

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

Consideration of Matching Methods and an Educational Practice of Programming Exercise with Board Game Strategy

山田 航平, 富永 浩之
Kohei YAMADA, Hiroyuki TOMINAGA
香川大学工学部
Faculty of Engineering, Kagawa University
Email: s12g482@stmail.eng.kagawa-u.ac.jp

あらまし：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 演習を提案する。また、ローカル側とサーバ側の支援環境 WinG を開発した。サーバ側では、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。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。2011 年度の演習実践の結果分析を行い、それに基づいて今後の対戦方法を検討する。

キーワード：ボードゲーム戦略、プログラミング演習、演習実践、競争型学習

1. はじめに

大学情報系では、プログラミング演習が必須とされ、複数の科目にわたって体系的で相補的なカリキュラムが組まれている。入門的な演習では、C 言語の文法事項の習得、データ構造の操作、整列や検索などの基本算法の理解に重点を置く。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、特定分野の課題に対する問題解決手法としてのプログラミングが扱われる。

しかし、提示した題材が学生の興味と程遠いと、具体的なイメージが湧きにくく、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。

2. 盤面ゲーム戦略プログラミング

以上の背景により、先行研究⁽¹⁾では、盤面ゲーム五五の戦略を題材とした Java プログラミング演習を提案している。演習の問題設定は、ゲーム戦略を Java で実装することとし、総当り対戦による順位付けとレポートによる評価を行う。二抜き連珠をベースに五五ゲームのルールを整備し、ゲーム実行のための基本ライブラリを実装した。戦略作成の基本方針を提示し、評価関数による戦略実装のためのサンプルコードを配布した。演習の形態として中間大会と最終大会、予備戦期間を設け、演習支援のための支援環境 WinG(図 1)を開発した。本研究では、2005 年度からの授業実践の成果を踏まえ⁽²⁾、2011 年度の教育実践に向け、演習方法の改善や支援システムの改良に取り組む。

3. ローカル支援ツール WinG-LA

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、戦略作成のローカル支援パッケージ WinG-LA(図 2)を提供する。WinG-LA は、4 つのモジュールから構成される。WinG-LA の各モジュールと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、試行錯誤による戦略の検討や修正を支援する。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。

これら既存のモジュールに加え、新たに追加した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。実行結果について、適切な着手かどうか判定する。また、デバッグを効率よく行えるよう、複数の局面に対して一括して実行する。個々の結果を総合的に評価する。機能は、試行する戦略を指定する。複数の初期局面を一括して読み込む。各局面に対して、着手を期待する/しない枡を指定する。各局面に対して、次の一手のみを連続して実行する。期待する/しない着手と実際の着手を照合して結果を表示する。

処理手順は、5 つのフェーズからなる。戦略指定フェーズでは、予め対戦用のクラスを手動で書き換えておく。局面選択フェーズでは、石の配置パターン毎に分類された初期局面データを選択する。着手指定フェーズでは、ユーザが改良した戦略が置くと期待する/しない枡を、盤面に設定する(図 3)。着手実行フェーズでは、各局面に対して、実際にどのよ

うな一手を打つか試行する。結果表示フェーズでは、個々の照合結果と正答率を表示する。

演習には最終版の開発が間に合わなかったが、試作版を一部の学生に試用してもらった。より多くのテストケースが欲しい、基本的な局面パターンだけでなく、複合的な石の配置を含むサンプル局面も欲しい、学生でも着手を期待する/しない枱を設定した局面データを作りたい、実際に着手した枱の評価値が知りたい、などの意見を得た。

4. 大会運営サーバ WinG-CS

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みさせるため、最終大会までの期間を、常に中間大会であるかのように、予備戦期間と定める。予備戦期間中に、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。予備戦後に、提出した戦略の強さを総合的に判断し、大会戦用の戦略を選択する。選択された戦略同士で大会戦として総当り戦を行う。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。

2011年度の大会実施に向けて、WinG-CSを全面的にリビルドし、新サーバに移行した。システムを、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に整理した。Javaでの対戦実行はJDK1.7、内部処理はRuby1.9.2で実装し、DBはXMLを用いる。

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、全戦略の総当り戦の結果を勝率順に表示する(図4)。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り戦を実施し、勝点で順位表示する。

5. 2011年度の演習実践

2011年度は、情報環境コースの3年次の必修科目「情報環境実験Ⅱ」にて、開発したシステムを運用し、本演習を実践した。受講者は、35名である。予備戦期間は約5週間で、強さの指標となる3つのダミー戦略を予め加えた。上位陣は、強レベルの戦略を倒そうと、当初から継続的に提出があった。中位・下位陣は、締切間際の提出が多かったが、中レベルの戦略を目標としていた。ほぼ全ての学生が、弱レベルの戦略を上回った。

6. おわりに

プログラミングを手段とする問題解決能力の養成を教育目標とし、盤面ゲーム戦略を題材とするJava演習を提案し、支援環境WinGを開発している。2011年度の授業実践では、ダミー戦略を取り入れたため、提出数が大幅に増えた。上位陣では早期からの積極的な取り組みが目立った。

一方、総当り戦に相当の時間が必要となった。そこで、次年度に向けて、少ない対戦数で戦略の妥当な評価を行える対戦方法を検討している。まず、試合数が異なっても比較できるように、勝点度を導入する。初期の弱い戦略による見かけ上の強さをなくすため、対戦相手の勝点度で重み付けする。勝点度でグレード分けし、各グレードから対戦相手を絞って、暫定的な強さを求める。これらの即時通知により、フィードバックのサイクルを短縮し、演習の活性化を促進する。また、複数のサーバによる分散処理の導入を検討する。

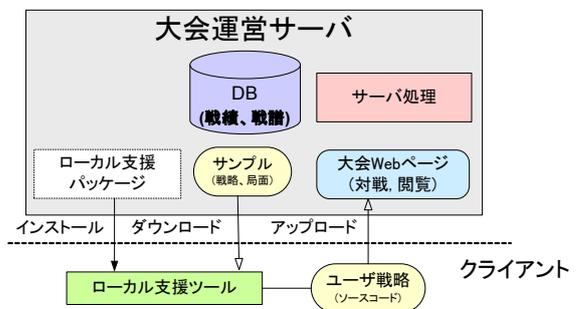


図1 支援環境 WinG の構成

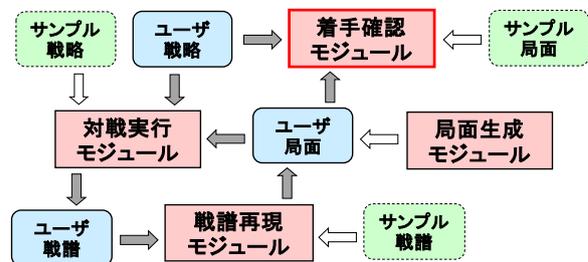


図2 ローカル支援パッケージ WinG-LA

7. 参考文献

参考文献

- (1) 尾崎浩和, 他: "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－", JSiSE 研究報告, Vol.22, No.4, pp.69-74, (2007).
- (2) 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－", 信学技報, Vol.111, No.141, pp.59-64, (2011).
- (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－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－", 信学技報, Vol.111, No.473, pp.19-24, (2012).