

# RoboCup サッカーにおける枝刈りを用いた行動制御

## Controlling Actions with Pruning in RoboCup Soccer

田中 翔<sup>†</sup>, 中島 智晴<sup>†</sup>, 秋山 英久<sup>‡</sup>

Sho Tanaka<sup>†</sup>, Tomoharu Nakashima<sup>†</sup>, Hidehisa Akiyama<sup>‡</sup>

大阪府立大学<sup>†</sup>, 福岡大学<sup>‡</sup>

Osaka prefecture University<sup>†</sup>, Fukuoka University<sup>‡</sup>

{sho.tanaka@cs, tomoharu.nakashima@kis}.osakafu-u.ac.jp, akym@fukuoka-u.ac.jp

### Abstract

This paper proposes a team strategy implementation method by directly controlling its behavior. The team behavior depends mainly on the decisions of the ball owner who selects the best action in an action space. Unfortunately, the action space is generally too large to explore in a reasonably short time. Furthermore, many solutions might be irrelevant to the strategy decided by the developer and can be ignored. In order to reduce the search space and make the ball owner focus only on relevant actions, a binary classifier is used. Actions classified as negative by the classifier are pruned from the search space. Computational experiments are conducted in order to show that the reduced search space produces tendencies in the team behavior that correspond to the strategy of interest.

### 1 はじめに

ロボット工学と人工知能の領域横断型研究プロジェクトとして RoboCup [1] が知られている。RoboCup には様々なリーグが存在しており、それぞれにおいて活発な研究、開発が行われている。本論文では、RoboCup のリーグの一つである RoboCup サッカーシミュレーション 2D リーグを研究対象とする。RoboCup サッカーでは、競技大会で勝利することを目的としている。ランダムな思いつきで作られた戦略を使って勝利するよりも、緻密なデータ分析や機械学習によるモデル化を活用して勝利に繋がる戦術を生成することが RoboCup の理念と合致する。

RoboCup サッカーシミュレーション 2D リーグでは、毎年各チームが変化を加えることにより、フィールドの片側

に選手を固め攻撃する戦術や守備を偏重する戦術など、多種多様な戦術が存在している。戦術の多様性と相性のため、1 つの戦術で全てのチームに勝つことは困難である。そのため、複数の戦術を用意し、相手に合わせて戦術を変更することで試合を有利に進めることが求められる。複数の戦術を用意するには、どのような方策で行動するかを決定し、それに合わせてプレイヤーの行動を制御する必要がある。本研究では、プレイヤーの行動制御に着目する。

プレイヤーの行動制御に関して、探索木を用いたプレイヤーの意思決定法が提案されている [2]。探索木で用いられる評価関数では、敵ゴールからの距離や敵との距離といったサッカー特有の特徴量の重み付き総和により、状態を評価する。そのため、評価関数における特徴量の重みを変更することでプレイヤーの行動を制御出来る。しかし、評価関数の重みは人手により調整されているため、戦略や戦術を表現する評価関数を新たに試作するだけでも、高い時間的コストが必要である。

プレイヤーの行動を制御する他の方法として、探索木生成段階で不要な行動列を直接制限することが考えられる。本論文では、この方針により探索木生成のプロセスを変更することで、評価関数を調整することなく、設計者の意図通りに戦術を実装する方法を提案する。

提案手法の概要を説明する。探索木生成段階において、行動が設計者の意図を反映しているかどうかを判断し、枝刈りを行う。枝刈り判定のため、サポート・ベクタ・マシン (SVM) により識別モデルを構築する。識別モデルの構築は教師あり学習に基づくため、モデルの構築には学習用データが必要である。学習用データを作成するため、まず試合を行い試合ログを作成する。作成した試合ログからプレイヤーの行った行動列を抽出しクラスタリングする。作成したクラスタを GUI アプリケーションにより人手でラベル付けすることで、学習用データを作成する。

数値実験では、10-分割交差検証により SVM の識別精度を調査する。また、学習した枝刈りによる行動制御への

有効性を示すため、実際に枝刈りを組み込んだチームで試合を行いキックの軌跡と性能を調査する。

## 2 RoboCup

RoboCup は、ロボット工学と人工知能の発展が目的の自律移動型ロボットによるサッカーやレスキュー、家庭内作業などを題材とした研究プロジェクトである。RoboCup には「西暦 2050 年までに、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チームを作る」という目標があり、この目標に向けて盛んに研究が行われている。

RoboCup にはサッカー以外にも、大規模災害への対応のシミュレーションや災害現場で活躍するロボットの開発を促進するレスキューリーグ、日常生活で人間を支援する自律ロボットによる競技を通じて、人とコミュニケーションしながら役に立つロボットの実現を目指す@ホームリーグの他に、次世代のロボット技術者育成を目的としているジュニアリーグも存在する。本論文では、RoboCup サッカーシミュレーションリーグを研究の対象とする。シミュレーションリーグは RoboCup 創設当初から存在する最も古いリーグの 1 つである。サッカーシミュレーションでは、実機を使用せずに、コンピュータ内に用意された仮想フィールド上でサッカー競技を行う。サッカーシミュレーションには 2D リーグと 3D リーグがある。Figure 1, 2 に 2D リーグと 3D リーグの試合の様子を示す。2D リーグでは、基本的な動作（キックやドリブルなど）はコマンドとして実装されている。そのため 2D リーグでは高レベルな意思決定を主な研究対象としている。一方、3D リーグでは、エージェントはヒューマノイドロボットで形成されているため、基本的な動作を関節から制御する必要があり、基本的な動作が非常に重要である。本論文では 2D リーグを扱う。2D リーグでは、二次元平面を仮想サッカーフィールドとし、円形のエージェントをプレイヤーとして競技を行う。また、プレイヤーやボールの位置と速度は全て二次元のベクトルとして表される。試合は前後半 3000 サイクルずつ合計 6000 サイクルからなる。1 サイクルは 0.1 秒で離散化されている。各プレイヤーはそれぞれ独立したエージェントとしてプログラムされており、制限された視覚情報や聴覚情報からドリブルやパス等の行動選択を行う。

## 3 関連研究

RoboCup サッカーシミュレーション 2D リーグにおいて、プレイヤーの行動制御に関する様々な研究が行われている。例えばプレイヤーレベルでの制御として、Gabel ら [3] は、強化学習を用いた Neuro Hassel により、ディフェンス時における積極的なボールのインターセプト動作獲得を行っている。しかし、各プレイヤーでの行動が洗練される

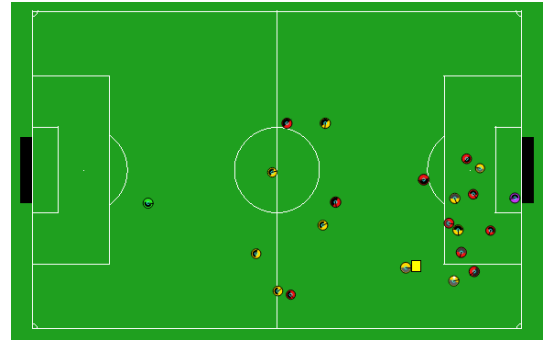


図 1: 2D Simulation League



図 2: 3D Simulation League

ことにより、チーム力は向上するが戦術レベルの行動制御に至ることは難しい。

チームレベルでの行動制御に関する研究として、Stone ら [4] は、特定の条件下において実行すべき戦術を事前に用意し共有することで、戦術レベルの行動をプレイヤーに知らせる Locker Room Agreement を提案している。Locker Room Agreement により、限定的な状況で戦術レベルの行動制御が可能であるが、戦術が固定的であるため、柔軟性に乏しいという問題がある。一方で、状況に応じて柔軟に対応し戦術レベルでプレイヤーの行動制御を行う方法も提案されている。Akiyama ら [5] は、Delaunay Triangulation を用いたボール位置に基づく直感的にわかりやすいプレイヤーのポジショニング手法を提案している。Akiyama ら [2] は、行動連鎖と呼ばれる探索木を生成し探索と評価により行動決定を行う木探索によるオンラインでの行動プランニングを提案している。また、Akiyama ら [6] は、上記 2 つの手法が実装された agent2d をオープンソースとして公開している。agent2d を基本としてプレイヤー開発を行っているチームは多く、行動制御に関する研究にもよく使われている。特に、チームの戦術を表現するよう行動制御を行うため、行動プランニング時に用いられる評価関数に関する研究が多く行われている。

評価関数に関する研究として、例えば三船ら [7] は、Single Input Rule Modules ファジィシステムを用いて評価

関数を自動作成する手法を提案している。田川ら [8] は、強化学習の 1 つである方策勾配法と人間の投票システムにより評価関数で用いる重みを学習する手法を提案している。牛腸ら [9] は、ランク SVM を用いたランキング学習と人間の教示による評価関数の作成手法を提案している。しかし、自動作成された評価関数が競技会に用いられることは少ない。このことから、人手により作成した評価関数のほうが優れており、自動作成された評価関数は未だ実践的なレベルではないと考えられる。そのため、複数の戦術を評価関数の自動作成により用意することは意思決定の質を落とす要因になる。そこで、評価関数を変更することなく、複数の戦術を用意する必要がある。

## 4 行動プランニングと枝刈りによる行動制御

本論文で使用するチームは、探索木により行動列を生成する。プレイヤーの行動を制御するため、行動プランニングにおける枝刈りを提案する。本章では、まず行動プランニングについて説明し、次に枝刈りによる行動制御について説明する。

### 4.1 行動プランニング

本論文では、行動意思決定のモデルとして秋山ら [2] の提案した木探索による協調行動プランニングを用いる。このモデルでは、ボールキック時において探索木を生成し探索することにより、行動プランを作成している。行動プランを次にプレイヤーが行うべき一定数の長さを持つ行動列と定義する。

本論文で使用するチームでは、探索木の走査アルゴリズムとして、最良優先探索を用いる。以下の手順により行動プランを作成する。まず、ルートノードに現在状態を格納する。そして、ルートノードから行動の候補を生成する。このとき、プレイヤーが観測した現在状態や予測状態を入力とし、自分と他者を含めた複数のエージェントによって実行される行動（パスやドリブル、シュートなど）を生成する。この時、実現可能な行動が計算し、不可能だと判断した行動は削除されるため、現実性のある行動のみ生成される。生成された行動を評価関数により評価値を計算し、行動と状態、評価値を探索木へ子ノードとして格納する。すべてのノードが追加された後、評価値が最大のノードを選択し、そのノードにおける予測状態からさらに行動の候補を生成する。これを繰り返すことで、探索ノード数があらかじめ設定された最大値に達するまで探索木を成長させる。ただし、木の深さがあらかじめ設定した値を越える場合や、ノードの予測状態から行動が生成できない場合、そして行動列の終了条件に設定されている行動（シュート）が生成された場合は、その葉ノードでの子ノード生成は行わないものとする。構築された木構造の中からノード列をつなげると、行動列が得られる。

探索終了後、生成された行動列の中で評価値が最大のものを行動プランとする。行動プランニングにより、プレイヤーは数手先の状況を考慮し、より戦術的価値が高い行動を選択することが可能となる。

### 4.2 枝刈りによる行動制御

行動プランニングにおいて、行動の意思決定は評価関数に強く依存する。新たに戦略や戦術を反映した行動選択のためには、それに対応した評価関数を作成する必要がある。評価関数は、ゴールからの距離、ペナルティエリアへの侵入、サイド攻撃といったサッカー特有の特徴量の重み付き総和により評価値を算出している。しかし、この重みは人手による調整の繰り返しによって定められているため、有用な評価関数を作成には高い時間的コストや開発の経験が必要であり困難である。評価関数を修正せずに行動制御を行う他の方法として、探索木生成段階で行動を制限することが考えられる。そこで、木探索時に枝刈りを行い、探索の選択と集中を行う手法を考える。本論文では、SVM により枝刈りの判断を行う。本節では、まず位置座標の離散化について説明し、SVM を用いた枝刈りについて説明する。

#### 4.2.1 位置情報の離散化

本論文では、戦術を行動列によって表現している。そこで、行動の遷移を明確にするため、 $x$  と  $y$  の 2 変数の連続値により表現される位置座標を離散化する。離散化のため、フィールドを  $n \times m$  個のグリッドに分割する。グリッド状に分割したフィールドにおいて、位置座標に対応したグリッドのみを 1、それ以外では 0 の値をとる  $n \times m$  次元ベクトルにより位置座標を表現する。つまり、仮に  $k$  番目のマス目内でキックした際、その位置座標を離散化して表現する場合、 $n \times m$  次元ベクトルにおいて  $k$  番目の要素のみ 1 の値をとり、残りの要素は 0 の値をとる、 $n \times m$  次元ベクトルとなる。

#### 4.2.2 SVM を用いた枝刈り

探索木生成段階において、設計者の意図と反する行動を枝刈りする。SVM をラベル付けした行動列により学習させる。ラベル付けの方法は 5 章で説明する。枝刈りの結果、設計者の意図した戦術を表現する行動列だけが残り、SVM への入力には、離散化したキック座標による行動列を用いる。行動列の長さは一定ではないため、可能なすべての長さに対する SVM を構築し、枝刈りを判断する。

## 5 GUI を用いた学習用データへのラベル付け

SVM は教師あり学習であり、モデルを構築するための学習用データが必要である。本研究では、学習用パターンを試合ログから生成する。まず、多数の試合を行い試合ログを作成する。次に、作成した試合ログからプレイヤー

が実行した行動列を抽出し、その行動列から長さ 4 の連続した部分行動列を生成する。これは、行動プランニングで生成される探索木の深さが最大 4 であるためである。そして、抽出した行動列にラベル付けを行い学習用データとする。学習用データへのラベル付けにおいて、製作者の意図した戦術を表現する行動列かどうかを完全に自動で判断することは難しい。そのため、既存研究において人間の教示による学習用データの作成が行われてきた。例えば、田川らの研究では人間による投票システムを用いた行動に対する報酬を決定するシステム [8] を作成している。また、牛腸らは GUI アプリケーションを用いて行動にランク付けを行うシステム [9] を作成している。そこで、本研究においても、GUI アプリケーションを用いることにより、人間による直観的なラベル付けを行う。しかし、いずれの研究においても人間による選択に必要な時間的コストが問題となっている。そこで、本研究ではクラスタリングにより似た行動列をまとめることにより、選択に必要なコストを削減する。クラスタリングには、ガウス混合分布と EM アルゴリズムを用いる。また、最適クラス数数を決定する基準として、ベイズ情報量規準を用いる。作成した GUI アプリケーションを Figure 3 に示す。GUI アプリケーションでは、Figure 3 のように、ウィンドウ上に一つのクラスに纏められた行動列が表示される。そして、下部に設置されたボタンから、設計者の意図した戦術を表現する行動列には 1、それ以外には -1 をラベル付けする。

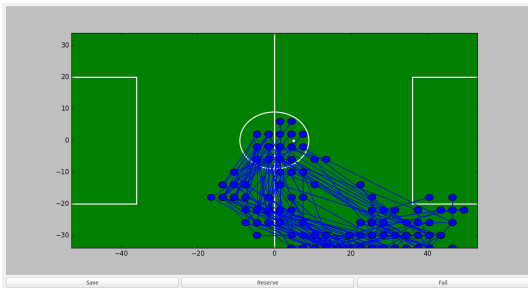


図 3: Screenshot of the developed GUI

## 6 数値実験

フィールドの片側サイドから攻撃するような学習用データを使って SVM を学習させ、識別精度を調査する。次に、枝刈りによる行動制御の有効性を示すため、枝刈りを組み込んだ HELIOS [10] で試合を行い、枝刈りを組み込んでいない従来の HELIOS と比較することで、行動の変化、チーム性能の変化について調査する。HELIOS は、本研究室と福岡大学により共同開発を行っている世界大会参加用のチームである。本実験において、SVM に入力する素性ベクトルは、4.2.1 項の離散化処理を行ったキック位置とする。また、行動プランニングにおいて、探索する

最大の深さを 4、最大探索ノード数を 1000 とする。連続するキックの中で、次にキックするプレイヤーが、キックしたプレイヤーと異なる場合をパス、同一プレイヤーが続けてキックを行いプレイヤーの位置座標が変化したものをドリブルと定義する。ドリブルは、キックによる位置座標の変化量が少なく、戦術的な意味合いが低いと考えられる。したがって、本論文の実験ではパスのみを枝刈りの対象とする。行動生成時の枝刈りにおいて、仮に [パス, パス, ドリブル, パス] といった長さ 4 の行動列が生成された場合、[パス, パス, パス] といったドリブルを排除した長さ 3 の行動列を識別対象とする。

### 6.1 SVM の学習と識別

HELIOS と Shiraz, FURY, CSU\_Yunlu, InfoGraphics, Ziziphus, HERMES, Ri-one, agent2d の 8 チームとそれぞれ 100 試合を行い試合ログを作成する。agent2d [6] はベースチームとして公開されているチームであり、InfoGraphics は RoboCup2014 に参加したチーム、HERMES は RoboCup2015 に参加したチーム、Shiraz, FURY, CSU\_Yunlu, Ziziphus, Ri-one は RoboCup2016 に参加したチームである。5 章で説明した操作により学習用データを作成する作成した学習用データを用いて、SVM を学習させる。作成した学習用データを用いて、10-分割交差検証を行う。グリッド数による識別精度への影響を調査するため、フィールドを  $7 \times 7$  個のグリッドに分割した際の識別結果を Table 1、フィールドを  $23 \times 26$  個のグリッドに分割した際の識別結果を Table 2 に示す。また、この時使用したカーネルは線形カーネルと RBF カーネルである。Table 1, 2 から、カーネルで比較した際、線形カーネルの識別精度が良いことがわかる。また、グリッド数で比較した際、 $23 \times 26$  個のグリッドに分割した際に識別結果が良いことがわかる。

RoboCup サッカーシミュレーション 2D リーグにおいて、意思決定は 1 サイクル (0.1 秒) の間に行わなければならない。そのため、提案手法を組み込んだ SVM を HELIOS に組み込み、agent2d と 10 試合行い学習した SVM の実行速度を調査した。SVM の識別一回当たりの平均実行速度は、線形カーネルでは 0.0024(msec)、RBF カーネルでは 4.7474(msec) であった。このことから、実行速度に関しても線形カーネルの方が有効であると言える。

表 1: SVM's accuracy rates for a  $7 \times 7$  grid

Kernel	Depth : 1	Depth : 2	Depth : 3	Depth : 4
Linear	80.31	81.98	83.65	84.60
RBF	67.93	67.93	67.93	67.93

表 2: SVM's accuracy rates for a  $23 \times 26$  grid

Kernel	Depth : 1	Depth : 2	Depth : 3	Depth : 4
Linear	83.44	85.08	86.58	87.45
RBF	67.93	67.93	67.93	67.93

## 6.2 枝刈りによる行動制御

提案手法では、行動プランニング時の探索において、生成された行動列が設計者の意図に適しているかどうかSVMにより判断し、枝刈りを行う。6.1節の実験により得られた知見から、識別モデルとして線形カーネルを用いたSVMを使用する。また、フィールドを $23 \times 26$ 個のグリッドに分割してキック位置を離散化する。HELIOSに提案手法を組み込み、Shiraz, FURY, CSU\_Yunlu, InfoGraphics, Ziziphus, HERMES, Ri-one, agent2dの8チームとそれぞれ50試合ずつ行う。行動の変化を表す例として、対FURY戦における枝刈りを組み込んでいない従来のチームでのパスの軌跡と提案手法を組み込んだチームでのパスの軌跡をFigure 4に示す。また、従来のチームでの試合結果をTable 3, 枝刈りを組み込んだチームでの試合結果をTable 4に示す。

Figure 4において、上側の図が従来のチームでのパスの軌跡を表しており、下側の図が枝刈りを組み込んだチームでのパスの軌跡を表している。また、左側が自陣、右側が敵陣である。赤い線がパスの軌跡を表しており、線の色の濃さにより頻度を表している。Figure 4から、枝刈りを組み込んだチームは、従来のチームと比較して片側のサイドにパスを出す傾向が強くなっていることがわかる。Table 3, 4において、Win rateは勝率、Goals forは平均得点、Goals againstは平均失点、Through passは平均スルーパス回数、Shootは平均シュート回数、Penalty areaは敵ペナルティエリアへの平均侵入回数を表している。この時、スルーパスを、敵のオフサイドラインより自陣側からキックを行い、次にキックする位置が一つ前の敵のオフサイドラインを超えており、なおかつフィールドを3分割した際の敵陣側でキックしたものと定義する。Table 3, 4から提案手法を組み込んだチームでは、従来のチームと比較した際、得点においてCSU\_Yunlu, Ziziphusでは同等程度、それ以外では減少している。得点が減少した原因として、スルーパスの回数が減少や、学習用データの作成において誤ったラベル付けをした可能性や誤識別の可能性などが考えられる。InfoGraphicsとの試合では、スルーパスの回数が著しく減少している。これは、InfoGraphicsの守備が、InfoGraphics側に向けてパスをさせないよう一列に並ぶという特徴に起因すると考える。従来は、これに対しサイドチェンジを繰り返し行い、スルーパスを行う機会を作り、突破していた。しかし、枝刈りを組み込んだチームでは、片方のサイドに行動を寄せているため、スルーパスを行う機会が減り、その結果としてシュート回数や得点が減少したことが考えられる。一方で、Ziziphusでは、ゴール前にプレイヤーを集め、守備を行うという特徴を持つ。そのため、片方のサイドから攻撃した場合でも、同様に得点に繋がる行動を行ったと考えられる。しかし、勝率で見ると、HERMES以外ではあまり変化してい

ない。このことから、勝利が可能な行動選択ができていたと言える。HERMESに対し勝利数が減少した原因として、HERMESは守備に偏重した陣形をとっており、得点が難しかったことが考えられる。

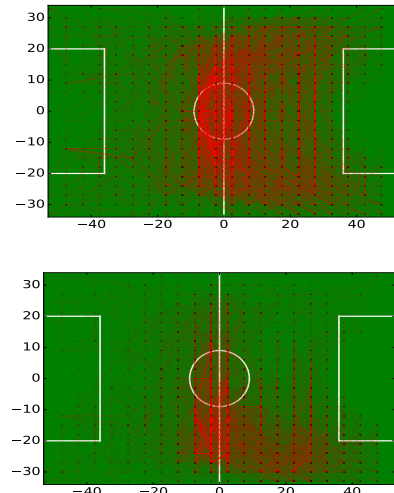


図 4: Pass distribution against FURY

## 7 おわりに

本論文では、評価関数の修正なしに行動制御をするため、探索木生成段階で枝刈りすることにより行動列を制御する手法を提案した。戦術を複数用意することが出来れば、より有利に試合を進めることが出来る。また、設計者の意図を反映した学習用データを作成するため、GUIアプリケーションによりラベル付けを行った。今後の課題として、ユーザビリティを考えたGUIアプリケーションの作成や、識別精度を高めるため識別モデルや入力に用いる特徴量の検討などが考えられる。

## 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, pp. 73-85, 1997.
- [2] 秋山 英久, "連続状態行動空間での木探索によるオンライン協調行動プランニング", 研究報告ゲーム情報学, 2012-GI-27 巻, 11号, 2012.
- [3] Thomas Gabel, Martin Riedmiller, and Florian Trost, "A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach", *RoboCup 2007: Robot Soccer World Cup XII*, pp. 61-72, 2008.
- [4] Peter Stone and Manuela Veloso, "Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic

表 3: Performance without the proposed method

Opponent	Win rate	Goals for	Goals against	Through pass	Shoot	Penalty area
Shiraz	0.94	4.1	0.21	20.04	6.1	11.27
FURY	0.78	2.63	0.1	8.67	3.69	6.18
CSU_Yunlu	0.88	2.49	0.06	11.63	4.24	7.47
InfoGraphics	0.96	3.55	0.0	11.35	4.96	9.06
Ziziphus	0.9	1.7	0.0	25.44	3.92	13.88
HERMES	0.92	1.88	0.14	11.44	3.22	6.64
Ri-one	0.84	2.06	0.1	10.44	2.72	5.66
agent2d	0.98	9.55	0.12	19.0	11.49	17.39

表 4: Performance with the proposed method

Opponent	Win rate	Goals for	Goals against	Through pass	Shoot	Penalty area
Shiraz	0.86	3.85	0.3	21.17	6.21	11.34
FURY	0.76	2.02	0.06	8.18	3.06	5.69
CSU_Yunlu	0.86	2.36	0.0	9.42	3.58	6.34
InfoGraphics	0.86	2.2	0.0	6.78	2.78	5.33
Ziziphus	0.96	1.74	0.0	24.18	3.48	12.58
HERMES	0.54	1.16	0.22	9.72	2.4	5.72
Ri-one	0.74	1.48	0.04	9.12	2.12	4.5
agent2d	0.98	8.2	0.14	17.37	10.69	16.27

Teamwork”, *Artificial Intelligence*, Vol. 110, No. 2, pp. 241-273, 1999.

*Germany*, 6 pages, 2015.

- [5] Hidehisa Akiyama and Itsuki Noda, “Multi-Agent Positioning Mechanism in the Dynamic Environment”, *RoboCup 2007 : Robot Soccer World Cup XI*, 2008.
- [6] Hidehisa Akiyama and Tomoharu Nakashima, “HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation”, *RoboCup 2013 : Robot World Cup XVII*, pp. 528-535, 2014.
- [7] 三船 哲史, 中島 智晴, 秋山 英久, 関 宏理, “RoboCup サッカーにおける SIRM’s ファジィシステムを用いたログからの行動評価”, 第 40 回人工知能学会 AI チャレンジ研究会論文集, pp. 20-25, 2014 .
- [8] 田川 諒, 五十嵐 治一, “サッカーエージェントにおける局面評価関数の強化学習”, 第 20 回ゲームプログラミングワークショップ 2015 論文集, pp. 78-83, 2015 .
- [9] 牛腸 俊寛, 秋山 英久, 荒牧 重登, “人間の行動選択ログを用いたサッカーエージェントの行動評価関数獲得”, 第 32 回ファジィシステムシンポジウム論文集, pp. 61-64, 2016 .
- [10] Hidehisa Akiyama, Tomoharu Nakashima, Jordan Henrio, Thomas Henn, Sho Tanaka, Tomonari Nakade and Takuya Fukushima, “HELIOS2016 Team Description Paper”, *RoboCup 2016 Leipzig*,