

# Gur Game パラダイムにおける自己組織的パターン生成

4 G-8

荒田秀樹

高井昌彰

北海道大学工学部

北海道大学大型計算機センター

## 1 はじめに

分散システムの分野において、外界からの命令なしに、もしくは簡単な命令だけにより、ある処理要素の集合が協調して何らかの仕事を成し遂げる、ということは非常に興味深い問題であり、多方面に渡り研究が進められてきた。

Gur Game は複数のエージェントの協調による最適化モデルの1つであり、マルチエージェントシステムにおける自律分散処理への応用が報告されている [1]。

本稿では、この Gur Game パラダイムをパターン生成に応用することを考え、その初期段階として、エージェントを直線上に整列させるを試みる。幾何図形の構成要素である点を、平面上を移動するエージェントとしてとらえ、それぞれを有限状態オートマトンで構成する。オートマトンは、生成させたいパターンの幾何学的制約の充足度にしたがい状態遷移を繰り返す。オートマトンの状態数や問題サイズの相違によるエージェントの整列具合を調べ、Gur Game パラダイムのパターン生成に対する応用の可能性を探る。

## 2 Gur Game

図1に Gur Game の基本的な概念を示す。ある環境に対して、 $A_1$  又は  $A_0$  のいずれかを出力するエージェント集団を考える。この環境には、あらかじめ2値出力の reward 関数（エージェントには未知）が1つだけ定義されており、あるエージェントが reward を得る確率は、 $A_1$  を出力したエージェントの個数（割合）の関数として与えられる。Gur Game とは、このような状況において reward 関数を最大化する問題である。

Gur Game を解くエージェントは、状態数  $2n$  ( $-n, \dots, -1, 1, \dots, n$ ) の有限状態オートマトンで構成できる。 $n$  はメモリサイズと呼ばれる。オートマトンは状態が正の時  $A_0$ 、負の時  $A_1$  を出力する。オートマトンの入力 (reward/penalty) に対する状態遷移を

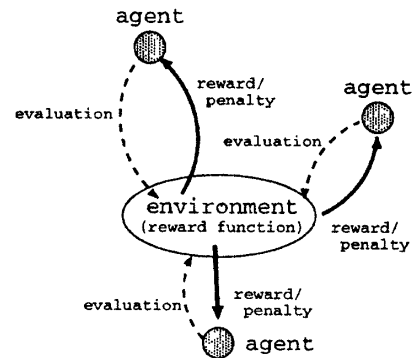


図1: Gur Game の枠組み

図2に示す。エージェントが  $m$  個ある時、これらの振舞いは  $m$  次元のランダムウォークと見なすことができる。ランダムな初期状態から出発し状態遷移を繰り返す後に、reward 関数が最大となる点が発見される。すなわち、 $A_1$  を出力するエージェント群と  $A_0$  を出力するエージェント群への最適な役割分担がおこる。

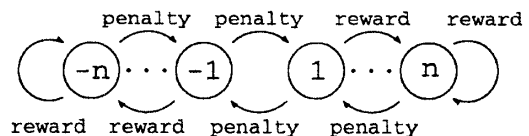


図2: Gur Game を解くオートマトン

## 3 パターン生成への応用

### 3.1 モデルの概要

Gur Game に基づくパターン生成モデルの概念図を図3に示す。各エージェントは幾何図形の構成要素である点を表し、平面上を自律的に移動する。エージェントの動きを評価する環境には、生成させたいパターンを示す幾何学的制約を与えておく。すなわち、エージェント集団は与えられた制約のソルバーとして機能することになる。

エージェントの環境に対する行動は、上下 (左右) への移動と停止の2種類であり、それぞれがオートマトンの出力と対応する。この行動に対し、エージェント

の座標値により幾何学的制約の充足度が環境によって計算され、その充足度にしたがって reward が与えられる。エージェントを構成するオートマトンは、reward が与えられるとその行動を続けるように、penalty が与えられると現在とは異なる行動をとるように状態遷移する。

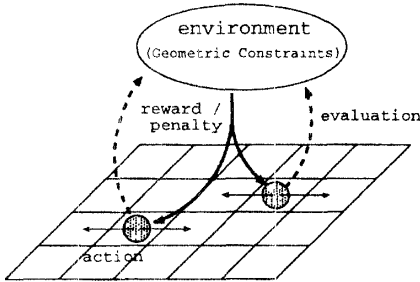


図 3: パターン生成モデルの概念図

### 3.2 直線の生成

パターン生成の初期段階として、2次元平面上でそれぞれ  $x$  座標の固定されたエージェント集団の直線上への整列を試みる。直線整列に用いるオートマトンは図 4 のように設計した。それぞれのエージェント

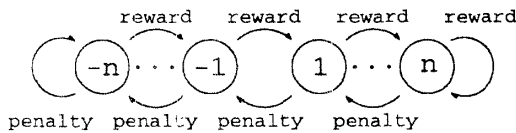


図 4: パターン生成に用いるオートマトン

は、オートマトンの出力が正のときに停止、負のとき  $y$  軸方向に 1 単位分ランダムに移動する。

エージェントに与える reward 関数は以下のように定義する。

$$\text{reward} = 1 - va/mva \quad (1)$$

ここで  $va$  は各エージェントの  $y$  座標の分散、 $mva$  は分散の最大値である。

実験では、メモリサイズと問題サイズの変更に対するエージェントの直線への整列具合を調べた。

#### 3.2.1 メモリサイズによる影響

エージェント数 5、メモリサイズが 2、3 の場合について実験を行った。メモリサイズが 2 (図 5 左) の場合は、高い reward を得られるが振動も大きい。これに対しメモリサイズ 3 (図 5 右) のときは、高い reward を安定して得られる特徴がある。

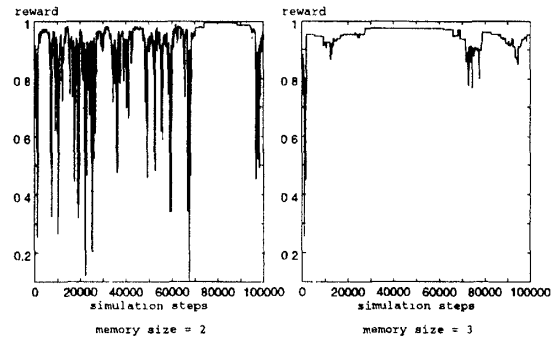


図 5: メモリサイズによる影響

#### 3.2.2 問題サイズによる影響

エージェント数が 10、 $y$  軸方向の移動範囲が 0 ~ 10、0 ~ 20 の場合について実験を行った。この場合、問題サイズは  $20^{10}/10^{10} = 1024$  倍となる。移動範囲が大きい方 (図 6 右) も小さい方 (図 6 左) も、ほぼ同程度の reward を得ていることから、更に大きな問題に対しても適用できると考えられる。

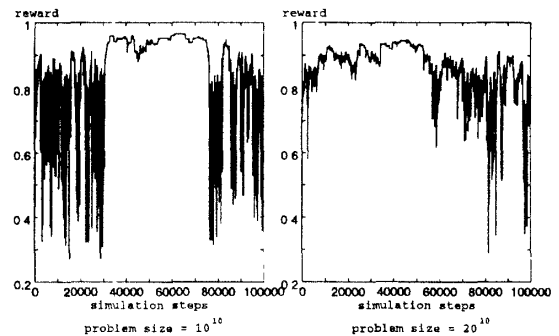


図 6: 問題サイズによる影響

## 4 まとめ

本稿では、Gur Game パラダイムのパターン生成への応用を試みた。簡単な直線の生成について実験を行い、その可能性を検証した。メモリサイズや reward 関数の形が整列に及ぼす影響について更に詳しく調べることが今後の課題である。

## 参考文献

- [1] B. Tung and L. Kleinrock: "Using finite state automata to produce self-optimization and self-control". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 4, pp. 439-448 (1996).