

RoboCup サッカー2D シミュレーションにおける 遺伝的アルゴリズムを使用したフォーメーション強化

晒健瑠[†] 河本敬子[‡]

近畿大学 生物理工学部 生命情報工学科^{†‡}

1. はじめに

RoboCup2D は、ロボット工学、人工知能の発展のために提唱された国際プロジェクトである。その中でも、2D シミュレーションリーグでは、高さのない 2 次元空間上のフィールドでサッカーの試合が行われる。各エージェント（選手）のステータス（キック力、走力など）には上限があり、どのチームもエージェントの能力に差はない。よって試合に勝利するには戦略が重要となるので、各エージェントが連携してプレイする必要がある。

強化すべき項目は複数存在するが、本研究ではフォーメーションについて注目して強化を行う。フォーメーションの項目としては、normal-formation、offense-formation、defense-formation などが存在する。また、キックインやゴールキック、キックオフなどのセットプレイのフォーメーションも存在する。各ファイルには、1 つのボール座標パターンに対して、各エージェント（11 人分）の x 座標、y 座標の目標座標が実数で設定され、それぞれのパターン毎に存在している。今回は normal-formation に注目して強化を行う。これをハンドコーディングで座標の編集を行おうとすると、重要なフォーメーションファイルにはボール座標パターンが 100 以上存在し、選手 11 人分の x 座標、y 座標を最適化させるのは現実的ではない。そこで、進化計算の中でも代表的な遺伝的アルゴリズム (Genetic Algorithm:GA) の使用を検討する。

2. 遺伝的アルゴリズム

GA では、知識の獲得対象となる情報を、染色体のパターンとして表現する。また、GA を使用することで、試合結果からデータ収集を行えば、自動的に多数のフォーメーションを生成することができるので、最終世代以外にも途中の世代の個体が良い結果を残している場合もあるので、より良いフォーメーションの生成が期待できる。[1]。

2.1 GA の項目について

・遺伝子表現

データの表現を遺伝子型と呼び、今回では、ゴールキーパー以外のエージェント 10 体の x 座標、y 座標の 2 つの実数で表される。また、ゴールキー

パーはあまり移動しないほうがよいと考え、移動させない。よって、遺伝子長はボール座標パターン数とエージェント数 (10 体) の積で表される。

・初期世代生成

強化したいチームのフォーメーションを元に、個体数分のフォーメーションを生成する。

・次世代の生成

個体数と同数の次世代の個体群を複数の方法で生成する。

・エリート保存

優れた個体を次世代に保存することにより世代を重ねても評価が極端に下がってしまうことを防ぎ、更なる最適解が期待できる。

・親候補の選択

個体群から一様な確率で複数の候補を選出する。これは同じ個体ばかりが選ばれてしまうと解が収束してしまうので、それを防止するために実装している。

・親の選択方法

交叉を行うために子の元となる親を 2 つ選択する。このとき、両方同じ親なら交叉にはならないので、重複しないように 2 つの親を選ぶ。

・トーナメント選択法

親候補をトーナメント形式で比較し、評価値が高い方が勝ち上がり、最終的に勝ち残った個体を親にする。メリットは、親候補から最も評価値が高い個体を選ばれるので、より期待できる解を生成する確率が高い。デメリットは、同じ個体ばかり選ばれる確率が高いので、解が収束してしまう場合がある。

・交叉方法

選ばれた親から 2 つの子を生成する。このとき、2 つの子の評価を行うことができないので、両方とも次世代に残す。

・一様交叉

遺伝子全体にマスクをかけ、遺伝子の入れ替えを行うかどうかを判断して交叉を行う。

・新規個体生成

現状の個体のみでは最適化が存在しない可能性がある。その場合、エリート保存、交叉のみでは最適解を生成することができない。なので、評価値の高い個体を元に新しい個体を生成する。

・突然変異

個体群に対し、一定の確率で行う。今回は新規生成で解の収束を防ぐので、解が発散することを

Formation enhancement using genetic algorithm in RoboCup soccer 2D simulation

[†] Takeru Sarashi, [‡] Keiko Kohmoto

^{†,‡} Department of Computational Systems Biology, Faculty of Biology-Oriented Science and Technology, Kindai University.

考え、行っていない。

2.1 評価方法

勝利数、敗北数、引分数、得点数、失点数の項目にそれぞれ重みをつけ、これらの合計を評価値とする。

2.2 その他の項目

- ベンチマークチームの設定
今回は agent2d を使用する。フォーメーションパターンを複数用意する。
- フォーメーションの生成
強化するチームからエージェントの座標データを抽出し、それに対し、乱数で新しいフォーメーションパターンを生成する。このとき、元のフォーメーションにはある程度整った配置となっている。これを一様乱数で移動させるとエージェントの配置が崩れすぎると考えたので、正規乱数を使用する。ここで、正規乱数は-2 から 2 の範囲で大部分の乱数が存在するので、正規乱数 r の範囲を $-2 < r < 2$ と考えることができる。よって、式(1)で新座標を生成する。

$$N = 0 + (r/2) \times m \quad (1)$$

N : 生成された座標, 0 : 元座標, m : 移動範囲

- データ収集
1 個体と 1 ベンチマークチームの試合を 1 試合のみでの試合結果から評価してしまうと、評価値の信頼性が十分とはいえないので、複数回試合を行う。

3. 実験

実際にフォーメーションを生成し、より良い結果が得られるかを実験する。各値の GA に関する項目の設定を表 1 に示す。また、評価値は式(2)を用いる。

$$\text{評価値} = \text{勝利数} \times 5 + \text{引分数} \times 1 + \text{敗北数} \times (-5) + \text{得点数} \times 3 + \text{失点数} \times (-3) \quad (2)$$

実験結果は、各世代の平均、最高値をそれぞれ図 1, 2 に示す。

4 考察

図 1 から、ばらつきがあるが、近似直線から評価値が上昇していることがわかる。よって、GA による次世代の生成は正常に行われていることがわかる。また、このばらつきは、試合数が 30 では足りておらず、データの信頼性が十分ではないからだと考えられる。

図 2 から、第 17 世代の最高値が最も評価が高い個体を作ることができている。よって、この世代の最高値の個体を強化したいチームに採用する。

これらにより、より強いチームを作ることができたことがわかる。

参考文献

[1] 小高知宏, 機械学習と深層学習-C 言語によるシミュレーション-, オーム社, 2017.

強化するチーム	agent2d-3.1.1
ベンチマークチーム	agent2d-3.1.1
強化フォーメーション	normal-formation.conf
X 軸移動範囲	50
Y 軸移動範囲	30
世代数	20
個体数	10
ベンチマークチーム数	3
試合数	30
エリート保存数	2
新規生成数	2
交叉生成数	6
親の候補数	4
親の選択方法	トーナメント選択法
交叉方法	一様交叉

表 1 本実験での設定項目

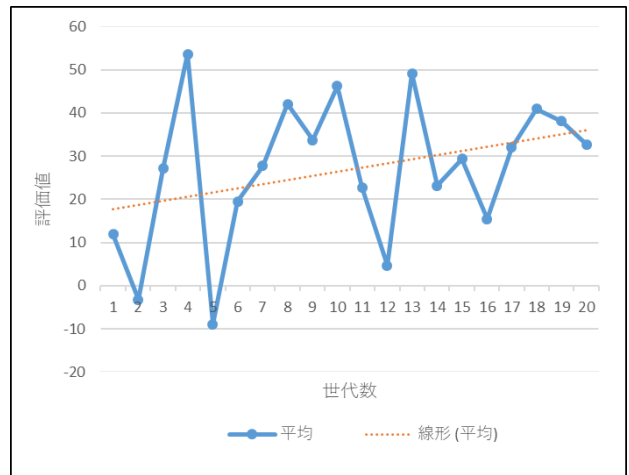


図 1 各世代の評価値の平均値

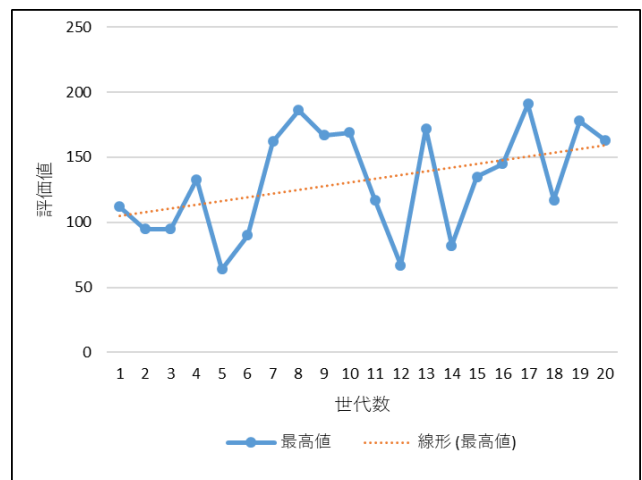


図 2 各世代の評価値の最高値