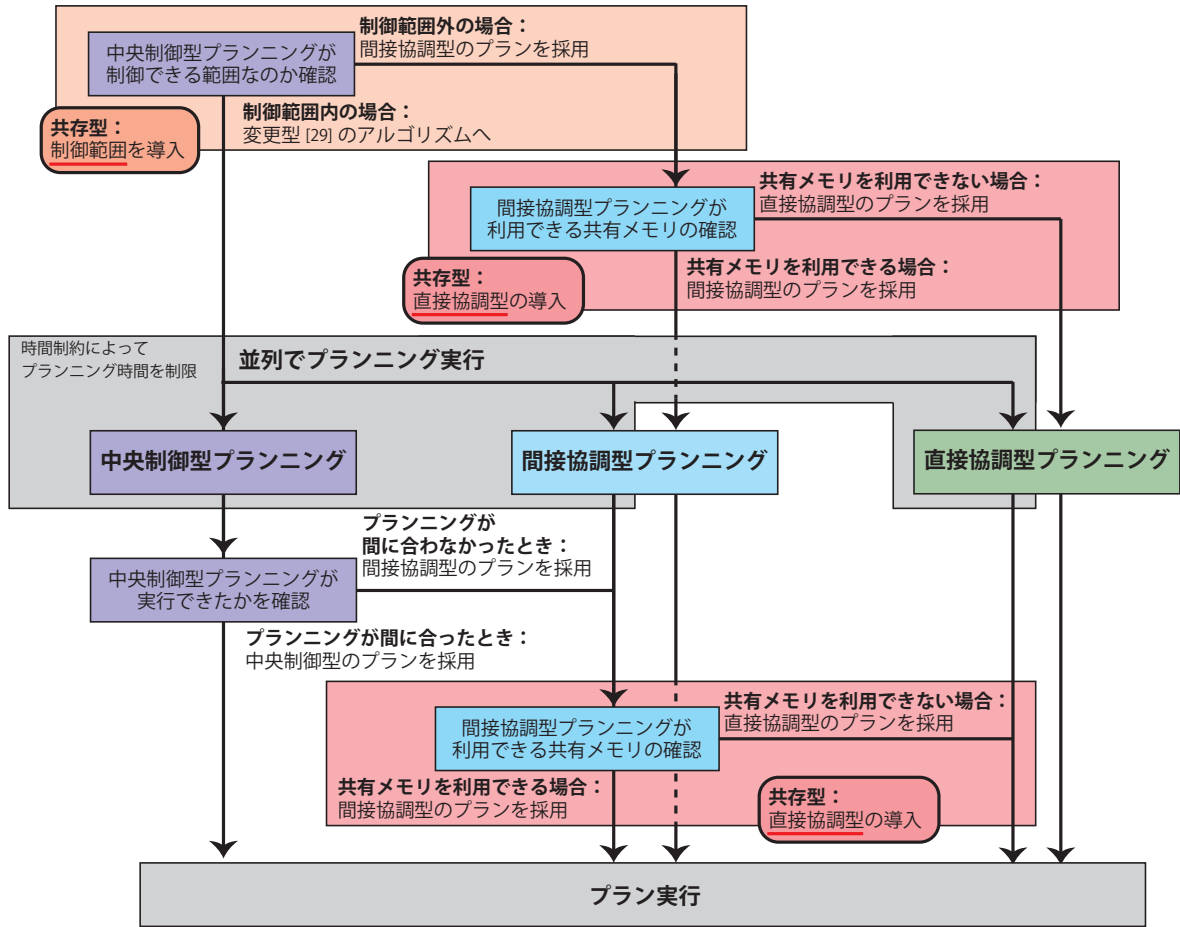


図 4.4: MAS における協調メカニズムの共存：アルゴリズム

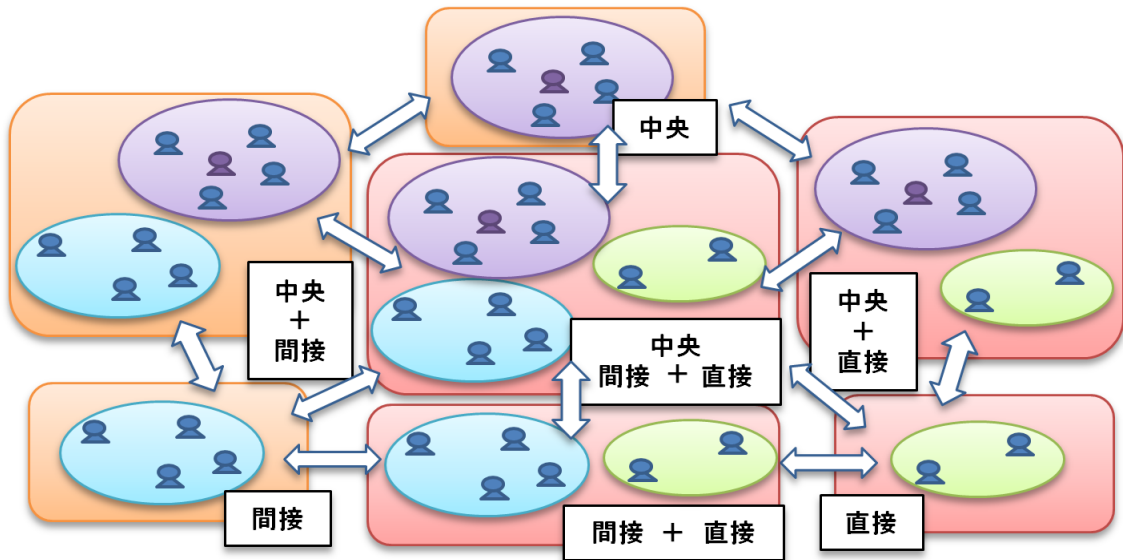


図 4.5: MAS における協調メカニズムの共存：概略図

# 第5章 評価用シミュレーション環境の構築： 中央制御型と間接協調型の共存型

## 5.1 マルチエージェントシミュレーション

マルチ・エージェント・シミュレーションを構築するツールとしては、Repast Agent Simulation Toolkit や swarm などがあるが、本研究では artisoc を使用する。artisoc とは、構造計画研究所<sup>1</sup> が以下の4つの実現をめざし開発したマルチエージェントシミュレータである。

どこでも マルチプラットフォームで動作

簡単に シンプルな操作性，モデル構築の支援

誰にでも 多言語対応，教科書の出版

つながる ネットワーク連携，データベース連携

artisoc における研究として、主に「情報通信」「建築・防災・環境」「製造」「意思決定」の4つの分野を得意としている [33][34][35]。本研究では、マルチエージェントの意思決定を artisoc 上で実装し、プランニングの行為系列を仮想の格子空間状に生成し、競合回避の様子をシミュレーションする。

---

<sup>1</sup><http://mas.kke.co.jp/>

5.2. エージェントの構成

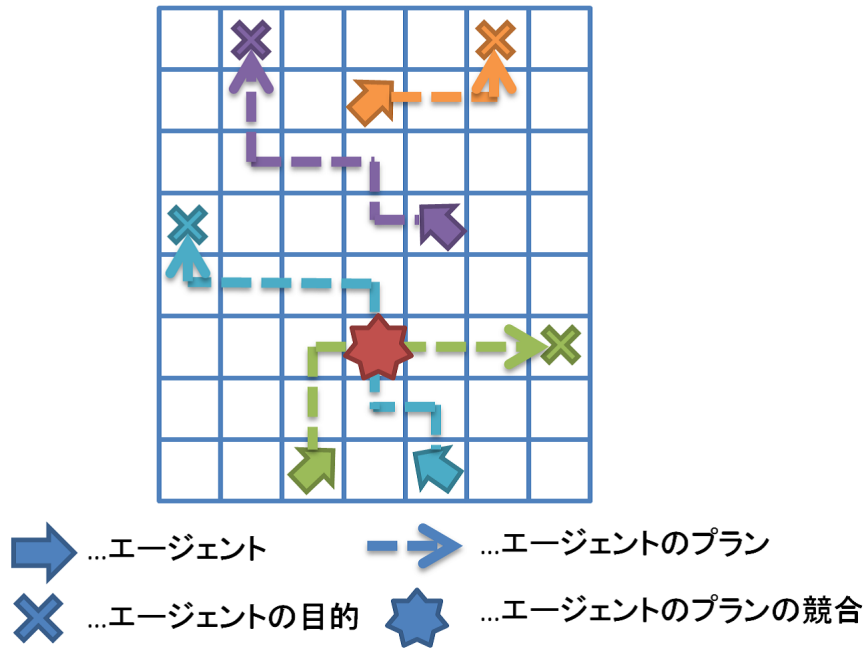


図 5.1: プランニング例

## 5.2 エージェントの構成

実際のアンビエント環境を想定した際に、各エージェントはそれぞれ異なる能力を持っている。例えば、各エージェントは目的によって全く異なる作業を行うため、視野や計算処理能力が異なる。一口に能力といっても様々であることが想定される。

本研究では、目的を達成するためにプランニングを行うエージェントを行動エージェントと定義する。また、各行動エージェントの目的や能力の違いを経路探索の目的地や移動速度、目的達成までのタスクの違いとして表現する（図 5.1）。行動エージェントは、自身の目的を達成するために経路（以後、プランと呼ぶ）を生成する。他の行動エージェントも同様にプランを生成するため、複数のエージェントが同じ時刻において同じタスクをこなさなければならないことが生じる。これを競合と呼び、本研究では、同時に同じ経路を移動することを指す。ある行動エージェントが作成したプランと他の行動エージェントが作成したプランにおいて競合が発生した場合、その競合を回避する必要がある。本研究では、競合回避の方法として中央制御型プランニングと間接協調型プランニングの2つを実装し評価する。

### 5.3. 実験環境

中央制御型プランニングにおいて行動エージェントの代わりに競合回避を図るエージェントをスケジューラエージェント、間接協調型プランニングにおいて共有メモリの役割を果たすものを掲示板と定義する。なおでは、スケジューラエージェントは視野を有している。この視野は制御可能範囲を示している。スケジューラエージェントは視野の範囲内の行動エージェント情報を受け取ることができる。また、一般に中央制御型プランニングは制御するエージェント数が多くなるほどプランニング時間が長くなる。そこで本研究では、プランニング時間に制限を設けるために、スケジューラエージェントには制御可能エージェント数  $\theta$  を設ける。

## 5.3 実験環境

artisoc にてシミュレーション環境を構築し、中央制御型と間接協調型の共存型の振る舞いを検証した。シミュレーションの過程を図 5.2 に示す。また、図 5.3 は、実験環境の一例として、 $20 \times 20$ (400 マス)の環境を用意し、障害物の数を環境に対して 25% (100 マス/400 マス)、行動エージェントを環境に対して 20% (80 マス/400 マス) 配置し、実行中の図である。また、障害物を■、スケジューラエージェントを▲、行動エージェントの位置を●、目的を×で示している。同じ番号の●と×が同一の行動エージェントの初期位置と目的を示している。なお、図 5.3 の上図は、各スケジューラエージェントの制御状態を示している。縦軸はスケジューラエージェントの制御可能範囲内にいる行動エージェントの数を、横軸は実行フェーズまでの試行回数を表している。スケジューラエージェントの制御可能範囲内にいる行動エージェントの数が、制御可能エージェント数  $\theta$  を超えなければ、中央制御型プランニングを、 $\theta$  超えると、間接協調型プランニングを採用している。



5.3. 実験環境

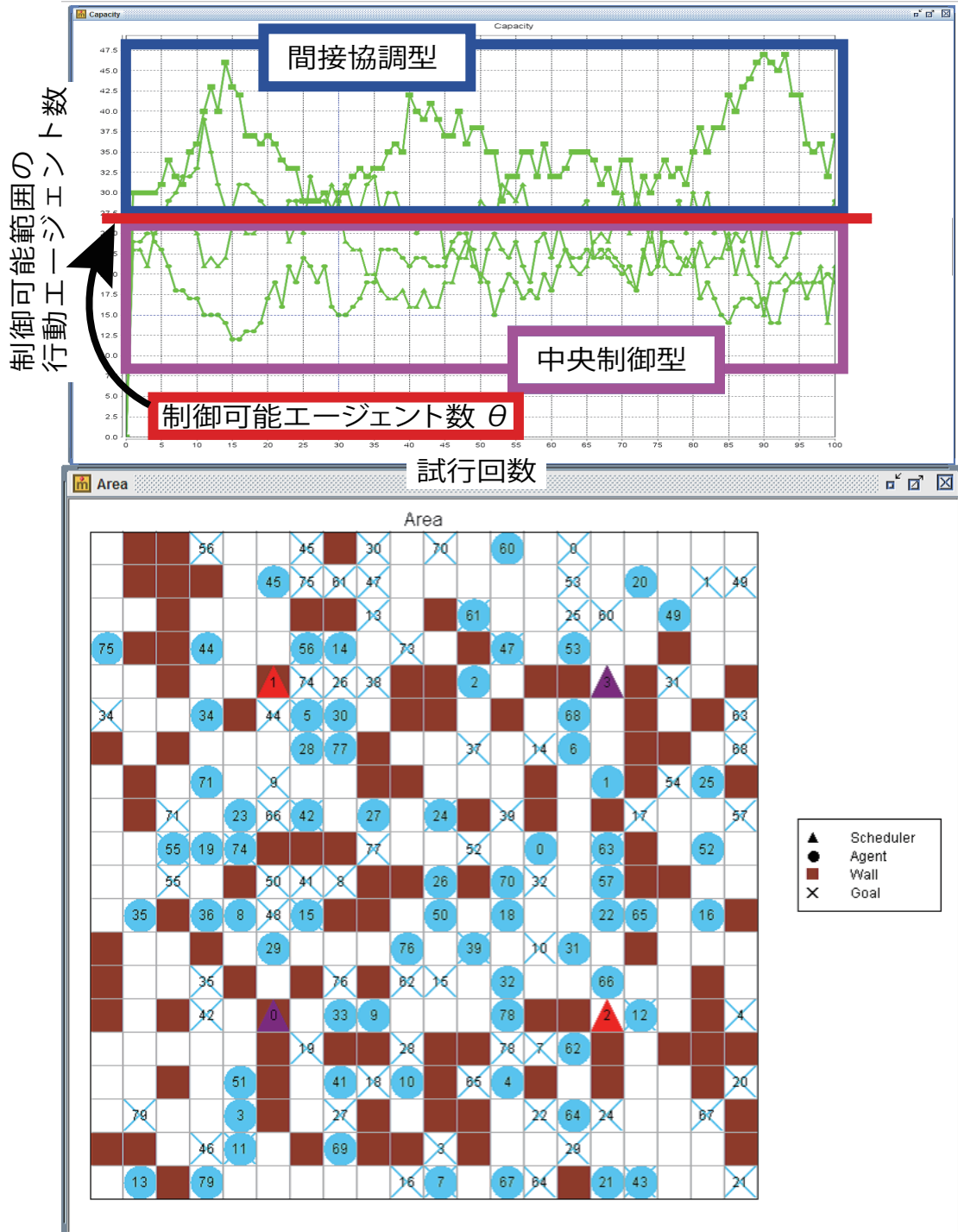


図 5.3: 実行動作例

### 5.3.1 前処理過程

前処理では、行動エージェントおよび動作環境の初期配置を行う。シミュレーションが開始されると、空間内に障害物を生成する。この壁は、空間が同じ大きさであれば他の条件が変化しても固定するように設定した。スケジューラエージェントの位置は任意でできるが、行動エージェント毎に初期位置と目的の位置をランダムに設定した。このとき、障害物、スケジューラエージェント、行動エージェントの初期位置、行動エージェントの目的の位置は同じマスに重複して配置されないようにしておいた。

### 5.3.2 行動エージェントの振る舞い

行動エージェントは図 5.2 に示したように、探索、プランニング選択、競合解消、実行の 4 つのフェーズによって振る舞いを決定する。以下にそれぞれのフェーズについて簡単に説明する。

#### 探索フェーズ

探索フェーズは、行動エージェントの現在の位置と目的の位置を読み込み、プランを生成するフェーズである。プラン作成には 2.2.1 節で述べた A\*探索を用いている。このときにはまだ他の行動エージェントとのプランに競合は解消されていない。プランを生成した後、次に行動する位置とその後の目的に達するまでの行動数を掲示板に書き込む。ここで書き込むその後の目的に達するまでの行動数が競合が生じた際の優先基準となる。

#### プランニング選択フェーズ

プランニング選択フェーズは、自身がスケジューラエージェントの視野内、つまり制御可能範囲内に位置するか否かを読み込み、次の競合解消フェーズにて競合を解消する協調形態を選択するフェーズである。制御可能範囲内であれば、中央制御型プランニングを選択し、制御可能範囲外であれば、間接協調型プランニングを選択する。

5.3. 実験環境

競合解消フェーズ

競合解消フェーズは、プランに競合があった場合に、回避するフェーズである。ここでは、スケジューラエージェントの制御可能エージェント数 =  $\theta$  によって、中央制御型プランニングか間接協調型プランニングのどちらかを選択する (図 5.4)。中央制御型プランニングでは、行動エージェントのプラン全てを比較し、間接協調型プランニングでは、行動エージェントの直後のプランのみを比較し競合を回避している。

中央制御型プランニングの場合、スケジューラエージェントが行動エージェントの現在のプランを読み込み、競合を回避する。

間接協調型プランニングの場合、掲示板の情報を読み込み、プランに競合があった場合、目的に達するまでの行動数を元に競合を回避する。

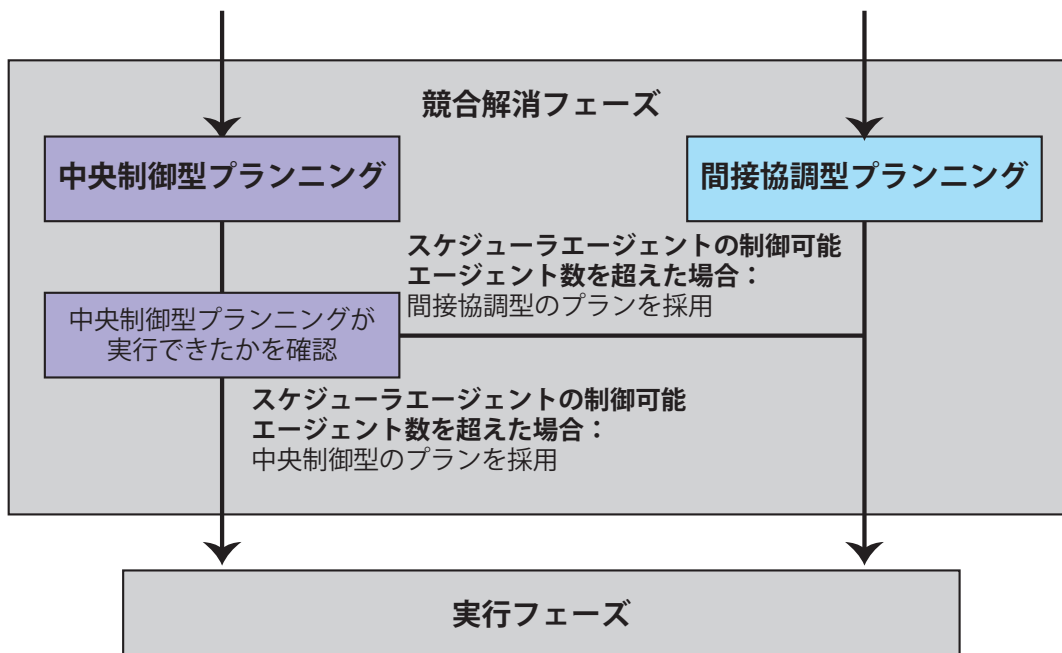


図 5.4: 競合解消フェーズ



### 実行フェーズ

実行フェーズは、自身の現在のプランに従って目的まで推移していくフェーズである。このフェーズの過程を図 5.5 に示す。

まず、設定した試行回数に達している場合はシミュレーションを終了する。設定した試行回数に満たなければ次の処理に移る。

競合検索フェーズにて競合がなかった場合、もしくは間接協調型プランニングで競合はあったが、自身のプランのほうが優先順位が高かった場合、プランを実行する。プラン実行後、中央制御型プランニング、間接協調型プランニングにかかわらず、作成したプランの次のタスクを掲示板に書き込む。プラン実行によって目的を達成していた場合、目的を新たに配置し、探索フェーズに移行する。目的を達成していない場合は、作成したプランは保持したまま、プランニング選択フェーズに移行する。

一方で、競合検索フェーズにて競合があった場合、協調形態によって遷移が異なる。間接協調型プランニングの場合は、優先順位が低いとき、プランニング選択フェーズに移行する。中央制御型プランニングの場合は、スケジューラエージェントからプランを変更するように指示があった場合、探索フェーズに移行する。スケジューラエージェントから指示ない場合、プランを実行することなく、プランニング選択フェーズに移行する。

つまり、中央制御型プランニングの場合、競合生じた際には、競合が生じた位置を制御しているスケジューラエージェントの制御可能範囲内の行動エージェントは、競合が全て解消されるまでプランの実行ができないことを意味する。これは、全体最適を目指す代わりにプランニング時間を要する、中央制御型プランニングの特徴を表している。

5.3. 実験環境

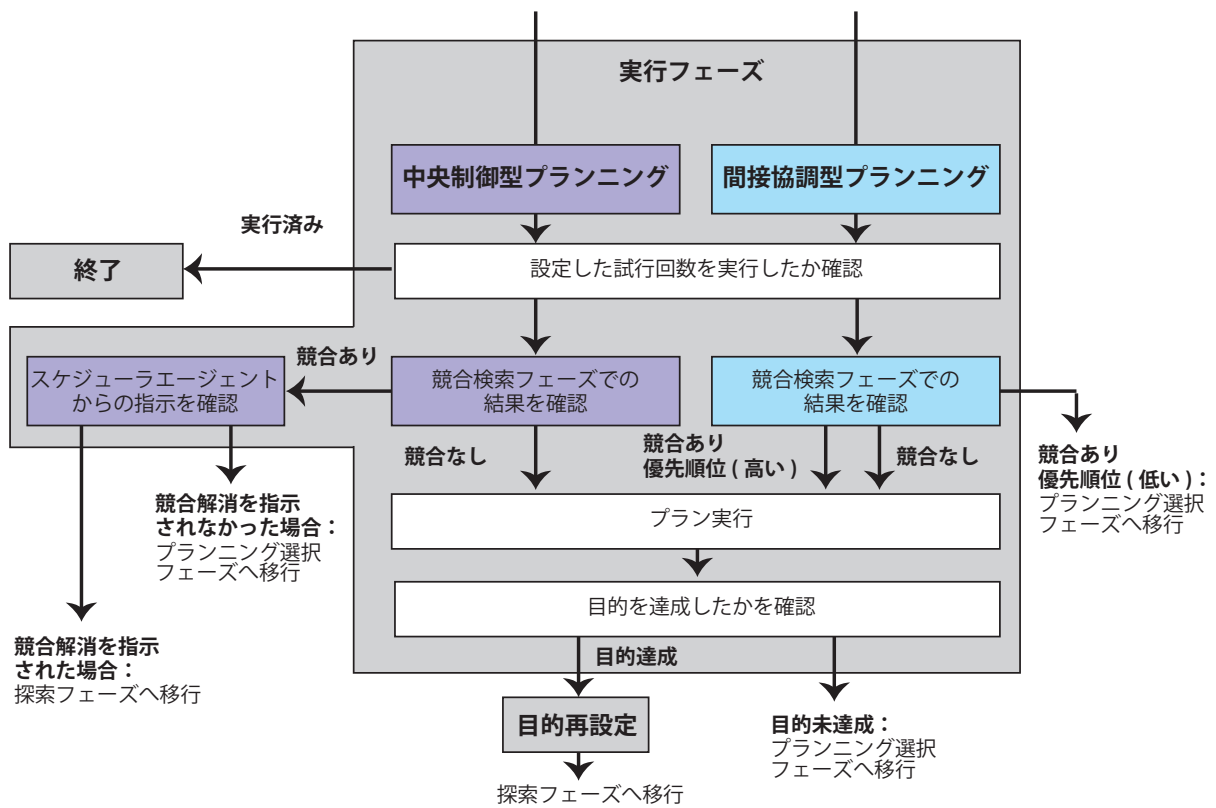


図 5.5: 実行フェーズ

## 第6章 実験と考察：

### 中央制御型と間接協調型の共存型

#### 6.1 実験1：制御可能エージェント数の違いによる

##### 各協調形態の共存の様子

実験環境として、 $20 \times 20$ の環境を用意し、障害物の数を環境に対して25%で固定し、スケジューラエージェントを4つ用意した。そこに、行動エージェントを1, 4, 20, 40, 60, 80, 100つで任意で配置し、スケジューラエージェントが行動エージェントを制御できる値として、制御可能エージェント数 $\theta$ を設け、0, 10, 20, 30, 40で任意に変更し、実行フェーズまでの行動回数を100として実験を行った(表6.1)。なお、行動エージェントは、目的に到達すると、次の目的を生成し、実行終了まで行動し続ける。

表 6.1: 実験環境1

プランニング空間	$20 \times 20$ (うち障害物 25%)
行動エージェント数	1, 4, 20, 40, 60, 80, 100
行動回数	100
スケジューラエージェント数	4
制御可能エージェント数 $\theta$	0, 10, 20, 30, 40

表 6.4 の実験環境の条件に対して、以下の 2 項目の検証をした。

- プランニング精度：目的達成までの理想と実行結果の差
- プランニング実行時間：目的を達成したエージェント数

上記の 2 つの検討項目に対して、表 6.2, 6.3 と図 6.3, 6.4 にまとめた。図 6.3, 6.4 の横軸は、実行フェーズまでの行動回数を示している。そして、縦軸ではそれぞれ、図 6.3 ではプランニング精度の尺度である、最初にプランニングした本来のプランと競合を回避した結果実際に行動したプランとの差を、図 6.4 ではプランニング実行時間の尺度である、行動回数 100 回の間で行動エージェントが目的に到達した回数を示している。

なお、表 6.2～6.3 の **紫色** は行動回数 100 回の間で全て中央制御型と同様の動きを、**水色** は行動回数 100 回の間で全て間接協調型と同様の動きを、**黄色** は行動回数 100 回の間で中央制御型と間接協調型が共存した動きをしている。また、プランニング精度は小さいほど、行動エージェントが目的に到達した回数は多いほど、より良い結果を示していることになる。

表 6.2: 実験 1 : プランニング精度

行動 エージェント	$\theta = 0$	$\theta = 10$	$\theta = 20$	$\theta = 30$	$\theta = 40$
100	673	673	420	783	676
80	400	400	368	345	345
60	86	86	248	257	257
40	107	54	49	49	49
20	12	20	13	13	13
4	1	1	1	1	1
1	0	0	0	0	0

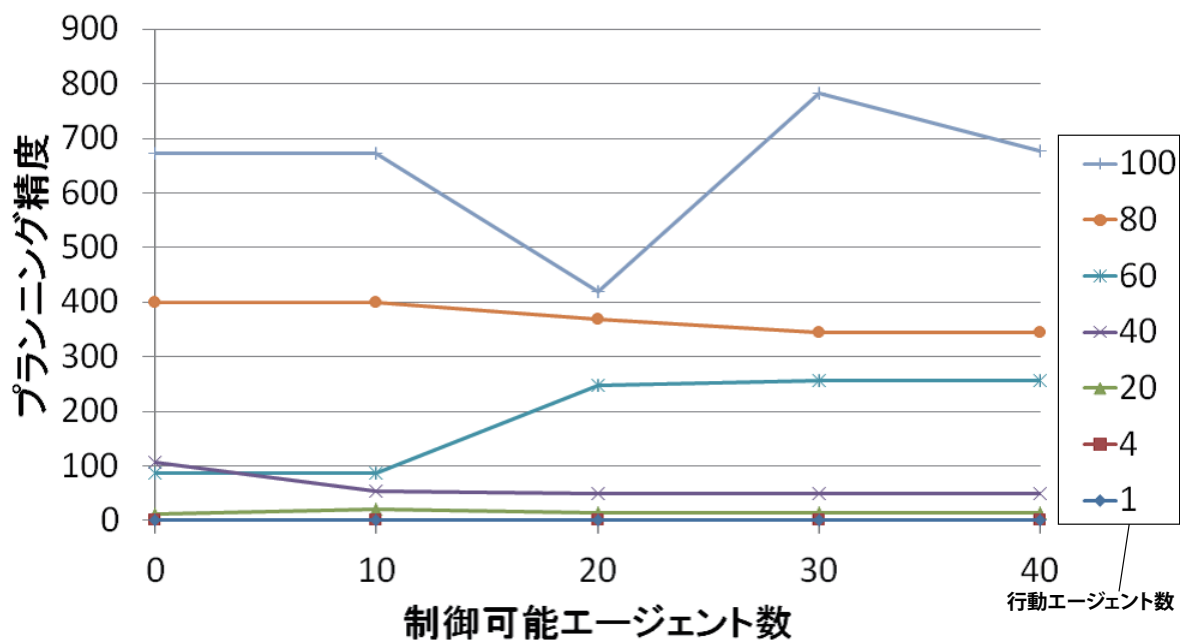


図 6.1: 実験 1 : プランニング精度

表 6.3: 実験 1 : プランニング実行時間

行動 エージェント	$\theta = 0$	$\theta = 10$	$\theta = 20$	$\theta = 30$	$\theta = 40$
100	771	771	769	712	721
80	610	610	600	641	641
60	489	494	444	438	438
40	312	306	306	306	306
20	149	153	154	154	154
4	28	28	28	28	28
1	6	6	6	6	6

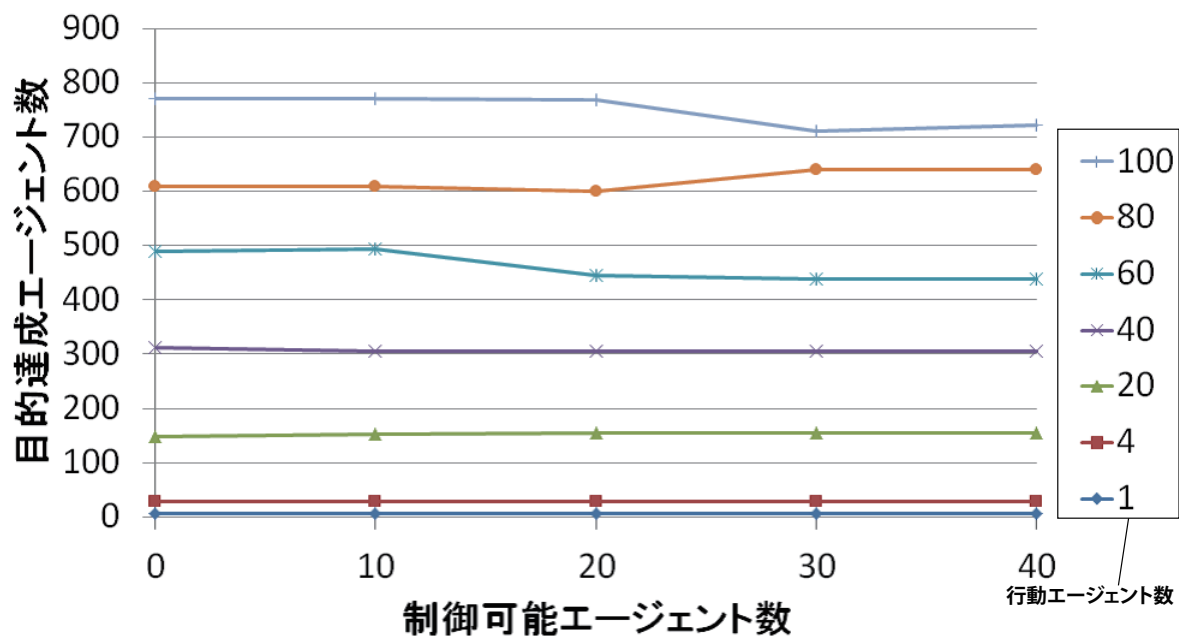


図 6.2: 実験 1 : プランニング実行時間

次に、プランニング精度をみると、スケジューラエージェントが行動エージェントを制御できる値である、制御可能しきい値  $\theta$  が大きくなるほど、つまり中央制御型の割合が大きくなるほど、目的までの理想ステップ数と到達ステップ数の差が小さくなっている。

そして、プランニング実行時間をみると、スケジューラエージェントが行動エージェントを制御できる値である、制御可能しきい値  $\theta$  が小さくなるほど、つまり間接協調型の割合が大きくなるほど、行動完了したエージェント数が増加している。

また、同じ行動エージェント数でみると、**紫色** 部分や **水色** はどの実験結果に対してもスケジューラエージェントが行動エージェントを制御できる値である、制御可能しきい値  $\theta$  に影響を受けていない。一方で、行動エージェント数が 60 と 100 のとき **黄色** が複数存在するが、スケジューラエージェントが行動エージェントを制御できる値である、制御可能しきい値  $\theta$  に大きく影響を受けている。これは、中央制御型と間接協調型が共存した動きをしている場合は、どちらの型の割合が大きいかによって結果に大きな影響を与えていることが考えられる。

そこで、次節に中央制御型と間接協調型の共存型の動きのより詳細な分析をまとめる。

## 6.2 実験 2 : 中央制御型と間接協調型の共存型の分析

実験 1 にて、スケジューラエージェントの制御可能エージェント数の違いにより、中央制御型プランニングと間接協調型プランニングの共存型の協調形態が変化することを確認した。また、中央制御型や間接協調型単独の場合には、行動エージェント数が同じであれば、制御可能エージェント数の違いによるプランニング精度とプランニング時間の変化は見られなかった。しかし、2つの協調形態が共存している場合、行動エージェント数が同じであっても、制御可能エージェント数の違いにより、明確な変化が確認できた。

そこで、行動エージェント数を 100 に固定した際に共存型であった、 $\theta$  の全てに対してより詳細な分析を行った。その結果実験環境は表 6.4 となった。

表 6.4: 実験環境 2

プランニング空間	20 × 20(うち障害物 25 %)
行動エージェント数	100
行動回数	100
スケジューラエージェント数	4
制御可能エージェント数 $\theta$	11~40



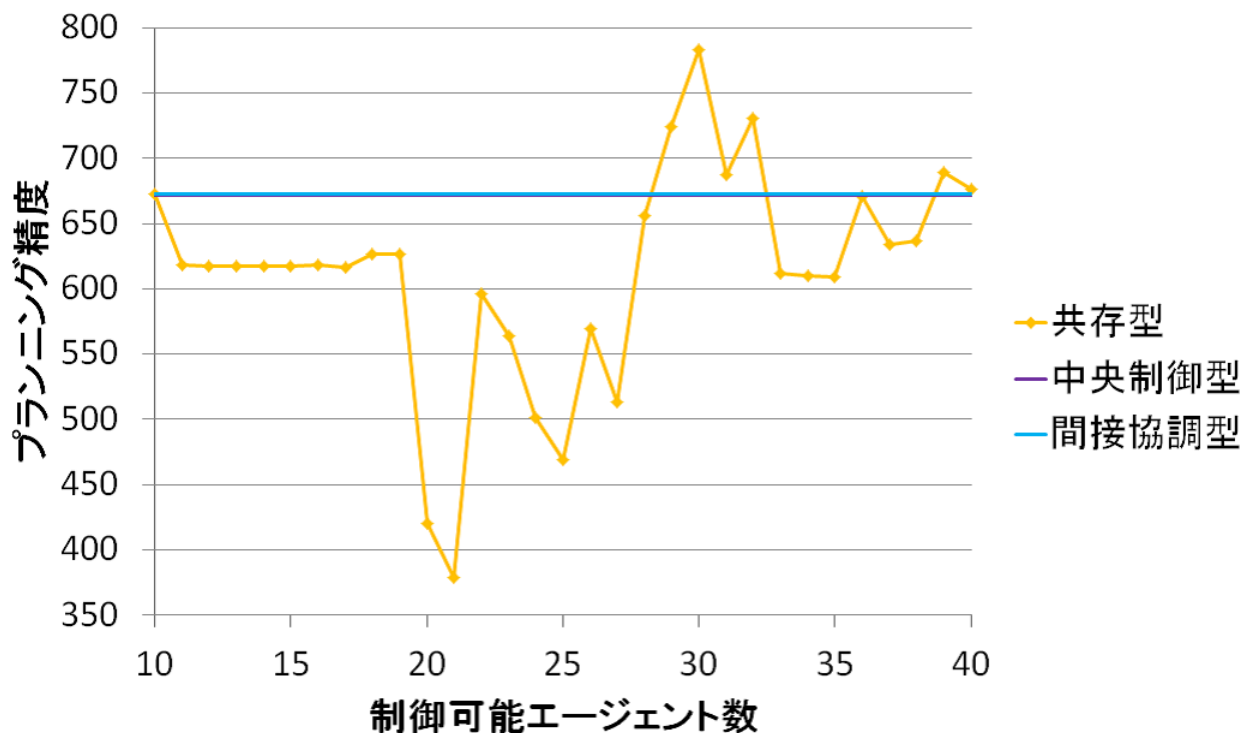


図 6.3: 実験 2 : プランニング精度

### 6.2.1 プランニング精度

プランニング精度の結果を図 6.3 に示す。縦軸のプランニング精度は、行動エージェントが最初にプランニングした本来のプランと競合を回避した結果実際に行動したプランとの差を指している。よって、数値が小さいほうがプランニングの精度は良いといえる。図 6.3 を見ると、表 6.4 の実験環境では、中央制御型と間接協調の違いによるプランニング精度の差は殆どなかった。しかし、共存型にすると、 $\theta$  の違いによって、プランニング精度が大きく異なった。この要因としては、共存型の場合、目的達成エージェントの数が  $\theta$  の違いによって変化することが考えられる。

### 6.2.2 プランニング実行時間

プランニング時間の結果を図 6.4 に示す。縦軸の目的達成エージェント数は、実装した環境ではプランニング実行時間の評価関数といえる。より数値の大きいほうがプランニング実行時間が短いといえる。同時に、競合を即応的に解消したことを示しており、一般的に間接協調型のほうがプランニング実行時間は短くなる。これは、中央制御型の場合、競合生じた際には、競合が生じた位置を制御しているスケジューラエージェントの制御可能範囲内の行動エージェントは、競合が全て解消されるまでプランの実行ができないためである。

共存型における中央制御型と間接協調型の割合は、 $\theta$  が大きいほど直接協調型が、 $\theta$  が小さいほど間接協調型が占める割合が多くなる。プランニング時間の結果に関しては、 $\theta$  が大きいほど直接協調型単独の、 $\theta$  が小さいほど間接協調型単独の結果に近づくと見ることがもできるが、プランニング精度に関しては同様の法則が一概に当てはまるとは言い難い。これは、今回想定している状況が大規模空間になった際の中央制御型の統率を司るエージェントが複数存在する状況であるため、そもそも中央制御型単独の場合でも空間全体の最適解を出すことが困難であり、その際には共存型を導入することでプランニング精度が向上する可能性を示しているのではないかと考えられる。

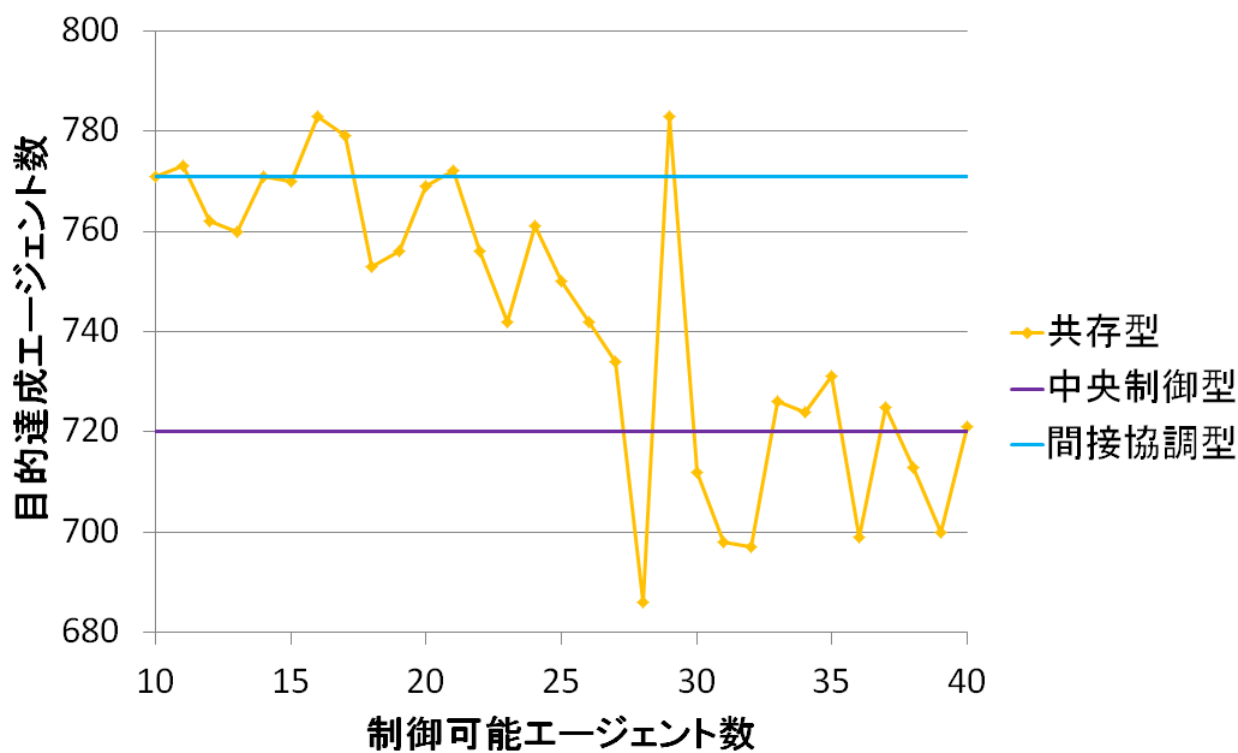


図 6.4: 実験 2 : プランニング実行時間

## 第7章 結論

本研究では、実世界でのデバイス運用のような環境が動的に変化する状況下においたマルチエージェントの協調形態を利用したプランニングを想定する。

実環境では環境の変化に対応するプラン修正能力と実際にプランが行われている中でプランの競合を解消させる即応力を求められる。更に協調形態が複数共存させるモデルを考えた場合、直接協調型がプラン修正能力と即応性に長けた状況も生じると考えられる。

そこで本研究では、これまでの中瀬 [32] らの変更型から共存モデルへの拡張及び、直接協調型の導入も含めた、中央制御型と直接協調型、間接協調型の3つの協調形態が共存するといったモデルを提案した。提案したモデルのうち、中央制御型と間接協調型の共存型に対して、マルチエージェントシミュレーション `artisoc` にて仮想プランニング空間を構築して実験を行った。実験では、プランニング精度とプランニング時間の検証を行い、大規模空間になった際の中央制御型の統率を司るエージェントが複数存在する状況では、共存型を導入することでプランニング精度が向上する可能性を示した。

今後の展望としては、本研究にて提案した直接協調型も含めた3つの協調形態が共存するモデルの実装と評価を行うことが考えられる。

# 謝辞

本研究を行うにあたり，主任指導教員である栗原聡教授には，日頃より熱心なご指導をいただき，多大な御助言を頂いたことを心より感謝いたします。研究に対する取り組み方や考え方など，多くのことを学ばさせて頂きました。

また，指導教員である田中健次教授，大須賀昭彦教授および，社会情報システム学講座の山本佳世子准教授には，学会，および発表会の場で有益なコメントを頂いたことを心より感謝いたします。

大阪大学大学院情報科学研究科の鬼塚真客員教授，奈良先端科学技術大学院大学情報科学研究科の諏訪博彦助教および，社会情報システム学講座の篠田孝祐助教には，研究の方向性について多くの示唆や，具体的な手法についてのアドバイス，多くのご協力を頂き大変有意義な研究生活を送ることができたことを深く感謝いたします。

また，栗原研究室で二年間苦楽を共にし，時には研究のアドバイスや技術的な支援もしていただいた，池田圭佑氏，高橋光紀氏，坪井一晃氏，楡井泰行氏，藤田真康氏には精神的にも支えられました。心より感謝致しています。

本研究は，これらの皆さまの御指導と御協力なしには決して完成し得なかったものであります。ここに心から感謝し御礼申し上げます

## 参考文献

- [1] 谷垣宏一, Chakraborty Goutam, 白鳥則郎, ”自然言語インタフェースに基づいた電子秘書システムの構成”, 情報処理学会研究報告, マルチメディア通信と分散処理研究会報告 95(22), 73-78, 1995.
- [2] 前田利之, 丹羽寿男, 吉田和美, 萱嶋一弘, 前田祐司, ”ネットワーク機能をもつ高齢者福祉ペット型ロボットシステム”, 電子情報通信学会技術研究報告. WIT, 福祉情報工学 102(420), 59-64, 2002.
- [3] 藤沢隆介, 今村光, 松野文俊, ”フェロモン・コミュニケーションによるロボット群の協調運搬”, 情報処理学会, Vol. 2011-MPS-83(9), 1-6, 2011.
- [4] 佐藤知正, ”部屋・家・地域まるごとロボット-ロボット技術の現状/課題と展望”, パナソニック技報, 56(3), 148-153, 2010.
- [5] S.J.Russell (著), P.Norvig (著), 古川康一 (翻訳), ”エージェントアプローチ人工知能 第2版”, 共立出版, pp.32-51, 2008.
- [6] 人工知能学会, ”人工知能学辞典”. 共立出版. 2005.
- [7] P.E. Agre and D. Chapman, ”A implementation of a theory of activity”, In Sixth National Conference on Artificial Intelligence, pp. 268-272, 1987.
- [8] R.E. Fikes and H.J. Nilsson, ”Strips:a new approach to the application of theorem proving to problem solving”, Artificial Intelligence, Vol. 2, pp. 189-208, 1971.
- [9] P. E. Hart, N. J. Nilsson, and B. Raphael, ”A formal basis for the heuristic determination of minimum cost paths”, pp.100-107, 1968.
- [10] Richard E. Korf, ”Real-time heuristic search”, Artificial Intelligence, Vol. 42, pp.189-211, 1990.
- [11] T. Ishida and R.E. Korf, ”Moving target search”, In IJCAI-91, pp. 204-210, 1991.
- [12] T. Ishida, ”Moving target search with intelligence”, In AAAI-92, pp. 525-532, 1992.
- [13] M. P. Georgeff, ”Reactive Resourcing and Planning”, In AAAI-87, pp. 677-682, 1987.

- 
- [14] Yamada, "Controlling deliberation with success probability in dynamic environment", In Third International Conference on Artificial Intelligence Planning Systems (AIPS-96), pp. 251-258, 1996.
- [15] E. Charniak, "Bayesian networks without tears", AI Magazine, Vol. 12, No. 4, pp. 50-63, 1991.
- [16] Shlomo Zilberstein, "Using anytime algorithms in intelligent systems", AI Mag, No. 17, pp. 73-83, 1996.
- [17] O.Sapene and E.Onaindia, "Planning in highly dynamic environments: an anytime approach for planning under time constraints", Applied Intelligence, 2007.
- [18] 三浦純, "ロボットにおけるプランニング", 人工知能学会誌, Vol. 16, No. 5, pp. 1-6, 2001.
- [19] R. Bonasso, R. Firby, and E. Gat, "Experiences with an architecture for intelligent, reactive agents", Experimental and Theoretical Artificial Intelligence, Vol. 9, No. 2, 1997.
- [20] E. Gat, "Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots", In AAAI-92, pp. 809-815, 1992.
- [21] B. Pell, D. Bernard, and S. Chien, "An autonomous spacecraft agent prototype", In Agent-97, pp. 809-815, 1997.
- [22] Satoshi Kurihara, Shigemi Aoyagi, Rikio Onai, and Toshiharu Sugawara, "Adaptive selection of reactive/deliberative planning for a dynamic environment", Robotics and autonomous systems, Vol. 24, No. 3-4, 1998.
- [23] G. ed. Weiss, "A Modern Approach to Distributed Artificial Intelligence", The MIT Press, 1999.
- [24] J. Cox and E. Durfee, "Discovering and exploiting synergy between hierarchical planning agents", In Proceedings of the Second International Joint Conference on Autonomous Agents and Multi agent Systems, pp.281-288, 2003.
- [25] Jeffrey S. Cox and Edmund H. Durfee, "An efficient algorithm for multiagent plan coordination", Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-06), pp.828-835, 2006.
- [26] E. Ephrati and J.S. Rosenschein, "Divide and conquer in multi-agent planning", In Proceedings of the twelfth National Conference on Artificial Intelligence, pp.375-380, 1994.

- 
- [27] M. P. Georgeff, "Communication and interaction in multi-agent planning", the Third National Conference on Artificial Intelligence, pp.125-129, 1983.
- [28] Mohamed Elkawagy and Susanne Biundo, "Hybrid Multi-agent Planning", Multiagent System Technologies 9th German Conference, MATES 2011, Berlin, Germany, October 6-7, 2011. Proceedings, pp.16-28, 2011.
- [29] Ko-Hsin Cindy Wang and Adi Botea, "MAPP: a Scalable Multi-Agent Path Planning Algorithm with Tractability and Completeness Guarantees", Journal of Artificial Intelligence Research 42, pp.55-90, 2011.
- [30] Jeffrey S. Cox and Edmund H. Durfee, "An efficient algorithm for multiagent plan coordination", Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-05), pp.828-835, 2005.
- [31] 本田誠一, 森山甲一, 沼尾正行, 栗原聡, "多ゴールが非同期に発生する環境下における実時間リアクティブプランニング法の研究", 人工知能学会全国大会 (第 22 回), 2008.
- [32] 中瀬絢哉, 沼尾正行, 栗原 聡, "マルチエージェントプランニングにおける協調形態の動的変更メカニズムの提案", (WSSIT-13), 2013.
- [33] 野田茂, 土居和加奈, "artisoc を用いた介護保険施設での避難行動シミュレーション", 土木学会第 64 回年次学術講演会, 2009.
- [34] 齊藤 たつき, "マルチエージェント・シミュレータを活用した教育—人工社会の基礎的演習・実験—", 情報科学 第 27 号 (別刷), 2007.
- [35] 浅見 拓哉, 関 良明, "テーマパーク問題における利用者行動シミュレーションを用いた誘導法に関する考察", 第 14 回社会情報システム学シンポジウム, 2008.



# 研究業績

## 研究会

- (1) 布施太章, 篠田考祐, 諏訪博彦, 栗原聡, ”マルチエージェントプランニングにおける動的協調形態共存メカニズムの評価”, 社会システムと情報技術研究ウィーク in ルスツリゾート, 北海道, 2015年3月
- (2) 布施太章, 篠田考祐, 諏訪博彦, 栗原聡, ”マルチエージェントプランニングにおける動的協調形態共存メカニズムに関する一考察 -artisocを用いた実装と評価-”, 社会情報システム学シンポジウム, 東京, 2015年1月
- (3) 布施太章, 篠田考祐, 諏訪博彦, 栗原聡, ”マルチエージェントプランニングにおける協調形態の変更メカニズムの検討”, 社会システムと情報技術研究ウィーク in ニセコ・甘露の森, 北海道, 2014年3月
- (4) 布施太章, 諏訪博彦, 栗原聡, ”環境の動的変化に応じて協調形態を柔軟に変化させるマルチエージェントプランニングの検討”, 情報処理学会知能システム研究会, 東京, 2014年1月

## 大会

- (1) 布施太章, 篠田考祐, 諏訪博彦, 栗原聡, ”マルチエージェントプランニングにおける協調形態の動的共存メカニズムの提案”, 人工知能学会全国大会, 愛媛, 2014年5月

## ポスター発表

- (1) 布施太章, 篠田考祐, 諏訪博彦, 栗原聡, ”マルチエージェントプランニングにおける動的協調形態変更メカニズムの検討”, Joint Agent Workshops and Symposium, 宮崎, 2014年10月
- (2) 池田圭佑, 諏訪博彦, 栗原聡, ”環境変化に応じたマルチエージェントの協調に関する研究”, Joint Agent Workshops and Symposium, 和歌山, 2013年9月

# 付 録 A 中央制御型と間接協調型の共存型 競合回避の詳細

## 具体的な動作例

### 【中央制御型】

行動エージェントがスケジューラエージェントの視野内に存在し、スケジューラエージェントの処理速度が十分に保たれる場合、その範囲内の行動エージェントは中央制御型のルールにて行動する。中央制御型では、行動エージェントではなくスケジューラエージェントがプランの生成を行う。ここで、スケジューラエージェントは中央制御型のエリアの数だけ存在する。そのため、環境全てを中央制御型で制御できる場合は、スケジューラエージェントの数は1つとなる。

まず、スケジューラエージェントは、制御している範囲の環境全てを読み込み、範囲内のエージェント全ての目的までのプランを生成する。その後、範囲内のエージェント全ての競合状態を調べ、競合解消を行う。この競合が全て解消され次第、行動エージェントにプランの実行を指示し、行動エージェントがプランを実行する。

### 【間接協調型】

行動エージェントがスケジューラエージェントの視野内に存在するが、スケジューラエージェントの処理速度が十分に保たれない場合、または、スケジューラエージェントの視野内に存在しなかった場合、その範囲内の行動エージェントは間接協調型または直接協調型のルールにて行動する。さらに、行動エージェント自身の視野内に外の行動エージェントが一定数以上の場合、間接協調型のルールにて行動する。間接協調型では、行動エージェント自身がプランの生成を行う。また、掲示板エージェントは間接協調型のエリアの数だけ存在する。そのため、環境全てを間接協調型で制御できる場合は、掲示板エージェントの数は1つとなる。

まず、行動エージェントは、視野内の自身に関する環境を読み込み、自身の目的までのプランを生成する。その後、他の行動エージェントと共有の掲示板エージェントに対して、自身のプランの書き込みと他の行動エージェントのプランの読み込みを行い、競合解消を行う。自身に関連する競合が解消され次第、それぞれの行動エージェントがプランを実行する。

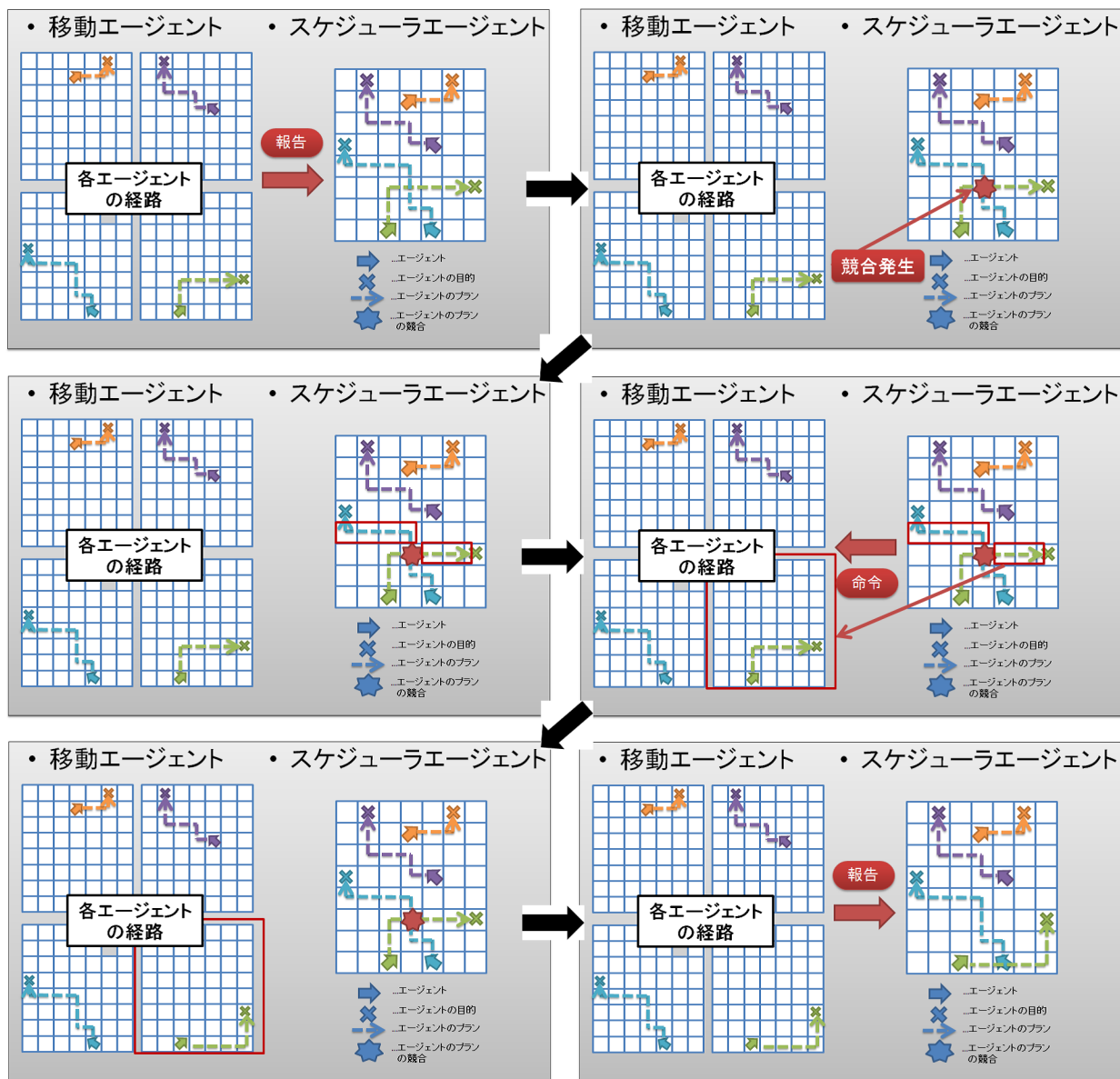


図 A.1: 中央制御型プランニングの競合回避の例

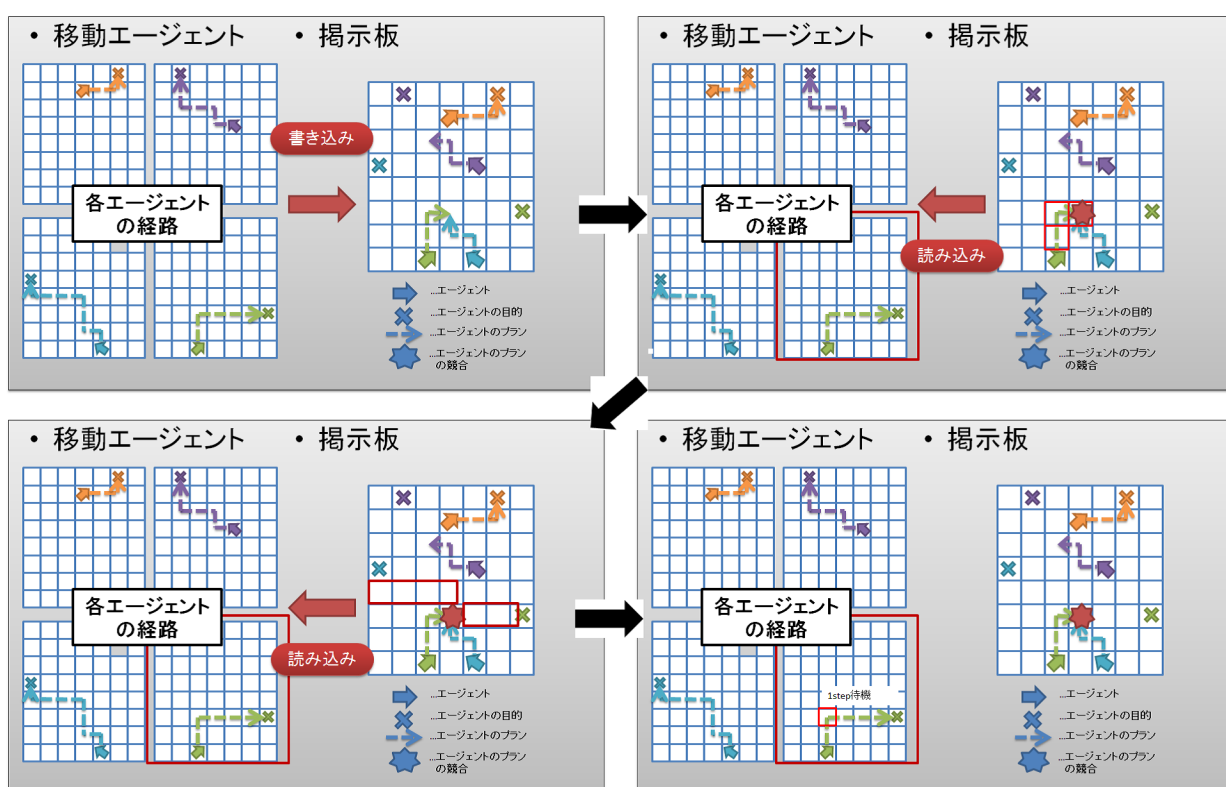


図 A.2: 間接協調型プランニングの競合回避の例

# 付録B 実験2： 中央制御型と間接協調型の共存型の分析 artisoc 実行画面

## 行動エージェントと $\theta$ の関係図

6.2の表6.4の条件で行った、行動エージェント数と $\theta$ の関係図のartisocでの実行画面を図B.2～図B.4に示す。各グラフの見方の例を図B.1に示す。縦軸はスケジューラエージェントの制御可能範囲内にある行動エージェントの数を、横軸は実行フェーズまでの試行回数を表している。スケジューラエージェントの制御可能範囲内にある行動エージェントの数が、制御可能エージェント数 $\theta$ を超えなければ、中央制御型プランニングを、 $\theta$ を超えると、間接協調型プランニングを採用している。

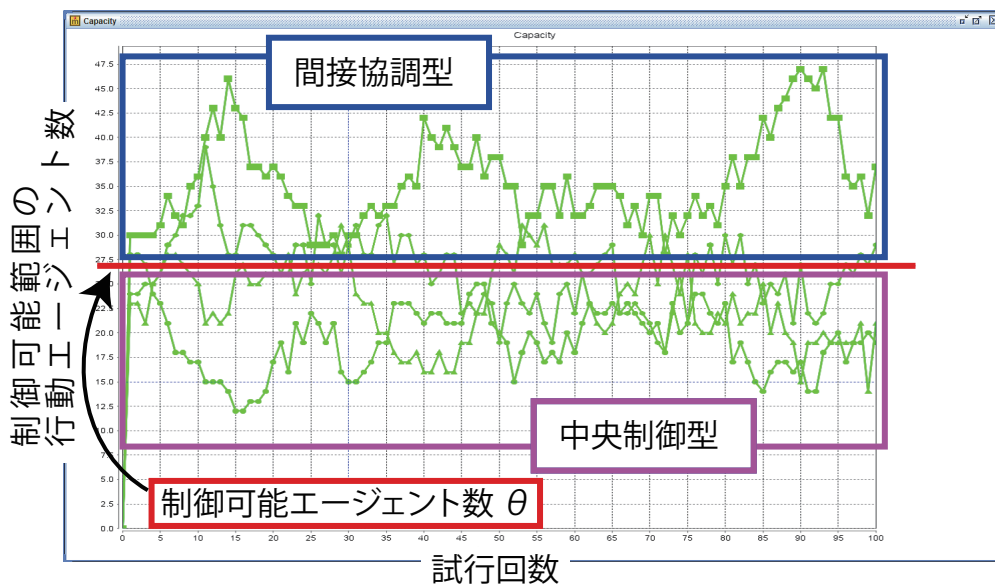


図 B.1: artisoc 実行画面：行動エージェント数と $\theta$ の関係図

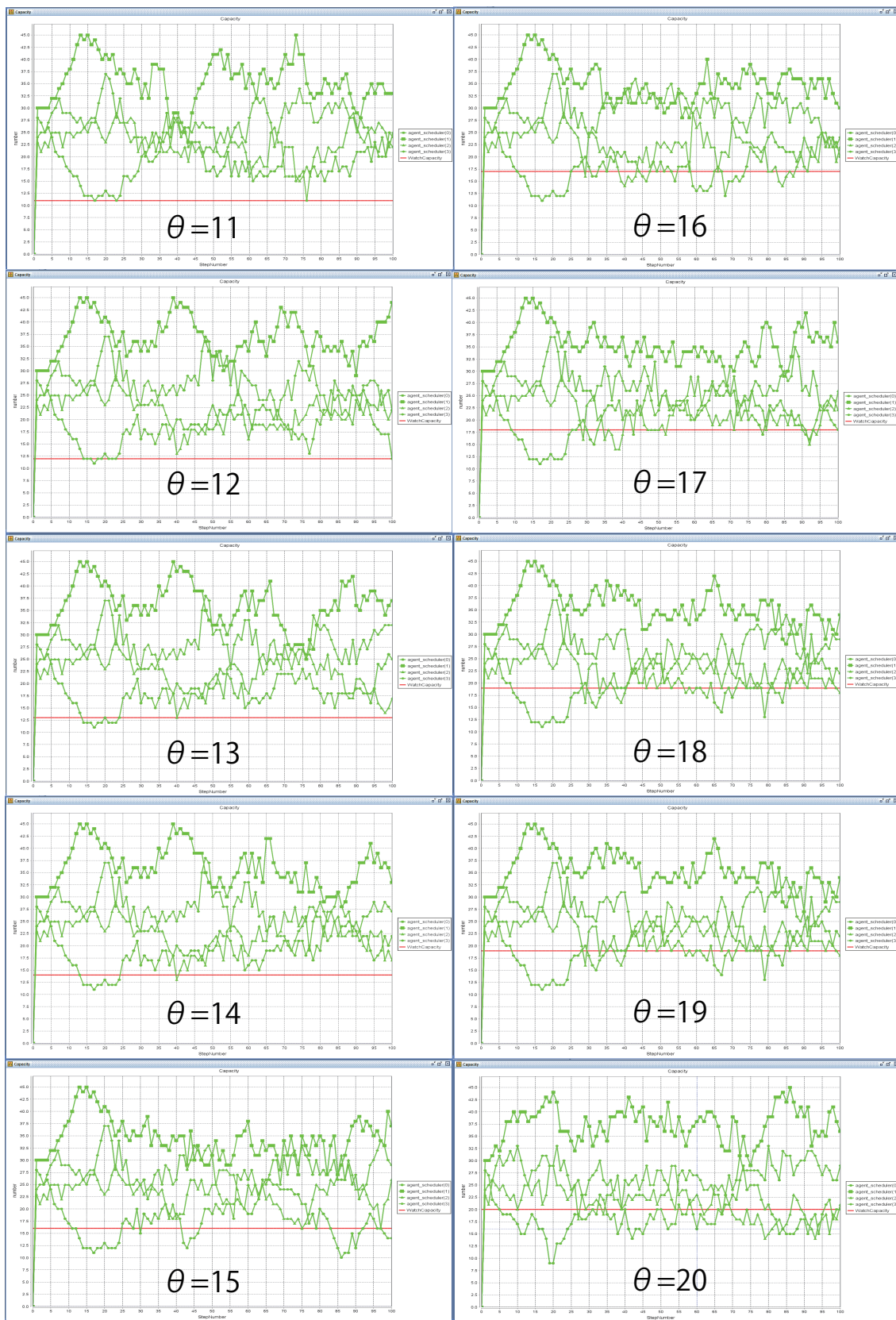


図 B.2: artisoc 実行画面 :  $\theta = 11 \sim 20$

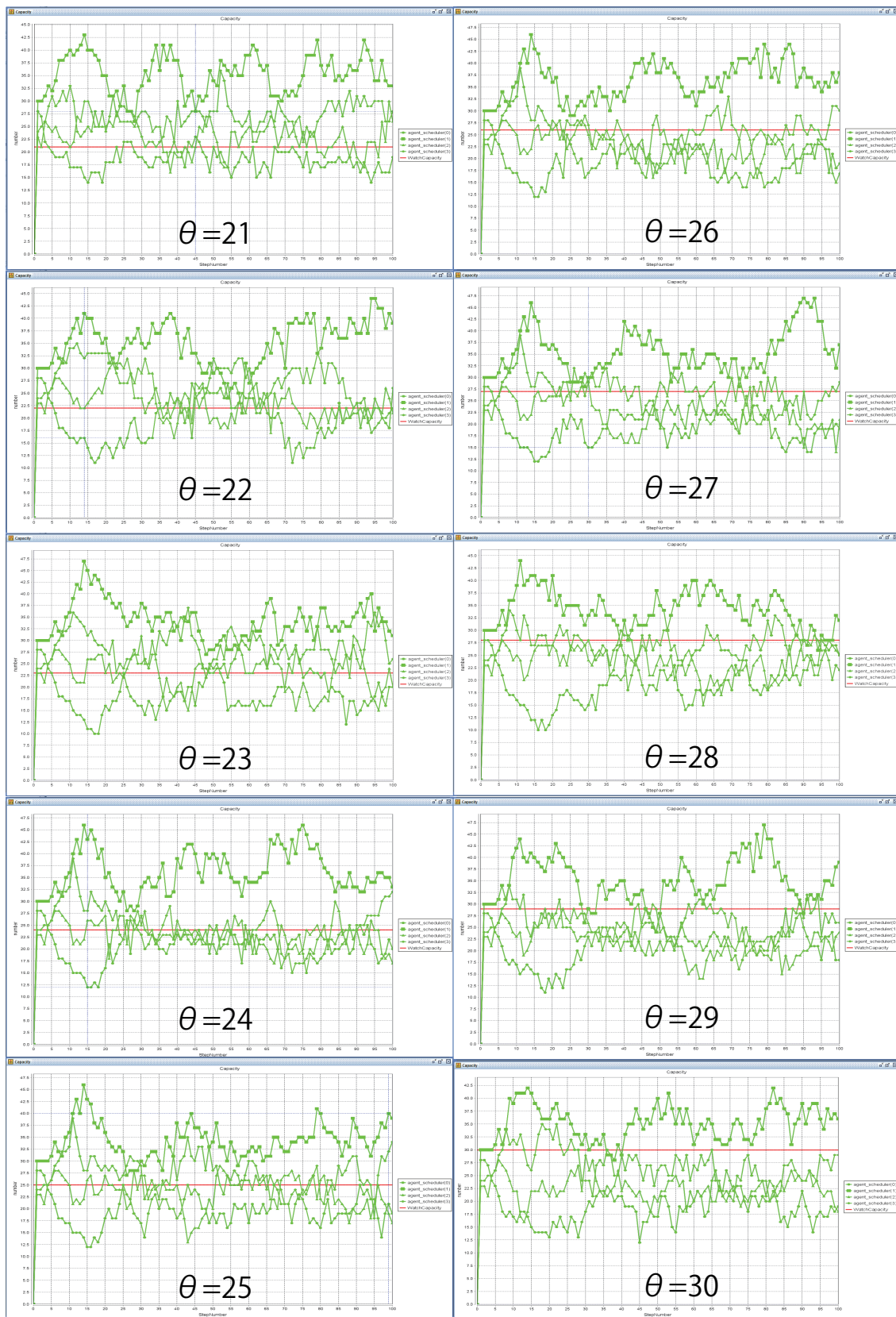


図 B.3: artisoc 実行画面 :  $\theta = 21 \sim 30$

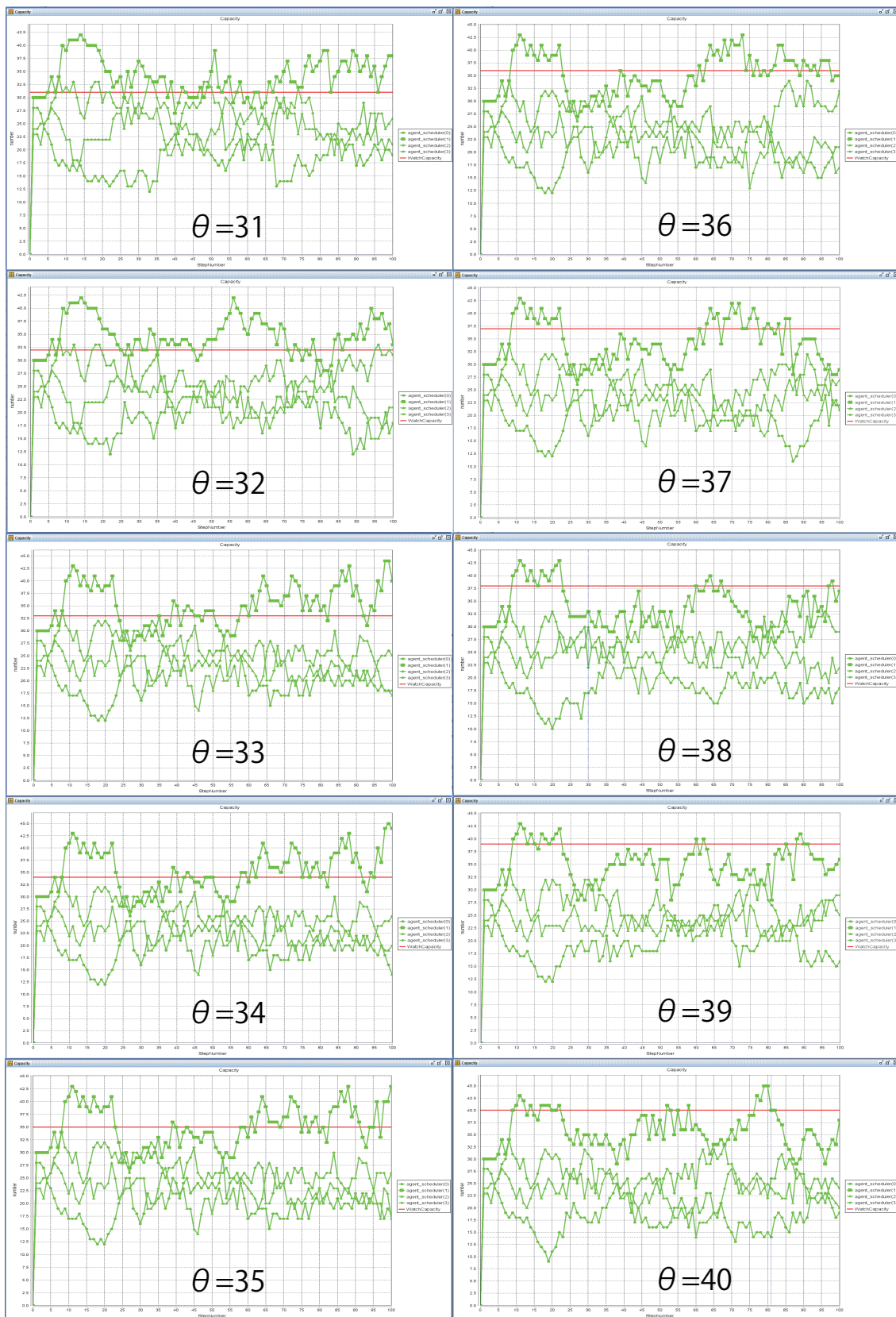


図 B.4: artisoc 実行画面 :  $\theta = 31 \sim 40$