

自動グループ構成手法 ADG によるマルチエージェントの行動制御

原 章[†] 長 尾 智 晴[†]

複雑な協調動作を行うエージェント集団を遺伝的プログラミング (GP) を用いて生成する場合, Homogeneous 集団よりも Heterogeneous 集団の方が性能が良くなると考えられる. しかし Heterogeneous 集団においてもエージェントの数が大規模になるにつれて, 探索領域の増大により探索効率が悪化してしまう. これに対しては, 問題ごとの必要な役割分担数などに応じてエージェントをいくつかのグループに分け, グループ内のエージェントを同じプログラムにより動作させるといった解決策が考えられるが, 最適なグループ数などをあらかじめ与えることは困難である. そこで本研究では GP による進化の過程で自動的に, 問題解決に必要なグループ数, 各グループへのエージェントの配分, 各グループの行動規則の最適化を同時に行うシステム ADG を提案する. 本手法を荷物搬送問題や拡張したタイルワールドへ適用した結果, 最適なグループ構造を獲得できることを確認した.

Action Control of Multiple Agents Using ADG; Automatically Defined Groups

AKIRA HARA[†] and TOMOHARU NAGAO[†]

In generating a multi-agent team which solves a problem cooperatively by means of Genetic Programming (GP), it seems that a heterogeneous team performs better than a homogeneous team. In a heterogeneous team, however, as the number of agents increases, the size of search space becomes vast and the efficiency of search declines. One of the solutions of this problem is to divide a team into the proper number of groups, and to provide the same program for the all agents belonging to the same group. However it is difficult to know the adequate team structure beforehand. In order to solve these problems, we propose a method called *ADG; Automatically Defined Groups*. In this paper, we applied this method to a simple load transportation problem and a modified Tile World problem, and confirmed that the optimal team structure was acquired by the method in each problem.

1. はじめに

近年, 多数のエージェントによる問題解決が注目を集めている. これはある種の問題を複数のエージェントの協調により効率的に解決することができるためである. 進化的計算法の分野でも, 遺伝的プログラミング (GP; Genetic Programming) を用いてエージェントの行動制御プログラムを自動生成する試みがさかに行われている.

GP は, 遺伝的アルゴリズム (GA; Genetic Algorithm)^{1), 2)}を, 木構造のような構造的な表現を扱うことができるように拡張した進化的計算法の一手法である³⁾. GP では木構造を表す個体に対して, GA と同様に交叉や突然変異を行い世代交代を繰り返すことで, より問題に適した木構造を持つ個体を獲得する. GP

によりマルチエージェントの行動規則を獲得する手法には大きく分けて次の 2 通りの方法が考えられる.

手法 1 各エージェントを GP の一個体とし, 各々のエージェントの行動を別々に評価し共進化などにより集団全体のパフォーマンスを向上させる手法

手法 2 エージェント集団を 1 つの個体として考え, 集団全体のパフォーマンスをもとに, 進化により効率的なエージェント集団を獲得する手法

エージェント集団が役割分担をしながら協調して問題解決を行う場合, 手法 1 では他のエージェントを助ける行動をとるような, 集団のパフォーマンスに直接的に反映されないエージェントの評価が難しい. たとえば, 複数のエージェントによる荷物搬送問題において, あるエージェントが他のエージェントの進行方向にある障害物を排除するような動作を行った場合でも, その動作が効果的であったかどうかを直接評価することは困難である. しかし, 手法 2 では集団のパフォーマンスにより集団に属するエージェントを等しく評価

[†] 東京工業大学工学部附属画像情報工学研究施設
Imaging Science and Engineering Laboratory, Faculty
of Engineering, Tokyo Institute of Technology

するため評価が容易である。先の荷物搬送問題の例では、一定時間内に運んだ荷物の数などにより集団内のすべてのエージェントを均等に評価すればよい。これにより集団のパフォーマンスに直接貢献していないエージェントの評価も可能となるため、役割分担などの複雑な協調動作の実現も可能になると考えられる。

手法2では集団を1つの個体として考える際に、集団中のすべてのエージェントが同一の行動規則を持つ Homogeneous モデルや、集団中の各エージェントが別々の行動規則を持つ Heterogeneous モデルなどが提案され性能の評価実験などが行われている^{4)~6)}。

Homogeneous モデルは環境内のすべてのエージェントが同一の行動規則に従って行動するモデルであり、GP においては図1のように、各エージェントは同じ木を参照する。各エージェントは同一の行動規則に従うが、各々の置かれた位置や状態に応じてさまざまな行動をとることが可能であるため、協調して問題を解決することが可能である。

一方、Heterogeneous モデルでは環境内の各エージェントが別々の行動規則に従って行動する。GP においては図2のように、一個体中に各エージェントが参照する複数の木を保持することになる。このため Heterogeneous モデルでは交叉の方法として、ある個体のエージェント1が参照する木は、別の個体のエージェント1が参照する木とのみ交叉を行うといったように、対応する木どうしに交叉を制限する Restricted Breeding や、任意の木どうしで交叉を行う Free Breeding などさまざまな手法が提案されている^{4),5),7)}。しかし GP の染色体である木構造を別々のプールに分解することで、多様性を維持し、エージェントの専門化、特殊化の促進により役割分担の発生を高めるという観点から、交叉を対応する木どうしに制限する Restricted Breeding が良いとされている⁴⁾。

大規模なエージェント集団による協調行動の創発に関する従来の研究では、Homogeneous モデルを用いて、同一の行動規則を持つエージェントの相互作用によって全体としてより高いレベルの秩序ある群行動を創発し、問題を解決する試みが数多くなされている^{3),8)}。しかし複雑なタスクの処理には、いくつかのグループが各々異なる役割を担い、チーム内で分業を行うといった協調行動が不可欠である。そこで本論文では GP による進化の過程で、各エージェントの行動規則とともに集団のグループ構造を最適化する新しい交配戦略として、自動グループ構成手法 (ADG; Automatically Defined Groups)^{9),10)}を提案する。

本論文の構成は以下のとおりである。2章では提案

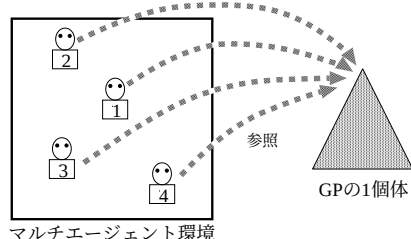


図1 Homogeneous モデル

Fig. 1 Homogeneous model.

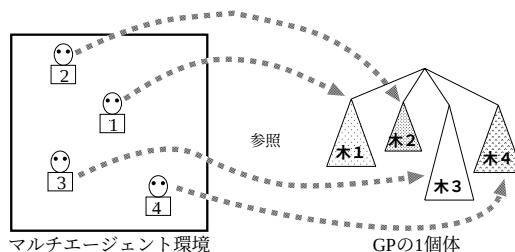


図2 Heterogeneous モデル

Fig. 2 Heterogeneous model.

手法 ADG について説明する。そして3章では ADG を簡単な荷物搬送問題に、4章では拡張したタイルワールド問題に適用し、各々実験の結果とその考察について述べる。最後に5章で結論と今後の課題について述べる。

2. 自動グループ構成手法 (ADG)

2.1 従来手法の問題点

Heterogeneous モデルでは、環境内のエージェントが各々の役割に応じて特殊化され、Homogeneous モデルに比べて、より複雑な協調動作を実現できると考えられる。しかし GP の一個体がエージェントの数だけ木構造を保持する必要があるため、環境内のエージェント数が増加すると、それに応じて木構造の探索領域が膨大になり GP による探索効率が悪化してしまう。実際にこれまでに行われている GP によるマルチエージェントの協調行動の創発に関する研究においても、探索効率の悪化などの要因から、Heterogeneous モデルの適用はエージェント数が2~4といった比較的小さな数の環境にとどまっている^{4)~6),11)}。

一方、自然界の蜂や蟻などは大規模なマルチエージェントシステムの例と見なすことができる。蜂や蟻のような社会性昆虫は生存に必要な役割分担数に応じていくつかの階級にグループ分けされ、各自が所属する階級に応じた仕事を行い、全体として高度なシステムを実現している。たとえば1つの巣の中の何百という働

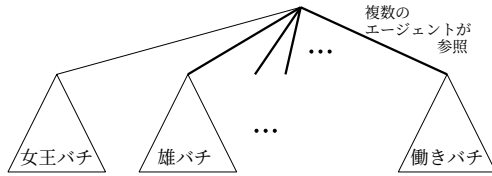


図3 蜂の階級

Fig. 3 Classes of bee colony.

きバチは同じ行動規則に従って行動していると考えられる。この集団を GP の個体として表現すると図 3 のように、全体としては Heterogeneous モデルのように複数の木構造を持つが、各々の木は Homogeneous モデルのように複数のエージェントが参照するといった 2 つのモデルが混在した形となる。この構造を GP に応用すれば探索領域が膨大になることを抑え、複雑な協調行動を実現できると考えられるが、あらかじめ問題解決に必要な役割分担数を知ることは困難である。そこで本手法では、進化の過程で最適なグループ構成の獲得を自動的に行う。これにより探索領域が膨大になることを抑え、効率的に探索を行うことができ、また逆に得られたグループ構成は問題解決に必要な役割分担や協調行動などを知る手がかりになると考えられる。

2.2 提案手法 (ADG) の定義

ここでは行動決定のために同一の木を参照するエージェントの集合をグループと定義する。GP の各個体はいくつかの木構造を持ち、それぞれの木は異なるグループの特別なプログラムとして機能する。以下に、ADG における初期集団の生成、遺伝操作、選択の方法について述べる。

- 初期集団の生成方法
ランダムにグループ分けされた個体を生成する。
 - 交叉の方法
基本的には Restricted Breeding のように対応する木の間で交叉を行う。具体的な処理手順を以下に示す。
- (1) GP の個体集団から任意に選んだ 2 個体に対し、エージェント集合から任意に 1 つのエージェントを選び、そのエージェントが参照する木を交叉に使用する木とする。
 - (2) 選択された木 T , T' を参照しているエージェントの集合 $A(T)$, $A(T')$ を各々求める。
 - (3) 選択された木 T , T' から交叉する部分木を各々任意に決定し交叉を行う。このとき木 T と木 T' を交叉する際に、
- (Type a) $A(T) = A(T')$ ならば、各個体のグルー

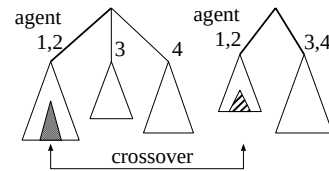


図4 交叉の例 (Type a)

Fig. 4 An example of crossover (Type a).

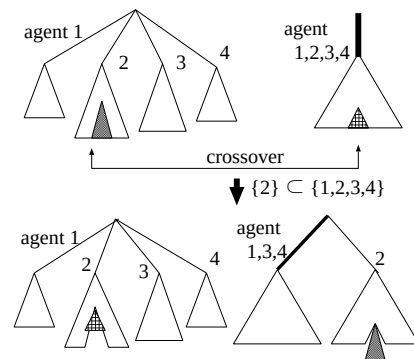


図5 交叉の例 (Type b)

Fig. 5 An example of crossover (Type b).

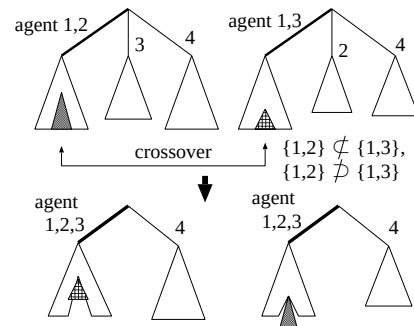


図6 交叉の例 (Type c)

Fig. 6 An example of crossover (Type c).

プ構造は変化しない。

たとえば、図 4 はエージェント 2 が参照している木どうしで交叉する場合である。交叉される木を参照するエージェントの集合が等しいため、Restricted Breeding と同様に部分木の交換を行う。各個体のグループ構成は変化しない。

(Type b) $A(T) \supset A(T')$ または $A(T) \subset A(T')$ ならば、 $A(T) \cap A(T')$ のみが交叉の影響を受けるよ

うに、グループの分割が起こる。 $A(T) \supset A(T')$ の場合は、 T を持つ個体にグループの分割が起こり、 T' を持つ個体のグループ構造は変化しない。逆に、 $A(T) \subset A(T')$ の場合は、 T' を持つ個体にグループの分割が起こり、 T を持つ個体のグループ構造は変化しない。

たとえば図 5 は、エージェント 2 が参照している木どうしで交叉する場合である。交叉する木を参照するエージェントの集合は、左側の個体は $\{2\}$ 、右側の個体は $\{1, 2, 3, 4\}$ であり、包含関係にある。左側の個体の交叉される部分木はエージェント 2 に特化したものであり、単純に交叉すると右側の個体のエージェント 1, 3, 4 にまで影響を与え、好ましくないと考えられる。そのため右側の個体で、エージェント 2 が参照している木と同じ木を新たに生成し、その木を参照する新しいグループにエージェント 2 だけを移す。結果として、2 つのエージェント集合に共通な要素であるエージェント 2 が参照する木のみが交叉に使用されるようにグループの分割が起こり、それから部分木の交換が行われる。共通部分集合中のエージェントのみが交叉の影響を受けるこのグループ分割操作は、従来の Restricted Breeding の考え方を適用したものであり、各グループの専門化、特殊化を促進すると考えられる。

(Type c) $A(T)$ と $A(T')$ が包含関係にない場合、どちらの個体も、 $A(T) \cup A(T')$ が同一の木を参照するようにグループの統合が起こる。

たとえば図 6 は、エージェント 1 が参照している木どうしで交叉する場合である。交叉する木を参照するエージェントの集合は、左側の個体は $\{1, 2\}$ 、右側の個体は $\{1, 3\}$ であり、包含関係にない。左側の個体はエージェント 1 と 2 が同じ行動規則に従い、右側の個体はエージェント 1 と 3 が同じ行動規則に従って行動しているため、次世代に生成される子はその和集合であるエージェント 1, 2, 3 が同じ行動規則を持つべきであると考え、それらが同じ木を参照するようにグループを統合した後、交叉を行う。このグループ統合操作は、2 つの親が各々持つエージェントのグループ関係という形質 (図 6 の例では、左側の親はエージェント 1, 2 が同じグループを形成し、右側の親はエージェント 1, 3 が同じグループを形成するという形質) をともに受け継ごうとした結果であると考えることができる。

● グループ突然変異

上述のとおり、ADG では交叉に使用される木を参照するエージェント集合の違いにより、グループの分割や統合が発生する。よって、もしすべての個体と同じグループ構造に収束した場合、グループ構造の変化が発生しない。そのため ADG では局所的な構造への収束を防ぐ目的で、グループ突然変異という遺伝操作を導入する。グループ突然変異では、あらかじめ定めたグループ突然変異率という確率で、エージェントを任意のグループに移す操作を行う。ただし移動先として選択されたグループが現在所属するグループであった場合は、現在参照している木と同一の木を新たに生成し、その木を参照する新しいグループにそのエージェントを移す。この操作は交叉によるグループ構造の変化を促進する働きを持つため、次世代の個体集団を生成する際に、交叉の前に実行する。

● 選択

次世代に子孫を残す親個体の選択には、トーナメント方式を用いる。ただし ADG では最も適応度の高い個体が複数存在した場合、グループ数の少ない方の個体を選択することにする。この条件により、グループの過度な分割を防ぐことができる。たとえば、Heterogeneous チームのすべてのエージェントが同じ内容の行動決定木を参照している場合、このチームの適応度は、すべてのエージェントが同じ行動決定木を参照する Homogeneous チームと同じであるかもしれない。この場合、Heterogeneous チームが持つ複数の木は余分なものであると考えることができる。このため選択の際に複数の最良個体が存在した場合は、グループ数の少ない方の個体を選択する。

このような枠組みで個体集団を進化させることにより、効率的に探索を行い、問題に対する適切な協調動作を獲得することが期待される。

3. 能力の異なる複数のエージェントによる荷物搬送問題への適用

3.1 問題定義

役割分担の発生の確認のため、マルチエージェントによる簡単な荷物搬送問題を設定し、本手法を適用した結果を示す。

設定した環境は以下のようなものである。

- 重さの異なる 2 種類の荷物を配置する。環境内に荷物の置かれている場所を 2 カ所設定し、一方には軽い荷物を、もう一方には重い荷物を配置する。

表 1 荷物搬送問題における終端・非終端記号の設定

Table 1 GP functions and terminals.

記号	機能
If_carrying_load(,)	荷物を持っているかどうかで分岐
If_load_here(,)	現在地の荷物の有無で分岐
Move_goal	ゴールの方へ向かう
Move_heavy_load	重い荷物の方へ向かう
Move_light_load	軽い荷物の方へ向かう
Pick_up	現在地に荷物があり、荷物を持っていない状態ならば拾う。ただしそれを持つ能力がなければ失敗。

各地点には大量の荷物が置かれており、なくなることはない。

- 荷物を運び込むゴール地点を 1 か所設定する。ゴール地点から各荷物置き場までの距離は等しい。
- 搬送能力の異なる複数のエージェントを配置する。全体のエージェント数は 20。そのうち番号 0, 4, 8, 12, 16 のエージェントは軽い荷物と重い荷物のどちらでも運ぶことができ、それ以外のエージェントは軽い荷物しか運べない。シミュレーション開始時は、すべてのエージェントはゴール地点からスタートし、単位時間ごとに各エージェントは自分が参照する GP 木を評価し行動を決定する。
- 一定時間内に、できるだけ多くの荷物を運ぶことを目標とする。チームの適応度として、定められた時間 (*Max_Step*) 内にゴール地点に運び込まれた荷物の数に応じて得点を与える。ただし重い荷物の得点 (*Heavy_Load_Score*) の方が軽い荷物の得点 (*Light_Load_Score*) よりも高いとする。なおエージェントは荷物を持ってゴール地点に着くと自動的に荷物を置き、その荷物は運び込まれた荷物としてカウントされる。
- 環境は 11×10 の 2 次元格子平面である。ゴール地点は、座標 (5,0) のセルに配置される。また軽い荷物置き場は (0,9)、重い荷物置き場は (10,9) に設定する。エージェントは上下左右に動くことができ、またエージェントや荷物などの各物体は同時に同じセルを占めることができる。

この問題における GP の終端・非終端記号の設定を表 1 に、GP や環境に関するパラメータの設定を表 2 にそれぞれ示す。木のノードの突然変異では、各個体に関して突然変異を適用するかについて、突然変異率の確率で決定し、選ばれた各個体の中で任意に選んだ 1 つのノードで突然変異を発生させる。

3.2 実験結果と考察

以上のような設定で ADG による実験を行った。図 7 にこの実験の結果を示す。この図は最大適応

表 2 GP、環境のパラメータ

Table 2 GP parameters.

個体数	1000
選択	トーナメント +エリート保存
トーナメントサイズ	5
交叉率	0.9
突然変異率 (各個体に関して 突然変異を適用するかどうかの確率)	0.9
グループ突然変異率	0.01
<i>Max_Step</i>	100
<i>Heavy_Load_Score</i>	5
<i>Light_Load_Score</i>	1

度を世代ごとにプロットしたものである。なおグラフは 10 回の実験の平均値である。今回の実験では、各エージェントは定められた時間内にゴール地点と荷物置き場を 3 往復することが可能であり、20 のエージェントの中で、5 つのエージェントが重い荷物も運ぶことができるので、とりうる最大適応度は、 $(5 \times \text{Heavy_Load_Score} + 15 \times \text{Light_Load_Score}) \times 3 = 120$ となる。このグラフから ADG では約 20 世代目で最適な行動規則が得られていることが分かる。

また 10 回の試行すべてにおいて、実行後獲得された木構造は、図 8 に示すように、あらかじめ重い荷物を運ぶことができるように設定した番号 0, 4, 8, 12, 16 のエージェントのグループと、それ以外のエージェントのグループの 2 つに分かれており、各々重い荷物を運ぶものと軽い荷物を運ぶものに役割が分担されていた。本手法で獲得された木構造を見ることにより、必要な役割分担数やどのエージェントが共通の役割を担っているかを知ることができる。

また図 9 に、ある 1 回の GP 実行時における各世代の個体群の平均グループ数の変化を示す。実行開始時はランダムにグループ分けされた個体群が、次第に最適なグループ数 2 に近付きながら探索を行っていることが分かる。

次に本手法の有効性を示すため、同じ環境、同じパラメータ設定を用いて、従来の Heterogeneous や Homogeneous 集団での実験も行った。このときの適応度の変化も図 7 に示されている。なおグラフは各々 10 回の実験の平均値である。Heterogeneous 集団では、10 回の試行のうち、最大の適応度が得られたのは最短でも 164 世代目であった。本手法に比べ進化に多くの時間が必要になったのは、Heterogeneous の個体ではエージェントの数と同じ 20 個の木構造を最適化する必要があり探索領域が膨大になったためと考えられる。またこの問題では探索に十分な時間をかける

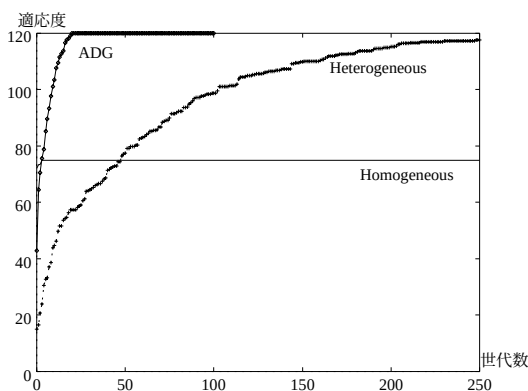


図7 ADGと従来手法における適応度の変化

Fig. 7 Comparison of the search performance among three methods.

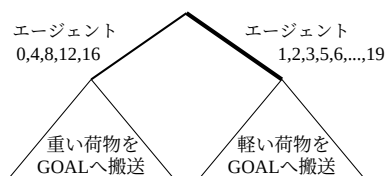


図8 獲得された木構造

Fig. 8 Acquired tree structure.

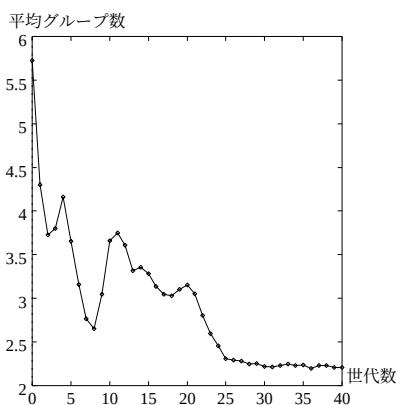


図9 平均グループ数の変化

Fig. 9 The change of the average number of groups.

ことにより、Heterogeneous 集団でも ADG と同じ適応度を最終的に得ることができるが、獲得された木構造では同じ働きをするエージェントどうしても別々のプログラムを参照しており、役割分担がどのように行われたかを知るには 1 つ 1 つのプログラムを解析する必要がある。ADG を使用することによりグループ化や分業の進化などを明示的に確認できることは、本手法の利点の 1 つである。一方、Homogeneous 集団では、1 つの木構造のみを最適化すればよいため、適応度は数世代で収束するが、最適な行動規則を得ること

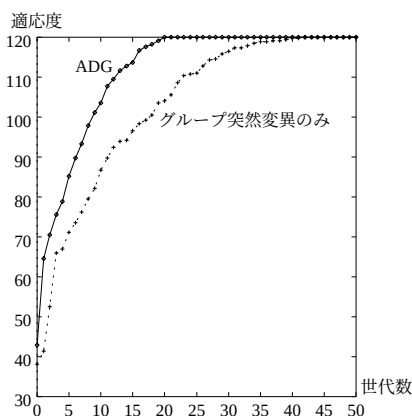


図10 ADGとグループ突然変異のみを用いた手法の比較

Fig. 10 ADG vs. Only Group Mutation.

はできなかった。これは能力の異なるエージェント集団を同一のプログラムで動作させるため、能力に応じた仕事ができないことが原因である。そのため獲得された行動は、能力によらずすべてのエージェントが重い荷物を運ぶを試みるもので、適応度の最大値は、 $5 \times \text{Heavy_Load_Score} \times 3 = 75$ となっている。この実験から、ADG を用いることにより、探索領域を抑えながら役割分担のような協調動作を実現できることが分かる。

さらに ADG における交叉によるグループ操作の効果を確認するため、グループ構造の変化にグループ突然変異のみを使用する手法との比較を行った。この手法における交叉は、グループの分割や統合を行わずに部分木の交換のみを行う Restricted Breeding であり、その他の遺伝操作などについては ADG と同様に行う。実験はともに表 2 に示すパラメータで行った。図 10 にこの実験の結果を示す。なおグラフは各々 10 回の実験の平均値である。このグラフから、収束速度において、グループ突然変異のみを用いる手法よりも ADG の方が優れていることが分かる。また ADG では、10 回の試行すべてで 2 つのグループからなる最適な構造が得られた。これに対して、グループ突然変異のみを用いた手法では、100 世代終了時に最適な構造が得られたのは 10 回の試行中 1 回だけであり、必要以上にグループ分けされた構造が多く見られた。以上のことから ADG の交叉による分割や統合が効果的であることが分かる。

4. タイルワールドへの適用

3 章では ADG が主に探索領域の抑制の点で効果的であり、また獲得された木構造は問題解決に必要な分業の理解に役立つことを示した。ただし獲得された行

表 3 タイルワールドにおける終端・非終端記号の設定
Table 3 GP functions and terminals for the Tile World.

記号	引数の数	機能
+	2	2つのベクトルを加える.
-	2	第1引数のベクトルから第2引数のベクトルを引く.
*2	1	ベクトルの大きさを2倍にする.
/2	1	ベクトルの大きさを半分にする.
->90	1	ベクトルを時計回りに90度回転させる.
inv	1	ベクトルを反転させる. たとえば, 入力があるとき $-\vec{v}$ を返す.
ifdot	4	4つの引数 x_1, x_2, x_3, x_4 に対して, ($x_1 \cdot x_2 \geq 0$) の場合は x_3 を, そうでない場合は x_4 を返す.
if>=	4	4つの引数 x_1, x_2, x_3, x_4 に対して, ($ x_1 \geq x_2 $) の場合は x_3 を, そうでない場合は x_4 を返す.
Last	0	エージェント (自分) に対する GP 木の最後の出力ベクトル. 最初に動くときは 0 ベクトルを返す.
Goal	0	エージェント (自分) からゴールへのベクトル.
Nearest-Tile	0	エージェント (自分) から最も近いタイルへのベクトル.
Same-Group	0	同じグループに属するエージェントのうち, 自分から最も近いエージェント (ただし同じ位置にいるエージェントは除く) へのベクトル.

動については Heterogeneous モデルによるものと同じであり, グループ化の有効性を示すものではなかった. この章ではグループ構造の違いによる効率への影響という点から ADG の有効性について検討する.

4.1 問題定義

マルチエージェントの標準的なテスト環境としてよく知られているものにタイルワールド¹²⁾がある. タイルワールドは 2次元格子上に, エージェント, タイル, ゴール, 障害物を配置した環境である. エージェントは上下左右に単位時間あたり 1 セル動くことができる. エージェントに隣接するセルにタイルがある場合, その方向にエージェントが移動することによりタイルを押すことができ, 各々が隣接するセルへと移動する. エージェントの目標はすべてのタイルをゴール地点へできる限り早く運ぶことである.

ここで扱う実験環境は, このタイルワールドを拡張した問題である. 具体的な問題設定は以下のとおりである. 環境はトラス状の 2次元格子平面であり, その上に複数のエージェントとタイル, そして 1つのゴール地点が配置される. 各タイルは 1つのセルを占め, 他のタイルやエージェントと重なることはないが, 複数のエージェントは同一のセルを占めることが可能である. エージェントに隣接するセルにタイルがある場合, その方向にエージェントが移動することによりタイルに力を加えることができる. この問題で最も重要な点は, タイルを動かすためには複数のエージェントが同時にかつ同じ方向にタイルを押すことが必要となることである. 各エージェントはタイル方向に移動することにより, タイルに大きさ 1 の力ベクトルをかけることができる. タイルにかかるすべての力の合成

ベクトルが, ある一定の値 *Power_Radius* よりも大きくなったとき, エージェントはそのタイルをその合成ベクトルの方向に運ぶことができる. この問題の目的は一定時間内に, ゴール地点にすべてのタイルをできる限り早く運ぶことである. また今回の実験では簡単のため, 障害物の配置やタイル, ゴール地点の確率的な出現消滅という条件は省略した.

この問題における GP の終端・非終端記号の設定を表 3 に示す. GP の木表現にはベクトル操作を選び, 終端記号の型と非終端記号の引数, 返り値の型がいずれも 2次元ベクトルとなるように設定した. これらの記号は, 文献 4), 5) における記号設定を参考に設定した. 各エージェントは単位時間ごとに自分の参照する GP 木を評価し行動を決定する. GP 木はつねに 2次元ベクトルを返し, そのベクトルがエージェントのとりうる行動に変換される. エージェントのとりうる行動は, その場にとどまる, 上下左右のいずれかに移動の 5つである.

GP 木の出力である 2次元ベクトル $\vec{v}$ から, エージェントのとりうる行動への変換は次のように行われる. ベクトルの大きさ $|\vec{v}|$ がパラメータ *Radius* 以下である場合, エージェントは現在の位置にとどまる. そうでなければ, ベクトル $\vec{v}$ を上下方向と左右方向の成分に分割し, 最大の成分の方向へ 1 セル移動する. タイルの移動方向も, かかる力の合成ベクトルとパラメータ *Power_Radius* を用いて同様に決定される. タイルが動かない場合は, その方向に移動しようとしたエージェントはその場にとどまることになる.

GP の各個体の適応度は, 一定時間 (*Max_Step*) 内でいかに効率良く仕事を完了できるかによって評価を

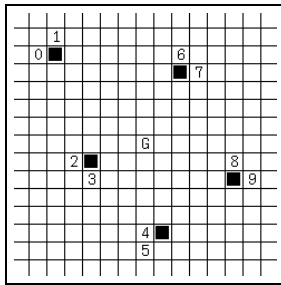


図 11 タイルワールドの環境 (初期配置)
(0, ..., 9: エージェント, ■: タイル, G: ゴール)
Fig. 11 The environment (initial position)
for the Tile World.

行う。そこで、以下の 3 つの要素を考慮する。

- ft : 時間内に運び込まれたタイルの数
- t_F : すべてのタイルをゴールへ運び終えるのにかかったタイムステップ数。仕事が完了しなかった場合は、 $t_F = \text{Max_Step}$ となる。
- 仕事が完了しなかった場合、タイルをゴールの近くまでどのくらい押すことができたか。

これらの要素を、パラメータ $Bonus$, $Speed$, $Dist$ で各々重み付けする。つまり適応度は次のように計算される。

$$f = Bonus \times ft + Speed \times (\text{Max_Step} - t_F) + Dist \times \sum_{t \in LT} \{MD(og(t), Goal) - MD(cr(t), Goal)\} \quad (1)$$

ここで LT はゴールへ運ばれずに残されたタイルの集合である。 $MD(x, y)$ は x と y のマンハッタン距離を表し、また $og(t)$, $cr(t)$ は各々、タイル t の初期位置と現在の位置を表す。よって、 $\{MD(og(t), Goal) - MD(cr(t), Goal)\}$ はゴールへのタイル t の移動距離を表す。この適応度は、各タイルをゴールへ近付けるほど高く、またすべてのタイルを運び終えた場合は費やしたステップ数が少ないほど高くなるように設定されており、できるだけ多く、効率的にタイルをゴールへ運ぶ行動が学習できる⁵⁾。

今回の実験では、10 のエージェント、5 つのタイル、1 つのゴール地点を図 11 に示すように重なりなく配置する。また $Power_Radius = 3.0$ 、すなわち 1 つのタイルの搬送には 3 つのエージェントが同じ方向に同時に押す必要があるように設定する。実験で利用したパラメータを表 4 に示す。

4.2 実験結果と考察

以上のような環境の下で ADG による実験を行った。図 12 は ADG により獲得された最良個体のチーム構造を、図 13 は実際にエージェント 1, 2, 4 が参照す

表 4 GP, 環境のパラメータ
Table 4 GP parameters.

個体数	1000
選択	トーナメント + エリート保存
トーナメントサイズ	5
交叉率	0.8
突然変異率	0.9
グループ突然変異率	0.075
$Radius$	1.0
$Power_Radius$	3.0
Max_Step	200
$Bonus$	2500
$Speed$	80
$Dist$	100

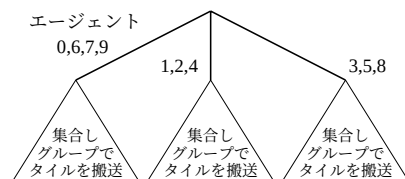


図 12 獲得された木構造
Fig. 12 Acquired tree structure.

る木構造を示したものである。図 12 から、この集団は 3 つのグループに分割されており、各グループが 3 つ以上のエージェントを含んでいることが分かる。また図 14 は獲得された最良個体による動作の一場面を示したものである。この動作の様子を観察すると、同じグループに属しているエージェントが集団を形成し、各々のグループがタイルをゴール地点へ運ぶというグループ行動が見られた。各グループのプログラムは複雑であり解析は困難であるが、そのプログラム中には、同じグループのエージェントが集合しタイルをゴール地点へ運ぶという行動規則が形成されていると考えられる。

また ADG の効果を確認するため、同じ環境の下で従来の Homogeneous や Heterogeneous モデルでの実験も行った。Homogeneous モデルはすべてのエージェントが同一グループに属する集団と見なすことができ、一方 Heterogeneous モデルでは各々が別々のグループを構成する集団と見なすことができる。ただし、Heterogeneous での実験では公平のため、終端記号 Same-Group ではなく最も近い別グループのエージェント (ただし同じ位置にいるエージェントを除く) へのベクトルを表す Different-Group を用いた。このときの適応度の変化を図 15 に示す。今回の実験では式 (1) から分かるように、5 つのタイルを運び終えたときに得られる適応度は、 $Bonus \times 5 = 12500$ 以上


```

(+ (->90 (+ Same-Group Nearest-Tile ))(ifdot
Nearest-Tile (- (if>= (->90 (if>= (/2 Last ))(inv
(if>= (/2 (+ (inv Goal ))(ifdot (if>= Nearest-Tile
Same-Group Last Last ))(->90 Goal ))(/2 Nearest-
Tile ))(if>= Last Goal (if>= (inv (inv Nearest-Tile
)))(ifdot (->90 Same-Group ))(ifdot Goal Nearest-
Tile Last Same-Group ))Last Same-Group ))(/2
(+ Same-Group Last ))(/2 Last ))Goal )))(inv
Goal ))(+ Nearest-Tile (if>= Last Nearest-Tile (-
Nearest-Tile Goal ))(*2 Last ))Same-Group ))(if-
dot Last (- (inv Goal )Goal ))(inv Nearest-Tile
))(->90 Goal ))(- Same-Group (inv Same-Group
)))(if>= (+ (ifdot (ifdot Last Same-Group Same-
Group (*2 Goal ))(if>= (->90 (if>= (- (+ Goal
Same-Group ))(/2 Same-Group ))Goal (if>= (inv
(+ Goal (inv (inv (inv Same-Group )))))(+ (ifdot
Same-Group Goal Last Goal ))(if>= Goal Same-
Group Goal Nearest-Tile ))(+ Nearest-Tile (if>=
(/2 (if>= Last (+ Last Same-Group ))Same-Group
(+ Goal Goal )))Same-Group Last Nearest-Tile
))(+ Last Goal ))(ifdot Nearest-Tile Nearest-Tile
(*2 (if>= Last Same-Group Last Goal ))(->90
(->90 Nearest-Tile ))))(- (inv (/2 Last ))(*2 (-
(inv Nearest-Tile ))(ifdot (->90 Last ))(+ (ifdot
Same-Group Same-Group (if>= Goal Nearest-Tile
(+ Same-Group Goal ))(inv Nearest-Tile ))(+ Last
Same-Group ))Goal ))(ifdot (- Same-Group Same-
Group ))(/2 (+ Last (ifdot Goal (inv (->90 (-
(- Goal Same-Group ))(->90 Last )))))(inv Last
))Last )))(if>= Goal Same-Group Last Nearest-
Tile ))Same-Group ))Same-Group ))))Goal Last
))(->90 (- (*2 (if>= (- (->90 Last ))(/2 (if>=
Goal Last Goal Last ))))Goal (+ Goal Nearest-
Tile ))Nearest-Tile ))Last ))Same-Group ))Last
))Nearest-Tile Nearest-Tile (->90 Last ))Nearest-
Tile Last ))Goal ))Last (ifdot Last Last (+ (if>=
(+ Last Last ))(+ (+ Same-Group Last ))Goal
))Nearest-Tile Nearest-Tile ))(+ Nearest-Tile Same-
Group ))(ifdot (- (/2 (inv (*2 Nearest-Tile )))(/2
(+ (->90 (+ Nearest-Tile Last ))(*2 (*2 Last
))))))Last Same-Group (/2 (->90 Last )))))))

```

図 13 エージェント 1, 2, 4 が参照する木構造

Fig. 13 Acquired program for agent 1, 2, 4.

であり、より短い時間で効率良くタスクを完了する個体ほど、さらに高い適応度を得ることができる。Homogeneous 集団では Same-Group により、次第に集団を構成してタイルをゴールへ運ぶ行動が獲得されているが、独立した小集団による並列処理が実現できておらず、効率では ADG に劣っている。また Heterogeneous 集団では、各エージェントの木を最適化する必要があるため探索領域が膨大になり、タスクを完了する集団は獲得できなかった。この実験から、効率的な行動を行うには集団の適切なグループ化が必要であり、本手法によりそれが実現できていることが分かる。

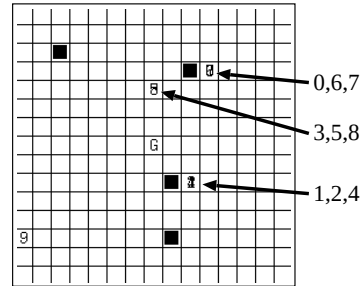


図 14 進化により得られたマルチエージェントの行動

Fig. 14 A snapshot of behavior of agents controlled by an evolved program.

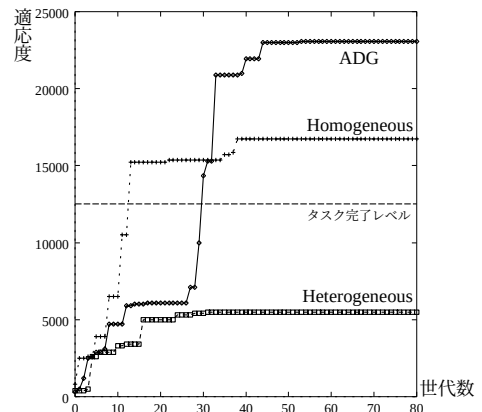


図 15 タイルワールドにおける ADG と従来手法との比較

Fig. 15 Comparison of the search performance among three methods for the Tile World.

5. おわりに

本論文では、GP による進化の過程で、マルチエージェントの最適なグループ構成を自動的に獲得する手法について述べた。本手法は、役割分担の発生などを促進するため交叉として Restricted Breeding を用い、その際交叉する木を参照するエージェントの集合の関係に応じて、グループの分割や統合を行うことによってチーム構成の最適化を行っている。このシステムをマルチエージェント環境における荷物搬送問題や拡張したタイルワールド問題に適用し、最適なチーム構成が獲得できることを確認した。また従来の Heterogeneous や Homogeneous モデルと進化の速度や性能を比較し、探索領域を抑え効率的に探索を行いながら、役割分担のような複雑な協調動作が実現できることを示した。今後はエージェントの数、必要な役割分担数、協調の形態などが異なるさまざまな問題に対して本手法を適用しその有効性を示すとともに、理論的な解析を行う必要がある。

参 考 文 献

- 1) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, The Univ. Michigan Press (1975).
- 2) Goldberg, D.E.: *Genetic Algorithms in search, optimization, and machine learning*, Addison Wesley (1989).
- 3) Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press (1992).
- 4) Luke, S. and Spector, L.: Evolving Teamwork and Coordination with Genetic Programming, *Genetic Programming 1996: Proc. 1st Annual Conference*, Koza, J.R., Goldberg, D.E., Fogel, D.B. and Riolo, R.L. (Eds.), pp.150–156, MIT Press (1996).
- 5) Iba, H.: Emergent Cooperation for Multiple Agents Using Genetic Programming, *Parallel Problem Solving from Nature IV, Proc. International Conference on Evolutionary Computation*, Voigt, H.-M., Ebeling, W., Rechenberg, I. and Schwefel, H.-P. (Eds.), LNCS, Vol.1141, pp.32–41, Springer Verlag (1996).
- 6) Iba, H.: Multiple-Agent Learning for a Robot Navigation Task by Genetic Programming, *Genetic Programming 1997: Proc. 2nd Annual Conference*, Koza, J.R., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M., Iba, H. and Riolo, R.L. (Eds.), pp.195–200, Morgan Kaufmann (1997).
- 7) Haynes, T. and Sen, S.: Crossover Operators for Evolving A Team, *Genetic Programming 1997: Proc. 2nd Annual Conference*, Koza, J.R., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M., Iba, H. and Riolo, R.L. (Eds.), pp.162–167, Morgan Kaufmann (1997).
- 8) Drogoul, A. and Ferber, J.: From Tom Thumb to the Dockers: Some Experiments with Foraging Robots, *From animals to animats 2*, pp.451–459, MIT Press (1993).
- 9) Hara, A. and Nagao, T.: ADG; Automatically Defined Groups for multi-agent cooperation, *Proc. 2nd Japan-Australia Joint Workshop on Intelligent and Evolutionary Systems*, pp.91–98 (1998).
- 10) Hara, A. and Nagao, T.: Emergence of cooperative behavior using ADG; Automatically Defined Groups, *GECCO-99: Proc. Genetic and Evolutionary Computation Conference*, pp.1039–1046, Morgan Kaufmann (1999).
- 11) Andre, D.: The Evolution of Agents that Build Mental Models and Create Simple Plans Using Genetic Programming, *Genetic Algorithms: Proc. 6th International Conference (ICGA95)*, Eshelman, L. (Ed.), pp.248–255, Morgan Kaufmann (1995).
- 12) Pollack, M. and Ringuette, M.: Introducing the Tileworld: Experimentally Evaluating Agent Architectures, *Proc. 8th National Conference on Artificial Intelligence*, pp.183–189, MIT Press (1990).

(平成 11 年 5 月 26 日受付)

(平成 12 年 2 月 4 日採録)

原 章 (学生会員)



1974 年生。1997 年東京工業大学電気・電子工学科卒業。1999 年同大学大学院総合理工学研究科物理情報工学専攻修士課程修了。現在、同大学院総合理工学研究科物理情報システム創造専攻博士後期課程在学中。進化的計算法、マルチエージェント等に関する研究に従事。

長尾 智晴 (正会員)



1959 年生。1985 年東京工業大学大学院博士後期課程中退。同年同大学工学部附属像情報工学研究施設助手。1995 年同大学工学部附属像情報工学研究施設助教授、現在に至る。工学博士。画像工学，進化的計算法，神経回路網，マルチエージェント，進化経済学，人工生命等に関する研究に従事。著書「画像の処理と認識」(昭晃堂)，「ジェネティックアルゴリズム」(昭晃堂)等。電子情報通信学会，人工知能学会，計測自動制御学会，進化経済学会，映像情報メディア学会，日本視覚学会，日本印刷学会，IEEE 等会員。