

# 実行テスト系列を取り入れた小コンテスト形式の初級 C 演習 — 学生の得点状況の時系列分析による活性度の推定 —

富永浩之<sup>†</sup> 西村智治<sup>†</sup>

実行テスト系列を取り入れた小コンテスト形式の初級 C 演習を提案し、大会支援サーバ tProgrEss を開発している。試験時間や時間調整点などの得点ルールを改良しながら、これまで数年にわたり授業実践を行ってきた。各コンテストにおける学生の提出状況と積算得点を時系列で捉え、成績で 4 群に分けて分析した。各群の特徴や振舞いパターンの類型化を議論し、活性度を推定する。

## Introductory C Programming Exercises Based on Contest Style Using Execution Test Series - Classification and Estimation of Student Behavior Patterns by Answering Situation Logs -

HIROYUKI TOMINAGA<sup>†</sup> TOMOHARU NISHIMURA<sup>†</sup>

To improve introductory C language lesson in an engineering college, we have proposed a small contest style exercise in classroom and homework. We have developed tProgrEss, the contest management Web server. It judges an uploaded program by an execution test with input and output data. To adjust a contest for beginners, we offer several preparation tests, which are for step-by-step sub-goals as partial specification. We set the rule of partial points and plus or minus points based on time. The system shows the result of judgment and exhibits the ranking page immediately. We carried out several educational practices in 2010 and 2011. For analyzing the results, we introduce two kinds of time series graph about total submission and personal points. We classify four groups by score and illustrate behavior patterns. We discuss estimation of student activity by the graph.

### 1. はじめに

本研究室では、授業中の演習を活性化させるため、初心者が気軽に参加できる小コンテスト形式の演習を提案している[1][2][3][4]。1回のコンテストで数問のプログラミング問題を提示する(図1)。ローカルPCで解答コードを作成し、Webから大会支援サーバ tProgrEss にアップロードする[5]。サーバ側でコンパイルした実行バイナリに、入力サンプルを与えて実行し、実行結果と出力サンプルを照合して、正誤を判定する(図2)。ただし、初心者がプログラムを完成するには時間がかかり、途中で諦めてしまう可能性がある。そこで、一部の仕様を満たす解答に対しても部分点を与えるように、実行テスト系列を導入する。また、各学生の進捗状況を即時的に公開し、競争意欲を高める。

### 2. 演習形式と大会支援サーバ tProgrEss の概要

#### 2.1 コンテスト形式の実施形態

コンテストの実施形態は、教室型/宿題型/試験型を用意する。これらは、異なる機能を提供するのではなく、コンテストの要項や問題編成の指針を示すテンプレートである。教室型は、毎回の授業の最後に、総まとめとして、教室内で一斉に実施する。基本的な事項で、例題の類題として、

15～30分で解ける問題を出題する。宿題型は、1～2週間の期間内とし、学外からのアクセスも許容する。応用的な事項の組合せで、60～120分の問題を出題する。試験型は、成績に関与する演習で利用する。教室型より長い、制限時間を厳格にし、難易度の高い問題を出題する。

#### 2.2 実行テスト系列による正誤判定

コーディングの進め方を誘導したり、プログラムの作成指針を示すため、複数の予備テストと1つの最終テストからなる実行テスト系列を採用する(図3)。予備テストは、題意の仕様を緩めたもので、3つ程度までの中間目標として構成し、徐々に完成に近付けさせる。予備テストは、入力制限と出力許容に分けられる[6]。入力制限では、仕様で想定される範囲内の一部のデータでのみ正しく実行できればよく、例外処理を必要としたり、継続条件や分岐条件で境界値となるものを避ける。出力許容では、出力の一部のみが正しければよい。また、仕様で要求される出力の書式において、照合基準を緩和する。例えば、データ列の平均と最大値を求める問題では、平均だけの出力が合えばよい。

#### 2.3 誤答減点と時間調整点による得点ルール

プログラムの判定結果には、時間と誤答による減点を反映した得点ルールを適用する(図4)。満点は、標準時間内に誤答なしで最終テストにパスした場合の得点である。ここから、誤答回数と経過時間で減点する。ただし、減点幅には上限を設け、最低でも基準点を与える。最終テストに正

<sup>†</sup>香川大学

<sup>†</sup>Kagawa University

答できなかった場合は、予備テストの部分点のみを与え、減点を行わない。

経過時間については、時間調整点として、早期の提出に対する加点と、標準的な期限からの遅延の提出に対する減点がある。ただし、最終的な期限は、別途に設け、それを過ぎると参加点程度の大幅な減点とする。誤答減点や時間調整点の設定により、素早く解答させるコンテスト、じっくり時間をかけて取り組ませるコンテストなど、コンテストの性格を調整することができる。

## 2.4 大会支援サーバ tProgrEss の機能

上記のような演習を実施するため、大会支援サーバ tProgrEss を開発し、実際の授業で運用している。学生の解答コードがアップロードされると、図5の6段階で判定する。提出履歴は全て記録され、コンテストの実施状況や学生の振舞いの分析に用いる。

学生へのインターフェースとして、判定結果、およびコンパイル時のメッセージと実行結果を、判定確認ページで提示する。学生は、これらの情報を基に、解答の修正や、次のテストに取り組む。また、順位表示ページで、各自の判定結果は教室全体にも公開される。他人の進捗状況を見て、競争意識を高め、意欲を持続させる。教師側ページでは、学生の提出状況や問題の着手状況を、時系列グラフで表示する[7][8]。また、学生の提出したソースコードも閲覧でき、自動判定では見逃される不十分なコードや理解不足の点に注意を与える。

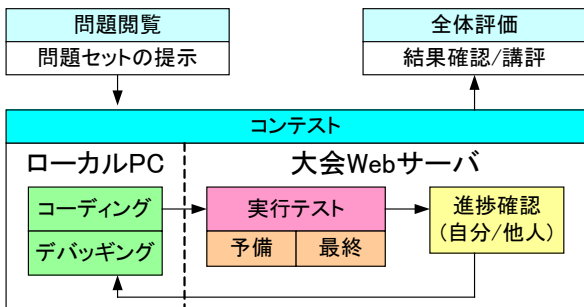


図1 コンテスト形式の演習の進行

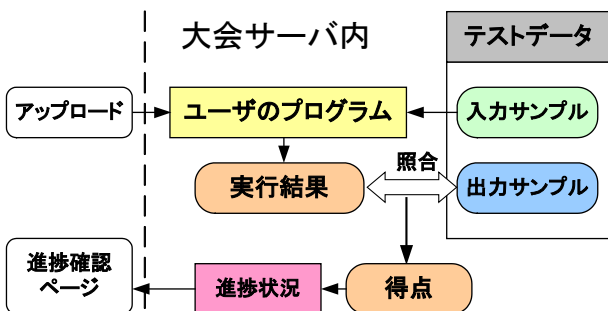
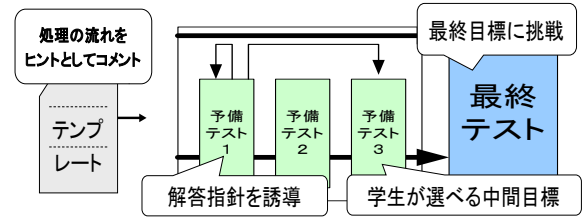


図2 解答コードの判定手順



|         | 予備テスト          | 最終テスト          |
|---------|----------------|----------------|
| 入出力サンプル | 部分的な仕様 データを明示  | 完全な仕様 データを隠蔽   |
| 正答の場合   | 解ける所まで取り組める部分点 | 解答時間を差し引いた得点   |
| 誤答の場合   | 減点なしで何度でも提出可能  | 回数による減点で慎重さを促す |

図3 実行テスト系列の予備テストと最終テスト

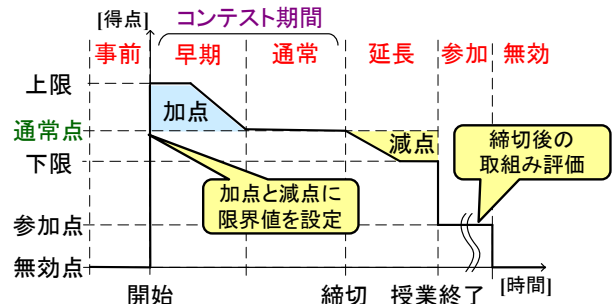


図4 時間調整点

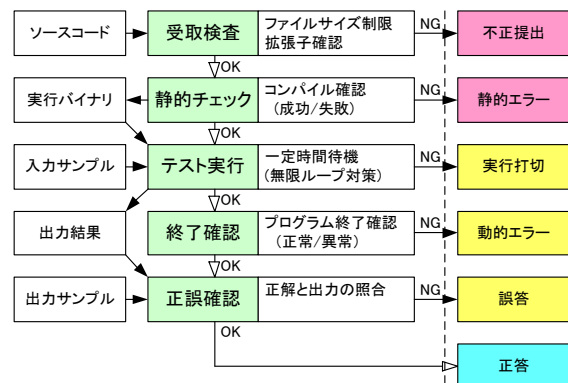


図5 実行テストの処理手順と判定結果

## 3. コンテストの実施結果の分析

### 3.1 提出状況グラフと積算得点グラフ

コンテスト中の学生の振舞いについて、時系列での提出状況と個人ごとの積算得点に着目し分析を行う。提出状況のグラフは、横軸をコンテスト開始からの時間とし、適当な時間幅で分ける。縦軸をその区間ごとの提出件数の棒グラフとし、予備と最終の内訳も示す。折線は、予備および最終の正答率である。積算得点グラフは、横軸をコンテスト開始からの時間とし、個人ごとの積算得点を折線で示す。コンテストの最終得点や授業の最終成績で、4群に分け、それぞれを示す。

本演習は、2010年度前期の授業「プログラミングⅡ」で

実施した[9]. 対象者は、情報系学科 2 年生 64 名である。この時点では、大会支援サーバ tProgrEss が旧版であり、時間調整点は導入していなかった。2010 年度後期の授業「プログラミング I」では、時間調整点を導入した新版で実施した。ただし、授業の担当者が異なるため、臨時的に 1 回だけの運用である。対象者は、情報系学科 1 年生 80 名である。前者の 1 年前の学年になる。また、2011 年度は、システムの修正に努め、授業では実施せず、自習用の試験運用とした。2010 年度前期の旧版での実施については、図 6 に教室型、図 7 に宿題型、図 8 に試験型の提出状況グラフと 4 群ごとの積算得点グラフを示す。2010 年度後期の新版での実施については、図 9 に宿題型のグラフを示す。

### 3.2 振舞いパターン

コンテスト中の学生の振舞いは、理解度と積極性の観点により 4 つのパターンに分類できると考えられる(図 10)。パターン 1 は、授業内容を理解し、コンテストへの参加も積極的な学生である。そのため、開始時から積極的に提出を行い時間調整点での加点を獲得でき、理解度も高いため多くの問題を正答し、最終的な得点は、最も高いものとなる。パターン 2 は、授業を理解しているが、コンテストへの参加が消極的な学生である。開始時はあまり提出せず、締切直前に提出するので時間調整点で減点されるが、理解度は高いため多くの問題を正答し、終了間際で得点が伸び、2 番目に高いものとなる。パターン 3 は、授業の理解は不足しているが、コンテストには積極的な学生である。開始時から提出を行い、時間調整点での加点をある程度得られるが、簡単な問題しか正答できない。最終的な得点は、3 番目となる。パターン 4 は、授業の理解が不足し、コンテストにも消極的な学生である。全般的にあまり提出を行わず締切直前になって取り組むが、簡単な問題しか正答できず、得点が伸びない。最終的な積算得点は、最も低いものとなる。

### 3.3 積算得点グラフの時系列分析

時間調整点の有無による相違として、宿題型の図 7 と図 9 について分析する。時間調整点有りの宿題型では、加点期間が終了する土曜日までに 56 名(70%)の学生がコンテストに着手し、減点期間が始まるまでには 65 名(81%)の学生が着手している。最後まで、提出しなかった学生が 4 名ほどいたが、これは自由参加のコンテストだったためだと考えられる。また、以前の宿題型コンテストよりも提出数が増えている。時間調整点の導入により、学生の解答の提出が前倒しになり、予備テストを活用した小まめな提出も増加している。また、以前の宿題型では、各学生が様々な時間に提出していたが、時間調整点の導入により、できる学生は早期に提出するようになった。これにより、問題に躓いている学生の発見が以前よりも容易になった。

## 4. 活性度による学生の振舞いの視覚化

これまでの 4 群での分析において、成績下位群の学生の提出数が少なく、得点に変化しない期間が続くことが確認された。成績下位群をコンテスト中に早期に発見し支援する必要がある。そのため、積算得点グラフの面積の大小を指標とした、活性度を定義した[10]。活性度は、時間経過による積算得点の変化に着目し、学生の振舞いを視覚化する。また、全てのコンテストで共通して扱うため、各学生の積算得点グラフの面積をグラフの総面積(経過時間×コンテストの設問の総得点)で割り、割合で表す。成績に関わるため学生の振舞いが顕著になる試験型のコンテストについて、時系列での 4 群ごとの活性度の変化を分析した。図 11 は、4 群の活性度の平均値を時系列で表したものである。

コンテスト終了時点での各群の活性度の平均は、第 1 群が 0.45、第 2 群が 0.32、第 3 群が 0.21、第 4 群が 0.08 であった。各群は、それぞれ活性度が約 0.1 の差が見られる。これらから、全体での活性度は、成績上位の第 1、2 群が高く、成績下位の第 3、4 群が低くなっていることが分かる。

さらに、学生の初期の活性度と最終的な得点の関係を分析した。図 12 における 60 分時点の各群における活性度の平均値を指標とし、4 つのパターンを定義した。そして、60 分時点での 4 群における各パターンの割合を分析した。

第 1 群は、全 17 名である。1 群パターンは、12 名で 70% と最も多い。2 群パターンは、3 名で 18%、3 群パターンは、1 名で 6%、4 群パターンは、1 名で 6%となっており、上位群パターンが 90%程度となっており、下位群パターンがあまり見られない。第 2 群は、全 17 名である。1 群パターンは、5 名で 29%である。2 群パターンは、7 名で 41%と最も多い。3 群パターンは、3 名で 18%、4 群パターンは、2 名で 12%となっており、上位群パターンが 70%となっている。第 1 群よりも上位群パターンが少ないが、全体的に過半数を占めている。第 3 群は、全 19 名である。1 群パターンは、2 名で 10%である。2 群パターンは、7 名で 37%、3 群パターンが 7 名で 37%となっており、2 群と 3 群パターンが多くなっている。4 群パターンは、3 名で 16%であった。2 群、3 群パターンが多くなっており、1 群、4 群パターンはあまり見られない。第 4 群は、全 11 名である。1 群パターン 0 名で 0%、2 群パターンは、1 名で 9%となっており、上位群パターンがあまり見られない。3 群パターンは、3 名で 27%である。4 群パターンが 7 名で 64%である。上位群パターンが 10%程度となっており、下位群パターンが 90%程度を占めている。これらの結果から、初期の活性度からおおよその最終的な成績を予測することが可能であると考えられる。そして、下位群になるだろう学生に対し、早期の支援を行うことを考えていく。

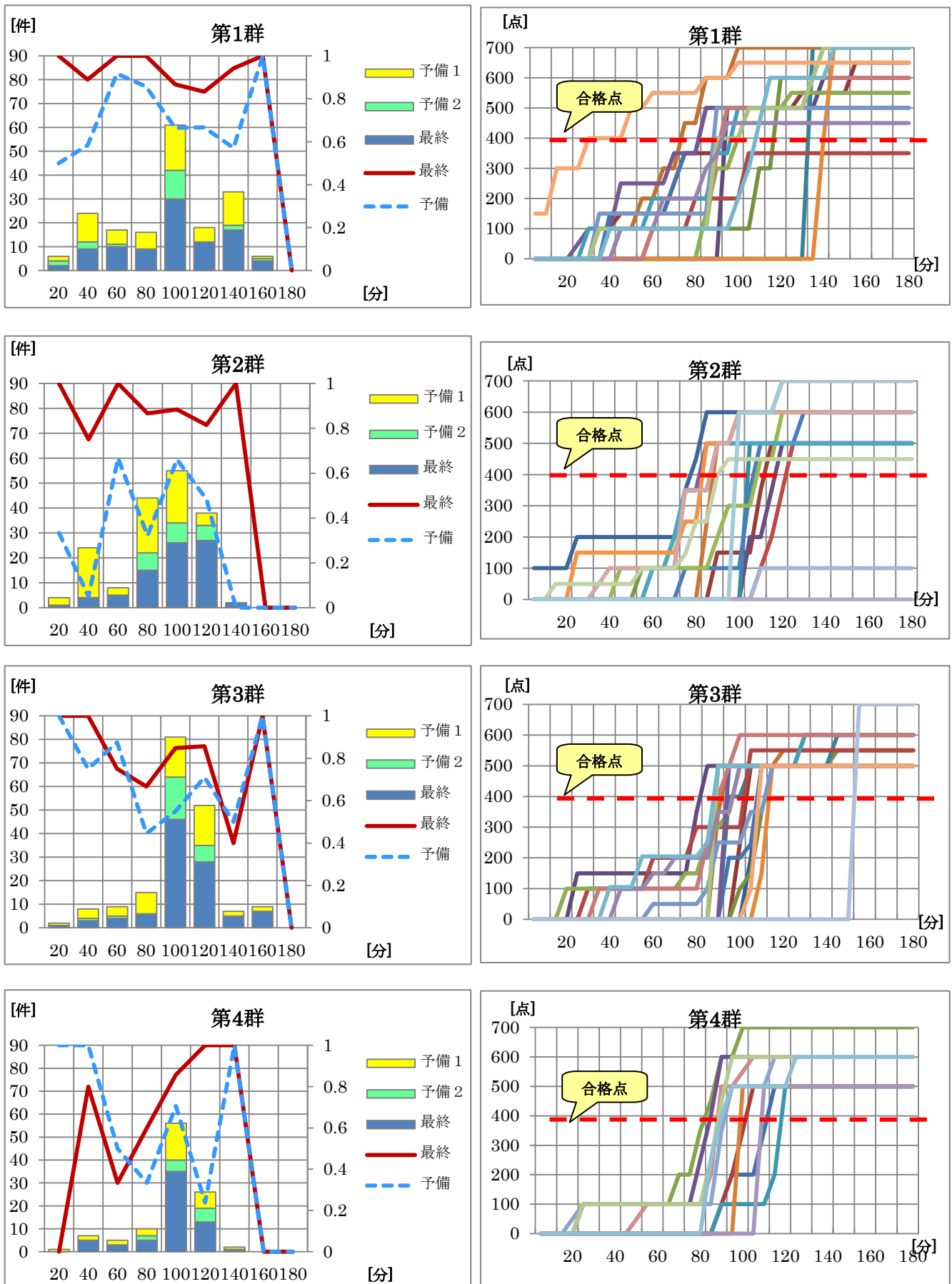


図6 「2010年度プログラミングⅡ」教室型の時系列による提出グラフと積算得点グラフ

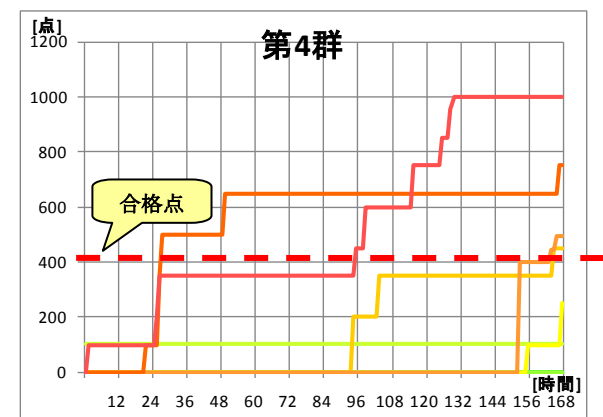
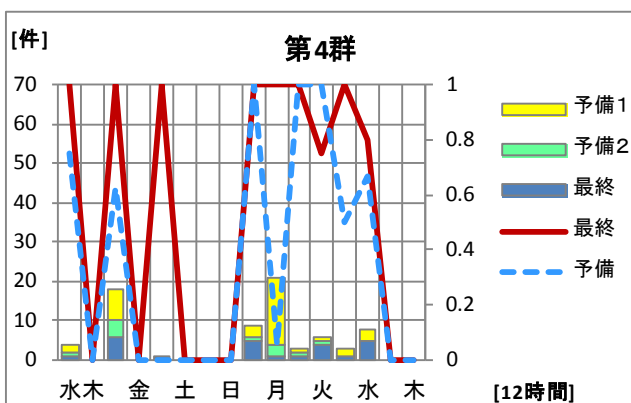
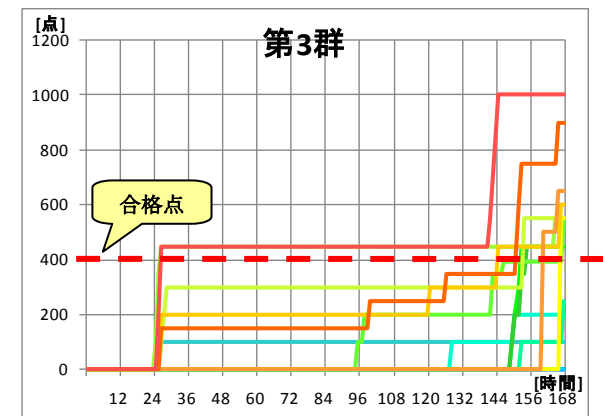
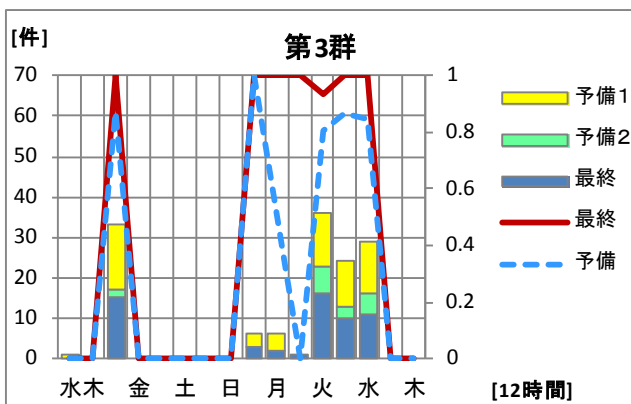
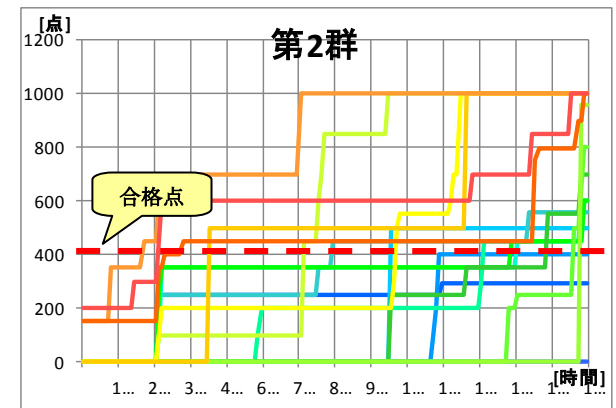
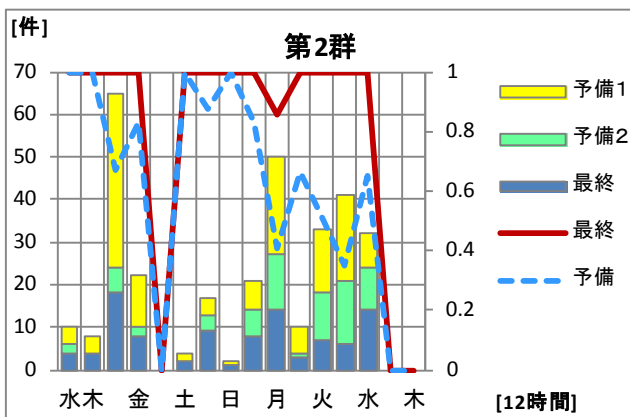
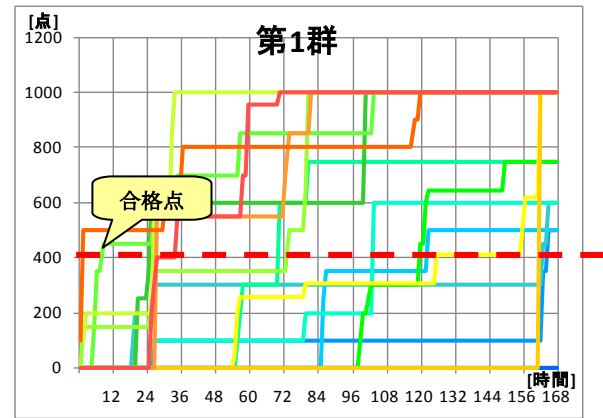
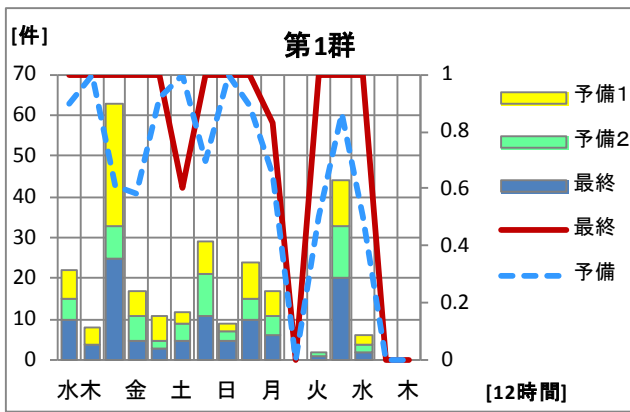


図7 「2010年度 プログラミングⅡ」宿題型の時系列による提出グラフと積算得点グラフ

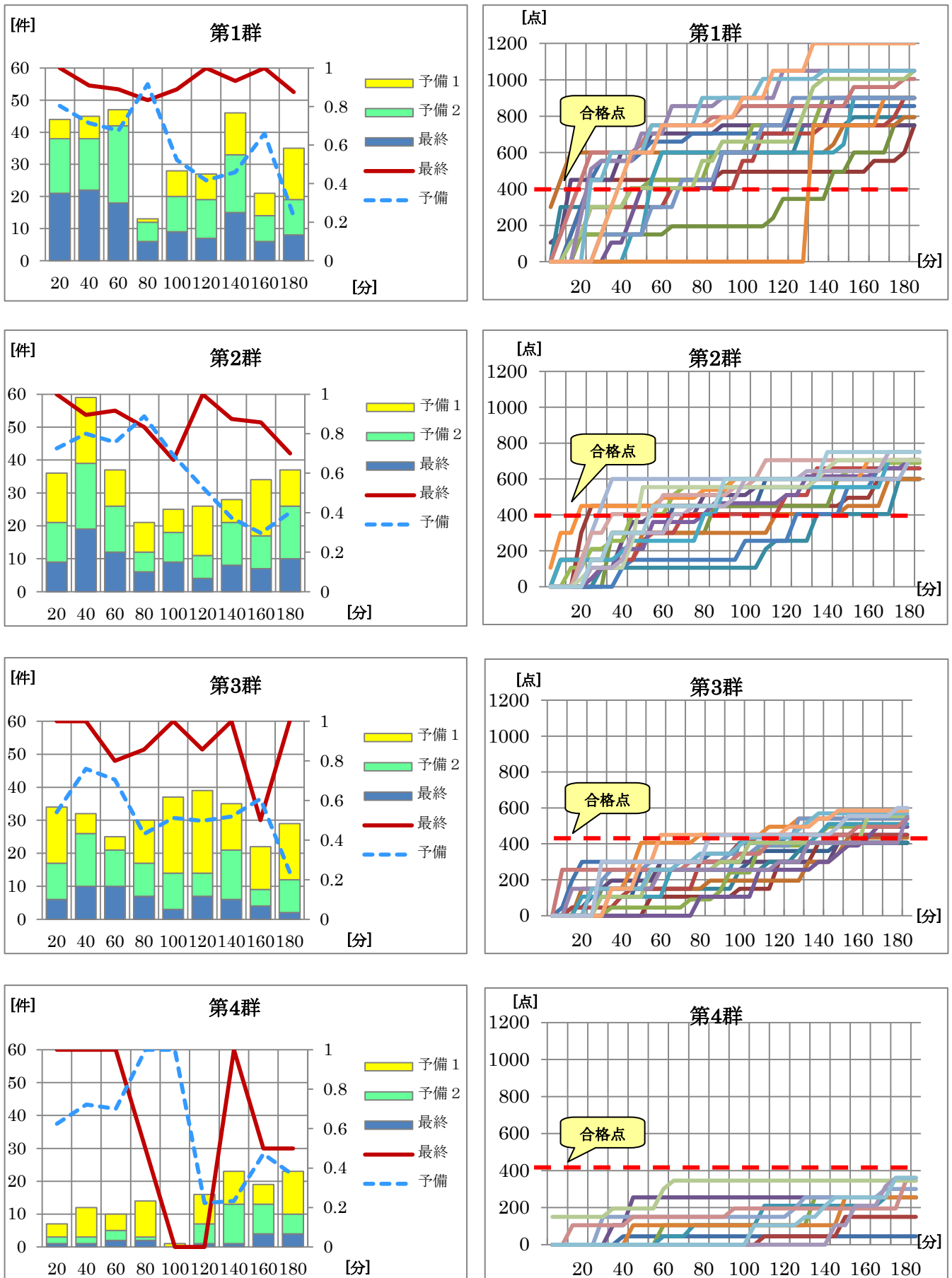


図 8 「2010 年度 プログラミング II」 試験型の時系列による提出グラフと積算得点グラフ

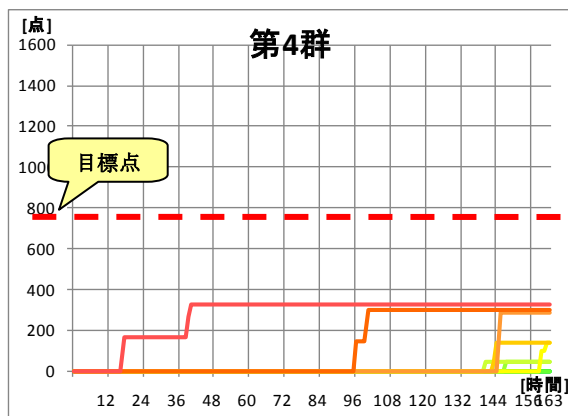
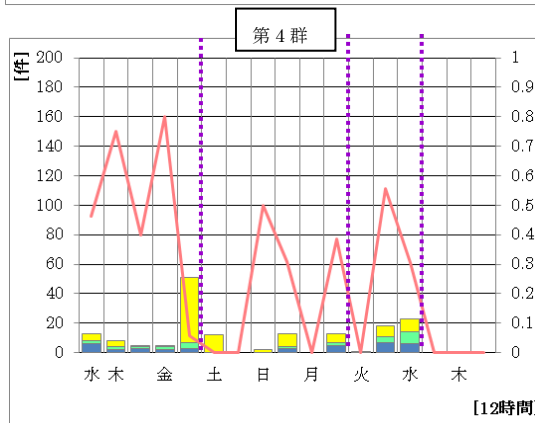
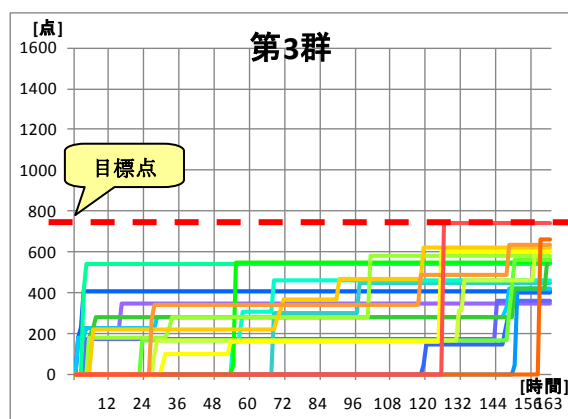
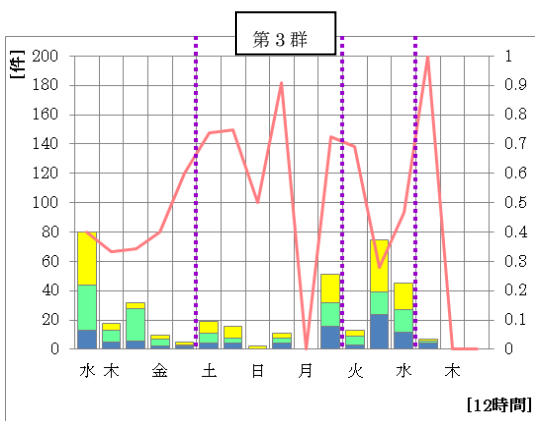
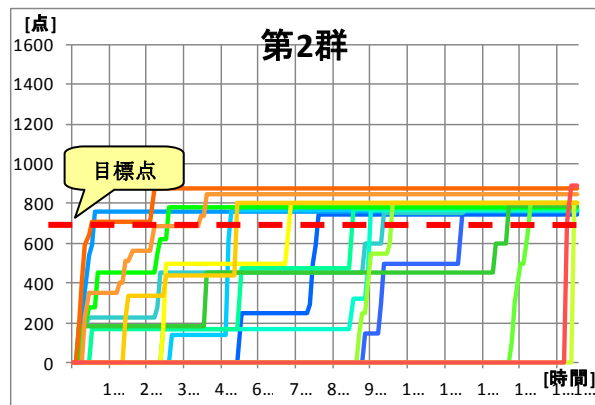
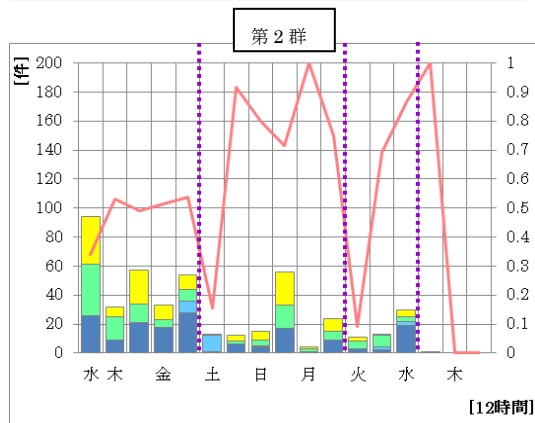
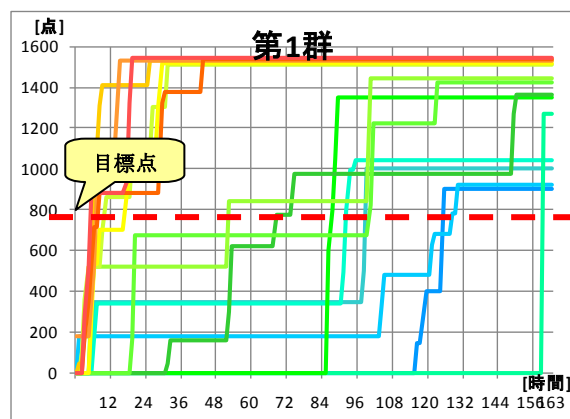
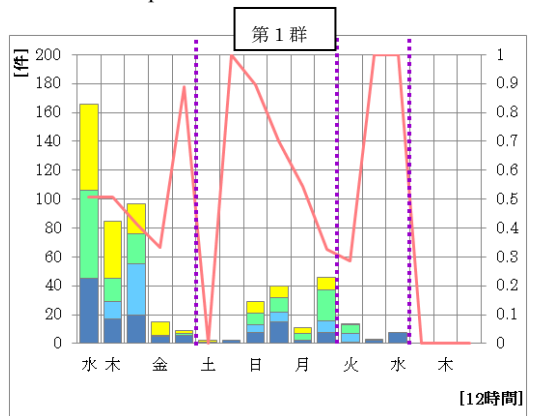


図9 「2010年度 プログラミング I」宿題型の時系列による提出グラフと積算得点グラフ

## 5. おわりに

大学情報系の入門C授業において、初心者向けの小コンテスト形式でのプログラミング演習を提案し、大会運営の支援サーバを開発している。実行テスト系列と時間調整点を含む得点ルールを採用し、演習への積極的な参加を促す。教育実践として、試験型のコンテストを実施した。学生の積算得点の推移を成績別にグラフにし、各群の特徴を分析した。その結果、上位陣と下位陣での振舞いの特徴を発見することができた。今後の課題として、即時的に積算得点をグラフ化する機能や、成績群に応じた支援機能の開発を行い、学生への助言ができるチュータリング機能の実装も目指す。

### 参考文献

- 1) 倉田英和, 富永浩之, 林敏浩, 垂水浩幸, 山崎敏範, "実行テストを用いたコンテスト形式の入門的Cプログラミング演習の大会運営サーバの開発", 情処研報, Vol.2006, No.108, pp.9-16, (2006).
- 2) 倉田英和, 富永浩之, 林敏浩, 垂水浩幸, "実行テストによるプログラム判定を用いた初級Cプログラミング演習支援と授業実践", 情処研報, Vol.2007, No.101, pp.11-18, (2007).
- 3) 富永浩之, 倉田英和, "コンテスト形式による初級Cプログラミングの演習支援", 情処研報, Vol.2008, No.42, pp.49-56, (2008).
- 4) 富永浩之, 川崎慎一郎, "競争型学習を取り入れた入門的Cプログラミング演習 -運用実験での実行テスト系列の利用状況-", 情処研報, Vol.2010-CE-104, No.4, pp.1-12, (2010).
- 5) 川崎慎一郎, 富永浩之, "競争型学習を取り入れた入門的Cプログラミング演習 -演習支援サーバ tProgrEss の出題解答と採点結果のページ表示の改良-", 信学技報, Vol.109, No.335, pp.187-192, (2009).
- 6) 川崎慎一郎, 富永浩之, "競争型学習を取り入れた入門的Cプログラミング演習 -実行テスト系列による部分採点のための柔軟な照合機能-", 情処研報, Vol.2010-CE-103, No.10, pp.1-8, (2010).
- 7) 西村智治, 川崎慎一郎, 富永浩之, "競争型学習を取り入れた初級Cプログラミング演習における教師支援 -モニタリング機能としての解答状況と提出状況の視覚化-", 信学技報, Vol.110, No.453, pp.163-168, (2011).
- 8) 西村智治, 川崎慎一郎, 富永浩之, "小コンテスト形式の初級C演習における教師支援 -解答状況の時系列表示によるモニタリング機能の試用実験-", 情処研報, Vol.2011-CE-110, pp.1-8, (2011).
- 9) 川崎慎一郎, 富永浩之, "競争型学習を取り入れた入門的Cプログラミング演習 -授業実践における解答コードの提出履歴の詳細分析-", 情処研報, Vol.2010-CE-107, No.10, pp.1-10, (2010).
- 10) 西村智治, 蔵本孝司, 富永浩之, "競争型学習を取り入れた初級Cプログラミング演習におけるモニタリング機能の開発", JSiSE 全国大会, Vol.37, pp.180-181, (2012).

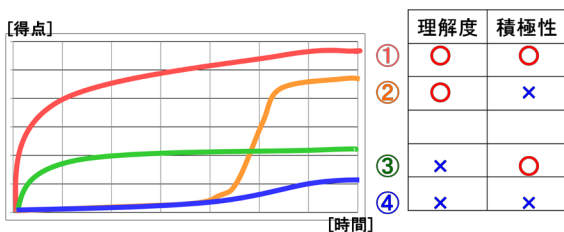


図10 学生の振舞いパターン

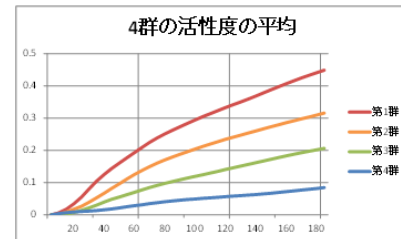
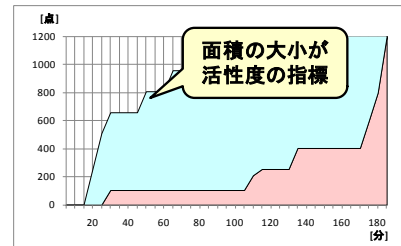


図11 活性度グラフ

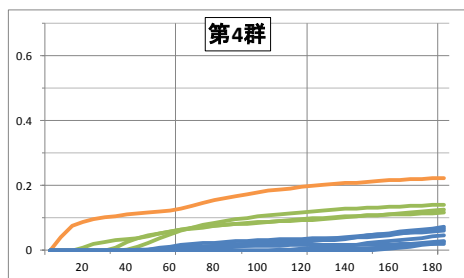
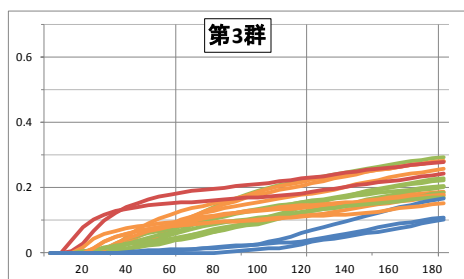
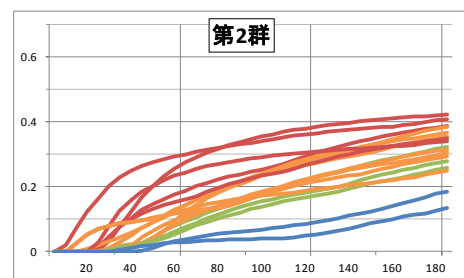
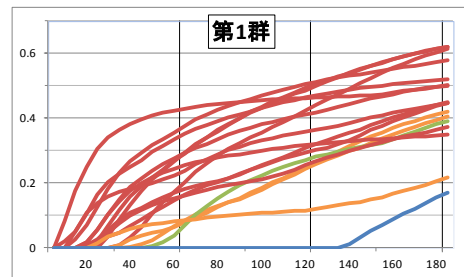


図12 試験型コンテストの活性度グラフ