

動的環境における マルチエージェントプランニングの提案

吉田 直人^{1,a)} 高屋 英知² 加藤 慶彦² 栗原 聡¹

概要：動的で複雑な環境において、エージェントが自律的な行動を取るためには、限られた時間内で効率的かつ柔軟性のあるプランニングを行うことが必須であり、本研究ではその手法の一つとして Agent Network Architecture を導入した。実験は Real-Time TileWorld という環境上で、ルールベースで行動するエージェントと ANA を導入したエージェントの振る舞いを比較することによって行った。ANA を導入したエージェントにはメタ的なプランニングの発現といった特徴が見られたが、ルールベースと同程度の効率的な振る舞いをするには、モジュール設計やパラメータをより複雑にする必要があり、ANA のパラメータ群に対してどのように強化学習などの枠組みを適用していくかが今後の課題である。

Proposal of Multi Agent Planning in Dynamic Environment

1. はじめに

動的で複雑な環境において、エージェントが自律的な行動を取るためには、限られた情報を用いて限られた時間内で効率的かつ柔軟性のあるプランニングを行うことを要求され、2 節で述べるように今まで多くの手法が提案されてきた。本研究ではそのような手法の一つとして Agent Network Architecture (以下 ANA) を導入した。ANA は 3 節で述べるようにエージェントによる分散型ネットワークで構成され、モジュール同士の活性伝播により行動の順番を決定する。そして、動的環境において簡単な目的を達成することはできるが、モジュールを増やすと様々な問題が発生する。本研究では ANA を用いたプランニングの特性について検証するため、動的で予測不能なシミュレーション環境下で、ルールベースで行動を選択するエージェントと ANA を導入したエージェントを比較する実験を行った。

2. 関連研究

2.1 最小コスト経路を求める手法

まず、RTT 上で特定のセル間の距離を算出するために用

いた A* アルゴリズムについて述べる。Hart ら [1] はグラフ理論とヒューリスティック情報を組み合わせてグラフ内の最小コスト経路を求める手法として、A* アルゴリズムを提案した。アルゴリズムは以下の順序で進行する。

A* アルゴリズム

- (1) スタートのセルを基準のセルとする。
- (2) 基準のセルの状態を open にする。
- (3) スタートのセルから基準のセルまでの実距離を求める。
- (4) 基準のセルから移動可能なセルの状態を open にし、基準のセルの状態を close にする。
- (5) 状態が open なセルの実距離と推定距離を求める。
- (6) 状態が open なセルのうち、実距離と推定距離の和が最小のセルを基準のセルとする。
- (7) 基準のセルには一つ前の基準のセルを親として記録する。
- (8) 基準のセルがゴールのセルになるまで 4~7 を繰り返す。
- (9) ゴールのセルからスタートのセルまで親のセルを辿っていき、最適経路とする。

2.2 動的で予測不能なシミュレーション環境

次に、本研究で用いた動的で予測不能なシミュレーション

¹ 慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

² 慶應義塾大学大学院理工学研究科

Graduate School of Science and Technology, Keio University

a) naoto_yoshida@keio.jp

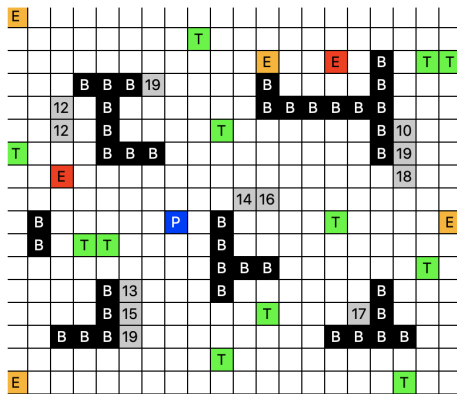


図 1 RTT

表 1 オブジェクトの一覧

オブジェクト	背景色	テキスト
障壁	黒色	B
ホール	灰色	数字 (得られるポイント)
タイル	緑色	T
エネルギー供給ベース	橙色	E
プレイヤーエージェント	青色	P
敵エージェント	赤色	E

環境について述べる。Pollack[2] は動的で予測不能なシミュレーション環境として TileWorld を提唱した。TileWorld は環境とエージェントが高度にパラメータ化されているため、パラメータの調整によって環境の変化とエージェントの特性を制御できるという点で優れている。また、その点を利用してメタレベルな推論機能・戦略を持つエージェントを同じ環境下で比較したり、環境を変化させてエージェントの性能を評価したりできる。

また、Bratman ら [3] は TileWorld 環境下において、有効なアーキテクチャとして Intelligent Resource-bounded Machine Architecture (以下 IRMA) を提案した。IRMA は、限られた時間内にプランニングを終えるために、目的の推定価値を数値化してフィルタリングすることで計算対象を絞り込んでいる。計算コストの低いプランニングを全ての目的に対して行った場合と、IRMA を用いて計算コストの高いプランニングを一部の目的に対して行った場合を比較すると、後者の方が効率よく目的を達成できることが報告されている。

Kurihara ら [4] は TileWorld を拡張し、より動的性と複雑性を持たせた Real-Time TileWorld (以下 RTT) というシミュレーション環境を提案した。RTT を図 1 に示す。また、RTT 環境下に配置されるオブジェクトは表 1 の通りである。プレイヤーエージェントはエネルギーを消費しながら移動し、タイルのセルに移動してタイルを持ち、ホールのセルに移動してタイルをホールに埋めることでポイントを獲得することができる。プレイヤーエージェントはエネルギー供給ベースでエネルギーを補給でき、敵エージェントはプレイヤーエージェントを追いかけ、エネルギーを奪う。

Kurihara らはさらに、RTT 上でより効率的にポイントを得るための即応性と熟考性を兼ね備えたプランニング手法として、マルチエージェント実時間リアクティブプランニング (MRR-Planning: Multi-agent real-time reactive planning) を提案した。MRR-Planning はプランニングエージェント、行動エージェント、行動選択エージェントの 3 つのエージェント群で構成される。

プランニングエージェントは 4 つのエージェントから構成される。各プランニングエージェントはそれぞれ以下のような目的を持ち、それぞれの目的だけを達成するためにプランニングを行う。

- P-agentA: 敵エージェントから逃げる
- P-agentB: 残存エネルギーがゼロにならない
- P-agentC: 移動経路上に出現したホールやタイルに即応的に反応すること
- P-agentD: あらゆる場合を考慮して最適なホールとタイルの巡り方を決定すること

行動エージェントは 7 つのエージェントから構成される。各行動エージェントはそれぞれ以下の単位行動を受け持つ。

- B-agent1: 1 セル上に移動する
- B-agent2: 1 セル下に移動する
- B-agent3: 1 セル左に移動する
- B-agent4: 1 セル右に移動する
- B-agent5: タイルを持つ
- B-agent6: タイルをホールに埋める
- B-agent7: エネルギーを補充する

まず、センサからの情報がプランニングエージェントに刺激を与える。閾値以上の刺激を受けたプランニングエージェントはプランニングを行い、プランの完了に必要な行動エージェントを刺激する。行動選択エージェントは行動エージェントの刺激の量を監視し、閾値以上の刺激を受けた行動エージェントが存在する場合、背反な行動が同時に選択されることのないように行動エージェントを活性化させる。従来のエージェントは環境の変化が激しくなると、急激にパフォーマンスを落とすが、MRR-Planning の場合はパフォーマンスの減少が小さいことが報告されている。

3. The Agent Network Architecture

2 節で述べた多くの手法は、センサからの情報が確実に得られることやプランした行動を行えることが前提となっているが、実環境では不確実な情報をもとにプランニングを行い、行動の結果をセンサからのフィードバックによって取得する必要がある。さらに、環境が複雑になると、目的を小目的に分解することも困難になる。そこで、本研究ではそれらの問題を克服するための手法として、Agent Network Architecture (以下 ANA) を導入した。本節では ANA について詳しく述べる。

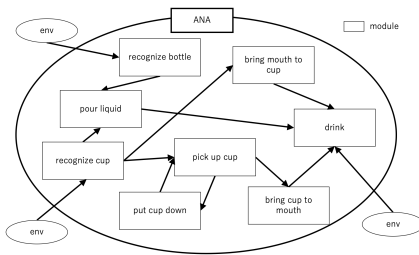


図 2 ANA

Maes[5] は自律エージェントのためのアーキテクチャを設計することを目的とし、エージェントが目的を達成するための行動の手順を提示する ANA というアーキテクチャを提案した。ANA の構成を図 2 に示す。ANA は簡単な機能を持つ多数のモジュールの集合であり、機能同士の前後関係がネットワークを構成している。各モジュールは一定の閾値以上の刺激を受け取ると活性化し、前工程のモジュールが全て活性化している場合はモジュールの機能を実行した後、後工程のモジュールに刺激を与える。また、前工程のモジュールが一つでも活性化していない場合は前工程のモジュールに刺激を与える。ある時刻における外部環境から ANA への刺激が各モジュールを刺激し、モジュール同士が相互に活性化し、次の時刻における ANA の内部状態を決定する。ANA 内部のモジュール間相互作用により、エージェントにテクニックやアルゴリズムを直接組み込むことなく、エージェントがより複雑な目的を持つことができ、その目的を達成するための機能も得ることができる。

ANA を組み込んだエージェントを用いて、生物の行動を再現しようとする試みや人間との対話で癒しや親しみを与えようとする試みが今まで数多く行われてきた。Tyrrell[6] はカモメを再現したエージェント、Tehseen ら [7] は特定の建物内の人間を再現したエージェント、Hayashi ら [8] はカメラとアームを持ったロボットのエージェント、Lim[9] は実世界の環境におけるサービスロボットのエージェント、Luis ら [10] は複数ロボットが協調するためのエージェント、Rohlfshagen ら [11] は TileWorld に似たシミュレーション環境でリソースを集めるエージェントに ANA を組み込み、エージェントの振る舞いに対する評価を行った。いずれの研究でも、エージェントが環境の中を移動するだけ、ものを取得するだけというような、ある程度単純な目的や行動に対しては ANA が適当な振る舞いを提案することを確認した。

一方、Pfeifer ら [12] は現実世界のような複雑な環境下で ANA を組み込んだエージェントを用いてプランニングを行う際の問題点として、以下の点を指摘した。

- 矛盾する目的が複数存在するときに刺激の伝播でデッドロックが起こる
- 行動が完了したか実行中かを判別できない

- 時間がかかる行動を待ってから知覚するので、反応が遅れることがある

そして、この問題点の解決策として、モジュールを階層化することと、センサからの値に変化があり、実行中の行動が不適切と判断したときにいつでも行動を中断できるようにすることを提案した。さらに Pfeifer らは行動選択のための情報や行動による結果自体をモジュールとし、知覚と行動が互いに依存しない形で保持することによって、Maes の ANA の問題点を解決した。

4. 実験環境

本研究の実験ではルールベースでプラン選択を行うプレイヤーエージェントと ANA を導入したプレイヤーエージェントを RTT 環境下で動かし、獲得したポイント数などを比較した。

4.1 ルールベースでプラン選択を行うプレイヤーエージェント

ルールベースでプラン選択を行うプレイヤーエージェントの構成を図 3 に示す。プレイヤーエージェントはプラン選択エージェント、プランニングエージェント、行動エージェントの 3 つのエージェント群で構成される。

プラン選択エージェントはプランニングエージェントの中から優先度の高いものを選択するエージェントである。

プランニングエージェントは 4 つのエージェントから構成される。各プランニングエージェントはそれぞれ以下のような目的を持ち、それぞれの目的だけを達成するために A* アルゴリズムを用いたプランニングを行う。

- P-agentA: エネルギー供給ベースに近づくこと
- P-agentB: 敵エージェントから逃げる
- P-agentC: ホールに近づくこと
- P-agentD: タイルに近づくこと

行動エージェントは 7 つのエージェントから構成される。各行動エージェントはそれぞれ以下の単位行動を受け持つ。

- B-agent1: 1 セル上に移動する
- B-agent2: 1 セル下に移動する
- B-agent3: 1 セル左に移動する
- B-agent4: 1 セル右に移動する
- B-agent5: タイルを持つ
- B-agent6: タイルをホールに埋める
- B-agent7: エネルギーを補充する

4.2 ANA を導入したプレイヤーエージェント

ANA を導入したプレイヤーエージェントの構成を図 4 に示す。ANA は一部のモジュールと繋がりのみ記している。ANA は主に以下の機能を持つモジュール群で構成した。

- Planning モジュール: センサの情報からそれぞれのプ

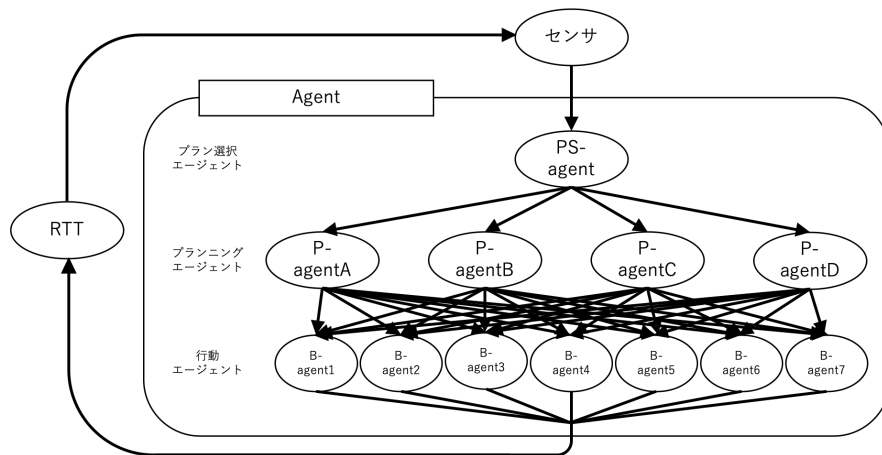


図 3 ルールベースでプラン選択を行うプレイヤーエージェントの構成

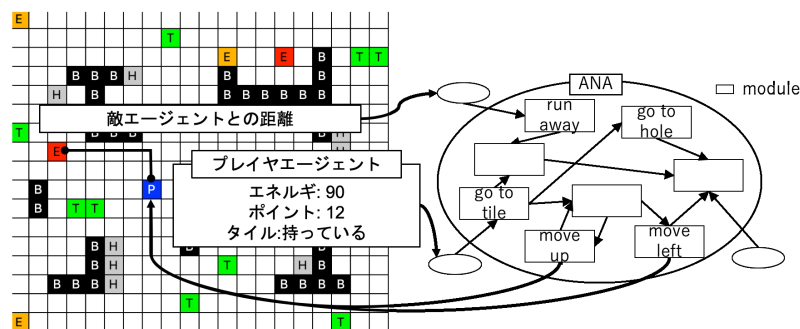


図 4 ANA を導入したプレイヤーエージェントの構成

ランの重要度を計算する機能を持つ

- Recognize モジュール: それぞれのセルにオブジェクトがあるという情報を持つ
- Action モジュール: プレイヤーエージェントを操作する機能を持つ

まず、環境やプレイヤーエージェントから Planning モジュールや Recognize モジュールに情報が与えられる。次に、モジュール同士で刺激の受け渡しを行う。Action モジュールが活性化すると、その行動が実行可能な場合は実行する。

5. 実験結果

ルールベースでプラン選択を行うプレイヤーエージェントと ANA を導入したプレイヤーエージェントを用いて 10 回ずつ実験を行い、ステップ数、ポイント数、往復回数、停止回数の観点で両者を比較した。

5.1 ステップ数

プレイヤーエージェントがエネルギーを切らすことなく行動し続けたステップ数を図 5 に示す。プレイヤーエージェントがエネルギー供給ベース上に留まることを選択し続けた場合、ゲームが終わらなくなってしまうので、200 ステップで強制的にゲーム終了としている。ルールベースでプラン選択を行うプレイヤーエージェントはほとんどの実験で 200 ステップ時にも十分な量のエネルギーを残していた。一方で、ANA を組み込んだエージェントはほとんどの実験でエネルギーを切らしてしまった。

この違いはエネルギー補給と敵エージェントから逃げるという 2 つの目的の優先順位の違いによるものと考えられる。ルールベースでプラン選択を行うプレイヤーエージェントはエネルギーが少なくなると、プラン選択の際に必ず敵エージェントからの逃亡よりもエネルギー補給を優先するので、エネルギーを切らすことがほとんどなかった。一方で、ANA を組み込んだエージェントはエネルギーが少ないときに敵エージェントからの逃亡やポイントの獲得を選択する

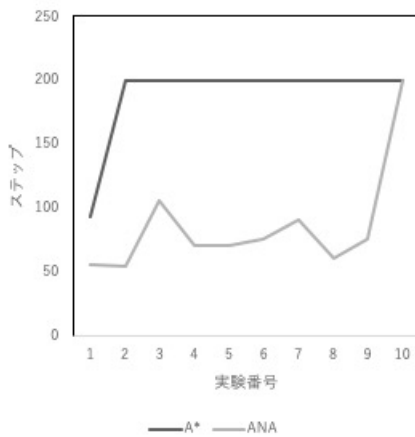


図 5 実験結果 (ステップ数)

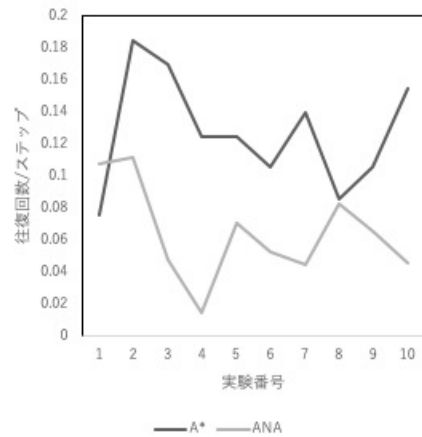


図 7 実験結果 (往復回数)

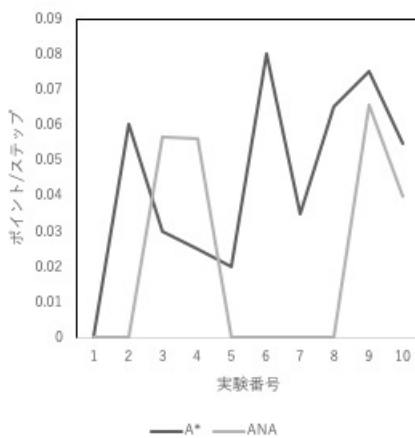


図 6 実験結果 (ポイント数)

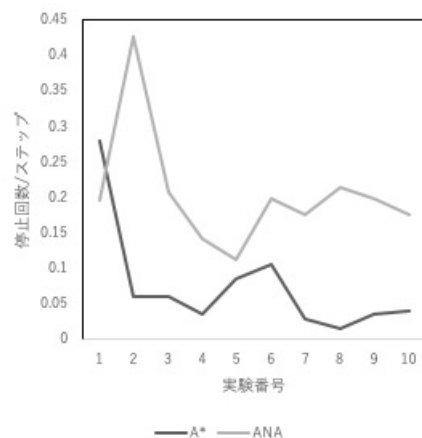


図 8 実験結果 (停止回数)

こともあり、結果としてエネルギーを切らしてしまった。この点はエネルギー供給に関わるモジュールのパラメータを調整することである程度解決できると考えられる。

5.2 ポイント数

プレイヤーエージェントが1ステップあたりに獲得したポイント数を図6に示す。ポイント数はルールベースの方が多くなった。

この違いはゲーム開始後の数ステップでのプラン選択の違いによるものと考えられる。ルールベースでプラン選択を行うプレイヤーエージェントはゲーム開始時には敵エージェントとの距離も残存エネルギー量も閾値以下なので、タイルを取得するためのプランニングを行う。結果、ほとんどのゲームで少なくとも1回はタイルを取得し、ポイントを獲得するという結果につながったと考えられる。一方で、ANAの場合、ゲーム開始時のタイルまでの距離は4、ホールまでの距離も4で、最初にどのプラン選択モジュールも活性化せず、その後のランダムな環境変化に応じてプラン選択モジュールが選択される。結果、ゲーム開始直後にタイルを取りにいかない場合があり、ルールベースよりも点数が低いという結果になったと考えられる。

5.3 往復回数

プレイヤーエージェントが1ステップあたりにどれだけ往復を行ったかを図7に示す。ここで、往復とは他のオブジェクトがないセルに移動し、次のステップで元のセルに戻ってくる行動のことを指す。往復回数はANAの方が少なくなった。これは、ANAがモジュールの活性値として、前のステップの環境を保持しているため、プランニングに連続性が生まれたためと考えられる。逆に、ルールベースでプラン選択を行う場合、そのステップの環境だけを用いてプラン選択を行うので、プランの連続性が失われ、無駄に往復するような行動につながったと考えられる。

5.4 停止回数

プレイヤーエージェントが1ステップあたりにどれだけ停止を行ったかを図8に示す。ここで、停止とは他のオブジェクトがないセル上で上下左右のいずれにも移動しないことを指し、選択した行動が実行不可能なとき(RTTの範囲外に移動しようとしている時、障壁のあるセルに移動しようとしている時など)や、1ステップの時間内にプランニングが終了せず、行動が決定されなかった時に発生する。

停止回数はルールベースでプラン選択を行うプレイヤーエージェントの方が少なかった。これは、ルールベースでプラン選択を行う場合は非常に短時間で計算が終了するプラン選択と A*アルゴリズムによるプランニングで構成されているので、本研究の実験環境では 1 ステップ内で計算が完了したのに対し、ANA は環境からの活性値が少ないと活性化するモジュール数が増えるので、計算に時間がかかり、行動選択まで至らないという場合があると考えられる。

6. 考察

ANA を導入したプレイヤーエージェントは往復回数の点でルールベースでプラン選択を行うプレイヤーエージェントよりも優れていたが、他の観点ではルールベースのプラン選択を上回ることができなかった。

まず、その原因と解決策について考察する。ルールベースでプラン選択を行うプレイヤーエージェントは RTT 上の全てのオブジェクトの座標を把握し、その情報に基づいてプラン選択、プランニングを行うのに対し、ANA はプレイヤーエージェントから距離 5 以内のオブジェクトの座標しか把握できないので、プランニング精度が低くなってしまっている。本実験では、実装の都合上距離 5 のセルを探索するモジュールまでしか作成しなかったが、距離 6 以上のモジュールを増やすことによってある程度精度を上げることができると考えられる。また、本実験では 1 セルに対して 1 モジュールを対応させたが、近くのセルの情報と遠くのセルの情報は重要度が異なるので、遠くの隣接するセルには共通のモジュールを割り当てるなどしてモジュール数を削減することで、同じ計算時間で同じモジュール数でもより遠くのセルの情報まで取得できるようになると考えられる。

次に、目的同士の関連について考察する。ルールベースでプラン選択を行うプレイヤーエージェントの場合、エネルギーが少ない時は敵エージェントやタイル、ホールを全て無視してエネルギーを補給するためのプランニングを行うが、RTT では敵エージェントに追いつかれるとエネルギーを奪われるので、本来はエネルギーの補給と敵エージェントからの逃亡は互いに複雑に影響を及ぼす関係でなければならない。ルールベースでプラン選択を行うプレイヤーエージェントで目的同士の相互作用を実現しようとする、プラン選択の際のルールを複雑にすることで実現できるが、その分計算コストが増してしまう。一方、ANA の場合、それぞれを目的とするモジュール同士を接続するだけで目的同士の相互作用を実現することができる。

最後に、ステップ数やポイント数と直接の関連はないが、ANA を導入したプレイヤーエージェントがエネルギー供給ベースがあるセルにいる時、近くにタイルやホールが出現するまでエネルギー供給ベース上に留まりエネルギーを補給し続け、タイルやホールが出現するとタイルを取得しホー

ルに埋めるという動作を繰り返し、敵エージェントを完全に無視した振る舞いをする様子を観察することができた。本実験の設定ではこの振る舞いを繰り返すことでエネルギーを枯渇させることなくポイントを増やすことができる。しかし、この振る舞いはエージェント設計時に想定していたものではないので、ルールベースでプラン選択を行う場合は決して発生することのない事象であり、ANA だから発生した事象ということができる。このようなメタ的なプランニングの発現は ANA の長所と言える。

7. まとめ

ルールベースでプラン選択を行うプレイヤーエージェントと ANA を導入したプレイヤーエージェントを RTT 上でプランニングさせたときにどのように振る舞うのかを比較することで、ANA を導入したプレイヤーエージェントの特徴や課題、解決策を考察することができた。ANA は往復回数の点でルールベースよりも優れていたが、他の観点ではルールベースを上回ることができなかった。そして、ゲーム開始直後と敵エージェント接近時のプラン選択の違いがステップ数やポイント数の違いとして表れることがわかった。

また、ANA を組み込んだエージェントを用いてより効率的にポイントを獲得するには強化学習を用いたモジュール設計、パラメータ調整が必要である。そして、人間に想像することができないメソッドやテクニックを生み出すためにも強化学習を用いたパラメータ設定は必要不可欠である。したがって、ANA のパラメータ群に対してどのように強化学習の枠組みを適用していくかが今後の課題である。

参考文献

- [1] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100–107, 1968.
- [2] M. E. Pollack. Introducing the tileworld: Experimentally evaluating agent architectures. *AAAI-90*, pp. 183–189, 1990.
- [3] Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, Vol. 4, No. 3, pp. 349–355, 1988.
- [4] Satoshi Kurihara, Shigemi Aoyagi, Rikio Onai, and Toshiharu Sugawara. Adaptive selection of reactive/deliberate planning for a dynamic environment. *Robotics and Autonomous Systems*, Vol. 24, No. 3, pp. 183–195, 1998.
- [5] P. Maes. The agent network architecture(ana). *SIGART Bulletin*, Vol. 2, No. 4, pp. 115–120, 1991.
- [6] Toby Tyrrell. An evaluation of maes' s bottom-up mechanism for behavior selection. *Adaptive Behavior*, Vol. 2, No. 4, pp. 307–348, 1994.
- [7] Tehseen Zia, Roland Lang, Harold Boley, Dietmar Bruckner, and Gerhard Zucker. An autonomous adaptive multiagent model for building automation. *IFAC Pro-*

- ceedings Volumes, Vol. 42, pp. 250–254, 2009.
- [8] Eiji Hayashi and Takahiro Yamasaki. Design of robotic arm’s action to imitate the mechanism of an animal’s consciousness. *Artificial Life and Robotics*, Vol. 15, pp. 565–570, 2010.
 - [9] Gi Hyun Lim. Shared representations of actions for alternative suggestion with incomplete information. *Robotics and Autonomous Systems*, Vol. 116, pp. 38–50, 2019.
 - [10] Luis Manso, Pablo Bustos, Juan Bandera, Adrián Romero-Garcés, Luis Calderita, Rebeca Marfil, Antonio Bandera. Deep representations for collaborative robotics. pp. 179–193, 12 2016.
 - [11] Philipp Rohlfshagen and Joanna Bryson. Flexible latching: A biologically- inspired mechanism for improving the management of homeostatic goals. *Cognitive Computation*, Vol. 2, pp. 230–241, 2010.
 - [12] R. Pfeifer, B. Blumberg, J. Meyer, and S. W. Wilson. An extension of Maes’ Action Selection Mechanism for Animats, pp. 153–158. 1998.