

Title	共有認知空間を用いたエージェントの協調探索
Author(s)	木崎, 徳次郎
Citation	
Issue Date	2002-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/1568">http://hdl.handle.net/10119/1568</a>
Rights	
Description	Supervisor:東条 敏, 情報科学研究科, 修士

# 修士論文

## 共有認知空間を用いた エージェントの協調探索

指導教官 東条敏 教授

北陸先端科学技術大学院大学  
情報科学研究科情報処理学専攻

木崎徳次郎

2002年2月15日

## 要旨

迷路に代表されるような探索問題に対してロボットを用いて探索を行うことを考えると、探索空間が大きな場合、一台での探索では探索に時間がかかり過ぎる。また、必要以上に多くの台数を用いることも連絡コストの増大や、多くのロボットを用意することのコストの増大などの点で問題がある。そこで本研究では、エージェントを少数使うことを前提として、協調探索を効果的に行うことを考える。

複数のエージェントの協調探索では、エージェントが共有する認知空間によって差異が生ずることが知られている。認知空間とは、エージェントがある一つの、例えば視覚に相当するセンサーを持つときに認知できる空間を一つの認知空間であるとしていて、どの認知空間をどのように共有させるかが、協調探索を成功させるための大きな要因となる。本研究では、共有認知空間に様々な異なるレベルを設けて実験を行い、協調探索におけるパフォーマンスの向上を目標とする。各レベルごとに探索におけるステップ数と計算時間を評価を行い、協調探索の効果を調べる。

共有認知空間の段階には、「実験空間の共有」、「実験空間と仮説空間の共有」の二段階を考え、これらの協調条件と、独立条件の探索結果を比較し、協調条件での探索結果のパフォーマンスが、独立条件での探索のパフォーマンスを上回ったときの利得を創発として本論文で定義し、協調の有効性を実験的に検証していく。

# 目次

<b>1</b>	<b>はじめに</b>	<b>1</b>
<b>2</b>	<b>共有認知空間と実時間探索</b>	<b>4</b>
2.1	共有認知空間 . . . . .	4
2.1.1	共有認知空間とは . . . . .	4
2.1.2	共有認知空間を用いた先行研究 . . . . .	5
2.1.3	共有認知空間を用いた Wason の 2-4-6 課題 . . . . .	6
2.2	実時間探索 . . . . .	11
2.2.1	実時間探索 . . . . .	11
2.2.2	マルチエージェント実時間探索 . . . . .	15
2.3	従来研究に対する提案 . . . . .	16
<b>3</b>	<b>MARTA*への共有認知空間の導入</b>	<b>17</b>
3.1	共有の段階 . . . . .	17
3.2	独立条件 . . . . .	21
3.3	協調条件 . . . . .	21
3.3.1	実験空間だけの共有 . . . . .	22
3.3.2	仮説空間と実験空間の共有 . . . . .	23
3.4	協調探索の評価 . . . . .	24
<b>4</b>	<b>実験モデル</b>	<b>26</b>
4.1	実験モデル . . . . .	26
4.2	実験空間 . . . . .	26
4.3	評価実験 . . . . .	29

<b>5</b>	<b>実験結果</b>	<b>35</b>
5.1	実験結果 . . . . .	35
5.1.1	迷路サイズ . . . . .	35
5.1.2	分岐 . . . . .	38
5.2	考察 . . . . .	39
5.2.1	実験空間だけの共有 . . . . .	39
5.2.2	仮説空間の共有 . . . . .	40
5.2.3	数人の探索 . . . . .	41
<b>6</b>	<b>まとめ</b>	<b>43</b>
6.1	貢献 . . . . .	43
6.2	今後の課題 . . . . .	43

# 第 1 章

## はじめに

近年コラボレーションという言葉をよく聞くようになった。コラボレーションとは共同研究、共同作業という意味を持つが、企業間で同一ブランドを作ったり、テレビコマーシャルなどに同じ俳優、同じ設定を用いて、商品販売の相乗効果を狙ったものまである。これらのコラボレーションは足し算の効果ではなく、掛け算の効果があると言われているが、実際どのくらいの効果があるのだろうか。ただ闇雲にコラボレーションを行えばいいという事ではないことは、成功しているコラボレーションもあれば、失敗しているコラボレーションもあることから明らかである。しかし、このような共同作業の流れは今後増えていくと思われ、効果的なコラボレーションを設計するための指針を明確にすることは重要なテーマである。

現在、情報工学の分野で研究されている様々な事柄は、いずれ実世界の様々なものに実装されていく。このとき、製作コスト、実労働のための時間、燃料などのコストまでを考えた実装を施していかなければならない。そこで本研究では、実世界における基本的な探索問題を実世界のロボットに実装することを想定して、探索を行うことを考える。

迷路に代表されるような探索問題に対してロボットを用いて探索を行うことを考えると、探索空間が大きな場合、一台での探索では探索に時間がかかり過ぎる。また、必要以上に多くの台数を用いることも連絡コストの増大や、多くのロボットを用意することのコストの増大などの点で問題がある。そこで本研究では、エージェントを少数使うことを前提として、協調探索を効果的に行うことを考える。

複数のエージェントの協調探索では、エージェントが共有する認知空間によって差異が生ずることが知られている。認知空間とは、エージェントがある一つの、例えば視覚に相

当するセンサーを持つときに認知できる空間を一つの認知空間であるとしていて, どの認知空間をどのように共有させるかが, 協調探索を成功させるための大きな要因となる.

90年代に入ってから, マルチエージェントによる協調研究が行われているが, 協調の方法として, 認知空間の共有を用いる方法が研究されている. 比較的高い段階での協調を対象にした研究として, 2-4-6 タスク問題 [Wason 60] や, 二人のカードゲーム問題 [Laughlin 85] が解かれている. [三輪 00] では, 2-4-6 タスク問題においては協調の段階は認知空間の違いであるとして, 共有の段階として, 「実験空間だけの共有」, 「実験空間と仮説空間の共有」, 「実験空間と仮説空間の統合」の各段階に分け, それぞれの協調の効果を調べたところ, 効果的な診断テストを行うことができれば「実験空間と仮説空間の統合」が最も効果的であるということを述べている.

そこで本論文では, 迷路問題に対して共有認知空間の概念を用いるときに, 「実験空間だけの共有」, 「実験空間と仮説空間の共有」の二つの共有段階を考え, それぞれを明確に定義した上で, それぞれの段階について, 共同の効果がどのように変動するのかを, 計算機シミュレーションを通して実験的に検討する. また, 迷路探索のアルゴリズムとしては, マルチエージェント実時間探索 (Multi-Agent Real-Time A\*: MARTA\*) [Knight 93] を用い, 共有認知空間を用いることにより, MARTA\* に用いられている, RTA\* アルゴリズムを修正する.

上に述べた問題を考えるに当たって, 共同の利得を考えるものさしが必要となる. そのために本論文では, 三輪の論文にならい, 「創発」という概念を定義する [亀田 97]. 通常, 一人で問題を解くよりも, 二人で問題を解いたときの方がより高いパフォーマンスを示すことは, 経験的にも理解できる. しかし, この場合, 相互作用による創発が現れたとはみなさない.

本論文では, 創発の定義に関して, 次のような状況を考える.

### 1. 独立条件

エージェント同士は, 相互作用することはなく, 推定コストの値  $\hat{h}(\cdot)$  も共有しない状態で, 別々に目標を探索する. これは二人のエージェントが, 独立して問題を解く状況に対応する.

### 2. 協調条件

ここでは、各エージェントは協同で目標を探索する。実社会で人間が通信機器を持って通信しあったり、情報を物を使って迷路上に残して連絡を取り合いながら行う探索に相当する。

ここで、後者のパフォーマンスが、前者を上回るとき、創発が現れたとみなす。

創発の定義は様々なものがあり、パフォーマンスの比較だけからでは創発の出現は確認されないという立場もあるが、ここでは、相互作用の利得がパフォーマンスに現れたことをもって創発の定義とする。本研究では、二人のエージェントによる探索が、創発を導く可能性を検討する。そのために、探索過程のモデルを計算機上に構築し、計算機シミュレーションを通じた実験的検討を行う。

2章では、本研究に用いる共有認知空間と実時間探索について説明し、3章で実時間探索に共有認知空間をどのように導入していくかについて述べる。4章では、3章で導入した共有認知空間を用いた協調探索をどのように計算機上に実装していくかを具体的に記述し、5章でその実験結果に対する考察を行う。最後に、6章ではまとめとして、貢献と今後の課題について述べる。

## 第 2 章

# 共有認知空間と実時間探索

この章では、本研究に用いる「共有認知空間」と「実時間探索」について詳しく述べ、本研究の提案について触れる。

### 2.1 共有認知空間

#### 2.1.1 共有認知空間とは

協同には様々な段階が存在する。例えば、同じ研究室で、日頃から意見を交換し合いながら一緒に研究するという協同のスタイルもあれば、一定の成果があがった段階ではじめて、その成果に対して意見を求めるという形もある。また、公刊された論文を介して、何十年も以前に行われた実験の結果を知るというような場合、も過去の研究者との間に生じた協同の一種であると考えることができる。

前者と後者の協同の違いは、共有される「認知空間」の違いによって整理される。前者の協同では、互いの仮説やアイデアの相互作用が存在するのに対して、後者では、実験結果としてのデータの相互作用しか生じない。すなわち、後者が「実験空間」だけしか分かち合っていないのに対して、前者では互いの「仮説空間」も共有されている。この共有認知空間の差異によって、協同の効果が大きく違うことが知られている。

## 2.1.2 共有認知空間を用いた先行研究

先行研究としては, Wason の 2-4-6 課題に対して共有認知空間を用いて共同の効果を計算機シミュレーションを通して実証的に検討したものがあある [三輪 00]. Wason の 2-4-6 課題は, 認知心理学の研究アプローチにおいて, 好んで使われてきた発見課題の一つである.

### The Wason's 2-4-6 task

表 2.1: Wason の 2-4-6 課題の発見過程の例

仮説	生成事例	方略	フィードバック
-	2,4,6	-	Yes
連続する偶数	4,6,8	Ptest	Yes
連続する偶数	20,22,24	Ptest	No
24 の約数	8,8,8	Ptest	Yes
24 の約数	18,100,2	Ntest	No
24 の約数	8,6,4	Ptest	Yes
24 の約数	24,12,8	Ptest	No
一桁の数	1,1,5	Ptest	Yes

以下, 2-4-6 課題の実験手続きを説明する. 2-4-6 課題における被験者の課題は, 次々に提示される 3 対の数字の組の規則性を発見するというものである. 表 2.1 は, 典型的な実験結果の例である. 典型的な実験では, はじめに [2-4-6] という数字の組が Yes というフィードバック (以下, Yes-FB と略す) と共に与えられていて, 被験者はこの 3 組の数字の規則性「一桁の数」を発見する. ここで, Yes-FB は, その数字の組が発見すべきターゲットの「正事例」であることを表し, 逆に No-FB は「負事例」であることを表す. 被験者は, 提示された事例に基づいて「仮説」を形成する. 表 2.1 ではまずはじめに, 「連続する偶数」という仮説を形成している.

次に被験者は実験を行う. 実験とは, 自ら数字列を作り出し, それを実験者に提示する

ことである。表 2.1 では、連続する偶数という仮説に基づいて、[4,6,8] という事例を用いて実験を行っている。被験者は、実験者より提示された事例に対して、Yes-FB , もしくは、No-FB いずれかを受け取る。そのフィードバックに基づいて、仮説を「確証」したり、「反証」したりしながら、最終的にターゲットを発見したと確信が得られるまで実験を繰り返す。

この課題に対して、仮説に対する正事例をポジティブテスト (Ptest) , 仮説に対する負事例をネガティブテスト (Ntest) とし、それに着目した実験を行うスタイルがあり [Klayman 87], 三輪はその問題について二つのプロダクションシステムが、協調してターゲットを発見する過程をシミュレーションしている。

### 2.1.3 共有認知空間を用いた Wason の 2-4-6 課題

以下に、協調における段階を 3 段階に分けて設定し、相互作用における創発の定義を行ったものを示す。

#### 独立条件

二つのシステムが、独立してターゲットを発見する状況を考える。この場合、二つのシステムは、相互作用しない。別々にターゲットを探索し、二つのシステムによって得られた二つの最終仮説のうち少なくとも一つが、ターゲットと一致していれば、二つのシステムは独立してターゲットを発見したものとみなす。

#### 協調条件

以下に続く三つの状況は、共有する認知空間の程度に応じて、協調的発見過程の三つの段階に対応する。ここでは、二つのシステムが共同でターゲットを探索し、独立条件と同じように二つの最終仮説のうち少なくとも一つが、ターゲットに一致すれば、二つのシステムは共同してターゲットを発見したことになる。

#### [1] 実験空間だけの共有

二つのシステムは、1 回目の実験ではシステム A によって、2 回目の実験はシステム B

によって、3 回目の実験はシステム A によって、というように、交互に実験を行う。実験結果は、両方のシステムにフィードバックされる。

この段階では、各システムは、相互の実験結果だけを知ることが許され、相手のシステムがどのような仮説を形成しているのかを知らない。これは単に実験結果だけを共有する状況に対応する。

## [2] 仮説空間の共有

これは、先の実験空間に加えて、「仮説空間」をも共有する段階である。この段階では、各システムは、仮説を形成するときに、相手のシステムが持つ仮説を参照して、自分の仮説を決定することができる。二人の人間が話し合いながら、ターゲットを探索するような状況がこれに対応する。三輪のシステムにおける相手のシステムの仮説参照の在り方は、相手の仮説を知った各システムは、以下のいずれかの原則に自分の仮説を形成する。(a) 相手の仮説と「異なった仮説」を形成する、(b) 相手の仮説に「オーバーラップした仮説」を形成する、(c) 相手の仮説を「特殊化した仮説」を形成する、(d) 相手の仮説を「一般化した仮説」を形成する。

仮説空間を共有する段階では、相手の仮説の情報は、自分の仮説を形成するときにしか使用することはできない。すなわち、実験において、相手の仮説の情報を参照して、事例生成することはできない。

## [3] 実験空間と仮説空間の統合

実験空間と仮説空間を統合することによって、実験においても、相手の仮説の情報を利用することができる。

## 創発の定義

二つのシステムの相互作用における創発は、次のように定義している。まず、独立条件におけるパフォーマンスをベースパフォーマンスとし、次に、協調条件の三つの段階のそれぞれのパフォーマンスが、そのベースパフォーマンスを上回れば、その協調の段階で創発現象が現れたものとみなす。

## 計算機モデル

上記でのべたようなプロセスを、計算機上でシミュレートするために、システム A、システム B の「プロダクションメモリ」、「ワーキングメモリ」（作業記憶）、および「共通の黑板」から構成される、協調プロダクションシステム（以下、Interactive Production System; IPS と略す）のアーキテクチャを開発し、実験している。二つのシステムは、片方のシステムが、そのワーキングメモリの内容の一部を共通の黑板に書き込み、他方のシステムがそれを共通の黑板から自分自身のワーキングメモリ内に読み込むことによって相互作用している。また、モデルは 3 対の数字の規則性についての知識を持っている。知識は、「属性-値リスト」の形で組織化されている。例えば、「昇降」という属性に対しては、「増加する数」、「減少する数」、「同一かもしくは増加する数」、「同一かもしくは減少する数」、「減少して増加する数」、「増加して減少する数」、「同一の数」がその値の例である。システムが持つ規則性の属性は、昇降、数字間の差、偶数/奇数、範囲、スロット、倍数、約数、和、積、関係式、その他である。

## シミュレーション結果

上の IPS を用い、共有される認知空間を三段階にコントロールして行ったシミュレーションの結果を示し、協調的発見における創発の可能性を検討する。

### シミュレーションの概要

以下のシミュレーションでは、独立条件、および協調条件の三段階のそれぞれにおいて、35 個のターゲットを発見させている。表 2.2 に、実験に用いたターゲットを示す。

まず、一つのターゲットに対して、計 30 回のシミュレーションを繰り返し、各ターゲットごとの正答率を求める。次に、35 個のターゲットの正答率の平均を求め、それをパフォーマンスの評価値とする。また、二つのシステムの仮説検証方略の組み合わせとして、(1) Ptest 方略 × Ptest 方略、(2) Ntest 方略 × Ntest 方略、(3) Ptest 方略 × Ntest 方略、(4) Rtest 方略 × Rtest 方略の 4 種類を考える。ここで Ptest 方略とは、仮説検証において常に Ptest を用いる方略、Ntest 方略とは、常に Ntest を用いる方略を示す。さらに、Rtest 方略とは、ランダムに事例を生成する方略を示し、実験において仮説を使用しない方略である。

表 2.2: シミュレーションに用いた 35 種類のターゲット

属性	ターゲット
昇降	増加する数, 等しいか増加する数
数字間の差	二つの数字の差が 2, 二つの数字の差が同じ
偶数/奇数	連続する偶数, 三つの偶数
範囲	一桁の数, 正の数
スロット	一番目が偶数, 二番目が偶数, 三番目が偶数
倍数	2 の倍数
約数	12 の約数, 24 の約数
和	合計が偶数, 合計が二桁の数, 合計が正の数, 合計が 12, 合計が 12 の倍数, 合計が 6 の倍数, 合計が 4 の倍数, 合計が 3 の倍数, 合計が 2 の倍数
積	積が偶数, 積が二桁の数, 積が正の数, 積が 48
関係式	三番目=一番目 + 二番目, 三番目=一番目 × 二番目-2, 三番目=一番目 × 3, 二番目=一番目 × 2, 三番目=一番目 × n, 二番目=一番目 × m
その他	三つの異なる数

### 実験空間だけの共有

一般的に見て、実験空間が共有されるだけでは、いかなる方略の組み合わせにおいても、創発は生じないばかりか、むしろ独立条件の方が、協調条件よりも高いパフォーマンスを示す場合があることが結果から見られた。しかし、 $P_{test}$  方略  $\times$   $N_{test}$  方略の組み合わせで、20 回の実験が許された場合において、協調条件のパフォーマンスが独立条件のそれを上回ることが観察された。以上は、二つのシステムが異なる仮説検証方略を用い、かつ十分な実験が許される場合には、創発の可能性が現れることを示唆している。

大切なことは、この実験空間が共有されるだけの協調では、それぞれのシステムにおいて、作業記憶の能力の拡張や、新しいプロダクションルールの追加を必要としないということである。二つのシステムは、ただ単に互いの実験結果を交換し合うだけである。このシミュレーション結果は、そのような単純な相互作用においても、創発の可能性が存在していることを示している。

### 仮説空間の共有

二つのシステムが共に  $P_{test}$  方略を用いる条件において、仮説空間が共有されることの効果は絶大であることが実験の結果から見られた。「特殊化された仮説」を生成する場合を除いて、協調条件のパフォーマンスは、独立条件のそれを著しく上回っている。一方、他の三つの仮説検証方略の組み合わせにおいては、このような顕著な創発現象は確認されない。

### 実験空間と仮説空間の統合

この段階では、システムは、実験における事例の生成において、相手の仮説の情報を用いることができるようになる。仮説空間と実験空間を統合することの重要な利得は、システムが「診断テスト」という仮説検証法を用いることができるようになるということである。システムが仮説空間および実験空間を統合し、診断テストを行うことができれば、協調条件のパフォーマンスは、独立条件のいずれの仮説検証の組み合わせのパフォーマンスをも、大きく上回ることができる。

## 検討とまとめ

これらの結果から、二つのシステムが十分な認知空間を共有することが可能ならば、「一緒に行く」ということには、十分な意義が存在することを示している。

三輪は、これまでの認知心理学的実験の結果は、創発の可能性に関して一貫した結果を示していないことを指摘している。その原因の一つとして、認知空間の共有の程度が統制されていないことが考えられると述べている。しかし、実質的な認知空間の共有の程度は、個人の能力や、課題の特性によって、大きな影響を受ける。

共有認知空間の程度を統制した実験は、認知科学だけではなく、これからの社会に必要で、あるモデルを仮定したときに、そこから演繹的に予測される結果を示しているという意味で、重要な実証的知見であり、将来の新しい実験デザインを導くものとしても重要であると結論している。

## 2.2 実時間探索

### 2.2.1 実時間探索

試行錯誤を伴うような非決定的問題解決のための基本的な手法として、これまでに様々な探索手法が研究されてきた [Stuart 95]。深さ優先探索や幅優先探索などの力づく探索手法は計算量<sup>1</sup>の爆発を招き、それに対処するために発見的知識を用いて枝刈り<sup>2</sup>(pruning)を行う A\* [Nilsson 71] や、反復深化 A\* (Iterative-Deepening A\*: IDA\*) [Korf 85] などのヒューリスティック探索手法<sup>3</sup> が提案されている。しかし、一般的にはその計算量は、問題のサイズ (状態数) に対して指数的に増加する。そこで、そのような問題に対処する手法の一つとして、実時間 A\* (Real-Time A\*: RTA\*) [Korf 90] が提案されている。本研究では、RTA\* を基礎的手法として、協調について考えていくが、RTA\* の説明をする準備として、用語について説明する。

---

<sup>1</sup>計算量には時間計算量 (time complexity) と空間計算量 (space complexity) があり、それぞれ回を見つけるのにかかる時間と、探索を行うためにどのくらいのメモリを必要とするかを表している。

<sup>2</sup>探索木の枝を先読みから取り除くプロセス。

<sup>3</sup>ゴールに到達するために予想されるコストを見積もりするために与えられるヒューリスティック関数 (heuristic function) を用いた探索手法。

## 用語の説明

### グラフに関する基礎概念

数学的な準備として、グラフに関するに基本的な用語を定義する。

グラフ (graph) は節点 (node) の集合  $N (\neq \phi)$  と (有向) 辺 (directed link) の集合  $L (\subseteq N \times N)$  の組  $\langle N, L \rangle$  で表される。ここでは、 $(n, n) \notin L$  とする。あるグラフ  $\langle N, L \rangle$  において、 $n_i, n_j \in N$  であれば、 $n_j$  は  $n_i$  の子 (child)、 $n_i$  は  $n_j$  の親 (parent) と呼ぶ。節点の列  $(n_0, n_1, \dots, n_m) (m \geq 1)$  は  $\forall k (0 \leq k \leq m-1) : (n_k, n_{k+1}) \in L$  のときに  $n_0$  から  $n_m$  への経路 (path) と呼び、 $m$  を経路の長さ (length) と呼ぶ。特に、 $n_0 = n_m$  のとき、その経路を閉路 (cycle) と呼ぶ。辺にコストが  $c : L \rightarrow R^+$  (ただし、 $R^+$  は正の実数の集合) として与えられる場合には、経路  $(n_0, n_1, \dots, n_m)$  のコストは  $\sum_{k=0}^{m-1} c(n_k, n_{k+1})$  で与えられる。

木 (tree) は親を持たない根 (root) (節点) を除くすべての節点がたった一つの親を持つグラフである。子を持たない木の節点は葉 (leaf) と呼ばれる。根からある節点  $n$  までの経路の長さを  $n$  の深さ (depth) と呼ぶ。

### 問題の定式化

非決定的な問題解決は次のような状態空間表現により定式化することができる。問題 (problem) は、組  $\langle S, O, s_I, G \rangle$  により与えられる。ここで、 $S$  は空でない状態 (state) の集合、 $O \subseteq S \times S$  は状態遷移を生じさせるオペレータ (operator) の集合、 $s_I \in S$  は初期状態 (initial state)、 $G \subseteq S$  は目標状態 (goal state) の集合である。このとき、組  $\langle S, O \rangle$  は状態を節点、オペレータを辺とみなすことによりグラフとなり、状態空間グラフ (state space graph) と呼ぶ。

求める問題の解 (solution) とは、初期状態  $s_I$  から目標状態  $s_G \in G$  に至る任意の経路  $(s_0, s_1, \dots, s_n)$  (ただし、 $s_0 = s_I, s_n = s_G$ ) である。オペレータにコスト ( $> 0$ ) が与えられているとき、解のコストはその経路のコストとして与えられる。あるコストの解が存在し、それよりも小さいコストの解が存在しなければ、その解は最適解 (optimal solution) であるという。

### ヒューリスティック探索

状態空間グラフ探索の最も基本的な操作は、ある状態にオペレータを適用して得られ

る子状態を求める状態生成 (state generation) である。特にすべての子を生成することを状態展開 (state expansion) と呼び、このとき親状態は展開された (expanded) という。一般的なグラフ探索は、初期状態からはじめて、目標状態が発見されるまで、次々と生成された状態を展開することにより実行される。通常、探索過程を記録しておくために、生成された子  $s'$  にはその親  $s$  に戻るポインタ (pointer)  $(s', s)$  がつけられ、生成された状態とポインタから初期状態を根とした探索木 (search tree) が形成される。

展開する状態の順序によって深さ優先や幅優先など、さまざまな探索手法が利用可能であるが、このような力づくの探索手法は状態空間が大きい問題に対しては計算量の組み合わせ的爆発により現実には実行不可能である。そこで、それぞれの状態に評価値を与え、評価値の高い状態の展開を優先することにより、探索効率の向上を図っている。最適解を求める探索では一般に、状態  $s$  の評価値  $f(s)$  は以下に示す評価関数により与えている。

$$f(s) = g(s) + h(s)$$

ここで、 $g(s)$  は初期状態から状態  $s$  までの最適経路のコスト、 $h(s)$  は状態  $s$  から目標状態までの最適経路のコストである。しかしながら、この  $g(s)$  と  $h(s)$  は探索の途中では正しい値を得ることができないので、 $A^*$  などのヒューリスティック探索では以下に示す推定値を評価値として用いる。

$$\hat{f}(s) = \hat{g}(s) + \hat{h}(s)$$

ここで、 $\hat{g}(s)$  はすでに得られている探索木のなかで、初期状態から状態  $s$  までの最適経路のコストとする。また、 $\hat{h}(s)$  は  $s$  から目標状態までの最適経路のコストの推定値であり、ヒューリスティック関数として発見的に与えられるものである。 $A^*$  アルゴリズムでは解の最適性を保証する適格性 (admissibility) の条件として、すべての状態において  $\hat{h}(s) \leq h(s)$  が満たされなければならない。

## 実時間探索

$A^*$  アルゴリズムは経路探索と移動が分離されたオフライン探索と見なすことができる。 $A^*$  アルゴリズムは最適解を保証するが、一般的には問題のサイズに応じて指数的な計算量を必要とするために大きな探索空間を持つ問題に適用することは困難であった。そこで、一定の先読み探索と移動を交互に行うことにより、最適解を保証することはで

きないが、計算量を削減することのできるオンライン探索手法として、RTA\*(Real-Time A\*)[Korf 90] アルゴリズムが提案された。計算量は、解の長さに対して線形となる。

RTA\* アルゴリズムを以下に示す。

## RTA\* アルゴリズム

Step1 [初期化]  $s_x = s_I$  とする。ただし、 $s_I$  は初期状態とする。

Step2 [展開]  $s_x$  を展開して、その子状態の集合を  $C(s_x)$  とする。

Step3 [終了判定]  $\exists s_G \in G : C(s_x) \cap G \neq \phi$  を満たす目標状態  $s_G$  が存在するなら、 $s_x = s_G$  として終了。ただし、 $s_G$  は目標状態とする。

Step4 [先読み探索] すべての  $s_y \in C(s_x)$  について、 $s_y$  を経由して目標状態に至る場合の  $s_x$  の評価値  $\hat{f}(s_x, s_y) = c(s_x, s_y) + \hat{f}(s_y)$  を計算する。ここで、 $\hat{f}(s_y)$  は  $s_y$  から深さ  $d$  までの先読み探索結果をもとに次のように計算する。

$$\hat{f}(s_y) = \min_{s_w \in W(s_y, d)} [\hat{c}(s_y, s_w) + \hat{h}(s_w)]$$

ただし、 $W(s_y, d)$  は  $s_y$  を根とする深さ  $d$  の先読み探索木の葉状態の集合とし、 $\hat{c}(s_y, s_w)$  はその時点で知られている  $s_y$  から  $s_w$  までの最適経路のコストとする。

Step5 [移動候補選択]  $\hat{f}(s_x, s'_y) = \min_{s_y \in C(s_x)} \hat{f}(s_x, s_y)$  となる隣接状態  $s_y$  を求める。複数存在するときはランダムに選択する。

Step6 [推定コスト更新]  $\hat{h}(s_x)$  の値を 2 番目に小さい  $\hat{f}(s_x, s''_y)$  に更新する。もし存在しなければ、 $\hat{h}(s_x) = \infty$  とする。

Step7 [移動]  $s_x = s'_y$  とする。

Step8 Step2 へ。

RTA\* では、エージェントが探索空間を移動しながら探索を行うので、状態の評価値には A\* のような初期状態からのコストは含まれない。したがって探索の経過が評価値には現れないので、同じ状態を何度も訪問する無限ループに陥る可能性がある。しかし、RTA\* では Step6 において  $\hat{h}(\cdot)$  が単調増加するように更新されるので、無限ループに陥らず、

解が存在するならば必ず発見されるというアルゴリズムの完全性 (completeness) が保証されている。また, Step6 で  $\hat{h}(s_x)$  として更新される  $\hat{f}(s_x, s'_y)$  の値は, 状態  $s_x$  から移動の対象となる  $s'_y$  以外の状態を経由して目標状態まで至る経路の最小推定コストを示している。RTA\* において  $\hat{h}(\cdot)$  を 2 番目に小さい評価値に更新することは, 以前に訪問した状態への再訪問を抑制して, 探索効率を向上させることになるが, この値は状態の推定コストとしては過大評価を与えている場合もある。

## 2.2.2 マルチエージェント実時間探索

RTA\* において解の質を改善する方法の一つに Knight が提案したマルチエージェント実時間探索 (Multi-Agent Real-Time A\*: MARTA\*) [Knight 93] がある。MARTA\* では, 複数のエージェントはそれぞれ自律的, 並行的に先読み深さ 1 で RTA\* を実行する。Step5 において最小評価値をもつ移動候補が複数存在するときには, エージェントはランダムに移動先を決定する。

MARTA\* の有効性はおもに次の 2 点である。

1. 発見効果 エージェントの数を増加させれば, それだけ異なる経路がより多く探索されることになり, 解の質が改善される。また, 一部のエージェントが探索に行き詰まっても残りのエージェントが探索を続けることができる。
2. 学習効果 エージェント数を増加させれば, 推定コストの更新が活発になり, 解の質が改善される。

また, RTA\* における探索時間は, 先読み深さの増大に対しては指数的に増加するが, エージェント数の増加に対しては線形にしか増大しないので, 計算量の面から MARTA\* は有利である。さらに, MARTA\* は並列システムへの実装が容易なことから, 探索の高速化が期待できる。

MARTA\* の問題点はおもに次の 2 点である。

1. 評価値の過大評価  $\hat{h}(\cdot)$  の更新手法として 2 番目に小さい評価値に更新する RTA\* 方式を用いていて, 複数のエージェントが  $\hat{h}(\cdot)$  を共有するとき, 状態の評価値としては過大評価になる危険があるこの手法では, 他のエージェントの探索を妨げる可能

性がある.

2. 移動候補地の重複 同等な評価値を持つ複数の子状態に遭遇した場合移動候補選択がランダムに行われているため, エージェントの探索経路が重複することを避けることができない. このような重複した経路探索を行う場合, その探索の冗長性からエージェントの群としての探索効率を低下させることになる.

これらの問題点を解決するために LRTA\*(Learning Real-Time A\*) を導入し, 推定コストの過大評価を防ぐ方法を導入したり, エージェントを組織化し, 同じ場所に向かわないようにする手法を導入したりしているものがある [北村 96].

## 2.3 従来研究に対する提案

先に述べた二つの従来の研究を用いて, 本研究では共有認知空間を用いたエージェントによる協調探索を行う. それは, 共有認知空間の程度を統制した実験は, 認知科学だけではなく, これからの社会に必要であるということと, 情報分野の研究の将来の新しい実験デザインを導くものとしても重要であると考えからである.

迷路探索を用いたのは, カード問題のみならず, より現実に近い問題設定に共有認知空間の概念を用いることが必要であるということと, 多くの木構造の問題が実際に存在し, また迷路探索問題が昔から好んで使われてきた探索問題であるということがある. また, 実社会で使われるときに必要であるという見方から実時間探索を用いた. しかし, 前述の MARTA\* はマルチエージェントを使った探索にも関わらず, 共有認知空間の概念を用いて, 協調の段階を分けた研究がなされていない. また, MARTA\* を共有認知空間の段階に分け, それぞれの段階でできうる協調を使えば, 比較的容易に MARTA\* の問題点を改良することができる. なぜなら, 目印を置きながらの探索では, 問題点の一つである移動候補地の重複を明示的に解決することができるからである. また, 評価値の過大評価の問題も, 共有する認知空間の違いで制御可能である.

以上をもとに, 次章では, MARTA\* に共有認知空間を導入し, エージェントが共有する認知空間によって探索にどのような機能を持たせることができるかを述べる.

## 第 3 章

# MARTA\*への共有認知空間の 導入

### 3.1 共有の段階

直感的には、共有される認知空間の範囲が大きくなるのにしたがって、協調の効果が大きく変動することが考えられるが、迷路探索問題などについて、認知空間の共有の程度を明確にいくつかの段階に分けて行われた実験的検討はなされていない。そこで、本研究では、共有の段階は三輪の共有認知空間の分け方 [三輪 00] にならって、「実験空間だけの共有」、「実験空間と仮説空間の共有」に分け、それぞれの段階について、協調の効果がどのように変動するのかを、計算機シミュレーションを通して実験的に検討する。各協調の段階における、エージェントが行う行動は、共有する情報に基づいた行動を行うものとする。

以下に、迷路探索問題における各エージェントの記憶と、共有認知空間について述べる。

- エージェントの持つ記憶

1. マップ記憶 各エージェントが参照することができる、マップ記憶。
2. 評価値記憶 各エージェントが参照する評価値記憶。副目標がある場合は副目標までの評価値記憶と、ゴールまでの評価値記憶の二つを持つ。
3. 副目標の情報 副目標が達成されたかどうかの情報。これにより、副目標へのプランを、ゴールへのプランへと変更することができる。

- 共有される認知空間

1. 実験空間 本研究の設定では, 迷路そのものである. 実験空間をお互いのエージェントが共有し, 各エージェントは, 実験空間を介してのみ情報を交換することが許される. 本研究では, 各エージェントが保持しているマップ記憶から, 「訪問済み」情報と, 「行き止まり」情報が実験空間に書き込まれ, それを他方のエージェントが読み込むことにより, 情報を共有する.
2. 仮説空間 本研究の設定では, 評価値記憶と, 副目標の情報の交換を表す. 仮説空間の情報の交換は, エージェント同士の通信に相当する情報交換によって各エージェントが保持する情報の一部を交換する. 本研究では, 評価値記憶は毎回の情報交換に情報交換コストがかかり, 副目標の交換には, 一回の情報交換コストがかかる.

次に, 共有認知空間を用いることにより, RTA\* アルゴリズムを共有認知空間用に以下のように変更する. 下線は変更した個所を表す.

#### RTA\* アルゴリズム (共有認知空間用)

Step1 [初期化]  $s_x = s_I$  とする. ただし,  $s_I$  は初期状態とする.

Step2 [プランの変更] 副目標があり, それを達成していた場合, プランを変更する.

Step3 [展開]  $s_x$  を展開して, その子状態の集合を  $C(s_x)$  とする. ここで, 「行き止まり」, 「訪問済み (訪問回数)」の情報を自分のマップ記憶から得ることができる. 訪問回数は, 自分が訪問した回数しか加算されない.

Step4 [終了判定]  $\exists s_G \in G : C(s_x) \cap G \neq \phi$  を満たす目標状態  $s_G$  が存在するなら,  $s_x = s_G$  として終了. ただし,  $s_G$  は目標状態とする.

Step5 [マップ記憶の更新] 回りの壁の情報, 行き止まりの情報, 訪問回数を自分のマップ記憶に更新する.

Step6 [先読み探索] すべての  $s_y \in C(s_x)$  について,  $s_y$  を経由して目標状態に至る場合の  $s_x$  の評価値  $\hat{f}(s_x, s_y) = c(s_x, s_y) + \hat{f}(s_y)$  を計算する. ここで,  $\hat{f}(s_y)$  は  $s_y$  が

ら深さ  $d$  までの先読み探索結果をもとに次のように計算する.

$$\hat{f}(s_y) = \min_{s_w \in W(s_y, d)} [\hat{c}(s_y, s_w) + \hat{h}(s_w)]$$

ただし,  $W(s_y, d)$  は  $s_y$  を根とする深さ  $d$  の先読み探索木の葉状態の集合とし,  $\hat{c}(s_y, s_w)$  はその時点で知られている  $s_y$  から  $s_w$  までの最適経路のコストとする.

**Step7** [移動候補選択]  $\hat{f}(s_x, s'_y) = \min_{s_y \in C(s_x)} \hat{f}(s_x, s_y)$  となる隣接状態  $s_y$  を求める.

候補地が複数ある場合, 自分のマップ記憶の訪問状態を参照して, 訪問済みの場所があれば, そこを候補からはずして, 残りの候補地からランダムに選択する. また候補地が, すべて訪問済みのときは, 訪問回数を参照し, 訪問回数が最小となる候補地に移動する.

ここで, 繰り返し訪問の抑制のために, 展開された場所の評価値に関係なく, 訪問回数が 3 回以上の差がついた場合, 訪問回数が一番小さい場所を候補地として選択する.

**Step8** [推定コスト更新]  $\hat{h}(s_x)$  の値を 2 番目に小さい  $\hat{f}(s_x, s''_y)$  に更新する. もし存在しなければ,  $\hat{h}(s_x) = \infty$  とする. また, 副目標がある場合, 副目標探索中にゴールを発見したとき, ゴール探索用の評価値記憶にゴール位置から移動する度に推定コストを 0 として更新する. 推定コストの更新は, 各エージェントごとに作られている評価値記憶に更新する.

**Step9** [移動]  $s_x = s'_y$  とする.

**Step10** Step2 へ.

Step7 で, 繰り返し回数の抑制のための候補地選択があるが, 図 3.1 のような状態があるときに無限ループに陥ることを避けるために付け加えた.

図 3.1 は, a から d までの 6 マスの状態空間を表していて, 数字はその状態空間に与えられているヒューリスティック値である. 先読み深さが 1, (a) の状態からエージェントが c から d に移動したとき, c のヒューリスティック値は更新されて 9 となり, (b) の状態に移る. 次に, (b) の状態から (c) の状態に移り変わることがあれば, 無限ループとな

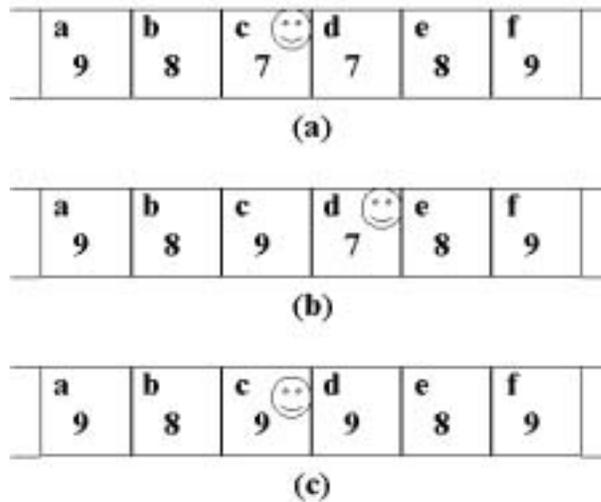


図 3.1: 繰り返しが起きる状態

り, エージェントは c d 間を行き来することになる. この条件は, 下の式のようになっている.

$$(\hat{h}(a) > \hat{h}(e)) \wedge (\hat{h}(f) > \hat{h}(e)) \wedge (\hat{h}(f) > \hat{h}(b)) \wedge (\hat{h}(d) > \hat{h}(b))$$

ここで,  $\hat{h}(a)$  は a のヒューリスティック値であり, 他のヒューリスティック値も同様になっている.

図 3.1 は無限ループが起きるときの例であり, 形状や, ヒューリスティック値は違うものであっても条件を満たせば, 無限ループに陥る. このため本研究では, 無限ループに陥ることを避けるために繰り返し回数の抑制のための候補地選択を付け加えた. 実際には, 探索した道を戻ってくるときにこの条件を満たすことがあり, 数値は図 3.1 の例のようにヒューリスティック値の差は 1 であることが確認されている. このような状態を避けるためには, 迷路の形状, ヒューリスティック値の初期値の与え方などの初期状態を適切なものに変えることが考えられるが, 様々な初期状態に対応する意味でも, 繰り返し回数の抑制のための候補地選択は重要である. また, 先読み深さを増やすことでも対応できる事が考えられるが, これは条件の値が変わり, 条件を満たす状態が少なくなるだけで, 問題の解決にはなっていない.

## 3.2 独立条件

まず,二人のエージェントが,独立して迷路を解く状況を考える(図3.2参照).このとき,エージェント同士は,相互作用することはない,推定コストの値 $\hat{h}(\cdot)$ も共有しない状態で,別々に目標を探索する.二人のエージェントのうち少なくとも一人が,目標状態に到達した場合,二人のエージェントが独立して迷路問題を解いたものとする.

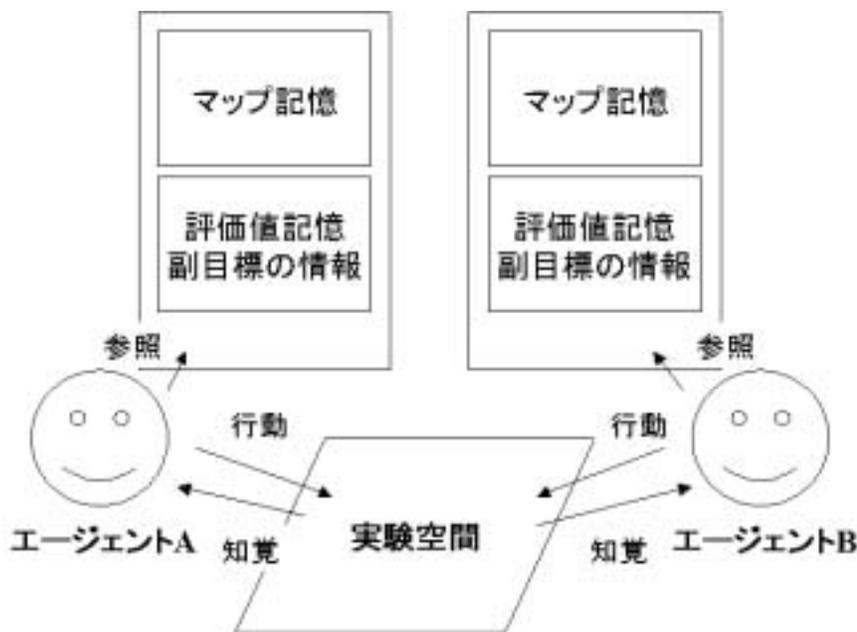


図 3.2: 独立条件の動作

## 3.3 協調条件

以下に続く二つの状況は,共有される認知空間の程度に応じた各段階に対応する.ここでは,各エージェントは協同で目標を探索する.独立条件のときと同じように,エージェントのどちらか一人が目標に到達すれば,二人のエージェントは協同して迷路問題を解いたものとする.

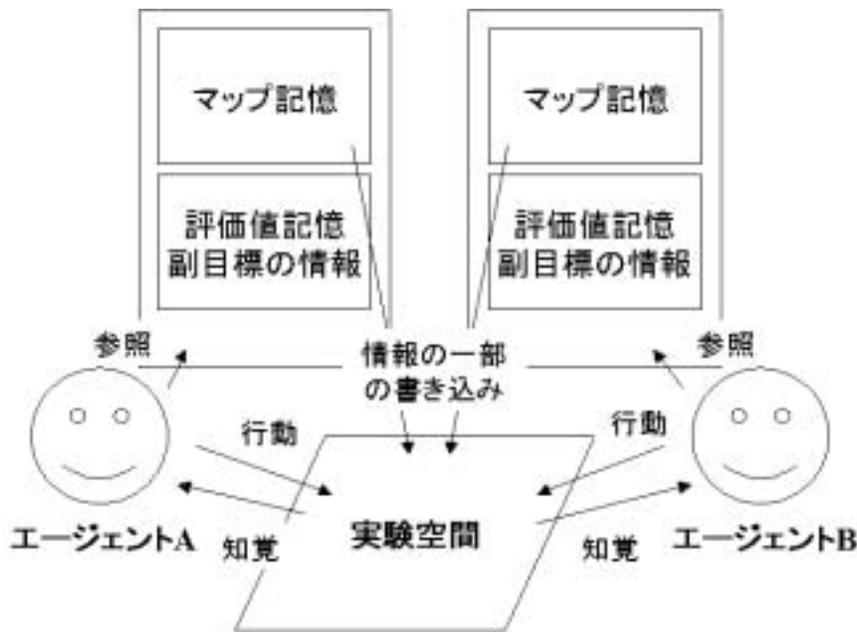


図 3.3: 実験空間を共有したときの動作

### 3.3.1 実験空間だけの共有

まず、「実験空間」だけを共有した段階を述べる (図 3.3 参照). 二人のエージェントは、まずエージェント A が RTA\* アルゴリズムによる推定値を計算し、移動候補を決定してから移動する。次にエージェント B が同様にして移動する。このように交互に移動を繰り返していく。その際、実験空間にあたる迷路そのものに行き止まりの情報や、ここは通ったことがある (足跡に対応) というような目印を残すことが許され、この残された情報だけを互いに交換することができる。この協調段階は、実社会で人間が通信機器を持っていないが、できるだけ情報を物を使って迷路上に残して伝えながら行う探索に相当し、システムでは実験空間そのものに情報を書き込むことと、その実験空間を互いが読み込むことによってのみ協調を行うことが可能である。通信機器を使わないという設定から、更新される推定コストもお互いが独自に保持するため、MARTA\* で挙げられている問題点である評価値  $\hat{h}(\cdot)$  の過大評価の問題はこの時点では発生しない。また、一度行った場所を書き込むことで、MARTA\* のもう一つの問題点であるランダム移動による移動先の重複を明示的に解消することが可能である。

ここで、RTA\* のアルゴリズムの Step3 , Step5 , Step7 は次のように修正される。

Step3 [展開]  $s_x$  を展開して, その子状態の集合を  $C(s_x)$  とする. ここで, 「行き止まり」, 「訪問済み (訪問回数)」の情報を自分のマップ記憶から得ることができる. 訪問回数は, 自分が訪問した回数しか加算されない.

Step5 [マップ記憶の更新] 回りの壁の情報, 行き止まりの情報, 訪問回数を自分のマップ記憶に更新する. また, そのなかから, 行き止まり情報と, 訪問済み情報 (回数は書き込まない) を実験空間に書き出す. 実験空間に書き出された情報は展開したときに, 各エージェントが読み込むことができる.

Step7 [移動候補選択]  $\hat{f}(s_x, s'_y) = \min_{s_y \in C(s_x)} \hat{f}(s_x, s_y)$  となる隣接状態  $s_y$  を求める. 候補地が複数ある場合, 実験空間から, その場所の訪問状態を参照して, 訪問済みの場所があればそこを候補からはずして, 残りの候補地からランダムに選択する. また候補地が, すべて訪問済みのときは, エージェントが持つマップ記憶から訪問回数を参照し, 訪問回数が最小となる候補地に移動する.

### 3.3.2 仮説空間と実験空間の共有

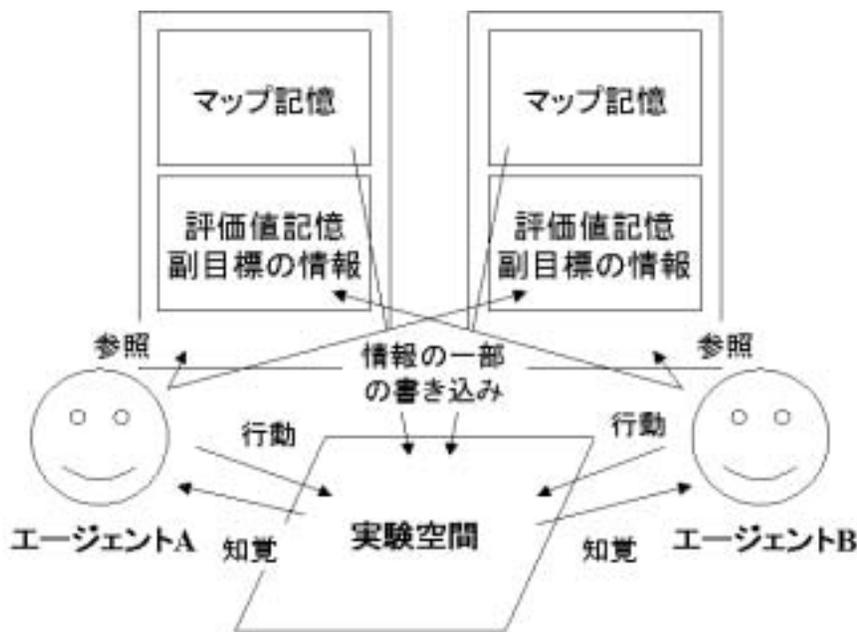


図 3.4: 仮説空間と実験空間を共有したときの動作

次に、先の実験空間に加えて、「仮説空間」を共有する段階について述べる (図 3.4 参照).

この段階では、エージェントが携帯電話のような情報伝達の道具を持ち、連絡を取り合いながら行う探索に相当し、推定コストの更新値や、副目標を達成した場合のプランの変更に関する情報交換もできる。これにより、実験空間だけを共有した場合に比べ、行われる情報交換が多くなり、探索がより早いステップ数で終了することが予想されるが、同時に MARTA\* で挙げられていた問題点である、評価値  $\hat{h}(\cdot)$  の過大評価の問題と、実験空間のときにはなかった情報交換のコストについての問題がでてくる。

ここで、RTA\* のアルゴリズムの実験空間のみ共有した場合に加えて、Step2, Step8 は次のように修正される。

Step2 [プランの変更] 副目標があり、それを達成していた場合、プランを変更する。また、もう一方のエージェントが副目標を達成していた場合、副目標の情報を読み取ることにより、自分のプランも変更する。

Step8 [推定コスト更新]  $\hat{h}(s_x)$  の値を 2 番目に小さい  $\hat{f}(s_x, s_y'')$  に更新する。もし存在しなければ、 $\hat{h}(s_x) = \infty$  とする。また、副目標がある場合、副目標探索中にゴールを発見したとき、ゴール探索用の評価値記憶にゴール位置から移動する度に推定コストを 0 として更新する。推定コストの更新は、各エージェントごとに作られている評価値記憶に更新する。また、もう一方のエージェントが持つ評価値記憶にも更新する。こうすることにより、各エージェントが同じ評価値記憶を持っているのと同じ状態を作る。

### 3.4 協調探索の評価

以上、独立条件と、二段階の協調条件について述べた。ここで、二人のエージェントの相互作用による協調探索の効果を測るために「創発」という概念を定義する。

まず、独立条件におけるパフォーマンス (ステップ数) をベースパフォーマンスとする。次に、協調条件の二つの段階のそれぞれのパフォーマンスが、そのベースパフォーマンスを上回れば、その協調の段階で創発現象が現れたとみなす。

情報交換にコストが生じるとき、認知空間として、実験空間と仮説空間を共有させた段階の協調が一番優れているということは一概には言えず、実験空間だけを共有する場合

との間で、探索に生じる差異がどれほどになるのかを実験的に示す必要がある。また、情報交換する内容も分割し、どの情報交換が探索に大きな影響を及ぼすかも検証する。このことにより、できるだけ少ない情報交換で仮説空間と実験空間を共有した段階の探索を行うことをめざす。

このとき行われる情報交換 1 回にかかるコストを  $cost_I$  とし、そのコストのステップ数に対する割合と探索効率を、実験空間だけ共有した場合と比較して、仮説空間を共有することの有効性を検証する。

# 第 4 章

## 実験モデル

### 4.1 実験モデル

3章で述べたようなプロセスを, 計算機上でシミュレートするために, システムを構築し実験を行う. 図 4.1 にエージェント数が 2 の場合の基本仕様を示す. システムは, エージェント A, エージェント B の「個人記憶」と, 「共通の黒板」で構成されている. 「個人記憶」は, マップ記憶, 評価値記憶, 副目標の情報のことを指す. また, 「共通の黒板」は, 共有認知空間として実験空間を共有したときは, 実験空間そのものであり, 仮説空間を共有したときは, それとは別に共通の黒板を用意する. 各エージェントは, 作業記憶の一部を共通の黒板に書き込み, それぞれのエージェントがそれを自分の作業記憶に読み込むことによって相互作用する.

次に述べるような実験空間のもとで, それぞれの共有段階に分けたシステムを実装し, 評価実験を行った.

### 4.2 実験空間

実験空間は以下のようにになっている.

- 格子状迷路において入り口 (初期状態) から出口 (目標状態) までの経路を求める問題である.
- エージェントは上下左右のみに移動し, 斜めには移動できないものとする.

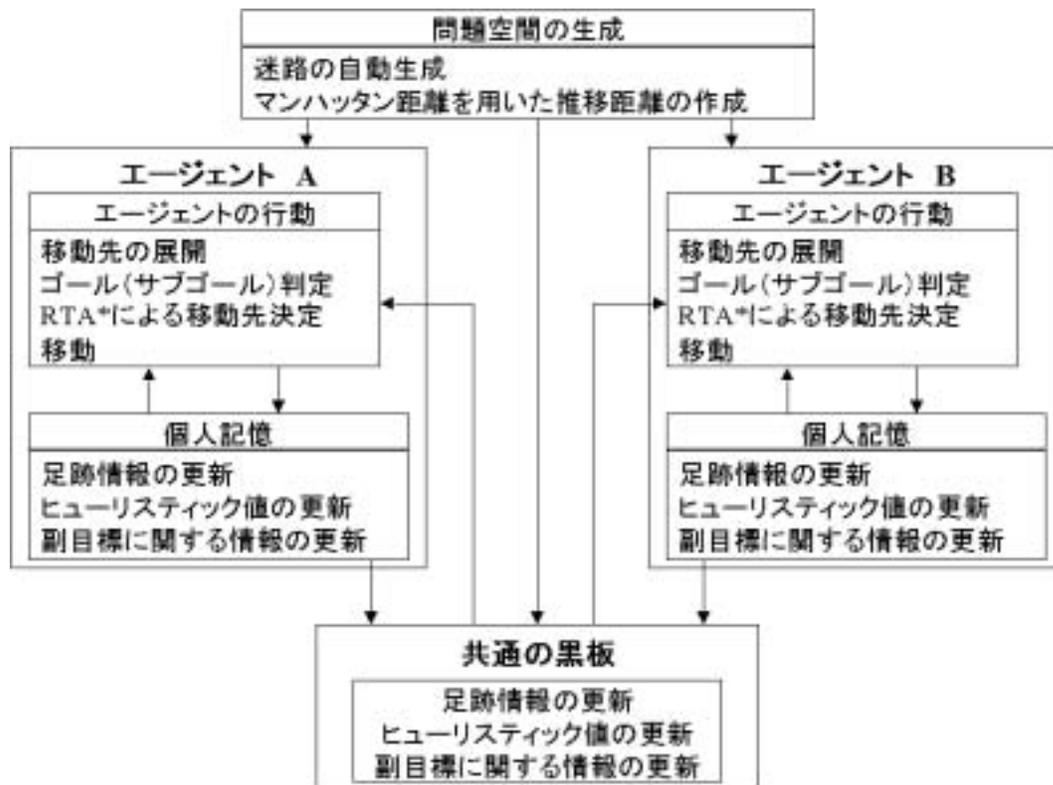


図 4.1: システムの動作

- エージェントの移動には1回の移動につきコストが1かかるとする.
- 迷路はランダムに道と壁とで作られた, 入り口を左上隅, 出口を右下隅とする  $25 \times 25$ ,  $51 \times 51$ ,  $75 \times 75$ , および  $101 \times 101$  の格子状グラフとを用いる.
- 入り口は  $(1,1)$ . 出口は  $(xSize-1,ySize-2)$ . ここで,  $xSize$ ,  $ySize$  は迷路の  $x$  方向のサイズと,  $y$  方向のサイズである.
- 推定コストの初期値には目標状態からのマンハッタン距離<sup>1</sup> (Manhattan distance) を用いる.
- 副目標を設置する場合, 迷路の問題空間のほぼ中央に設置する. 以下に実際に副目標がおかれた座標を示す.
  - $25 \times 25$  :  $(14,14)$
  - $51 \times 51$  :  $(26,26)$
  - $75 \times 75$  :  $(38,38)$
  - $101 \times 101$  :  $(50,50)$

また, 実験空間となる迷路はランダムに生成されるが, そのアルゴリズムは以下のようになっている.

### 迷路生成アルゴリズム

**Step1 [初期設定]** 迷路サイズ  $(x_{size}, y_{size})$  を読み込み,  $(0,0)$  から  $(x_{size} + 1, y_{size} + 1)$  までの空間を準備する. また,  $(0, y_{all})$ ,  $(x_{all}, 0)$ ,  $(x_{size} + 1, y_{all})$ ,  $(x_{all}, y_{size} + 1)$  となる4辺を道 (road) とし, 残りの場所を壁 (wall) とする. ここで,  $x_{all}$  と  $y_{all}$  は, すべての  $x$  と  $y$  を表す.

**Step2 [初期化]**  $s_x = s_I$  とし, その場所を道とする. ただし,  $s_I$  は初期状態で  $s_I = (2, 2)$

**Step3 [展開]**  $s_x$  の2マス隣りを展開して, その子状態の集合を  $C(s_x)$  とする. そのとき, 壁の場所を展開可能とし, 道であれば展開できない.

<sup>1</sup>ゴール状態からの距離の和. 計算する距離は縦と横の距離の和である.

Step4 [終了判定] すべての  $C(s_x)$  が道なら,  $s_x = (x_{before}, y_{before} + 1)$  として Step3 へ.  $y_{before} = y_{size} - 1$  なら,  $s_x = (x_{before} + 1, 2)$  として Step3 へ. このとき,  $x_{before} = x_{size} - 1$  なら Step9 へ. ここで,  $(x_{before}, y_{before})$  は, 展開し始めた最初の点.

Step5 [移動先選択] すべての  $s_y \in C(s_x)$  から 移動先  $s_y$  をランダムに選択する.

Step6 [マップ更新] 現在の場所  $s_x$  から展開された先  $s_y$  までの 2 マスを道とする.

Step7 [移動]  $s_x = s_y$  とする.

Step8 Step3 へ.

Step9 [ゴールの設置] ゴールを  $(x_{size}, y_{size} - 1)$  に設置する. 副目標が無い場合は終了. 副目標がある場合には Step10 へ.

Step10 [副目標の設置] 副目標がある場合は任意の場所に副目標を設置して終了.

ここで, 迷路サイズを上下左右に大きくしたのは, アルゴリズム上の Step4 で終了判定をするときに必要な操作で, これにより外側の壁を生成することができる. この周りに作った道は, 迷路生成が終了したときに無くしてもかまわない. また, Step4 から Step3 に戻るときに迷路の分岐が生成される.

図 4.2 に実験空間である, ランダム生成させた迷路の例 (サイズは  $51 \times 51$ ) を示す.

実験結果は, ランダムな移動候補地選択が探索結果に影響をおよぼすため, 各共有段階ごとに一つの迷路に対して, 1000 回のシミュレーションを繰り返し, 各迷路に対しての平均探索ステップ数を求める. 次に, 1000 個の迷路についてそれを繰り返し, その平均値を各共有段階の探索にかかる平均ステップ数とし, その平均ステップ数をその段階のパフォーマンスとする.

### 4.3 評価実験

エージェントは並列に動作可能で, 1 回の移動ごとに 1 単位時間かかると仮定している. ただし, 先読み探索や, 実験空間の情報を読み書きするときにかかる計算コストは含

迷路の大きさは51\*51 branchnumber67

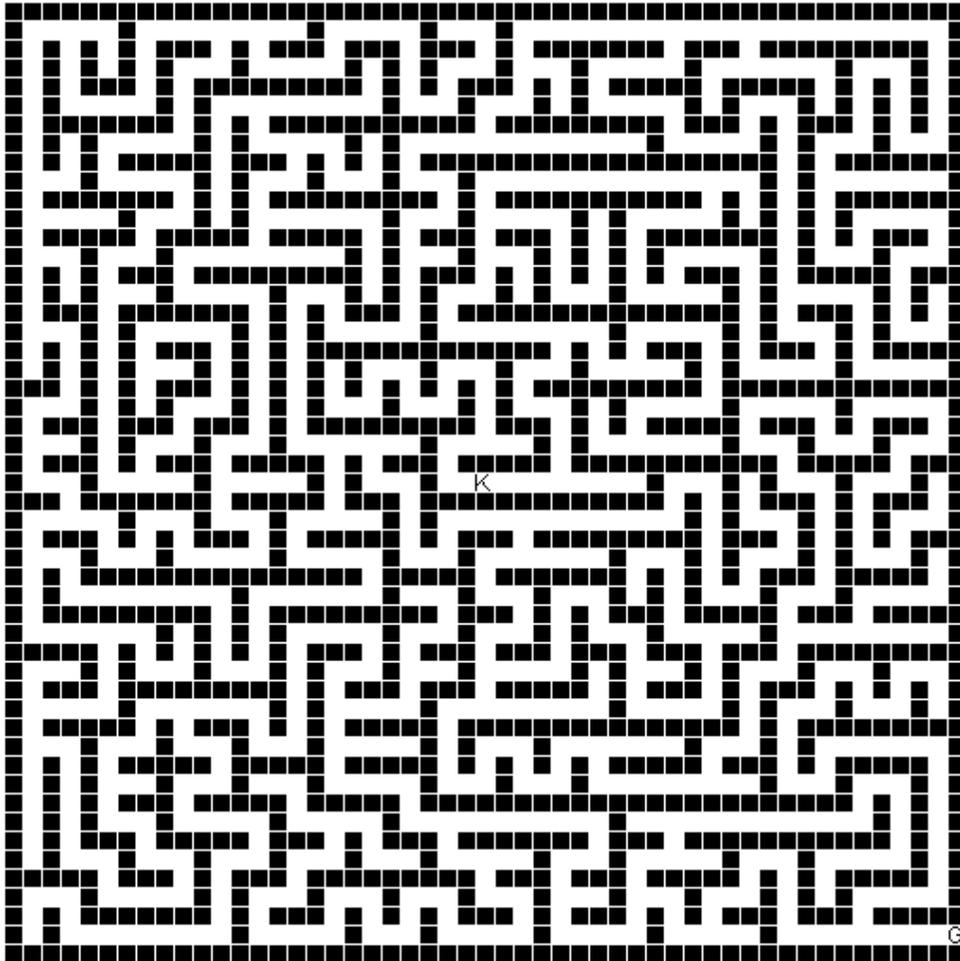


図 4.2: ランダム生成された迷路の例

めず、協調の段階によってできる差異がもたらす違いに注目する。

実験では、探索にかかるステップ数を測定するが、二人のエージェントのなかで最初に目標状態に到達したエージェントの所要時間が探索時間となる。また、実験空間上に副目標を設置した場合の探索時間の違いも調べる。副目標が入ることによって、問題の難度があがることが考えられるが、情報交換ができる場合は他の段階に比べて有利に探索ができることが予想され、各協調の段階がどのような探索に向いているのかを細かいレベルで知ることができる。

以下に、一つの迷路に対する繰り返し回数の検討と、実験に用いる迷路数の検討を行った結果をのせる。

#### 一つの迷路に対する繰り返し回数の検討

実験を行うにあたって、ランダムな移動候補地選択が、探索結果に影響をおよぼすために一つの迷路に対して、ステップ数の平均をとるが、繰り返し回数が 1000 回で妥当であるかどうかを実際に一つの迷路に対して、10000 回試行を繰り返し検証した。

副目標のない、迷路サイズ  $101 \times 101$  の 10 種類の同一迷路に対して独立条件で探索を行い、その平均をとった。それぞれ 10000 回試行したものと、1000 回試行したものを図 4.3 と図 4.4 に示す。

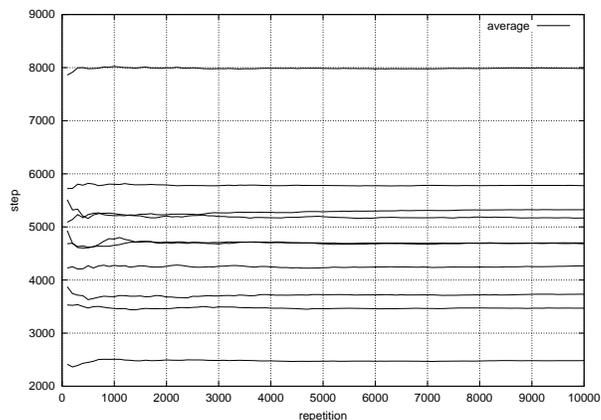


図 4.3:  $101 \times 101$ ; 独立条件; 副目標無し; 同一迷路でのステップ数の平均値 (10000 回)

また、副目標のある、迷路サイズ  $101 \times 101$  の 10 種類の同一迷路に対して独立条件で

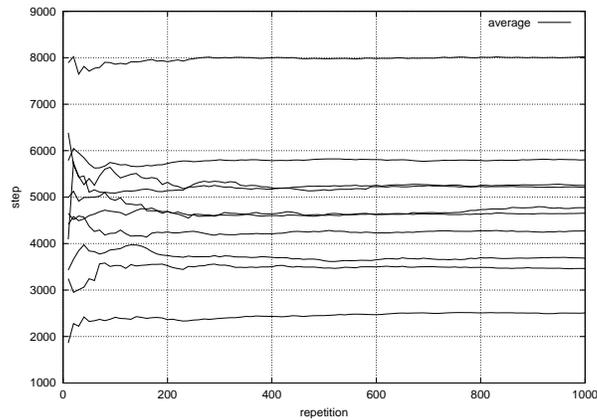


図 4.4:  $101 \times 101$ ; 独立条件; 副目標無し; 同一迷路でのステップ数の平均値 (1000 回)

も同様の探索を行ったが同じような傾向を示した. これらの結果から, 1000 回の繰り返し試行で十分な平均値がとれることがわかる.

次に, 協調空間での繰り返し回数についても同様の実験を行った. 副目標のない  $101 \times 101$  サイズの迷路での, 実験空間を共有させた場合, 実験空間と仮説空間を共有させた場合についての 1000 回の繰り返しを, それぞれ図 4.5 と図 4.6 に示す.

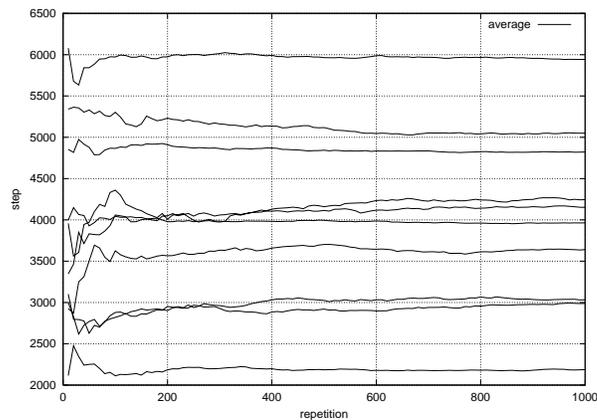


図 4.5:  $101 \times 101$ ; 実験空間の共有; 副目標無し; 同一迷路でのステップ数の平均値 (1000 回)

また, 副目標のある  $101 \times 101$  サイズの迷路での, 実験空間を共有させた場合も同様な

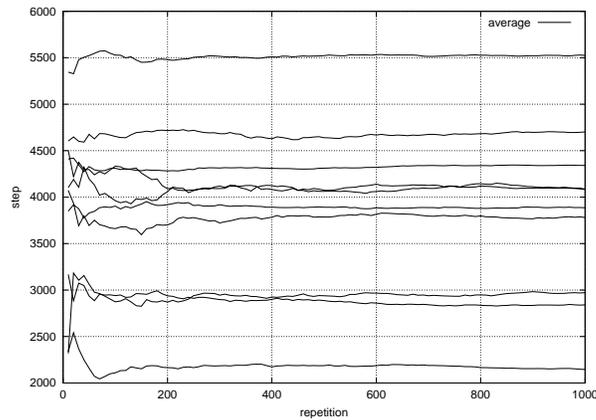


図 4.6:  $101 \times 101$ ; 実験空間と仮説空間の共有; 副目標無し; 同一迷路でのステップ数の平均値 (1000 回)

実験を行ったが、これも同じような傾向を示した。

これらの結果から、1000 回の繰り返し試行で各協調条件すべてについて、十分な平均値がとれることがわかる。他の迷路サイズでもこの実験を繰り返し、1000 回の繰り返し回数が適切であることを確認した。

### 迷路数の検討

各実験設定に対し、一つの迷路を 1000 回繰り返し、ステップ数を平均したものを 1000 個の迷路について行い、さらに平均して、各設定でのステップ数とする。ここで、迷路数が 1000 個で妥当かを検証する。

図 4.8 と図 4.7 は、一つの迷路で 1000 回の試行をした、1000 個の迷路を平均したものである。 $101 \times 101$  サイズの迷路で、副目標のないものと、副目標のあるものを示す。

このことから、1000 個の迷路からの平均をとることで、安定したパフォーマンスを示すことがわかる。この結果より、1000 個の迷路に対して、それぞれ 1000 回の試行を行い、実験結果をとった。

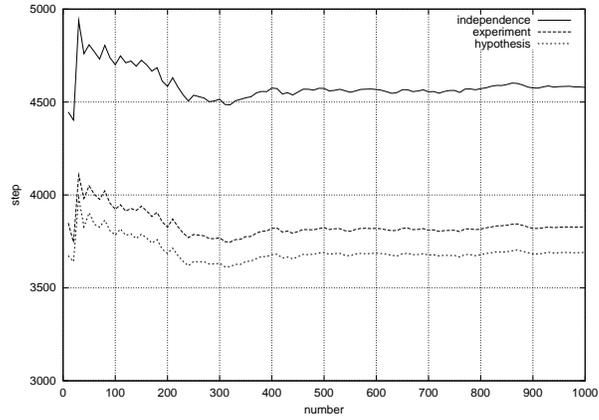


図 4.7:  $101 \times 101$ ; 副目標有り; ステップ数の平均値 (1000 個)

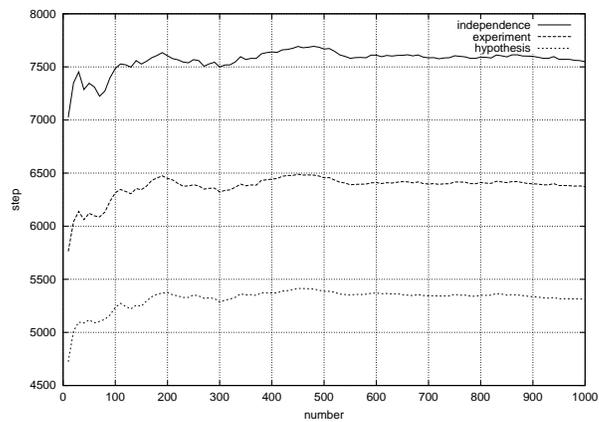


図 4.8:  $101 \times 101$ ; 副目標有り; ステップ数の平均値 (1000 個)

# 第 5 章

## 実験結果

4 章で述べたような実験システムを用いて、独立条件と二つの協調条件について、迷路に「迷路サイズ」と「分岐数」の違いを持たせて、共有認知空間が協調に与える影響を調べた。

### 5.1 実験結果

#### 5.1.1 迷路サイズ

迷路サイズは、問題空間の大きさであり、探索問題に共通する難しさと言える。そのため、まず、迷路サイズごとに共有認知空間を用いた探索がどのような効果を見せるのかを実験的に検証していく。

副目標のない場合のサイズごとの探索結果を表 5.1 と図 5.1 に示す。

表 5.1: 迷路サイズごとのステップの平均値 (副目標無し)

迷路サイズ	独立条件	実験空間	実験空間と仮説空間
25	226.6	180.8	174.4
51	1089.8	886.0	862.0
75	2441.5	2040.9	1974.6
101	4578.7	3827.8	3690.7

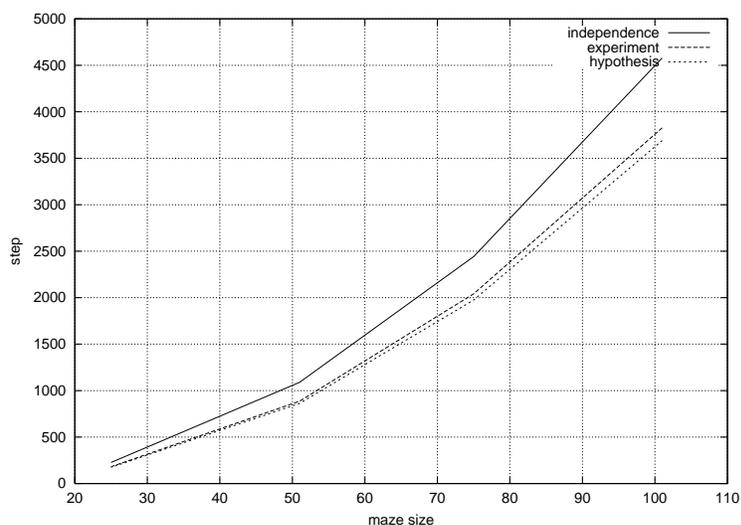


図 5.1: 副目標無し; ステップ数の平均値

次に, 副目標がある場合のサイズごとの探索結果を表 5.2 と図 5.2 に示す.

表 5.2: 迷路サイズごとのステップの平均値 (副目標有り)

迷路サイズ	独立条件	実験空間	実験空間と仮説空間
25	347.1	306.1	237.8
51	1727.5	1464.2	1215.8
75	4001.1	3391.7	2833.3
101	7548.6	6372.3	5309.0

図 5.1 から, 独立条件のときの探索にかかるステップ数よりも, 協調条件の探索にかかるステップ数の方が, 迷路サイズが大きくなるほど小さくなっていくことから, 問題空間のサイズが大きくなるにつれて協調の効果が大きくなっていることがわかる. また, 図 5.2 との比較より, 副目標がある場合に比べて実験空間のみを共有した協調と実験空間と仮説空間を共有したものとの差異がほとんどないことを観察した. このとき, 実験空間と仮説空間を共有している協調探索と, 実験空間だけを共有している探索との情報交換の差は, 実際には推定コストの更新値を前者が共有しているということだけなので, この結果から, ヒューリスティック値の共有は探索を助ける大きな情報交換ではないことが推測

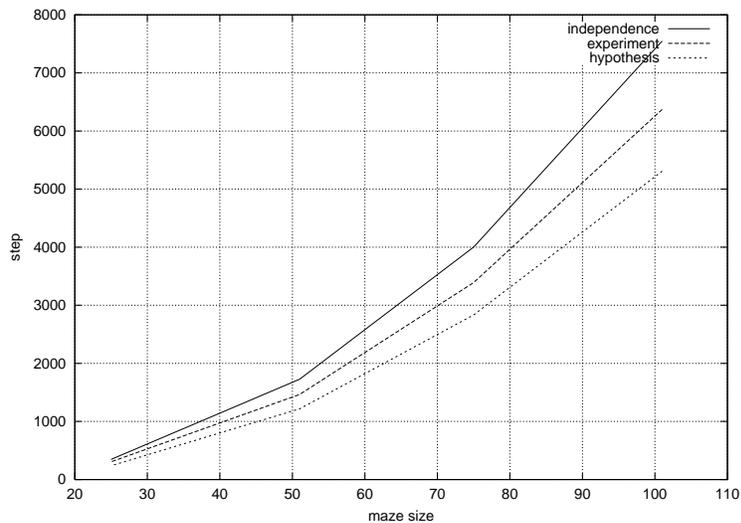


図 5.2: 副目標有り; ステップ数の平均値

できる.

実際に  $101 \times 101$  の迷路でのパフォーマンスは, 独立条件が 4578.7 に対して実験空間のみの協調が 3827.8, 実験空間と仮説空間の協調が 3690.7 となっていて, パフォーマンス比は 1:0.836 と 1:0.806 となっている. 実験空間を共有した場合の 16% ものパフォーマンスの向上に比べて, 実験空間と仮説空間を共有したパフォーマンスの向上はわずか 3% の向上となっている. 推定コストの更新値の交換のための情報交換コスト  $cost_I$  は, 両協調条件の差である 137.1 以下でなければ推定コストの更新値を共有するだけの共有認知空間を用いることに大きな意味はない. これは, 実験空間と仮説空間を共有したときのパフォーマンスとの比率から, 移動に対して 0.037 ステップでこの情報の交換ができなければならないことを示す. また, 実験空間の共有をした場合, エージェントにはセンサーと実験空間への書き込みができる機能のみを実装すればよいのに対し, 情報交換のための通信機を実装することを考えると, 単に迷路を解くという作業に対してはこの場合の探索において, 推定コストの更新値の共有は探索を助ける大きな要因ではない.

しかし, 副目標があり, プランの変更が必要な探索について見ると, 情報を交換するという事には大きな意味があることが確認できる. 実際に  $101 \times 101$  の迷路でのパフォーマンスは, 独立条件が 7548.6 に対して実験空間のみの協調が 6372.3, 実験空間と仮説空間の協調が 5309.0 となっていて, パフォーマンス比は 1:0.844 と 1:0.703 となっている.

これは、それぞれ 15 % と 14 % のパフォーマンスの向上であり、仮説空間を共有することに大きなメリットがある。

情報交換のためにかかるコスト  $cost_I$  は、1063.3 以下であればよく、また、実際にプランを変更するために副目標達成の情報を交換するのは一回きりであり、一回の情報交換にかかるコストが 1063.3 ステップ以下であればいい。このうち、推定コストの更新値の共有によるパフォーマンスの向上は 3 % であることから、904 ステップ以内のコストで一回の通信ができるときに、この仮説空間の共有が意味を持つようになる。

### 5.1.2 分岐

次に、迷路の分岐の数が協調探索にどのような影響を与えるかを調べた。MARTA\* の移動候補地の重複を考えたとき、分岐は移動候補を選択する重要なポイントであることから、分岐数をパラメータとした。図 5.3 と、図 5.4 は、それぞれ迷路サイズ  $25 \times 25$  と、迷路サイズ  $101 \times 101$  についての結果を示す。結果は、それぞれ各分岐ごとに、40 個の迷路で平均をとって、1 個の迷路では、1000 回試行している。また、分岐の数は、ランダム生成された迷路の中で、十分な数が得られたものを用いている。

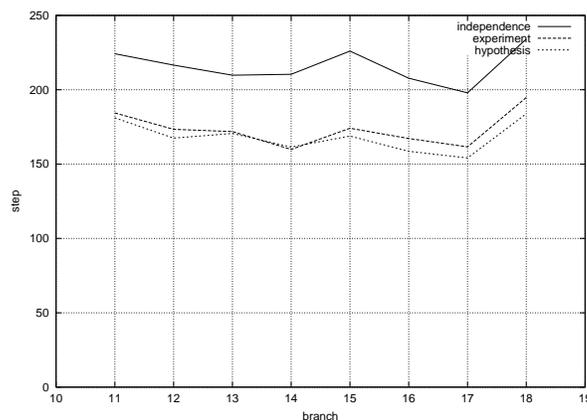


図 5.3: 分岐ごとのステップ数;  $25 \times 25$

結果から、一つの分岐ごとにもっと多くの迷路で平均をとると、サイズごとの平均に近づくことが予想される。これは、迷路においてできる分岐は、ある程度の分岐数以上になると、協調探索の効率に影響してこなくなる事が考えられる。それは、この実験が二人の

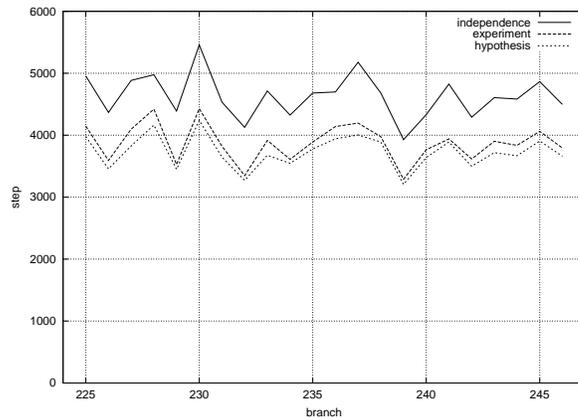


図 5.4: 分岐ごとのステップ数; 101 × 101

エージェントによるもので、大きな迷路になればなるほどエージェント同士が近くで探索をしているという可能性が少なくなり、一人での探索をしているような振る舞いをすることが多くなるということや、一つの迷路において、探索の早く終わったものと、遅く終わったものの十分な平均をとっているために、探索において分岐以外の、より大きく探索に影響を与える要素が結果に出てきたことが考えられる。

木構造の探索空間に対して、迷路サイズと分岐の結果から、分岐の数よりも展開できる節点の数が協調探索に影響を与えていると予想できる。展開できる節点数は、迷路サイズに比例することから、格子状のループのない迷路では、迷路サイズが共有認知空間を用いた協調探索の効率に影響しているということがわかる。

## 5.2 考察

### 5.2.1 実験空間だけの共有

探索の結果から、実験空間が共有されることで、パフォーマンスに約 15 % の向上が見られた。そして、エージェントにセンサーと何らかの書き込み機能を備えるだけで通信機能を持たなくても創発が現れている。

迷路サイズと分岐の結果から、このような迷路では、分岐の数よりも展開できる節点の数が協調探索に影響を与えていて、展開できる節点数は、迷路サイズに比例することから、

格子状のループのない迷路では、迷路サイズが共有認知空間を用いた協調探索の効率に影響しているということがわかった。

通信機器を使わないという設定から、更新される推定コストもお互いが独自に保持するため、MARTA\* で挙げられている問題点である評価値  $\hat{h}(\cdot)$  の過大評価の問題は発生しておらず、MARTA\* の改良についても可能性を示した。さらに、一度行った場所を書き込むことから、MARTA\* のもう一つの問題点であるランダム移動による移動先の重複を解消して、推定コストの更新に頼らない協調による探索を可能にしている。さらに細かな実験空間に書き込み可能な情報を加えていけば、より推定コストを用いる探索の効率に近づいていくと推測できる。

大切なことは、この実験空間が共有されるだけの協調では、それぞれのシステムにおいて、作業記憶の能力の拡張や、新しいプロダクションルールの追加を必要としないということである。二つのシステムは、ただ単に互いの実験結果を交換し合うだけである。このシミュレーション結果は、そのような単純な相互作用においても、創発が存在していることを示している。

### 5.2.2 仮説空間の共有

探索が独立条件のときより早いステップ数で終わり、創発が現れていることが確認できた。しかし、副目標のない場合の結果から、情報交換が多くなる分だけその効果には疑問が残った。

探索の結果から、推定コストの更新値の共有は探索を助ける大きな要因にはなり得ないことが観測された。推定コストの更新値の過大評価の問題や、推定コストをもう一方のエージェントに伝える情報交換について考えると、毎回の通信にかかるコストや、その通信システムを実装するコストを考えても、推定コストを共有するための共有認知空間は必ずしも必要ではないことが考えられる。しかし、プランの変更を伴う副目標を設置した場合、プランを変更するときだけの通信で済み、しかもその効果は大きなものとなり、実社会のエージェントにプラン変更の情報交換の機能を備えることは、大きな効果があると期待できる。また、共有できる情報を増やすことによって、この探索はより効果的に行われることが予想できる。

### 5.2.3 数人の探索

最後に, エージェント数が共有認知空間を用いた探索に与える影響を見るために, 副目標のない  $51 \times 51$  サイズの迷路に対して, 二人の探索に加え, 三人での探索と, 四人の探索を行った. その探索の結果を表 5.3 と, 図 5.5 に示す.

表 5.3: 数人の探索  $51 \times 51$

人数	独立条件	実験空間	実験空間と仮説空間
2	1089.8	886.0	862.0
3	960.1	758.3	716.9
4	877.2	696.5	642.0

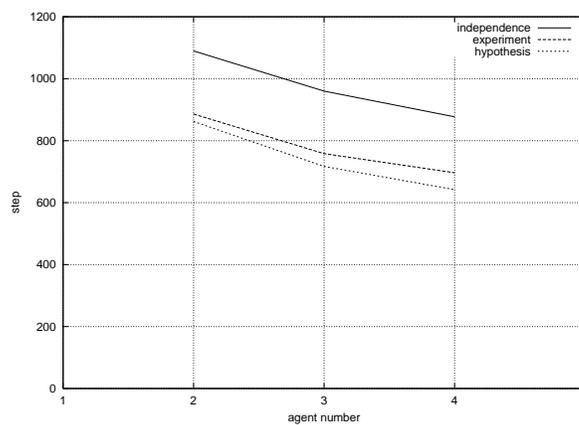


図 5.5: 数人の探索  $51 \times 51$

また, 表 5.4 に, パフォーマンス比を示す.

この結果から, 人数が増えることにより, 共有認知空間を用いた協調の効果はより大きくなることを見ることができる.  $51 \times 51$  サイズの迷路では, 二人での探索は, 実験空間だけ共した状態で, 約 12%, 実験空間と仮説空間を共有した状態では, 約 20% 探索効率が上昇している. 三人の探索はそれぞれ約 14% と, 約 21%. 四人の探索では約 17%, 25% となっている. これは, 人数が大きくなることによって, 協調の効果は減少すること無く, 少しずつではあるが確実に協調の効果が増していくということを示している. そし

表 5.4: 数人の探索のパフォーマンス比  $51 \times 51$

人数	独立条件	実験空間	実験空間と仮説空間
2	1	0.881	0.805
3	1	0.856	0.786
4	1	0.832	0.745

て, 推定コストの更新値の共有については, 人数が増えることによって, 実験空間を共有することによる協調よりも, 探索に有効になってくることを確認した. また, 図 5.5 から, 迷路サイズに対してある一定のステップ数に収束していくことが予想され, ある迷路サイズに, ある共有の段階を与えたときに必要とされる人数は, ある程度決定できることであることがわかった.

# 第 6 章

## まとめ

### 6.1 貢献

本論文では, MARTA\* に共有認知空間の概念を用いて, 協調の段階を分けた実験的な検証を行った.

本論文では, 実世界に実際に実装していくための協調について, 共有認知空間という概念を用いた方法があるということを, 迷路の実時間探索という題材を用いて実験的に示した.

実験空間だけを共有した探索では, MARTA\* の問題点の一つである移動候補地の重複を明示的に解決することができる. また, 共有認知空間を統制することにより, 評価値の過大評価の問題も, 共有する認知空間の違いで制御可能であることを示した. そのような見方から, MARTA\* についての改良についての見識を示した. 実験結果から, ヒューリスティック値の交換は計算機上の情報交換としては有効であるが, 実世界に実装する場合, 探索の大きな助けにはなり得ないことがわかった.

そして, 共有認知空間を用いた協調のデザインは, 今後実用化される様々なものに必要なデザインであり, そのための指針を示した.

### 6.2 今後の課題

さらに詳細な共有認知空間ごとに現れる情報交換については今後の課題と言える. また, どのくらいのサイズにどの程度の性能を持つエージェントがどのくらい必要かを知

るための実験を伴う検証が必要である。なぜなら、このような機能を持つエージェントが一つ作られるのにも、コストがかかり、情報の交換も複雑になるためである。このことから、サイズや問題に適した、有効エージェント数を知ることも今後の課題である。また、迷路問題だけでなく、15 パズルなどの問題にも共有認知空間を用いた探索を行い、どのような傾向を持つ問題により効果的に共有認知空間を用いた協調の効果が現れるのかを実験的に検証しなければならない。さらに、探索問題に限らず、実社会の協調作業に目を向けた研究が必要である。

## 参考文献

- [亀田 97] 亀田 達也 : 合議の知を求めて, グループの意思決定, 共立出版, 1997.
- [北村 96] 北村 泰彦, & 寺西 憲一, & 辰巳 昭治 : マルチエージェント実時間探索における組織化とその評価, 人工知能学会誌, Vol. 11, No. 3, pp.470-477, 1996.
- [Klayman 87] Klayman, J., & Ha, Y.-W. : Confirmation, disconfirmation, and information in hypothesis testing. *Psychological Review*, 94, pp.211-228. 1987.
- [Knight 93] Knight, K. : Are many reactive agent better than a few deliberative ones?, *Proc. 1993 Int. Joint Conf. on Artificial Intelligence*, pp.432-437, 1993.
- [Korf 85] Korf, R. E. : Depth-first iterative-deepening, An optimal admissible tree search, *Artif. Intell.*, Vol. 27, pp.97-109, 1985.
- [Korf 90] Korf, R. E. : Real-time heuristic search, *Artif. Intell.*, Vol. 42, pp.189-211, 1990.
- [Laughlin 85] Laughlin, P. R., & Futoran, C. G. : Collective induction, Social combination and sequential transition, *Journal of Personality and Social Psychology*, 48, pp.608-613, 1985.
- [三輪 00] 三輪 和久 : 共有認知空間の差異が協調的発見に与える影響, 人工知能学会誌, vol. 15, No. 5, pp.854-861, 2000.
- [Nilsson 71] Nilsson, N. J. : *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill, New York, 1971.

[Stuart 95] Stuart J. Russell, & Peter Norvig : Artificial Intelligence, A Modern Approach, Prentice-Hall, Inc., 1995.

[Wason 60] Wason, P. : On the failure to eliminate hypotheses in aconceptual task, Quarterly Journal of Experimental Psychology, 12, pp.129-140, 1960.