CF-006

反実仮想後悔最小化による アメリカンフットボールにおけるオフェンス戦略の均衡推定

島野 雄貴 * 阿部 拳之 † 岩崎 敦 ‡ 大河原 一憲 ‡
Yuki Shimano Kenshi Abe Atsushi Iwasaki Kazunori Ohkawara

概要

本論文では，アメリカンフットボールをゲーム理論の枠組みで定式化しオフェンスの均衡を導出する．スポーツにおいて，状況に応じて適切な戦術を決定することは勝率を上げることに繋がる．近年は，情報処理技術の発展や試合データの高度化・取得の容易化から，計算機を用いて戦術の分析・推定を行うことが主流となりつつある．この際重要となるのは，そのスポーツの特性を担保したまま分析を行うという点である．そこで本論文では，アメリカンフットボールをオフェンスおよびディフェンスからなる 2 人零和不完全情報展開型ゲームとして定式化し，そのゲームのナッシュ均衡を求める．ナッシュ均衡の導出にはテキサスホールデムポーカーをはじめとする不完全情報ゲームの解を求めることで有名な反実仮想後悔最小化を用いる．このアルゴリズムを用いて，NFL の試合データと合わせて様々な設定で近似ナッシュ均衡を計算し，得た戦略について吟味する．

その結果，現在の状況に応じた戦略，つまり過去に互いがどのような戦術を選択し，その結果何 yd 進んだのかという履歴に応じた戦略がオフェンスの均衡となることを確認した．また，行動空間を拡張して分析を行うことが，従来研究のように行動空間を狭めて分析することよりも優位であることを示した．更に，試合データから算出した戦術の比率に従って行動するよりも，ナッシュ均衡に従って行動することで勝率が上がることを示した．

1 はじめに

スポーツでは長年，経験則や勘など，非科学的な要素を用いて作戦を決定・分析することが主流だった．しかし，セイバーメトリクスを題材として扱った映画「マネー・ボール」が 2011 年に放映されてい以降，スポーツにおいて統計学を背景とした作戦決定・分析が急速に広まった．このようなスポーツにおけるデータ分析をスポーツアナリティクスと呼び，データサイエンスと密接な関わりを持つ．情報処理技術の発達，データの高度化，データ取得の容易化から，これまで以上にデータの活用が進んでいる．ここ近年でデータの活用が最も進んでいるスポーツの一つにアメリカンフットボールがある．National Football League（NFL）では，各選手に装着した RFID から選手の軌道をキャプチャし，機械学習などを活用することで，戦術の評価を行っている [*1]．また，戦術評価の方法はデータ分析プラットフォームである Kaggle [*2]にてコンペティション形式で募集するなど，計算機を用いた分析やデータ活用に対する熱量が伺える．

ここで，アメリカンフットボール（以降，アメフト）とはどのようなスポーツか説明をする．アメフトは，アメリカ合衆国における 4 大スポーツの一つであり，その中でも最も人気の高いスポーツである．アメフトの試合は，大まかにオフェンス・ディフェンスの 2 つのチームに分かれて展開する．オフェンスチームは最大で 4 回の攻撃権を持ち，一般に各攻撃回を 1st Down〜4th Down と呼称する．その攻撃回数の中で 10 ヤード（以降，yd）以上前進することができれば，新たに 4 回の攻撃権を獲得する．これを 1st Down の更新（フレッシュ）と呼ぶ．オフェンスチームはフレッシュを繰り返し行い，エンドゾーンと呼ばれる得点を獲得できるエリアまでボールを運ぶこと（タッチダウン）が最終的な目的となる．オフェンスチームは如何にボールを前進させるか，また一方でディフェンスチームは如何にそれを阻止するか，互いの戦術を高度に読み合いながら戦略を組み立てるのがアメフトの大きな特徴である．その他の特徴として，アメフトはラグビーやサッカーと異なり，ボールを持っている人がタックルされて倒れたり，プレイが失敗した際にゲームは一度そこで止まるため，プレイの区切りが明確な点がある．また，野球のようにオフェンス側とディフェンス側が明確に分かれている点も特徴の一つと言える．以上の特徴から，その時のシチュエーションや相手の取りうる戦略に対して迅速かつ効率的に，合理的な作戦を決定することが勝利への鍵となる．

アメフトのルールや特性から，データサイエンスと親和性が高く，戦略の推定・分析に関してこれまでに様々な研究が行われてきた．島野ら [1] は関東学生アメリカンフットボールの試合から得たデータからシミュレータを構築し，オフェンスの最適な戦略選択について考察を行った．また，高柳ら [2] は

* 無所属
† 株式会社サイバーエージェント
‡ 電気通信大学

[*1] NFL Next Gen Stats
https://nextgenstats.nfl.com
[*2] 予測モデリング及び分析手法関連のプラットフォーム
https://www.kaggle.com

深層強化学習を用いて，アメフトのあるシチュエーションにおいて，どのような戦術をどのプレイヤが行うのが望ましいかを分析した．いずれの研究もディフェンスを固定環境として扱い，戦略の推定・分析を行っていた点が課題である．先に述べた通り，アメフトはオフェンス・ディフェンス双方が高度に互いの戦略を読み合い，その時々の戦略を決定するスポーツである．このような性質から，ディフェンスを固定環境として扱うことは非現実的であると言える．

以上から，アメフトにおける戦略推定・分析はゲーム理論の枠組みで行うことが望ましいと考える．ゲーム理論を用いた推定・分析はこれまでにも行われてきた [3, 4, 5] が，いくつか課題がある．一つは戦略空間が狭い点である．従来の分析では，オフェンスが選択可能な行動（アメフトにおける戦術）をランかパスの 2 種に限定してゲームの構築を行っている．一般に，オフェンスの戦術は数多く存在しており，各戦術に特徴が存在するため，それらの特徴を反映しないまま戦略空間を設定することは望ましくない．もう一つは標準型ゲームとして分析を行っている点である．過去の結果も加味した上で戦術を決定するゲーム性から，一回限りのゲームとして分析を行うことは非現実的である．

そこで本論文では，従来の研究から戦略空間を拡張した上で，アメフトを 2 人零和不完全情報展開型ゲームとして定式化し，そのゲームのナッシュ均衡を求める．ナッシュ均衡の導出にはテキサスホールデムポーカーをはじめとする，不完全情報ゲームの近似ナッシュ均衡を求めることで知られている反実仮想後悔最小化（Counterfactual Regret Minimization [6]）を用いる．このアルゴリズムを用いて，NFL の試合データと合わせて様々な設定で近似ナッシュ均衡を計算し，得た戦略について吟味する．その結果，現在の状況に応じた戦略，つまり過去に互いがどのような戦術を選択し，その結果何 yd 進んだのかという履歴に応じた戦略がオフェンスの均衡となることを確認した．また，行動空間を拡張して分析を行うことが，行動空間を狭めて分析することよりも優位であることを示した．更に，試合データから算出した戦術の比率に従って行動するよりも，ナッシュ均衡に従って行動することで勝率が上がることを示した．

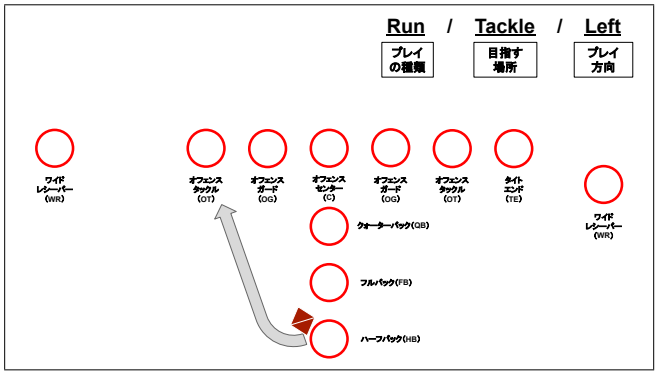

図 1: オフェンスの戦術例（$Run|Tackle|Left$）

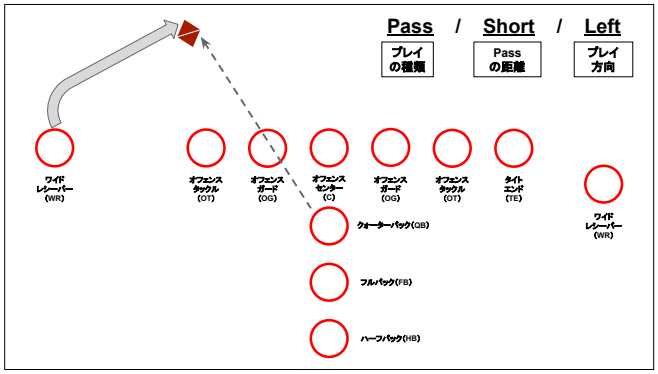

図 2: オフェンスの戦術例（$Pass|Short|Left$）

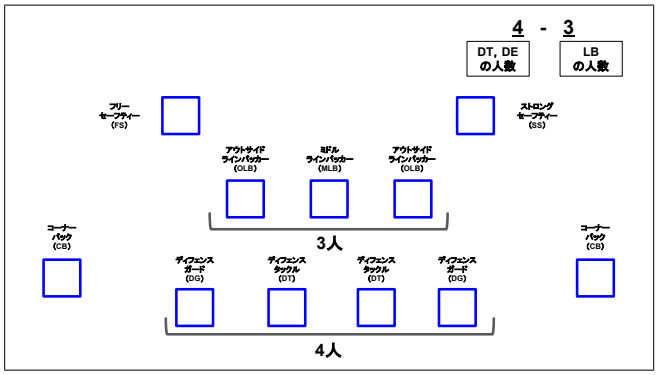

図 3: ディフェンスの戦術例（$4-3$）

2　問題設定

2.1　2 人零和不完全情報展開型ゲーム

本論文では，アメフトを 2 人零和不完全情報展開型ゲームとして定式化する．一般的な 2 人零和不完全情報展開型ゲームを以下の要素で定義する．

- プレイヤの有限集合 $N = \{1, ..., n\}$．アメフトにおける 2 人ゲームでは $N = \{Offense, Defense\}$ となる．
- 履歴 h の有限集合 H．
  すべての h は $h \in H$ を満たし，かつ H は空列を含む．
- 終端履歴 z の有限集合 Z．
  すべての z は $z \in Z$ かつ $Z \subset H$ を満たす．
- 非終端履歴 $h \in H \backslash Z$ における行動するプレイヤを表す関数 $P(h) \in N \cup \{c\}$．$P(h) = c$ ならば，次の行動はある確率分布に従う．
- 非終端履歴 $h \in H \backslash Z$ において，選択可能な行動 a の集合 $A(h)$．
- $P(h) = c$ の時の行動の確率分布を示す関数 σ_c．
  $\sigma_c(a \mid h)$ は h を所与とした場合の行動 a の確率分布を与える．これらは異なる h について独立である．
- プレイヤ i の情報分割 $\mathcal{I}_i$ と，その要素である情報集合 $I_i \in \mathcal{I}_i$．プレイヤ i から $h \in H$ と $h' \in H$ の区別がつかない場合，二つの履歴は同一の情報集合として扱う．また，$P(h)$，$A(h)$ をそれぞれ $P(I)$，$A(I)$ と表すこともある．

- プレイヤ i の利得関数 u_i.
  u_i の定義域は終端履歴 Z であり，その終端履歴 z における利得実数値を返す．零和ゲームのため，すべての z において $\sum_{i \in N} u_i(z) = 0$ を満たす．

2.2 各プレイヤの行動空間

続いて，本論文における各プレイヤが選択可能な行動（アメフトにおける戦術）を定義する．まず，オフェンスの行動について定義する．オフェンスが選択可能な行動は Run と Pass の 2 種類に大別する．そのうち，Run は【目指すポジション】と【方向】で細分化し，Pass は【距離】と【方向】で細分化する．以上から，本論文におけるオフェンスが選択可能な行動は以下の 13 種類となる．なお，Run の方向が $Middle$ の時のみ目指すポジションを定義する必要がないため N/A（Not applicable，該当なし）とする．

$$\{Run\} \times \{N/A\} \times \{Middle\}$$
$$\{Run\} \times \{Guard,\ Tackle,\ End\} \times \{Right,\ Left\}$$
$$\{Pass\} \times \{Short,\ Deep\} \times \{Middle,\ Right,\ Left\}$$

Run および Pass の戦術例を図 1 および図 2 に示す．戦略が $Run|Tackle|Left$ の場合は，「$Left$ の $Tackle$ 方向に Run をプレイする」ことを意味する．そして，戦略が $Pass|Short|Left$ の場合は「$Left$ に $Short$ の $Pass$ をプレイする」ことを意味する．一般に、Run は獲得できる yd は多くないがプレイが成功する確率が高い．一方で，Pass は獲得できる yd が多いが成功する確率は Run と比べると低い．ディフェンスの行動は【フォーメーション】で大別し，以下の 3 種類となる．

$$\{4-2,\ 4-3,\ 3-4\}$$

ディフェンスの戦術例を図 3 に示す．ディフェンスのポジションは大きく前衛・中衛・後衛の 3 層で構成する，フォーメーションは，前衛および中衛にどれだけ人を配置するかで分けることができる．戦略が $4-3$ の場合は，「前衛の人数が 4 人，中衛の人数が 3 人」のフォーメーションを選択することを意味する．一般に，前衛の人数が多ければ Run に強くなり，前衛・中衛の総人数が少なければ後衛の人数が増えるため Pass に強くなる．

2.3 ゲーム構造と利得

はじめに本論文におけるアメフトのゲーム設定について定義する．アメフトのゲームを全て記述した場合，計算量が膨大になるため，本論文では以下の設定とする．

- 1st Down〜3rd Down までのゲームとする．
- 新たな攻撃権を獲得したらそこでゲーム終了とする．
- フィールド上の現在地やタッチダウンまでの距離は考慮しない．

オフェンスは 4 回攻撃権を所有しているため，最大 4th Down まで攻撃が可能である．しかし一般に，4th Down は陣地回復のために攻撃権を放棄することが多いため 3rd Down までのゲームとして設定した．

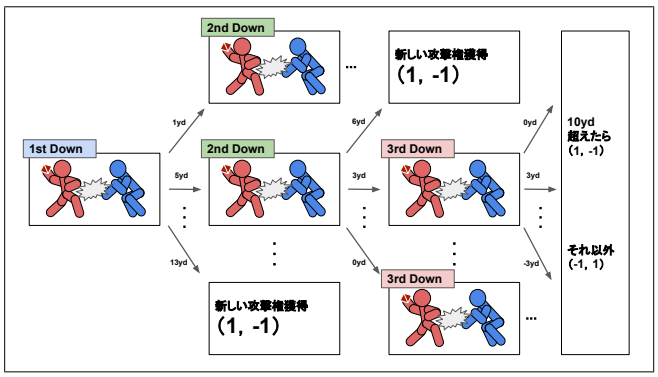

図 4: ゲーム構造概観

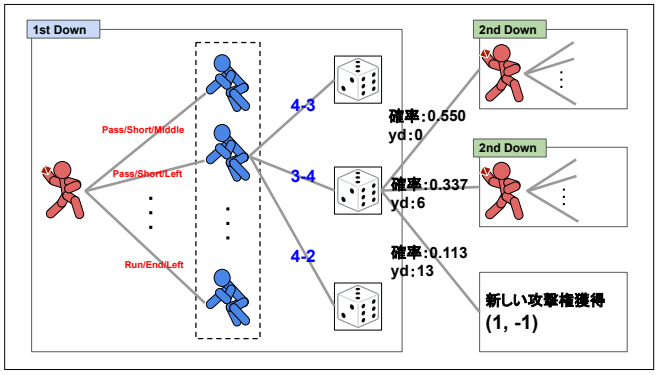

図 5: ゲーム詳細構造

以上を前提として，ゲームの構造と利得について定義する．ゲーム構造概観を図 4 に示す．ゲーム開始時点は 1st Down であり，次の攻撃権獲得までに必要な yd 数，つまりオフェンスが正の利得を獲得するのに必要な yd 数は 10 yd となる．各 Down にて，各プレイヤが選択した戦術の組に応じてオフェンスが獲得する yd が決まる．獲得した yd が過去分も含めて 10 yd を超えた時点でゲームが終了し，その時はオフェンスが 1，ディフェンスが-1 の利得を得る．一方，3rd Down 終了時点で獲得 yd が 10 yd を超えていなかった場合もゲームは終了し，この時はオフェンスが-1，ディフェンスが 1 の利得を得る．

続いて，ゲームの詳細構造を図 5 に示す．まず初めにオフェンスが戦術を選択し，その後にディフェンスが戦術を選択する構造となることが分かる．しかし，各 Down においてオフェンスとディフェンスは同時に戦術を選択するため，ディフェンスの意思決定点は全て同じ情報集合に属する．その後，各プレイヤが選択した戦術の組に応じて獲得する yd が決まり，ゲームが継続するか終了するかの判定を行う．なお，一般に，ある戦術の組に対する獲得 yd は一意に決まっていることはないため，ある確率分布に従い決定することとする．

ここで，1st Down でオフェンスが $Pass|Short|Left$ を選択し，ディフェンスが $3-4$ を選択した時の利得表を表 1 に示す．ここでオフェンスは，ある確率で 0 yd，6 yd，13 yd のいずれかを獲得するものとする．13 yd を獲得した場合は，そこ

表 1: 1st Down 残り 10 yd の利得表（抜粋）

	$3-4$	
$Pass\|Short\|Left$	$(u_{\text{off}}(h'),\ u_{\text{def}}(h'))$	if got 0 yd
	$(u_{\text{off}}(h''),\ u_{\text{def}}(h''))$	if got 6 yd
	$(1,\ -1)$	if got 13 yd

でゲーム終了となり，オフェンスが 1，ディフェンスが-1 の利得を獲得する．それ以外であればゲームは継続するので，それぞれが 2nd Down 以降の期待利得 u_{off}，u_{def} を獲得する．なお，h' および h'' は Down 開始時の履歴 h から，互いに戦術を選択し，その結果から yd を獲得した後の履歴を示す．ここの例では，1st Down 残り 10 yd はゲームの開始時点となるため $h = \varnothing$ であり，各履歴は以下で記述できる．

$$h' = \{Pass|Short|Left,\ 3-4,\ 0\ yd\}$$
$$h'' = \{Pass|Short|Left,\ 3-4,\ 6\ yd\}$$

なお，$\{Pass|Short|Left,\ 3-4,\ 13\ yd\} \in Z$ である．

2.4 試合データによる獲得 yd 数の決定

ある戦術の組に対する獲得 yd は σ_c に従い決まることとし，σ_c は 2017 年の NFL の試合データから求める．その試合データから戦術の組に対する獲得 yd を集計し，外れ値を除外後に任意の bin 数でヒストグラムを作成する．なお bin 数とは，ヒストグラムにおけるグラフの柱の数を指す．bin=3 であれば，ヒストグラムは 3 本の柱で構成される．そして，各 bin 内の最頻値を獲得 yd，各 bin の面積をその yd を選択する確率とする．オフェンスが $Pass|Short|Left$ を選択し，ディフェンスが $3-4$ を選択した際の bin=3 でのヒストグラムを図 6 に示す．このヒストグラムを作成した結果，$Pass|Short|Left$ と $3-4$ の組み合わせではオフェンスは 0 yd を 0.550，6 yd を 0.337，13 yd を 0.113 の確率で獲得することとなる．bin 数が大きければ大きいほど，σ_c は実際の獲得 yd の分布に近づく．なお，獲得 yd が 6 yd で，その選択確率が 0.337 の場合の σ_c は以下で記述できる．

$$\sigma_c(6\ yd) = 0.337$$

3 反実仮想後悔最小化

反実仮想後悔最小化は，同じゲームを何度も繰り返すことで近似ナッシュ均衡を求めるアルゴリズムである．各ゲームで「別の戦術を選択していればより利得を獲得できた」という後悔値を算出し，その後悔値をもとに次のゲームの戦略を更新していき，近似ナッシュ均衡を求める．

3.1 戦略とナッシュ均衡

プレイヤ i の戦略は，情報集合 I_i における可能な行動 $a \in A(I_i)$ の確率分布 σ_i となる．また，プレイヤ i の戦略 σ_i の全体集合は戦略集合 Σ_i となる．プレイヤ全体の戦略の集合は戦略プロファイル σ であり，戦略プロファイルからプレイヤ i の戦略のみを除いたものは σ_{-i} となる．

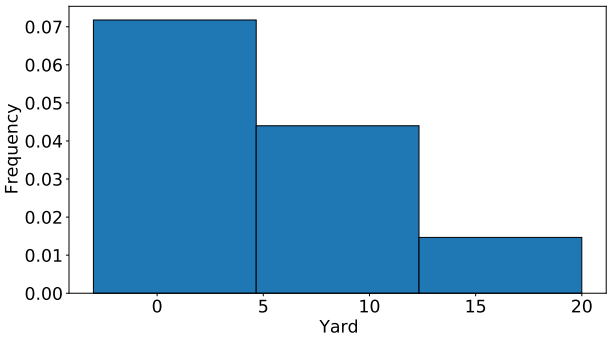

図 6: $Pass|Short|Left$ と $3-4$ における獲得 yd 分布（bin=3）

2 人展開型ゲームにおける一般的な解をナッシュ均衡と呼ぶ．ナッシュ均衡は，以下の式を満たす戦略プロファイル σ^* である．

$$u_i(\sigma_i^*,\ \sigma_{-i}^*) \geq \max_{\sigma_i \in \Sigma_i} u_i(\sigma_i^{'},\ \sigma_{-i}^*) \quad \forall i \in N$$

また，以下の式を満たす戦略プロファイル σ^* を ϵ-ナッシュ均衡と呼ぶ．

$$u_i(\sigma_i^*,\ \sigma_{-i}^*) + \epsilon \geq \max_{\sigma_i \in \Sigma_i} u_i(\sigma_i^{'},\ \sigma_{-i}^*) \quad \forall i \in N$$

2 人零和不完全情報ゲームにおいて，ナッシュ均衡を適切に導出することは重要である．それは，ナッシュ均衡は 2 人零和不完全情報ゲームにおいて，利得の損失を最小限に抑えることができるという性質を持つため，相手の様々な戦略に対して安定的に強い戦略であると言えるからである．ナッシュ均衡は前述した定義から，全プレイヤについて自分だけ戦略を変更してもそれ以上期待利得を上げることができない戦略の組であることを示している．特に 2 人零和ゲームの場合，ナッシュ均衡 σ^* に従うことで相手の戦略に関係なく自分の利得を $u_i(\sigma_i^*, \sigma_{-i}^*)$ にすることができる．

3.2 Regret Minimization

$t \in [T]$ 期におけるプレイヤ i の戦略を σ_i^t としたとき，T 期におけるプレイヤ i の average overall regret を以下で定義する．

$$R_i^T = \frac{1}{T} \max_{\sigma_i^* \in \Sigma_i} \sum_{t=1}^{T} \left(u_i(\sigma_i^*,\ \sigma_{-i}^t) - u_i(\sigma^t) \right)$$

更に，プレイヤ i の T 期までの average strategy を $\overline{\sigma}_i^T$ とする．特に，各情報集合 $I \in \mathcal{I}_i$ および各行動 $a \in A(I)$ における average strategy を以下で定義する．

$$\overline{\sigma}_i^T(I, a) = \frac{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)\sigma_i^t(I, a)}{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)}$$

2 人零和ゲームにおいて，T 期における両プレイヤの R_i^T が ϵ 以下であれば，$\overline{\sigma}^T$ は 2ϵ 均衡となる [6]．つまり，$|N| = 2$ において，average overall regret が収束すれば ϵ-ナッシュ均衡に収束する average strategy である $\overline{\sigma}_i^T$ が導出できることを示している．

3.3 Counterfactual Regret

ここで，プレイヤ i が情報集合 I に到達するように行動し，他のプレイヤが σ^t に従い行動した場合の counterfactual value を以下で定義する．

$$v_i(\sigma^t, I) = \sum_{h \in I} \pi^{\sigma^t}_{-i}(h) \sum_{z \in Z_h} \pi^{\sigma^t}(h, z) u_i(z)$$

なお，$\pi^{\sigma^t}_{-i}(h)$ はプレイヤ i 以外が σ に従って行動し，i が必ず h に向かうように行動した場合の h への到達確率を示す．また，$\pi^{\sigma^t}(h, z)$ は全プレイヤが σ^t に従って h に到達した状態で z に到達する確率を示す．以上から，新たに immediate counterfactual regret を以下で定義する．

$$R^T_{i,\mathrm{imm}}(I) = \frac{1}{T} \max_{a \in A(I)} \sum_{t=1}^{T} \left(v_i(\sigma^t_{I \to a}, I) - v_i(\sigma^t, I) \right)$$

なお，$\sigma^t_{I \to a}$ はプレイヤ i が情報集合 I で行動 a を選択し，それ以降の行動選択は σ^t に従う戦略を示す．$R^{T,+}_{i,\mathrm{imm}}(I) = \max(R^T_i(I),\, 0)$ としたとき，immediate counterfactual regret と average overall regret の間には以下の関係が成り立つ [6].

$$R^T_i \leq \sum_{I \in \mathcal{I}_i} R^{T,+}_{i,\mathrm{imm}}(I)$$

つまり，$\sum_{I \in \mathcal{I}_i} R^{T,+}_{i,\mathrm{imm}}(I)$ を最小化できれば R^T_i が最小化でき，R^T_i と $\overline{\sigma}^T_i$ の関係から ϵ-ナッシュ均衡を求めることができることを示している．最終的に各情報集合毎に独立して regret を更新する．このとき，

$$R^T_i(I, a) = \frac{1}{T} \sum_{t=1}^{T} \left(v_i(\sigma^t_{I \to a}, I) - v_i(\sigma^t, I) \right)$$

とし，$R^{T,+}_i(I, a) = \max(R^T_i(I, a),\, 0)$ として，$T+1$ 期の戦略を以下の式に基づいて更新する．

$$\sigma^{T+1}_i(I, a) = \begin{cases} \dfrac{R^{T,+}_i(I,a)}{\sum_{a \in A(I)} R^{T,+}_i(I,a)} & \text{if } \sum_{a \in A(I)} R^{T,+}_i(I, a) > 0 \\ \dfrac{1}{|A(I)|} & \text{otherwise.} \end{cases}$$

この繰り返しアルゴリズムを用いることで average overall regret は 0 に収束し，近似的にナッシュ均衡を用いることができる [6]．この繰り返しのアルゴリズムこそが反実仮想後悔最小化となる．疑似アルゴリズムを Algorithm 1 に示す．

Algorithm 1 反実仮想後悔最小化

$\forall I, a \in A(I)$: $R(I, a) \leftarrow 0$
$\forall I, a \in A(I)$: $\overline{\sigma}(I, a) \leftarrow 0$
$\forall I, a \in A(I)$: $\sigma^1(I, a) \leftarrow 1/|A(I)|$
function CFR(h, i, t, π_i, π_{-i})
  if h is terminal **then**
    return $u_i(h)$
  else if h is chance node **then**
    return $\sum_{a \in A(I)} \sigma_c(h, a)$ CFR($ha, i, t, \pi_i, \sigma_c(h, a) \cdot \pi_{-1}$)
  Let I be the information set containing h.
  for $a \in A(I)$ **do**

$$\sigma^{T+1}_i(I)(a) = \begin{cases} \dfrac{R^{T,+}_i(I,a)}{\sum_{a \in A(I)} R^{T,+}_i(I,a)} & \text{if } \sum_{a \in A(I)} R^{T,+}_i(I, a) > 0 \\ \dfrac{1}{|A(I)|} & \text{otherwise.} \end{cases}$$

  $v_\sigma \leftarrow 0$
  $v_{\sigma_{I \to a}}(a) \leftarrow 0$ for all $a \in A(I)$
  for $a \in A(I)$ **do**
    if $P(h) = i$ **then**
      $v_{\sigma_{I \to a}}(a) \leftarrow$ CFR($ha, i, t, \sigma_c(h, a) \cdot \pi_i, \pi_{-1}$)
    else
      $v_{\sigma_{I \to a}}(a) \leftarrow$ CFR($ha, i, t, \pi_i, \sigma_c(h, a) \cdot \pi_{-1}$)
    $v_\sigma \leftarrow v_\sigma + \sigma^t(I, a) \cdot v_{\sigma_{I \to a}}(a)$
  if $P(h) = i$ **then**
    for $a \in A(I)$ **do**
      $R(I, a) \leftarrow R(I, a) + \pi_{-i} \cdot (v_{\sigma_{I \to a}}(a) - v_\sigma)$
      $\overline{\sigma}(I, a) \leftarrow \overline{\sigma}(I, a) + \pi_i \cdot \sigma^t(I, a)$
  return v_σ
function SOLVE
  for $t = 1, 2, ..., T$ **do**
    for $i \in \{1, 2\}$ **do**
      CFR($\varnothing, i, t, 1, 1$)
  return $\overline{\sigma}$

4 計算機実験

前章で述べた設定におけるオフェンス側の均衡戦略を，反実仮想後悔最小化アルゴリズムで求める計算機実験を行う．本実験において，アルゴリズムの反復回数は 10,000 回とし，ヒストグラムの bin 数は {3, 4, 5} とする．各設定で反実仮想後悔最小化により導出したオフェンスの均衡戦略，つまり各履歴ごとの行動選択確率の一部を表 4 に示す [*3]．表 4 は，反実仮想後悔最小化アルゴリズムでも求めた近似ナッシュ均衡を構成する戦略が，実際の試合で観測されている例になっていることを示している．例えば，2nd Down 残り 2 yd で履歴 $\{Pass|Short|Middle,\, 4-3,\, 8\ yd\}$ を観測したあとの均衡は，$Run|Tackle|Left$ を 0.570, $Run|Tackle|Right$ を 0.419, $Run|Guard|Right$ を 0.011 となっている．このように，残り yd が少ない時は，獲得しうる yd 数は少ないが成功確率が高い Run 系をより選択するなど，実際の試合でも観測可能な戦略が均衡となることを確認した．

以降は，本計算機実験で導出した近似ナッシュ均衡を用いて追加の議論を行う．

4.1 Exploitability の算出

導出した結果がどれだけナッシュ均衡に近いのか評価をするため，Exploitability という指標 [7] を用いる．2 人零和ゲー

[*3] 全結果は以下 URL を参照 https://www.dropbox.com/sh/0yd2b7vnrtoq43j/AADhBpWThn6dL5PA6AUP09h2a?dl=0

表 2: 戦略の違いによる各プレイヤの勝利数

		ディフェンス	
		統計データ	ナッシュ均衡
オフェンス	統計データ	(5,226, 4,774)	(4,778, 5,222)
	ナッシュ均衡	(7,716, 2,284)	(7,676, 2,324)

ムにおける Exploitability を以下で定義する．

$$\epsilon_\sigma = \max_{\sigma_1'} u_1(\sigma_1', \sigma_2) + \max_{\sigma_2'} u_2(\sigma_1, \sigma_2')$$

ϵ_σ の値が大きいほど σ がナッシュ均衡から遠い戦略であることを示す．σ がナッシュ均衡のときは，

$$\begin{aligned}\epsilon_\sigma &= \max_{\sigma_1'} u_1(\sigma_1', \sigma_2) + \max_{\sigma_2'} u_2(\sigma_1, \sigma_2') \\ &= u_1(\sigma) + u_2(\sigma) \\ &= u_1(\sigma) - u_1(\sigma) \\ &= 0\end{aligned}$$

となるため，ϵ_σ が 0 に近いほど σ がナッシュ均衡に近い戦略であることを示している．この Exploitability をアルゴリズム内の任意の期で算出し，その推移について確認する．Exploitability の推移を示したものを図 7 に示す．最終期における Exploitability は bin = 3 のときに 2.03×10^{-3}，bin = 4 のときに 1.98×10^{-3}，bin = 5 のときに 1.42×10^{-3} となっており，いずれも 10^{-3} 近傍まで収束してる．従来研究の結果から，今回導出した均衡は十分にナッシュ均衡に近いと述べることができる [8, 9, 10].

最終期における Exploitability は bin が大きい時のほうが，bin が小さいときよりも小さくなるが，これは bin 数が大きければ大きいほど戦術を細かく記述できるため，行動経路上にあるノード数が少なくなるためだとと予想している．ここで bin 数を 3 もしくは 5 として，履歴 $\{Pass|Short|Left,\ 4-3,\ 8\ yd,\ Run|Tackle|Right,\ 4-2,\ 0\ yd\}$ における，平均戦略 $\bar{\sigma}^t_{\text{off}}(I, a)$ の推移を表したものを図 8 および図 9 に示す．$t = 100$ 時点で，$\bar{\sigma}^t_{\text{off}}(I, a)$ が 0 以上のものは bin=3 では 6 種存在するのに対して，bin=5 では 3 種のみとなっている．

4.2 統計データに対するナッシュ均衡の有効性

本節では，本論文で導出した均衡戦略に従うことがどれだけ有効かを検証する．具体的には，均衡戦略に従って戦術を選択するプレイヤと，統計データに従って戦術を確率的に選択するプレイヤを対戦させる．ここで，統計データに従うプレイヤは，2017 年の NFL の試合データに対して【現在の Down】と【攻撃権更新までの残り yd】毎に算出した確率に従って戦術を選択するものとする．実験設定を以下に示す．

- オフェンスは 3 回の攻撃権を所有し，その中で 10 yd 以上獲得したらオフェンスの勝利．獲得できなかった場合はディフェンスの勝利とする．
- 各プレイヤは統計データ，ナッシュ均衡どちらかに従ってゲームを行う．
- bin 数は 3 で固定．
- どちらかの勝利が確定するまでを 1 シミュレーションとし，それを 10,000 回実施．
- 10,000 回のシミュレーションを 10 セット行い，その平均勝利数を算出．

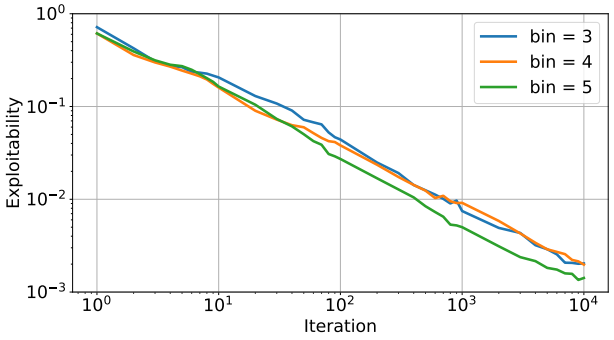

図 7: 各 bin における Exploitability の推移

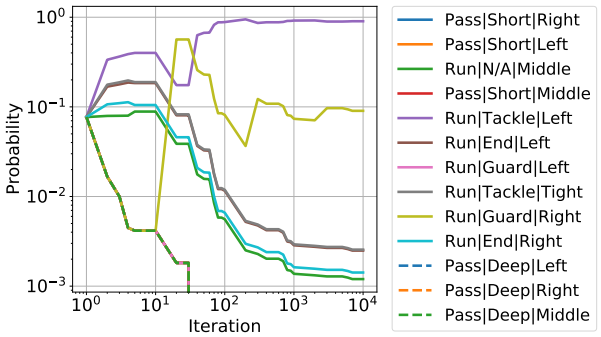

図 8: bin=3 における各戦術の確率の遷移

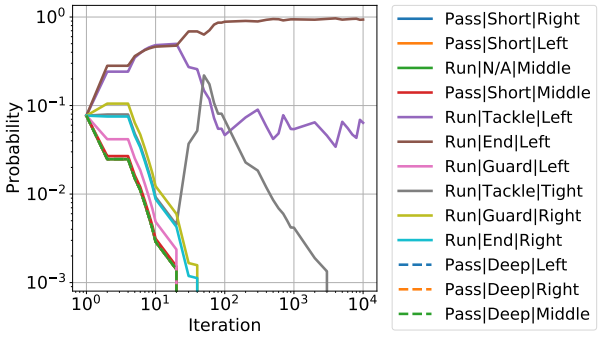

図 9: bin=5 における各戦術の確率の遷移

シミュレーションの結果を表 2 に示す．表 2 は，例えばオフェンス・ディフェンスが共に統計データに従い戦術を選択した場合に，オフェンスは平均で 5,226 回，ディフェンスは平均で 4,774 回勝利したことを示している．

オフェンスがナッシュ均衡に従い戦術を選択すると，ディフェンスが統計データに従い戦術を選択した場合は 7,716 回，ディフェンスがナッシュ均衡に従った際は 7,676 回オフェン

表 3: Exploitability の平均と標準偏差

設定	Exploitability	標準偏差
行動空間大：bin = 3	0.194	0.032
行動空間大：bin = 4	0.195	0.035
行動空間大：bin = 5	0.192	0.022
行動空間小：bin = 3	0.286	0.030
行動空間小：bin = 4	0.398	0.030
行動空間小：bin = 5	0.285	0.031
行動空間小：bin = 20	0.277	0.030

スが勝利することが表 2 から分かる．オフェンスがナッシュ均衡に従い戦術を選択することで，勝率を 5 割前後から 7 割後半まで引き上げており，実際にナッシュ均衡に従って戦術を決定することが勝率を上げるには有効であることを示している．以上から，ナッシュ均衡に従い戦術を選択することの重要性を本実験で示すことができた．

4.3 行動空間の大小による Exploitability の比較

本節では，行動空間の大きさが Exploitability に与える影響を吟味する．文献 [3] では，オフェンス側の行動空間を Run か Pass の 2 通りとしてナッシュ均衡を導出している．一方で，本論文ではオフェンス側の行動空間を 13 通りとしてナッシュ均衡を導出している．そこで，前者を【行動空間小】，後者を【行動空間大】として，それぞれの設定で計算した戦略の Exploitability を比較する．実験設定を以下に示す．

- アメフトのゲームの設定，およびディフェンスの行動空間は本論文のものに準拠する．
- 以下の 7 つの設定で均衡を導出．
  行動空間大：bin $\in \{3, 4, 5\}$
  行動空間小：bin $\in \{3, 4, 5, 20\}$
- 各設定で導出した均衡を【戦略空間大】の【bin = 30】のゲームにおける Exploitabiloity で評価．

算出した結果を表 3 に示す．例えば，表 3 における【行動空間大：bin = 3】の結果は，オフェンスは【行動空間大：bin = 3】の均衡戦略とし，各設定で導出したディフェンスの均衡戦略それぞれで Exploitability を導出したときの平均が 0.194 で，その標準偏差が 0.032 であることを示している．

行動空間小の設定における Exploitability の平均は 0.311 であり，行動空間大の設定における Exploitability の平均は 0.194 のため，行動空間大の設定で導出した戦略がよりナッシュ均衡に近いことを示している．先の実験から，ナッシュ均衡に従い戦術を選択することで勝率が上がることが判明しているので，以上から行動空間を広げて均衡を導出することが有効であると言える．

一方で，bin の大小に着目すると，bin が大きいからといって Exploitability が小さくなるとは限らないことが分かる．bin が大きくなるということは，戦術の組に対する獲得 yd の分布が現実の分布に近づくことを意味するため，Exploitability と標準偏差は共に小さくなると予想したが，そのような結果にはならなかった．bin の粒度と均衡の関係性については今後の研究課題となる．

5 おわりに

本論文では，アメリカンフットボールを元にした 2 人零和不完全情報展開型ゲームを定式化し，その近似ナッシュ均衡を反実仮想後悔最小化アルゴリズムを用いて様々な設定で計算し，得た戦略を吟味した．そこでまず，状況や行動空間を制限したゲームを構築し，アルゴリズムが現実的な時間で解を出せるように調整した．次に求めたオフェンスの戦略が，ナッシュ均衡を近似することを数値的に示した後，求めた戦略のいくつかが，実際のゲームでもしばしば観察する，理にかなった戦略であることを確認した．最後に，求めた戦略と統計データの確率に単純に従う戦略が対戦できるよう，異なる設定で計算した戦略を共通の設定で扱うマッピングを構築した．この結果，統計データに基づく戦略より，均衡を近似した戦略の方が優れていることがわかった．今後の課題として，設定の一般化 [11, 10] やアルゴリズムの高速化 [7] が挙げられる．

参考文献

[1] 島野雄貴, 福島稜規, 伊藤毅志, 岩崎敦, 大河原一憲. モンテカルロシミュレーションを用いたレッドゾーン内における最適な戦略推定. 第 34 回ゲーム情報学研究会, pp. 1–8, 2015.

[2] 高柳里紗, Dinesh B. Malla, 酒井剛, 曽我部東馬. 深層強化学習を用いたアメリカンフットボールコーチング戦略の研究. 第 34 回人工知能学会全国大会, 2020.

[3] Christopher B. Adams. *Learning Microeconometrics with R*. Chapman and Hall/CRC, 2020.

[4] Understanding balanced, effective play-calling with game theory and nash's equilibrium. `https://www.fieldgulls.com/seahawks-analysis/2015/1/9/7517551/understanding-effective-play-calling-with-game-theory`.

[5] Game theory and run/pass balance. `http://archive.advancedfootballanalytics.com/2008/06/game-theory-and-runpass-balance.html`.

[6] Martin Zinkevich, Micahel Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. *Proceedings of the 20th International Conference on Neural Information Processing Systems*, p. 1729–1736, 2007.

[7] Micahel Johanson, Kevin Waugh, Micahel Bowling, and Martin Zinkevich. Accelerating best response calculation in large extensive games. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, p. 258–265, 2011.

表 4: オフェンスの均衡（抜粋）

bin	Down	残り yd	履歴 h	均衡戦略
3	1	10	$\varnothing$	[Pass|Short|Left, Pass|Short|Middle, Pass|Deep|Right] = [0.001, 0.510, 0.489]
	2	10	{Pass|Short|Left, 4-3, 0 yd}	[Pass|Short|Middle, Pass|Deep|Left, Pass|Deep|Right, Pass|Deep|Middle] = [0.474, 0.002, 0.414, 0.110]
			{Pass|Short|Middle, 4-2, 0 yd}	[Pass|Short|Middle, Pass|Deep|Right, Pass|Deep|Middle] = [0.057, 0.298, 0.645]
		2	{Pass|Short|Left, 4-3, 8 yd}	[Run|Tackle|Left, Run|End|Left, Run|Tackle|Right, Run|Guard|Right] = [0.288, 0.005, 0.568, 0.139]
			{Pass|Short|Middle, 4-3, 8 yd}	[Run|Tackle|Left, Run|Tackle|Right, Run|Guard|Right] = [0.570, 0.419, 0.011]
4	1	10	$\varnothing$	[Pass|Short|Right, Pass|Short|Left, Pass|Short|Middle, Pass|Deep|Right] = [0.024, 0.410, 0.413, 0.153]
	2	10	{Pass|Short|Left, 4-3, 0 yd}	[Pass|Short|Middle, Pass|Short|Middle, Pass|Deep|Right] = [0.117, 0.413, 0.470]
			{Pass|Short|Middle, 4-2, 0 yd}	[Pass|Short|Right, Pass|Short|Left, Pass|Short|Middle, Pass|Deep|Right, Pass|Deep|Middle] = [0.122, 0.004, 0.494, 0.370, 0.010]
		2	{Pass|Short|Left, 4-3, 8 yd}	[Run/Tackle/Left, Run/Tackle/Right] = [0.368, 0.632]
			{Pass|Short|Middle, 4-2, 8 yd}	[Pass|Short|Right, Run|Tackle|Left, Run|Tackle|Right, Run|Guard|Right, Run|End|Right, Pass|Deep|Middle] = [0.004, 0.396, 0.513, 0.037, 0.040, 0.010]
5	1	10	$\varnothing$	[Pass|Short|Left, Pass|Short|Middle] = [0.607, 0393]
	2	10	{Pass|Short|Left, 4-3, 0 yd}	[Pass|Short|Middle, Pass|Deep|Right, Pass|Deep|Middle] = [0.401, 0.453, 0.146]
			{Pass|Short|Middle, 4-2, 0 yd}	[Pass|Short|Middle, Pass|Deep|Right] = [0.558, 0.442]
		2	{Pass|Short|Left, 4-3, 8 yd}	[Run|Tackle|Left, Run|Tackle|Right, Run|Guard|Right] = [0.803, 0.195, 0.002]
			{Pass|Short|Middle, 4-2, 8 yd}	[Run|Tackle|Left, Run|End|Left, Run|Tackle|Right, Run|Guard|Right] = [0.414, 0.351, 0.005, 0.230]

[8] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, p. 1078–1086, 2009.

[9] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. *Proceedings of the 36th International Conference on Machine Learning*, p. 793–802, 2019.

[10] Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019.

[11] Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, Vol. 359, No. 6374, pp. 418–424, 2018.