

1. はじめに

ある種のタスクを複数のエージェント間の部分タスクに分離することをマルチエージェントシステムという。マルチエージェントシステムで、あるエージェントの行動が結果的に他のエージェントのタスク遂行を助けるとき、その行動は協調的であるといい、複数のエージェントたちを協調に導くような規則を発見することにより効率的に問題を解くことができる。

そこで遺伝的プログラミング (GP) を用いて多数のエージェントのための協調的な行動を探す。

2. 遺伝的プログラミング

遺伝的アルゴリズム (GA) は、進化論的な考え方に基づいてデータを操作し、最適化の問題や学習、推論を扱う手法である。GA では、以下のようなアルゴリズムに従う。

1. ランダムに初期世代の集団 $M(0)$ を生成する
2. 現在の集団 $M(t)$ 内の各個体 m に対して適合度 $u(m)$ を計算する
3. $u(m)$ に比例する確率分布を用いて $M(t)$ から個体 m を選び出す
4. 選び出された個体に GA オペレータを作用させて、次の世代の集団 $M(t+1)$ を生成する
5. 2. に戻る

遺伝的プログラミング (GP) は、GA の遺伝子型としてグラフ構造や木構造などの構造的表現を用いたものである。今回は、木と呼ばれるサイクルを持たないグラフを扱う。

3. タイルワールド

実験には仮想的なロボットエージェントのシミュレーションであるタイルワールドを用いる。この世界はエージェント、タイル、障害物、穴からなる。エージェントは障害物や壁に当たらない限り、上下左右に動くことができ、またタイルに隣接するエージェントはその方向に動くことでタイルを押すことができる。ただし、タイルの先が壁や障害物であるときは押せない。タイルの動いた先が穴であったとき、そのタイルは穴に落ちて消える。エージェントの目標は全てのタイルをできるだけ早く穴に落とすことである。

図 1 のような単純な場合 (TW1) を考える。



T : タイル
: 障害物
V : 穴
Ai : エージェント i

図 1 TW1

エージェント A0 だけで全てのタイルを穴に入れようとする 17 ステップが必要となる。しかし両エージェントが協力して仕事に当たればそれより低いステップで全てのタイルを落とすことができる。

4. タイルワールドと遺伝的プログラミング

エージェントの行動は GP で進化する木のプログラムで規定される。木を解釈することでエージェントのとりべき行動が決定される。エージェントのプログラムの終端・非終端記号の一部を表 1 のようにする。引数 0 の記号は終端記号である。非終端ノードはベクトル操作を行い、各部分木は 2 次元ベクトルを値として返す。GP の一つの木構造 (プログラム) は、エージェントがあるステップでいかに動くかをあらわす。そのために、wrapper 関数を GP 木の出力ベクトルに適用し、エージェントの動きを決定する。

wrapper 関数は 2 次元ベクトルから行動への写像である。もし出力ベクトルの大きさがパラメータ Radius 以下であるならば現在の位置に留まる (STAY)、さもなければその方向に応じて上下左右に 1 ステップ動く、となる。

名前	引数	内容
Tile	0	エージェントから一番近いタイルのベクトル
Hole	0	エージェントから一番近い穴へのベクトル
Agi	0	エージェントからエージェント Ai へのベクトル
If_dot	4	第 1 番目と第 2 番目の引数の内積により、第 3 または第 4 引数を評価する
If >=	4	第 1 番目と第 2 番目の引数の大きさを比較し第 3 または第 4 引数を評価する

表 1

エージェントは各タイムステップで Wrapper に従って行動する。タイムステップの上限は変数 Eval で設定する。

適合度 f は以下の要素によって決まる

1. 穴に落としたタイルの数
2. 仕事が終了した場合は残されたタイムステップ
3. 終了しなければタイルを穴に近づけた升数
つまり適合度は次のようになる

$$f = Bonus \times ft + Speed_Up \times (Evals - t_f) +$$

$$C_T \times \sum_{i \in LE} \{Dist(og(t), nr(t)) - Dist(cr(t), nr(t))\}$$

ft は実行終了後に穴に落とされたタイル数、 t_f

はすべてのタイルを穴に落とすのに費やされるタイムステップ、 $Dist(x,y)$ は x と y の距離を示す。 $og(t), cr(t), nr(t)$ はそれぞれタイル t の元の位置、現在地、最も近い穴の位置である。

5. 均質的交配戦略と異質的交配戦略

均質的交配戦略では、すべてのエージェントが同じプログラムに従って行動する。

それに対して異質的交配戦略では各エージェントは異なるプログラムに従う。N個のエージェントを用いる場合、N個の木のまとまりを1個体とし、それぞれの木が各エージェントのプログラムとなる。また、交叉は同じエージェントに相当する木同士のみにも適用される。(図2)

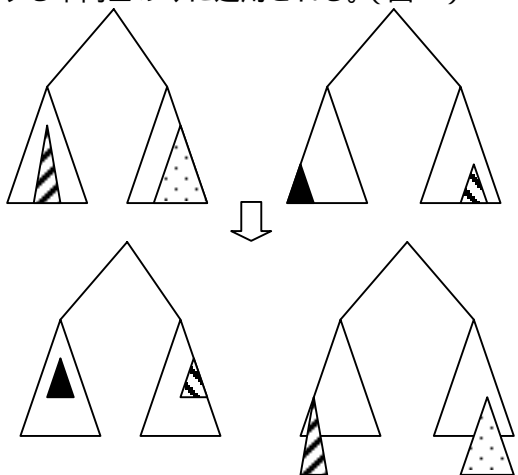


図2

表2のようなパラメータで、二つの手法の実験をした。グラフの適合度は10回の平均値をとる。

交叉確率	0.8	突然変異確率	0.1
Eval	50	最大世代数	50
Bonus	3000	Radius	1.0
Cr	100	Speed_Up	80

表2

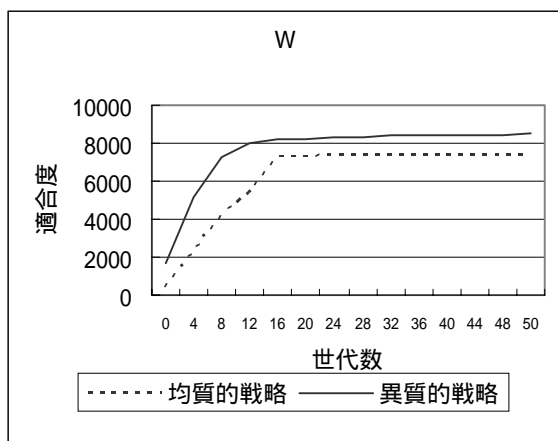


図3

図3はその結果である。仕事が完了した場合の適合度は6000となる。つまり、均質的交配戦略では14世代目、異質的交配戦略では6世代目にはすべてのタイルを穴に落とすことができた。

次に、図4のような場合(TW2)を考える。

A0	A1	A2	A3	A4		#	
						V	
			T	#			
	T						
						T	
#					V		

表3は10回実行したときの平均適合度である。(括弧内は最も成績の良かった回のタスク完了までのステップ数、-はタスク完了した回がなかったことを表す)

図4 TW2

Agent 数	均質的交配戦略	異質的交配戦略
1	1060(-)	
2	10488(28)	7990(39)
3	10504(27)	6352(-)
4	10808(23)	5814(37)
5	9036(23)	4880(-)

表3

また、タイル数を2にして同様に実験した(表4)

Agent 数	均質的交配戦略	異質的交配戦略
1	3120(-)	
2	8336(19)	8048(24)
3	7240(22)	7408(27)
4	7060(21)	6456(25)
5	6748(22)	5036(29)

表4

6. 考察

TW1のような狭い環境では異質的交配戦略のほうが成績が良かった。しかしTW2のような広い環境で複雑な動作を必要とする場合、均質的交配戦略のほうが成績が良い。これは、異質的交配戦略は多数の種類の木を操作するため、良い働きをする部分木(スキーマと呼ぶ)が後の世代に残る確率が低いからではないかと考えられる。

さらに、このプログラムにおいて協調の行動がみられ、複数のエージェントが協力しあって仕事が完了した。しかし、使用するエージェントの数が増えるにつれて、エージェントが他のエージェントの動きを邪魔するような行動をとり、成績が悪くなる様子もみられた。

配置するタイルの数によって、もっとも成績の良くなるエージェントの数は違う。また、今回採用した手法以外にも交叉の手法がないかを考えていきたい。

7. 参考文献

- [1]伊庭齊志:遺伝的プログラミング,p264,東京電機大学出版局(1996)
- [2]北野宏明編:遺伝的アルゴリズム3,p374,産業図書(1997)