

# ボードゲームの戦略プログラミングを題材とした Java 演習支援 —間引き対戦の導入と提出戦略の詳細分析—

山田航平<sup>†1</sup> 富永浩之<sup>†1</sup>

問題解決型の応用プログラミングとして、ボードゲーム戦略を題材とする対戦形式での Java 演習を提案している。試行錯誤的なプログラミングを体験させ、持続的な戦略修正への動機付けを行う。過去の実践結果を踏まえ、支援環境 WinG の改良を行った。ローカル側では、各モジュールを統合し、各種サンプル素材も含めた、利用方法のモデルを構築した。サーバ側では、指標戦略と重付勝点度を導入し、大会運営を改善した。また、間引対戦によって対戦処理を効率化した。

## Programming Exercise for Problem Solving with Board-Game Strategy

### - Introduction to Thinning of Round-Robin Matching and Analysis of Submission Strategies -

KOHEI YAMADA<sup>†1</sup> HIROYUKI TOMINAGA<sup>†1</sup>

We have proposed an applied Java programming exercise by board-game strategy for problem solving learning. During implementation of hand method of Gogo game, students learn realization of idea as algorithm and revision with trial and error by execution result. We also have developed support system WinG, which consists of the local review package and the contest support server. The server maintains a preliminary and the final league, which decide students' score by the result of round-robin matching. We performed an educational practice in 2011. By introducing three kinds of dummy strategies as the standard of strength, the number of submission increased. We analyze the relation and tendency of the ranking in both leagues. We discuss efficiency of process in round-robin matching and approximate ranking method for immediately response.

## 1. はじめに

大学情報系では、C 言語などの入門的なプログラミング演習が必修とされる。続いて、応用的な演習では、C++/Java 言語などオブジェクト指向の導入、ソフトウェア開発手法の実践などが中心となる。また、データベースやネットワークなど、多様な情報系技術との融合も扱われる。さらに、特定分野の課題に対する問題解決の手段としてのプログラミングが重視される。

このような総合的な能力の向上を教育目標とするには、指示された通りの要件や仕様を満たすプログラムを作成するだけでは不十分である。むしろ、オープンプロブレムの課題を提示し、各自の創意工夫を取り入れる余地がなければならない。また、コンパイルに成功し、サンプルデータで動作すれば終わりではなく、試行錯誤しながら改良と修正を重ね、効率性や精度など品質の向上を図る過程こそが重要である。

そのためには、魅力的な題材と効果的な演習方法が求められる。しかし、学生の興味と程遠い題材では、具体的なイメージが湧かず、プログラミングの到達目標を描きにくい。また、成果物としてのソフトウェアへの愛着が得られない。そこで、知識情報処理の分野では、ゲーム戦略を題

材とする演習が試みられている。これには、競争型学習の要素も盛り込まれ、動機付けへの効果が期待される。

ゲーム戦略は、ルールが明確で可能な着手を実際に確認でき、題意を把握しやすい。背景知識を必要とせず、学生の関心と興味を惹き、取掛りを容易にする。個々の着手は明快でも、その組合せである戦略は多様である。1 つの定まった正解を求めるのではなく、より強い戦略を目指すというオープンプロブレムである。その過程で、何らかの基準による取捨選択や多くの試行錯誤を必要とし、発見的な問題解決の訓練となる。

## 2. 本演習の概要

### 2.1 ボードゲームと戦略プログラミング

以上の背景に基づき、本研究では、ボードゲーム戦略を題材とする対戦形式での Java 演習を提案している[1][2]。ボードゲームとしては、五目並べに石取りを加えた、二抜き連珠のルールを表 1 のように整備し、五五と名付けた(図 1)。五五は、石を取ることで局面が大きく変化する。連と取という 2 つの勝利条件があり(表 2)、それぞれに攻撃と防御の優先度が考えられ、初心者の段階でも戦略の個性が出やすい(図 2)。

問題設定として、Java 言語で作成したゲーム実行ライブラリを提示し、13×13 の盤面での五五の戦略を Java プロ

<sup>†1</sup> 香川大学  
Kagawa University

グラミングで実装させる。実行ライブラリのオブジェクト構成は、図3の通りである。学生は、Computer クラスを継承したサブクラスで、着手メソッド calc\_hand() をオーバーライドする。calc\_hand() は、局面を引数とし、次の1手を返す。局面は、State クラスのインスタンスで、盤面の石の配置や取った石の個数を保持している。

戦略の作成手順は、図4のように、戦略方針に従って、各枞の評価値を求め、最高点の位置を着手とする。評価値は、経験的に割り当てた値から、実戦を通して調整していく必要がある。また、局面パターンのより詳細な判別に基づいて精密化していく。学生には、プロトタイプのソースコードを提示し、最低限必要な処理をコメントで指示しておく。典型的な配置パターンの実装から始め、独自の局面分析に進んでいく。

対戦では、先手後手の1組で1試合とし、勝敗で勝点を付ける。1勝1敗では、取った石の数で優勢を決め、同数は引分とする。戦略の評価は、総当たり対戦での勝点の合計で順位を決める。ただし、全体の評価は、戦績だけでなく、戦略の自己評価を行った総括レポートも加味する。

## 2.2 授業実践の概要

本演習は、先行研究の段階として、2005年度から2009年度まで、本学科の3年次の演習科目で試行的に取り入れていた。その後、カリキュラムの改定により、2011年度から現行の形態による本格的な実践として再開された。実施する科目は、情報環境コースの3年次の必修科目「情報環境実験Ⅱ」である。後期に配置され、火曜 3・4 時限(13:00-16:10)の開講である。必要に応じて、5時限にも演習を延長する。授業期間は、前半と後半で、担当教員を変え、異なる内容を扱う。前半6回は、香川教員が担当で、Javaの応用プログラミングで、サブレットを利用したアプリケーションやシューティングゲームの改良を課題とする。後半9回は、著者の教員の担当で、個人課題のゲーム戦略プログラミングと、グループ課題のLEGOロボットの制御プログラミングを並行して進める。本演習は、前者のゲーム戦略プログラミングの一環である。

前提科目として、「オブジェクト指向言語」、「情報数学」、「知識工学」、「情報環境実験Ⅰ」が挙げられる。特に、「オブジェクト指向言語」および「情報環境実験Ⅱ」自体の前半でのJavaプログラミング、および「知識工学」での探索手法やヒューリスティック評価などが、本演習の内容に関連する。日程は、表3の通りである。ただし、実際には、当初の予定より、少しずつこんだ。授業中は、主にグループ課題のLEGO演習に充てており、本演習は30分程度の時間をとって、手短かに説明を行った。そのため、配布した実行環境にHTML形式のドキュメントを含め、五五のルール、実行環境やローカル支援環境の操作方法、大会の進め方などを自習してもらった。

## 2.3 支援環境 WinG

本演習の戦略作成を支援するため、支援環境 WinG を開発している(図5)。ローカル側 WinG-LA では、戦略のテスト実行を効率的に行うモジュールを提供し、戦略検討に用いる各種のサンプルを用意する。サーバ側 WinG-CS では、提出された戦略同士を対戦させる大会を運営し、ランキングや戦績を公開する。これにより、試行錯誤的なプログラミングを体験させ、持続的な戦略修正への動機付けを行う。

先行研究としての2005年度からの第1版のシステムを大幅に開発し直し、2011年度から第2版のシステムを運用している。その後も、毎年の授業実践の結果を踏まえ、改良を重ねている。本論では、特に、大会運営の方法の改善と、大会運営サーバの改良について述べる。

## 2.4 ローカル支援ツール WinG-LA

ローカル支援ツール WinG-LA は、学生の躓きを減らし、全体的な戦略のレベルアップを図るための支援を行う。WinG-LA およびゲームの実行ライブラリは、大会運営サーバから事前にダウンロードしておき、戦略検討の各種のサンプルと合わせて利用する。プログラミング自体は、既存のテキストエディタや、Eclipse などのプログラミング統合環境で行う。

WinG-LA は、4つのモジュールから構成される(図6)。対戦実行モジュールは、実装した戦略の確認のため、人間または他の戦略プログラムとの対戦を行う。局面生成モジュールは、着手のデバッグのため、任意の初期局面を生成する。戦譜再現モジュールは、参考とすべき戦譜からの対戦を鑑賞する。着手確認モジュールは、指定された戦略と局面に対して、次の一手のみ実行する[8][9]。

表1 五五のルール

着手	先手が黒石、後手が白石を使用し、交互に打つ 2個並んだ相手の石を両側から挟んで取れる 1手で複数の方向の2連を同時に取ることも可能 後から石間に置いて2連になったものは取れない
勝利条件	完全な五連を作るか、10個(5回)石を取ると勝ち 完全な五連とは、挟んで取られない五連のこと 長連は五と認められない
禁手	「三々」は、先手後手共に禁手である 石を取った後の「三々」は、禁手とならない

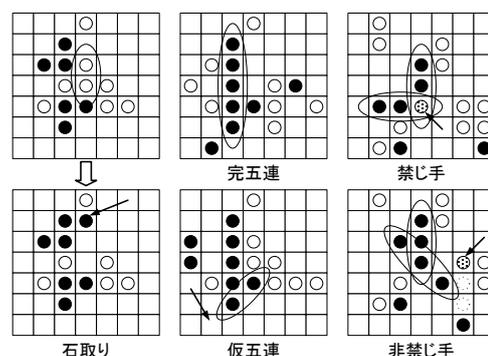


図1 五五のルールと局面

表 2 勝敗判定の手順

1. 置けない打手(盤外, 重置, 連打)は反則負け
2. 打って(石取り前に)三々になったら, 禁手で負け
3. 石を取っても, 相手の五連を崩せなければ負け
4. 相手の長連から石を取り, 五連ができれば負け
5. 自分で石を取り, 五組に達すれば勝ち
6. 相手に崩されない五連を作ったら勝ち
7. 全ての枡が埋まっていたら引分
8. 時間内に打たないときは投了で負け

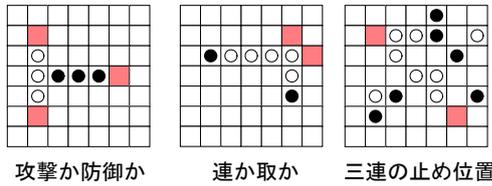


図 2 戦略の分岐点

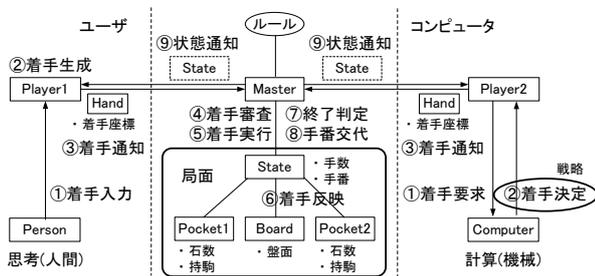


図 3 WinG の実行ライブラリ

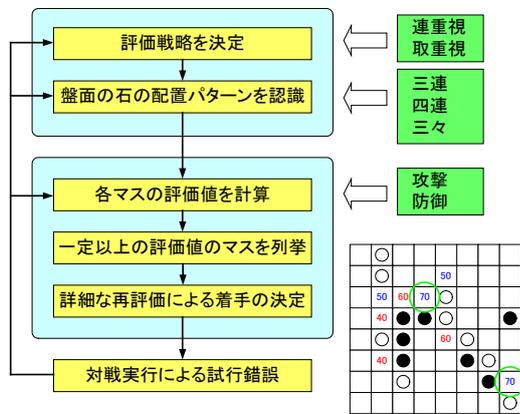


図 4 戦略プログラムの組立て方

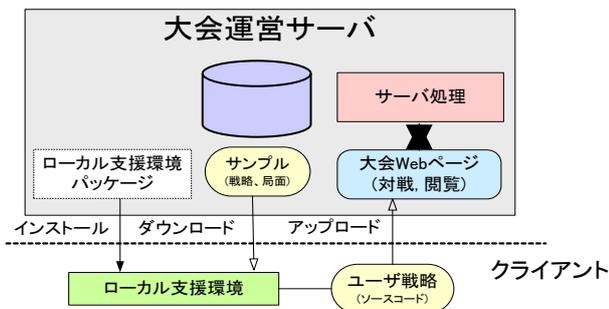


図 5 支援環境 WinG のシステム構成

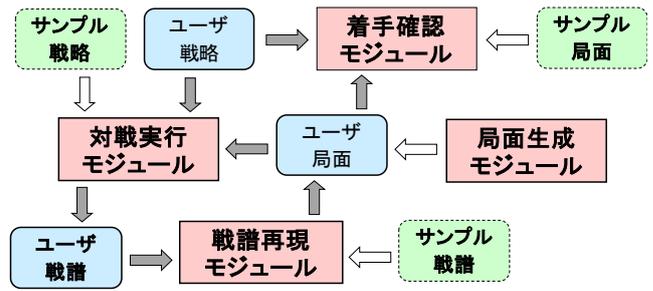


図 6 WinG-LA のモジュール構成

### 3. 予備大会と最終大会

#### 3.1 予備大会中の対戦方式

作成中の戦略にフィードバックをかけて, 持続的に演習に取り組ませるため, 最終大会の締切までを予備大会の期間とする(図 7). 期間中に提出された戦略は, サーバ上で他の戦略と対戦し, 定期的に結果が更新され, 順位が公開される. 順位の推移を見て, 自分の戦略を再検討し, 状況に応じて戦略を修正していく. 期間後に, 提出した戦略の強さを総合的に判断し, 最終大会の戦略を選択する. これらの戦略同士で総当たり戦を行う. この結果から最終順位を決定し, 成績に反映させる. このように, 自分の戦略を常に評価する機会を設けることで, 試行錯誤の繰返しを動機付ける.

#### 3.2 以前の授業実践における予備大会の問題点

2009 年度以前の試行的な授業実践では, 予備大会の序盤における受講者による戦略の提出が少ない傾向にあった. その理由としては, 大会序盤では, 受講者が提出した戦略は, 弱いものがほとんどで, 対戦相手も少ないため, 対戦結果を見ても, 強くなっているのかが分かりづらいからだと考えられる. また, 戦略の強さとしての合格基準が不明瞭だったため, どこまで取り組みればよいのかが受講者にイメージしづらかったことも上げられる.

#### 3.3 強さの基準となる指標戦略の導入

そこで, 2011 年度から, 戦略の強さの指標となる 3 段階の指標戦略を導入した. 戦略作成の指針となるような戦略を対戦相手にするすることで, 受講者の競争意欲を刺激し, 大会序盤からの積極的な戦略提出を促す. また, 演習の到達目標としての合格基準を明確にすることで, 予備大会を活性化させる. さらに, 最終大会において, 戦略の優劣によって順位が決定する相対評価だけでなく, 成績基準としての絶対評価にも用いる. 指標戦略のレベル設定は, 3 の通りである.

弱レベルは, 受講者に配布しているプロトタイプ戦略や, 過去の実践の下位戦略である. あえて実装が不十分なままにしてあり, 特定の局面で最善手を見逃すような振舞いをする. これらに勝つことが, 最初にクリアすべき最低目標

である。これらより順位が低いということは、独自の戦略が検討されていないか、プログラムのどこかにミスがあるということであり、デバッグの必要性を認識させることに繋がる。また、出足の遅い受講者が、他の受講者が提出した戦略に全く勝てないままだと、やる気を失う恐れがある。少し取り組めば勝てるような相手(噛ませ犬)を用意しておくことには意味がある。

中レベルは、プロトタイプ戦略にコメントとして示していた戦術を全て実装したものであり、これに勝つことは、課題としての合格基準の要求を満たすことに相当する。強レベルは、過去の演習で上位に入った戦略から選抜したものである。攻撃・防衛優先、連・取優先、戦略の強弱など、多彩なプログラムを用意する。プログラムの振舞から戦略を推測し、リバースエンジニアリングもさせる。これらに勝つことは、上位陣における最高目標である。また、最上位陣に対しては、超えるべき最終目標として、過去の授業実践で先輩が作成した歴代の王者である戦略を用意する。当初は、3段階のレベル分けだったが、より受講者のレベルに適した指標戦略を用意するため、2013年度からは5段階に細分化している。

### 3.4 勝点と勝率による順位の相違

2011年度までは、予備大会では勝率で、最終大会では勝点で順位を決めていた。勝点は、1組の対戦に対し、先手後手を入れ替えた2試合を行い、その勝敗で付ける。まず、2勝なら完勝4点、2敗なら完敗0点とする。1勝1敗のときは、取った石の数が多の方が僅勝3点、少ない方が僅敗1点、同数なら引分2点とする。最終大会は、40名程度の受講者が1人1つの戦略を選択し、総当り戦を実施し、勝点の合計で平等に順位が決まる。予備大会では、期間の最初と最後では提出された戦略数も異なり、加算方式の得点では優劣が決めにくい。そこで、単純に勝敗による勝率を採用していたが、完勝と僅勝の差がなくなり、最終大会の結果とややずれが生じた。最終大会の戦略に限っても、勝率と勝点による順位を比較すると、上位と下位の戦略では余り変化が見られないが、中位の戦略では順位が異なるものがあつた(図11)。これには、勝点では、中位の戦略の点差が明確につきにくいという要因も影響している。

### 3.5 単純勝点度 SWG の検討

2012年度に向けた改善案として、まず、勝点の合計を対戦数の4倍で割った単純勝点度 SWG に着目した。SWG は、区間 [0,1] の値であり、全試合で完勝ならば 1.0 となり、完敗ならば 0.0 となる。しかし、予備大会の性質上、戦略の全体的傾向による影響を受けやすい。例えば、予備大会の序盤は、弱い戦略が多く、それらばかりに勝って SWG が高くても、真の強さを反映していないと思われる。また、意欲的な少数の学生が多く戦略を提出するため、同じタイプの戦略に対する相性が強く出てしまう。場合によっては、ある戦略にわざと負けるような捨て戦略(キングメーカー)

を大量に用意し、勝たせたい戦略の SWG を高上げする可能性も考えられる。

### 3.6 重付勝点度 WWG の採用

そこで、このような見かけ上の強さや相性によるバイアスを減らすため、個々の勝点を対戦相手の勝点度で重み付けした重付勝点度 WWG を採用した。WWG は、各戦略のレーティングとしての意味を持つ。WWG は、区間 [0,1] の値であり、初期値 SWG から再帰的に計算する。戦略  $s$  の  $t$  回目の WWG の仮の値を  $w[t](s)$  とし、戦略  $s$  が相手  $k$  との対戦による勝点を  $p(s,k)$  および対戦数を  $n(s)$  とすると、 $w[t+1](s) = \sum \{w[t](k) \times P(s,k)\} / n(s) / \gamma$  で計算する。ここで、正規化係数  $\gamma$  は、 $\gamma = \sum \{w[t](k) \times 4\}$  とする。これは、全試合に完勝した場合の WWG である。この計算を、値の変化が閾値以下になるまで、あるいは順位が収束するまで行う。図8は、2012年度の最終大会の戦略の勝敗を用いて WWG を計算した際の、 $k$  回目と  $k+1$  回目の仮値をプロットしたグラフである。 $k=4$  ではほぼ  $Y=X$  の直線上に乗っている。これは、 $w[4](s) \approx w[5](s)$  を意味しており、4回程度の反復計算で値が収束すると言える。

### 3.7 重付勝点度 WWG の効果

WWG では、ノイズのような弱い戦略(雑魚)に幾ら勝っても、重みが低いため勝点度はそれほど上がらず、それらに対する勝敗の影響が軽減される。逆に、強い戦略に勝つと、金星のように大きな意味を持つ。また、結果として、中位の戦略でもどれに勝ったかで勝点度への寄与が異なり、戦略の優劣がより明確になる。図9は、SWG と WWG の相関を見たグラフである。丸で囲まれた部分は、SWG では同程度の強さと見なされていた戦略が、WWG によって強さの差が明確になっていることを表している。

表3 指標戦略の段階

段階	区分	内容
弱	1	演習開始時に配布したプロトタイプ戦略戦略をスケルトンコードとして記述
	2	過去の実践の下位戦略 プロトタイプ戦略の戦術の一部を実装 演習の最低目標
中	3	過去の実践の中位戦略 プロトタイプ戦略の戦術をほぼ実装 仮五連崩し、禁じ手などの実装 演習の到達目標
強	4	過去の実践の上位戦略 仮五連崩し、飛び連を高い精度で実装 局面の詳細な評価(四四、四三) 演習の最高目標
	5	過去の演習の優勝戦略 強戦略に順位で上回ること解禁 最上位陣の超えるべき最終目標

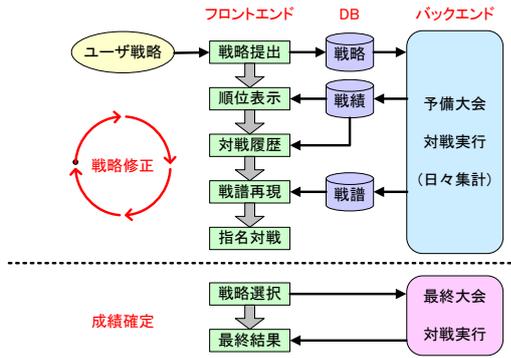


図7 予備大会と最終大会の運営

■ 順位

タイプレーヤ無し順位表

順位	勝率	戦勝	敗	分	戦名	作成者	登録日時			
1	0.625	16	10	4	2	40	Player05_02	Player05	2012.01.14	13:48:40
2	0.579	19	11	6	2	46	Player02_01	Player02	2012.01.14	13:35:34
3	0.562	16	9	6	1	36	Player02_02	Player02	2012.01.14	13:47:39
4	0.562	16	9	7	0	37	Player03_02	Player03	2012.01.14	13:47:56
5	0.562	16	9	6	1	38	Player01_04	Player01	2012.01.14	14:04:07
6	0.526	19	10	9	0	40	Player01_01	Player01	2012.01.14	13:34:30
7	0.500	16	8	7	1	31	Player04_02	Player04	2012.01.14	13:48:16
8	0.500	16	8	5	3	36	Player01_03	Player01	2012.01.14	14:03:54
9	0.438	16	7	8	1	29	Player01_02	Player01	2012.01.14	13:47:17
10	0.421	19	8	8	3	36	Player04_01	Player04	2012.01.14	13:36:12
11	0.421	19	8	11	0	34	Player05_01	Player05	2012.01.14	13:36:32
12	0.417	12	5	7	0	21	Dummy_02	Dummy	2012.01.14	14:07:51
13	0.417	12	5	6	1	22	Dummy_04	Dummy	2012.01.14	14:11:24
14	0.333	12	4	7	1	20	Dummy_05	Dummy	2012.01.14	14:11:35
15	0.333	12	4	7	1	22	Dummy_03	Dummy	2012.01.14	14:08:01
16	0.263	19	5	10	4	31	Player03_01	Player03	2012.01.14	13:35:54
17	0.250	12	3	7	2	18	Dummy_01	Dummy	2012.01.14	14:07:35

図11 大会運営サーバ WinG-CS の動作画面

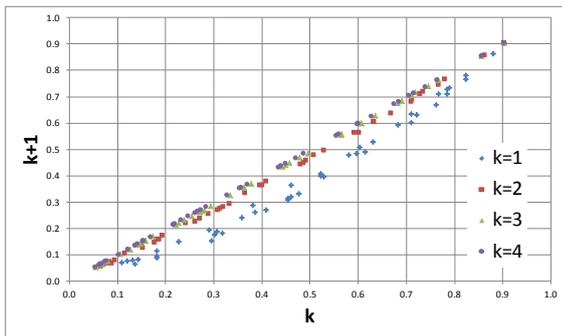


図8 WWG の計算の収束過程

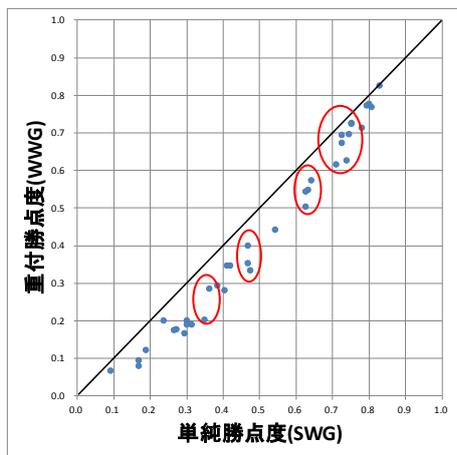


図9 最終戦略の SWG と WWG の相関性

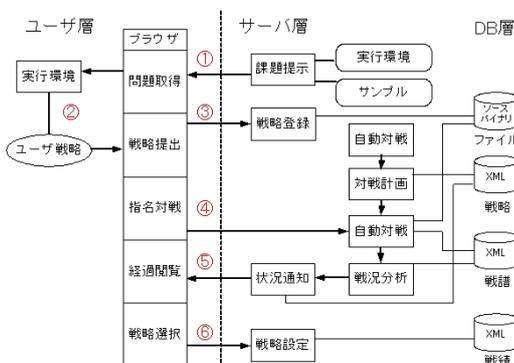


図10 大会運営サーバ WinG-CS のモジュール構成

## 4. 大会運営サーバ WinG-CS の改良

### 4.1 大会運営サーバ WinG-CS のモジュール構成

予備大会および最終大会を支援するため、大会運営サーバ WinG-CS を運用している(図 10)。システムは、ユーザ管理部、戦略提出部、戦略管理部、全体結果部、予備対戦部、最終対戦部の各モジュールから構成される[5][6]。戦略プログラムの対戦実行は JDK1.7、内部処理は Ruby1.9.2 で実装している。DB は、戦譜に XML、戦略と戦績の管理に SQL を用いている。

戦略提出ページでは、アップロードする戦略に、名前やコメントを付けられる。戦略が実行可能かどうかを学生に通知する。順位表示ページでは、全戦略の総当たり戦の結果を勝率順に表示する(図 11)。最強戦略による個人毎の順位も表示する。戦略管理ページでは、自分が提出した戦略について、戦績などを集約して表示する。対戦履歴ページでは、対戦ごとに、勝因や手数などを確認できる。指名対戦ページでは、任意の戦略を選んで対戦し、戦譜を再現できる。ただし、他人のソースコードを閲覧することはできない。最終結果ページでは、各自が選択した戦略で総当たり戦を実施し、勝点で順位表示する。

### 4.2 WinG-CS の内部処理

予備大会におけるシステムの内部処理を述べる(図 10)。戦略ソースコードが提出されると、サーバ内に保存する。ユーザ名から、パッケージ名を付け直し、ソースコードの該当部分を書き直す。次に、提出されたソースコードをコンパイルする。最後に、最低限のエラーチェックのため、弱いサンプル戦略との確認対戦を行う。実行時エラーが発生しなければ、提出を受け付ける。提出が受理された戦略は、その日の提出リストに加えられる。コンパイルか確認対戦でエラーが発生した場合、それを学生に通知する。すなわち、戦略のアップロード時には、受理のみとし、他の提出された戦略との対戦は行っていない。

教師が指定した時間になると、提出リストに入っている全ての戦略について、それまでに提出されている戦略との総当たり戦を実行する。デフォルトでは、1日ごとに提出リストを確定し、深夜に実施する。対戦結果は戦譜として保存し、各戦略の対戦履歴に登録する。提出リストの全戦略の対戦が終了した後、勝率を基準とした順位の再計算を行う。再計算された順位に基づいて、順位表などを更新する。

#### 4.3 対戦処理の実行時間

2012年度の予備大会では、これまでより戦略の提出数が大幅に増加した。全体で824個の提出があり、試合数は30万を超えた。1人当りの提出数は、平均19個であった。最大95個もの提出をする学生もいた。

2012年度のシステムでは、戦略のアップロード時には、受理のみとし、その場での対戦は行っていない。予備大会での定期対戦は、1分間隔で専用のスクリプトが起動し、新規に戦略の提出があれば定期対戦を実行する。既に提出されている実行可能な全ての戦略を対戦相手とし、その時点での総当たり戦を実現する。そのため、戦略数が増える大会終盤では、試合数が数万から数十万となり、相当の時間を要した。

予備大会の教育効果を高めるには、戦略のアップロード時に対戦結果の即時通知が必要である。予備大会の順位は、暫定的なものであり、その都度更新されるものである。すなわち、戦略の強さのおおよその指標となればよい。その結果を踏まえ、すぐに戦略の修正に着手し、再アップロードまでのサイクルが短い方がよい。

2012年度のシステムの予備対戦部の定期対戦の機能では、戦略の新規提出リストある戦略に対して、順次、対戦処理を実行する。定期対戦1回あたりの所要時間が長いと、最後に提出された戦略の結果が反映されるまでに何時間と掛かってしまう場合もある。したがって、完全な総当たり戦の実行を求めるより、戦略評価の判断にほぼ十分な程度の実行を実現する方が望ましい。

#### 4.4 定期対戦のトランザクション処理の変更

現行のWinG-CSでは、データベースにXMLを使用し、戦略や戦績の情報を構造化して管理している。しかし、戦略数が増加すると、それだけデータベースに書込む頻度は増え、書込まれたデータ量も膨大になる。そのため、大会中盤以降では、XMLの読み書きが著しく遅くなる問題がある。例えば、約300の試合を実行する定期対戦の処理に掛かる時間は、30分以上にもなっている。

そこで、この問題の影響を抑えるため、予備大会における定期対戦のトランザクションの変更を行った。従来の定期対戦では、対戦実行、戦績記録、順位計算の3つの大きな処理が1つのトランザクションとして扱われていた。そのため、各トランザクションで行われる処理が多く、戦略提出から対戦結果の反映までに時間を要する原因の1つとなっていた。また、各処理において、トランザクションの

原子性が確保されないため、何らかの問題で定期対戦が中断された場合に、データの整合性が失われることがあった。そこで、3つの処理を分離し、独立して動作させることで、対戦結果の通知を迅速化した。また、各処理において、適切に例外処理をすることによって、トランザクションの原子性を確保した。

#### 4.5 間引対戦の導入

重付勝点度による戦略評価の精密化を踏まえて、予備大会における対戦方式を、総当たり戦から間引対戦へ移行することを検討した。サーバの性能も向上させたため、2012年度の予備大会も、当初は総当たり対戦を行っていた。しかし、前年度を超える数の戦略が提出され、予備戦期間の終盤で、処理の遅延が発生したり、途中で対戦処理にトラブルが生じたりした。さらに、終盤は、特に提出過多になるため、結果の通知が遅いと、受講者のモチベーションの低下を招いてしまう可能性がある。また、予備大会の序盤に提出された不完全な戦略や、同一プレイヤーの似た戦略との対戦は、試合結果に悪影響を与える。そこで、迅速に定期対戦の結果を反映し、戦略修正の繰り返しを活性化させることと、不適切な戦略(雑魚、噛ませ犬、銅鉄)による不当な順位への影響を排除することを目的とし、間引対戦を導入する。間引対戦の方針は次の通りである。試合数は、基準となる試合数を決め、原則として、それを上回る試合を行う。また、予備大会の序盤で基準試合数に満たない場合は、通常の前半戦を実施する。さらに、中盤以降で、戦略数が増えた場合は、基準試合数で打ち切る。

- 各指標戦略を選出する方法
  - ・ 合格基準を達成したかの判断材料になる
- 各プレイヤーの選択戦略を選出する方法
  - ・ 最終大会と同等の状況で定期対戦を実施できる
  - ・ 意図的に弱い戦略を選択戦略にできてしまう
- 各プレイヤーの最高順位の戦略を選出する方法
  - ・ 選択戦略より強い戦略も選出できる
- 弱レベル以下の戦略を候補から除外
  - ・ ノイズとなる戦略との対戦を回避できる
- 順位表からランダムで選出する方法
  - ・ 全ての戦略と対戦する機会がある
  - ・ 同一プレイヤーの大量提出がある場合、対戦相手が偏る

#### 4.6 改良の結果

2013年度は、予備大会の終盤から、定期対戦に間引対戦を導入し、試験的に運営した。選出した対戦相手は、全指標戦略、全プレイヤーの選択戦略、全プレイヤーの最高順位の戦略である。基準試合数は100とし、それに満たない場合は、順位表からプレイヤーが重複しないようにランダムで選出した。図11は、予備大会における戦略数と定期対戦の平均実行時間の推移を表したグラフである。当初は、総当

り戦 300 試合で 45 分掛かっていた定期対戦が、間引対戦を導入することで、100 試合で 5 分～10 分程度で終わるようになった。これによって、戦略の提出から順位が反映されるまでの時間が大幅に高速化された。

また、戦略提出ページにて、戦略のアップロード時に、定期対戦の予想終了時間を通知するようにした。予想終了時間 T の計算は、直前の定期対戦の実行時間 S、実行待ちの提出戦略の待ち数、定期対戦スクリプトの実行間隔(cron 間隔)を用いて、 $T=(S+ \text{cron 間隔}) \times (\text{待ち数}+1)$  と計算する。

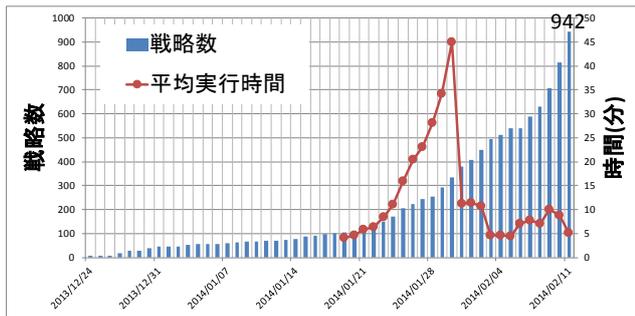


図 12 予備大会における戦略数と定期対戦時間の推移

表 4 各年度の実施状況

年度	2011	2012	2013	
期間(週)	5	8	5	
人数	35	44	37	
指標戦略 投入数	弱-1	2	1	1
	弱-2	2	1	1
	中-3	3	2	1
	強-4	1	1	1
	強-5	0	0	2
戦略 提出数	平均	8	19	25
	最大	46	95	150

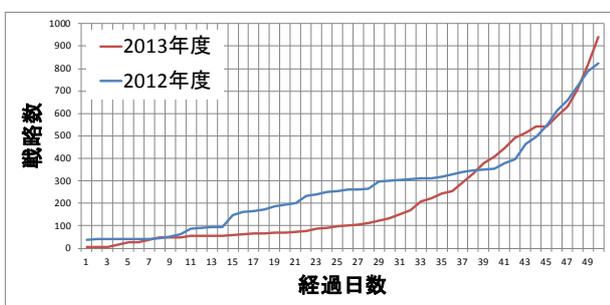


図 13 予備大会における戦略数の推移

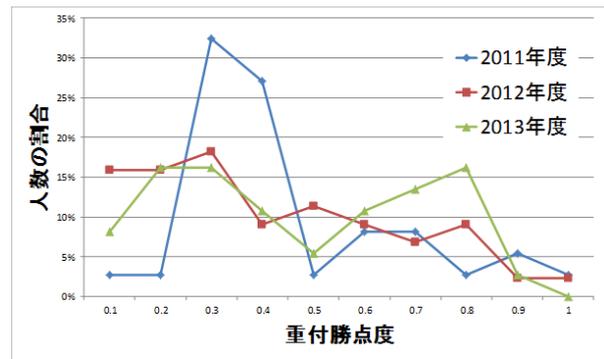


図 14 最終戦略の重付勝点度の頻度

## 5. 2013 年度の演習実践の結果分析

### 5.1 2013 年度の演習の概要

2013 年度の予備戦期間の提出状況は、図 11 の通りである。予備戦期間は約 5 週間で、指標戦略は、弱レベルを 2 つ、中レベルを 1 つ、強レベルを 1 つ、を大会序盤に投入した。さらに、大会終盤では、強レベルを上回る戦略が現れたため、強レベル 5 として、2011 年度と 2012 年度の優勝戦略を投入した。戦略の提出数は、全体で 942 個、1 人あたり約 25 個となり、去年の提出数をさらに超える結果となった。また、最大で 150 個近い戦略を提出する受講者もいた。図 13 は、2012 年度と 2013 年度の予備大会における戦略数の推移の比較である。2013 年度は、2012 年度と比べて、予備大会の中盤までは勢いが低調だった。大会終盤に突入すると、急激に提出数が増加し、最終的には、去年の提出数を上回った。

### 5.2 2011 年度から 2013 年度の実践結果

図 14 は、2011 年度から 2013 年度の 3 年間の演習実践にて、最終大会の最終戦略の重付勝点度の頻度を表したグラフである。2011 年度は、重付勝点度が 0.3, 0.4 前後の戦略が最も多く、中位から上位の戦略は少ない。積極的に上を目指そうという受講者は少なく、最低限のところまで演習を止めてしまっている。続いて、2012 年度は、重付勝点度が 0.1, 0.2 の戦略が増えたものの、全体的に下位の戦略数は減少し、中位に前後に達した戦略数が増加している。また、重付勝点度が 0.8 以上の上位の戦略も増加している。全体的に、下位から上位にかけて緩やかに人数が減少しており、演習実践の成果としては、レベルが均されてきたと言える。最後に、2013 年度は、2012 年度に比べると下位の戦略が減少し、上位の戦略が増加している。一方で、重付勝点度が 0.5 前後の戦略は減少しており、全体的に、強い戦略と弱い戦略の二極化が起こっている。これは、中位陣のレベルが底上げされて上位陣に食い込んできたためと考えられる。過去、2 年間と比較しても、下位は少なく、上位が多いため、演習実践の成果としては、良い傾向にあると言える。これを踏まえて、2014 年度は、下位を中位まで底上げし、

中位・上位をさらに増やすことを目標とする。また、表 5-10 は、2011 年度から 2013 年度の最終大会の戦略の総合順位である。2011 年度の優勝戦略が他年度と比べても圧倒的に強い。2013 年度は、圧倒的に強い戦略はないものの、それなりの強さの戦略が多い。一方で、中位付近の戦略は少なく、下位付近で少し固まっている。

## 6. まとめ

本研究では、知識情報処理の分野で、Java によるオブジェクト指向プログラミングの応用として、ボードゲーム戦略を題材とする問題解決型の演習を提案してきた。また、競争型学習のアプローチで、演習方法に競技系のコンテストを採用し、予備大会と最終大会を運営する。システムとしては、戦略作成を支援するローカル支援ツール WinG-LA と、大会運営サーバ WinG-CS を開発している。

2011 年度までの演習実践を踏まえ、強さの基準となる指標戦略を幾つか導入した。初期からの対戦相手として、予備大会に登録しておき、手頃な達成目標とした。これにより、予備大会で大幅に提出が増え、上位陣の競争意欲を刺激した。全体的な実装の度合いも高まり、教育効果も確認できた。

一方、対戦処理に時間がかかり、結果の早期の反映が困難となった。また、予備大会における弱い戦略との対戦がノイズとなる状況も生じた。そこで、安定的な強さの尺度として、対戦相手の強さで重み付けした勝点度を採用した。これにより、対戦数の違いや相性に影響されにくい、適切な順位付けが行えた。2012 年度の演習実践の結果では、これらの効果が確認できた。2013 年度では、間引対戦を導入し、十分かつ最小の試合数に抑え、対戦処理の効率性を大幅に向上させた。2013 年度については、最終大会の結果を分析中である。

## 参考文献

- 1) 尾崎浩和, 富永浩之: ボードゲームの戦略プログラミングを題材とした Java 演習支援 - 五五ゲームの実行環境と大会運営方法 -, 電子情報通信学会 技術研究報告, Vol.106, No.35, pp.55-60, (2006).
- 2) 尾崎浩和, 富永浩之, 林敏浩, 垂水浩幸: ボードゲーム戦略を題材とする問題解決型プログラミング演習支援 - 試行錯誤的な戦略作成の支援環境とサンプル提示 -, 教育システム情報学会研究報告, Vol.22, No.4, pp.69-74, (2007).
- 3) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材とした Java 演習支援 - 演習実践と対戦結果の分析 -, 電子情報通信学会 技術研究報告, Vol.111, No.141, pp.59-64 (2011).
- 4) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材とした Java 演習支援の実践状況, 平成 23 年度電気関係学会四国支部連合大会, pp.312 (2011).
- 5) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材とした Java 演習支援 - 対戦結果の順位分析と対戦方法の考察 -, 電子情報通信学会 技術研究報告, Vol.112, No.166, pp.29-34 (2012).
- 6) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題

材とした演習実践の結果分析と対戦方法の考察, 教育システム情報学会 第 37 回全国大会, pp.130-131 (2012).

7) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材とした Java 演習の大会運営サーバの効率化, 情報科学技術フォーラム, No.3, pp.627-628 (2012).

8) 山田航平, 富永浩之: 盤面ゲームの戦略を題材とするプログラミング演習支援システムにおける着手確認モジュール, ゲーム学会 第 10 回全国大会, pp.21-22 (2011).

9) 山田航平, 富永浩之: ボードゲームの戦略プログラミングを題材とした Java 演習支援 - 着手確認モジュールの導入と大会支援サーバの GUI の改良 -, 電子情報通信学会 技術研究報告, Vol.111, No.473, pp.19-24 (2011).