

マルチエージェントシステムの共生進化モデルの構築

江口 徹[†] 平澤 宏太郎[†] 古月 敬之[†]

近年、複雑化かつ巨大化するシステムを柔軟に制御・処理する手段として、人間に代わる実行主体であるエージェントを基本単位とした、マルチエージェントシステム(MAS)に関する研究が行われている。また、MASの新たな学習・進化の方式として、生態系に見られる共生現象を考慮した共生学習進化型マルチエージェントシステム(Masbiole)が提案されている。本論文ではMasbioleの2つの方式のうち、共生進化方式に注目し、これをMASの代表的例題であるタイルワールドモデルに適用し、共生進化モデルについての詳細な検討を行う。

Construction of Symbiotic Evolutional Model in Multiagent Systems

TORU EGUCHI,[†] KOTARO HIRASAWA[†] and TAKAYUKI FURUZUKI[†]

Recently, Multiagent Systems (MAS) which regard an "agent" as the basic element in the systems have been studied to control large and complex systems efficiently. As one of these studies, Multiagent Systems with Symbiotic Learning and Evolution named Masbiole have been proposed and studied introducing symbiotic phenomena in the ecosystem into MAS. In this paper, as an application of Masbiole, the method of symbiotic evolution has been applied to the tile-world model which is a test bed of MAS study. It is clarified that the proposed symbiotic evolutionary model is effective from the simulations.

1. はじめに

複雑化したシステムを柔軟に制御/処理するため、従来より人工知能や人工生命を応用した様々な研究がなされている。これらは設計者による中央集権的な制御システムよりもむしろ、複数の自律の実行主体(エージェント)による自律分散制御システムのほうが優れた手法であるという考えに基づいている。

このような流れの中で、複数のエージェントによってシステムの問題を処理していくマルチエージェントシステム(Multiagent Systems: MAS)が研究されてきた^{1)~3)}。MASではそれぞれ独自の目的を持つエージェントが社会を形成し、相互作用を及ぼし合うことにより、問題(タスク)の効率的な処理を図る手法である。またMASの持つ社会性に着目した生命現象や社会現象をモデル化する研究も行われている。

上記の背景に基づき、生態系に見られる共生現象⁴⁾を考慮したMASの学習/進化モデルである共生学習進化型マルチエージェントシステム(Multiagent Systems with Symbiotic Learning and Evolution: Masbiole)

が提案され、研究がなされている^{5)~7)}。Masbioleは搾取や競争といった共生現象に基づき、各エージェントが自分と相手エージェントの利害得失を考慮するように学習/進化を行う共生学習進化方式を用いる。すなわち各エージェントが自身の評価のみを考慮して学習/進化を行う従来のMASに比べ、Masbioleではエージェントの共生学習進化によって、従来のMASでは得られなかった複雑な創発現象や柔軟で効率的な問題の解決を実現することが可能になる。

従来のMasbioleの研究^{5)~7)}では適用するモデルの単純さから、Masbioleの特徴であるエージェントの多様な「共生的挙動」、複雑な創発現象の解析、およびMasbioleの問題解決への有効性に関する検討が十分に行われていなかった。そこで本論文では、Masbioleの進化方式に注目し、新たにエージェントの複雑な動的挙動を表すテストベッドであるタイルワールドモデル⁸⁾を提案、適用し、エージェントの意思決定のメカニズムとして進化論的計算手法であるGenetic Network Programming(GNP)^{9),10)}を用いて上記の詳細な検討を行っている。

本論文の構成は以下のとおりである。2章ではMasbioleの基本概念、Masbioleの共生進化のアルゴリズム、および共生パレート解について要約する。3章で

[†] 早稲田大学大学院情報生産システム研究科
Graduate School of Information, Production and Systems, Waseda University

は提案する対戦型のタイルワールドを用いた Masbirole のシミュレーションを行う．そして 4 章では結論を述べる．

2. 共生学習進化型マルチエージェントシステム (Masbirole)

2.1 基本概念

従来の MAS では，エージェントは目的の達成度である行動評価 (Evaluation) を基に行動戦略 (Strategy) を変更し，問題に適應する．その際互いに相互作用を及ぼし合うことで，MAS 全体として問題の効率的な処理を実現する．しかし，エージェントは自分の利益のみを考慮する (個人合理性) ため，相互作用によってエージェント間に利益の衝突が生じ競争解に陥ってしまうことがある．この競争解をナッシュ均衡解 (Nash Equilibrium) といい，これに陥ることによって MAS の挙動の柔軟性や多様性が失われるという弊害が生じる．

本論文で用いる Masbirole では，エージェントが搾取や競争といった生態系の共生現象にヒントを得た「自分と相手の利害得失を考慮する (集団合理性) 」，共生学習進化を行う．ここで学習とは 1 個の個体で構成されたエージェントが個体の行動戦略を変更することであり，進化とは複数の個体群 (集団) で構成されたエージェントが集団の行動戦略を変更することと定義する．すなわち共生学習進化はエージェントの構成により共生学習方式と共生進化方式に分類される．共生学習進化では各エージェントに相手エージェントに対する共生関係 (Symbiotic Relation) が設定され，エージェントはそれを考慮しながら行動戦略の変更を行う．その結果，多様な共生関係に基づく学習/進化が複雑に相互作用してナッシュ均衡解を回避し，柔軟で多様な問題の解決を可能とする．したがって，基本的には Masbirole は複雑なエージェント間の共生関係に基づく学習/進化によるシステムの挙動を検討する数理モデルであり，企業競争モデル等の社会モデルの構築に資することを目的とする．また共生関係を「自分と相手の利益を考慮する」共利に設定することにより，ナッシュ均衡解より適切な解を求める手法としても Masbirole を適用することが可能である．図 1 に以上の議論に基づいた Masbirole と従来の MAS の基本構造を示す．

共生関係は自エージェントおよび相手エージェントの行動評価の改善/改悪の組合せから表 1 に示す共利 (Mutualism) ，競争 (Competition) ，搾取 (Predation) ，利他 (Altruism) ，自改善 (Self Im-

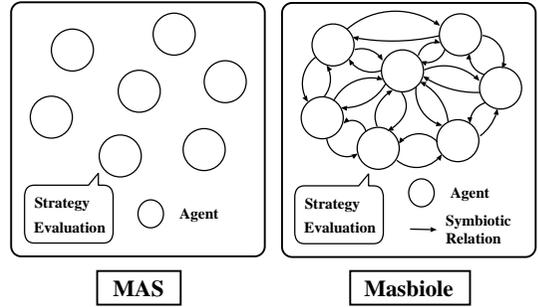


図 1 Masbirole と MAS の基本構造
Fig. 1 Basic structures of Masbirole and MAS.

表 1 自エージェント s の相手エージェント o に対する共生関係
Table 1 Symbiotic relations of self agent s toward opponent agent o .

Symbiotic Relation	Agent s	Agent o
Mutualism	Improve	Improve
Competition	Deteriorate	Deteriorate
Predation	Improve	Deteriorate
Altruism	Deteriorate	Improve
Self Improvement	Improve	-
Self Deterioration	Deteriorate	-

provement) ，自改悪 (Self Deterioration) の 6 種類が定義されている．これらは任意の 2 個のエージェント間に設定され，共生学習進化はシステム内の共生関係ごとに順次行われることになる．たとえば搾取の共生学習の場合，エージェント (個体) は自エージェントの行動評価が改善し，相手エージェントの行動評価が改悪するように行動戦略を変更することと考える．また自改善および自改悪は相手を考慮せず自分のみを考慮する共生関係として定義しており，特に自改善は従来の MAS の学習/進化の方式に対応する．つまり Masbirole は広義には MAS の学習/進化を含めたモデルとして考えることができる．

Masbirole では設定した共生関係ごとに学習/進化が実行されるが，その際行動戦略を変更するエージェント以外エージェントは行動戦略を固定する．このような局所的な学習/進化の方式をとることで，複雑なモデルにおいてもアルゴリズムの実装が容易に行える．

2.2 共生進化のアルゴリズム

本節では本論文で用いる進化型 Masbirole の共生進化方式のアルゴリズムについて説明する．進化型 Masbirole ではエージェントが複数の個体群 (集団) で構成され，各個体が行動戦略および行動評価を持つため，エージェントの行動戦略の変更は多目的遺伝的アルゴリズム (Multiobjective Genetic Algorithms: MO-GAs)¹⁾ を応用した進化的手法によって行う．まず共生進化の全体的な流れを以下に示す．

共生進化のアルゴリズム

- 1: 各エージェント間の共生関係およびエージェントの行動戦略を初期設定する.
- 2: 行動戦略の変更を行う自エージェントと、その相手エージェントを選択する.
- 3: 自エージェントの行動戦略を遺伝的操作によって変更する.
- 4: 各エージェントの個体を問題空間上で実行し、行動評価を計算する.
- 5: 多目的ランキング方式を用いて相手エージェントに対する共生関係を考慮した、自エージェントの各個体のランクを計算する.
- 6: ランクに基づき、自エージェントの次世代へ残す個体群を選ぶ.
- 7: 終了条件を満たせば、エージェントの各個体が獲得した解を共生パレート解として終了する. 満たさない場合は手順2へ戻る.

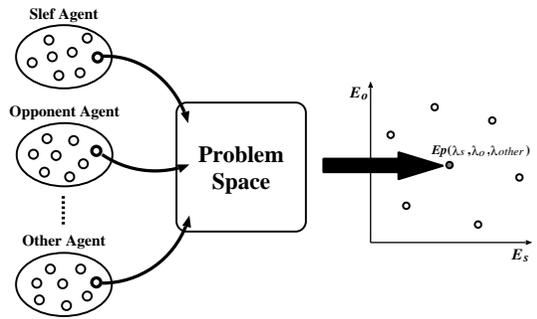


図2 行動評価点の計算例

Fig. 2 An example of calculation of evaluation points.

次に手順4および5について詳しく説明する. 手順4および5では、遺伝的操作によって新たに生成された自エージェントの個体群に対して、それと相手およびそれ以外のエージェントの個体を組み合わせて個体対とする. そしてその個体対を問題空間上で実行して得られた行動評価に対して、共生関係に基づくMOGAsの多目的ランキング計算を行い、ランクの良い自エージェントの個体を次世代の行動戦略として残していく.

ここで、自エージェントを s 、相手エージェントを o 、システム内のそれ以外のエージェントを $other$ として説明を行うことにする. 各エージェントに対して、

- λ_i : エージェント i の個体の行動戦略
- E_i : エージェント i の個体の行動評価値

と定義すると、エージェント $s, o, other$ から選んだ行動戦略(個体)の組合せ $(\lambda_s, \lambda_o, \lambda_{other})$ に対して、エージェント s およびエージェント o の個体の行動評価は式(1)のようになる.

$$\begin{cases} E_s = E_s(\lambda_s, \lambda_o, \lambda_{other}) \\ E_o = E_o(\lambda_s, \lambda_o, \lambda_{other}) \end{cases} \quad (1)$$

式(1)の行動評価の組合せ (E_s, E_o) を行動評価点 $E_p(\lambda_s, \lambda_o, \lambda_{other})$ という. 図2は様々な行動戦略の組合せを問題空間上で実行し、得られた行動評価点を (E_s, E_o) 平面上にプロットしたものである.

図2のような行動評価点に対し、エージェント s の o に対する共生関係に基づくランクの計算を行う. 各行動評価点のランクは以下のルールに基づき計算する.

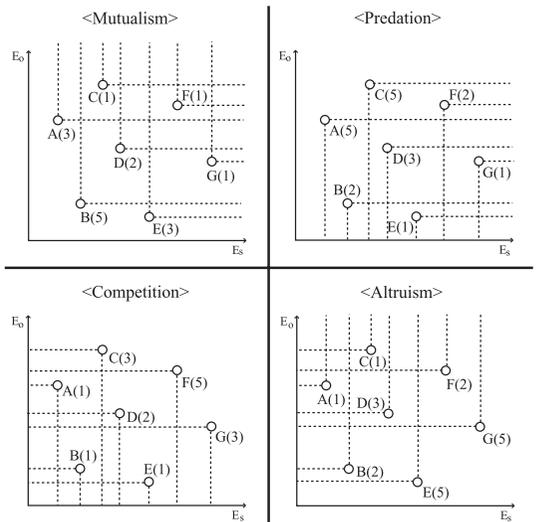


図3 各共生関係におけるランキングの例

Fig. 3 Examples of ranking for each symbiotic relation.

ランク計算のルール

ある行動評価点が、エージェント s のエージェント o に対する共生関係の下で R 個の他の行動評価点に支配されているとき、その行動評価点のランクを $R + 1$ と計算する.

図3に行動評価点 $E_p(\lambda_s, \lambda_o, \lambda_{other})$ の共利、競争、搾取、利他の各場合のランク計算の例を示す. 図3中の A, B, C, \dots は各行動評価点、()内の数字が行動評価点のランクを示しており、行動評価の改善は行動評価の増加として考える. たとえば共利の場合、行動評価点 C, F, G は E_s, E_o が改善される方向に1つも他の行動評価点を含んでいない(支配されていない)ので、ランクは1となる. 行動評価点 D は F によって支配されているのでランクは2、同様に A, E は他の2つの行動評価点に支配されているのでランクは3、 B は4つの行動評価点に支配されているのでランクは5と計算される. また自改善および自改善

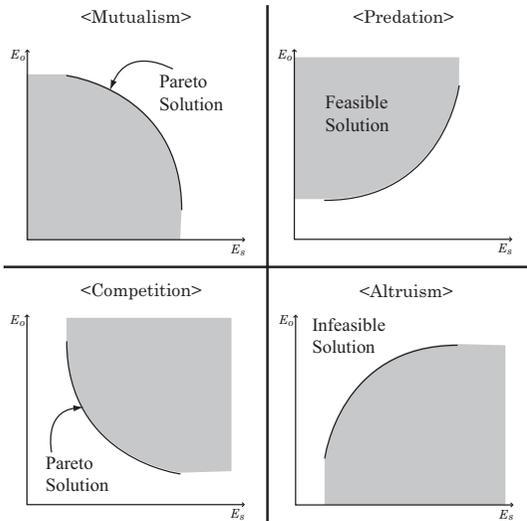


図4 各共生関係の共生パレート解

Fig. 4 Symbiotic Pareto solutions for each symbiotic relation.

の場合は, E_s の高い(低い)行動評価点から順にランクを計算する.

2.3 共生パレート解

共生進化によってエージェントが獲得した解は自エージェントの個体と相手エージェントの個体の行動評価を考慮しており,パレート最適性を満たす.このような Masbiole のパレート最適解を共生パレート解と呼び,これは次のように定義される.

Masbiole の共生パレート解

自エージェント s の行動戦略の集合 Λ_s と相手エージェントおよびその他のエージェントの行動戦略 $\lambda_o, \lambda_{other}$ に対して,下記の条件を満足する $\lambda_s \in \Lambda_s$ が存在しないとき, $\lambda_s^* \in \Lambda_s$ は $\lambda_o, \lambda_{other}$ のもとでの自エージェント s の相手エージェント o に対する共生パレート解である.

<条件>

$$E_s = E_s(\lambda_s, \lambda_o, \lambda_{other}) \sim E_s(\lambda_s^*, \lambda_o, \lambda_{other}) \quad (2)$$

かつ

$$E_o = E_o(\lambda_s, \lambda_o, \lambda_{other}) \approx E_o(\lambda_s^*, \lambda_o, \lambda_{other}) \quad (3)$$

式(2),式(3)の記号 \sim および \approx は,エージェント s のエージェント o に対する共生関係に対して次のようになる.

共利 $\implies \sim : >, \approx : >$

競争 $\implies \sim : <, \approx : <$

搾取 $\implies \sim : >, \approx : <$

利他 $\implies \sim : <, \approx : >$

これらの各共生関係の共生パレート解のイメージ図

を (E_s, E_o) 平面上に描くと図4のようになる.

3. シミュレーション

本章では Masbiole の共生進化方式の例題として,タイルワールドモデルを用いた3種類のシミュレーションを行う.シミュレーション1では2個のエージェント間の比較的単純なモデルから得られるエージェントの共生的挙動について検討する.シミュレーション2ではより複雑なモデルを実現するためエージェント数を3個に増やし,共生進化によって得られる複雑な創発現象の検討を行う.シミュレーション3では,シミュレーション1と同じ2個のエージェントのモデルを用いて,各エージェントが共利をとる Masbiole と従来の MAS の性能比較を行う.

3.1 基本設定

3.1.1 シミュレーションモデル

シミュレーションモデルであるタイルワールドモデルとは2次元格子状の仮想的な環境であり,動的環境のテストベッドとして知られている⁸⁾.タイルワールドは動的素子(ユニット),タイル,障害物,穴からなり,それぞれが格子の1マス(セル)を占める.この動的素子のことを一般にはエージェントと呼ぶが,本論文ではこの「エージェント」と Masbiole を構成する「エージェント」を混同してしまう可能性がある.そこでこれらを区別するために,本論文ではタイルワールド上の動的素子を「ユニット」と呼び, Masbiole のエージェントと区別する.

従来のタイルワールドに関する研究で使用されてきたモデルは,タイルを穴へ落とすタスクを最短時間で処理することを前提とした構成になっており,共生関係に基づくエージェントの挙動を実現できる構成ではない.したがって本論文では,より Masbiole に即したモデルとして,「対戦型タイルワールドモデル」を新たに提案し使用している.シミュレーションで用いる対戦型タイルワールドモデルの1例を図5に示す.図5では,22×22の大きさの環境内に2種のエージェント(エージェント1,2)に属するユニットが3個ずつ分散して存在しており,相手ユニットと対戦を行う.穴は各エージェントにつき5個ずつ分散して存在し(Hole 1, Hole 2),穴の添え字はその穴が属するエージェントの添え字を表している.タイルには得点用タイル(Tile S)と妨害用タイル(Tile D)の2種類があり,それぞれ10個ずつ各穴の1~5近傍に存在する.これらのタイルをユニットが穴に落とすことで各エージェントのユニットは得点する.Tile S と Tile D の違いであるが,たとえば Hole 1 に Tile S を落

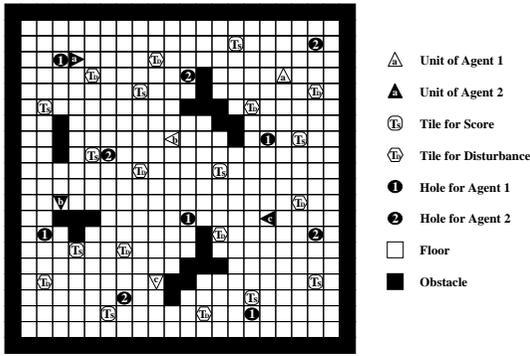


図5 シミュレーションで用いるタイルワールドの例

Fig. 5 An example of tile-world model in the simulations.

とすとエージェント1の得点になるが、Tile Dを落としてもエージェント1の得点にはならず、穴が埋まり得点源が消滅し得点の上昇が困難になる。以上の構成によって、共利、競争、搾取、利他等の共生関係に基づくエージェントの多様な共生的挙動の実現が可能になる。

3.1.2 GNPによるエージェントの構成

本論文では共生進化を行うエージェントを進化論的計算手法の1つである Genetic Network Programming (GNP) を用いて構成する。タイルワールドをはじめとするマルチエージェントに関する研究としては、同じく進化論的計算手法である Genetic Programming (GP) を適用した研究¹²⁾がこれまで一般的であった。しかしながら木構造のGPに比較し、GNPはネットワーク構造であるため、過去の行動や情報に基づいてエージェントの行動系列を生成できる。したがって、タイルワールドのような動的なモデルに対して、GNPではGPより効率的な処理を期待でき、また実際にGPより良い結果を得ている⁹⁾。

GNPは判定ノードと処理ノードがネットワーク状に接続された構造をしており、判定ノードは環境からの情報を判定し判定結果に従って次ノードへ遷移させる役割を持っている。ノード関数の内容(判定内容)は設計者によって問題に即したものが用意される。処理ノードではエージェントの問題空間上における行動/処理を実行する。GNPでは、GPの終端ノードのように行動/処理の後に根ノードへ戻ることはなく判定および処理に応じて次ノードへ遷移する。このようにGNPでは特定の初期起動ノードから処理が始まり、判定ノードでは判定結果に従い次ノードへ遷移し、処理ノードでは行動/処理を行うというノード遷移をネットワーク構造に従って繰り返し、これによってエージェントの行動系列を生成する。

図6に、GNPによるエージェントの構成とユニッ

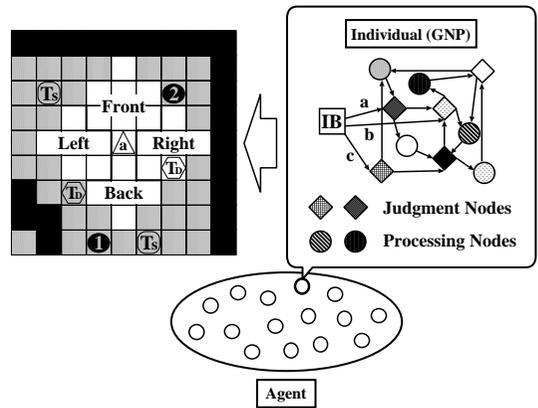


図6 シミュレーションの基本構造

Fig. 6 A basic structure of simulation.

トへの適用のメカニズムを示す。図6において、ユニットは自分を中心とした n 近傍分の視野を有し、視野内にある環境の情報を GNP の判定ノードに取り入れることができる。また各ユニットは位置および前後左右の相対的な方向の情報を持つ。同じエージェントに属するユニットは同一の GNP プログラム(各エージェントの1個体)によって行動系列が生成される。各ユニットには行動する順番が設定され、すべてのユニットが1回行動することを1ステップとし、定められたステップ数行動することを1エピソードと定義する。

本シミュレーションでは、各ユニットは前(MF)、後(MB)、左(ML)、右(MR)のセルへ動け、その場にとどまる(SH)ことができる。また各ユニットは、自分の前(SF)、後(SB)、左(SL)、右(SR)のセルに何があるか、および視野内で最も近くにあるタイル(NTS, NTD)、穴(NH i ($i = 1, 2, \dots$)), ユニット(NU i ($i = 1, 2, \dots$))の情報を知ることができる。したがって、たとえば図5の環境におけるGNPの判定/処理ノード群としては、表2に示すものが用意される。表2では、判定ノードSF, SB, SL, SRは8個の判定結果(Tile S, Tile D, ユニット1, 2, Hole 1, 2, 床, 障害物)を、NTS, NTD, NH1, NH2, NU1, NU2は6個の判定結果(前, 後, 左, 右, 複数, なし)をそれぞれ返す。

またエージェントの行動戦略の変更は GNP の遺伝的操作(交叉, 突然変異, エリート保存戦略, トーナメント選択)によって行う。

GNPの全ノードに対し確率 P_c で特定のノードを選択し、そのノードからの接続を親 GNP 間で交換し子 GNP を生成する。GNPの全ノードの接続に対し確率 P_m で特定の接続を選択し、選択された接続をランダムに変更する。

表 2 ノード関数群の例

Table 2 Examples of node function sets.

Node type	Node functions	Branch
Judgment node	SF, SB, SL, SR	8
	NTS, NTD, NH1	6
	NH2, NU1, NU2	
Processing node	MF, MB, ML, MR, SH	1
Initial boot node	IB	1

3.1.3 行動評価関数

エージェント i の各個体の行動評価 E_i は、式 (4) によって計算する。

$$E_i = k_T \times T_i + \sum_{l=1}^L (k_{TN}^l \times TN_i^l) + k_C \times C_i + k_N \times N_i \quad (4)$$

ここで式 (4) 中の各項は

- T_i : 穴 i に落とされた Tile S の数
- TN_i^l : 穴 i の l 近傍にある Tile S の数 ($l = 1, 2, 3, 4$)
- C_i : エージェント i のユニットが移動した平均ステップ数
- N_i : エージェント i の個体 (GNP) の 1 ステップあたりの平均ノード遷移回数
- k_T, k_{TN}^l, k_C, k_N : 重み付け係数

を意味する。 T_i, TN_i^l は各エージェントが Tile S を穴へ近づけて落とすというタスクの達成度を表しており、 C_i, N_i は行動評価点を (E_s, E_o) 平面上に広く分散させ、進化の局所的な収束を避けるために考慮している。各重み付け係数は、 k_T および k_{TN}^l に関しては、それぞれ $k_T=200, k_{TN}^1=80, k_{TN}^2=60, k_{TN}^3=40, k_{TN}^4=20$ に設定した。すなわち、Tile S を穴へ 1 近傍近づけるごとにその穴が属するエージェントは 20 ずつ行動評価を増加できる。Tile S を穴へ落とすと 200 の行動評価を獲得する。したがって、エージェントは Tile S を穴へ近づけ、最終的に穴に落とすように進化する。また k_C, k_N はそれぞれ 1 に設定した。図 5 を見ると、初期状態ですでに穴 1, 2 の 1~4 近傍に Tile S が 1 個ずつ存在しており、ユニットが何もしなくてもエージェントは 200 程度の行動評価を得ることになる。これは競争や利他のように、行動評価を改悪させるような行動戦略の変更が行われることを可能にするためである。本シミュレーションでは行動評価の増加を改善としている。

3.2 シミュレーション 1

3.2.1 シミュレーション条件

シミュレーション 1 では図 5 のタイルワールド環

表 3 シミュレーション 1 のパラメータ

Table 3 Parameter conditions in simulation 1.

最大世代数 = 2000/共生関係 (合計 4000 世代)
エージェントのサイズ (個体数) = 300
エリート保存個体数 = 1
1 エピソードの最大ステップ数 = 50
各ユニットの視野 = 5
交叉確率 $P_c = 0.1$, 突然変異確率 $P_m = 0.01$
子個体の交叉/突然変異比 = 2 : 3
トーナメントサイズ = 2
GNP のノード数 = 15/各種類+初期起動ノード

表 4 シミュレーション 1 における各エージェントの共生関係

Table 4 Symbiotic relation of each agent in simulation 1.

Case	Agent 1 → Agent 2	Agent 2 → Agent 1
1	Mutualism	Mutualism
2	Competition	Competition
3	Altruism	Altruism

境を用いて 2 個のエージェント間の基本的な共生進化モデルによって、エージェントの進化の様子、共生パレート解および共生的挙動について検討を行う。GNP の構成に関しては、表 2 に示すノード群を用いる。進化はエージェント $1 \rightarrow 2 \rightarrow 1 \dots$ の順に交互に行い、進化のための各パラメータは表 3 のように設定した。

以上の条件の下、各エージェントの共生関係を表 4 の 3 通りに設定してそれぞれ実行する。

3.2.2 結果と考察

図 7 に表 4 の各 Case のシミュレーション結果を示す。図 7 は、Case ごとにエージェントの個体の行動評価の平均 (縦軸) の世代 (横軸) ごとの推移、各エージェントが獲得した共生パレート解 (E_1, E_2) 平面上に図示、各エージェントが穴に落としたタイル数の平均を示している。共生パレート解は横軸が自エージェントの行動評価、縦軸が相手エージェントの行動評価をそれぞれ表しており、落としたタイルの数を表す棒グラフの横軸の各系列は、左から (1) 自分の利益、(2) 相手の利益、(3) 自分の利益の消滅、(4) 相手の利益の消滅、を順にそれぞれ示す。たとえば、エージェント 1 の場合は横軸の系列は左から順に「Hole 1 に落とした Tile S 」「Hole 2 に落とした Tile S 」「Hole 1 に落とした Tile D 」「Hole 2 に落とした Tile D 」の数の平均となる。また結果はすべて、乱数系列 5 通りに対する実行結果の平均である。

各 Case の共生進化による行動評価の推移について考察する。エージェント 1, 2 がともに共利をとる Case 1 では、双方の行動評価が改善されほぼ同じ値で収束している。これは互いに自分と相手の利益を考慮することによって各々の進化に相乗効果をもたらし

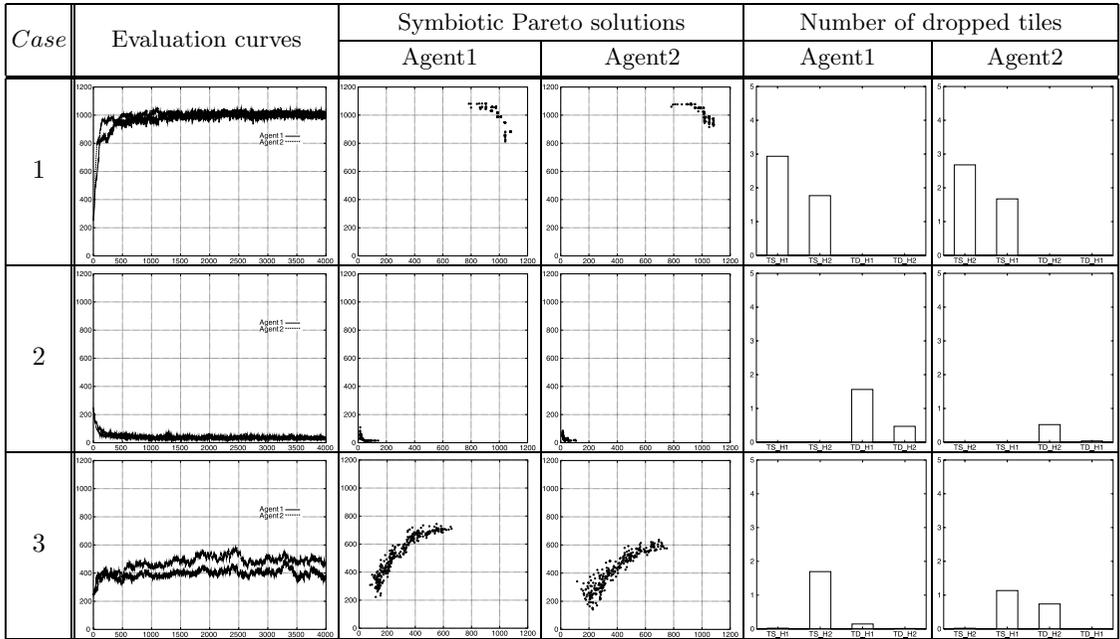


図 7 シミュレーション 1 の各 Case の結果

Fig. 7 Results of each symbiotic evolution in simulation 1.

たためと考えることができる．エージェント 1, 2 が競争をとる Case 2 では, 双方の行動評価が改悪され, ともに 0 に近い値で収束している．これも Case 1 と同様に, 互いに自分と相手の行動評価の改悪を考慮することで各々の進化に影響を及ぼし合い, 早い段階で収束している．エージェント 1, 2 がともに利他をとる Case 3 では Case 1, 2 とは異なりそれぞれの行動評価は一定の値に収束せずに行動評価値 400 ~ 500 付近で振動している．これは各エージェントの共生関係が相反するため, 一定の値に収束せず交互に行動戦略を変更するたびに振動を起こしているからである．

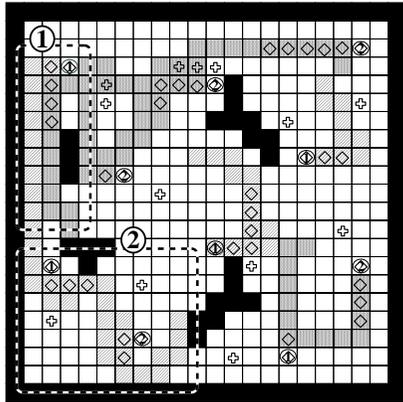
次に, 各 Case で得られた共生パレート解について考察する．Case 1 では各エージェントとも, 図 4 のイメージ図のような共利の共生パレート解を得ていることが分かる．それに対し Case 2 ではいずれもイメージ図のような競争の共生パレート解が得られていない．これは式 (4) において行動評価が負の値をとらないことから, 共生パレート解が 0 以下の小さい値をとれず原点付近に集中してしまったと考えられる．Case 3 では各エージェントともイメージ図のような利他の共生パレート解を得ていることが分かる．

最後に図 7 のエージェントが穴に落としたタイルの数の棒グラフ, そして図 8 および表 5 を用いて, 進化によってエージェントが獲得した共生的挙動について検討を行う．ここで図 8(a) ~ (c) は各 Case にお

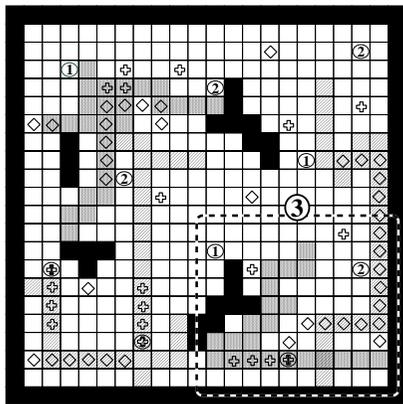
けるユニット, タイルがタイルワールド内を移動した軌跡と, その中で特に注目すべき部分を点線枠で囲み, 右側に一定ステップごとのスナップショットとして示している．表 5 は図 8 の各スナップショットにおいてエージェントの GNP (個体) が使用したノード関数群の頻度を示している．

まず Case 1 の共利対共利について検討する．図 7 および図 8 (a) の軌跡から分かるように, 各エージェントは共利によって自分のみならず相手の利益となるようにタイルを穴へ落とす挙動 (Tile S → Hole 1, Tile S → Hole 2) を見せている．その中でも特に点線枠で囲んだ 2 つの部分についてスナップショットを見ると, ①では 2 ~ 9 ステップにおいてエージェント 2 のユニット *b* が Tile S を Hole 1 の近くまで運び, その後 24 ~ 34 ステップにおいてエージェント 1 のユニット *c* がそれを穴へ落とすという, タスクの共同作業を行っていることが分かる．この挙動は, 共利の共生進化が単に自分と相手の行動評価が改善するような挙動を獲得するだけでなく, 相手のタスクを助けるような協力的な挙動も獲得しうることを示している．①の挙動に対して表 5 を見ると, エージェント 2 は Tile S を Hole 1 へ近づけるために SF, NTS, NH1 によって Tile S と Hole 1 の位置情報を取り入れつつ行動し, エージェント 1 は SR を主に用いて自分の右側のセルに注意しながら前進し, 右側にタイルを発見した時点 (33

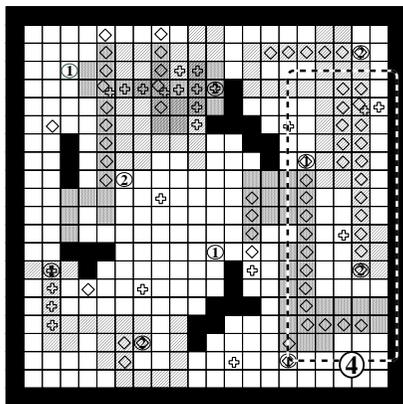
- Trajectories of Units of Agent 1
- ◇ Trajectories of Tile for Score
- Trajectories of Units of Agent 2
- ⊕ Trajectories of Tile for Disturbance
- ① Hole 1
- ② Hole 2



(a) Case 1



(b) Case 2



(c) Case 3

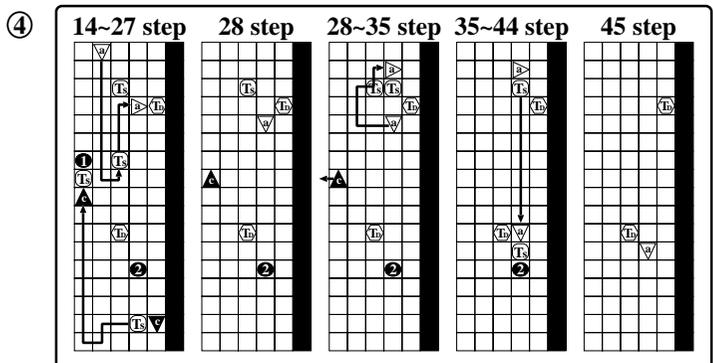
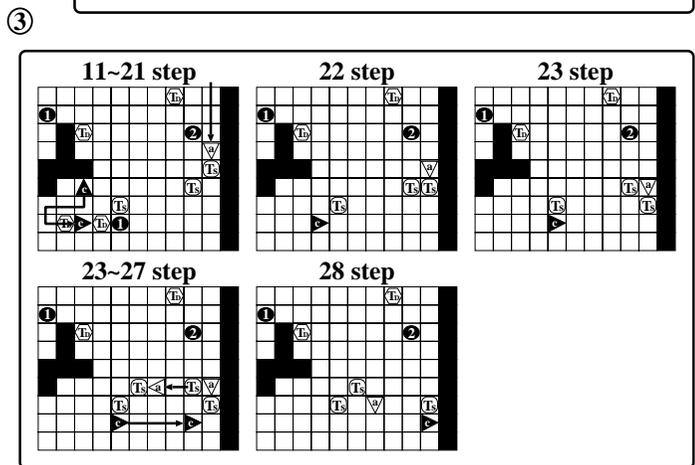
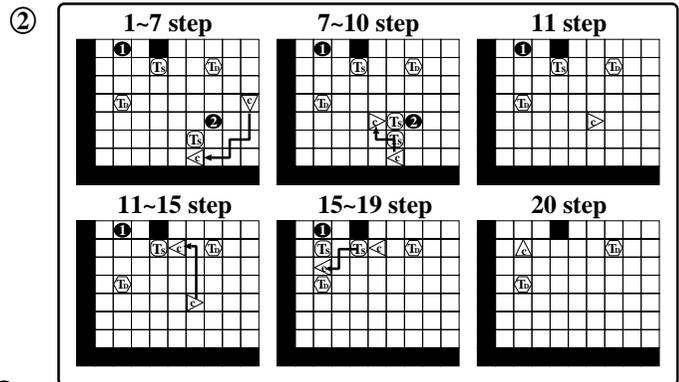
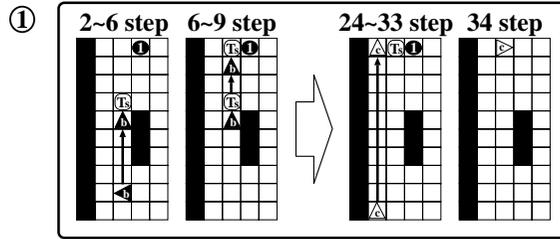


図 8 各 Case における典型的な共生的挙動
Fig. 8 Typical symbiotic behaviors of each Case.

表 5 スナップショットでエージェントの GNP (个体) が使用したノード関数群の頻度
Table 5 The number of usage of node functions of GNP in the snap shot of each Case.

Case	Behavior	Agent	Step	Judgment nodes										Processing nodes					
				SF	SB	SL	SR	NTS	NTD	NH1	NH2	NU1	NU2	MF	MB	ML	MR	SH	
1	①	1	24~34	1	0	3	14	3	16	0	1	7	1	10	0	0	1	0	
		2	2~9	3	0	6	3	4	4	5	4	4	4	6	0	1	1	0	
1	②	1	1~11	0	2	0	4	0	1	4	9	5	2	3	0	2	5	1	
			12~20	2	3	1	6	8	5	3	6	1	2	2	0	4	3	0	
2	③	1	11~23	1	1	0	1	5	3	1	0	2	5	8	0	2	3	0	
			24~28	2	1	2	1	2	2	4	1	0	3	2	0	1	1	1	
2	③	2	11~22	0	1	2	2	1	3	3	2	0	1	5	0	3	4	0	
			14~27	3	6	1	7	5	1	0	0	8	1	10	0	3	1	0	
3	④	1	28~45	5	4	5	5	6	3	0	4	10	1	11	0	1	6	0	
			2	14~28	1	2	1	0	10	0	0	1	0	1	10	0	1	4	0

ステップ目)で右に進み Tile S を落としていることが分かる。このケースではエージェント 1 と 2 の共同作業に時間的な差があったため、NU1, NU2 といった相手の位置を把握する判定ノード群が有効に働いているかの判断は難しいが、各エージェントともこれらのノードを用いており、相手エージェントの存在に関する情報がある程度は有効に働いているものと考えられる。②は①とは対照的にエージェント 1 のユニット c が単独で自分と相手の利益となるように Tile S を Hole 1 および 2 に最適な経路で落としていく挙動である。そのため表 5 を見ると、1~11 ステップでは NH2 を主に利用して Hole 2 の位置をつねに確認しつつ、ユニットが右回りに動きながら SR によって 7 ステップ目で Tile S を発見し、これらの情報を利用して穴へタイルを落としている。12~20 ステップでは NTS によって Tile S の位置を見つけ、それに近づきつつ穴の情報も NH1, NH2 によって取り入れ、Hole 1 へ Tile S を落とすように行動している。

次に Case 2 の競争対競争の③の挙動について検討する。図 7 からは各エージェントが自分と相手の利益を減少させるようにタイルを穴へ落とす挙動 (Tile D → Hole 1, Tile D → Hole 2) が得られていることが分かる。ところが図 8 (b) を見ると、Tile D を穴へ落とすだけでなく、穴から Tile S を遠ざけるような挙動も見られる。これは行動評価を式 (4) のように穴の 1~4 近傍にある Tile S の数を考慮して計算することから、Tile S を穴から 5 近傍以上へ遠ざけることでも利益の減少が可能とエージェントが判断したためである。ここで点線枠の部分および表 5 に注目すると、エージェント 1 のユニット a は NTS, NTD, NU1, NU2 を利用しタイルおよびユニットの位置情報を得ながら、Tile S を押して 23 ステップ目で 1 個の Tile S を Hole 2 の 5 近傍に移動させている。これに対してエージェント 2 のユニット c は NTD,

NH1, NH2 によって Tile D と穴の位置情報を取り入れ、Tile D を押して 22 ステップ目で Hole 1 へ落とすことでエージェント 1 の得点源を消滅させている。したがって埋まった Hole 1 の近傍も同時に消滅することになり、これを受けて 23~28 ステップでエージェント 1 は NH1, NH2 により穴の位置情報を確認し、Hole 2 の 3 近傍に位置する Tile S を Hole 1 があつた付近へと運び、Hole 2 から遠ざけている。

最後に Case 3 の利他対利他の④の挙動について検討する。図 7 および図 8 (c) の軌跡を見ると、各エージェントとも利他的な挙動として自分の穴に Tile D を落とし、相手の穴に Tile S を落とすような挙動が得られていることが分かる。そして同時に、自分の穴の近傍に存在する Tile S を相手の穴に落とすという、より利他の意味合いの強い挙動も得られている。図 8 の点線枠の部分は、この挙動が相互に影響を及ぼし合った興味深い例を示している。まず 14~28 ステップではエージェント 1 のユニット a は自分の穴 (Hole 1) の近くの Tile S を遠ざけるように、エージェント 2 のユニット c は自分の穴 (Hole 2) の近くの Tile S を遠ざけつつ相手の穴 (Hole 1) へと落とすように、それぞれ動いている。するとエージェント 2 の行動によって Hole 1 およびその近傍が消滅するため、28 ステップ目以降エージェント 1 はいったん遠ざけた Tile S を再び Hole 1 があつた方向へと押し始め、最終的に 45 ステップ目で相手の穴 (Hole 2) へと落とす利他的な挙動を見せている。また表 5 を見ると、エージェント 1 は 14~27 ステップでは SL, SB, NTS 等によって Tile S と Hole 1 の位置情報を確認し、Tile S を遠ざける行動をとり、28~45 ステップでは SF, SL, SR, NTS, NH2 を利用し Tile S を落とすべき Hole 2 を探し、それに落とすように行動していることが分かる。この挙動では NU1 が頻繁に用いられているが、周囲に仲間ユニットがないためこの挙動に対

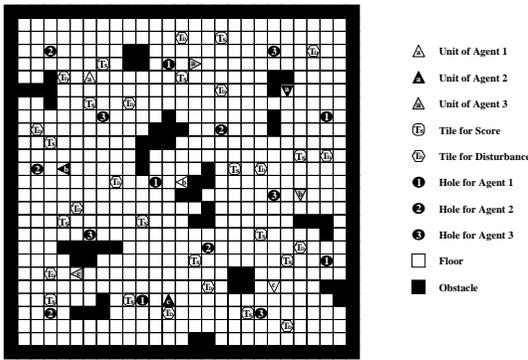


図 9 シミュレーション 2 のタイルワールド環境

Fig. 9 The tile-world environment in simulation 2.

表 6 シミュレーション 2 で用いる GNP のノード関数群

Table 6 Node function sets of GNP used in simulation 2.

Node type	Node functions	Branch
Judgment node	SF, SB, SL, SR	10
	NTS, NTD, NH1, NH2 NH3, NU1, NU2, NU3	6
Processing node	MF, MB, ML, MR, SH	1
Initial boot node	IB	1

しては有効には働いていないと考えられる。エージェント 2 は NTS を主に利用し上記のような挙動をとっている。

このように共生進化によって多様なエージェント間の挙動が見られ、それらはエージェントを構成する GNP が環境および相手エージェントの挙動を入力情報として取り入れることにより生成されている。換言すると、進化によって獲得された GNP の遺伝子情報の中に、共生の概念に由来する部分が断片的であるが獲得できていると考えられる。

3.3 シミュレーション 2

3.3.1 シミュレーション条件

シミュレーション 2 では図 9 に示すタイルワールド環境を用いて 3 個のエージェント間の共生進化モデルについて検討を行う。GNP の構成に関しては、表 6 に示すノード群を用いる。表 6 では、判定ノードとして新たに NH3, NU3 が加わり、SF, SB, SL, SR の判定結果も (Tile S, Tile D, ユニット 1~3, Hole 1~3, 床, 障害物) の 10 個となっている。またエージェント間の進化の頻度を等しくするため、6 通りの共生関係の進化をエージェント 1→2, 2→3, 3→1, 1→3, 2→1, 3→2 の順に、すなわち 6 世代ですべての共生関係の進化が一巡するようにしている。進化のための各パラメータは表 7 のように設定した。

上記の条件に基づき、3 個のエージェント間の 6 個の共生関係を表 8 のように 2 種類 (Pattern) に 2

表 7 シミュレーション 2 のパラメータ
Table 7 Parameter conditions in simulation 2.

最大世代数 = 1500/共生関係 (合計 9000 世代)
エージェントのサイズ (個体数) = 300
エリート保存個体数 = 1
1 エピソードの最大ステップ数 = 50
各ユニットの視野 = 4
交叉確率 $P_c = 0.1$, 突然変異確率 $P_m = 0.01$
子個体の交叉/突然変異比 = 2 : 3
トーナメントサイズ = 2
GNP のノード数 = 15/各種類 + 初期起動ノード

表 8 シミュレーション 2 における各エージェントの共生関係
Table 8 Symbiotic relation of each agent in simulation 2.

Pattern A		Pattern B	
Case 1	Case 2	Case 1	Case 2

M : Mutualism, C : Competition,
P : Predation, A : Altruism

通り (Case) ずつ、合計 4 通りに設定した。表 8 の Pattern A は 6 個の共生関係を様々に設定し共生進化させるモデルであり、Pattern B は相手に対して様々な戦略を有する企業競争モデルである。

3.3.2 結果と考察

図 10 にシミュレーション結果を示す。図 10 は、各 Pattern の Case ごとにエージェントの個体の行動評価の平均 (縦軸) の世代 (横軸) ごとの推移を示している。また結果はすべて、乱数系列 5 通りに対する実行結果の平均である。

Pattern A は 3 エージェント間の共生関係を表 8 のように共利、競争、搾取、利他の複雑な組合せとしたものである。Case 1 と 2 ではエージェント 1→2 の共生関係が異なるだけで、それによって進化にどのような違いが生じるかを検討する。まず Case 1 では図 10 を見ると、エージェント 1 は 2 と 3 に対して共利をとりエージェント 2 から利他の共生進化を受けるため、最も行動評価が良くなっている。一方エージェント 2 は 3 に対する搾取の共生進化が反映されず、両者の行動評価は同程度の値になっている。これは各エージェントが様々な共生関係をとることで、エージェント 2→3 の搾取を抑制する作用が働いたためと考えられる。Case 2 では、エージェント 1→2 の共生関係が共利から競争に変更された結果、エージェント 1 と 2 の行動評価は競争によって Case 1 よりも悪くなっている。しかしながら、この変更には直接的に関係ないエージェント 3 の行動評価が良くなっている。これ

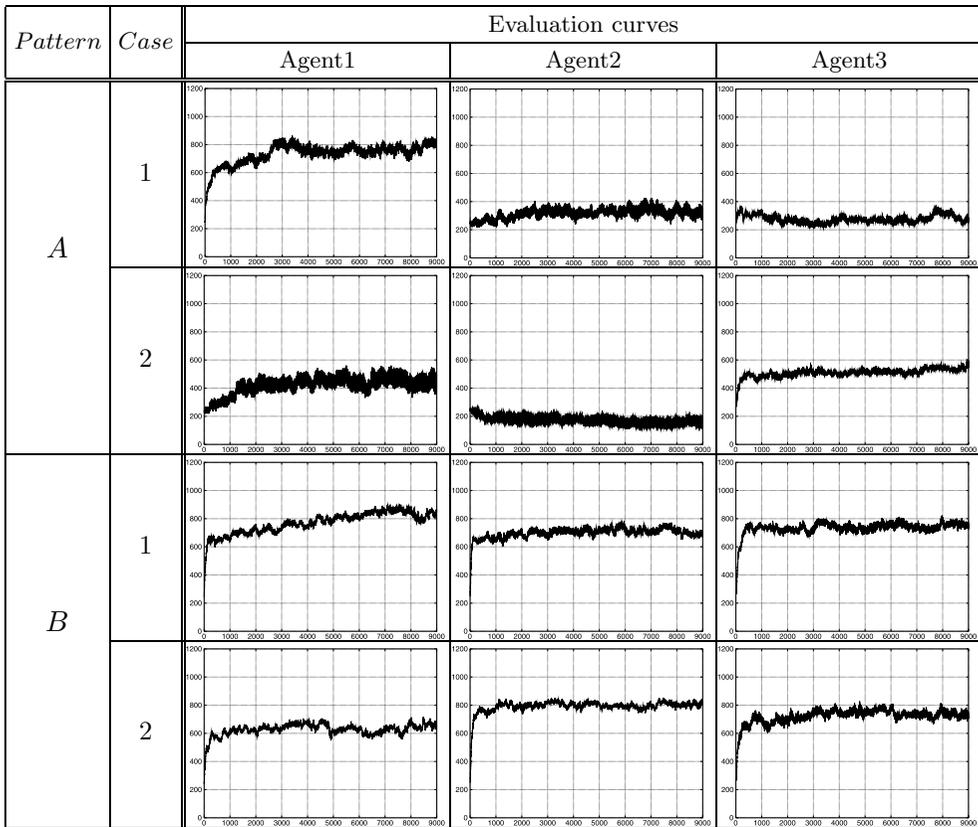


図 10 シミュレーション 2 の各 Case の結果
 Fig. 10 Results of each symbiotic evolution in simulation 2.

はエージェント 1 から 2 に対する競争関係によって、エージェント 1 と 2 の利益の一部がエージェント 3 に還元されたためと考えられる。

Pattern B は各エージェントを企業と見なし、相手企業に対して様々な戦略（共生関係）をとる企業モデルである。ここではエージェント 2 と 3 がお互いに共利をとる協力関係にある。Case 1 ではエージェント 2 と 3 は 1 に対して共利をとりエージェント 1 と協力しようとするが、1 はそれを出し抜く形で搾取をとっている。したがって、相手を淘汰しつつも相手から利益を受けるエージェント 1 が最も行動評価が良くなっている。一方 Case 2 では逆にエージェント 2 と 3 が 1 の搾取を警戒して 1 に対して搾取をとっている。そのためエージェント 1 は 2 と 3 を搾取によって淘汰しようとするが、2 と 3 の協力関係によって行動評価が伸び悩み、エージェント 2, 3 よりも行動評価が悪くなっている。このように複数の相手どうしが協力し合っているグループに対して、うまく出し抜いて搾取をとる場合は有効であるが、その搾取を相手に警戒された場合は逆に行動評価が下がってしまう。こ

表 9 シミュレーション 3 における各エージェントの共生関係
 Table 9 Symbiotic relation of each agent in simulation 3.

Case	Agent1 → Agent2	Agent2 → Agent1
1	Mutualism	Mutualism
2	Self Improvement	Self Improvement
3	Whole Improvement	Whole Improvement

れは、企業が市場経済の中で継続的に利益をあげるためには、どの企業とも協力せずに相手を淘汰する戦略のみでは困難であるという企業モデルの 1 つの構図を説明している。

3.4 シミュレーション 3

3.4.1 シミュレーション条件

シミュレーション 3 では図 5 のタイルワールド環境を用いて、システム全体としての行動評価の最適化を目指した場合の、Masbiole と従来の MAS の性能比較を行う。GNP、パラメータおよび進化に関する設定はシミュレーション 1 とすべて同じとし、表 9 に示す 3 通りの共生関係の組合せのシミュレーションを行った。

表 9 の Case 1 は各エージェントが共利をとる共

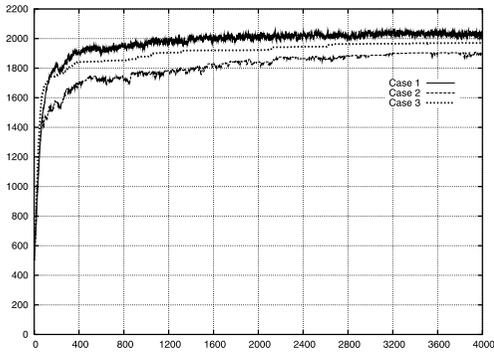


図 11 シミュレーション 3 の各行動評価曲線の比較

Fig. 11 Comparison of evaluation curves in simulation 3.

利型 Masbiole であり, *Case 2* は各エージェントが自改善をとる従来の MAS である. *Case 3* の Whole Improvement とは, エージェントが進化を行うときに「自分と相手の行動評価の和」の高い順にランクを計算し, 選択/淘汰を行う手法である.

3.4.2 結果と考察

図 11, 図 12 に表 9 の各 *Case* のシミュレーション結果を示す. 図 11 はシステム全体の行動評価 ($E_1 + E_2$) (縦軸) の世代 (横軸) ごとの推移を, 図 12 は各 *Case* のエージェント 1 が進化によって獲得した行動評価点を行動評価平面 (E_1, E_2) 上にそれぞれプロットしたものを示している. また結果はすべて, 乱数系列 10 通りに対する実行結果の平均である.

図 11 の結果の比較より, 各エージェントが共利をとる *Case 1* はシステム全体の行動評価が最も良くなっている. これは図 12 を見ると分かるように, *Case 2* や *Case 3* では行動評価点に対して 1 目的のランクの計算を行うため, 狭い範囲の解の探索しか行えず進化の過程で行動評価点が局所解に収束してしまうからである. 一方, *Case 1* では自分と相手の 2 目的のランクの計算によって幅広い解の探索を行うため, 局所解 (ナッシュ均衡解) に陥ることなく適切な共生パレート解が求まったものと考えられる.

またシステム全体で最適な行動評価値について考えると, 図 5 では各エージェントに 5 個ずつ穴が用意されているのでそれらすべてに Tile *S* を落とすことが最適な処理であり, 式 (4) から第 3, 4 項も考慮して 2100 程度が行動評価の最適値である. 図 11 では *Case 1* が集団の平均値として 2000 を超える行動評価を得ており, エージェント 1, 2 とも集団の大部分の個体が最適な処理を行うように進化できているといえる. それに対し *Case 2, 3* では行動評価が 2000 を下回っており, エージェントは最適な処理を獲得できていない.

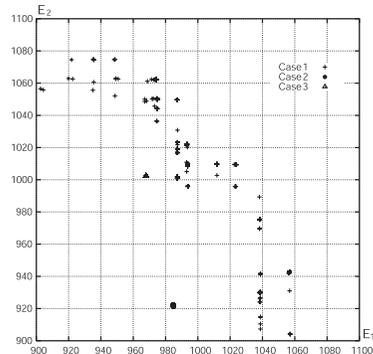


図 12 エージェント 1 の各行動評価点の比較

Fig. 12 Comparison of evaluation points of agent 1 in each case.

さらに詳しい性能比較を行うため, 1 エピソードのステップ数および各エージェントに属するユニット数を, $\{30, 50, 70\}$, $\{1, 3, 5\}$ と変更して得られたエージェント 1 と 2 行動評価の平均の和の最良値を比較したものを表 10 に示す. 表 10 中の太字は 3 つの *Case* の中で最も良かったものを表す. 表 10 の結果より, *Case 1* はステップ数 50, 70 およびユニット数 3 の条件のときに最も良い結果をあげ, その他の条件においても大体良い結果を示している. また *Case 1* ではステップ数が 50 およびユニット数が 3 の条件において行動評価の平均値は 2000 を超え, エージェントの大部分の個体が最適解を獲得しているが, *Case 2, 3* ではステップ数およびユニット数がより大きい条件において *Case 1* と同様の最適解を得ている. したがって, 共利型 Masbiole は通常の MAS よりコストを抑えてより最適な処理を実現することができる.

以上の結果から, Masbiole の共利に基づいて各エージェントを進化させることによって, 従来の MAS より良いシステムを構築できることが期待される. 具体的には, サッカーロボットをはじめとするロボット間の連携による作業システムに対し Masbiole を導入し, より効率的な作業分担の実現を図ることが可能になる.

4. 結 論

本論文では生態系に見られる共生現象にヒントを得た, 共生物学進化型マルチエージェントシステム (Masbiole) に対して, タイルワールドモデルの 3 種類のシミュレーションを行い, 共生的挙動の検討, 複雑な創発現象の考察および Masbiole と従来の MAS の性能比較を行った. その結果, Masbiole では MAS に比べ多様な共生的挙動が見られ, 複雑な創発現象や企業競争モデル等の構築に応用が可能であることを示した. またエージェントが共利関係を採用する Masbiole

表 10 ステップ数とユニット数の各設定に対する行動評価の最良値の比較

Table 10 Comparison of best evaluation of each case for the step and unit conditions.

Case	Number of step			Number of unit		
	30	50	70	1	3	5
1	1939.87	2065.33	2086.57	1609.76	2065.33	2096.39
2	1840.84	1906.96	1937.27	1468.79	1906.96	2109.96
3	1950.60	1970.10	2006.60	1670.70	1970.10	2071.30

が、従来の MAS や単一評価システムよりも効率的な問題の解決を実現できることを明らかにした。

今後は迷路問題や追跡問題、人工蟻社会モデル等の各種マルチエージェントシステムや多目的最適化問題に Masbiole を導入し、さらに詳しい検討を行っていく予定である。またそのための拡張として、現段階では設計者によって与えられているエージェントの共生関係をエージェント自身が適応的に変更していくメカニズムを導入する予定である。

参考文献

- 1) 沼岡千里, 大沢英一, 長尾 確: マルチエージェントシステム, 共立出版 (1998).
- 2) 生天目章: マルチエージェントと複雑系, 森北出版 (1998).
- 3) Wooldridge, M.: *An Introduction to Multi-Agent Systems*, Wiley (2002).
- 4) 松田裕之: 「共生」とは何か, 現代書館 (1995).
- 5) 平澤, 吉田, 中西, 胡: マルチエージェントシステムの共生学習進化 (Masbiole) の基礎検討, 電学論 C, Vol.122-C, No.3, pp.346-354 (2002).
- 6) 平澤, 中西, 江口, 胡: 共生学習進化型マルチエージェントシステムとその応用, 電学論 C, Vol.123-C, No.1, pp.67-74 (2003).
- 7) 江口, 平澤, 胡, 村田: Genetic Network Programming を用いた共生学習進化型マルチエージェントシステム, 電学論 C, Vol.123-C, No.3, pp.517-526 (2003).
- 8) Pollack, M.E. and Ringuette, M.: Introducing the tile-world, Experimentally evaluating agent architectures, *Proc. Conference of the American Association for Artificial Intelligence*, pp.183-189 (1990).
- 9) 平澤, 大久保, 片桐, 胡, 村田: 蟻の行動進化における Genetic Network Programming と Genetic Programming の性能比較, 電学論 C, Vol.121-C, No.6, pp.1001-1009 (2001).
- 10) 片桐, 平澤, 胡, 村田: Genetic Network Programming とそのマルチエージェントシステムへの応用, 電学論 C, Vol.122-C, No.12, pp.2149-2156 (2002).
- 11) Bagchi, T.P.: *Multiobjective Scheduling By*

Genetic Algorithms, Kluwer Academic Publishers (1999).

- 12) 伊庭斉志: 進化論的計算の方法, 東京大学出版会 (1999).

(平成 15 年 4 月 11 日受付)

(平成 15 年 6 月 25 日再受付)

(平成 15 年 7 月 11 日採録)



江口 徹

1978 年生。2003 年九州大学大学院システム情報科学府修士課程修了。同年早稲田大学大学院情報生産システム研究科博士後期課程入学。現在、情報アーキテクチャ分野に在学中。



平澤宏太郎 (正会員)

1942 年生。1964 年九州大学工学部電気工学科卒業。1966 年同大学大学院工学研究科修士課程電気工学専攻修了。同年 (株) 日立製作所入社, 日立研究所勤務, 1989 年同研究所副所長。1991 年同大みか工場主管技師長。1992 年九州大学工学部教授, 1996 年同大学大学院システム情報科学研究科教授。2002 年早稲田大学大学院情報生産システム研究科教授, 現在に至る。計測自動制御学会, 電気学会, IEEE の各会員。工学博士。



古月 敬之

1962 年生。1985 年中国中山大学大学院修士課程修了。同年同大学電子工学科助手, 1988 年同講師。1993 年来日。1997 年九州工業大学情報工学研究科博士後期課程修了。同年九州大学ベンチャービジネスラボラトリ非常勤研究員を経て, 同年同大学システム情報科学研究科助手。2003 年早稲田大学大学院情報生産システム研究科助教授, 現在に至る。計測自動制御学会, 電気学会の各会員。情報工学博士。