5ZC-09

ボードゲーム戦略を題材としたJava演習の予備大会における提出コードの個人進捗の分析手法

花川直己† 富永浩之†
香川大学† 香川大学†

1. はじめに

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[1]．2005 年度より，情報系学科 3 年生の必修科目において，課題の 1 つとして実践している．ゲームとして，五目並べに石取りを加えた五五を採用している．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，それぞれに攻撃と防御の優先度が考えられ，初心者でも戦略の個性が出やすい．

戦略の作成手順は，戦略方針に従って，各枡の評価値を求め，最高点の位置を着手とする(図 1)．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する．学生はまず，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．

2. 予備大会と最終大会

作成された戦略同士を対戦させる大会を運営するために，大会運営サーバ WinG-CS を開発している．WinG-CS は，提出された戦略同士を総当り的に対戦させ，戦績や順位を公開する．演習期間中は，予備大会として，対戦結果を参考に，何度でも戦略を提出できる．締切時に各自の最強戦略で総当りの最終大会を実施する．これにより，試行錯誤的なプログラミングを通して，持続的な戦略修正を動機付ける．

2011 年度から，新サーバに移行した．強さの指針として，3 段階の指標戦略を導入し，具体的な目標を提示した．2012 年度からは，レーティングとして，重付勝点度(WWG)を採用し，対戦相手の強弱を戦績に反映させた．2013 年度からは，WWG をレーティングに用いて間引対戦を取り入れ，対戦結果の反映を迅速に行った[2].

Analysis Methods for Personal Situation of Submission Codes in a Preliminary Contest of Java Programming Exercise with Board-Game Strategy
†Naoki HANAKAWA, Kagawa University
†Hiroyuki TOMINAGA, Kagawa University

3. 最終大会の戦略のコード指標による分析

本提案の演習では，戦略同士の対戦による戦績をプログラムの評価として用いている．しかし，WWG の高い戦略が，必ずしも質の良いコードとは限らない．実行結果という外面的な評価だけでなく，モジュール設計やコードの書法など，内面的な品質も評価しなければならない．

そこで，コードの品質を定量的かつ自動的に評価する手法として，ソフトウェアメトリクスに着目する．ただし，戦略は，if-then 形式のプロダクションルールの実現に近く，一般的なプログラムとは異なった特徴を有している．また，オブジェクト指向の特性や実行ライブラリの影響もある．一方，ルールという共通仕様は固定され，得点や成績という外部評価の目的関数が明確で，それを目指したコードの品質や改良の方向性が分析しやすいという側面もある．

2013 年度の最終大会の各自の最強戦略を対象とし，4 つのコード指標について，WWG による順位に沿って分析した[3]．コードの行数については，200~1000 であり，例外的な奇異なコードもみられるが，WWG と正の相関が見られる．冗長性は，ZIP によるファイル圧縮率を指標とする．類似の処理が多いと，圧縮率が高い．多くが 15～21%のサイズ比に収まるが，WWG との相関は余り見られなかった．構造化の指標には，メソッド定義の個数を用いる．ピークは 5～10 個である．上位陣でもばらつきが多く，成績群による顕著な傾向の違いは見られない．モジュール構造の循環的複雑度は，分岐や反復の構文数に比例する．複雑度が高いと，修正や拡張が困難になる．上位陣は，コードの量が多い割に，複雑度が低く，保守しやすいコードが多い．

4. 予備大会の提出コードの個人進捗の分析

以上の分析からは，成績群とコード指標との明確な相関性は必ずしも得られていない．コード量に影響する部分も大きく，複数の指標の組合せが必要と考えられる．また，最終大会の最強戦略は，締切時において，各学生にとって最も WWG の高いプログラムであるが，最も品質の良いプログラムとは限らない．特に，上位陣は，機能を追加する段階と，その後にリファクタリ

ングを行う段階のサイクルで実装していると思われる．そこで，予備大会にも着目し，提出コードの個人進捗を追跡することも重要である．本論では，その分析手法について検討し，予備的な調査を実施した結果を議論する．

予備大会の期間中に各学生が提出した戦略は，その時点で既に提出されている他の戦略と対戦する．最初は総当りであるが，提出数が多くなると間引対戦となる．戦略の WWG の値は，他の戦略の提出によって変化する．ここでは，予備大会が終了した時点での対戦成績から得た WWG を基にする．また，指標としては，コードの行数を取り上げる．ただし，事前処理として，コメントや空行を削除しておく．

各学生に対し，提出された全てのコードを，横軸を行数，縦軸を WWG としたグラフにプロットし，提出順に追跡する．これを TDQ2(Trace in Diagram of Quantity and Quality)と呼ぶ(図 2)．この手法は，ポーカー戦略を題材とする C 演習にも適用できる[4]．横軸は，コード品質の内部評価として，冗長性や複雑度に関連するコードの量的な指標を用いる．縦軸は，結果出力の外部評価として，得点や勝率などの成績を用いる．

5. TDQ2 上の Q2V によるコード更新の類型化

TDQ2 では，ある提出と次とのプロット間のベクトル Q2V の向きに着目し，どのような更新を行ったかを類型化する(図 3)．右上は，コードの増加と得点の上昇が相関し，適切な修正である．逆方向の左下は，以前のコードに戻すロールバックである．真上は，コードの変更が少ないのに得点が上がっているので，パラメタの調整の成功である．逆の真下はその失敗である．真左は，出力結果を変えない内部の改善なので，リファクタリングとみなせる．逆の真右は，行数が増えても得点が上がらず，効果の薄い報われない変更である．左上は，コード量が減って得点が上がるという，理想的な改良である．先読みなど，新たなアイデアに基づく戦略の発見などである．逆の右下は，バグを含んだ誤った改悪といえる．右下の直後の左上は，これに気付いたロールバックと言える．

6. おわりに

ボードゲーム戦略を題材とする Java プログラミング演習において，学生の提出コードの分析を行っている．内部評価として，4 つのコード指標を取り入れた．最終大会では，WWG による成績とコード指標との相関性を分析した．予備大会では，各自の提出コードの状態を追跡し，個人進捗の分析を行った．指標と成績の組をプロットする TDQ2 を導入し，プロット間を結ぶベクトル Q2V に着目して，その向きでコード更新を類型化した．

このような類型化に対し，実際のコードの目視による確認を行った(図 4)．上位陣で提出数が多いケースについては，ほぼ推測が適切であった．特に，リファクタリングの過程が確認できた．今後の課題として，さらに分析を深め，精緻な結果を求めたい．

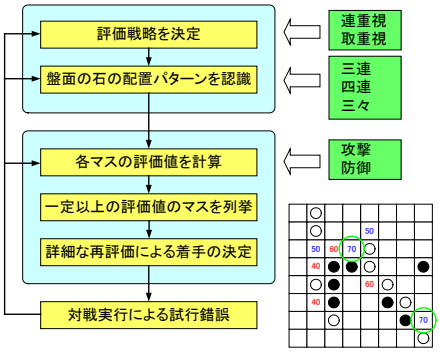

図 1 戦略プログラムの組立て方

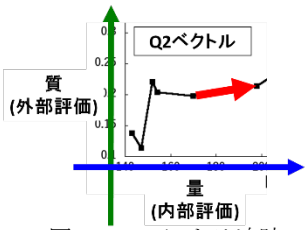

図 2 TDQ2 による追跡

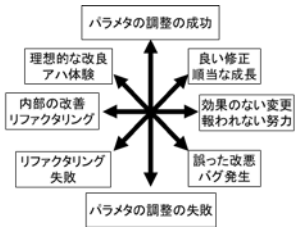

図 3 Q2V の類型化

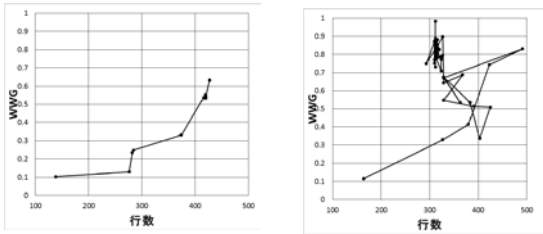

図 4 TDQ2 と Q2V による個人進捗の分析例

参考文献

1) 尾崎浩和, 富永浩之, 林敏浩, 山崎敏範: ボードゲームの戦略プログラミングを題材とした Java 演習の支援システムの開発, 情処研報, Vol.2006, No.108, pp.1-8 (2006).

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 - 指標戦略の導入と重み付き勝点度による結果分析 - , 教育システム情報学会 研究報告, Vol.28, No.2, pp.127-134 (2013).

3) 花川直己, 山田航平, 富永浩之: ボードゲーム戦略を題材としたプログラミング演習支援 - 最終大会の提出コードの特徴分析 - , 信学技報, Vol.114, No.121, pp.13-16 (2014).

4) 玄馬史也, 吉田亜未, 大川昌寛, 山田航平, 富永浩之: カードゲーム戦略を題材としたプログラミング演習支援 - 最終大会の提出コードの特徴分析 - , 信学技報, Vol.114, No.121, pp.17-22 (2014).