

PBL 型ゲーム AI 作成授業とその成果 PBL type classes to Create A Game AI

神谷 涼斗[†] 小寺 智仁[†] 丹羽 拓実[†] 田島 孝治[†] 出口 利憲[†]
Ryoto Kamiya Tomohito Kodera Takumi Niwa Koji Tajima Toshinori Deguchi

1. はじめに

岐阜高専電気情報工学科では、第五学年に PBL 型の実験があり、研究室対抗のコンテストを毎年実施している。本稿では、このコンテストにおいて実施したゲーム AI の作成について、対象のオリジナルゲームを紹介するとともに、授業の実施によって得られた成果をまとめる。

コンテストの課題は電気電子コース (以下 E コース) と、情報工学コース (以下 J コース) で異なる。本稿では特に J コースのコンテストについて述べる。このコンテストは前期 (4月~9月) と後期 (10月~3月) で2度、同じテーマで開催され、前期は第五学年のみが参加できる。課題は4月に提示され、7月末にプレコンテスト、9月末に本コンテストを行う。後期は任意の学生同士がチームを組んで参加可能で、10月にプレ大会と、1月末に最終大会という形で実施している。

コンテストの課題は毎年異なる[1]が、プログラムをすることで解決し、スコアを2チームが対戦して競い合う競技という形は統一している。2012年以前は画像処理などコンテストも実施していたが、2013年以降は将棋やチェスのような対戦型のオリジナルゲームを解かせている。

2. 競技課題の設計

2.1 競技課題の設計方針

本コンテストの実施は、学生に次の点を学ばせることが目的である。

- (A) 既存でないゲームの特徴を分析し、適応するアルゴリズムを自ら思考する能力の育成
- (B) コンピュータネットワークにおけるプロトコルについての実践的な理解
- (C) グループでのソフトウェア開発による、ソフトウェアの共同開発に対する理解と、スケジューリングや役割分担の実践

以上の観点から、ゲームは既存のものは用いない。しかし、あまり特殊なルールのゲームを作ってしまうと、ルールの理解が困難になってしまう。そこで、ゲームジャンルとしては既存のボードゲームにあるものを採用し、駒やピースの種類を変更する、リソースやゲームバランスを変更するという形で実装してきた。

2013年度~2015年度は、将棋やチェスのような駒を動かして相手の駒を封じていくゲーム、2016年度~2017年度はブロックのような陣取りゲームを実施してきた。これらのゲームでは先読みが大切なものの、短期間で学生が制作するため、盤面の評価関数を作るだけで手いっぱいになっている印象があった。

近年の人工知能やビッグデータの解析などを踏まえると、ゲームを学習させ、数手先をお互いに読みながら、戦って

いくプログラムを作らせたい。このため 2018 年度以降は、新たな競技課題を、次の方針で作成することにした。

- (1) ランダム要素が入ることは避ける
- (2) いつでも相手を妨害するような指し手がある
- (3) 盤面評価は分かりやすいが、プレイヤーの状況により評価値が変わる
- (4) スコアを得る道筋が複数あり、戦略が分かりやすい

2.2 2018 年度版ゲームの開発とその課題

盤面の評価のわかりやすさを高めるために、ボードゲームのジャンルとしては、Worker placement (または Action drafting) を採用する。この種類のゲームでは、アクションを選択するためのボードを用意し、ここから自分の行いたいアクションに Worker を置いて実行する。アクションの実行によって資源を得て、これらの資源をまた別のアクションを通して、勝利点に変換していく。ただし、自分が実行したアクションを相手は選べない。また、ターンごとに利用できる Worker の数は定められており、交互にアクションを実行していくと、お互いに Worker が一つもない状況になる。ここでターンを終了し、すべての Worker を回収してから次のターンを始める。Worker を増やすアクションも用意され、Worker が多いほど手数が増えるので有利であるが、ターン後の支払いが増えるなどのペナルティが追加されているものもある。また、ターン数は有限で、決まったターン数の終了がゲーム終了となる。

このジャンルのゲームの特徴は、各アクションによって得られる資源量があらかじめ決まっていることである。資源の種類が複数あることは多いが、資源の種類ごとに重みを定めれば、アクションの価値を評価することは容易である。このため自分の行動選択が簡単になるだけでなく、相手が必要としている資源を事前に奪うなど先読みが重要となる。資源は最低でも、ゲームの得点である勝利点とワーカーへの支払いのための要素 (一般的にはお金や食料)、勝利点の材料となる要素 (建物を建てる資材など) の3種類は必要で、対戦相手がちょうど欲しいものを、先に奪うべきか、自分の欲しいものを取るべきかがトレードオフになっていると良い。

テーマは本質ではないが、ちょうど卒業研究を始めたばかりの5年生を対象とするため、農業や工場経営などではなく、研究活動をテーマとするゲームを開発した。

本ゲームのアクションボードを図1に示す。ゲームのアクションが徐々に増えると先読みを行うプログラムが作りにくいと考え、すべてのアクションはリソースさえ用意できれば、最初から選択可能とした。さらに、リソースの種類についてもあまり多いと難しいと考え、(1)実験や人件費に必要なお金、(2)実験やゼミにより得られる研究成果、(3)研究成果を発表や論文化することで得られる業績 (=勝利点) の3種類に絞った。

[†]岐阜工業高等専門学校 電気情報工学科・専攻科
National Institute of Technology (KOSEN) Gifu College

ターンは季節として表し、3年間とした。各季節で支払いが必要だが、半期ごとに業績が多いとボーナスとしてお金を追加で得られる。これにより、研究成果を貯めてから一気に業績にするか、少しずつでも良いので業績にするかを戦略的に選べる。一回のアクションで一気に研究成果を業績に変換すると効率が良いが、そのためには変換用の多くの成果が必要で、これを貯めている間はお金が少なくなりやすい。さらに、相手に変換用のアクションを先に取りられてしまうと効率が落ちるというジレンマを持たせた。一方で、盤面の評価は極めて簡単で、得たいリソースを最も多く増やす手段は明確である。

別の観点として、学生のプログラム作成負担が大きく増えるのを防ぐために、通信部分やプロトコルの決め方は例年の方法を参照し、特に単なる通信部分プログラムは昨年度のものを用いるようにした。これにより、研究室の先輩等に昨年度の基本的なプログラムを教えることもできるようになった。

2018 年度のコンテストは 9 月 27 日実施され、5 チーム中 4 チームがゲーム木を作り、相手の手を予想した上で、最適な手を打っていくアルゴリズムを制作した。この結果は、先読みをさせるプログラムを作りたいという目的に十分かなうものであった。一方で、ゲームバランスとして見てみると、やや問題があった。それは、研究成果の取得方法に起因するもので、実験は後にするほど多く成果が得られ、この差を利用して相手の業績を僅差で上回る手を後手であれば必ず打つことができるため、後手が必ず有利であった。優勝したチームはこのゲームバランスに気が付き、初めからワーカを増やすアクションを切り捨て、最初の 2 ターンはあえて負けることで、以後をすべて僅差で勝つことを目指していた。

2.3 2019 年度版ゲームへの更新

後手が有利である問題と、Worker を増やすことが強みになっていない問題を解消するために、2019 年度ではアクションの種類の見直しを行った。まずは選択肢を狭めるほうにゲーム設計を変更した。この結果(1)Worker を増やすアクションを廃止、(2)駒の種類を 1 種類減らし 2 種類に変更、(3)駒の種類によって得られる資源数が変わる仕様を廃止、(4)同じ種類のアクションに駒を置いた順によって得られる資源が変わる仕様を廃止とした。この結果ゲームが

なり単純化された。これでは結果の予想が簡単すぎるため、ここからアクションのバリエーションを増やすことにした。具体的には実験や調査によって得られる資源の種類を 2 種類に増やし、上限を設けることで、資源の取得と得点の取得のアクションに幅を持たせ、相手の妨害をするか、自分の欲しい資源を得るかを考えられるようにした。また、これにより、先手を取ることが有利になるように切り替えた。

さらに、戦略をより強力にするが相手にもその戦略がわかりやすくなってしまおうというデメリットを持つ「機材」という概念を導入し、「機材」があれば資源をより多く得られるが維持費が必要とすることで、ターン終了後に支払う必要のあるコストを切り替えた。

3. 学生が制作したゲーム AI

3.1 Minimax アルゴリズムの応用

高専本科第 5 学生の参加チームのほとんどが採用したアルゴリズムは Minimax により、自身のアクションを決定するものであった。研究室対抗戦で優勝したゲーム AI もこの方式を応用したものである。優勝チームの実装における工夫は、盤面の評価ではなく、最終的に得られる資源をもとに Minimax を適用したことである。簡単な実装では各アクションの強さを、アクションの効果によって増加する資源に重みをつけて定める。この方法で考えればすべてのアクションに強さは両プレイヤーで一致しているはずである。しかし、このゲームではその資源が有用か否かはプレイヤーの状況によるため、盤面における自分の最善手が、相手の最悪手になっていない。このため、順にアクションを実行し、最終的な得点や、相手チームより 1 点でも多くターンを終えるなど、ターンの目標を評価指標にして、Minimax を実装したことが、このゲーム AI を強くしている。

3.2 GA による評価関数決定法

GA によって AI の評価関数を更新するプログラムを作成した。

AI のアルゴリズムは Minimax を採用した。季節終了まで探索し、その時点でのリソースに対し評価関数を適用することで評価を行う。この評価関数の係数を GA によって決定することで AI を作成した。

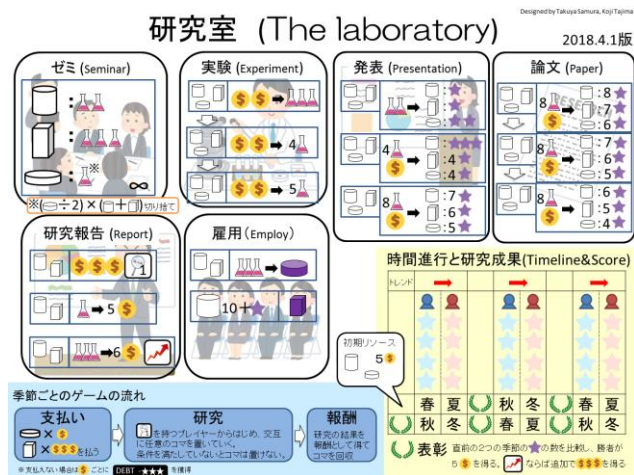


図 1 2018 年度版のゲームボード

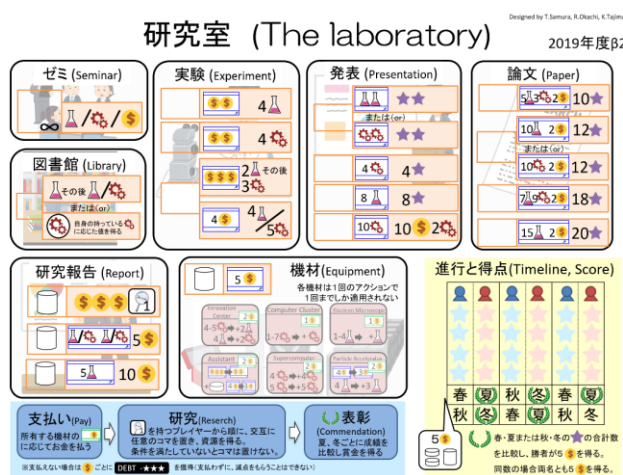


図 2 2019 年度版のゲームボード

本 AI では、自分のリソースと相手のリソースのそれぞれに対して評価関数を適用し、計算された自分の評価値から相手の評価値を引いたものを、その状況の評価値とした。評価関数 f は以下のようにになっている。

$$f = x_1F + x_2F^2 + x_3F^3 + x_4G + x_5G^2 + x_6G^3 \\ + x_7T + x_8T^2 + x_9T^3 + x_{10}S + x_{11}S^2 + x_{12}S^3 \\ + X_1(P)M + X_2(P)M^2 + X_3(P)M^3 \\ + X_4(Mc) + X_5(A)St$$

F : フラスコの数

G : ギアの数

T : トレンドの得点差

S : 総得点の得点差

P : 機材の支払いの金額

M : お金の数

Mc : 機材の所持状況

A : アシスタントの雇用状況

St : スタートプレイヤーマーカーの有無

評価関数の中の数値のうち、 x_1 から x_{12} はリソース等の状況によらず一定の値を持つ係数であり、 X_1 から X_5 は関数のようにリソース等の状況によって値が変化する係数である。 X_1 から X_3 は支払い、 X_4 は機材、 X_5 はアシスタントの状況によって変化する。 X_1 から X_3 はそれぞれ 11 パターン、 X_4 は 22 パターン、 X_5 は 3 パターンの状況に対応する数値を持つ。また、 x_1 から x_{12} までと X_1 から X_5 までのすべての数値を季節ごとに定めた。そのため、用意する数値は 70 個 \times 12 季節の 840 個となった。この 840 個の数値を GA により決定する。

次に、GA による係数の決定方法を説明する。840 個の遺伝子のまとまりを GA における 1 つの個体とした。まずランダムに 10 個の個体を生成する。この 10 個の個体の総当たりで対戦を行う。対戦は先手と後手を入れ替えて計 2 戦行う。対戦の結果、勝ち数の多い上位 3 個をエリートとする。このエリート 3 個と、エリート以外の 7 個の中からランダムに選ばれた 1 個を次の世代に引き継ぐ。次にエリート同士を交配し、新たな個体を 6 個生成する。エリート以外から選ばれた個体と交配により生成した個体には、突然変異が起こる可能性がある。5% の確率で突然変異する個体が抽選される。抽選された個体から、2.5% の確率で遺伝子を抽出し、ランダムな値に書き換える。このようにしてできた 10 個の個体を新たな世代として繰り返すことで評価関数の重みを更新する。

3.3 GA+NN 法

次にニューラルネットワーク(以下 NN)の結合荷重を遺伝的アルゴリズム(以下 GA)により定め、これを基にアクションを決めるものについて述べる。作成した NN のモデルを示す。作成した NN のモデルを図 3 に示す。階層型の NN となっており、入力層に 60 個、中間層 1 に 32 個、中間層 2 に 16 個、出力層に 32 個のニューロンで構成されている。入力層の 60 個のニューロンの内訳は次の通りである。

- 自分と相手のお金 (12 個)
- 自分と相手のフラスコ (8 個)
- 自分と相手のギア (8 個)
- 自分と相手の機材 (12 個)
- 総得点差 (6 個)
- 現在のシーズンの得点差 (6 個)

- 現在までに自分が何手打ったか (7 個)
- 先手かどうか (1 個)

盤面の状況から得られる情報をできる限り入力層に入れ、NN 自身に必要な入力を判断させるようにした。また、出力層のニューロンの数は配置できるコマの場所の数と同じ数になっている。

NN の計算方法は前の階層のニューロンの値とお互いを繋ぐ結合荷重の積の総和に評価関数を通して計算しており、中間層の評価関数は step 関数を採用している。そのため、入力層と中間層は 1,0 の二値での出力となっており、出力層はニューロンの値と結合荷重の積の総和をそのまま出力している。そのターンに決定するアクションは出力層の値が最も大きいニューロンに対応するアクションを採用する。

NN の結合荷重の値は GA により決定している。結合荷重の総数は 2,944 個となっており、全てランダムな状態から学習をしている。結合荷重の値の範囲は -1 以上 1 以下となっており、その状態に陥ることによるメリットやデメリットを再現させるためこの範囲設定とした。GA の学習方法は以下の順で行った。

1. 50 個分の結合荷重をランダムに作成する
2. 各個体を先手にランダム 30 体と対戦
3. 各個体を後手にランダム 30 体と対戦
4. 総獲得スコアが多い順に個体をソート
5. 上位 35 体を次世代へ引き継ぎ
6. 5 以外から 4 体を次世代へ引き継ぎ
7. 5 で選ばれた者同士の結合荷重を組み合わせ、6 体生成して次世代へ引き継ぎ
8. 結合荷重が全て乱数の新しい個体を 5 体追加
9. 2 へ戻り繰り返す

また、手順 5 の段階で 35 体の引継ぎを行う際 5% の確率で個体を選び、その個体の結合荷重の 2.5% を新しい乱数に置き換えることにより最高得点への収束の時間が短くさせた。

4. 対戦による評価

4.1 相互対戦の結果

制作したゲーム AI を評価するため、5 種類のゲーム AI を用意し、総当たりによる対戦を行った。GA+NN 法によるゲーム AI は、同一のアルゴリズムで複数の AI を制作したところ、学習段階の乱数の影響で 3 種類の戦術ができたため、それぞれ別の AI として対戦させた。対戦は先手後手を入れ替えたもの、自分自身との対戦を含めて全部で 25 試合行った。順位は勝ち数が多いものから順番に付け、勝ち数が同じ場合は直接対決の結果を参照して付けた。

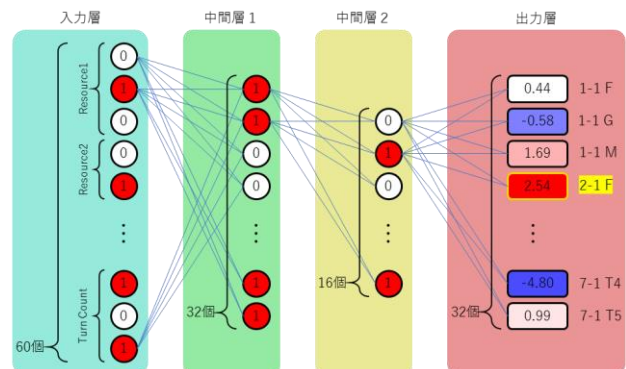


図 3 NN の構成図

対戦結果の順位表を表 1 に、各試合における得点一覧を図 4 に示す。試合は 25 試合中、D (先手) 対 A (後手) では、A のバグにより試合が成立せず、E (先手) 対 A (後手) では引き分けとなったが、それ以外の試合では勝敗が付いた。図 4 では勝ったほうの得点を太字にしている。

4.2 AI が持つ戦術の特性調査

相互対戦では、相手との相性によっても試合結果が変わるため、相手を「常に 1-1 (ゼミ) にコマを置き続けるもの」にして、AI が持つ戦術が妨害されないようにして対戦も行った。獲得した得点の一覧を表 2 に示す。

4.3 結果の考察

A の AI は、順位では 3 位であるものの、どの試合でも安定して 70 点以上を獲得しており、1 試合あたりの平均点は 84.4 点であった。これは、A よりも順位の高い E の 81.9 点よりも高い値であり、ここから相手が変わっても安定して高得点を取りにいけることが分かる。しかし、表 2 から分かる通り、相手に妨害されない場合の点数が 105 点と他よりも少し低く、理想としている戦術がまだ不十分であったことが考えられる。ここから、Minimax などの先読みは、戦術が十分であれば相手によらず安定した動作をすることができるが、その戦術は人間が考える必要があり、ここが不十分だと強い AI にならないことが分かる。

B の AI は、全体的に獲得点数が低く、ほとんどの試合で勝つことができなかった。理論上では、先読みができるため相手の妨害にも強く点数を伸ばすことができるはずであるが、そうならなかった理由として、係数学習方法の不十分さが挙げられる。本 AI は個体同士の相互対戦によって係数の学習を進めたが、世代が進むごとにエリートと似ている個体のみが残ってしまい、結果として自分自身には強い個体が出来上がってしまう。実際 B の得点を確認してみると、自分自身と対戦しているときに一番高い点数が出ている事が分かる。ここから、単純に自分自身との対戦を繰り返すだけではなく、相手の手に乱数を加えて様々な状況に対応できるように学習をさせると効果的であることが予想される。また、評価関数自体に関しても、各リソースの相関が反映されない形になってしまっているため、特定のリソースが多い時や少ない時に取りに行くと良いリソースを見逃しているという問題がある。

C、D、E の AI は、戦術の特性調査の結果から高い点数を取る戦術を見つけることができたと考えられる。この AI では、相互対戦では無く、ランダムとの対戦によって学習を進めたため、常に弱い個体との対戦を繰り返し、自分自身の点数の最大化ができたためであると言える。一方で、

		後手				
		A	B	C	D	E
先手	A	84 - 72	100 - 38	88 - 94	102 - 80	78 - 87
	B	59 - 86	78 - 64	65 - 90	55 - 117	69 - 60
	C	104 - 78	108 - 45	92 - 80	96 - 52	88 - 78
	D	不成立	105 - 73	107 - 108	65 - 40	99 - 90
	E	72 - 72	100 - 52	93 - 78	114 - 78	73 - 52

図 4 相互対戦時の得点一覧

表 1 相互対戦の順位表

AI	アルゴリズム	戦術	順位
A	Minimax 法	-	3
B	GA による 評価関数法	-	5
C	GA+NN 法	Flask・Gear 両方	1
D	GA+NN 法	Flask 中心	4
E	GA+NN 法	Gear 中心	2

表 2 AI が持つ戦術の特性調査の結果

AI	先手	後手	平均
A	100	110	105.0
B	94	86	90.0
C	114	114	114.0
D	125	99	112.0
E	114	114	114.0

先読みをせず、盤面評価に相手の手を考慮しないため妨害に弱いと言う欠点も見られた。特に自分自身と同じような戦略を取ろうとされると、自分の打ちたい場所と同じ場所が先に打たれてしまうことが多く、点数が伸びにくくなってしまふ。今回の総当りでは C が 1 位となったが、他の AI の戦術を考えると、たまたま C の戦術が他と被っていないため、C の点数が伸びたと考えられる。ここから、相手の妨害に対して対応できるように NN の入力に盤面の情報を加えるなどをすると、より安定して勝つことができる AI を作成できると考えられる。

5. おわりに

本稿では、岐阜高専電気情報工学科で実施しているオリジナルゲームのゲーム AI 開発をテーマとする PBL 教育について述べた。また、学生らが開発したゲーム AI について、簡単にまとめた。オリジナルゲームはワーカプレースメント型で研究室の運営をテーマとしており、ランダム性を排除すると共に、先読みにより強くなるように設計した。これに対して学生は自ら情報を収集し、Minimax を中心とするプログラムや、GA や NN を利用したゲーム AI などを開発した。これらのアルゴリズムは該当する学年へのプログラミング教育と比べても高度であり、自主的な学びを推進することができたといえる。また、ゲーム AI 同士の対戦により、それぞれの持つアルゴリズムの特徴も明らかにすることができた。今後はゲームを改良すると共に、ゲーム形式とアルゴリズムの関係性などを調査していく。

参考文献

- [1] 田島 孝治, 山田 博文, 出口 利憲, 安田 真, “電気情報工学科情報コースコンテストの変遷と成果”, 岐阜工業高等専門学校紀要, Vol.54, pp.26-29, (2019).