

カードゲーム戦略を題材とした応用 C 演習における 提出コードの ABC サイズによる品質評価とサーバ上での提示機能

玄馬史也[†] 富永浩之[†]

香川大学[†] 香川大学[†]

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。本研究では、カードゲームのポーカーの戦略を題材とする応用 C 演習を提案している[1]。本演習の戦略プログラムは、5枚の手札から不要な1枚を捨て、山札からの1枚と交換していく。既定のチェンジ数までに、配点が与えられた9種類の手役のうち、高い手役を作る。十分にシャッフルされた1つの山札で、このテイクを繰り返す。作った手役の素点を、テイクごとの傾斜掛率で重付けして集計する。大量のランダムな山札による平均を得点とする。

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、その中で得点を競う。その支援として、大会運営サーバ WinT を運用する。受講者は、提供された実行環境の下、各自のローカル PC 上で戦略プログラムを作成する。WinT は、提出された戦略のソースコードをコンパイルし、ゲームを実行する。サーバ側では、予め10000個のランダムな山札リストを用意しておく。これらによる実行結果を平均し、サーバ上での戦略プログラムの得点とする。チェンジ数とテイク数、傾斜掛率は、年度ごとに変更し、レギュレーションと呼ぶ。

2. 戦略プログラムの外部評価と内部評価

本演習では、単に与えられた仕様を満たすプログラムの作成ではなく、持続的な改良を目指す開発体験を重視する。そこで、戦略プログラムの評価として、2つの側面を取り入れる。外部評価は、プログラムの実行性能を測るものであり、実行結果としての得点である。内部評価は、ソースコードの品質に関するものであり、本質的には定性的なものである。得点が高い戦略が、必ずしも質の良い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

ードが含まれている場合がある。このようなコードは、リファクタリングすべきである。

コードの品質は、静的解析ツールを利用したコードメトリクス(CM)で、定量的かつ自動的な評価をある程度は行える。最終的には、実際にコードを黙視する必要があるが、CMが極端な値を示すコードは、何らかの意味で不適切な可能性が高い。本研究では、これらの評価に対する考察と改善を試み、WinTに組み込んでいく。

3. 外部評価としての理想得点と達成度の導入

外部評価として、前述の単純な得点は、同じレギュレーションの下での比較には問題ないが、異なる年度の比較には不適切である。そこで、遺伝的アルゴリズム(GA)を用いたシミュレーションを採用する[2]。各山札に対し、理想的な捨札列をGAで探索し、最高点の近似値を得る。これを理想得点 IP と呼ぶ。ただし、この手法は、任意の山札に対する具体的な最良戦略を求めるものではない。各年度の個々の戦略プログラムに対し、単純な得点を IP で割った相対得点 ADIP を求める。ADIP は、理想得点に対する達成度を意味しており、レギュレーションの差によらない比較に有用である。各年度の ADIP の最高点は、50%程度で、大きな差はない。また、平均点や分布は、年度ごとの傾向をより適切に示している。

さらに、各年度の IP に対し、傾斜掛率の和である重みで割った値を基礎得点 BP と呼ぶ。BP は、1テイク当たりの理想得点である。チェンジ数が多いと、BP が高くなる傾向がある。これは、交換回数が増えれば、より高い手役を狙うことができるためである。2011年度と2014年度は、チェンジ数は同じだが、2011年度の方がBPが高い。これは、カードの残り枚数が0になる可能性のある2014年度の方が、戦略的に難易度が高いためである。

4. 内部評価としての SMP と RCM の導入

内部評価については、これまで、4つの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[3]。個々のCMの値は、戦略の内容にも大きく左右される。そこで、各コードのCMとADIPをプロット

した散布図 SMP を導入した. さらに, 各ドットと回帰直線 L との距離を標準偏差で割った回帰的指標 RCM を計算する. L 上にあれば, RCM は 0 となる. RCM の絶対値が大きいと, 得点の割に品質が低い「特異な」コードと判断される.

WinT では, 戦略プログラムの提出の反映として, 得点や順位を個人および全員に公開する. 順位の変移を見て, 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, 試行錯誤の繰り返しの動機付ける. 締切時に, 提出された各自の戦略のうち, 最高得点となる最良戦略を最終結果とし, 成績に反映する. また, コード品質についても, 全体の SMP を公開し, 各自の RCM も通知する. これにより, 大会中に, 特異なコードに気付かせ, 修正を促進する.

5. ABC サイズの導入とサーバ上での提示機能

現在の CM については, 複数の指標が羅列的で意図が伝わりづらいという問題点がある. そのため, 受講者による活用は十分とは言えなかった. そこで, 新たな CM として, ABC サイズを採用する[4]. これは, 関数やメソッドごとの指標であり, これまでの CM を統合したものといえる. 変数への代入 A(Assignment), 関数の呼出 B(Branch), 条件文 C(Condition)の個数から二乗和の平方根 $\sqrt{A^2 + B^2 + C^2}$ で算出する.

ここで, C 言語のプログラムを対象とする静的解析ツール AdLint を利用する[5]. AdLint は, 対象となるコードを部分的に実行し, 多種のメトリクスを計測する. AdLint の雑多な解析結果から, 関数ごとに必要な情報を抽出し, ABC サイズを求める Ruby スクリプトを開発した.

ABC サイズは, 冗長性や複雑さの点で, 不適切さを示す指標である. リファクタリングの判断材料にもなる. プログラムの言語や種類にもよるが, 本演習では, 30 以上を特異であるとみなす. ただし, 関数ごとの指標であり, 1 つのコード内でも, ばらつきがある. そのため, コード全体に対する簡易な指標も必要である. そこで, コードに現れる全ての関数について, 二乗平均を取った値 Q-ABC を導入する.

WinT の新版では, 2 つの提示機能を実装した. 戦略一覧ページでは, 各自の全ての戦略コードに対し, Q-ABC と ADIP の推移を図示する(図 1). 戦略分析ページでは, 各コードに対し, 関数ごとの ABC サイズおよび個々の CM を一覧する(図 2). ここで, ABC サイズが 30 以上を赤色で警告とし, 20 以上を黄色で注意とする. これらにより, 性能と品質の両方を常に意識させ, 持続的なコードの改良を誘導する.

6. 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. 提出された戦略コードの実行結果を公開する大会運営サーバ WinT を開発し, 授業で運用している.

実行性能を測る外部評価の精密化のため, GA による理想得点 IP を導入し, 相対得点 ADIP を採用した. コード品質を測る内部評価として, ADIP とコード指標 CM との散布図 SMP を導入し, 回帰的指標 RCM を求めた. その精密化のため, 静的解析ツール AdLint を用いて, 関数ごとの ABC サイズを求めるスクリプトを開発した. さらに, コード全体の指標 Q-ABC に拡張した.

これらの情報は, WinT に提示機能として実装した. 特に, ABC サイズの通知は, 不適切なコードへの注意や警告を与え, リファクタリングを誘導する. 2016 年度の演習で実践し, その結果を分析して, 妥当性や有用性を検証する.

参考文献

- 1) 玄馬史也, 吉田垂未, 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).
- 2) 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 - 遺伝的アルゴリズムによる理想得点と傾斜掛率による達成度の傾向分析 -, 信学技報, Vol.116, No.126, pp.7-12 (2016).
- 3) 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.2014-CE-128, No.9, pp.1-6 (2015).
- 4) 玄馬史也, 富永浩之: ポーカー戦略を題材とする応用 C 演習の大会運営サーバにおける解答コードのメトリクス提示機能, 情報処理学会 第 78 回全国大会, pp.889-890 (2016).
- 5) 矢野尾裕: AdLint, <http://adlint.sourceforge.net/>.



図 1 戦略一覧ページ

行番号	関数名	ABCサイズ	代入回数	関数呼び回数	条件文回数
0	main	10,506	10	0	0
07	checkage	15,768	10	7	10
077	find_min_or_max	34,028	23	7	20
103	is_prime	49,118	40	3	20
209	flush	27,052	20	0	14
340	full_house	27,368	20	0	18
357	three_of_a_kind	5,477	3	1	5
416	two_pair	18,301	19	1	6
450	one_pair	14,836	14	0	5
473	full	90,038	46	2	20
500	straight_flush	8,244	5	0	3
673	straight_flush	8,244	5	0	3
699	cherry_blossom	5,274	7	1	6
616	control	6,706	0	0	3
614	bet	6,087	8	0	4
663	hand_check	18,851	16	1	10
670	bet_size	2,236	3	0	1

図 2 戦略分析ページ