JSiSE Research Report
vol.37,no.7(2023-3)

プログラミングにおける補助問題を用いた複雑な課題の理解を促す課題系列の提案とそのシステムの開発・評価

増田　敢太[*1], 前田　新太郎[*2], 東本　崇仁[*1]
*1 東京工芸大学工学部, *2 東京工芸大学大学院工学研究科

Proposal of Task Sequences Using Auxiliary Problems to Promote Understanding of Complex Tasks in Programming and System Development and Evaluation

Kanta Masuda[*1], Shintaro Maeda[*2], Takahito Tomoto[*1]
*1 Faculty of Engineering, Tokyo Polytechnic University
*2 Graduate School of Engineering, Tokyo Polytechnic University

プログラミング学習において，初学者は単純ソートのような複雑な課題を理解することは難しい．そこで本研究では，行き詰まりを解消する補助問題に着目した．補助問題は学習者の誤答箇所に応じた部分問題を提示する手法で，複雑な課題の理解を促すことが期待される．本稿では，補助問題による行き詰まり解消を指向したプログラミング学習支援システムの開発と評価を行った．評価から，プログラミング学習を対象とした補助問題を提示することによる学習効果が示唆された．

キーワード: プログラミング学習，適応的支援，補助問題，課題系列

1. はじめに

プログラミング学習において，複雑な課題を初学者がすぐに理解することは難しい．例えば，「要素数 n の配列 a の全体を昇順単純ソートで並べ替えよ.」というプログラミング課題を解くためには，一時変数を用いた入れ替えや，入れ替えを用いた並べ替え，最小値探索等のいくつかのアルゴリズムを組み合わせる必要がある．しかし，これらを組み合わせる活動は初学者にとって複雑であり，問題解決に行き詰まってしまうと考えられる．学習者が問題解決に行き詰まる原因として，何らかの理解が不十分な要素が存在することがあげられる．

学習者の行き詰まりを解消する支援の手法として，補助問題[(1)]があげられる．補助問題とは，もとの問題を簡単化した問題である．学習者が問題解決に行き詰まった際，補助問題を学習者に与えることで，学習者は問題解決の行き詰まりを解消する効果が期待できる．相川らは，力学の作図問題において，学習者の誤答箇所に適応的な補助問題を提示する学習支援システムを開発した[(2)]．学習者の誤答箇所に対応する補助問題を提示することで，学習者の問題解決の行き詰まりを解消することができた．

そこで本研究では，単純ソートのような複雑なプログラミング課題の学習を支援するため，補助問題を作成し，コードの意味的なまとまりごとに関係付け，課題系列を作成した．次に，作成した補助問題の課題系列を基に，学習者の誤答箇所に対応する補助問題を提示し，行き詰まりを解消する学習支援システムを開発した．開発したシステムは，学習者が誤答をした場合に学習者の解答から正答に不足している要素を分析し，誤答を繰り返した場合に誤答箇所に対応する補助問題

を提示する．学習者に誤答箇所に対応する補助問題を提示することで，個々の学習者の行き詰まりを解消し，複雑なプログラミング課題の学習を支援する．

2. 先行研究

2.1 力学における補助問題提示システム

相川らは力学の作図問題を学習題材として，Error-based Simulation（以下，EBS）を用いた補助問題提示システムを開発してきた．従来の EBS では，学習者の解答にもとづいたシミュレーションをフィードバックする．しかし，学習者の誤った解答のシミュレーションのみ提示するため，正解の提示が行われない．そのため，解答の修正すべき箇所を理解できず，正解に辿り着けない学習者の支援が行われていない．そこで相川らは，EBS を拡張し，学習者が誤答を繰り返した場合に誤答箇所に応じた補助問題を提示するシステムを開発した．誤答箇所に応じた補助問題を提示することにより，学習者の問題解決の行き詰まりを解消し，もとの問題の理解を促すことができた．

相川らのシステムでは，物体にはたらく力のベクトルを学習者に入力させ，その解答の中で欠落している力を分析し，対応した補助問題を提示する．力のベクトルは例えば重力や垂直抗力といった，その意味によって分類することができ，問題どうしを結び付けることができる．しかし，プログラミング学習では解答は複数行のコードであり，1 行ごとの処理を結びつけることは難しい．プログラミングでは，コードのまとまりごとに意味をもつ．そのため，その意味をもつコードのまとまりごとにプログラミング課題を結び付けることができる．

そこで本研究では，ある学習対象とするコードの全体（例えば，単純ソート）から，意味をもつコードのまとまりごとに補助問題として分け，それぞれの問題を結びつける課題系列を構築する．

2.2 従来のプログラミング学習における学習支援

古池ら[3]はプログラミングの構造的理解を支援するため，1 行ごとのコードにあらたな処理や機能をもつコードを加え，機能を変化させる活動による機能を拡張する学習の手法を提案した．学習者が問題に正解すると，その問題で学習した機能を部品として獲得する．その部品にあらたな部品やコードを加える問題を解くことで，機能を拡張し，より大きな部品を獲得する．小さな部品を拡張し，より大きな部品を作る活動を繰り返すことで，プログラミングを構造的に理解することを促した．また，併せて古池ら[4]は学習者の理解状態に適応的な課題提示を実現するため，プログラミング課題の構造化に取り組んできた．課題どうしは全体―部分関係と一般―特殊関係で結び付けられている．全体―部分関係は部分問題と部分問題を用いてあらたな機能を学習する全体課題の関係である．そして，一般―特殊関係は特殊課題の振舞いをより汎用的にした一般課題の関係である．これにより，課題を段階的に学習することが可能になる．

このような学習方法で誤答した場合，学習不足による誤答であれば，それよりも 1 段階簡単な問題を復習することが有効である．しかし，必要な問題を十分に学習しているにもかかわらず誤りがおこることがある．その場合はヒントによって解決過程を 1 つ進めることが有効である．そこで藤島ら[5]は，学習者が誤答した際，その部品となる問題の習熟状態によって適応的にフィードバックをする支援システムを提案した．学習者が誤答した際，その部品となる問題でヒントを使わずに正解している場合，詳細ヒントによって過程を 1 つ進める．ヒントを用いて正解している場合，簡易ヒントを提示し，さらに誤答した場合は未習熟部品として部品の問題の復習に移る．このように，部品問題の習熟状態によって適応的にフィードバックをすることで，個々の学習者の習熟状態に応じた支援を実現した．

しかし，藤島らの提案するフィードバックは部品問題の学習が必要となる．そのため，部品問題を学習していない学習者への支援は行えていない．そこで本研究では，問題の解答内容から習熟状態を分析し，個々の学習者の習熟状態に適応的に支援する．

3. 提案手法

3.1 課題系列

本研究では，古池らのプログラミング課題の構造化を用いて補助問題を作成する．そして，作成した問題どうしを全体―部分関係，一般―特殊関係によって関

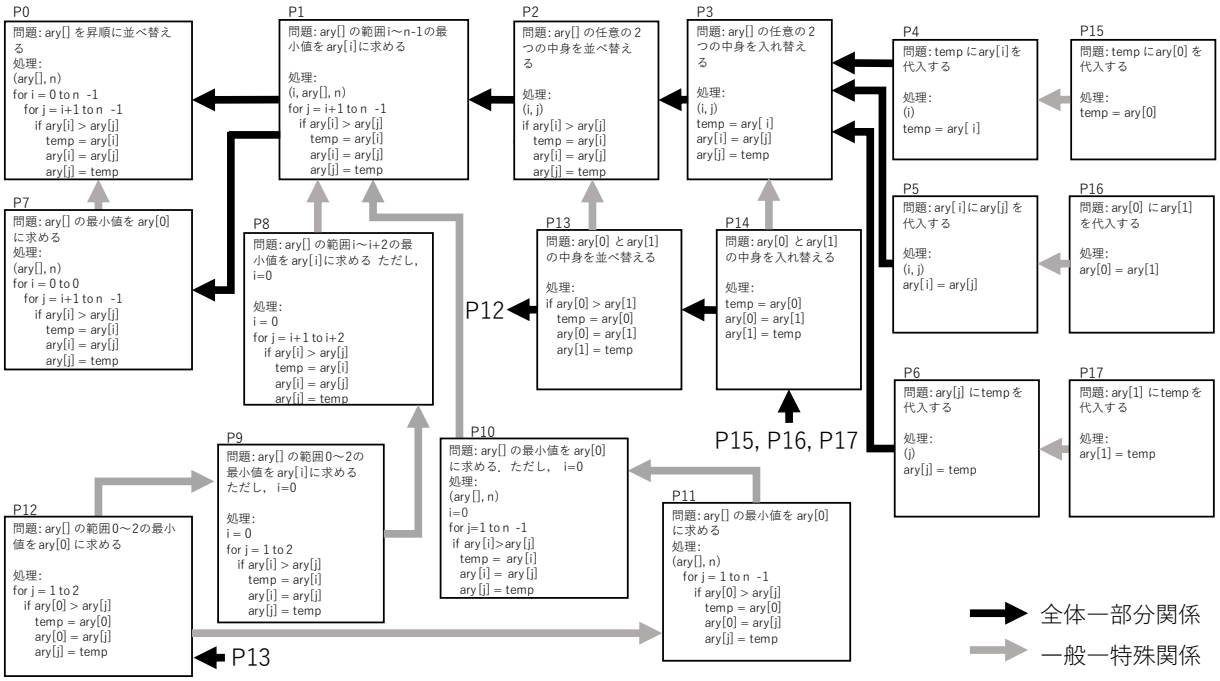

図 1 単純ソートの問題をもとに作成した課題系列

係付け，課題系列を作成する．図 1 は単純ソートの問題をもとに古池らのプログラミング課題の構造化によって作成した課題系列である．図中の灰色の矢印が一般—特殊関係，黒色の矢印が全体—部分関係である．P0 は要素数 n の配列 ary を昇順単純ソートする問題であり，その部分課題として，機能の一部を切り取った P1 の要素数 n の配列 ary の最小値を求める問題がある． そして特殊課題として，P0 の for 文の 1 回分の振舞いの一部を切り取った P7 の問題がある．

3.2 誤答箇所の分析

本研究では，正答のコードを機能ごとに分解し，まとまりごとに正誤判定をする．図 2 は単純ソートの問題を機能で分解した例である．単純ソートには最小値探索，並べ替え，入れ替えの機能が含まれており，全体—部分関係により小さい機能を大きい機能が包含する形となっている．例えば，並べ替えは入れ替えの機能を含んでおり，最小値探索はその並べ替えの機能を含んでいる．そのため，入れ替え部分，並べ替え部分，最小値探索部分，単純ソート部分の順に小さい機能をもつコードから分析を行い，学習者の解答の誤答箇所を判定する．例えば，図 3 左部のような解答では，入れ替えのまとまりが含まれているが，並べ替えのまとまりが含まれていない．この場合は，並べ替えの誤りとなる．

また，学習者がコードの振舞いを理解できていない場合，変数や条件式などを誤ってしまうと考えられる．例えば，図 3 右部のように for 文の条件式を誤っている場合，最小値探索部分の for 文の条件式の誤りとなる．

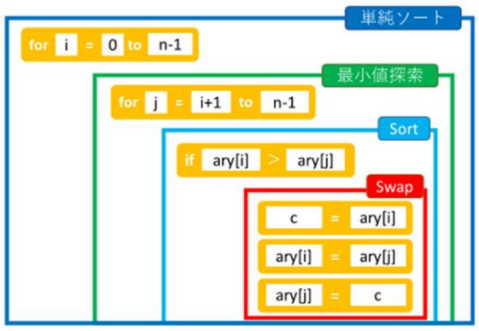

図 2 単純ソートを機能ごとに分解した例

```
for i = 0 to n-1
  for j = i+1 to n-1
    if ary[i] > ary[j]
      ary[i] = ary[j]
      ary[j] = ary[i]
```

```
for i = 0 to n
  for j = 0 to n
    if ary[i] > ary[j]
      c = ary[i]
      ary[i] = ary[j]
      ary[j] = c
```

図 3 単純ソートの問題の誤答の例

3.3 問題の移行

学習者が誤答を繰り返した場合は，3.2 の手法でえられた分析結果から，最も多い誤答箇所に対応する補助問題に移行する． 例えば P0 の問題で図 3 左部のよ

うな誤答した場合，P0 の部分課題である P1 に移行する．また，図 3 右部のように単純ソート部分の for 文の条件式のみ誤っている場合，単純ソートのコードの振舞いを学習するための P7 の問題に移行する．そして，補助問題に正解すると図 1 の課題系列に沿って P0 に向かって課題を移行する．例えば，P0 から P1，P2 を辿り P3 に移行したのち，問題にすべて正解した場合，図 1 の課題系列に沿って P3→P2→P1→P0 の順に問題を移行する．

補助問題に正解した際，移行先が複数ある場合，その問題に辿り着くまでに行った移行をなぞり，問題を移行する．例えば，P9 で誤答を繰り返し P12 に移行したのち，P12 を正解した場合は P9 に移行する．ただし，その 2 つの問題間の移行を 3 回以上行っている場合は，別の問題へ移行する．たとえば，P9 から P12 に移行し，P12 を正解したとき，P9－P12 間の移行を 3 回繰り返していた場合は P11 に移行する．

4. 提案システム

4.1 システム概要

図 4 は学習者が使用するシステム画面である．画面左上に問題が提示され，右上に解答の条件が提示される．そして作業スペース上部に引数が提示される．図 5 では「問．要素数 n の配列 ary の全体を昇順単純ソートで並べ替える」という問題文と，条件として「変数について配列は ary，要素数は n，記憶変数 c，添え字は i，j の順で用いること．」，引数として「ary[], n」が提示されている．学習者は画面左側のブロックリストからブロックを画面中央の作業スペースに追加し，ブロックを操作することでプログラムを作成する．

学習者は解答を作成したのち，画面右下の解答ボタンを押し，正誤判定を行う．解答したのち，画面右側に表示される実行結果で学習者が作成したプログラムの機能を確認できる．図 5 は「要素数 n の配列 ary の最小値を ary[0]に求めよ．」という問題で，学習者が作成した解答と，その実行結果である．解答ボタンを押すと変数の初期値がランダムで生成され，「初期値」の欄に表示される．学習者が作成したプログラム実行後の終了値を「終了値」の欄に表示する．まず，「初期値」にプログラムで使用する変数「ary[0]，ary[1]，ary[2]，ary[3]，ary[4]，c」にランダムな値「39，34，23，34，47，28」が割り当てられている．この問題は ary[0]と他の要素を並べ替えていき，ary[0]に最小値を求める問題である．正答の場合は配列 ary の出力結果は「ary[0] = 23，ary[1] = 39，ary[2] = 34，ary[3] = 34，ary[4] = 47」となる．しかし図 5 の解答では，2 つ目の for 文の条件式を誤っているため，誤った出力結果が表示される．そして，学習者はこの実行結果をもとに解答の修正を行う．*1

図 4 システム画面

図 5 解答のプログラムの実行結果

4.2 誤答箇所の分析

学習者が誤答をした場合，3.2 の手法により誤答箇所を分析する．図 5 の解答では入れ替え，並べ替えのコードのまとまりが含まれている．そして，最小値探索部分で必要な繰り返しブロックも含まれているが，その条件式が誤っている．この場合，最小値探索部分の条件式の誤りと診断する．

4.3 問題の移行

本システムでは，答箇所ごとに誤答回数を数え，合計で 3 回以上誤答を繰り返した場合，「次へ」ボタン

*1 エラーによって結果を表示できない場合，終了値には「コンパイルエラー」と表示される．

が表示される．そして「次へ」ボタンを押すと，3.3 の手法にもとづいて最も多い誤答箇所に対応する補助問題へ移行する．また，補助問題に移行せず 6 回誤答を繰り返した場合，「次へ」ボタンを押してない場合も補助問題に移行する．補助問題で正解した場合は図 1 の課題系列に沿って移行していき，最終的に P0 の問題へ戻る．また，移行した問題で学習者が誤答をした場合も同様に分析をし，同様の条件で問題を移行する．学習者が図 5 のような誤答を繰り返した場合，図 6 のような「問．配列 ary の範囲 i～i+2 の最小値を ary[i]に求める．ただし，i=0 とする．」という問題に移行し，学習者は補助問題の学習に移る．このように，課題系列に沿って繰り返し補助問題を解くことで，学習者の理解が不足している要素を段階的に学習する．このような学習活動によって，もとの課題で生じた行き詰まりを解消することが期待される．

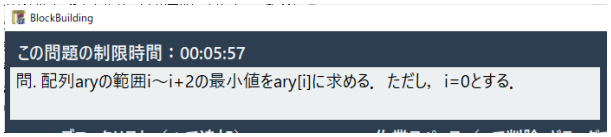

図 6 誤答を繰り返した場合の問題の移行の例

4.4 問題マップの表示

本システムでは，学習者の誤答箇所に応じて補助問題を提示する．しかし，プログラミングを題材としているため，問題文の提示による変化のみでは，もとの問題とあらたに提示された問題との差分を学習者は理解しづらいと考えられる．そのため本システムでは，課題系列をもとにした問題マップ（図 7）をシステム画面（図 4）とともに提示する．問題マップは，学習者の学習状況によって問題文や正答例がマップ上に表示される．また，学習者が現在学習中の問題は，青色で，学習者が既に到達したことがある問題は白色で問題文の背景がハイライトされる．既に正解した問題は赤色にハイライトされ，問題文に加えて正答例が表示される．図中では学習者は「ary[i]と ary[j]を昇順に並べ替えよ．」などの問題にすでに到達しているため，白い背景で問題文が表示されている．そして，「配列 ary の i～i+2 の最小値を ary[i]に求める．ただし，i=0 とする．」という問題に解答中のため，青い背景で表示されている．そして，その補助問題である「配列 ary の範囲 0～2 の最小値を ary[i]に求める．ただし，

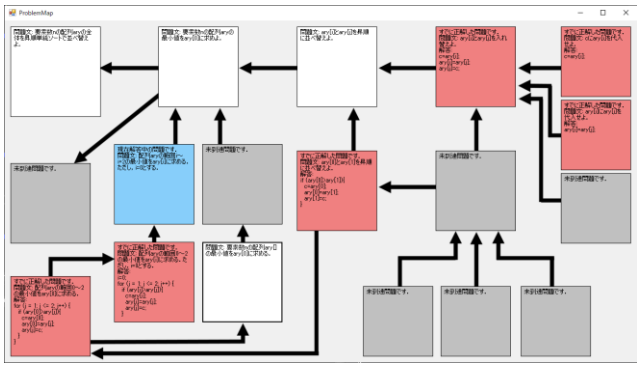

図 7 問題マップの例

i=0 とする．」などの問題を正解しているため，赤い背景で正答例が表示されている．

5. 評価実験

5.1 実験概要

本実験では，提案システムによる補助問題の提示は学習者自身が問題を選択する学習よりも有効性があるか検証する．対象者は工学部の学生 17 名である．実験は実験群 10 名と，統制群 7 名にわけて行う．実験群では本システムを使用する．統制群では本システムからシステムによる問題選択機能を削除し，代わりに学習者自身が問題を選択し学習するシステムを使用する．

まず，15 分間の事前テストを行う．その後，被験者は 60 分間システムを使用する．そしてシステム利用後に 15 分間の事後テストを行う．最後にアンケートを実施する．

事前テストではシステムであつかっている単純ソートをもとにしたプログラミング課題を 6 問出題し，システム利用前と利用後のプログラミング学習の理解度を調査する．実験群のシステムは，まず，もとの課題である「要素数 n の配列 ary を昇順単純ソートせよ．」という課題を提示する．そして，学習者が誤答をした場合は誤答箇所ごとに誤答回数を数える．学習者が繰り返し誤答している場合は，これまでの解答の中で一番多い誤答箇所に対応する補助問題を提示する．補助問題に正解した場合は課題系列に沿ってもとの課題に向かって移行を繰り返す．統制群のシステムは，実験群のシステムから，学習者の解答の一番多い誤答箇所によって補助問題を提示する機能と問題マップによる問題どうしの関係の可視化機能を削除したシステムで

ある．システム利用中は学習者自身が自由に問題を選択する．事後テストでは事前テストと同様の問題を出題する．

5.2 実験結果

5.2.1 テスト結果

テストの結果を表 1 に示す．テストは単純ソートを題材とした問題を 6 問提示し，各問題で 1 点ずつの 6 点満点で採点する．採点の結果，平均点が実験群では事前テストが 2.00 点，事後テストが 3.20 点となった．統制群では事前テストが 2.00 点，事後テストが 3.14 点となった． そして，効果量 d を求めた結果，実験群は 1.02 で効果量大，統制群は 0.53 で効果量中という結果になった．

表 1 事前・事後テストの結果

	平均		標準偏差		効果量(d)
	事前	事後	事前	事後	
実験群	2.00	3.20	1.00	1.32	1.02
統制群	2.00	3.14	1.60	2.53	0.53

5.2.2 ログ分析結果

統制群の被験者のうち，事後テストで満点をとった 2 名の問題の移行を表 2 に示す．また，学習者が選択した問題が実験群のシステムで選択する問題と一致している箇所を表中に下線で示す．この表から，C3 は，P0 から P17 までを順に選択し，時間内にすべて正解したのち，P0 を学習しなおしている．C7 は自身で選択した問題がシステムによって選択する問題と一致している箇所が多いことがわかる．このことから，適切に問題選択することができていたため，学習効果があったと考えられる．このことから，学習が必要な問題を学習することができているため，点数が向上したと考えられる．

5.2.3 アンケート結果

アンケート結果の一部を表 3 に示す．6 件法を用いて，6 が「とてもそう思う」，5 が「そう思う」，4 が「どちらかといえばそう思う」，3 が「どちらかといえばそう思わない」，1 が「まったくそう思わない」として平均を集計した．アンケートの結果から，「補助問題の提示機能はプログラミング学習に有効であると思いますか？」「問題マップの表示機能はプログラミング学習に有効であると思いますか？」の項目において，実験群から肯定的な評価をえられた．また，「自分で問題を選択する機能はプログラミング学習に有効であると思いますか？」の項目において，統制群からは肯定的な評価をえられなかった．そして，「本システムによる学習はプログラミング学習において有効だと思いますか？」の項目において，統制群と比べて実験群に肯定的な評価をえることができた．

また，「本システムがプログラミング学習の妨げになる点 等，悪かった点について記述してください.」について，実験群の学習者から「コンパイルエラーの表示のみではどこで間違えたのか確認するのが難しかった」という意見があった．

表 2 C3,C7 の問題の移行

	学習者による問題の移行
C3	P0→P1→P2→P3→……→P16→P17→P0
C7	P0→P1→P17→P5→P4→P3→P2→P1→P12→P11→P13→P12→P11→P12→P11→P12→P11

表 3 アンケートの結果

質問内容	実験群	統制群
システムで提示された補助問題ともとの問題との関係を意識しましたか？	4.5	-
システムから提示された補助問題は問題の解き方を理解するヒントになったと思いますか？	4.1	-
補助問題の提示機能はプログラミング学習に有効であると思いますか？	4.9	-
自分で選択した問題ともとの問題との関係を意識しましたか？	-	3.0
自分で選択した問題はもとの問題の解き方を理解するヒントになったと思いますか？	-	2.7
自分で問題を選択する機能はプログラミング学習に有効であると思いますか？	-	2.3
システムで提示された問題マップは問題を解くヒントになったと思いますか？	3.7	-
問題マップの表示機能はプログラミング学習に有効であると思いますか？	4.3	-
本システムによる学習はプログラミング学習において有効だと思いますか？	4.4	3.2

5.3 考察

テストの結果から実験群，統制群ともに点数の伸びがみられた．このことから，プログラミング学習における補助問題による学習効果が示唆された．そして，実験群の方が統制群と比べ，効果量が高いことが示唆された．また，統制群ではテストの点数の伸びにばらつきがみられた．そこで，事前・事後テストで点数が向上した統制群の被験者 C3，C7 の 2 名のログ分析を行った．その結果，C7 は自身で選択した問題と実験群のシステムで選択する問題の一致している箇所が多かった．そのため，補助問題を適切に選択できる学習者については学習効果があったと考える．また，C3 は問題選択の一致はないが，すべての問題に着手し，正解することができていることが確認できた．そのため，学習が必要な問題を網羅していたため，点数が向上したと考える．点数が向上した 2 名を除いた統制群は平均点が事前テストで 2.00 点，事後テストで 2.00 点であり，点数の伸びがみられなかった．そのため，問題を適切に選択できなかった学習者は学習効果が少なかったと考える．一方実験群では標準偏差が低く，事前・事後テストの点数では一定の伸びがみられた．この結果から提案システムの学習効果が示唆された．

また，アンケート結果から，実験群の学習者の方が統制群の学習者と比べ，システムでの学習はプログラミング学習に有効であると考えていることがわかった．しかし，解答内容によって実行結果を表示できない場合に表示される「コンパイルエラー」だけではなぜ誤答しているのかがわからず，解答を考える妨げになっていることがわかった．今後はエラー内容を表示することで，実行結果を表示できない原因を示す必要があると考えられる．

6. おわりに

プログラミング初学者にとって，複雑なプログラミング課題を理解することは難しく，問題解決に行き詰まってしまうと考えられる．初学者がそのような複雑な課題を理解するためには，学習者の行き詰まりを解消する必要がある．学習者の行き詰まりを解消する手法の 1 つとして補助問題の提示があげられる．

先行研究では，力学の作図問題において，学習者が行き詰まった場合にもとの問題を簡単化した補助問題を誤答箇所に応じて提示するシステムを開発した．システムにより誤答箇所に対応した補助問題を提示することで，力学の作図問題における学習者の問題解決の行き詰まりを解消する支援を提案した．

そこで本研究では，学習者がプログラミング課題で問題解決に行き詰まった際，誤り箇所を分析し誤答箇所に応じた補助問題を提示する．それにより，個々の学習者の問題解決の行き詰まりを解消し，複雑なプログラミング課題の理解を支援するシステムを開発した．

結果，プログラミング学習において補助問題による一定の学習効果が示唆された．そして，提案システムを用いることで，個々の学習者に適切な問題を選択することができた．

今後の課題として，実行結果を表示できない解答をした場合にフィードバックとしてエラーの内容を提示する機能の実装があげられる．

謝辞

本研究の一部は JSPS 科研費 JP22K12322，JP21H03565，JP20H01730 の助成による．

参 考 文 献

(1) 平嶋宗, 東正造, 柏原昭博, 豊田純一: “補助問題の定式化”, 人工知能学会誌, Vol.10, No.3, pp.413-420, (1995)

(2) 相川野々香, 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: “力学を対象とした Error-based Simulation における行き詰まりの解消を指向した補助問題の提示システム”, 電子情報通信学会論文誌 D, Vol.J106-D, No.2, in press, (2023)

(3) 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: “プログラミングの構造的理解を指向した部品の段階的拡張手法の提案と支援システムの開発・評価”, 教育システム情報学会誌, Vol.36, No.3, pp.190-202, (2019)

(4) 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: “プログラミング課題を構造化することによる理解状態に適応した課題提示方法の検討”, 人工知能学会第 83 回先進的学習科学と工学研究会, pp.35-40, (2018)

(5) 藤島優希, 古池謙人, 東本崇仁: “知識の再利用性向上を目的としたプログラミング学習における理解状態に基づく適応的フィードバックの開発と評価”, 第 91 回先進的学習科学と工学研究会, pp.60-65, (2021)