Vol.2015-CE-128 No.9
2015/2/14

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 ー大会運営サーバ WinT の提出状況とコード比較の機能の追加ー

玄馬史也[†1]　　富永浩之[†1]

ポーカー戦略を題材とする応用 C プログラミング演習を提案している．配布された手札から 1 枚ずつの交換を指定回数だけ繰り返し，手役を確定させる．実行環境を提供し，上記の戦略を実装させる．運営サーバ WinT を開発し，大会期間を設け，作成した戦略コードをアップロードさせる．受講者には，実行結果としての得点を通知し，順位を公開する．本研究では，コードの評価として，コードメトリクスと得点の相関性を考慮した回帰的指標 RCM を導入する．これらは，不適切な書法を含む，特異なコードの検出に有用である．2014 年度に向けた WinT の改良として，提出状況，および RCM と散布図 SMP を受講者に通知し，コードの修正へのフィードバックを促す．本論では，2014 年度の演習実践における提出状況と達成状況についても報告する．

Programming Exercise for Problem Solving with Card-Game Strategy -Functions of Submission Situation and Code Comparison in Contest Management Server-

FUMIYA GEMBA[†1]　HIROYUKI TOMINAGA[†1]

We have proposed an applied C programming exercise with a card-game strategy. An approach of our exercise adopts a simple Poker game for single player and a contest style in a classroom. We have offered an execution environment and also developed a contest management server WinT. It manages uploaded strategies of each student during a contest. It calculates a game score of each strategy by many executions with decks. It notifies a result and makes a ranking table public. The methods stimulate students' competitive volition and promote their motivation to revise a program in trial-and-errors. The final score of each student is the best score from his submitted strategies. For feature analysis of a code, we consider correlation between several code metrics and the score to introduce regressive code metrics RCM and the scatter plot SMP. They are useful to detect abnormal and irrelevant codes. In this paper, we discuss new functions for time-series view of submission and for visualization of code feature analysis. We also report results of an educational practice in 2014.

1. はじめに

近年，学習者の興味と意欲を高めるため，プログラミング実習の題材にパズルやゲームが取り上げられている[1]. 企業や団体が主催する情報系コンテストも盛んになってきており，競争意欲を刺激するため，学内のプログラミング演習をコンテスト形式で行うところも出てきている．本学科でも，2010 年度からのカリキュラムで，情報コースの 3 年次の必修科目で，ボードゲームの戦略を題材とする応用 Java 演習を実施している[2].

本研究では，カードゲームの戦略を題材とする問題解決型の応用 C 演習を提案している[3]. ポーカーを具体的な課題とし，ルールの設定，支援システムの開発，演習実践の結果を論じる．本提案も，上記の必修科目の演習内容の一部となっている．2014 年度は，大会運営サーバ WinT を改良し, 提出状況とコード比較の機能を追加した. 本論では，それらの機能について議論し，提出状況などの実践結果について報告する．

†1 香川大学
Kagawa University

2. ポーカー戦略のプログラミング演習

ポーカーは，手札の 5 枚で 9 種類の手役のうちの 1 つを作るハンドメイキング型のトランプゲームである．手役には，ペア系，フラッシュ系，ストレート系の 3 つの要素があり, これらの組合せで難度が変わる. 単純ドロー式では，手札から不要な 1 枚を捨て，山札からの 1 枚と交換していく．制限回数までに高い手役ができれば交換を打ち切る．本研究が提案する課題では，人間のプレイヤの代わりにポーカーを行う人工知能的なプログラムを作成する．ルールとしては，まず，手役の配点を与える．手役の配点は，ランダムに揃う確率に基づくのではなく，アルゴリズムの実装の難易を考慮して，麻雀のように倍々ベースとする．そのため，簡単なフラッシュと安全なフルハウスは低めの配点とし，ストレートを高くする．

十分にシャッフルされた 1 つの山札で，各プレイヤがポーカーゲームのテイクを繰り返し，作った手役の合計点を素点とする．トランプが 52 枚であることにより，交換の制限回数(チェンジ数)によって，可能なテイクの回数(テイク数)は異なる．ランダムな山札で 10000 回ゲームを繰り返し，その平均点を得点とする(図 1)．実現すべき戦略とは，各時

点の手札に対し，どの札を捨てるか，それとも交換を打ち切るかを決めることである．ここで，1 つの山札で既に使われた札も考慮する必要がある．残りの山札の確率的な傾向から，進行に応じた戦略のパラメタを調整させる．具体的には，図 2 のように，手札と場札の列，チェンジとテイクの数を引数とし，捨札を返す関数 strategy()を実装する．

C 言語の学習項目としては，カードの符号化と配列による手札の管理，数位と種類によるビンソートなどの整列算法，ストレートの待ち判定などのパターンマッチ，山札の乱数シミュレーション，再帰による先読みなどがある．文法的には C プログラミングの初級者でも挑戦できる．ただし，戦略のアイデアを実際のアルゴリズムで表現し，ソースコードとして記述するには，実践的なプログラミング経験が必要となる．また，補助関数によるコードの整理，十分なデータによるテスト，変数の値確認によるデバッギング，バージョン管理，非機能要件としての性能向上など，ソフトウェア開発手法の初歩も教育目標に含まれる．

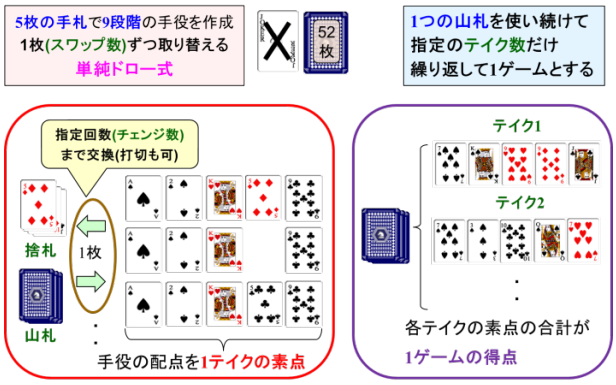

図 1 本演習のポーカーのルール

```
int strategy(int hd[], int fd[],     // 手札と場札
             int cg, int tk,         // チェンジとテイク
             int ud[], int us) {     // 以前の場札

  //----  本体処理
  // poker_point()は手役を判定するライブラリ関数
  // P3 以上の役ができていれば、手役を確定させる
  if ( poker_point(hd) >= P3 ) { return -1; }
  // テイク 1 と 2 では、2 回までしか交換しない
  if ( tk < 2 && cg >= 2 ) { return -1; }
  // テイク 3 と 4 では、4 回までしか交換しない
  if ( tk < 4 && cg >= 4 ) { return -1; }
  //----  返却処理
  return 0;     // 先頭のカード
```

図 2 関数 strategy()の実装サンプル

3. 大会運営サーバ WinT

個々の戦略プログラムは単独で実行されるが，受講者全体を 1 つのリーグと捉え，その中での得点を競わせる．その支援として，大会運営サーバ WinT を開発している[4]. WinT のシステム構成は，図 3 の通りである．現行版は，CentOS 6.5 の Linux マシンで動作し，Ruby 2.0 上の Rails 4.0 で開発している．GUI の実装の一部には，jQuery を用いている．データベースは，PostgreSQL 9.1 で構築している．

受講者は，共有フォルダから実行環境をダウンロードし，各自のローカル PC 上で戦略プログラムを作成する．WinT では，演習期間中，作成された戦略のソースコードの提出を受け付け，サーバ側でポーカーを実行する(図 4). 戦略コードは，提出時にサイズや様式などがチェックされ，提出 DB に保存される．gcc でコンパイルに成功すると，戦略 DB に保存される．不適なソースコードや実行エラーとなるバイナリがあれば，ユーザに警告が通知される．サーバ側では，予め 10000 個の山札リストを用意しておき，全ての戦略に対し，この山札リストを入力データとして用いる．

提出の反映として，得点や順位を個人および全員に公開する．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．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．締切時に，提出された各自の戦略のうち，最高の得点となる最良戦略を最終結果とし，成績に反映させる．

学生側サイトは，図 5 の通りである．ユーザ認証の後，個人のトップページであるマイページ(a)に進む．ここでは，戦績一覧やコンパイル失敗などの情報を閲覧する．提出ボタンを押すと，戦略提出フォーム(b)がモーダルとして表示される．ここでは，戦略に対する題目やコメントを付けて，戦略コードをアップロードする．サーバ内でのコンパイルと実行の後，得点が表示される．順位表示ページ(c)では，各学生の最良戦略の得点ランキングが表示される．戦略詳細ページ(d)では，各自が提出した戦略について，手役の割合などを集計して表示する．2014 年度からの新機能として，ソースコードの分析の情報も表示する．

教師側サイトでは，リーグ詳細ページで，提出状況を確認する．順位表示ページでは，個人の順位と全戦略の順位をタブの切替えで表示する．また，プレイヤ詳細ページでは，指定したプレイヤの得点の推移，各戦略のソースコードを閲覧する(図 6). リーグの設定や受講者の登録などの管理ページも用意されている．

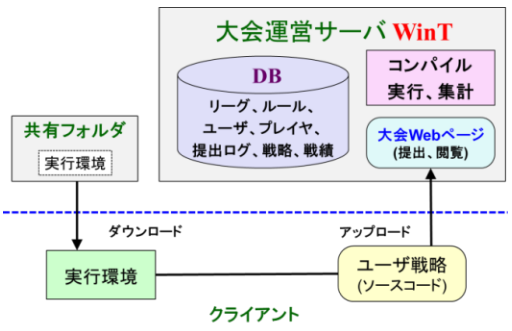

図 3 大会運営サーバ WinT のシステム構成

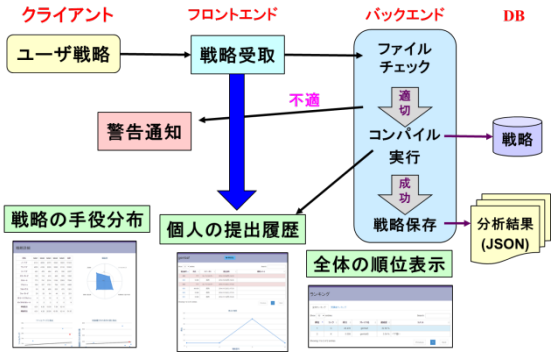

図 4 戦略提出の手順とサーバ側の処理

図 5 学生側の各ページ

図 6 教師側のプレイヤ詳細ページ

4. コードメトリクスに基づく回帰的指標

本演習では，戦略の平均得点を，プログラムの外部評価として用いている．しかし，得点が高い戦略が，必ずしも質の良いソースコードとは限らない．コーディング書法として，効率的でない，可読性が低い，構造化が不十分である，などの不適切なコードが存在することがある．そこで，コードの内部評価を定量的かつ自動的に行う手法として，ソフトウェアメトリクスに着目する[4][5]．

本研究では，4 つのコードメトリクス(CM)として，コード量に関してファイルサイズ，冗長性に関して制御構文の条件数と圧縮率，構造化に関して関数の定義数を採用している．分析の予備処理として，コメントは削除している．また，マクロなどプリプロセッサの影響を排除するため，プリコンパイルしたコードを用いる．圧縮率は，コードを ZIP で圧縮したファイルのサイズを，コードと提示したテンプレートを合わせたサイズで割った値である．相互情報量に基づくコルモゴロフ記述量として，冗長性を計測できる[6]．テンプレートを含めているのは，コードのサイズが小さいとき，圧縮率の値の妥当性が低くなるためである．

2013 年度の最終大会の戦略コードについて，成績で 4 群に分けてメトリクスの分布を調べ，傾向を分析した．その結果，コード量の割に得点が低い，if 構文の羅列で同様のコードの繰返しが多い，などの特異なコードが検出できた．

しかし，効果的な演習支援のためには，これらの傾向を大会中にフィードバックさせることが重要である．すなわち，戦略コードを提出したときに，即時に CM の分析を行い，何らかの指標を提示する．個々の CM の値については，実装しようとする戦略の質と量に大きく左右され，絶対的な評価は難しい．しかし，外部評価である得点との相関性をみれば，特異なコードの指摘に有用であると考えられる．

そこで，各 CM において，得点との散布図を作成し，回帰直線を求めて相関係数とともに図示したものを実装する．それを SMP とする．さらに，各点と回帰直線との距離を標準偏差で割って標準化した回帰的指標 RCM を導入する．ここで，標準偏差は，CM の単純な総和と二乗和から求め

たものである．回帰直線上にあれば，RCM の値は 0 となる．RCM の絶対値が大きいと，得点の割に「特異な」コードであると判断される．ただし，RCM の値そのものは，コードの良し悪しを必ずしも意味しない．現在は，上記の 4 つの CM を対象としているが，SMP と RCM は，他の CM に対しても有効な手法である．例えば，複雑に関する CM として，循環的複雑度などを採用することも考えられる．

5. 戦略詳細ページでの SMP と RCM の表示

本演習では，予備大会の期間中，大量の戦略コードが提出される．全ての戦略コードを教員が確認し，個別に指導をするのは現実的ではない．そこで，戦略コードを提出するごとに，各学生の戦略コードに対する分析結果を通知する．具体的には，戦略詳細ページにおいて，4 つの CM に対する SMP を提示する(図 7)．学生は，RCM の値および SMP における自分のコードの位置を把握し，特異なコードであるかどうかを認識する．図 7 では，丸印で囲まれたプロットが特異なコードである．また，4 つの RCM をまとめた星形チャートも表示する．ただし，RCM の値は正負にまたがり，値も理論的には限定されない．しかし，実際には，-50〜+50 の範囲に収まることを想定し，原点を-50，外枠を+50，中央の星形を 0 とする座標軸を設ける．

教師側での SMP と RCM の表示は，順位表示ページと戦略分析ページで行う．順位表示ページでは，各学生の最良戦略に対する RCM を表示する．ソート可能であり，特異なコードを数値的に見つけやすくなっている．戦略分析ページでは，受講側の SMP とほぼ同じグラフを表示する．ただし，各プロットをマウスオーバーすることで，どの受講生の最良戦略かが分かる．目視による定性的な評価で，特異なコードを確認し，個別の指導を行う．

SMP および RCM を導入した理由として，予備大会の序盤と終盤では，得点だけでなくコードの質と量が大きく異なることが挙げられる．最初の演習では，サーバの動作テストや学生の操作の確認もあり，ほとんどテンプレートのままの戦略コードが提出される．この状態で，CM を求めても意味がない．そのような数値の提示は，かえって学生の混乱を招く．また，上位陣の学生は，コードの提出を繰り返すが，得点を高める修正と，質を高めるコードへの改良と，2 つの側面が考えられる．以上の点から，これらの学生への通知は，予備大会がある程度経過した後，教師の判断で始めるものとする．

また，各学生の最良戦略のみで SMP を作成し，RCM を計算する．したがって，誰かの戦略コードが提出されるたびに，SMP と各コードの RCM は再計算が必要であるが，実際には戦略詳細ページにアクセスされた時点で再計算する．また，シミュレーションや再帰探索など高度な戦略を含むコードの実行には時間がかかるため，サーバ内では，Sidekiq3.2.5 を使って，バックグラウンドジョブとして処理している．このため，各自の RCM の反映には若干の遅延が発生する可能性がある．しかし，この点は，データベースの結果整合性(EC)として許容されると考えている．

その他に追加した機能として，自分との比較を取り入れた．対戦ではないので，提出の時期によっては，順位表示が効果的でない．具体的には，自身の得点の推移グラフを表示することで，学習意欲を高めさせる．また，受講側の順位表示ページでは，達成度によってランクを表示する．ランクは，下から X，C，B，A，S，SS，SSS と分けられており，自分の中での短期的な目標として用いる．

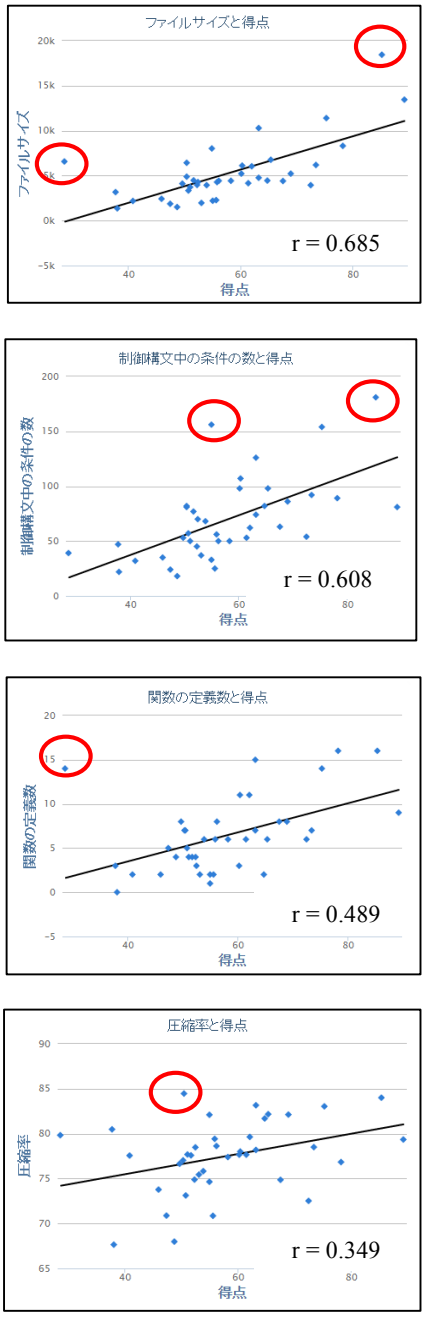

図 7 4 つの CM に対する SMP

6. 演習実践の課題設定と結果の概要

2010 年度からの新カリキュラムで，本学科の情報環境コースの 3 年次後期に必修科目「情報環境実験Ⅱ」が開講された．その課題の 1 つとして，ポーカー戦略の C プログラミング演習を実施している(図 8)．毎年の受講者は 40 名程度で，期間は 4 週間前後である．2010 年度は，2 週目に中間大会として暫定的に完成した戦略を提出させた．その得点と順位を公開し，それを踏まえて，3 週目に最終大会を行った．2011 年度からは，大会運営サーバ WinT を本格的に運用し，最終大会に至るまでの約 3 週間，継続的に戦略を提出させるようにした．ただし，2012 年度は，システムの障害が発生し，大会期間は 1 週間のみであった．

また，各年度で，ゲームのレギュレーションとして，チェンジ数とテイク数を少しずつ変えて実施している．2013 年度からは，テイクごとに掛率を変えるように拡張した．これは，2012 年度までは，残りの山札によって戦略を切り換えるものが少なかったからである．ただし，後半で山札が不足し，ゲームが打切りとなる場合が生じる．残りの山札の枚数で，テイクごとに戦略のパラメタを調整する必要がある．2014 年度は， 7 チェンジの 5 テイクとした．傾斜掛率は，図 8 の付表のようにした．2013 年度は，終盤のテイクの掛率が高すぎ，序盤を「捨てゲーム」にする戦略が見られたため，2014 年度は中盤を高くした．

2014 年度は 4 週間の演習とした．受講者は 36 名で，聴講者も 3 名が参加した．大会中は，全体で 1498 件，受講者からは 1453 件の提出があった(図 9)．各自の平均は 38 件，最高は 187 件であった．前年度までに比べ，大幅に増加した．例外もあるが，得点の上位陣は提出回数が多く，下位陣は少なめであった．2013 年度までは，毎週の授業の前後に提出が増え，締切直前も大幅に増えていた．2014 年度は，最初にシステムの動作テストを兼ねて，全員に提出させた．その後，他の課題と重なる時期は伸び悩んだが，3 週目からは，コンスタントに提出数が増えていった．

		2010	2011	2012	2013	2014
ルール	チェンジ テイク	5-5	7-4	8-4	6-6	7-5
	最大使用	50	48	52	52(66)	52(60)
	傾斜掛率	×	×	×	○	○
大会	演習期間	3 週	3 週	3 週	5 週	4 週
	大会期間	—	3 週	1 週	5 週	4 週
	中間提出	○	○	○	×	×
参加者	受講者	35	35	43	37	36
	聴講者	—	—	5	4	3
予備大会 戦略数	受講者	—	136	446	773	1453
	聴講者	—	—	119	89	45
	計	—	136	565	862	1498

テイクごとの掛率		1	2	3	4	5	6
付表	2013	1.0	1.0	1.5	1.5	2.0	2.0
	2014	1.0	1.0	1.5	1.5	1.0	—

図 8 ポーカー戦略演習の各年度の概要

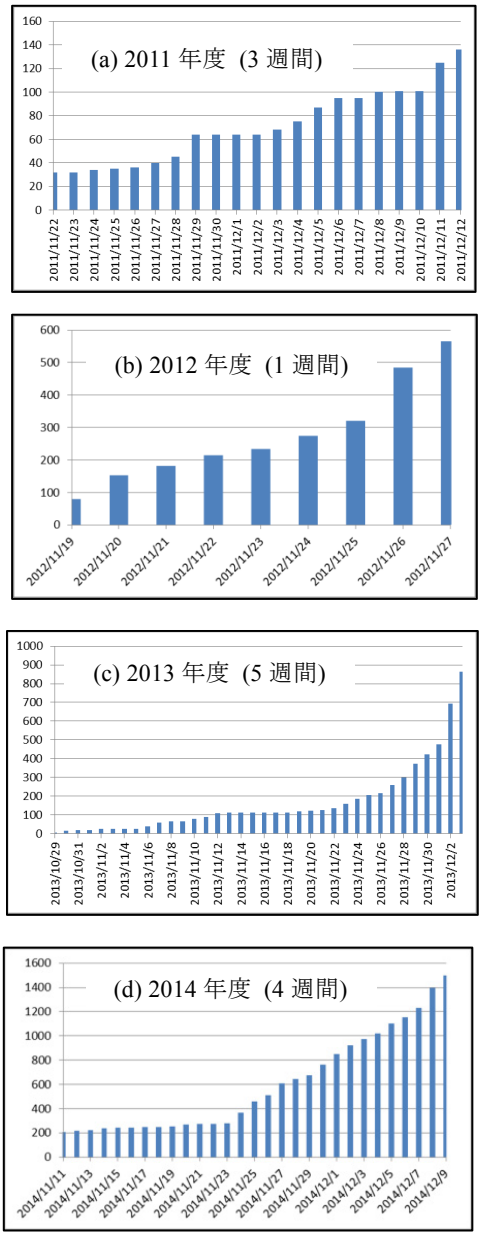

図 9 各年度の戦略の提出状況

7. 得点状況の比較と考察

本演習のような戦略プログラミングでは，常に最高得点を出力し，正解と言える模範コードを提示することは難しい．また，レギュレーションが異なるので，単純な得点では，年度ごとの比較が適切でない．そこで，大会で用いた各年度の山札リストに対し，遺伝的アルゴリズムを用いて，最高得点を近似的に求めた．これは，大量のシミュレーションを実施し，与えられた山札にのみ適応した，カードの捨て方を求めるものである．この方法で求めた最高の理想得点を基準として，各戦略の得点を比率に換算して，達成度とした．

図 10 は，各年度の達成度を比較したものである．横軸を達成度として，各年度の分布を示している．受講者数も

異なるので, 縦軸は人数ではなく相対度数としている. 2012 年度は，2011 年度と比べ，達成率の平均と最高が向上している．分布も，全体的に右にシフトしており，2011 年度の双峰が単峰に変わっている．2013 年度は，傾斜掛率の導入という大きなルール変更があったが，上位陣は例年通りの達成度を保っていた．一方，下位陣には新ルールに対応できず, 取り残された層もいて, 双峰となった. 2014 年度は，2013 年度よりも平均は高くなり，単峰の分布となった．1 名を除いて，下位陣が大きく底上げされた．達成度の最高値も更新し，上位陣の層が厚くなった．全体的には，達成度は向上したといえる．

演習の聴講者は，前年度までの受講者である上級生と，意欲的で希望のあった下級生である．例年，聴講者の達成度は高く，受講者を超えている場合もある．提出は特定の時期のみであるが，これによって受講者の競争意欲を大きく刺激している．また，達成度としては，最高でも 50%強であり，遺伝的アルゴリズムによる理想得点は，目標に対する度合いとしては，妥当といえる．

	理想	受講者				聴講者
		平均	最高	最低	標準偏差	最高
2011	118	29.0%	44.6%	10.3%	7.5	―
2012	125	39.6%	52.3%	2.8%	9.1	50.9%
2013	202	36.8%	48.2%	22.6%	7.9	49.5%
2014	150	37.7%	52.1%	19.1%	6.9	59.3%

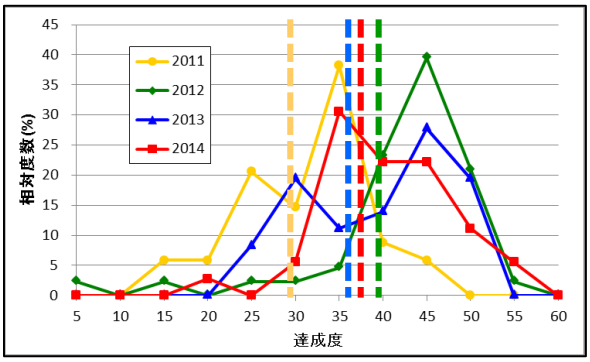

図 10 各年度の達成度の状況

8. おわりに

ポーカーの戦略プログラミングを題材とする C 言語の応用演習を提案している．具体的な課題を設定し，実行環境を構築している．提出された戦略を管理し，実行結果を公開する運営サーバ WinT を開発している．受講者は，ランダムな山札に対して，不要な手札を捨て，高い手役を作る戦略プログラムを作成する. 得点や順位をサーバで確認し，最良戦略を目指す.

2013 年度の結果分析に基づき，2014 年度の演習実践に向けて，WinT の機能を追加し，GUI を改良した．戦略コードの外部評価に相当する実行結果の得点だけでなく，内部評価に関連するコードメトリクスに着目した．そこで，得点との相関性に基づく回帰的指標 RCM を導入した．具体的には，コード量に関してファイルサイズ，冗長性に関して制御構文の条件数と圧縮率，構造化に関して関数の定義数を計測した．大会中の受講者へのフィードバックを促すため，戦略コードの提出状況および回帰的指標 RCM と散布図 SMP の表示の機能を追加した．

2014 年度の演習実践では，聴講者の参加や提出状況の時系列表示が意欲を刺激し，提出数が大幅に増えた．得点分布は，単峰となり，平均点が高くなった．下位陣が大きく底上げされ，上位陣の層も厚くなった．図 7 は，実際の最終大会において, 4 つの CM に対する SMP を示したものである．図上に相関係数も添えている．ファイルサイズと制御構文の条件数については, 得点との相関が大きかったが, 関数の定義数と圧縮率については，それほど相関がみられなかった. 現在, より詳細な実践結果の分析を行っている.

今後の課題として，より適切なコードメトリクスの選択を検討する．また，これらの指標を，コーディングの進捗状況の把握[7][8][9]，コーディングの個人傾向の推定[10]などにも応用する．さらに，受講者への効果的なフィードバックの手法を実現する．

参考文献

1) 富永浩之 ：プログラミング実習の題材としてのゲームと支援環境，ゲーム学会 第 1 回合同研究会, Vol.1, No.1, pp.39-42 (2003).
2) 山田航平，富永浩之 ：ボードゲーム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, No.5, pp.1-2 (2014).
4) 玄馬史也，吉田亜未，大川昌寛，山田航平，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 ―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).
5) 花川直己，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 ―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) 上田和志，富永浩之 ：類似性に基づくレポート剽窃の検出ツールのプログラミング課題への適用，情処研報, Vol.2010, No.9, pp.1-8 (2010).
7) 加藤利康，石川孝 :プログラミング演習支援システムにおける学習状況把握機能の提案，情処研報, Vol.2013-CE-120, No.2, pp.1-8 (2013).
8) 内藤広志，齋藤隆 ：プログラミング演習の自動採点システムの評価法と進捗状況，情処研報, Vol.2013-CE-120, No.1, pp.1-7 (2013).
9) 蜂巣吉成，吉田敦，阿草清滋 ：プログラミング演習におけるコーディング状況把握方法の考察，情処研報, Vol.2014-CE-125, No.3, pp.1-8 (2013).
10) 伏田享平，玉田春昭，井垣宏，藤原賢二，吉田則裕 ：プログラミング演習における初学者を対象としたコーディング傾向の分析，信学技報, Vol.111, No.481, pp.67-72 (2012).