RoboCup サッカーシミュレーションリーグ 2D における 局面評価関数の学習

谷川俊策[†1]　五十嵐治一[†1]　石原聖司[†2]

RoboCup サッカーシミュレーションリーグ 2D はコンピュータ上の仮想フィールド内でエージェント同士がサッカーの試合を行う競技であり，マルチエージェントシステムの研究用テストベットとしても利用されている．本研究では，プレイヤエージェントがボールを保持した際の行動決定に，ゲーム木探索により行動後の局面を予測し，その局面での評価関数を用いる方策を採用する．今回はこの局面評価関数をサッカーに関する人間のヒューリスティクスを用いて構築し，その中のパラメータを強化学習の一手法である方策勾配法により学習する方法を提案する．

A Learning Algorithm of State Evaluation Functions in RoboCup Soccer Simulation League 2D

SHUNSAKU TANIGAWA[†1]　HARUKAZU IGARASHI[†1]
SEIJI ISHIHARA[†2]

RoboCup Soccer Simulation League 2D is a game competition where independently moving software players (agents) play soccer on a virtual field inside a computer. This league is used as a testbed in research on multiagent systems. This paper deals with a policy of an agent when he holds a ball. In Agent2D, the policy uses a game tree whose nodes are predicted states caused by agents' actions and a positional evaluation function that evaluates the predicted states. We designed this positional evaluation using heuristics on soccer and applied the policy gradient reinforcement learning algorithm to learning weight parameters included in the evaluation function.

1. はじめに

RoboCup[1)]サッカーシミュレーションリーグ 2D は，コンピュータ上の仮想フィールド内でエージェント同士がサッカーの試合を行う競技部門 [2)]であり，マルチエージェントシステムの研究用テストベットとしても利用されている．特に，不完全な情報しか得られない自律エージェントが，周囲の状況が刻々と変化する中で，実時間で行動を決定する必要がある．さらに，状態と行動も連続空間であることが，状態数や行動数，あるいはその組み合わせ数の大幅な増大を引き起こすことも問題を難しくしている．

本研究は，このようなマルチエージェントシステムにおいて，プレイヤエージェント間の協調行動を強化学習によって実現させるための一般的な手法の確立を目指している．今回は，ボールを保持したプレイヤエージェントが次に行うべき行動を決定する際に，複数の味方プレイヤの協力を考慮することに研究対象を限定した．この行動決定問題に対しては，最近ではチェスや将棋で採用されているゲーム探索木や局面評価関数が用いられ始めている [8)]．本研究もこの方針を取り，サッカーシミュレーションリーグ 2D 向けのゲーム木探索法と局面評価関数の学習法とを提案する．

2. サッカーシミュレーションと agent2d

2.1 サッカーシミュレーションリーグ 2D

本リーグのシミュレータは，計算機上に仮想的なサッカーフィールドを用意して，ボールやプレイヤの物理的な運動（2 次元）をシミュレートする．このシミュレータはサーバ・クライアント方式を採用しており，サーバはクライアント（プレイヤエージェント）へ視覚・聴覚情報の送信，物理計算，状態更新などの作業を行う．プレイヤエージェントはこの知覚情報を元に状況を判断し，行動を決定し，行動コマンドをサーバへ送信する．各プレイヤエージェントのプログラムは完全に独立しており，エージェント間の通信もサーバを介する必要がある．また，プレイヤエージェントの他に，2 種類のコーチエージェントが用意されており，フィールド上の物体の完全情報を利用したいときにはよく用いられている．本リーグのシミュレータについては文献 3)が，最近の世界大会の動向については文献 4)の解説記事や本年度(2013)の世界大会におけるチーム紹介文書 [5)]が詳しい．なお，本リーグに関する研究発表については文献 6)に一覧表としてまとめられている．

2.2 agent2d について

本リーグでは，いくつかのチームプログラムのソースコードが公開されている．これらをベースプログラムとして利用するチームも多い．その中で，最近よく使用されているチームプログラムの一つが agent2d[7)]である．agent2d は 2010 年の RoboCup で優勝した HELIOS というチームの簡易版プログラムである．この agent2d(v3.1.0)ではボールを保持してキック可能な場合，いくつかの行動に候補を絞り込んだ後で，局面をノード，行動を枝とする探索木（chain action[8)]）を生成し，末端ノードの局面のうち評価値の最大なノードを実現する行動を選択している（先読み探索）．しかし，行動実行後の遷移局面では，ボールとレシーバ以外のプレイヤは静止していると仮定し，局面評価関数はボー

†1 芝浦工業大学工学部情報工学科
Shibaura Institute of Technology
†2 東京電機大学
Tokyo Denki University

ルの位置だけを評価しているにすぎない．

そこで，この chain action の改良もいくつか試みられている．例えば，agent2d の作者自身による相手プレイヤの位置予測をニューラルネットワークで行う研究[9]や，Mello らの評価関数中の重み係数を PSO (Particle Swarm Optimization) で最適化する試み[10]などがある．

この chain action を用いた行動決定方式は，チェスや将棋などのゲームで用いられているゲーム木探索と局面評価関数とを用いた着手決定方式の手法を，サッカーというマルチエージェントシステムにおけるエージェントの行動決定の問題へ持ち込んだと解釈することもできる．しかし，サッカーの場合は，22 人のプレイヤが独立して行動することから行動数や状態数が膨大となり，かつ，状態遷移における不確実性が大きいことから，探索木の生成自体が難しく，局面評価関数の設計も自明ではない．また，チェスや将棋で用いられている min-max 戦略を単純に適用することにも検討の余地があり，探索木のどのノードの評価値をどのように用いて行動を決定するかということも新たに考える必要がある．本研究では，この chain action を用いた行動決定方式において，探索木の利用法と局面評価関数の設計・学習について新しい提案を行う．

3. Chain action を用いた行動決定

2.で述べたように，agent2d ではボールを保持したプレイヤは，chain action と呼ばれる探索木を用いて行動決定を行う．図 1 に例を示す．この探索木のルートノードからはボール保持者の行動（パス，シュート，ドリブル）に対応する枝が出ている．これらの行動を a_i (i=1,2,…,N)とする．探索木のノードは行動実行後に遷移した状態を表している．ルートノード以外からも枝が出ているが，それは元のボール保持者の行動とは限らず，他のプレイヤの行動を表す場合もある．例えば，図 1 において，ルートノードでのボール保持者 O の行動 a が他の味方プレイヤ A へのパスであれば，ノード s_a はそれを A がレシーブした状態を表し，さらに s_a から出ている枝は A が別の味方プレイヤ B へパスを出す行動（パスの連鎖）などを表している．つまり，枝はボール保持者自身または味方プレイヤの行動を表し，ノードはその行動実行後に生じる予測状態である．

現行の agent2d においては，最良優先探索を行い，探索木中のノード(leaf とは限らない)の評価値が最大であるノードへつながる行動を決定論的に選択している．しかし，本研究では局面評価関数の学習用に次のような確率的な方策を考える．まず，探索の深さを固定して，leaf だけではなく探索木中の全ノードの評価値を計算する．出現したルート局面 s における行動 a の評価関数 $E_a(a,s;\omega)$を，その行動 a から派生した部分木の全ノードにおける最大評価値 $E_s(s_a;\omega)$で定義する．s_a はこのときの最大評価値を与えるノード局面（末端ノートとは限らない）である．この行動評価関数 $E_a(a,s;\omega)$を目的関数とする確率的方策を次のボルツマン分布で定義する．

$$\pi(a \mid s;\omega) \equiv \frac{e^{E_a(a,s;\omega)/T}}{\sum_x e^{E_a(x,s;\omega)/T}} = \frac{e^{E_s(s_a;\omega)/T}}{\sum_x e^{E_s(s_x;\omega)/T}} \quad (1)$$

図 1 は現在局面 s からの行動を枝とする探索木の例を示しており，数字はその局面の局面評価関数 $E_s(s;\omega)$の値を表している．この例では，行動 a, b, c の行動評価関数 $E_a(a,s;\omega)$ の値はそれぞれ，80, 30, 100 となるが，それらの値は，s_a, s_b, s_c のノードを局面評価関数 E_s により評価した値である．

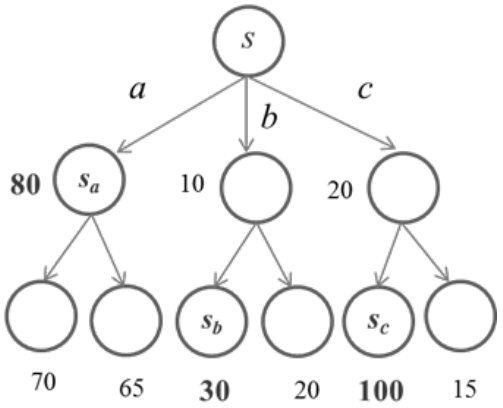

図 1　探索木と局面評価関数の値の例

Figure 1 An example of a search tree and node values given by a positional evaluation function.

4. 局面評価関数の設計

局面 s が自チームとりどれだけ優勢であるかを評価する局面評価関数 $E_s(s;\omega)$を次のように設計した．

$$E_s(s;\omega) = U_0(s)\left[\omega_1 U_1(s) + \omega_2 U_2(s)\right] + \sum_{i=3}^{5} \omega_i U_i(s) \quad (2)$$

ここで，U_i(s) (i = 0,1,⋯,5)は局面 s を評価する際に有効だと思われる 6 つの先見的知識を表した関数であり，ω_i (i = 1,2, ⋯,5) は重み係数である．(2)の局面評価関数 $E_s(s;\omega)$は，正／負で絶対値が大きいほど自チームが優勢／劣勢な局面を表すように定義する．今回設計した(2)の各項の意味を簡単に述べると，$U_0 \in \{-1,0,1\}$は，ボールを所有しているプレイヤであるパサーが所属しているチームを表す．U_1 はパサーの安全度，U_2 はパサーから見て敵ゴール側にいる敵味方の人数比，U_3 はボールの位置，U_4 と U_5 はボールまでのプレイヤの最短距離や分布を考慮した局面特徴量である．また，U_1 と U_2 は[0,10]に，U_3 ～U_5 は[-10,10]の区間内の値を取るように正規化した．これら各項の定義を付録に記した．

図 2 に agent2d 同士の試合中に出現したある局面における(2)の評価値の一例を記す． ただし，重み係数 ω は適当な値に設定してある．図 2 は，この局面を各プレイヤが (2)の局面評価関数を用いて評価値を計算し，それぞれのプレイヤの上に表示させた図である．ボールから遠いプレイヤほどボール周りの観測情報が曖昧で，コーチの評価値と大きく異なる評価値を出す傾向があることがわかる．

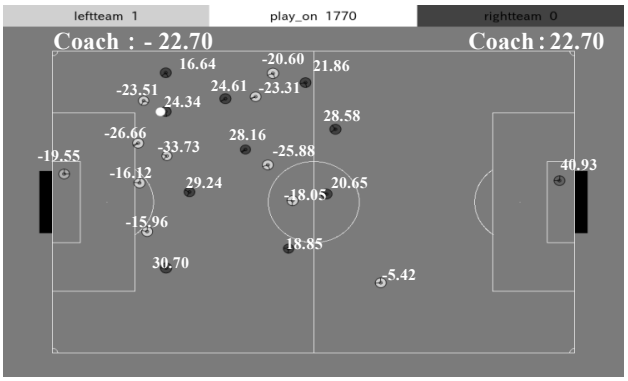

図 2　各プレイヤが算出した局面評価関数値の例
Figure 2　Players' positional evaluation values calculated by the positional evaluation function.

5. 方策勾配法による局面評価関数の学習

エピソードを定義し，エピソード終了後にエピソード全体を評価して報酬 r を与える．この報酬の期待値を最大にすることを考える．(1)の方策中のパラメータ ω の学習則は，強化学習の一種である方策勾配法[11),12)]によると，

$$\Delta\omega = \varepsilon \cdot r \sum_{t=1}^{L} e_{\omega}(t) \tag{3}$$

$$e_{\omega}(t) \equiv \partial \ln \pi\left(a(t)\middle|s(t);\omega\right) \Big/ \partial\omega \tag{4}$$

と表される．ただし，$s(t)$は時刻 t における局面，$a(t)$は選択された行動，L はエピソード長，ε は学習係数である．(4)に(1)を代入すると $e_{\omega}(t)$は次のように表される．

$$e_{\omega}(t) = (1/T)\left[\partial E_s\left(s_{a(t)};\omega\right) \Big/ \partial\omega - \sum_b \pi\left(b\middle|s(t);\omega\right)\partial E_s\left(s_b;\omega\right)/\partial\omega\right] \tag{5}$$

6. 実験

6.1 状態 s と行動 a

各プレイヤが得た敵味方合わせて 22 人のプレイヤとボールの位置情報を状態 s と定義する．また，agent2d では，パス，ドリブル，シュート，ホールド（ボールの保持）の行動に対応する関数として，Bhv_PasskickFindReceiver(), Bhv_NormalDribble(), Body_ForceShoot(), Body_Hold()が用意されている．今回はこの 4 種を行動 a として用いた．

6.2 エピソードと報酬 r

自チームのプレイヤがボールを保持してから，敵プレイヤに捕られるか，ファールなどで審判によってプレーが中断されるまでの間を 1 エピソードと定義した．また，報酬 r は次のように定義し，エピソード終了時に与えた．

$$r = \mathrm{sgn}(\Delta l)(\Delta l / C)^2 + goal\_point \tag{6}$$

ボールと敵ゴール中心との距離を l とする．Δl はエピソード開始時と終了時における l の変化量である．C と $goal\_point$ は定数であり，$goal\_point$ はエピソード中に自チームが得点したときに与える．sgn( )は符号関数である．次節の学習実験では，C=5[m]，$goal\_poin$=200 と設定した．

6.3 学習実験

学習エージェントは，agent2d において MF の役割を持つ 3 人と FW の役割を持つ 3 人の合計 6 人とした．ただし，重み ω は学習中には 6 人ともに同一の値を取るように設定した．温度 T=10，学習率 ε=0.001，重み ω_i の初期値はすべて 5 という条件で， オリジナルの agent2d(ver3.1.1) と 500 試合対戦させる学習実験を行った．結果を以下に示す．

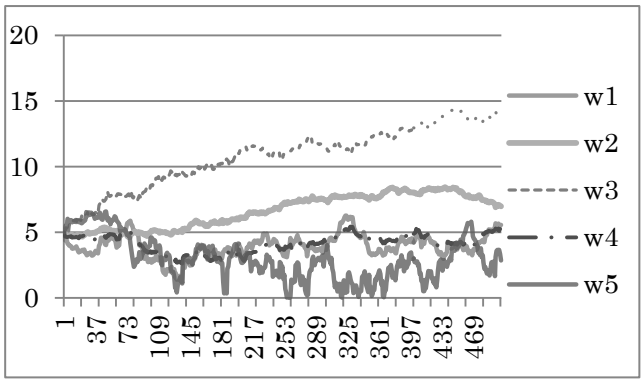

図 3　重みの推移
Figure 3　Change of weight parameters.

図 3 からボールの位置を評価する関数 U_3 の重み ω_3 の値が最大となった．これは，報酬が敵ゴールに近い程多く得られることからも妥当と考えられる．ω_1 と ω_4 の推移はよく似ている．今回のようにパサーのみが学習する場合は，U_1 と U_4 はともにパサーの安全度を評価する関数となるので重みも類似した推移になったと考えられる．ω_2 と ω_5 は，両者ともパサー以外のプレイヤの位置を評価する重みである．ω_2 は攻めに加担する味方が多いことを評価する関数の重みであり，この評価が高いほど自チームに有利な前方へのボール運びが可能となるので，今回の報酬の下では ω_2 の値は強化された．また，U_5 は，ボール周辺での敵味方プレイヤ数の差という安全性の評価関数だが，高い報酬を獲得する可能性が高い敵ゴール前付近では低い値を取りやすい．この結果，高報酬を得るためには ω_5 を小さくしてこの項を抑え気味にする必要があったと解釈できる．以上により，ω_2 と ω_3 の値が大きく強化され，1 試合ごとの平均エピソード報酬値が増加して行ったと考えられる（図 4）．

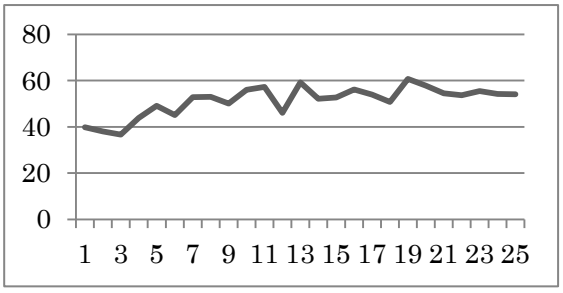

図 4　平均エピソード報酬値の推移
（20 試合ごとの区間平均値）．
Figure 4　Reward per episode (averaged over 20 games).

7. おわりに

本研究では，マルチエージェントシステムの協調行動の学習の一例として，RoboCup サッカーシミュレーションリーグ 2D におけるオープンソースプログラム agent2D の chain action の探索木の利用法と局面評価関数の設計・学習について提案を行った．学習実験を行なった結果，獲得報酬が増えて学習の有効性を確認することができた. 今後は，役割やプレイヤごとに局面評価関数を学習して行きたい．

参考文献

1) RoboCup のホームページ， http://www.robocup.org/.
2) RoboCup サッカーシミュレーションリーグ 2D のホームページ，http://wiki.robocup.org/wiki/Soccer_Simulation_League
3) 秋山英久 : RoboCup サッカー2D シミュレーションリーグ解説：仕組みと環境構築，知能と情報，Vol. 23, No. 5, pp. 714-720 (2011).
4) 秋山英久，中島智晴：RoboCup サッカーシミュレーションリーグ解説 : 2011 年世界大会レポートと今後の展望，知能と情報，Vol. 24, No. 1, pp. 14-18 (2012).
5) Team Description Papers: RoboCup Soccer 2D, http://staff.science.uva.nl/~arnoud/activities/robocup/RoboCup2013/Symposium/TeamDescriptionPapers/SoccerSimulation/index.html
6) Stone, P. : http://www.cs.utexas.edu/~pstone/tmp/sim-league-research.pdf
7) 秋山英久 : RoboCup サッカー2D シミュレーションリーグ解説：サンプルエージェントを使ったチーム開発，知能と情報，Vol. 23, No. 6, pp. 838-844 (2011).
8) 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).
9) Akiyama, H., Nakashima, T., and Yamashita, K. : HELIOS2013 Team Description Paper, http://staff.science.uva.nl/~arnoud/activities/robocup/RoboCup2013/Symposium/TeamDescriptionPapers/SoccerSimulation/Soccer2D/TDP_HELIOS2013.pdf
10) Mello, F., Ramos, L., Maximo, M., Ferreira, R., and Moura, V., http://www.socsim.robocup.org/files/2D/tdp/RoboCup2012/TDP_ITAndroids.pdf
11) Williams, R. J. : Simple Statistical Gradient- Following Algorithms for Connectionist Reinforcement Learning, Machine Learning, Vol.8, pp.229-256 (1992).
12) 五十嵐治一，石原聖司，木村昌臣：非マルコフ決定過程における強化学習―特徴的適正度の統計的性質―，電子情報通信学会論文誌 D, Vol.J90-D, No.9, pp.2271-2280 (2007).

付録：局面評価関数の各項の定義

(i) U_0：*passer* の所属チームの判定

ボール保持者である *passer* の所属チームを表す. *passer* が味方プレイヤであれば 1 を，敵プレイヤであれば-1 を，どちらが持っているかわからない場合を 0 とする．

(ii) U_1：*passer* の安全性を評価する関数

$$U_1 \equiv 10 \Big/ \left(1 + e^{-(p\_safty-7)}\right) \tag{7}$$

p-safty は *passer* とそれに最近接の敵プレイヤとの距離である．この距離が大きい程，高い評価となる．

(iii) U_2：ボール前方の敵味方の人数差を評価する関数

$$U_2 \equiv 10 \Big/ \left(1 + e^{-(l_2 - \frac{4}{9})}\right) \quad \left[l_2 \equiv \frac{mate\_cnt + 1}{opp\_cnt + 1} \right] \tag{8}$$

passer が攻める方向を前方としたとき，ボールの前方にいる味方の人数を *mate_cnt*，敵の人数を *opp_cnt* で表している．また，*passer* が敵ゴール付近にいる場合は、ペナルティエリア内にいる味方の人数を *mate_cnt*，敵の人数を *opp_cnt* で表している．ボール前方にいる味方が多い程，また，敵が少ない程，高い評価となる．

(iv) U_3：ボールの位置を評価する関数

$$U_3 \equiv 10 \Big/ \left(1 + e^{dist\_mateGoal}\right) - 10 \Big/ \left(1 + e^{dist\_oppGoal}\right) \tag{9}$$

dist_mateGoal はボールと自ゴールとの距離を，*dist_oppGoal* はボールと敵ゴールとの距離を表している．敵ゴールにボールが近い程，高い評価となる．

(v) U_4：プレイヤのボールまでの距離を評価する関数

$$U_4 \equiv 10 \Big/ \left(1 + e^{dist\_mate - dist\_opp}\right) \tag{10}$$

両チームにおいて，ボールに最も近いプレイヤとボールとの距離の差を評価する関数である．dist_mate は自チームの中でのボールに最も近いプレイヤとボールとの距離を，dist_opp は敵チームの中でボールに最も近いプレイヤとボールとの距離を表している．自チームのプレイヤがボールに近ければ近い程，敵プレイヤがボールから遠ければ遠い程，高い評価となる．

(vi) U_5：各チームのプレイヤのボールへの距離分布

$$U_5 \equiv 10 \Big/ \left(1 + e^{point\_m - point\_p}\right) \tag{11}$$

ボールから一定の範囲内にいるプレイヤの数の差を評価する関数である．*point_m* は自チームが得た評価値，*point_o* は敵チームが得た評価値を表している．ただし，評価値の定義は図 5 に示す. この例では, ○印のチームの評価値は, 1+2+0=3 点, △印のチームの評価値は, 2+2+2=6 点となる，U_5 はプレイヤのボールへの密集度に関する味方と敵の違いを評価している．

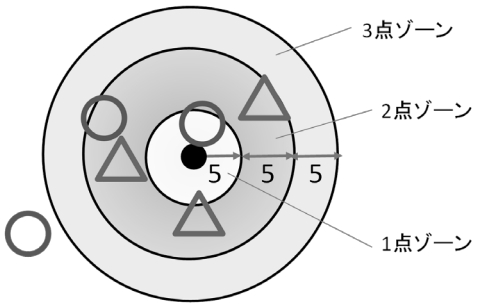

図 5　U_5 で用いた評価値の計算例．各プレイヤが位置するゾーンの点数を合計する．

Figure 5　Calculation of the evaluation points in U_5. Sum zone points of the team mates around a ball.