

引力斥力モデルに基づく集団行動の獲得

守田 了 東 慎吾

山口大学工学部

宇部市常盤台 2557

ロボットの集団行動は、ロボットによる工場の自動化や、顕微鏡サイズのマイクロロボットによる医療活動などに応用できる。個別に行動する自律ロボットを設計するよりもさらに対象やタスク、環境との相互作用が複雑になるため、ロボットの集団行動の実現は困難である。本稿では、ロボットの集団行動を簡易に実現するためにオブジェクト間の引力と斥力のモデルを導入する。集団行動を獲得するために必要な制御パラメータを、目的の集団行動を獲得するための評価関数の最小化問題を解くことによって獲得する。実際に複数のグループが混じっている集団が、グループごとに分かれる行動と衝突回避を行いながらゴミを集める行動を獲得することにより、本モデルの有効性を示す。

キーワード 集団行動, 引力斥力モデル, 学習

Learning Cooperative Behavior based on Attraction Repulsion Model

Satoru MORITA Shingo AZUMA

Faculty of Engineering, Yamaguchi University

2557 Tokiwadai, Ube, 755, Japan

Cooperative behavior of robots can be applied for the factory automation and the medical field supporting the doctor using micro robots. In this paper, we introduce the attraction repulsion model to realize the cooperative behavior of robots. Optimal control parameters is gotten by solving the minimization problem of the evaluation function to acquire the cooperative behavior of robots. Actually, we shows the effectiveness of this model by simulating the behavior that a group is divided into some groups and robots collect scattered many papers in a garbage box.

keyword: cooperative behaviour, attraction repulsion model, learning

1 はじめに

ロボット研究 [1] の目標の一つに、実環境で知的に行動するロボットの實現 [2] がある。その一連の研究の中で、従来の記号世界での推論を否定した Brooks の行動に基づく [3] の成功により、実環境でのロボットの行動 [4] への関心が高まりつつある。しかしながら、Behavior-based ロボットシステムは人間が設計した規則群によって制御されており、対象やタスク、環境との相互作用が複雑になるにつれて設計が複雑になる。

ロボットの集団行動は、ロボットによる工場の自動化や、近い将来實現されるであろう顕微鏡サイズのロボット [5] による医療活動などに応用できる反面、個別に行動する自律ロボットを設計するよりもさらに対象やタスク、環境との相互作用が複雑になるため實現が困難である。

Arkin は、AuRA (Autonomous Robot Architecture) [6] としてロボットの外部からの刺激を元に動作する反応制御に、ロボットの存在する世界の知識とロボットの行動を基にしたスキーマを組み合わせた制御法を提案した。AuRA で組み合わせられたスキーマにより障害を避けながら動くロボットを試作している。T. Balch [7] は Arkin の考えを基にロボットのチームにおいて個々のロボットが学習することにより、チームが多様な振舞いをすることを示した。Khatib はロボットの制御にポテンシャル (力場) を用いる手法を提案した [8]。ロボットの目的地に陽の電荷を、障害物に陰の電荷を置くことにより、ロボットは低いポテンシャルの障害物を避け、最終的にポテンシャルの高い目的地に着く。Reif と Wang [9] はポテンシャルを発展させ陰と陽の組み合わせを許可する方法を提案した。しかし、ここでは目的の集団行動を得るためのパラメータの決定法については述べられていない。

そこで、オブジェクト間の力の計算 [9] を用いて集団行動を定義し、集団行動に必要な制御パラメータを最小化問題を解くことによって獲得する手法を提案する。実際に複数のグループが混じっている集団が、グループごとに分かれる行動と、衝突回避を行いながらゴミを集める行動を獲得することにより、本モデルの有効性を示す。

2. ではロボットの制御を与えるための力の定義と制御に必要なパラメータを獲得する学習法について述べる。3. では集団行動の例をあげ、その集団行動を獲得するために必要な評価関数について述べる。4. では実際に集団を形成する行動と、ゴミ捨てる行動をシミュレーションすることにより、本モデルの有効性を示す。

2 力の計算によるロボットの制御

3 引力斥力モデル

簡易にロボットの集団行動を實現するために引力斥力モデル [9] を導入する。本節ではこの引力斥力モデルを定義する。

ある時間において、オブジェクト $1, 2, \dots, n$ がユークリッド空間内の X_1, X_2, \dots, X_n に存在している時、オブジェクト ij 間のユークリッド距離は、

$$r_{ij} = \|X_i - X_j\|_2 \quad (1)$$

である。この値を次式 (2) に代入し、オブジェクト j がロボット i に及ぼす力 F_{ij} を計算する。

$$F_{i,j}(r_{ij}) = \left(\sum_{k=1}^L \frac{c_{i,j}^{(k)}}{r_{ij} \sigma_{i,j}^{(k)}} \right) \left(\frac{X_j - X_i}{r_{ij}} \right) \quad (2)$$

$c_{i,j}^{(1)}, c_{i,j}^{(2)}, \dots, c_{i,j}^{(L)}$ と $\sigma_{i,j}^{(1)}, \sigma_{i,j}^{(2)}, \dots, \sigma_{i,j}^{(L)}$ はオブジェクト i, j の組と k によって定まるパラメータであり、ロボットの動きを決める要素である。 L は力の組合せの数である。

各項において、 σ は通常、正の数であることを要求するが、 c は正負どちらでも良い。 c の値が正である場合は引き付ける力、すなわちロボットを集団としてまとめるための引力として、負である場合はロボット同士の衝突回避を行うための斥力として力は働く。この式を他の全てのオブジェクトに対し適用し、その合計値 F_i を得る。

$$F_i = \sum_{j \neq i} F_{i,j}(r_{ij}) \quad (3)$$

この F_i をもとに、ロボット i の動作を決定する。例えば、 F_i の示す方向に F_i だけ動かす、 F_i の方向に一定量だけ動かす、 F_i をロボットの加速度に用いるなどの方法がある。式 (2) において、 $c_{i,j}^{(k)}$ が正ならその項は引力を、負なら斥力を示す。この2つをうまく組み合わせると離れているときには近寄り、近すぎるときには離れる行動を取らせてロボットを制御する。 $\sigma_{i,j}^{(k)}$ の値は、その2つの力が効く範囲に影響する。 σ が大きくなれば、遠くのオブジェクトからの影響力は小さくなり、逆に σ が小さければ遠くでも影響は小さくなくない。また、オブジェクト間の距離が近いときは σ の大小による差は小さくなる。

3.1 ロボットの制御のための力の計算

引力と斥力を用いてロボットを簡易に制御するために、この2つの力を1つずつ用いた最も簡単なモデルを

考える。引力と斥力を1つずつ持たせるため、式(2)において $L = 2$, $c_1 < 0$, $c_2 > 0$, また衝突回避の観点から $\sigma_1 > \sigma_2 > 0$ とする。ここではすべてのロボットで同じパラメータを使用し、平面上にロボットを展開することにより、最も単純なモデルを作成する。

このとき、すべてのロボットで作用する力は、

$$F(r) = -\frac{c_1}{r^{\sigma_1}} + \frac{c_2}{r^{\sigma_2}} \quad c_1, c_2 \geq 0, \sigma_1 > \sigma_2 > 0 \quad (4)$$

となる。このモデルにおいて2つしかロボットがない場合、平衡状態つまり $F(r) = 0$ となる時のロボット間の距離は、

$$d = \left(\frac{c_1}{c_2}\right)^{\frac{1}{\sigma_1 - \sigma_2}} \quad (5)$$

となる。

ロボットは、引力と斥力それぞれの力の合計がつかう場所に静止する。平面上のシミュレーションではロボットが4つ以上存在する場合、ロボットが他の全てのロボットに対し互いにこの距離を保つことが出来なくなるため、ロボット間の距離は次第に狭くなる。ロボット全てが統一されたパラメータを持ち、パラメータの変動がないならばロボットは1つのクラスタを形成する。クラスタの大きさは先に書いたようにパラメータとロボットの数に依存する。

図1は、 $L = 2$, $c_1 < 0$, $c_2 > 0$, $\sigma_1 > \sigma_2 > 0$ の条件を満たすように $c_1 = -60.0$, $\sigma_1 = 2.0$, $c_2 = 1.0$, $\sigma_2 = 1.0$ と設定したモデルにおいて、ロボットの数を2~100まで変化させたときの最終状態における最も近いロボット間の距離の平均をとったものである。この図は横軸にロボットの数、縦軸にロボット間の平均距離をとってグラフにしたものである。ロボットの数が増えるにつれて最も近いロボットまでの間隔が短くなっていくのがわかる。

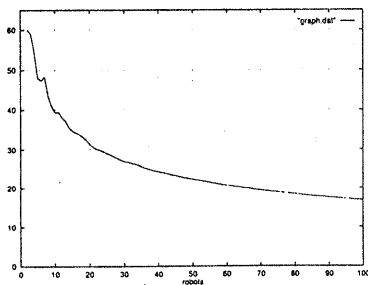


図1: ロボット数と平均最短距離

ロボット数が6から7に変化したときだけ平均距離が延びている。この理由であるが、ロボット数が7の場合ほぼ六角形の頂点と中心の位置にロボットが配置されるため、外側のロボット間の距離と中心のロボットとの距離はほぼ同じになる。一方ロボット数が6の場合は最終状態において五角形の頂点とその中心にロボットが位置するため、外側のロボット間の距離に対し中心のロボットとの距離が短くなり、ロボット数が8の場合は7角形とその中心に分布するために外側のロボット間の距離が中心のロボットとの距離より短くなる。そのため最も近いロボットとの平均距離をとった場合ではロボット数7よりロボット数6, 8の方が短くなる(図2)。しかし基本的には最も近いロボット間の平均距離はロボット数が増えるにしたがって単調減少する。



図2: ロボット数6,7,8の最終状態

また、図3は、 $c_1 = -60.0$, $\sigma_1 = 2.0$, $c_2 = 1.0$, $\sigma_2 = 1.0$ を用いたときの式(4)のグラフである。図3より、引力よりも距離が近い場合の斥力が急激に伸びていることが判る。式(5)にパラメータを代入して示される $r = 60$ を境に正負が入れ替わる。 $r = 120$ 付近で引力は最大になり、あとは単調減少する。このモデルの利点は、距離が遠いロボットほど影響力が小さくなることにある。それにより遠いロボットを制御から無視できる範囲に収めることが可能になる。また正負が均衡する距離付近において急激に影響力が減衰する。均衡付近までは大きな力で移動し、均衡が近づくと小さな力で少しずつ調整することができる。本稿のシミュレーションでは全てこのモデルを使用している。

3.2 パラメータ学習

この引力斥力に基づく集団行動の制御の問題点は制御用のパラメータ c, σ の決定が難しいことである。ロボットが2つの場合はパラメータの意味は式(5)に示されている。しかし、パラメータが変化なくてもロボットの数の変化により挙動が変化するため、目標に沿ったパラメータを決定することは難しい問題である。さらには全てのロボットのパラメータを異にする場合はより難しい問題となる。目的の集団行動を実現する評価関数を最小

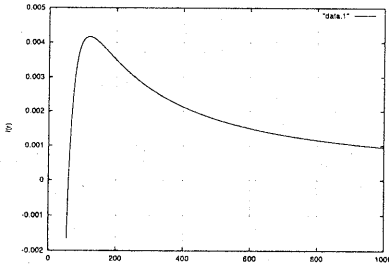


図 3: 2点間の距離と力 ($c_1 = -60.0, \sigma_1 = 2.0, c_2 = 1.0, \sigma_2 = 1.0$)

化する最適なパラメータを以下のアルゴリズムを用いて求める。

ステップ t における評価関数 $f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t)$ と、各パラメータに対応した $f(c_1^t + \Delta c_1, c_2^t, \sigma_1^t, \sigma_2^t)$, $f(c_1^t, c_2^t + \Delta c_2, \sigma_1^t, \sigma_2^t)$, $f(c_1^t, c_2^t, \sigma_1^t + \Delta \sigma_1, \sigma_2^t)$, $f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t + \Delta \sigma_2)$ を用いて各パラメータごとに評価を行い、ステップ $t+1$ のパラメータを定める。以下パラメータごとの学習のステップを示す。

1. c_1 の判定

$$\begin{aligned} f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t) &> f(c_1^t + \Delta c_1, c_2^t, \sigma_1^t, \sigma_2^t) \\ &\Rightarrow c_1^{t+1} = c_1^t + \Delta c_1 \\ f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t) &\leq f(c_1^t + \Delta c_1, c_2^t, \sigma_1^t, \sigma_2^t) \\ &\Rightarrow c_1^{t+1} = c_1^t \\ &\Delta c_1 = -\Delta c_1 \end{aligned}$$

2. c_2 の判定

$$\begin{aligned} f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t) &> f(c_1^t, c_2^t + \Delta c_2, \sigma_1^t, \sigma_2^t) \\ &\Rightarrow c_2^{t+1} = c_2^t + \Delta c_2 \\ f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t) &\leq f(c_1^t, c_2^t + \Delta c_2, \sigma_1^t, \sigma_2^t) \\ &\Rightarrow c_2^{t+1} = c_2^t \\ &\Delta c_2 = -\Delta c_2 \end{aligned}$$

3. σ_1 の判定

$$f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t) > f(c_1^t, c_2^t, \sigma_1^t + \Delta \sigma_1, \sigma_2^t)$$

$$\Rightarrow \sigma_1^{t+1} = \sigma_1^t + \Delta \sigma_1$$

$$f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t) \leq f(c_1^t, c_2^t, \sigma_1^t + \Delta \sigma_1, \sigma_2^t)$$

$$\Rightarrow \sigma_1^{t+1} = \sigma_1^t$$

$$\Delta \sigma_1 = -\Delta \sigma_1$$

4. σ_2 の判定

$$f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t) > f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t + \Delta \sigma_2)$$

$$\Rightarrow \sigma_2^{t+1} = \sigma_2^t + \Delta \sigma_2$$

$$f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t) \leq f(c_1^t, c_2^t, \sigma_1^t, \sigma_2^t + \Delta \sigma_2)$$

$$\Rightarrow \sigma_2^{t+1} = \sigma_2^t$$

$$\Delta \sigma_2 = -\Delta \sigma_2$$

5. 次のステップ ($t+1$) へ

これを繰り返すことにより評価関数の最適の点を求める。評価関数を導入できるのであれば、この方法を用いて制御パラメータを学習により獲得することが可能になる。

4 集団行動の評価関数

4.1 システムの構成

シミュレーションを行なうシステムをサーバクライアントにより構築する。移動するロボットの機能をもつクライアントプログラムをロボットの台数分並列に走らせることによりロボットの動きのシミュレーションを行う。サーバは要求があれば、フィールド上に存在する物の全ての座標値をクライアントに提供する。またクライアントからの座標の変更要求に応じ、そのクライアントの座標値を変更する。このとき、クライアントの衝突判定も行い、衝突が発生する場合は座標の変更を許可しない。基本的なサーバの役割はこの2つであるが、学習ステップ変更やそれに伴う初期化もサーバが行なう。クライアントはサーバから提供された座標値をもとに力の計算を行い、次に移動する座標をサーバに返す。

4.2 集団形成の評価関数

平面上に置かれた、単一の c, σ の組を持つロボットは一定の間隔をもって分布し、1個の円状の集団を形成する。 c, σ の値を変化させることでロボットの間隔は変化し、集団の大きさ、形成過程も変化する。ロボットを2個の集団に分ける場合、パラメータの引力もしくは斥力の強さに違いをつけた2組のパラメータを用意する。集

団内のロボットと集団外のロボットで違うパラメータを適用することで2つの集団を形成する。3つ以上の集団に分ける場合、違う集団のロボットを別々の対象物としてパラメータを設定するのではなく、この2種類のパラメータを適用することで集団を分割する。

ロボットをA, Bの2種類用意する場合、パラメータはA-A間, A-B間, B-A間, B-B間の4通り×4つのパラメータ($c_1, c_2, \sigma_1, \sigma_2$)の16個必要になる。しかしグループAとBのクラスタ形成条件に差をつけないとすると、A-A間とB-B間, A-B間とB-A間については同じパラメータでよい必要なパラメータは8個となる。さらに各クラスタの形成に必要なA-A間とB-B間のパラメータを学習前の1つのクラスタ形成時のパラメータを用いることにより、学習によって導出が必要なパラメータはA-B間とB-A間で共通して用いる4つのパラメータとなる。

収束状態においてすべてのロボットが自身のクラスタに属している場合が望まれる状態であるため、収束状態におけるクラスタが自身のクラスタに属した場合識別できたものとみなすと、異なるクラスに属しているロボットの数を評価関数に用いる。

評価値はロボットの分布よりロボットのグループごとの重心を求め各ロボットから重心への距離を比較することによりどちらのグループに属しているかを識別し、その結果異なるクラスタに属すると判断されたロボットの数を示す。

4.3 ゴミ捨ての評価関数

平面領域上にランダムに配置されたロボットがゴミを収集する。ロボットはゴミを拾った後、領域中央のゴミ捨て場に運ぶ。ゴミ捨て場でロボットはゴミを捨て、その後再びゴミを拾いに行く。このシミュレーションでは、ロボットがゴミを持っている時と持っていない時の2つの状態がある。この2つの状態でパラメータの値を変化させることにより、ロボットの行動の制御を行う。領域上にはロボットとゴミ、ゴミ捨て場の3種類のオブジェクトがあり、ロボットはそれぞれに対してパラメータを持つ。ゴミを持っているロボットはゴミに対して興味を持たず、ゴミを持たないロボットはゴミ捨て場に興味を持たない。ロボットによる具体的なゴミ拾いを実現するため、2つのルールを適用する。

- ・ゴミを持っていないロボットがゴミに達した時、ロボットはゴミを持った状態になる。このときロボットのパラメータは変更され、ロボットはゴミに対する興味を失い、ゴミ箱に対し興味を持つ。

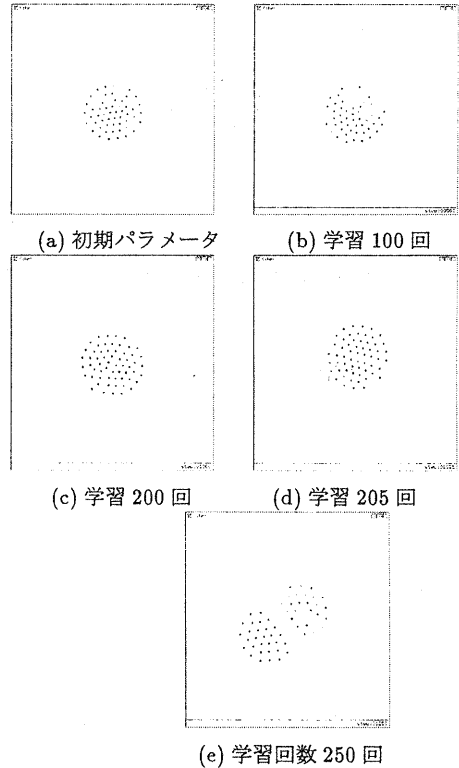


図 4: 2 集団への分離学習過程

- ・ゴミを持ったロボットがゴミ捨て場に達した時、ロボットはゴミをゴミ捨て場に捨て、ゴミを持っていない状態になる。パラメータは変更され、ロボットはゴミに対して興味を持ち、ゴミ箱に対する興味を失う。

他のロボットにぶつかると作業を妨げるため、収束状態に達するまでに他のロボットに衝突する回数を評価値に用いる。

5 集団行動のシミュレーション

5.1 集団形成

ロボットの集団形成では、1つの集団を形成するパラメータから2つの集団に分離するパラメータの導出を行った。初期パラメータとして1集団を形成するものを与え、パラメータの学習により2集団への分離を行った。 $c_1 = -60.0, \sigma_1 = 2.0, c_2 = 1.0, \sigma_2 = 1.0$ を持つ60個のロボットを用意する。これを30個の2つのクラスタに分けるためのパラメータを獲得するシミュレーション

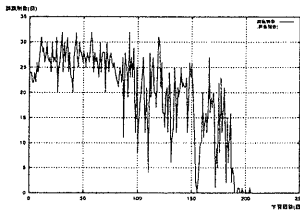


図 5: 学習回数と評価値の変化

を行なった。学習の過程の全てにおいて初期配置はランダムとした。

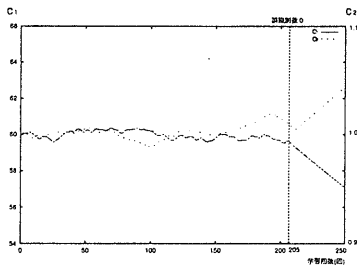


図 6: 学習回数とパラメータの c_1, c_2 の変化

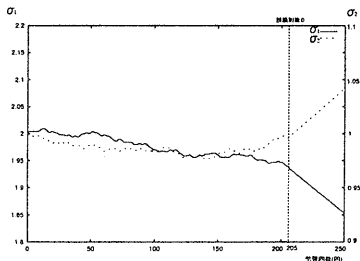


図 7: 学習回数とパラメータ σ_1, σ_2 の変化

図 4(a)(b)(c)(d)(e) は 2 つの集団へ分離が獲得される過程を示している。フィールドの大きさはロボットの大きさを 1 とした場合 600×600 ，黒い四角がグループ A のロボット，白い四角がグループ B のロボットである。全てのロボットが安定する最終状態までシミュレーションを行った後，クラスタリングを行い評価値を求める。

図 4(a) は初期パラメータによる結果であり，1 つの

クラスタを形成する。各ロボットに与えられるパラメータは全て同じであり，クラスタを形成したときの配置はほぼ初期配置に依存する。クラスタ分けは全く行なわれないため，両クラスタの重心位置は全体の集団の重心付近となる。評価値は全ロボット数 60 の半分の 30 前後となる。

図 4(b) は学習回数 100 回の最終状態の分布である。黒いロボットは 1 つのクラスタにまとまり，白いグループも大きなクラスタと分離した 4 つのロボットによるクラスタの 2 つの集団があるだけである。学習によりロボットの分布のクラスタ分けが進んでいる様子がわかる。しかし白いロボットの大きなクラスタを取り囲むように黒いロボットのクラスタが存在しているため，黒いロボット集団の重心位置は白いロボット集団の重心と近くなり，評価関数の値はロボットのクラスタ分けの見かけ以上に悪い。

図 4(c) は学習回数 200 回のロボットによるシミュレーション結果の分布である。クラスタ分けはほぼ終わっており，白グループのロボットが 1 つ孤立しているだけで後は全て黒白ロボットによるクラスタ分離は終わっている。この状態で異なるクラスに識別されるのは孤立している白ロボットだけである。このようにクラスタ分離が拮抗した状態の場合，初期配置によって最終的に分離されるかどうか決定する。そのため，学習回数 200 回では評価値は 0 ではないが，学習回数 200 回以前に評価値が 0 になったときがある。そのため評価関数が 0 になっても，同じパラメータで何度か追試を行なう必要がある。本アルゴリズムでは $f(t)$ と $f(t + \Delta)$ の比較結果が同じならば前回と同じ方向にパラメータを変動させることによって，評価関数が 0 になっても学習を止めないようにしている。

図 4(d) は学習回数 205 回におけるロボットの分布である。黒白のロボットにより 2 つのクラスタが完全に分割されているのがわかる。このときの各パラメータの値は， $s_1 = 1.940777$ ， $s_2 = 1.003950$ ， $c_1 = -60.116693$ ， $c_2 = 1.008986$ であった。初期値が $s_1 = 2.0$ ， $s_2 = 1.0$ ， $c_1 = -60.0$ ， $c_2 = 1.0$ であるから，パラメータの微妙な変化が制御に大きく関わることがわかる。

このシミュレーションでは学習回数 205 回以降において評価関数が常に 0 であった。その後の学習結果をパラメータの妥当性の追試とみなすことにより，205 回時点のパラメータを最終的に得られたパラメータとした。

図 4(e) は学習回数 250 回におけるロボットの分布である。クラスタは 2 つに分かれているため，学習回数 250 回で用いたパラメータも解である。但し学習回数

205回に比べクラスタが互いに距離が大きくなっている。クラスタが離れる方向にパラメータの変更が進んでいるため、学習を繰り返しても評価値が下がっていない。

図5は2つの集団へ分離が獲得される過程における評価値の変化を示している。このグラフは大きく上下に振れており、70回前後において評価関数が傾きだすとそのまま収束に向かっていく。図6にパラメータ c_1, c_2 の変化を、図7にパラメータ σ_1, σ_2 の変化を示す。図5において評価値が0になるのは学習回数205回のときである。その後図6,7のパラメータの値は変化しているにもかかわらず、評価値は0から変化しない。これは学習終了時点である評価値が0で安定する時点でシミュレーションを止めていないためである。評価関数ではクラスタ間の距離を評価にしないで、2つのクラスタに分かれることを評価している。クラスタ間の距離やパラメータは変化しているにもかかわらず、評価値は0のまま変わらず、目的の集団行動は得られている。

5.1.1 獲得されたパラメータによる集団形成

学習回数205回において獲得したパラメータをロボットに与え集団形成のシミュレーションを行なった。初期配置を図8(a)に示す。ロボットは 400×400 の領域にランダムに配置した。この状態から各ロボットは力の計算を行い、それぞれのグループに応じた集団を形成した。その過程を図8(a)-(g)に示す。

図8(a)(b)(c)より、シミュレーション開始直後はランダムに配置されたロボットは中央に集まった。これはシミュレーションで用いた初期パラメータが1つの集団を形成するパラメータを用いているためである。評価値が0に落ち着く最短の学習回数である205回のパラメータは、目的を達するパラメータの中で学習前のパラメータに最も近い値である。そのためにロボットは全体として1つの集団を形成している。その後図8(d)(e)(f)(g)では、全体としての1つの集団を崩すことなく2つのクラスタに分離した。最終的に2つのクラスタに分離できたことから、学習により獲得したパラメータが妥当であることがわかる。

5.2 ゴミ捨て

フィールド上にランダムに配置されたゴミを収集し、ゴミ捨て場にゴミを廃棄するロボットを用いたゴミ捨てのシミュレーションを行なった。

シミュレーションでは、 400×400 のフィールドを用意し、そこにロボットを60体、ランダムに配置した。ゴミは100個用意し、同じくランダムに配置、また

ゴミ箱を4箇所設置した。ロボットに拾われたゴミはロボットの手元に移る。ロボットが持つゴミはゴミ捨て場にロボットが行くことによって廃棄され、完全にシミュレーション上から消滅する。ロボットはゴミを1つしか持てないものとした。

ロボットにはゴミを持った状態とゴミを持たない2つの状態がある。この2つの状態でパラメータを変化させることによりロボットの行動の制御を行なう。 $c_1 \leq 0, c_2 \geq 0$ とした場合、ゴミをもつ、持たないにかかわらずゴミ、ゴミ捨て場に対する斥力は必要ないため、ゴミ、ゴミ捨て場に対しては $c_1 = 0$ である。ゴミを持つロボットはゴミ捨て場に対し引力を持ち、これ以上ゴミを持たないため他のゴミは無視するように対ゴミ捨て場では $c_2 = 100$ 、対ゴミでは $c_2 = 0$ とパラメータを設定した。またゴミを持たないロボットはゴミに対して引力を持ち、ゴミ捨て場を無視するよう、対ゴミでは $c_2 = 10$ 、対ゴミ捨て場では $c_2 = 0$ とした。このようにパラメータを設定し、状況に応じてパラメータの選択を行なうことにより、ロボットはその時必要な物にのみ興味を発生させる。ロボットはゴミを持つ、あるいはゴミを捨てることにより、自らの状態とともに制御用のパラメータも変動させる。ゴミの有無によってロボットの内部状態は異なるが、対ロボットにおけるパラメータの違いはないものとした。

このモデルにおいて、ロボット間の相関が0、つまりロボット間に全く力が働かない状態から学習によってパラメータ導出を行なった。求めるパラメータはロボット間に働く力のパラメータ4つ($c_1, c_2, \sigma_1, \sigma_2$)である。初期値として $c_1 = c_2 = 0, \sigma_1 = \sigma_2 = 1.0$ を与えた。この値では $c_1 c_2$ ともに0のため、ロボット間の相関がない。それぞれのロボットは好き勝手にゴミやゴミ捨て場に向かう。そのため初期状態ではロボット間の衝突が発生する。評価関数にロボットの衝突回数を扱い、その回数を0に収束させることにより、衝突回避をさせながらゴミ捨てを行なうパラメータを導出した。図9にこのシミュレーションにおける評価関数の変化を表す。グラフは大きく上下に揺れながら0に収束している。獲得したパラメータを用いてロボットにゴミ捨ての作業を行わせた結果、初期配置を変えても、60体のロボットは衝突回避をしながらすべてのゴミをゴミ捨て場に捨てる作業を行った。

6 おわりに

集団行動にオブジェクト間の引力と斥力の計算を導入し、ロボット間に加わる力を制御するパラメータを評価

関数の最小化問題を解くことによって獲得した。実際に複数のグループが混じっている集団が、グループごとに分かれる行動と衝突回避を行いながらゴミを集める行動が実現されている。

参考文献

- [1] John j. Craig, "Robots," Addison-Wesley, 1991.
- [2] Uwe R. Zimmer "Robust world-modelling and navigation in a real world," NeuroComputing, Vol. 12, Nos. 2-4, 1996.
- [3] R. Brooks, "A robust layered control system for a mobile robot," IEEE J. Robotics and Automation, vol. RA-2, pp. 14-23, 1986.
- [4] L. Steels, "The artificial life roots of artificial intelligence," Journal of Artificial Life, vol. 1, no. 1/2, pp. 75-110, MIT Press, 1994.
- [5] 福田 敏男, 石原 秀則, "マイクロマシンの自律分散制御", 日本ロボット学会誌 vol. 12, No. 4, pp. 25-30, 1994.
- [6] Ronald C. Arkin, "Integrating behavioral, perceptual, and world knowledge in reactive navigation," Robotics and Autonomous Systems 6, pp. 105-122, 1990.
- [7] T. Balch, "Learning roles: Behavioral diversity in robot teams," AAAI-97 Workshop on Multiagent Learning, 1997.
- [8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," Int. J. Robotics Reserch, pp. 90-99, 1986.
- [9] John H. Reif and Hongyan Wang, "Social potential fields: a distributed behavioral control for autonomous robots," Algorithmic Foundations of Robotics, pp. 331-345, 1995.

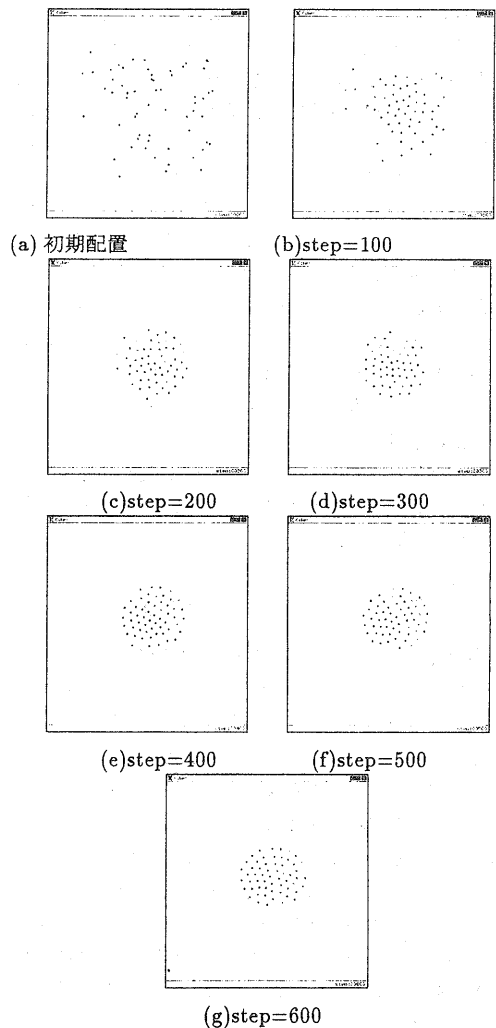


図 8: 獲得されたパラメータによる集団形成

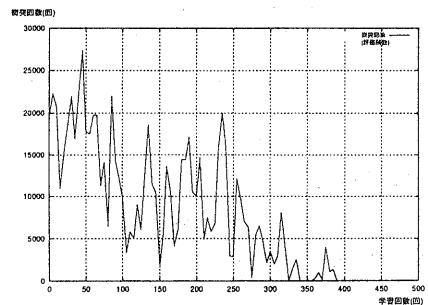


図 9: 学習回数と衝突回数の変化