

RoboCupSoccer における実数値 GA を用いた ボール非保持者の移動方法の学習

水島諒^{†1} 穴田一^{†1}

概要 : RoboCup と呼ばれる自律型ロボットによる世界チャンピオンチームに勝てることを目標にした、サッカーの世界大会が毎年行われている。この RoboCup には 5 つのリーグがあり、リーグごとに異なる特徴がある。本研究ではこの 5 つのリーグの中で、各選手が人工知能を搭載し、それぞれ意思決定を行い、人間のような戦術的なサッカーが行われている 2D リーグを扱う。この 2D リーグは、移動可能範囲が広いことや、リアルタイムで試合が進むため計算時間が制限されていること、11 人同士の対戦であることから、将棋や囲碁より難しいと考えられている。2D リーグで使用されているチームの多くの選手は、戦況に応じた意思決定ができていない。しかし、戦況によって最適な意思決定をする必要がある。そこで、本研究では遺伝的アルゴリズムの一種である実数値 GA を用いて戦況に応じたボール非保持者の意思決定を学習し、その有効性を確認した。

キーワード : RoboCup, 遺伝的アルゴリズム, 実数値 GA, 人工知能

Decision Making of Player Using Real-Coded GA for RoboCupSoccer

RYO MIZUSHIMA^{†1} HAJIME ANADA^{†1}

1. はじめに

近年、「ゲーム AI」の開発が盛んに行われている。例えば、チェスや将棋、囲碁といったゲームが挙げられる。そして、これらのゲームにおいては AI が人間のチャンピオンに勝利するといった事も起きている。また、RoboCup と呼ばれる、自律型ロボットによるサッカーの世界大会が毎年行われている[1]。RoboCup とは、西暦 2050 年迄にサッカーの世界チャンピオンチームに勝てる、自律型ロボットのチームを作ることを目標とした大会である。この RoboCup には 5 つのリーグがあり、リーグごとに異なる特徴がある。本研究では 5 つのリーグのうち、各選手の人工知能がそれぞれ思考し、人間のような戦術的なサッカーが行われている 2D リーグを扱う。2D リーグは、移動可能範囲が広いこと、リアルタイムに計算し判断を下す必要があること、11 人同士の対戦であることなどから、前述のチェスや将棋、囲碁といったゲームより難しいと考えられている。

秋山は RoboCup の 2D リーグで使用可能な agent2d (Ver 3.1.1) というチームモデルを公開している[2]。このチームモデルでは、全ての選手がボールの位置のみを使用し、移動先を決定するという、秋山が開発したフォーメーションシステムを用いている。しかし、このシステムではボールの位置が同じであれば、どのような戦況においても同じポジショニングを目指すという問題がある。

その問題を解決するため、本研究ではボールを保持していない選手の移動方法に、ゲーム木探索[3]を参考に秋山が提案したアルゴリズム (Chain Action モデル) を適用することを考えた。ところが、Chain Action モデルを含め、これまでの研究ではサッカーの専門的な知識を用いて、選手の意思決定のアルゴリズムを経験則で構築することが多い。そのため、意思決定のアルゴリズムを改良する際は人手で調整しなければならないが、人手で調整するには限界がある。

そこで、本研究では遺伝的アルゴリズムの一種である実数値 GA を用いて選手の移動方法を学習させ、その有効性を確認した。

2. ゲーム AI

2.1 RoboCup

RoboCup とはロボット工学と人工知能の融合、発展のため、日本の研究者らによって提唱された自律型ロボットによるサッカーの世界大会である。RoboCup で行われている「ロボカップサッカー」は、『西暦 2050 年迄にサッカーの世界チャンピオンチームに勝てる、自律型ロボットのチームを作る』ということを目指し、ロボット工学や人工知能などの研究を推進し、様々な分野の基礎技術として波及させることを目的としたランドマーク・プロジェクトである[4]。現在は、サッカーだけでなく大規模災害へのロボットの応用を目指した「ロボカップレスキュー」、移動可能な産

^{†1} 東京都市大学
Tokyo City University

業用ロボットを用いて製造現場の最適化を目指した「ロボカップインダストリアル」, 日常生活におけるロボットと人間の共同作業を題材にした「ロボカップ@ホーム」というプロジェクトも存在する。

2.1.1 ロボカップサッカー

ロボカップサッカーは, 人間のサッカーの試合と同じように, 自分で考えて動く自律型ロボットを使った競技会形式で行われる[5]。現在, 自律型ロボット(小型, 中型, 標準プラットフォーム, ヒューマノイド)の4リーグとシミュレーションの1リーグを合わせた計5リーグが存在するプロジェクトである。

● 小型リーグ

1 チーム 6 台の車輪型ロボットが試合を行う。フィールド全体を見渡すカメラを用いて1つのマザーコンピュータが各ロボットに指示を出す。

● 中型リーグ

1 チーム 5 台の車輪型の自律型ロボットが試合を行う。ロボットには全方向を見渡せるカメラを搭載することが多く, ロボットは搭載されたカメラのみで環境を認識し, その情報を用いて行動する。

● 標準プラットフォームリーグ

1 チーム 5 台の人型の自律型ロボットが試合を行う。他の自律型ロボットのリーグとは異なり, 各チームが同じプラットフォームロボットを使用する。そのため, ロボットの性能の優劣はなく人工知能の完成度を競う。

● ヒューマノイドリーグ

1 チーム 3 台の人型サイズの自律型ロボットが試合を行う。各リーグの困難な部分を集めたような最も難易度の高いリーグであり, 2050年の目標へ向かうリーグである。

● シミュレーションリーグ

1 チーム 11 人の人工知能を搭載した選手がコンピュータ上で試合を行う。このリーグは標準プラットフォームリーグをそのままシミュレーションに落とし込んだ 3D リーグと, 高さの概念がない 2D リーグがある。そして, このシミュレーションリーグは外見上市販のサッカーゲームと同じように見えるが, 中身は大きく異なる。市販されているサッカーゲームは, チェスや将棋と同じように, 1 つのマザーコンピュータが全状況を把握できる。そのため, 全状況を把握した上で, 全選手の操作が可能で, フォーメーションやチームワークを持たせることが容易である。しかし, このシミュレーションリーグは, それぞれの選手が独立した人工知能を搭載している。そして, 選手の視野範囲は顔

が向いている方向だけであること, ボールや選手との距離を目視では正確に測ることができないこと, 蹴ったボールは正確に飛ばないこと, 声に相当する通信情報量は少なく遠い選手には届かないことなど, 選手の特徴が人間に近くなっている。そのため, 不完全な情報の獲得, 正確な行動ができない, 仲間の位置が予測できないなどの, 選手が行動する上で難しい問題を多く含んでいるリーグとなっており, 連携したプレーを行うことが難しいリーグとなっている。

2.1.2 ロボカップレスキュー

ロボカップレスキューは, ロボカップサッカーで培われた技術を災害救助に利用しようというプロジェクトである。このプロジェクトには, 地震などの大規模災害時を想定したシミュレーションを用いて, 救助戦略を研究するリーグと, 災害現場で救助に役立つ自律型ロボットの開発を行うリーグがある。

2.1.3 ロボカップインダストリアル

ロボカップインダストリアルは, ロボカップサッカーで培われた技術を物流や倉庫管理システムで活用しようというプロジェクトである。このプロジェクトには, 製品を作るための複数の素材を, 複数の加工機械が協調し, 効率よく運べるかを競うリーグと, 移動型ロボットアームがいかにかうまく部品を検知して回収し, 目的地へ運べるかを競うリーグがある。

2.1.4 ロボカップ@ホーム

ロボカップ@ホームは, ロボカップサッカーで培われた技術を日常生活で活用することを目指し, 人間とロボットが共に暮らしていけるのかという可能性を探るプロジェクトである。キッチンやリビングルームでの利用を想定し, ロボットがいかにか人間と共に作業を遂行できるか, その技術を競技形式で評価する。

2.2 2D リーグ

2D リーグの試合時間は, 前半, 後半 300 秒ずつで1試合 600 秒となっていて, 0.1 秒間隔で離散化されている。そして, 2 次元平面で表わされる仮想のサッカーフィールド上で, 1 チーム 11 人の円形の選手が試合を行う。本研究では 2D リーグで使用されている RoboCup Soccer Simulator (RCSS) というシミュレータを使用する。

2.2.1 RoboCup Soccer Simulator (RCSS)

2D リーグでは, RCSS というシミュレータを利用している。RCSS は図 2.1 のように選手同士は直接通信することができず, rcssserver というシミュレータ本体を介して通信を行う。

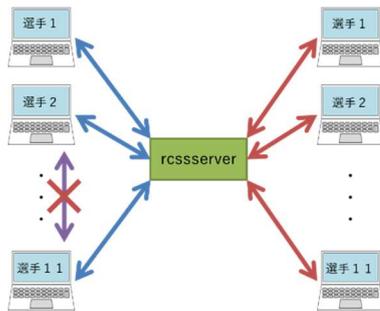


図 2.1. RCSS の通信方法

選手同士は直接通信することができず、rcssserver というシミュレータ本体を介して通信を行う。

RCSS の物理モデルは実世界を一部簡素化してモデル化したものとなっている。そして、RCSS のフィールドの座標系はフィールド中央を原点とし、フィールドの長辺方向を x 軸、短辺方向を y 軸とした直交座標系で、 x 軸は自チームのゴール側が負の値となっている。フィールドの長辺は 105m、短辺は 68m である。

2.2.2 RCSS の特徴

RCSS の選手は、人間と同様の様々な制約を受ける。主な制約として、選手がそれぞれ独自に考えて動く「分散型マルチ選手システム」、選手が正確に情報を把握できない「不完全情報処理」、選手ごとに得意なことや不得意なことがある「選手の能力差」などが挙げられる。以下に 3 つの特徴の詳細な説明を記す。

- 分散型マルチエージェントシステム

人間はそれぞれが得た情報を用いて、独自に考え行動し、互いの考えを共有することはできない。そのため、RCSS は、複数の選手がそれぞれ得た情報を用いて、独自に考え行動するシステムである、分散型マルチ選手システムをとっている。

- 不完全情報処理

人間が視覚などから得られる情報には誤差がある。また、全ての情報を知覚することはできない。例えば、ボールや選手との距離を目視では正確に測ることができないことが挙げられる。また、図 2.2 のように 1 番の選手が知覚した場合、2 番の選手は知覚できるが 2 番の選手の影になっている 4 番の選手や、視野範囲外にいる 3 番の選手を知覚することが出来ないなどが挙げられる。そのため、RCSS の選手はボールや選手までの距離を正確に測ることができないことや、視野範囲外のボールや選手を知覚できないことなどの不完全情報を処理しなければならない。

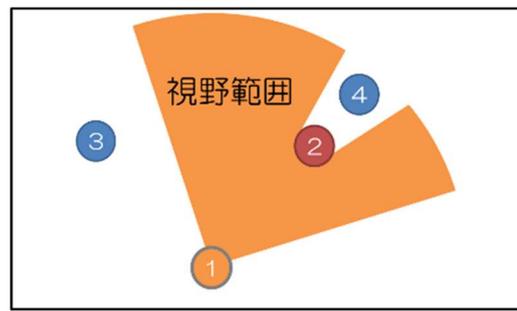


図 2.2. RCSS における不完全情報処理

●, ●, ● は選手を表し、番号は背番号を表す。そして、オレンジ色の扇は 1 番の選手が知覚できる視野範囲を表している。1 番の選手は 2 番の選手を知覚することができるが、2 番の選手の影になっている 4 番の選手や視野範囲外にいる 3 番の選手を知覚することが出来ない。また、2 番の選手との距離の目測も若干誤差がでるようになっている。

- 選手の能力差

人間はそれぞれ得意不得意があり、能力に差がある。そのため、RCSS では選手ごとに移動速度や体の向きを変え、回転速度などの能力が異なるようになっている。

3. 既存研究のチームモデル

秋山は RoboCup の 2D リーグで使用可能な agent2d (Ver 3.1.1) というチームモデルを公開している[2]。agent2d は、選手の移動先の決定を、秋山が提案したフォーメーションシステムを用いて行う。

3.1 フォーメーションシステム

フォーメーションシステムでは、ボールの位置のみを用いて移動先を決定している。このシステムでは、事前にフィールド上の複数の位置に、その位置にボールがあった場合の 11 人の選手の最適位置を、それぞれ設定している。そして、選手の移動先の決定方法は、フォーメーションシステムで設定した位置にボールがある場合と、ない場合で異なる。以下にそれぞれの移動先の決定方法を記す。

- 設定した位置にボールがある場合

選手は、今いる場所からフォーメーションシステムで事前に設定したフォーメーションに従って、指定された目標位置を目指して進む。図 3.1 はある設定した位置にボールがあった時の移動先の決定した例である。

- 設定した位置にボールがない場合

選手は、フォーメーションシステムで事前に設定したボールの位置の集合から、ドロネー図を用いてフィールドを

三角形に分割し、ボールを含む三角形の頂点3点を選ぶ。選んだ3点には、それらの位置にボールがある場合のフォーメーションを事前に設定してある。これら3つのフォーメーションを用いてフォーメーションの補間を行い、今いる場所から補間を行ったフォーメーションに従って、目標位置を目指して進む[5]。図3.2はドローネー図を用いてボールを含む三角形の頂点3点を選び、その3点のフォーメーションとそこから補間して求めたフォーメーションの例である。

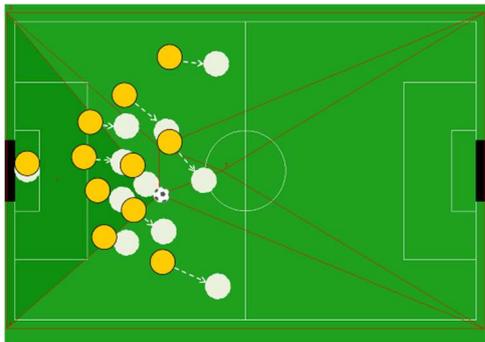


図 3.1. フォーメーションシステムを用いた移動先の例
 自チームのゴールが左で、相手チームのゴールは右である。フィールド上にある ● は選手の今いる場所, ● は選手の移動先の場所, ● はボールを表す。各選手は今いる ● から指定された移動先 ● を目指して移動する。

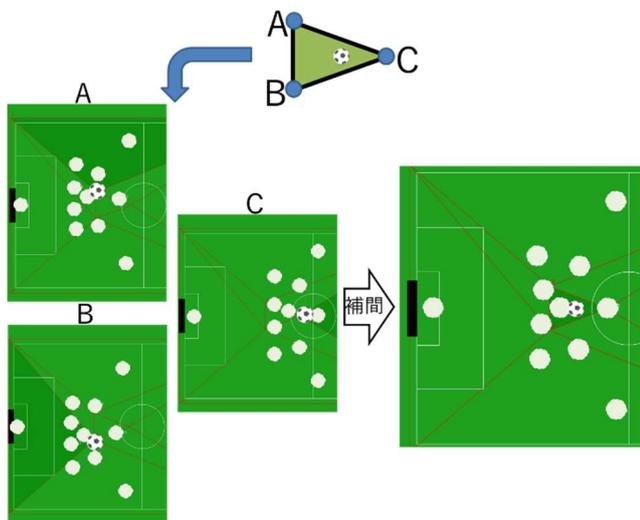


図 3.2. フォーメーションシステムでの補間の例
 ●のボールを含む三角形 ABC からフォーメーションを補間する例である。これらの図は全て自チームのゴールが左であり、左側のフィールド半面を切り取った図となる。フィールド上にある ● は自チームの選手を表す。ボールを含む三角形の頂点 A, B, C で設定してあるフォーメーションを用いて補間を行い、補間したフォーメーションを用いて移動先を決定する。

3.2 既存研究のチームモデルの問題点

ボールの位置のみを用いて移動先を決定しているため、ボールの位置が同じであれば、どのような戦況においても各選手が同じポジショニングを目指して移動するという問題がある。

4. 提案モデル

前述の問題を避けるため、ボールを保持していない選手の移動方法に、ゲーム木探索[3]を参考に秋山が提案したアルゴリズム (Chain Action モデル) を適用する。

本研究では、図 4.1 のように現在の全ての選手の配置を初期状態とし、ボールを保持していないある選手の移動によって、局面がどのように展開するかをツリー構造で表す。そして、その全ての展開を評価し、最も評価が高い展開を選択する。

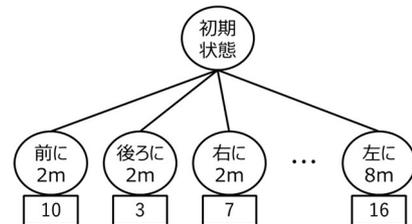


図 4.1. Chain Action モデルで作成したツリー構造の例
 ○は可能な移動候補を示し、中には移動方法を示す。評価値 V の値は□の中に表す。全ての展開の中で最良の展開は V の値が最も高い「左に 8m 移動」という展開である。

4.1 ChainAction モデルの適用方法

本研究で用いる ChainAction モデルの基本的な枠組みを述べる。このモデルでは、移動候補の生成と移動候補の除外処理を行った後に、残った移動候補を評価する。以下にその条件を記す。

- 移動候補の生成
 - I. 周囲 20 方向に 2m 刻みで 2m~10m の候補
 - II. 移動しない
 計 101 個の移動候補を生成
- 移動候補の除外処理
 - I. 移動候補がフィールドのラインから 1m 以内の場合
 - II. オフサイドラインを超えた場合

4.2 移動方法を学習するための評価値 V

移動方法は味方がボールを持っている時 (攻撃時) と敵が持っている時 (守備時) で異なると考えた。そこで、評価値 V の算出方法をそれぞれ用意した。

4.2.1 味方がボールを持っている時（攻撃時）の評価値 V

攻撃時の評価値 V を次のように定義した。

$$V = \sum_{i=1}^7 \omega_i U_i \quad (1)$$

ここで、 ω_i は評価項 U_i の重みを表す。

評価項 U_i には以下のような7つの評価項目を用意した。

$$U_1 = \frac{x_s + 52.5}{105.0} \quad (2)$$

$$U_2 = \frac{\max\{0.0, dist_{thr} - dist_{s,g}\}}{dist_{thr}} \quad (3)$$

$$U_3 = \frac{dist_{l,0n}}{10.0} \quad (4)$$

$$U_4 = \frac{ang_{g1,s,g2}}{180.0} \quad (5)$$

$$U_5 = \min\left\{1.0, 1.0 - \frac{dist_{s,h} - Min_{thr}}{52.5 - Min_{thr}}\right\} \quad (6)$$

$$U_6 = \frac{\min\{dist_{thr2}, dist_{s,t_n}\}}{dist_{thr2}} \quad (7)$$

$$U_7 = \min\left\{1.0, 1.0 - \frac{dist_{s,b} - Min_{thr2}}{52.5 - Min_{thr2}}\right\} \quad (8)$$

ここで、 x_s は移動候補の x 座標、 $dist_{s,g}$ は移動候補とゴールとの距離、 $dist_{l,0n}$ は移動候補とボールとのライン上から一番近い敵との距離、 $ang_{g1,s,g2}$ はゴールとのなす角、 $dist_{s,h}$ は移動候補とフォーメーションシステムによって決定した移動先との距離、 $dist_{s,t_n}$ は移動候補から一番近い敵との距離、 $dist_{thr}, dist_{thr2}, Min_{thr}, Min_{thr2}$ は閾値である。そして、評価項目 U_i はおおよそ 0 以上 1 以下の範囲をとるように設計した。

4.2.2 敵がボールを持っている時（守備時）の評価値 V

守備時の評価値 V を次のように定義した。

$$V = \sum_{i=1}^4 \omega_i U_i \quad (9)$$

ここで、 ω_i は評価項 U_i の重みを表す。

評価項 U_i には以下のような4つの評価項目を用意した。

$$U_1 = \frac{\max\{0.0, dist_{thr3} - dist_{s,g}\}}{dist_{thr3}} \quad (10)$$

$$U_2 = \min\left\{1.0, 1.0 - \frac{dist_{s,b} - Min_{thr3}}{52.5 - Min_{thr3}}\right\} \quad (11)$$

$$U_3 = \frac{\min\{dist_{thr4}, dist_{s,t_n}\}}{dist_{thr4}} \quad (12)$$

$$U_4 = \min\left\{1.0, 1.0 - \frac{dist_{s,b} - Min_{thr4}}{52.5 - Min_{thr4}}\right\} \quad (13)$$

ここで、 $dist_{s,g}$ は移動候補とゴールとの距離、 $dist_{s,b}$ は移動候補とボールとの距離、 $dist_{s,t_n}$ は移動候補から一番近い

敵との距離、 $dist_{thr3}, dist_{thr4}, Min_{thr3}, Min_{thr4}$ は閾値である。そして先ほどと同様に、評価項目 U_i はおおよそ 0 以上 1 以下の範囲をとるように設計した。

4.2.3 評価値 V で設定すべきパラメータ

味方がボールを持っている時（攻撃時）に設定するパラメータは、(1) 式より重みは 7 個、(2)~(8)式より閾値は 4 個である。(9) 式より重みは 4 個、(10)~(13)式より閾値は 4 個であるため、総パラメータ数は 19 個である。

また、選手の移動方法はポジションごとに異なると考えたため、ポジションごとにパラメータを用意した。今回は 6 つのポジションに分けたため、総パラメータ数は 114 個である。この 114 個のパラメータそれぞれを遺伝子とし、遺伝的アルゴリズムを用いてパラメータを進化させる。

4.3 遺伝的アルゴリズムの適用方法

本研究では用意した 4 チームに勝てるように学習を行った。チームの評価は用意した 4 チームとそれぞれ 20 回ずつ対戦させ、次式で表される適合度 G を用いて行う。

$$G = \frac{\sum_{i=1}^4 \sum_{j=1}^{20} (P_{ij} + D_{ij}/1000)}{4 \times 20} \quad (14)$$

ここで、 P_{ij} はチーム i の j 回目の勝ち点であり、勝利なら 3、引分なら 1、敗北なら 0 を表す。 D_{ij} はその試合の得失点差を表す。そのため、チームの評価は 80 試合の勝ち点の平均を用い、勝ち点の平均が同じ際は得失点差を考慮する。遺伝的アルゴリズムの流れは以下の通りである。

I. 初期世代

114 個のパラメータをランダムに決定したチームを 12 チーム作成する。用意した 4 チームと対戦し(14)式を用いて適合度 G を求める。

II. 交叉

ランキング選択[6]によって 2 チームを選択することを繰り返し 6 組作成する。そして、1 組ごとに 2 チームで BLX- α [7]を用いて交叉を行い、新たに 2 チームを作成し、計 12 チームを作成する。

III. 突然変異

新たに作成したチームのパラメータ全てに 5% の確率で、指定されたパラメータ範囲内のランダムな値に変更する。

IV. 適合度計算

残存したチームと新たに作成したチームの計 24 チームを、それぞれ用意した 4 チームと対戦し(14)式を用いて適合度 G を求める。

V. 選択

エリート戦略[6]に基づき現在の世代における上位半分の 12 チームを次の世代に残す。

VI. 終了条件

II~V を指定した世代まで繰り返す。

4.4 メモリーの導入

2D リーグは、選手ごとに能力が異なることや、ボールや選手との距離を目視では正確に測ることができないこと、蹴ったボールは正確に飛ばないことなどの影響があり、試合結果がばらつきやすい。また、1 試合あたりの計算時間もかかる。

そこで、メモリーという過去 5 世代分の適合度を記憶できるスペースを持たせる。そして、対戦して求めた適合度とメモリーに記憶してある適合度を用いて平均適合度を求め、交叉選択の際に使用する。それにより、適合度の大きな変化を抑えた。また、5 世代分の適合度をメモリーに記憶したチームは新たに対戦せずに、メモリーに記憶した適合度から平均適合度を求める。これにより計算時間を減少させた。

5. モデルの評価

5.1 実験設定

本研究では、遺伝的アルゴリズムの適合度計算の際に使用する 4 チームを表 5.1 に記す。

表 5.1 適合度に用いる 4 チーム

チーム名	説明
Fifty-Storms	公開されている最新の大会『RoboCup2011 日本大会』にて動作するチーム
Puppets	
HELIOS-FU	
agent2d	既存研究のチームモデル(Ver.3.1.1)

初期に作成するチーム数は 12、毎世代新たに作成するチーム数は 12、1 チームあたりのパラメータ数は 114、1 世代あたり上記の 4 チームとそれぞれ 20 回対戦し、終了世代は 50 である。各パラメータの範囲と刻み幅は表 5.2 に記す。

表 5.2 各パラメータの範囲と刻み幅

パラメータ名	範囲	刻み幅
ω_i	[0.00 , 1.00]	0.05
$dist_{thr}$	[0 , 100]	5
Min_{thr}	[0.000 , 52.500]	2.625

5.2 実験結果

表 5.3 に各世代における最良チーム(1 位)の適合度を示す。適合度は世代数が増すと概ね高くなっていることが分かる。またメモリーの効果を見るため、0 世代目(初期世代)で最良チームとされたチームの順位変化を表 5.4 に示す。

表 5.3 各世代における最良チームの適合度

	0 世代	1 世代	2 世代	3 世代	4 世代	5 世代
適合度	0.572	0.547	0.379	0.410	0.597	0.797

表 5.4 0 世代目の最良チームの推移

	0 世代	1 世代	2 世代	3 世代	4 世代	5 世代
適合度	0.572	0.522	0.043	0.131	0.471	
平均適合度	0.572	0.547	0.379	0.317	0.348	0.348
順位	1	1	1	3	8	13

表 5.4 から 0 世代目に計算した適合度がたまたま高かったことが分かる。そのため、メモリーを導入し、平均適合度を求めたことにより、適合度の大きな変化を抑えることが出来ていることが分かる。

次に、メモリーの導入による計算時間の短縮効果を見るため、5 世代目における残存したチームのメモリーに記憶してある適合度の数を確認し、表 5.5 に示す。

表 5.5 5 世代目における各チームのメモリーの記憶数

順位	メモリーの記憶数	順位	メモリーの記憶数
1 位	1	7 位	2
2 位	1	8 位	1
3 位	2	9 位	2
4 位	2	10 位	1
5 位	2	11 位	2
6 位	2	12 位	2

この表から、適合度を求めるため、5 世代分のメモリーのみを使用しているチームはなかったことが分かる。5 世代目の最良チーム(1 位)は、対戦して求めた適合度がたまたま高かったため、1 位となった可能性が高い。

発表時に終了世代まで繰り返した結果と考察を述べる。

参考文献

- [1] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa and Hitoshi Matsubara, "RoboCup: A Challenge Problem for AI", AI Magazine, Vol.18, No.1, 1997, pp.73-85.
- [2] 秋山 英久, "アクション連鎖探索によるオンライン戦術プランニング", 人工知能学会研究会資料, SIG-Challenge-B101-6, pp.23-28 (2011).
- [3] 新谷虎松, 大園忠親, 白松俊: "知識システムの実相基礎-スライドで理解する人工知能技術-", コロナ社, (2012).
- [4] "ロボカップ日本委員会".
<http://www.robocup.or.jp/original/about.html>, (参照 2016-10-31).
- [5] 大島真樹:"Java でつくる RoboCup サッカー選手プログラム", 森北出版株式会社 (2005).
- [6] 佐藤 浩, 小野 功, 小林 重信, "遺伝的アルゴリズムにおける世代交代モデルの提案と評価", 人工知能学会誌, Vol.12, No.5, pp.734-743 (1997).
- [7] Eshelman, L.J, "Real-Coded Genetic Algorithm and Interval Schema ta", Foundations of Genetic Algorithm 2, pp.187-202, 1993.