

追跡ゲームを用いたマルチエージェント協調行動の 獲得手法の検討

09E27 早崎孝明

1. はじめに 本研究の目的は、追跡ゲームを題材としてこれらのエージェントの振る舞いについて調べ、エージェント間通信を利用した不均一で階層的なエージェントが最も協調的な捕獲能力に優れていることを示すことである。追跡ゲームとは、無限に広がる2次元格子上に2種類のエージェントが存在し、1種類のエージェントがもう一方のエージェントを捕獲することを目的としたものである。

2. 追跡ゲームの主な内容 追跡ゲームを使用するにあたって主な取り決めは、無限に広がる2次元格子(フィールド)上に、ハンター(H)と呼ばれるエージェントと、獲物(P)と呼ばれるエージェントがいる。エージェントたちは、2次元格子を縦横方向に、一度に1セルずつ移動することができる。もちろんしばらく動かないでいることも可能である。各エージェントの視界は限られており、フィールド全体を見ることはできない。各ハンターエージェント同士の通信は可能であるがハンター同士がお互いの相対座標を知るためには、自分の視界に相手のハンターを捕らえている必要がある。ハンターエージェントを2体、獲物エージェント1体の配置は、無限に広がる2次元格子にランダムに初期配置される。獲物エージェントを捕獲するための条件は、左右か上下の方向からはさむことである。

3. プログラムの主な流れ プログラムの流れとしては、まず search prey 関数で獲物エージェントを発見し、follow prey 関数で獲物エージェントを追いかけて、sub routine 関数で獲物エージェントを捕獲する。search prey 関数とは、2次元格子にランダムに初期配置された状態から、獲物エージェントを発見する(視界に捕らえる)ための関数である。次に follow prey 関数とは、発見した獲物エージェントを視界から逃がさないようにする関数である。最後に sub routine 関数とは、視界に捕らえた獲物エージェントを2体のハンターエージェントで左右か上下の方向からはさむ関数である。

4. 実験方法 実験方法としては、まず均一なエージェントと不均一なエージェントを3.で述べたプログラムの流れで作成する。均一なエージェントとは、同じ状況下で2つのハンターエージェントを入れ替えても、前そこにいたハンターエージェントとまったく同じ動きをするような、プログラミングをされたエージェントのことである。不均一なエージェン

トとは、エージェント内に階層構造があり他のエージェントに命令をできる状態を指す。この2種類のエージェントを使って、どちらがより協調的な捕獲能力に優れているかを検討する。

5. 結果および考察 不均一なエージェントの結果と均一なエージェントの結果を図1に示す。まず獲物エージェントの発見率では、エージェントが不均一の場合、視界に広さとフィールドの広さの差があまりに大きい場合を除いては、ほぼ100%の確率で発見できた。それに比べてエージェントが均一な場合は、視界の広さとフィールドの広さの差が少ないときでも発見率が低いことが分かった。次に捕獲率についても、エージェントが不均一の場合は同様のことが言える。エージェントが均一な場合は、発見率のときよりもさらに低い結果となった。これらの結果を比べて言えることは、エージェントが均一な場合それぞれのエージェントが単独で探索するため発見が悪く、発見した後も捕獲するための行動を1つに絞ることができなかつたために、取り逃がすことが多くあった結果だと考えられる。逆にエージェントが不均一の場合は、探索する場合にも命令によって協力することができるため発見効率もよくなり、命令により捕獲するための行動を1つに絞ることができるため、取り逃がすことなく捕獲できる結果だと考えられる。

6. まとめ 結論としては、均一なエージェントだとエージェント同士が協調した動きができず、その結果無駄な動きが多く発見率、捕獲率ともに低くなった。その比べ不均一なエージェントでは、命令によってエージェント同士が協調した動きができ、それによって無駄な動きがなくなりさらに捕獲条件をクリアでき、発見率、捕獲率ともに向上する結果となった。

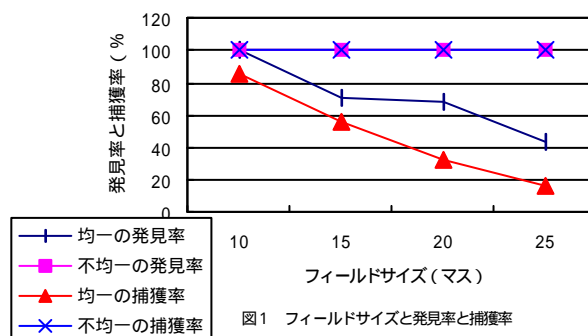


図1 フィールドサイズと発見率と捕獲率