ポケモン対戦に対する UCT アルゴリズムの有効性の調査

猪原弘之 [1, a)]　小山聡 [1]　栗原正仁 [1]

概要：ゲーム AI はコンピュータ黎明期から盛んに研究されており，コンピュータが扱いやすいものから順に研究していく意味で特に完全情報ゲームの研究が中心だった．しかし，チェス，将棋，囲碁などの主要な完全情報ゲームにおける研究は人間のトッププロよりも高い実力が発揮できるレベルまでに達し，単純に強いだけの完全情報ゲームの AI を作る研究は終焉を迎えつつある．そこで，本稿では数多くある不完全情報ゲームの中でも，盛んに研究されているポーカーや麻雀などにないゲーム特性を持ったポケモン対戦を対象として，そのゲームが持つ特徴を紹介し，それぞれ異なる工夫を施したいくつかの UCT アルゴリズムの有効性を検証した．その結果，ポケモン対戦で UCT の実力を一番引き出す方法が分かったが，モンテカルロ法と UCT の実力に有意差がなく，ポケモン対戦には UCT アルゴリズムが有効とは言えない可能性があることが分かった．

Evaluating the Effectiveness of UCT Algorithms for Pokémon Battles

Hiroyuki Ihara[1, a)]　Satoshi Oyama[1]　Masahito Kurihara[1]

Abstract: The study of perfect information games attracted a lot of attention in the field of game AI. Recently, however, the study of major perfect information games such as chess, shogi, and the game of Go, has almost reached its goal, because their game AI became stronger than human top professionals. What should be our next goal? In this paper, we introduce Pokémon, which has unique characteristics other imperfect games do not have and evaluate the effectiveness of UCT algorithms for Pokémon Battles. The results show that there are no significant differences between the Monte Carlo method and the UCT algorithms in effectiveness, and there is a possibility that the UCT algorithms are not effective for Pokémon Battles.

1. はじめに

チェスなどの深い読みが必要なゲームは知性の象徴であり，コンピュータ黎明期から AI の分野で盛んに研究されてきた．長年の研究によりチェスや将棋などの主要なボードゲームの AI はトッププロ以上の力を発揮できるようになってきた．しかしながらゲームの特性上，評価関数が作りづらい囲碁では中々プロレベルの AI を作れずにいた．

試行錯誤の末，評価関数を作ることが難しいコンピュータ囲碁では，事前の知識を使わずに乱数によって終局図を数多く得ることで最善手を求めるモンテカルロ法と，探索空間が小さいゲームではαβ法などで解を得ることができる木探索の両方の良さを取り込んだ MCTS (Monte-Carlo Tree Search)が開発された．MCTS は囲碁の AI の実力を一段階上げることに成功し，今日では探索に UCB (Upper Confidence Bound)[1]を用いた UCT (UCB applied to Trees)[2]が主に使用されている．また，2016 年，MCTS とディープニューラルネットワークを合わせた方法で Alpha Go[3]が韓国の囲碁のチャンピオンに勝利したことで，完全情報ゲームに分類される古典的なテーブルゲームの強いだけの AI を作る研究は終わりを迎えつつある．

一方で，相手の状態が完全には分からない麻雀や人狼，カタンなどの不完全情報ゲームの強い AI を作る研究はまだまだ発展途上であり，これからゲーム AI 分野で注力すべき領域だと言える．UCT は囲碁などの完全情報ゲームのみならず，Skat[4]や麻雀[5]などの不完全情報ゲームでも有効性が確認されているが，全ての不完全情報ゲームにおいて有効性が確認されたわけではない．

UCT が有効であるか試されていない不完全情報ゲームの中に不完全不確定情報ゲームに分類される turn-based RPG のポケモン対戦がある．ポケモン対戦は典型的な turn-based RPG と同じく合法手の評価に相手のパラメータの情報が必須であるため，相手の状態推定が重要である特徴を持つ．また，ほぼ全ての合法手による状態遷移に乱数が関係している特徴も持つ．これらの特徴により，プレイアウトベースの合法手の評価の収束に時間がかかると予想されるため，UCT アルゴリズムが有効に働かない可能性がある．

ポケモン対戦は非常に有名であり，上記のような興味深い特徴を有していながらこれまであまり研究されてこなかった．そこで本稿ではポケモン対戦の概要を説明する．また，他の不完全情報ゲームに UCT アルゴリズムを用いた既存研究の方法を参考にしてポケモン対戦に対して UCT を実装し，有効性を検証した．

2. ポケモン対戦について

2.1 ポケモン対戦概要

ポケモンはゲームフリーク，クリーチャーズ，および任天堂から開発，発売された 1996 年より続く世界中で大人気の turn-based RPG のビデオゲームのシリーズである．ポケモン対戦には形式とルールが様々あるが，本稿におけるポケモン対戦とは，第 6 世代のシングルバトルのフラットルールを指すことにする．

第 6 世代の商品は X，Y，アルファルビー，オメガサファイアの四つがそれにあたる．ポケモンバトルにおいて世代と言うのはポケモンのシリーズの第何作目であるかを示

1 北海道大学大学院情報科学研究科
a) ihara_h@complex.ist.hokudai.ac.jp

しており，違う商品名の物であっても同じ世代であれば技の追加効果などの細かいルールは共通しているため，統一して扱うことができる．しかしながら，細かい効果の仕様の詳細などは開発元から公式に公表されているわけではない.プレイヤーは大体の仕様を把握しながら戦っているが，まれなケースで効果同士の衝突がどのように処理されるかは正しく把握できていないこともある．今後の課題で後述するが，これは研究者全体でポケモン対戦のゲーム AI のコンペティションなどを行おうとしたときの妨げになると思われる．

シングルバトルとはプレイヤーが場に出しておけるポケモンが一匹ずつであるルールで，他にダブルバトルやトリプルバトル，ローテーションバトルなどがあるが最も競技人口が多く，人気があるのはシングルバトルである．そのため，本稿ではシングルバトルを扱う．

フラットルールとはポケモンのレベルと呼ばれる値が 50 以下に統一されるルールである．レベルは一般的に高ければ高いほど強いので 50 以下に統一して戦わせるほうが良い勝負になりやすい．そのため公式大会などではフラットルールが採用されている．

第 6 世代のシングルバトルのフラットルールではまず，二人のプレイヤーが六匹一組のポケモンを持ち寄ったところから対戦がスタートする．この六匹一組をパーティと一般的に呼ぶ．その後，対戦開始とともに相手のパーティに含まれるポケモンの種類が開示される．開示された相手のパーティを考えたうえで，プレイヤーは互いに自分のパーティから実際に対戦に使う三匹を選択する．このとき選ばれなかった三匹については対戦中に使われることはない．その後，実際に相手とポケモンを戦わせる本当の意味での対戦が開始される．対戦では現在自分が操作しているポケモンに技を使用させて相手のポケモンにダメージを与えるか，操作しているポケモンを自分の控えのポケモンと交代させるかを 2 人のプレイヤーがそれぞれ 1 ターンに 1 度同時に選び，相手より先に相手の全てのポケモンの HP と呼ばれる値を 0 にして勝利することを目指す．

このように，ポケモン対戦はパーティ選択，ポケモン選出，技選択による戦闘の三つのフェーズから成る．ポケモンではタイプと呼ばれる相性が重要であり，ポケモンはそれぞれ自分にとって有利なポケモンと不利なポケモンが必ず存在するため，最強のポケモンは存在せず，単体で強いポケモンだけを集めたパーティが強いわけではない．そのため，パーティ内で相性の不利を補いながら相手がどのようなパーティで来ても選出と実際の戦闘次第で勝てるようにパーティを作成する．選出についても三匹の不利な相性が重ならないように選出せねばならず，実際の戦闘で技を選択するときにもミスをしてはいけない．このように三つのフェーズそれぞれが高い次元に達しており，なおかつ互いにかみ合っていなければ強いポケモン対戦プレイヤーとは言えない．ポケモン対戦の強いゲーム AI を作っていくためにはいずれのフェーズも研究しなければならない．

2.2 ポケモン対戦の不完全情報要素

ポケモンの個体それぞれには多様なパラメータがあり，プレイヤーが育成の仕方を変えることである程度自由にパラメータを調整できる．中には値を少し変えただけではポケモンの強さがあまり変わらないパラメータもあるが，少し値を変えただけで鋭敏にポケモンの強さが変化するパラメータもある．そのため，相手のポケモンのパラメータの把握は大変重要であるが，相手側のほとんどのパラメータがゲーム開始時には把握できない．

表 1　ポケモンの代表的なパラメータの例

パラメータ	簡易説明	不完全情報
HP	体力のこと. 相手の体力は最大値に対する割合しかわからず, 最大値も分からない.	○
攻撃	技のダメージ計算に使用.	○
防御	技のダメージ計算に使用.	○
特攻	技のダメージ計算に使用.	○
特防	技のダメージ計算に使用.	○
素早さ	そのターンにどちらが早く技を出すかを決める.	○
基礎ポイント	HP，攻撃，防御，特攻，特防, 素早さを決めるのに使う値. それぞれに 252 まで, 全部で 508 までプレイヤーが割り振れる.	○
種族値	HP，攻撃，防御，特攻，特防, 素早さを決めるのに使う値. 種族ごとに固定.	×
個体値	HP，攻撃，防御，特攻，特防, 素早さを決めるのに使う値. ポケモンの個体ごとに固定.	○
技	ポケモンの種族ごとに数十種類から四つだけ持つことができる.	○
特性	ポケモンの種族ごとに数個から一つだけ選択される.	○
性格	25 種類から一つだけ選択できる.	○
持ち物	数百種類から一つだけ選択できる.	○
タイプ	18 種類から二つまで種族ごとに固定されている.	×

各プレイヤーが毎ターンに取ることのできる合法手の数は高々六つと少ないが，相手のポケモンのパラメータの状態数は概算で10^{34}もある上にそのほとんどが不完全情報で，1 試合中に取りうる状態数は概算で10^{142}もある．表 1 にポケモンのパラメータの一部をまとめた．

ポーカーや麻雀などの他の不完全情報ゲームでも不完全情報の推定は重要な事柄である．しかし，ポーカーでのショーダウン時の相手のハンドの強さや麻雀での相手の当たり牌や聴牌の推定は確率を使って処理することで回避することも可能であり，相手の不完全情報を推定できないことは必ずしも負けに直結しているわけではない．一方で一般的な turn-based RPG は互いの合法手がどれだけ他方にダメージを与えられるかが重要な評価指標である．しかし，ポケモン対戦は合法手のダメージ計算式中に相手の不完全情報の項が含まれているため，相手の不完全情報の推定を回避できず，相手の不完全情報の推定の誤りはゲームの敗北に直結する．この点は頻繁に研究対象とされているポーカーや麻雀と異なる点である．

2.3 ポケモン対戦の不確定情報要素

ポケモン対戦は 1 回の合法手による状態遷移に平均して 4 回ほど乱数による分岐があり，同じ状態で同じ合法手を選択しても同じ結果にならない可能性が非常に高く，不確定情報ゲームの側面も強いといえる．

ポケモンは HP が 0 になると行動不能になり，対戦に関与できなくなるためポケモンの HP が 0 か 1 以上かでは大きな差がある．しかしながら，ダメージを決定する計算式には乱数の項があり，0.85 から 1.00 まで 0.01 刻みで 16 段階の補正がかかる．また，毎回 16 分の 1 の確率でダメージが 1.5 倍になるなど乱数がかかわってくる部分が非常に多く，非常に稀だが運だけで初級者が上級者に勝ってしまうこともないわけではない．そのため，ゲーム AI の実力を測るために実際に対戦させてみる際は試行回数を多くとる必要がある．

3. 従来手法と検証した手法

UCT は Kocsis ら[2]が 2006 年に提案した手法で，MCTS の中でよく使われている手法の一つである．以下の手順を回数や時間の制限内に繰り返すことで探索する．

(1) UCB 値が高いノードを選択して根接点から末端接点まで木を辿る．
(2) (1)の動作でたどり着いた末端接点の訪問回数が閾値を超えていればその接点を展開して新たな接点を作り，再び末端接点まで木を辿る．
(3) 展開できない末端接点まで到達したらそこからは乱数により手を決定して終局まで行き，勝敗を得る．この行為をプレイアウトと呼ぶ．
(4) (3)のプレイアウトで得られた勝敗を今度は末端接点から根接点までフィードバックし，UCB 値を更新する．

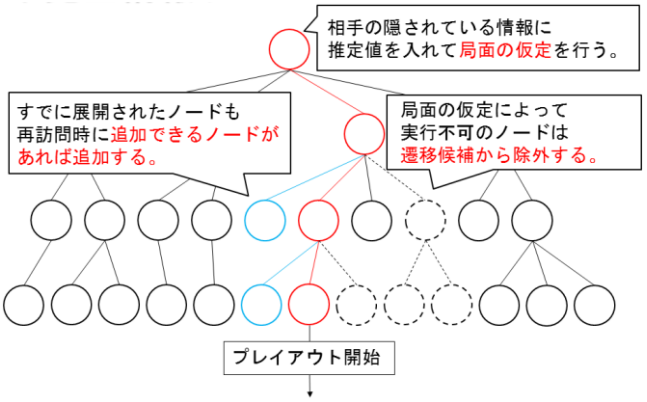

図 1　不完全情報ゲームの工夫を施した UCT

完全情報ゲームのアルゴリズムとして開発された UCT を不完全情報ゲームに用いる場合はいくつか工夫を施す必要がある．工夫の仕方については後述するが，簡易的な説明を図 1 にまとめた．

まず，不完全情報をありうる範囲の具体的な数値で固定し，各接点での合法手を確定するために局面の仮定をする必要がある．ポケモンでは相手の合法手も不完全情報であるため，UCT の手順(1)から開始するときには必ず局面の仮定を完了していなければならず，UCT の 1 イテレーションごとに局面の仮定を行う場合は(4)が終わって(1)から次のイテレーションが行われる間に局面の仮定をし直すことになる．他の不完全情報ゲームでも局面の仮定によって最善手が変わることがあるが，ポケモン対戦においてはそれが大変顕著であり，色々な局面を仮定してプレイアウトを作成することで慎重に合法手の評価をしなければならない．

また，ゲーム木における接点の区別も完全情報ゲームとは異なっている．本稿の実験に用いた UCT のゲーム木ではその接点に到達するまでに互いが使用した技のみを判別に用いており，使用した技以外が異なる状態同士であっても互いに使用した技が同じであればその二つを同じ状態，同じ接点として扱っている．これには利点と欠点の両方が存在する．まず，ポケモン対戦は乱数の影響が非常に強いため完全情報ゲームのときのように厳密に状態数が異なる場合を区別して接点をゲーム木に追加すると自然を含めなくとも接点が膨大な数になってしまい，現実的な時間で探索ができなくなってしまう．そのために最も重要な要素である毎ターンにどの技を選択したかで接点を区別した．一方で，実験の考察でも後述するが，異なる状態を一つにまとめてしまったために最適な手が異なる状態同士をまとめてしまう可能性も生じてしまった．これは直前までの探索で有効だった手の評価値は高くするという UCB のコンセプトと衝突してしまう可能性を含んでいる．ポケモン対戦では不完全不確定情報ゲームであるために状態数は膨大になってしまうのだが，状態数の集約と UCB の有効性のトレードオフをどのように解消するかがうまく UCT を適用す

表 2　比較した手法

	局面の仮定	ノードの追加	ノードの削除
MC	イテレーションごとに行う	行わない	行わない
UCT1	最初のイテレーションに 1 度だけ行う	行わない	行わない
UCT2	一定時間ごとに 5 回行い最終的に平均をとる	行わない	行わない
UCT3	イテレーションごとに行う	行わない	行わない
UCT4	イテレーションごとに行う	行わない	行う
UCT5	イテレーションごとに行う	行う	行う

るための課題であると言える．

また，状態数を集約した一連の UCT アルゴリズム中に何回も局面の仮定を行う場合，イテレーションごとに相手の意思決定接点で取れる合法手が変わってくるため，本来遷移できるはずのノードが存在しなかったり本来遷移できるはずのないノードがゲーム木上に存在したりしてしまうことがある．ポケモンは不確定情報ゲームの側面も持つため，ゲーム木の状態を集約すると，この問題は局面の仮定を 1 度しか行わない場合の自分の意思決定ノードでも発生してしまうことがある．この問題を回避するためには展開済みのノードに到達するたびに遷移できないノードの一時的な削除と遷移可能なノードの追加をする必要がある．

状態数の集約を前提として，局面の仮定とノードの追加，削除を施すことでポケモン対戦でも UCT アルゴリズムのゲーム木探索は問題なく行えるようになる．一方で，局面の仮定を 1 回だけ行うことで 1 つの局面に対して深い読みを実現でき，ノードの追加と削除をしないことで頑なに UCB1 値が一番高いノードに遷移することができる．そのため，これらの工夫を施さないことで本来の UCT のコンセプトに一番沿う形となる．ポケモン対戦は同じ選択肢を選んでも起こる事象は毎回違うため，ノードの遷移に失敗したら親ノードに戻ることでノードの遷移に失敗したところまでの探索は無駄になってしまうが，不完全情報ゲーム用の工夫を施さなくても UCT は実装することができる．

以上の理由から局面の仮定とノードの追加，ノードの削除の三つの工夫を施す場合と施さない場合のどちらにも利点が存在する．そこで本研究では三つの工夫をそれぞれ施したり施さなかったりした 5 パターンの UCT アルゴリズムを用意し，また，比較のためにゲーム木探索をしない原始的なモンテカルロ法（以後，MC）と完全にランダムな手を選択するプレイヤーの二つを加えた計七つのアルゴリズム同士を戦わせることでポケモン対戦には UCT アルゴリズムをどのように使えばよいのかを検証した．表 2 に今回比較した手法と特徴をまとめた．UCT1 は工夫を全く取り入れておらず，UCT5 は工夫を全て取り入れている．

4. 実験結果と考察

実験は筆者が作成したシミュレーション上で行った．ポケモン対戦は使うポケモン自体も勝敗に影響するため，パーティの育成，ポケモンの選出に関しては各手法で同じものを採用した．パーティは九つの中から毎試合ランダムで選んだ．また，相手のポケモンのパラメータ予測はポケモンが公式に発表している web サイト[6]から取得した事前分布に従って技，特性，性格，持ち物について行った．一方で，最も重要な基礎ポイントの数値に関する事前分布は存在しておらず，筆者の知識に頼った予測をするしかなかった．

各アルゴリズムの思考時間は公式ルールに基づき 1 ターン 1 分とした．各アルゴリズムを総当りで 200 回ずつ対戦させた結果を表 3 にまとめた．

表 3 の各数字は第 1 列に書かれたアルゴリズムから見た第 1 行に書かれたアルゴリズムと対戦したときの勝率である．UCT に振られた 1 から 5 までの数字は本研究において区別のために割り振られた数字であり，数字が大きいものほど 3 章で触れた工夫を多く取り入れている．

表 3 より，MC から UCT5 までの六つは順当にランダムプレイヤーに勝利したことが分かる．ランダムプレイヤーに対する勝率が 100%ではなかった理由はポケモン対戦が不完全不確定情報ゲームだからである．不完全な情報があるため読みあいが発生した場合は運悪く読みを外すこともあり，また，不確定な情報があるため正しい選択をしても運次第では間違いとなる．それゆえ 100%の勝利ができないことはポケモン対戦のゲーム性が不完全不確定情報である証明にもなっている．また，ランダムプレイヤーに対する勝率が 50%でもないことからきちんと正しい手を選択できれば勝つ可能性が高くなることが示されたと言える．

表 3 の結果から UCT に割り振られた数字の大きい順，つまり，3 章で述べた工夫を多く取り入れた順に強かったため，他の不完全情報ゲームで使われているような UCT を不完全情報ゲームに適用するための工夫はポケモン対戦でも

表 3　性能比較実験の結果（勝率）

	MC	UCT1	UCT2	UCT3	UCT4	UCT5	random
MC	-	0.665	0.665	0.605	0.565	0.510	0.785
UCT1	0.335	-	0.455	0.460	0.375	0.340	0.710
UCT2	0.335	0.545	-	0.485	0.490	0.330	0.665
UCT3	0.395	0.540	0.515	-	0.465	0.445	0.765
UCT4	0.435	0.625	0.510	0.535	-	0.455	0.760
UCT5	0.490	0.660	0.670	0.555	0.545	-	0.860
random	0.215	0.290	0.345	0.235	0.240	0.140	-

表 4　追実験の結果（勝率）

プレイアウトの質の向上	UCT5 から見た対 MC の勝率
あり	0.490
なし	0.550

有効であるということができる．

一方で，表 3 からゲーム木探索をしない MC と全ての工夫を取り入れた UCT5 の実力に有意差がなかったことも読み取れる． UCT は MC の上位互換であるためこの現象は本来起こりえないことである． しかしながら，今回の実験ではゲーム木の接点を集約しているため，有効な合法手が異なる接点同士を一つにまとめてしまっている可能性がある．そのようなことが生じている場合，前回までの評価を使って探索する UCT は良い探索を行えない．それどころか乱数に依ってプレイされるプレイアウト部分よりも木探索部分が高確率で悪い手を選んでいたらプレイアウト部分が大半を占める MC が UCT に勝つ可能性が高くなる．よって，プレイアウトの質がどのような影響を施しているかを調べる追実験を実施した．

プレイアウトの質を高めると AI の実力も上昇することは広く知られているが，今回の七つの AI を用意した実験でもランダムプレイヤー以外のプレイアウトを行う AI に関しては筆者の知識を使って良さそうな手を選択しやすく改良している．MC と UCT5 の結果の考察から UCB による木探索部分よりもプレイアウト部分の方が良い手を選びやすくなっているのであればプレイアウト部分が多くを占める MC の方が強くなるのは不思議ではないと考えた．もしも，この仮説が正しかった場合はプレイアウトの質を落とすことで相対的に木探索部分の合法手選択がよくなれば今度は UCT5 の方が強くなるはずである．そこで，プレイアウトの質の向上を全く施さなかった MC と UCT5 の勝率を前の実験と同じ条件で計測した結果が表 4 である．

これを見るとプレイアウトの質の向上の有無によって MC と UCT5 の実力が逆転していることが分かる．今回行った実験では色々な要素が複合的に絡み合ったうえでの結果であるため断言することはできないが今回用意した UCT が MC に勝てなかったことは状態数の集約方法と UCT の探索方針が噛み合っていなかった可能性が高まった．

5. まとめと今後の課題

今回の実験で，ポケモン対戦に対して他の不完全情報ゲームに対する UCT の工夫は有効であるが，UCT 自体があまり有効でない可能性があることが分かり，ポケモン対戦は研究対象として非常に興味深いものであることが分かった．一方で，UCT をどのようにポケモン対戦に活用したらよいかを主眼に置いて実験を進めたため，MC と UCT の実力差に有意差がなかった理由などが明確にできなかった．今後は実験設定を注意深く考えて設定し，一つ一つ基本的なことから確認していきたい．また，ターンごとに不完全情報から完全情報になった情報を 100%活用した探索になっていなかったことも反省すべきであり，今後改善したい部分の一つである．

また，研究者全体でポケモン対戦を研究する運びになった場合に公式から詳細なルールが発表されておらず，シミュレータを一意に作れないなどフレームワークの整備に対する点も浮き彫りになったため，多くの研究者が参加できるような研究プラットフォームを築いていくことも必要だと感じた．

参考文献

[1] Auer, P., Cesa-Bianchi, N. and Fisher,: P. Finite-time Analysis of the Multi-armed Bandit Problem. Machine Learning, Vol. 47, pp.235-256 (2002).

[2] Kocsis, L. and Szepesvári, C.: Bandit Based Monte-Carlo Planning, Proceeding of the 15th European Conference on Machine Learning. pp.282-293 (2006).

[3] David, D., Huang, A. Maddison, CJ., et al.: Mastering the game of Go with deep neural networks and tree search. Nature, Vol.529 pp.484-489 (2016).

[4] Schäfer, J. Buro, M. and Hartmann, K.: The UCT algorithm applied to games with imperfect information. Diploma, Otto-Von-Guericke Univ. Magdeburg, Magdeburg, Germany, (2008).

[5] 三木理斗，三輪誠，近山隆, UCT 探索による不完全情報下の行動決定．ゲームプログラミングワークショップ 2009 論文集, 2009, 43‐50, (2009-11).

[6] ポケモングローバルリンク https://3ds.pokemon-gl.com/battle/oras/#single