

7人制ラグビー最適攻撃プレー算出法に対する並列計算 Parallel Computing for Optimization Method of Attack Play in Seven-a-side Rugby

八代 航太郎[†] 中田 洋平[†]
Kotaro Yashiro Yohei Nakada

1. はじめに

近年、日本では、急速にラグビーへの注目が集まってきている。そのような中、新たなファンの獲得やファンの固定化のため、試合放送時などに試合展開を分かりやすくする付加情報を提供する重要性が、ますます高まってきている。このような背景の下、著者らは、これまでに7人制ラグビーを対象として、選手位置情報から、ランとハンドパスを含む最適な攻撃プレーを算出する方法を提案してきた[1][2]。さらに、いくつかの実装上の工夫を導入し、その計算処理の高速化を進めてきた[3]。本稿では、その実装に並列計算を導入し、さらなる高速化を図ることを目的とする。そして、4つの仮想的フォーメーション例を用いて、その並列計算の導入の有効性を検証する。

2. 最適攻撃プレー算出法の概要

本節には、高速化の対象となる最適攻撃プレー算出法[1][2]の概要について記す。

2.1 攻撃プレーに対する最適化問題

最適攻撃プレー算出法[1]-[3]では、選手とボールの運動モデルに基づいてランとハンドパスのシミュレーションを行い、その結果に基づいて攻撃プレーの評価値を計算する。ここで、ある有効な攻撃プレー p の評価値 $F(p)$ は次式で表される。

$$F(p) = T(p) + \lambda N(p) \quad (1)$$

ただし、 $T(p)$ は攻撃プレー p での最短トライ時間、 $N(p)$ は攻撃プレー p での期待タックル数を表している。また、パラメータ $\lambda \in [0, \infty)$ は2つの指標のトレードオフを調整するためのパラメータである。本手法では、この評価値(1)を最小化することで、最適攻撃プレーを算出する。

2.2 ランプレーのシミュレーション

本手法におけるランプレーのシミュレーションでは、ボール保持選手は、以下の2式に示す選手運動モデルに基づいて、最適な方向に力を加え動くものとする。

$$dr(t)/dt = v(t) \quad (2)$$

$$dv(t)/dt = aVu - av(t) \quad (3)$$

ここで、ベクトル $r(t)$ と $v(t)$ は、それぞれ選手の位置ベクトルと速度ベクトルを、パラメータ $a [s^{-1}]$ 、 $V [m/s]$ は、それぞれ選手の加速能力、最大速度を表している。また、ベクトル u は加速方向を表す単位ベクトルである。さらに、最適な方向は、ボール保持選手がトライまで同じ方向に力を加え続けるという条件で、評価値(1)を最小化することによって決定される。

2.3 ハンドパスプレーのシミュレーション

本手法では、ルールに抵触する危険性を考慮し、ボール保持選手の右手の方向を $0 [deg]$ とし、ハンドパスの方向角が $[-\epsilon, -(180 - \epsilon)] [deg]$ の範囲内にあるハンドパスをシミュレーションする。パラメータ ϵ は、実行可能なハンドパスを定義するために用いる角度値である。本研究の実装では、ハンドパスの方向角を $\zeta [deg]$ の幅で量子化して実行可能なハンドパスを探索する。なお、ハンドパスされたボールは有効射程距離 $l [m]$ までは等速直線運動をするものとし、それ以降は無効なプレーと定めている。また、各味方選手は、ハンドパスを出した時点のボール保持選手の位置より前方で捕球することはできないとしている。各味方選手の実行可能なパスを探索した後、評価値(1)を用いて、各味方選手について最適なハンドパスを評価する。この評価は、パスボールを捕球した後、各味方選手が最適な角度方向でトライするまで走り続けると仮定して行う。なお、最適な方向角は、ボールを捕球してから時間幅 $\Delta [s]$ ごとに更新している。

2.4 分枝限定法による最適攻撃プレーの算出

本手法では、ランとハンドパスの攻撃プレーの分岐を木構造により表現する。各ノードでは、対応するボール保持選手が、時間幅 $\Delta [s]$ ごとに走り続けるか、または各味方選手にパスをするかを決定する。このような設定の下では、最適な攻撃プレーは、対象となる問題をその部分問題に分割するための分枝操作と、不要な部分問題を終端とするための限定操作を用いた分枝限定法を用いて計算することができる。本実装では、あるノードまでの評価値の下界値が評価値の上界値を超えた場合、限定操作により対応する部分問題を終端とする。ここで、上界値は、トライまでシミュレーションされた実行可能な攻撃プレーの評価値の中で、最も小さいものとしている。また、実行可能なハンドパスを持たない味方選手に対応する部分問題も終端となる。本実装では、後述する各ノードでの目的関数の下界値に基づく最良優先探索法を適用している。

3. 最適攻撃プレー算出の高速化

まず、これまで著者らが取り組んできている以下に記す2つの改良[3]について記す。なお、これら2つの改良についても、後述の検証実験では、処理時間を計測し、比較している。まず、1つ目の改良では、ボール保持選手のランのシミュレーションの方向角を、7人制ラグビーの性質上、全方向から攻撃方向のみに限定している。2つ目の改良では、分枝限定法における下界値の算出方式の改良を実施している。具体的に述べれば、これまで最適攻撃プレー算出法[1][2]では、評価値(1)の時間に対する非減少性から、その時点 t までの評価値(1)を用いていた。しかし、著者ら文献[3]において、時点 t のボール位置とゴールラインまでの

[†] 明治大学 先端数理科学研究科
Graduate School of Advanced Mathematical Sciences,
Meiji University

最短距離を用いて、トライまでにかかる時間の下界値を算出し、それを加えた以下の下界値 $L(\cdot)$ に改良した。

$$L(p, t) = f(p, t) + (x_p(t) - x_G) / V^* \quad (4)$$

ここで、 $f(p, t)$ は攻撃プレー p における時刻 t までの評価値(1)であり、以前用いていた下界値となる。なお、攻撃プレー p での最短トライ時間 $T(p)$ を用いると、 $F(p) = f(p, T(p))$ という関係がある。また、 $x_p(t)$ は攻撃プレー p における時刻 t でのボールの長軸方向の位置を、 x_G は相手チーム側のゴールラインの長軸方向における位置を表しており、 V^* は全味方選手の速度の上界値を表している。なお、この下界値 $L(\cdot)$ は、 $0 \leq t \leq T(p)$ の範囲において、以下の関係を満たし、以前用いていた下界値 $f(p, t)$ よりも、タイトな下界値となっている。

$$F(p) \geq L(p, t) \geq f(p, t) \quad (5)$$

すなわち、この下界値の算出により、最適な攻撃プレーとはならない攻撃プレーの分岐をより早く終端とすることが可能となり、計算時間の削減が可能となる。

本稿では、これら 2 つの改良に加え、更にボール保持選手のハンドパスのシミュレーションの実装に対し、並列計算が可能となるよう改良を施している。また、前述したように方向角について $[-\epsilon, -(180 - \epsilon)]$ [deg] の範囲を ζ [deg] の幅で量子化して実行可能なパスを探索している。更に、各パスボールを捕球した位置から、評価値(1)の意味で、最適な角度方向でトライするまで走り続けると仮定してシミュレーションを行い、各味方選手の最適なハンドパスを評価している。そのため、ハンドパスのシミュレーションには、膨大な計算が必要となる。しかし、各ハンドパスのシミュレーションを並列化することは比較的容易であり、その効果も高いと見込まれる。

4. 仮想的フォーメーション例での検証実験

本稿では、4 つの仮想的なフォーメーション例を用いて、前節で述べた高速化のための改良や並列計算の導入後の条件において、処理時間の比較検証を行う。なお、パラメータの設定値は文献[2]に記載のものと同様としている。図 1 に最適攻撃プレーの算出結果を示す。また、全選手の初速度は 0 としており、青い点が味方選手の到達可能領域[4]の中心位置やボール保持選手の位置、赤い点が敵選手の到達可能領域の中心位置を表している。初期ボール保持選手は、(a)と(b)が青の 1 番下の選手、(c)と(d)が青の下から 3 番目の選手としている。また、黄色い線がボールの軌跡、青色の円と赤色の円は、それぞれ味方選手と敵選手の到達可能領域を表している。

表 2 に、各フォーメーション例での以下の条件下における平均処理時間の比較結果を示す。条件(i)がランのシミュレーション法における角度範囲の改良を実施したもの、条件(ii)は、条件(i)に加え、更に分枝限定法における下界値の算出方式の改良を加えたもの、条件(iii)は、条件(ii)に加え、更にハンドパスのシミュレーション法に並列計算を導入したのとなっている。なお、それぞれ、10 回の試行による平均処理時間を記載している。また、本検証実験は 24 コア 48 スレッドマシン (OS: CentOS Linux release 7.6.1810 (Core), CPU: AMD EPYC 7402 (24Core 2.8GHz 128MB)×2, メモリ: 16GB DDR4-3200 REG ECC (合計 256GB)) で実施している。なお、実装は、Java(jdk-11.0.7)により行い、並列計算には、非同期処理を行うために用いられる

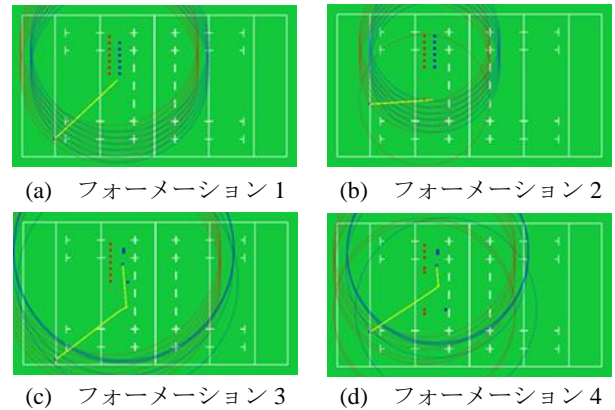


図 1 各フォーメーション例での算出結果

表 1 平均処理時間(秒)の比較

| | (a) | (b) | (c) | (d) |
|---------|--------|-------|--------|----------|
| 改良なし | 19.375 | 4.315 | 68.804 | 1786.619 |
| 条件(i) | 8.200 | 2.158 | 17.614 | 339.955 |
| 条件(ii) | 8.068 | 2.064 | 17.550 | 30.296 |
| 条件(iii) | 3.072 | 1.720 | 3.545 | 4.107 |

CompletableFuture クラスを使用している。表 1 に示されるように、いずれのフォーメーション例でも、以前導入した 2 つの改良[3]に加え、新たに並列計算を導入したことにより、大きく高速化されたことが見て取れる。また、その結果、リアルタイム処理とまではいかなかったが、いずれのフォーメーション例でも数秒での計算ができていたことも確認できる。

5. おわりに

本稿では、最適攻撃プレー算出法の実装に並列計算を導入し、仮想的フォーメーション例を用いて、その有効性を検証した。今後は、更なる高速化を目指した改良を加えるとともに、試合動画から、選手やボールの位置や速度のデータを取得し、より多くのフォーメーションでの検証をしていく。

謝辞

本研究を進めるにあたり、同研究室の皆様には、多くの貴重なコメントやアドバイスを頂きました。深く感謝申し上げます。

参考文献

- [1] 八代 航太朗, 中田 洋平, “7 人制ラグビーにおけるパスとランを考慮した最適攻撃プレーの算出法”, 2019 年映像情報メディア学会冬季大会, 11A-4 (2019).
- [2] K. Yashiro and Y. Nakada, “Computational Method for Optimal Attack Play Consisting of Run Plays and Hand-pass Plays for Sevenside Rugby”, Proc. 22nd IEEE International Symposium on Multimedia, pp. 145-148 (2020).
- [3] 八代 航太朗, 中田 洋平, “7 人制ラグビーにおけるランとハンドパスを考慮した最適攻撃プレー算出法的高速化”, 2021 年電子情報通信学会総合大会, D-12-19 (2021).
- [4] 藤村 光, 杉原 厚吉, “優勢領域に基づいたスポーツチームワークの定量的評価”, 電子情報通信学会論文誌, vol. J87-D-II, no. 3, pp. 818-828 (2004).