

ボードゲーム戦略を題材とした Java 演習の 大会中の提出コードの行数と勝点度による個人進捗の視覚化

花川直己^{†1} 富永浩之^{†1}

概要: ボードゲーム戦略を題材とする対戦形式の応用 Java プログラミング演習を実践している。提出された戦略同士を対戦させるラウンドロビン・トーナメントの大会を運営し、重付勝点度によるランキングを公開する。学生は、戦績に基づいて、戦略コードを改良し、提出を繰り返す。本研究では、提出コードの更新状況を視覚化し、個人進捗の状況を把握する。重付勝点度と行数を軸とする散布図に各コードをプロットし、その追跡ベクトルの方向を類型化する。本論では、2013年度の演習実践に適用し、修正の有効性やコード整理の傾向を考察する。

キーワード: ボードゲーム戦略、応用 Java プログラミング演習、大会運営と持続的なコード提出、対戦結果と重付勝点度、コード更新の視覚化と類型化

Visualization Methods of Personal Progress Situation with Line of Code and Winning Point of Strategy Codes in Java Programming Exercise with Board Game Strategy

NAOKI HANAKAWA^{†1} HIROYUKI Tominaga^{†1}

Abstract: We have proposed an applied Java programming exercise with board game strategy. During implementation of a 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 tournament support server. The server maintains a preliminary and the final league, which decide students' score by the result of round-robin matching. We try to realize visualization methods of code modification for grasping personal progress situation. We plot all codes in scatter diagram of the axis of WWG and LoC. We classify these vectors into 8 patterns by direction. In this paper, we discuss analysis in 2013 with our methods, and consider the tendency of effectiveness of modification or code modification.

Keyword: board game strategy, applied Java programming, pattern of code modification, personal progress situation

1. はじめに

本研究では、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]。ボードゲームとして採用している五五は、五目並べに石取りを加えたものである(図1)。盤面は、13×13とし、マス目の中に石を置く。五五の特徴として、石を取ることで局面が大きく変化する。連と取という2つの勝利条件があるため、それぞれに攻撃と防御の優先度が考えられる(図2)。そのため、初心者でも戦略の個性が出やすい。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。戦略の評価として、総当たり戦での勝点の合計で順位を決める。

本演習を実施するための支援環境として、戦略作成支援のスタンドアロンツール WinG-LA と、対戦実行の大会運営サーバ WinG-CS を開発している。2005年度より、情報系学科3年生の応用プログラミングの課題として取り入れている。2011年度からは必修科目となり、より本格的な課題として位置付けられた[2]。WinG も大幅に実装し直し、

その後も機能の改良を重ねている。

2. 本演習の概要

基本的な戦略プログラムの組立て方は、図3のようになる。戦略方針に従って、盤面の配置パターンに着目し、各マス目への評価値を設定する。これをプロダクションルールとして実装する。評価値が最高となるマス目を実際の着手とする。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。また、配置パターンも、より詳細な判別に基づいて精密化していく。

問題設定として、Java で作成したゲーム実行ライブラリを提供し、五五の戦略を Java で実装させる。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、局面と直前の相手の着手位置を引数とし、各マス目の評価値を求め、次の着手位置を Hand クラスのインスタンスとして返却する。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。学生には、プロトタイプソースコードを提示し、必要な最低限の処理をコメ

^{†1} 香川大学
Kagawa University

ントとして提示する。

課題の糸口として、まずは、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。先手後手の戦略の変更も必要である。他の戦略プログラムとの対戦結果の戦譜を再現しながらの検討、テストとしての特定の初期局面からの「次の一手」の確認も重要である。これらは、WinG-LA の機能として装備されている。さらには、再帰的な先読み、探索木の枝刈り、ランダムシミュレーションの実施など、より高度な戦略の導入も有り得る。

演習では、数週間の大会期間を設け、対戦形式の予備大会と最終大会を実施する。予備大会は、約5週間とし、その期間中は、戦略を何度も提出できる。提出された戦略は、WinG-CS 上で他の戦略との対戦を行う。戦績は、レーティングを考慮した勝敗である重付勝点度 WWG で提示され、その順位を競う。なお、総当りの対戦では、膨大な実行時間が必要となるため、現在は WWG に基づく間引対戦を導入している。

WinG-CS では、提出された他人の戦略コードは閲覧できないが、対戦結果の手数や勝因を確認したり、戦譜を再現することができる。また、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ている。これらの情報は、戦略修正のフィードバックに有効である。このように、自身の戦略を常に評価する機会を設けることで、試行錯誤の繰返しによる持続的な戦略修正を動機付ける。

予備大会の終了後、学生は提出した戦略のうち、最良と思う戦略を総合的に判断し、最終大会の戦略を選択する。最終大会では、その選択された戦略同士で総当り対戦を行い、成績に反映させる。また、最終大会の後には、戦略検討、概略設計、実装方法、実験結果の考察も記述した総括レポートも提出し、成績に加味する。

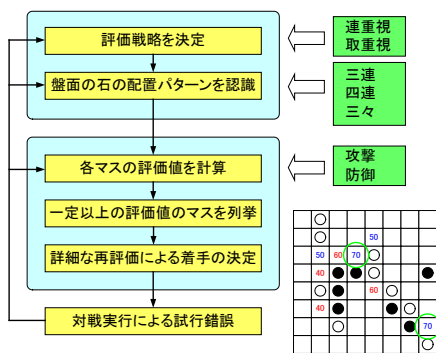


図1 戦略プログラムの組立て方

```

if ( 自分の四連が存在 ) then { 自分の五連を実現 }
if ( 自分が四取 ) then { 取を重視 }
if ( 相手の二連の石が存在 && 端に自分の石が存在 )
    then { 石取を実施 }

if ( 相手の五連が完成 ) then { 相手の五連を崩す }
if ( 相手の四連が完成 ) then { 相手の五連を阻止 }
    
```

図2 戦略のプロダクションルールの例

3. 演習実践における最終大会の得点状況

同様の演習課題として、カードゲーム戦略を題材とする得点形式でのCプログラミング演習も実施している[3]。採用しているトランプのカードゲームは、独り遊びとして、1つの山札で連続試行し、ベットによる賭けは行わず、得点だけを競うドロー式のポーカーである。戦略とは、5枚の手札から交換すべき不要な捨札を選ぶか、交換を打ち切って手役を確定させることである。ただし、大量の山札に対する実行結果の平均を得点とする。こちらの演習についても、スタンドアロンな実行環境と対戦実行の大会運営サーバ WinT を開発している。科目としては、先にカードゲーム戦略の方を約4週間で実施している。

両者は、予備大会での戦略コードの繰返しの提出、戦績や戦譜の結果公開など、共通点がある。カードゲームでは、山札ごとの捨札列が戦譜であり、戦績は手役の分布である。また、他人にコードを見せたり、良いアイデアを教えると、自分の順位が下がって不利になる可能性があるため、安易な剽窃を防ぐ側面がある。一方、手続型のC言語とオブジェクト指向型のJava言語、不完全情報で確率に左右されるカードゲームと完全情報で確定的なボードゲーム、得点形式と対戦形式などの相違があり、補完的である。

本研究では、これまでの演習実践で提出された戦略プログラムについて、教育効果の有効性や戦略の傾向を確認するため、様々な分析に取り組んでいる。まず、最終大会での各自の戦略プログラムを対象に、得点分布を図示し、年度ごとの傾向を分析した。これは、得点や勝敗など、実行結果の性能というプログラムの外部評価に基づくものである。また、学生の取組状況についても、提出された総括レポートから分析を行っている[4]。

次に、プログラムの内部評価としてコード品質にも焦点を当てた。これは、得点が悪くても、冗長で不適切な戦略コードが見受けられ、異なる観点からの評価も必要と判断したからである。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、ソフトウェアメトリクスの中でも、コードの静的な解析に用いるコード指標として、コード量、冗長性、構造化、複雑度の4分野に着目した。冗長性については、コルモゴロフ情報量に基づく圧縮比などを具体的な指標として採用した。これは、年度の異なる戦略コードの剽窃に対する類似度判定にも効果があった[5][6][7][8]。

ただし、一般的なソフトウェア工学における完成品に対するメトリクスを用いるのではなく、教育における適用を踏まえる必要があった。そこで、コード指標と得点との相関性に着目し、視覚化する方法を提案した。これはルールに沿った戦略の実現という仕様は同じであるが、性能を競うプログラムを対象にしているからである。また、不十分

な完成度のコードに対しても傾向を把握し、早期の警告や指導を与えたいためである。

以上から、各種のコード指標において、各コードの指標値と得点とをプロットした散布 SMP を提案した。なお、予備大会の間は、各時点での各自の最良な戦略コードに対して適用する。また、各コードに対する回帰的指標 RCM を導入し、その値が大きいとき、得点の割に効果の薄い異常なコードであると検出できるようにした[9]。

4. 演習実践における予備大会の進捗状況

予備大会においては、まず、教室全体での予備大会の提出推移を累積度数曲線として、年度ごとに比較した。これにより、システムの機能の追加と GUI の改良、大会運営の改善による意欲の向上を確認できた。特に、上級生などの聴講生によるオープン参加で、レベルの高い戦略が大会の早期から順位表に現れることが、競争意識を刺激することがアンケートなどから伺えた。また、他人との競争意識だけでなく、自分の得点推移の確認がプログラムの修正への大きな動機付けとなることも伺えた。

教師側でも、個人の進捗状況を把握することが重要である。本研究では、さらに、予備大会における学生の進捗状況を推測するアプローチとして、提出コードの更新傾向を視覚化する手法を検討している。そこで、コード品質の指標と実行性能の組をプロットする TDQP(Trace in Diagram of Quantity and Performance)を導入する[10][11]。図3のように、横の X 軸は、コード量に関する具体的な指標を用いる。縦の Y 軸は、カードゲームの得点やボードゲームの勝率などの成績を用いる。各学生に対し、予備大会で提出された全てのコードを、TDQP のグラフ上にプロットし、提出順に追跡する。

また、更新前後のプロット間を結ぶ差分ベクトル QPV(Vector by Quantity and Performance)にも着目する。図4のように、その向きでコード更新の特徴を8種に類型化する。これにより、効果的なコードの改良、バグの混入、ロールバック、パラメタ調整の成功と失敗、リファクタリングなどの傾向の可能性を把握しやすくなる。ただし、現時点では、目視による確認も必要である。

本論では、さらに、全ての QPV の始点を原点中心に集め、終点の分布状況を見やすくした QPV-SP も導入する。QPV の8種の類型化は、現時点では、成分のスケールに妥当な基準はなく、定性的な概念である。将来的には、QPV-SP における定量的で自動的な分類を検討している。

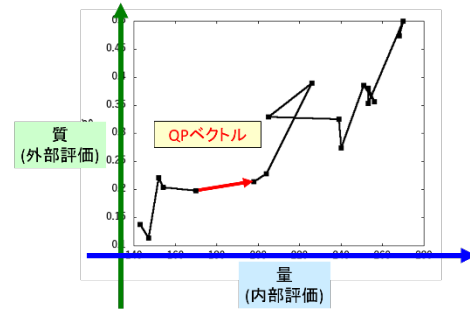


図3 TDQPによる追跡

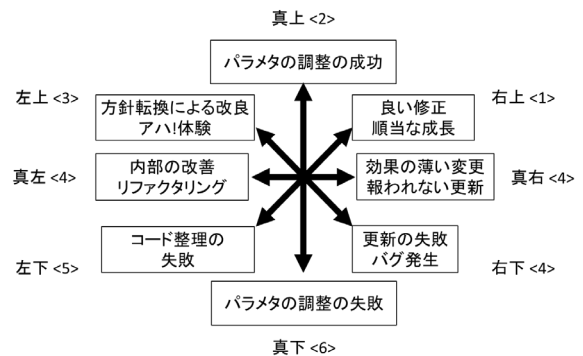


図4 QPVの8方向の類型化

5. 2013年度の最終大会の分析と考察

2011年度からの演習実践の概要について、表1、表2、表3にまとめる。2014年度については、サーバトラブルが発生し、大会の開始が遅れてしまった。しかし、開発環境は例年と同じ頃に提示しており、実質の開発期間は長い。提出件数の推移グラフは、図5の通りである。

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、図6のようにSMPを描いてみた。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。

本分析が対象とする戦略は、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

という側面もある。

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が重要とある。2013 年度は、提出数は 1 人あたり 25 件である(表 2)。これは、個人進捗の分析を行う上で十分であると考えられる。

	項目	2011	2012	2013	2014
演習実施	授業	必修	必修	必修	必修
	期間(週)	5	8	5	6
	人数	35	44	37	35
	リーグ	1	1	1	1
作成支援	サンプル戦略	○	○	○	○
	対戦実行	○	○	○	○
	局面生成	○	○	○	○
	戦譜再現	○	○	○	○
大会運営	着手確認	◎	○	○	○
	中間大会	-	-	-	-
	予備大会	○	○	○	○
	指標戦略	◎	○	○	○
	重付勝点度		◎	○	○
	間引対戦			◎	○

表 1 各年度の演習の実施状況

年度	2011	2012	2013	2014
期間(週)	5	8	5	6
受講生	35	44	37	35
総提出数	213 件	824 件	942 件	377 件
平均提出数	6 件/人	18 件/人	25 件/人	11 件/人
最大	32 件	67 件	142 件	32 件

表 2 各年度の予備大会の提出状況

分類	項目	2011	2012	2013	2014
実施	受講者	35	44	37	35
	対戦数	595	946	630	595
	試合数	1190	1892	1260	1190
勝者	黒対白	52-48	51-49	51-49	48-52
勝因	五連	55%	74%	68%	69%
	五取	36%	21%	27%	27%
	反則	9%	5%	5%	4%
手数	平均	37	34	33	36
戦況	完勝	66%	73%	66%	64%
	僅勝	26%	20%	23%	27%
	引分	8%	8%	11%	10%

表 2 各年度の最終大会の勝敗状況

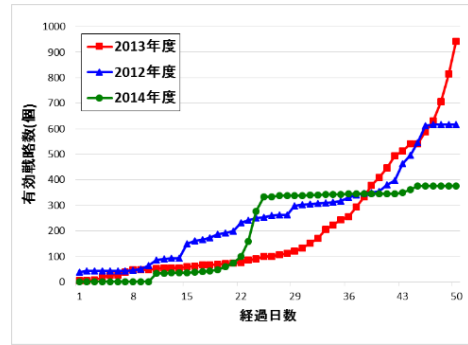


図 5 提出件数の推移グラフ

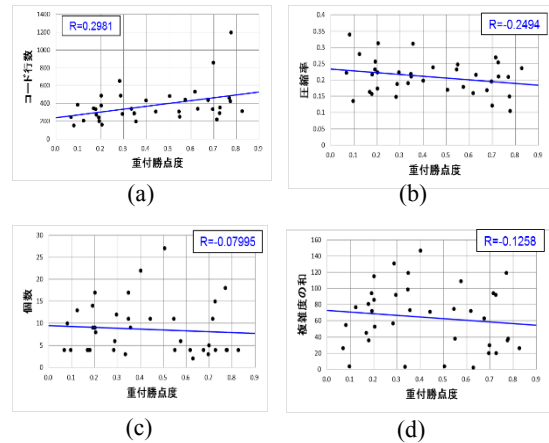


図 6 4つのコード指標による SMP

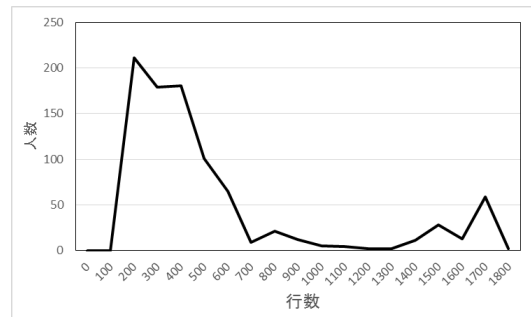


図 7 全戦略の行数の分布

6. 2013 年度の個人進捗の分析と考察

TDQP と QPV を用いて、2013 年度に対する実際の分析と考察について述べる。37 人の受講者に対して、5 週間の予備大会で 905 個の戦略コードの提出があった。全戦略の行数の分布は、図 7 のようになる。図 8 は、2013 年度の予備大会における学生の TDQP と QPV-SP である。TDQP においては、コード量としての X 軸に行数、実行性能として Y 軸に WWG を用いる。QPV-SP は、注記のない限り、X 軸は -400~+400、Y 軸は -0.4~+0.4 の範囲をとっている。

第 1 群については大きく右上方向に進む QPV が多く見られた。(a)(b)については、特に 100 行以内の追加による WWG の増加が多い。(c)については、大きな行数の追加は少ないが、垂直方向の QPV が多い。評価値を調整し、こま

めに提出を行なっているのではないかと考えられる。

第2群については、第1群に比べ点が全体的に水平方向へのQPVが多く見られた。(d)については、真下方向<6>へ大きく落ち込むQPVが見られたが、QPV-SPで見ると、行数の差が0であることから、評価値の調整に失敗したものであると考えられる。(e)については、少ないが右下方向<7>へのQPVが見られた。実際に目視にて確認をしたところ、飛石の判定など新しいパターンマッチングを導入していた。その直後の提出では、ミスをしている部分を除去し、点数を戻していた。

第3群の(f)は、余り集中しておらず、QPVの方向がバラバラである。しかし、QPV-SPにおいて、<7>方向のものが非常に少ないことから、堅実に実装を進めていると考えられる。(g)についてのみ、QPV-SPの行数は-600~600の範囲としている。左上方向<3>と右下方向<7>が交互に発生している。行数の増減と点数の増減が大きいため、実装とロールバックを繰り返しているのではないかと考えられる。実際に目視にて確認したところ、確かにロールバックと改良を繰り返していた。

第4群の(h)では、提出数は少ないが、堅実な実装を進めている。途中、大きく点数が下がる提出があるが、実際にコードを見たところ、反復処理の条件の記述ミスによるものであった。(i)は、途中で試行錯誤が見られる。(j)は、WWGの増加する方向へのQPVが少ない。実装方法がよくわかっていない可能性が考えられる、演習の期間中に指導する必要があると考えられる。

7. おわりに

ボードゲーム五五の戦略を題材とし、対戦形式のJavaプログラミングの演習を提案している。情報系学科の応用課題として、2011年度から授業で実施している。支援環境として、ローカル支援ツール WinG-LA と大会運営サーバを開発し、機能の改良を重ねている。

演習実践の結果分析として、最終大会については、得点分布やコード指標との相関性について分析してきた。予備大会については、教室全体の提出状況だけでなく、個人の進捗状況にも着目している。特に、コード更新の傾向を視覚化するTDQPと、更新前後の差分ベクトルQPVを紹介した。これらは、関連研究のカードゲームのポーカーの戦略を題材とする得点形式のCプログラミング演習と同様の手法を用いている。

本論では、これまでのカードゲーム戦略に対する分析手法を整理した上で、ボードゲーム戦略への適用における問題点を議論した。特に、TDQPにおけるQPVの分布状況を確認するため、QPV-SPを新たに導入した。

2013年度の演習実践においては、37人の受講者に対して、5週間の予備大会で905個の戦略コードの提出があっ

た。各自の戦略コードをTDQPにプロットし、QPV-SPも描いた。これらの図表を最終大会で成績4群に分けて傾向を考察し、実際のコードも目視で確認した。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。

今後の課題として、本演習における適切な指標について検討を行う。コード量については、Javaプログラムを行数で評価することができない。そこで、トークン数やJVMの中間言語を利用することが考えられる。また、行数の増減の差分ではなく、前回との変更があった行数の数を比較するなど考えられる。実行性能については、現在WWGは対戦毎に変動してしまう。そこで、指標戦略を準備し、これとの対戦結果による評価を採用することも考えられる。また、提出されたコードの目視での確認による、本手法の効果についても評価を行う。QPVの定量的な分析を行い、WinGのシステムへの統合を行う。

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] 吉田亜未, 大川昌寛, 玄馬史也, 富永浩之: ポーカー戦略を題材とする応用Cプログラミング演習の支援と実践, 教育システム情報学会 学生研究発表会 四国会場, Vol.2013, No.5, pp.9-10 (2014).
- [4] 花川直己, 富永浩之: ボードゲーム戦略を題材としたJavaプログラミング演習における学生の取組み状況とアンケート結果の分析, 情報処理学会 情報教育シンポジウム SSS2015 論文集, Vol.2015, pp.171-176 (2015).
- [5] 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).
- [6] 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).
- [7] 花川直己, 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の大会運営サーバにおけるコード品質の確認機能, 教育システム情報学会 第40回 全国大会, Vol.40, pp.251-252 (2015).
- [8] 花川直己, 富永浩之: ボードゲーム戦略を題材としたJava演習における提出コードのメトリクス分析, 情報処理学会 第77回 全国大会, pp.953-954 (2015).
- [9] 玄馬史也, 富永浩之: ポーカー戦略を題材とする応用Cプログラミング演習の支援と実践 - 大会運営サーバ WinT の提出状況とコード比較の機能の追加 -, 情処研報, Vol.2015-CE-128, No.9, pp.1-6 (2015).
- [10] 花川直己, 玄馬史也, 阿部隆幸, 富永浩之: カードゲーム戦略を題材とする応用Cプログラミング演習の支援と実践 - 大会中の提出コードの更新状況による個人進捗の分析手法 -, 信学技報, Vol.115, No.492, pp.51-58 (2016).
- [11] 花川直己, 富永浩之: 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, 情報処理学会 第78回 全国大会, Vol.78, No.4, pp.901-902 (2016).

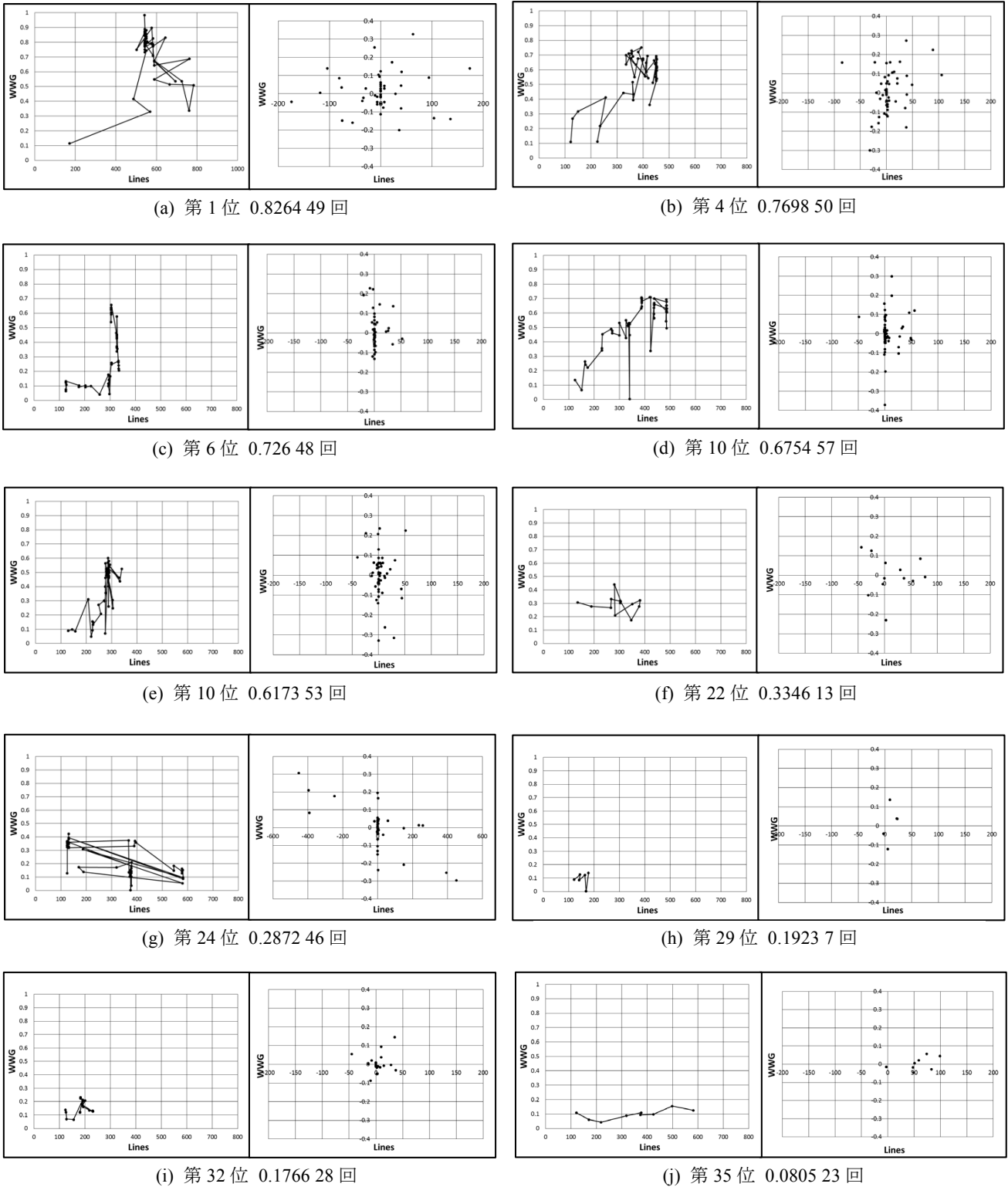


図8 成績群ごとのTDQPとQPV-SP